



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Εργαστήριο Βιομηχανικής και Ενεργειακής Οικονομίας
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ - ΤΟΜΕΑΣ ΙΙ

**ΠΡΟΣΑΡΜΟΣΤΙΚΑ ΜΑΘΗΣΙΑΚΑ ΠΑΙΓΝΙΑ ΚΑΤΑ NASH
ΣΕ ΑΠΕΛΕΥΘΕΡΩΜΕΝΕΣ ΑΓΟΡΕΣ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ**

Adaptive Learning Nash Games in Deregulated Power Markets

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΧΡΗΣΤΟΣ Κ. ΣΚΟΥΛΙΔΑΣ

Αθήνα, Ιανουάριος 2017

Η έγκριση της διδακτορικής διατριβής από την Ανωτάτη Σχολή Χημικών Μηχανικών του Ε.Μ.Πολυτεχνείου δεν υποδηλώνει αποδοχή των γνώμων του συγγραφέα. (Ν. 5343/1932, Άρθρο 202).

Στον αείμνηστο Παιδάγωγό μου Παιδαγωγικό

Στα παιδιά μου
Κωνσταντίνα και Γιώργο

ΠΡΟΛΟΓΟΣ

Στον σύντομο αυτό πρόλογο θα ήθελα να ευχαριστήσω όλους εκείνους που βοήθησαν στην ολοκλήρωση της παρούσας διατριβής συμβάλλοντας ο καθένας ξεχωριστά με τον τρόπο του. Και πρώτον απ' όλους τον αείμνηστο καθηγητή και δάσκαλο *Λευτέρη Παπαγιαννάκη* χάριν στις προσπάθειες και τη συνδρομή του οποίου ξεκίνησε το όλο εγχείρημα. Την επιβλέπουσα της συμβουλευτικής επιτροπής καθηγήτρια *κα.Δανάη Διακουλάκη* της οποίας η ανεξάντλητη επιμονή (και την ευχαριστώ ιδιαίτερα γ' αυτό) συνέβαλε καθοριστικά στην ολοκλήρωση της διατριβής. Τον καθηγητή και φίλο πλέον, *κ.Γιώργο Παπαβασιλόπουλο* για την καθοδήγησή του, τις ιδέες, τη συνδρομή και τη στήριξη που μου προσέφερε κατά τη διάρκεια της έρευνας αλλά και για τις ατέρμονες φιλοσοφικές συζητήσεις επί των παιγνίων και όχι μόνο. Τον καθηγητή *κ.Σπύρο Λιούκα* για τη συνδρομή του και την υπομονετική του διάθεση. Τον καθηγητή *κ.Κώστα Βουρνά* για την πολύτιμη βοήθειά του και τις παρατηρήσεις του στις εργασίες που δημοσιεύσαμε από κοινού. Τον καθηγητή *κ.Γιάννη Καλογήρου* για τη βοήθεια του σε διάφορες φάσεις της εκπόνησης της διατριβής. Τον φίλο και συνεργάτη *κ.Στέφανο Τσινόπουλο* για την σημαντική βοήθειά του στα θέματα του κώδικα προγραμματισμού. Τον *κ.Παναγιώτη Κοντογιώργο* για τη συνδρομή του στο θέμα της σύγχρονης βιβλιογραφίας.

Τέλος, θα ήθελα να αναφερθεί ότι τμήμα της επιστημονικής έρευνας που πραγματοποιήθηκε στο πλαίσιο της παρούσης Διδακτορικής Διατριβής χρηματοδοτήθηκε:

- από την ΓΓΕΤ στο πλαίσιο του Ερευνητικού Προγράμματος ΠΑΒΕΤ 2000 με συμμετέχοντες την εταιρεία 01 ΠΛΗΡΟΦΟΡΙΚΗ Α.Ε και το ΕΠΙΣΕΥ του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Εθνικού Μετσόβιου Πολυτεχνείου.
- από το Ευρωπαϊκό Ερευνητικό Πρόγραμμα του 6^{ου} Κοινοτικού Πλαισίου Στήριξης, EU-DEEP μέσω του ΕΠΙΣΕΥ του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή	9
2. Συνοπτική Περιγραφή του Έργου	10
2.1. Αντικείμενο και Στόχοι της Διατριβής	10
2.2. Διάρθρωση της Διδακτορικής Διατριβής	10
2.3. Πρωτοτυπία της Διδακτορικής Διατριβής	13
3. Η Θεωρητική Ανάλυση	14
3.1. Απελευθερωμένες Αγορές Ηλεκτρικής Ενέργειας	14
3.1.1. Γενική Επισκόπηση	14
3.1.2. Χρηματιστηριακές Αγορές Ηλεκτρικής Ενέργειας (Spot Markets)	18
3.1.3. Διμερείς Συναλλαγές Ηλεκτρικής Ενέργειας	21
3.1.4. Χρέωση Ισχύος	22
3.1.5. Χρηματοοικονομικές αγορές	24
3.1.6. Χρέωση Μεταφοράς και Διευθέτηση Συμφορήσεων	26
3.2. Θεωρία Παιγνίων	35
3.2.1. Σημείο Ισορροπίας κατά Nash και οι λύσεις Pareto	35
3.2.2. Παίγνιο Stackelberg	36
4. Το Μοντέλο Προσομοίωσης της Αγοράς Ηλεκτρικής Ενέργειας	39
4.1. Το Μοντέλο ενός Συστήματος Ηλεκτρικής Ενέργειας	39
4.1.1. Γενική Περιγραφή του Μοντέλου	39
4.1.2. Μέθοδοι Τιμολόγησης και Προσδιορισμός της Τιμής	45
4.1.3. Το πρόβλημα κατανομής των παραγωγών και ο αλγόριθμος επίλυσής του	48
4.1.4. Η Συνάρτηση της Ζήτησης	50
4.2. Το Μοντέλο Δύο Διασυνδεδεμένων Αγορών Ηλεκτρικής Ενέργειας	52
4.2.1. Γενική Περιγραφή του Μοντέλου	52
4.2.2. Περίπτωση υποβολής προσφοράς και στα δύο Συστήματα	53
4.2.3. Περίπτωση υποβολής προσφοράς μόνο στο ένα Σύστημα	57
4.2.4. Περιορισμός στη δυναμικότητας διασύνδεσης	57
4.3. Το Μοντέλο Αγοράς με Προσφορές Ζήτησης και Κατανεμημένη Παραγωγή Ενέργειας	60
4.3.1. Γενική Περιγραφή του Μοντέλου	60
4.3.2. Προσφορές Κατανεμημένης Παραγωγής Ενέργειας (DER generation)	64
4.3.3. Προσφορές Ζήτησης (Demand Side Bidding) με Περικοπή Φορτίου και χρήση Κατανεμημένης Παραγωγής Ενέργειας (DER)	67
5. Το Λογισμικό Προσομοίωσης	69

6. Παιγνία και Αποτελέσματα.....	75
6.1. Παιγνία σε ένα Ηλεκτρικό Σύστημα.....	75
6.1.1. Παιγνίο 1: Οριακή Τιμή Συστήματος και Αριθμός Παικτών με τετραγωνική συνάρτηση οριακού κόστους σε Uniform Pricing τιμολόγηση.....	75
6.1.2. Παιγνίο 2: Οριακή Τιμή Συστήματος και Αριθμός Παικτών με γραμμική συνάρτηση οριακού κόστους και διαφορετικό βήμα προσαρμογής σε Uniform Pricing τιμολόγηση.....	81
6.1.3. Παιγνίο 3: Οριακή Τιμή Συστήματος για διαφορετικό Ανώτατο Όριο Προσφορών (Price Cap) και μεταβλητό βήμα προσαρμογής e	84
6.1.4. Παιγνίο 4: Σύγκριση Uniform Pricing και Pay-As-Bid τιμολόγησης.....	88
6.2. Παιγνία σε δύο Διασυνδεδεμένα Ηλεκτρικά Συστήματα.....	92
6.3. Παιγνία σε Αγορά με Προσφορές Ζήτησης και Κατανεμημένη Παραγωγή Ενέργειας (DER)....	96
6.4. Σύγκριση της Θεωρητικής Λύσης του Παιγνίου Nash και των Αποτελεσμάτων του Μοντέλου	98
6.4.1. Παιγνίο με Uniform Pricing.....	98
6.4.2. Παιγνίο με Pay-As-Bid.....	103
7. Επισκόπηση του Έργου.....	107
7.1. Συμπεράσματα.....	107
7.2. Κυριότερα προβλήματα κατά την εκπόνηση της διατριβής.....	108
7.2.1. Το πρόβλημα της κατανομής των παραγωγών.....	108
7.2.2. Η Συνάρτησης Ζήτησης.....	108
7.2.3. Ο Προσαρμοστικός Μαθησιακός Αλγόριθμος.....	108
7.3. Αξιοποίηση και Μελλοντικές Επεκτάσεις της Έρευνας.....	110
7.4. Επιστημονικές Δημοσιεύσεις και Ανακοινώσεις.....	111
7.5. Βιβλιογραφία.....	112
7.6. Citations.....	115
Παράρτημα.....	117
Ο κώδικας του Λογισμικού EM Simulator.....	118-224

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.	Παράσταση των καμπυλών αντίδρασης δύο παικτών, του σημείου ισορροπίας κατά Nash και των λύσεων Pareto σε δισδιάστατο διάγραμμα καμπυλών ισοδύναμου αποτελέσματος.	36
Σχήμα 2.	Παίγνιο Stackelberg όπου και οι δύο παίκτες έχουν συμφέρον να αναλάβουν το ρόλο του ακολούθου.	37
Σχήμα 3.	Παίγνιο Stackelberg όπου και οι δύο παίκτες έχουν συμφέρον (α) να αναλάβει συγκεκριμένος παίκτης το ρόλο του αρχηγού (β) να αναλάβουν οι ίδιοι το ρόλο του αρχηγού.	37
Σχήμα 4.	Προσδιορισμός της Οριακής Τιμής και των μεριδίων αγοράς σε ένα Σύστημα με δύο Παραγωγούς με τετραγωνικό Οριακό Κόστος και σύστημα τιμολόγησης Uniform Pricing.	40
Σχήμα 5.	Προσδιορισμός Κέρδους για τον Παραγωγό στο Σύστημα με Uniform Pricing Τιμολόγηση	41
Σχήμα 6.α	Η Μαθησιακή Διαδικασία: Η Πρώτη Προσφορά.	43
Σχήμα 6.β	Η Μαθησιακή Διαδικασία: Τυχαία Επιλογή Δράσης για έναν από τους συντελεστές και διαμόρφωση της επόμενης προσφοράς.	43
Σχήμα 6.γ	Η Μαθησιακή Διαδικασία: Σύγκριση των αποτελεσμάτων J1 και J2, επαναπροσδιορισμός του στοχαστικού προφίλ συμπεριφοράς του συντελεστή με τη βοήθεια ενός βήματος θ και επιβράβευση ή τιμωρία της πιθανότητας της δράσης που επιλέχθηκε.	44
Σχήμα 6.δ	Η Μαθησιακή Διαδικασία: Νέα Προσφορά βάσει του προσαρμοσμένου προφίλ συμπεριφοράς.	44
Σχήμα 7.	Σχέση Συνολικού και Οριακού κόστους, συνάρτησης προσφοράς και Οριακής Τιμής Συστήματος Παραγωγού με γραμμικό οριακό κόστος σε αγορά ηλεκτρικής ενέργειας με Uniform Pricing.	46
Σχήμα 8.	Σχέση Συνολικού και Οριακού κόστους, συνάρτησης προσφοράς και Οριακής Τιμής Συστήματος Παραγωγού με γραμμικό οριακό κόστος σε αγορά ηλεκτρικής ενέργειας με Pay-As-Bid.	47
Σχήμα 9.	Το αντεστραμμένο διάγραμμα Προσφερόμενης Ισχύος-Τιμής όπου διακρίνονται οι περιοχές ασυνέχειας.	49
Σχήμα 10.	Η συνάρτηση Ζήτησης σε σχέση με την ανελαστική ζήτηση και το Price Cap.	50
Σχήμα 11.	Η επίλυση του προβλήματος της κατανομής των παραγωγών με γραμμική συνάρτηση ζήτησης.	51
Σχήμα 12.	Σχηματική απεικόνιση των εμπορικών συναλλαγών μεταξύ δύο διασυνδεδεμένων αγορών ηλεκτρικής ενέργειας σε σχέση με τις αντίστοιχες φυσικές ροές ηλεκτρικής ενέργειας.	52
Σχήμα 13.	Απεικόνιση προσφοράς παραγωγού σε δύο συστήματα και αντιστοίχων πωλούμενων ποσοτήτων και εσόδων με Uniform Pricing τιμολόγηση.	53
Σχήμα 14.	Σύγκριση παραγωγού που πωλεί το σύνολο της δυναμικότητάς του σε ένα και σε δύο Συστήματα.	54
Σχήμα 15.	Ο δείκτης αρχικής κατανομής α και ο συσχετισμός των μεγεθών σε παραγωγό που υποβάλλει προσφορά και στα δύο Συστήματα.	55
Σχήμα 16.α	Ο μετασχηματισμός της συνάρτησης προσφοράς στο Δευτερεύον Σύστημα.	56
Σχήμα 16.β	Ο μετασχηματισμός της συνάρτησης οριακού κόστους στο Δευτερεύον Σύστημα.	56
Σχήμα 17.	Ο Προσαρμοστικός αλγόριθμος μάθησης για την επιλογή του Συστήματος που θα υποβληθεί η προσφορά.	57
Σχήμα 18.	Πρώτη επίλυση Συστήματος με συμμετοχή αλλοδαπών παραγωγών και υφιστάμενο περιορισμό δυναμικότητας διασύνδεσης.	58
Σχήμα 19.	Καθορισμός των ποσοτήτων που πωλούν αλλοδαποί παραγωγοί λαμβάνοντας υπόψη τον περιορισμό της δυναμικότητας διασύνδεσης.	59
Σχήμα 20.	Τελική επίλυση Συστήματος με συμμετοχή αλλοδαπών παραγωγών και υφιστάμενο περιορισμό δυναμικότητας διασύνδεσης.	59
Σχήμα 21.	Προσφορές των παραγωγών $F_i(x)$ και προσδιορισμός των αντίστοιχων ποσοτήτων παραγωγής (x_i) που κατανέμονται	61
Σχήμα 22.	Συνολική Καμπύλη Προσφοράς σύμφωνα με τις επιμέρους Προσφορές των Παραγωγών και Προσδιορισμός της Οριακής Τιμής Συστήματος.	62
Σχήμα 23.	Γενική Επισκόπηση του Μοντέλου της Αγοράς με Προσφορές Ζήτησης και συμμετοχή Κατανεμημένης Παραγωγής Ενέργειας (DER) .	63
Σχήμα 24.	Συνάρτηση Κατανομής Πιθανότητας της Διαθεσιμότητας Παραγωγής από μικρά Αιολικά Πάρκα.	64

Σχήμα 25.	Η συνάρτηση κόστους του Εκπροσώπου Παραγωγής, η μορφή της προσφοράς του και ο προσδιορισμός του κέρδους του.	66
Σχήμα 26.	Προσδιορισμός της Οριακής Τιμής Συστήματος σε μια αγορά με Προσφορές Ζήτησης.	67
Σχήμα 27.	Η συνάρτηση κόστους του Εκπροσώπου Ζήτησης, η μορφή της προσφοράς του και ο προσδιορισμός του κέρδους του.	68
Σχήμα 28.	EM Simulator: Η αρχική οθόνη γενικών επιλογών του παιχνιδιού.	70
Σχήμα 29.α	EM Simulator: Η οθόνη εισαγωγής δεδομένων και γενικών χαρακτηριστικών του παιχνιδιού και της εποπτικής εικόνας παραγωγών που συμμετέχουν στο παιχνίδι.	71
Σχήμα 29.β	EM Simulator: Η οθόνη εισαγωγής δεδομένων και χαρακτηριστικών ενός παραγωγού.	71
Σχήμα 30.	EM Simulator: Οι οθόνες εισαγωγής δεδομένων και χαρακτηριστικών των Εκπροσώπων Παραγωγής/Ζήτησης που συμμετέχουν στο παιχνίδι.	72
Σχήμα 31.	EM Simulator: Η Οθόνη δυναμικής προσομοίωσης παιχνιδιού σε ένα Σύστημα με Προσφορές Ζήτησης.	73
Σχήμα 32.	EM Simulator: Εμφάνιση αποτελεσμάτων και διαγραμμάτων παιχνιδιού σε ένα Σύστημα με Προσφορές Ζήτησης.	73
Σχήμα 33.	EM Simulator: Η δομή του προγράμματος όπως φαίνεται στο περιβάλλον της Visual Fortran .	74
Σχήμα 34.	Πίνακας με τα χαρακτηριστικά μεγέθη των παραγωγών που συμμετέχουν στο παιχνίδι.	76
Σχήμα 35.	Η εξέλιξη της Οριακής Τιμής του Συστήματος και η κινητή Μέση Τιμή της σε παιχνίδι 10 Παραγωγών.	76
Σχήμα 36.	Σχέση Αριθμού Παραγωγών-Παικτών και τιμής σύγκλισης της ΟΤΣ στα αντίστοιχα παιχνίδια.	77
Σχήμα 37.	Σχέση τιμών σύγκλισης της Μ.Τ. των Συντελεστών A, B και C και των αντιστοίχων συντελεστών κόστους a, b, c για κάθε παραγωγό.	78
Σχήμα 38.α	Σχέση της τιμής σύγκλισης της Μ.Τ. του Συντελεστή A ως προς την τιμή του αντιστοίχου συντελεστή κόστους a, για τους Παραγωγούς 1,2 και 3 ανά είδος παιχνιδιού.	79
Σχήμα 38.β	Σχέση της τιμής σύγκλισης της Μ.Τ. του Συντελεστή B ως προς την τιμή του αντιστοίχου συντελεστή κόστους b, για τους Παραγωγούς 1,2 και 3 ανά είδος παιχνιδιού.	79
Σχήμα 38.γ	Σχέση της τιμής σύγκλισης της Μ.Τ. του Συντελεστή C ως προς την τιμή του αντιστοίχου συντελεστή κόστους c, για τους Παραγωγούς 1,2 και 3 ανά είδος παιχνιδιού.	80
Σχήμα 39.	Πίνακας με τα χαρακτηριστικά μεγέθη των Παραγωγών που συμμετέχουν στο παιχνίδι	82
Σχήμα 40.	Σχέση της Οριακής Τιμής του Συστήματος σε σχέση με τον αριθμό των παραγωγών που συμμετέχουν στην αγορά για διαφορετικές τιμές του βήματος προσαρμογής e.	83
Σχήμα 41.	Σχέση της επιπλέον ισχύος που αναγκάζεται να αγοράσει ο Διαχειριστής του Συστήματος για να καλύψει τη ζήτηση, και του αριθμού των παραγωγών που συμμετέχουν στην αγορά για διαφορετικές τιμές του βήματος προσαρμογής e.	83
Σχήμα 42.	Διαγράμματα ΟΤΣ σε Uniform Pricing παιχνίδια με διαφορετικό Price Cap και διαφορετική συμπεριφορά του βήματος προσαρμογής e.	85
Σχήμα 43.	Διαγράμματα Τιμής Ηλεκτρικής Ενέργειας σε Pay-As-Bid παιχνίδια με διαφορετικό Price Cap και διαφορετική συμπεριφορά του βήματος προσαρμογής e.	86
Σχήμα 44.	Διαγράμματα μέσων μεριδίων αγοράς των παικτών σε τύπο παιχνιδιού έξι παικτών και με τις δύο μεθόδους τιμολόγησης και μειούμενο βήμα προσαρμογής e.	87
Σχήμα 45.	Διάγραμμα σύγκρισης της τιμής της ηλεκτρικής ενέργειας συναρτήσει του αριθμού των παραγωγών σε Συστήματα με διαφορετική μέθοδο τιμολόγησης (Uniform Pricing και Pay-As-Bid).	89
Σχήμα 46.	Διάγραμμα σύγκρισης του μέσου προφίλ προσφορών σε Συστήματα με 2, 6 και 20 παραγωγούς και με διαφορετική μέθοδο τιμολόγησης: Uniform Pricing και Pay-As-Bid.	89
Σχήμα 47.	Διάγραμμα σύγκρισης της μέσης τιμής του σταθερού συντελεστή A της προσφοράς των παραγωγών σε Συστήματα με διαφορετική μέθοδο τιμολόγησης και σε συνάρτηση με τον αριθμό των συμμετεχόντων παραγωγών.	90
Σχήμα 48.	Διάγραμμα σύγκρισης της μέσης τιμής του γραμμικού συντελεστή B της προσφοράς των παραγωγών σε Συστήματα με διαφορετική μέθοδο τιμολόγησης και σε συνάρτηση με τον αριθμό των συμμετεχόντων παραγωγών.	90
Σχήμα 49.	Μέσος όρος πωλήσεων για κάθε παίκτη σε παιχνίδι 20 παραγωγών και σε Συστήματα με διαφορετική μέθοδο τιμολόγησης.	91

Σχήμα 50.	Συγκριτικός πίνακας της τιμής της ηλεκτρικής ενέργειας σε παίγνια με διαφορετικό αριθμό παραγωγών σε ένα και σε δύο ηλεκτρικά συστήματα χωρίς περιορισμό στη δυναμικότητα διασύνδεσής.	93
Σχήμα 51.	Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με κοινή μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και δυνατότητα διάσπασης της προσφοράς.	94
Σχήμα 52.	Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με διαφορετική μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και δυνατότητα διάσπασης της προσφοράς.	94
Σχήμα 53.	Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με κοινή μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και χωρίς δυνατότητα διάσπασης της προσφοράς.	95
Σχήμα 54.	Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με διαφορετική μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και χωρίς δυνατότητα διάσπασης της προσφοράς.	95
Σχήμα 55.	Μέση Κατανεμόμενη Ισχύς για τους συμμετέχοντες και στους τέσσερεις τύπους παιγνίων.	97
Σχήμα 56.	Εγκατεστημένη (Προσφερόμενη) Ισχύς και Μέση Κατανεμόμενη για όλες τις τεχνολογίες παραγωγής της Κατανεμημένης Παραγωγής Ενέργειας (DER generation) και της Αποκοπής Φορτίου.	97
Σχήμα 57.	Προσδιορισμός του βέλτιστου επιπέδου παραγωγής και αντιστοίχων προσφορών του Παικτή 1 για δεδομένη Προσφορά του Παικτή 2	101
Σχήμα 58.	Σύγκριση των σημείων ισορροπίας κατά Nash (αναλυτικά υπολογισμένων) και των σημείων σύγκλισης του προσαρμοστικού μαθησιακού μοντέλου στο Uniform Pricing Παίγνιο με 3 παίκτες .	102
Σχήμα 59.	Σχηματική απεικόνιση των καμπυλών της πρώτης αναλυτικής λύσης του παιγνίου δύο παικτών με Pay-As-Bid τιμολόγηση.	104
Σχήμα 60.	Σχηματική απεικόνιση των καμπυλών της δεύτερης αναλυτικής λύσης του παιγνίου δύο παικτών με Pay-As-Bid τιμολόγηση.	105
Σχήμα 61.	Σύγκριση των σημείων ισορροπίας κατά Nash (αναλυτικά υπολογισμένων) και των σημείων σύγκλισης του προσαρμοστικού μαθησιακού μοντέλου στο Pay-As-Bid Παίγνιο με 3 παίκτες .	106
Σχήμα 62.	Οι δράσεις της στοχαστικής κατανομής απόφασης για την προσαρμογή της γραμμικής συνάρτησης προσφοράς των παικτών με ταυτόχρονη μεταβολή και των δύο συντελεστών της συνάρτησης.	109

1. Εισαγωγή

Η παρούσα Διδακτορική Διατριβή με τίτλο «Προσαρμοστικά Μαθησιακά Παίγνια κατά Nash σε Απελευθερωμένες Αγορές Ηλεκτρικής Ενέργειας» στοχεύει στην ανάλυση της νέας πραγματικότητας που είχε αρχίσει να διαμορφώνεται την τελευταία δεκαπενταετία στον χώρο των ενεργειακών αγορών σε παγκόσμιο επίπεδο και πιο πρόσφατα στην Ελλάδα. Οι αναδιαρθρώσεις των ενεργειακών αγορών με οδηγό τις αγορές της ηλεκτρικής ενέργειας δημιούργησαν ένα εντελώς νέο τοπίο σ'αυτόν τον κρίσιμο, για την παγκόσμια οικονομία, τομέα που σε τίποτε δε θυμίζει την ως τώρα γνωστή και επί δεκαετίες παγιωμένη μορφή του. Η βίαιη σχεδόν είσοδος του ανταγωνισμού σε έναν τομέα που στις περισσότερες χώρες επί σειρά ετών χαρακτηριζόταν από την πιο ισχυρή ίσως μορφή μονοπωλιακής διάρθρωσης και νοοτροπίας, συμπίπτει χρονικά με την διαφαινόμενη τάση παγκοσμιοποίησης της οικονομίας και εκφράζεται μακροπρόθεσμα με την ανάγκη για ενοποίηση των ενεργειακών δικτύων και αγορών. Ωστόσο, όπως κάθε μεγάλη αλλαγή που συντελείται σε τόσο ευρεία κλίμακα, δημιουργεί νέα δεδομένα στο παγκόσμιο και τοπικό επιχειρηματικό περιβάλλον και χαρακτηρίζεται από μια αρχική περίοδο προσαρμογής, αναζήτησης και επαναπροσδιορισμού των αναγκών, των στόχων και των μέσων που τη δημιούργησαν. Σε αυτό το γενικότερο πλαίσιο, τα αποτελέσματα της παρούσας διατριβής φιλοδοξεί να συμβάλει στην καλύτερη κατανόηση του νέου αυτού περιβάλλοντος διερευνώντας, αναλύοντας και προτείνοντας νέα εργαλεία και μεθοδολογίες με στόχο να φανεί χρήσιμο τόσο σε επίπεδο σχεδιασμού στις ρυθμιστικές και διαχειριστικές αρχές όσο και σε καθαρά επιχειρηματικό επίπεδο στους παίκτες και επενδυτές που δραστηριοποιούνται στο χώρο αυτό.

2. Συνοπτική Περιγραφή του Έργου

2.1. Αντικείμενο και Στόχοι της Διατριβής

Αντικείμενο της διατριβής είναι η προσομοίωση και μελέτη απελευθερωμένων αγορών ηλεκτρικής ενέργειας με χρήση προσαρμοστικών στοχαστικών μαθησιακών παιγνίων κατά Nash. Το έργο επικεντρώνεται σε θέματα διάρθρωσης, κανόνων και συμπεριφοράς των αγορών ηλεκτρικής ενέργειας αλλά και των συμμετεχόντων σε αυτές, βασιζόμενο στην ανάλυση της κατάστασης μέσω της μελέτης ενός παιγνίου αποφάσεων, όπου συμμετέχουν πολλοί παίκτες με αντικρουόμενα συμφέροντα, έκαστος των οποίων ενδιαφέρεται για την επίτευξη του δικού του σκοπού. Σκοπός της έρευνας ήταν να αναπτύξει ένα *πρωτότυπο μοντέλο* βάσει του οποίου να προσομοιώσει και να μελετήσει απελευθερωμένες αγορές ηλεκτρικής ενέργειας που λειτουργούν κάτω από διαφορετικές συνθήκες και κανόνες και να παρατηρήσει φαινόμενα που αφορούν στη συμπεριφορά των συμμετεχόντων αλλά και της ίδιας της αγοράς όπως τιμές, ευστάθεια, επιπτώσεις εφαρμογής διαφορετικών συνθηκών και κανόνων λειτουργίας.

2.2. Διάρθρωση της Διδακτορικής Διατριβής

Στο **Κεφάλαιο 3** γίνεται μια **θεωρητική ανασκόπηση** αναφορικά με τις αναδιαρθρώσεις αγορών ηλεκτρικής ενέργειας όπου αυτές πραγματοποιήθηκαν και των αντίστοιχων μοντέλων αγορών (Ενότητα 3.1) καθώς και μια αναφορά σε γενικές έννοιες της **Θεωρίας Παιγνίων** (Ενότητα 3.2) και της εφαρμογής της σε θέματα μοντελοποίησης ενεργειακών αγορών.

Το **Κεφάλαιο 4** είναι αφιερωμένο στην αναλυτική **περιγραφή του μοντέλου** λειτουργίας της αγοράς ηλεκτρικής ενέργειας ως προσαρμοστικό μαθησιακό παίγνιο και των διαφόρων παραλλαγών του. Δίδεται η **γενική περιγραφή του μοντέλου** προσομοίωσης που αναπτύχθηκε για **μία** απελευθερωμένη αγορά ηλεκτρικής ενέργειας (Ενότητα 4.1) αποτελούμενη από παραγωγούς (παίκτες) με συγκεκριμένο κόστος παραγωγής (τετραγωνικής ή κυβικής μορφής) και έναν Ανεξάρτητο Διαχειριστή του Συστήματος στον οποίο οι παραγωγοί υποβάλλουν τις προσφορές τους υπό τη μορφή της καμπύλης του οριακού κόστους (γραμμική ή τετραγωνική αντίστοιχα). Ο Διαχειριστής αγοράζει ποσότητες ηλεκτρικής ενέργειας ξεκινώντας από τις οικονομικότερες προσφορές με στόχο την κάλυψη τη ζήτησης του Συστήματος και προσδιορίζει βάσει αυτών την οριακή τιμή του Συστήματος (ακριβότερη μονάδα ενέργειας που αγοράσθηκε). Οι παραγωγοί δεν γνωρίζουν τα κόστη, τις επιλογές και τα αποτελέσματα των αντιπάλων τους (μη συνεργατικό Nash παίγνιο) και χρησιμοποιούν ένα προσαρμοστικό στοχαστικό μαθησιακό μοντέλο για να αντισταθίσουν αυτή την έλλειψη πληροφορίας. Οι παραγωγοί αναπροσαρμόζουν τις προσφορές τους, αλλάζοντας κάθε φορά έναν από τους συντελεστές της καμπύλης προσφοράς, για συγκεκριμένο αριθμό επαναλήψεων. Τυχαία επιλεγμένες τιμές από το αντίστοιχο στοχαστικό προφίλ συμπεριφοράς του κάθε παραγωγού -το οποίο διαμορφώνεται σταδιακά, με την αξιολόγηση της επίπτωσης που είχε στο κέρδος του η τελευταία αναπροσαρμογή της προσφοράς του - καθορίζουν την επόμενη προσφορά. Η διαδικασία αυτή επαναλαμβάνεται μερικές χιλιάδες φορές μέχρις ότου η οριακή τιμή παρουσιάσει σύγκλιση οπότε και τερματίζεται το παίγνιο. Το ίδιο παίγνιο μπορεί να επαναληφθεί μερικές εκατοντάδες φορές προκειμένου να επιβεβαιωθεί ότι συγκλίνει στο ίδιο αποτέλεσμα (payoff). Το αποτέλεσμα κάθε τέτοιου παιγνίου, των χιλιάδων επαναλήψεων, προσπαθεί να προσεγγίσει αυτό που στην πραγματικότητα θα ήταν το αποτέλεσμα, για παράδειγμα, κατά τη δημοπρασία κάλυψης της ζήτησης μίας ώρας της ημέρας σε μια αγορά ηλεκτρικής ενέργειας με τους ίδιους παραγωγούς, συνθήκες, κόστη και περιορισμούς.

Σε πρώτη φάση εξετάζονται οι επιπτώσεις και η ευαισθησία του μοντέλου στις τεχνικές παραμέτρους του παιγνίου όπως:

- η δυνατότητα των παικτών να μεταβάλλουν ποσοτικά την προσφορά τους κάθε φορά (βήμα προσαρμογής), ο τύπος της μεταβολής (σταθερή ή μειούμενη) καθώς και η συχνότητα εναλλαγής των συντελεστών της προσφοράς τους που μεταβάλλονται.

- οι αρχικές συνθήκες του στοχαστικού μοντέλου απόφασης που χρησιμοποιείται για να ληφθεί η απόφαση του κάθε παίκτη και ο βαθμός προσαρμογής του κατά τη διάρκεια του παιχνιδιού.
- η ύπαρξη ή μη ανωτάτου ορίου επιτρεπόμενων προσφορών (Price Cap) και το ύψος αυτού.
- το πλήθος των συμμετεχόντων

Τα αποτελέσματα που εξετάζονται αφορούν στη διαμόρφωση της τιμής και στην ταχύτητα σύγκλισής της, των μεριδίων αγοράς και στη συμπεριφορά των συντελεστών της προσφοράς των παικτών

Επιπλέον παρουσιάζονται και περιγράφονται αναλυτικά (Ενότητα 4.1.2) οι διαφορετικές **μέθοδοι τιμολόγησης** που ενσωματώθηκαν στο μοντέλο προκειμένου να γίνουν συγκρίσεις στα αποτελέσματά τους τόσο στην αγορά όσο και στη συμπεριφορά των συμμετεχόντων. Συγκεκριμένα μοντελοποιούνται και συγκρίνονται οι ακόλουθοι μέθοδοι τιμολόγησης:

α) **Ενιαία Τιμολόγηση (Uniform Pricing)** όπου οι παραγωγοί πληρώνονται το σύνολο της παραγωγής τους που απορρόφησε ο Διαχειριστής στην τιμή της ακριβότερης μονάδας ενέργειας που αγοράσθηκε στο Σύστημα (Οριακή Τιμή του Συστήματος).

β) **Τιμολόγηση βάσει προσφοράς (Pay-As-Bid)** όπου οι παραγωγοί πληρώνονται για κάθε μονάδα ενέργειας σύμφωνα με την προσφορά που υπέβαλλαν.

Ενδιαφέροντα συμπεράσματα εξάγονται όχι μόνο από τη σύγκριση των δύο συστημάτων τιμολόγησης αλλά και από την παραμετροποίηση του Συστήματος σε σχέση με τον αριθμό των συμμετεχόντων παραγωγών, τα κόστη τους καθώς και τον τρόπο και την ταχύτητα που αντιδρούν και προσαρμόζουν τη συμπεριφορά τους στα μηνύματα που λαμβάνουν από την αγορά.

Ακολούθως παρουσιάζεται το πρόβλημα της κατανομής των παραγωγών και του προσδιορισμού της Οριακής Τιμής του Συστήματος και ο αλγόριθμος επίλυσής του (Ενότητα 4.1.3), καθώς και η περιγραφή του τρόπου εισαγωγής στο μοντέλο της συνάρτησης της Ζήτησης (Ενότητα 4.1.4).

Στο επόμενο στάδιο περιγράφεται η **μοντελοποίηση δύο διασυνδεδεμένων ηλεκτρικών αγορών-συστημάτων** (Ενότητα 4.2) τα οποία διενεργούν παράλληλα «πλειστηριασμούς» για την κάλυψη της ζήτησης με τη διαφορά ότι παραγωγοί του ενός συστήματος μπορούν να υποβάλλουν προσφορές και στο άλλο σύστημα. Εισάγεται στο μοντέλο η δυνατότητα επιλογής ώστε οι παραγωγοί να μπορούν (α) να προσφέρουν κάθε φορά το σύνολο της ποσότητας που παράγουν σε ένα μόνο σύστημα της επιλογής τους, είτε (β) να υποβάλουν προσφορά για ένα μόνο τμήμα της προσφοράς τους στο ένα σύστημα και το υπόλοιπο στο άλλο. Η επιλογή του συστήματος στην πρώτη περίπτωση και η κατανομή της προσφερόμενης ποσότητας στα δύο συστήματα στη δεύτερη περίπτωση, γίνεται με τη χρήση ενός δεύτερου προσαρμοστικού στοχαστικού μαθησιακού αλγορίθμου που προσομοιώνει τη διαδικασία της απόφασης. Εδώ μελετάται και συγκρίνεται η συμπεριφορά της ενιαίας αγοράς που αποτελούν τα διασυνδεδεμένα συστήματα υπό διαφορετικές συνθήκες κάθε φορά, όπως διαφορετικά συστήματα τιμολόγησης και διαφορετικό πλήθος παραγωγών, σε σχέση με το πώς θα λειτουργούσαν οι αγορές αυτές αν ήταν ανεξάρτητες. Επιπλέον με την εισαγωγή δυνατότητας περιορισμού στη δυναμικότητα διασύνδεσης εξετάζεται η συμπεριφορά των δύο αγορών και των συμμετεχόντων σε αυτές, τόσο στη θεωρητική περίπτωση της άπειρης δυναμικότητας όσο και σε αυτή της διασυνδεδεμένης λειτουργίας με περιορισμένη δυναμικότητα διασύνδεσης.

Στη συνέχεια παρουσιάζεται η **μοντελοποίηση μιας αγοράς ηλεκτρικής ενέργειας με μεταβαλλόμενη ζήτηση** (Ενότητα 4.3) όπου υπάρχει παράλληλα και υποβολή προσφορών από την πλευρά της ζήτησης (Demand Side Bidding) υπό μορφή αντίστοιχης αλλά αντίστροφης καμπύλης προσφοράς. Παράλληλα εισάγεται στο μοντέλο η δυνατότητα συγκεντρωτικών προσφορών Κατανεμημένης Παραγωγής (DER: Distributed Energy Resources) μέσω του «Εκπροσώπου» Παραγωγής ή Ζήτησης (Supply/Demand Aggregator) ο οποίος λειτουργεί ως εξισορροπιστής

της αγοράς. Από την πλευρά της Προσφοράς εισάγεται στο μοντέλο η οντότητα του Εκπροσώπου Παραγωγής ο οποίος διαθέτει ένα portfolio από μικρούς τοπικούς παραγωγούς (ΑΠΕ, συμπαραγωγή, κυψέλες καυσίμου) με συγκεκριμένα κόστη, δυναμικότητες και διαθεσιμότητα παραγωγής, στοχαστικά προσδιοριζόμενης για τις τεχνολογίες μη ελεγχόμενης διαθεσιμότητας όπως τα αιολικά και τα φωτοβολταϊκά πάρκα. Επιπλέον ορίζεται μια σειρά ιεράρχησης κατανομής για τις τεχνολογίες των μικροπαραγωγών που εκπροσωπεί ο Εκπρόσωπος Παραγωγής η οποία στηρίζεται αποκλειστικά στο κόστος παραγωγής της κάθε τεχνολογίας και καθορίζει ποιές από αυτές θα διαθέσουν την παραγωγή τους στο σύστημα και σε ποιο βαθμό (ποσότητα), ανάλογα με την συγκεντρωτική προσφορά του Εκπροσώπου, τη συνολική ποσότητα για την οποία αυτός κατανέμεται μετά το κλείσιμο της αγοράς και τη διαθεσιμότητα παραγωγής των τεχνολογιών παραγωγής. Από την πλευρά της Ζήτησης, ο Εκπρόσωπος Ζήτησης, ο οποίος διαθέτει ανάλογο μίγμα τεχνολογιών τοπικής παραγωγής (πλην κυψελών καυσίμου), υποβάλει οικονομική προσφορά βάσει της οποίας προσδιορίζει το τρόπο με τον οποίο είναι διατεθειμένος να καλύψει τοπικά τις ανάγκες της ζήτησης και να μειώσει έτσι τη συνολική ζήτηση η οποία σε διαφορετική περίπτωση θα καλυπτόταν μέσω των παραγωγών που συμμετέχουν στην αγορά ηλεκτρικής ενέργειας. Στην προσφορά του εκτός από τη δυναμικότητα των επιμέρους τεχνολογιών Κατανεμημένης Παραγωγής (DER) ενσωματώνει και προσφορές για Περικοπή Φορτίου από μεγάλους καταναλωτές που είναι διατεθειμένοι από κάποιο επίπεδο τιμών και πάνω να περιορίσουν την κατανάλωσή τους λαμβάνοντας αντίστοιχο όφελος. Ο Εκπρόσωπος Ζήτησης αποζημιώνεται στην οριακή τιμή του Συστήματος που προκύπτει, για την ποσότητα κατά την οποία απομείωσε τη συνολική ζήτηση ενώ ο Διαχειριστής του Συστήματος (και κατ'επέκταση οι τελικοί καταναλωτές ηλεκτρικής ενέργειας) ωφελείται κατά τη διαφορά του ποσού που θα αποζημιώνε τους παραγωγούς για την ίδια συνολική ποσότητα ενέργειας σε υψηλότερη ωστόσο οριακή τιμή. Η ανωτέρω μοντελοποίηση του παιγνίου πραγματοποιήθηκε προκειμένου να συγκριθεί η επίδρασή που έχει στην τιμή της ηλεκτρικής ενέργειας η επιλογή της αύξησης της εγκατεστημένης δυναμικότητας παραγωγής από την πλευρά της Προσφοράς με την προσθήκη ενός νέου συμβατικού παραγωγού σε σχέση: (α) με την εισαγωγή ενός Εκπροσώπου Κατανεμημένης Παραγωγής αντίστοιχης συνολικής δυναμικότητας και (β) με την επιλογή «ρύθμισης» της ζήτησης μέσω υποβολής προσφορών Ζήτησης, επίσης αντίστοιχης δυναμικότητας, από έναν Εκπρόσωπο Ζήτησης.

Στο **Κεφάλαιο 5** παρουσιάζεται συνοπτικά το **λογισμικό (EM Simulator)** που χρησιμοποιήθηκε για την υλοποίηση των ανωτέρω μοντελοποιήσεων και τη διεξαγωγή των παιγνίων στο επόμενο κεφάλαιο της διατριβής Το λογισμικό αναπτύχθηκε σε γλώσσα προγραμματισμού Visual Fortran 6.6, αποτελείται από 29.000 γραμμές πηγαίου κώδικα και παρέχει φιλικό στο χρήστη γραφικό περιβάλλον τόσο για τη δυνατότητα εισαγωγής των παραμέτρων των παιγνίων όσο και την δυναμική απεικόνιση (animation) της εξέλιξης των παιγνίων και των αποτελεσμάτων τους στα αντίστοιχα γραφήματα. Ο κώδικας του λογισμικού παρατίθεται στο Παράρτημα της παρούσης διατριβής.

Στο **Κεφάλαιο 6** γίνεται η **παρουσίαση των παιγνίων** που μελετήθηκαν, των **αποτελεσμάτων και των συμπερασμάτων** που προέκυψαν από αυτά. Συγκεκριμένα, παρουσιάζονται αρχικά τα παίγνια που διεξήχθησαν στο απλό μοντέλο του **ενός Συστήματος** ηλεκτρικής ενέργειας (Ενότητα 6.1).

- Το πρώτο παίγνιο (Ενότητα 6.1.1) το οποίο αφορά στη μοντελοποίηση μιας αγοράς ηλεκτρικής ενέργειας με Ενιαία Τιμολόγηση και παραγωγούς με τετραγωνικό οριακό κόστος, επικεντρώνεται στη σύγκλιση της οριακής τιμής σε σχέση με τον αριθμό συμμετεχόντων, αλλά και στη συμπεριφορά των παικτών εξετάζοντας την εξέλιξη και τη σύγκλιση των τιμών των συντελεστών των προσφορών τους σε σχέση με τις αντίστοιχες τιμές των συντελεστών της συνάρτησης του οριακού τους κόστους.
- Το δεύτερο παίγνιο (Ενότητα 6.1.2) μοντελοποιεί μιας αγοράς ηλεκτρικής ενέργειας με Ενιαία Τιμολόγηση και παραγωγούς με γραμμικό οριακό κόστος προκειμένου να μελετήσει την επίδραση του βήματος προσαρμογής του μαθησιακού αλγορίθμου στα αποτελέσματα του παιγνίου και κυρίως το επίπεδο των τιμών και την ταχύτητα σύγκλισης.

- Το επόμενο παίγνιο (Ενότητα 6.1.3) ως μια πιο πολύπλοκη παραλλαγή του προηγούμενου χρησιμοποιεί συγκεκριμένο σταθερό αριθμό παικτών και εξετάζει τη σύγκλιση της οριακής τιμής για διαφορετικά επίπεδα Ανωτάτου Ορίου Προσφορών, για δύο διαφορετικές μεθόδους τιμολόγησης και για σταθερό και μεταβαλλόμενο βήμα προσαρμογής (κατά τη διάρκεια του ίδιου παιγνίου). Επιπλέον πέραν της τιμής εξετάζονται και τα μερίδια αγοράς των παικτών στα οποία συγκλίνουν τα παίγνια καθώς και οι τιμές των συντελεστών προσφοράς των παικτών.
- Το τελευταίο παίγνιο (Ενότητα 6.1.4) της πρώτης ενότητας παιγνίων σε ένα Σύστημα επικεντρώνει στη εξονυχιστική σύγκριση των δύο μεθόδων τιμολόγησης. Συμμετέχουν παίκτες με γραμμικό οριακό κόστος και εξετάζονται η οριακή τιμή συστήματος, τα μερίδια αγοράς, οι τιμές των συντελεστών προσφοράς και το μέσο προφίλ προσφοράς τους σε παίγνια με διαφορετική μέθοδο τιμολόγησης και με διαφορετικό πλήθος συμμετεχόντων.

Στο παίγνιο των **δύο διασυνδεδεμένων αγορών** ηλεκτρικής ενέργειας (Ενότητα 6.2) εξετάζονται τα αποτελέσματα σύγκλισης της τιμής των δύο συστημάτων σε σχέση με αυτά ενός απομονωμένου συστήματος σε παίγνια με διαφορετικό αριθμό παικτών, για διαφορετικούς συνδυασμούς μεθόδων τιμολόγησης στα δυο συστήματα (κοινό και διαφορετικό) και για διαφορετικό επίπεδο δυναμικότητας διασύνδεσης μεταξύ των δύο συστημάτων.

Στο παίγνιο μιας αγοράς ηλεκτρικής ενέργειας με **προσφορές ζήτησης και χρήση κατανεμημένης παραγωγής ενέργειας** (Ενότητα 6.3) και συγκεκριμένο αριθμό παικτών εξετάζονται οι τρεις εναλλακτικές περιπτώσεις που αναφέρθηκαν ανωτέρω κατά την αντίστοιχη περιγραφή του μοντέλου της συγκεκριμένης αγοράς. Εκτός από την τιμή του συστήματος εξετάζονται και τα μέσα μερίδια αγοράς που κατανέμονται στους Εκπροσώπους Προσφοράς/Ζήτησης αλλά και στις επιμέρους τεχνολογίες παραγωγής που αυτοί εκπροσωπούν.

Τέλος, για κάποια παίγνια με μικρό αριθμό παικτών γίνεται αναλυτική επίλυση του παιγνίου Nash (Ενότητα 6.4) και ακολούθως σύγκριση των τιμών σύγκλισης που προκύπτουν από το προσαρμοστικό στοχαστικό παίγνιο και των πολλαπλών σημείων ισορροπίας κατά Nash.

Στο **Κεφάλαιο 7** συνοψίζονται τα σημαντικότερα συμπεράσματα της διατριβής, τα κυριότερα προβλήματα που αντιμετωπίστηκαν κατά την εκπόνησή της και αναφέρονται κάποιες περιοχές στις οποίες θα μπορούσε να προσανατολισθεί η μελλοντική έρευνα.

2.3. Πρωτοτυπία της Διδακτορικής Διατριβής

Η παρούσα διατριβή όταν ξεκίνησε είχε σαν σκοπό να μελετήσει τις απελευθερωμένες αγορές ηλεκτρικής ενέργειας, έναν νέο σχετικά τύπο αγορών όπου λίγη πρακτική εμπειρία υπήρχε, καθώς οι αγορές αυτές έκαναν τα πρώτα τους βήματα και στην ουσία ήταν υπό διαμόρφωση. Η μοντελοποίηση μιας τέτοιας αγοράς με κάποιον ικανοποιητικό τρόπο θα μπορούσε να επιτρέψει τη μελέτη του τρόπου λειτουργίας της και τη δοκιμή κάποιων κανόνων ή περιορισμών και ενδεχομένως να προβλέψει κάποια φαινόμενα και συμπεριφορές τόσο της ίδιας της αγοράς όσο και των συμμετεχόντων σε αυτή. Η θεωρία παιγνίων φαινόταν εκ πρώτης ένα επιστημονικό πεδίο βάσει του οποίου θα μπορούσε κανείς να επιτύχει μια ικανοποιητική προσέγγιση καθώς τα χαρακτηριστικά των αγορών αυτών (παίκτες με αντικρουόμενα συμφέροντα, ελλιπής πληροφόρηση για τα κόστη και τις κινήσεις των υπολοίπων) τις έκαναν να μοιάζουν με ένα μη συνεργατικό παίγνιο μηδενικού αθροίσματος με ατελή πληροφόρηση. Και ενώ η χρήση της θεωρίας παιγνίων για την μελέτη τέτοιων αγορών ήταν ελάχιστες, η συγκεκριμένη μοντελοποίηση και προσομοίωση μιας απελευθερωμένης αγοράς ηλεκτρικής ενέργειας ως προσαρμοστικό στοχαστικό μαθησιακό Nash παίγνιο [55-56], ήταν η πρώτη που περιγράφηκε και πραγματοποιήθηκε.

3. Η Θεωρητική Ανάλυση

3.1. Απελευθερωμένες Αγορές Ηλεκτρικής Ενέργειας

3.1.1. Γενική Επισκόπηση

Η θεωρητική ανάλυση των απελευθερωμένων αγορών εξετάζει την πορεία της απελευθέρωσης, στις περισσότερες χώρες που έχουν αποφασίσει να απελευθερώσουν τις αγορές ηλεκτρικής ενέργειας. Στις περισσότερες χώρες οι αγορές ηλεκτρικής ενέργειας έχουν απελευθερωθεί πλήρως, τόσο για τους βιομηχανικούς όσο και τους οικιακούς καταναλωτές. Ενώ η απελευθέρωση των αγορών ηλεκτρικής ενέργειας ξεκίνησε αρχικά στη δεκαετία του '90 στις Σκανδιναβικές χώρες, τη Μ.Βρετανία, τη Γερμανία, τη Νέα Ζηλανδία, καθώς και σε αρκετές πολιτείες των ΗΠΑ και της Αυστραλίας μέσα στην επόμενη δεκαετία ακολούθησαν όλες οι περισσότεροι από 500 εκατομμύρια άνθρωποι (και όλοι οι βιομηχανικοί καταναλωτές) στις αναπτυσσόμενες χώρες έχουν τη δυνατότητα να επιλέγουν τον προμηθευτή ηλεκτρικής ενέργειας.

Η ανάλυση που πραγματοποιήθηκε διαπραγματεύεται την εξέλιξη της ανάπτυξης του ανταγωνισμού και του δικαιώματος επιλογής προμηθευτή στις αγορές Ηλεκτρικής Ενέργειας. Κάνοντας μια επισκόπηση της διεθνούς εμπειρίας, περιγράφει τις κυριότερες προσεγγίσεις που αναπτύσσονται, αναφέρεται στα κύρια σημεία της αποτελεσματικής αναδιάρθρωσης των αγορών ηλεκτρικής ενέργειας και κάνει μια αποτίμηση της διαφαινόμενης προσέγγισης αυτής της αναδιάρθρωσης. Θεωρητικά, υπάρχουν πολλές διαφορετικές μέθοδοι με τις οποίες είναι δυνατόν να γίνει η αναδιάρθρωση μιας αγοράς ηλεκτρικής ενέργειας, ανάλογα με το ποιες δραστηριότητες της βιομηχανίας ηλεκτρισμού απελευθερώνονται (π.χ. παραγωγή), με ποιόν τρόπο ρυθμίζονται οι δραστηριότητες που δεν υπόκεινται σε καθεστώς απελευθέρωσης (π.χ. μεταφορά) και σε ποιους παίκτες επιτρέπεται να συμμετέχουν στις διαφορετικές αγορές. Ωστόσο, στην πράξη και στις περισσότερες περιπτώσεις, παρατηρείται όλο και περισσότερο μία σύγκλιση στον τρόπο προσέγγισης της αναδιάρθρωσης της αγοράς ηλεκτρικής ενέργειας, όπου αυτή επιχειρείται.

Οι αναδιρθρώσεις αγορών ηλεκτρικής ενέργειας που έχουν συντελεστεί τα τελευταία χρόνια φαίνεται να παρουσιάζουν τα ακόλουθα κοινά σημεία:

- Ταχύτατη εισαγωγή της δυνατότητας πλήρους επιλογής για τον καταναλωτή.
- Διαχωρισμός των δραστηριοτήτων της Παραγωγής, της Μεταφοράς, της Διανομής και της Προμήθειας (εμπορίας) ηλεκτρικής ενέργειας.
- Υποχρέωση για παροχή δυνατότητας αμερόληπτης πρόσβασης σε τρίτους στα δίκτυα Μεταφοράς και Διανομής
- Απελευθέρωση των συναλλαγών ηλεκτρικής ενέργειας έτσι ώστε αυτές να έχουν τη δυνατότητα να πραγματοποιούνται παράλληλα μέσω οργανωμένων χονδρεμπορικών αγορών («χρηματιστήρια ενέργειας») αλλά και διμερών συμβολαίων.

Ο συνδυασμός του πλήρους ανοίγματος της αγοράς, του διαχωρισμού των δραστηριοτήτων Μεταφοράς, της ρυθμιζόμενης πρόσβασης στο δίκτυο και της απελευθέρωσης των συναλλαγών ηλεκτρικής ενέργειας είναι γνωστός ως ελεύθερος ανταγωνισμός (retail competition). Στον ελεύθερο ανταγωνισμό οι συναλλαγές μεταξύ των παραγωγών, των τελικών καταναλωτών και ενός αριθμού πιθανών ενδιάμεσων, όπως οι προμηθευτές ή οι χρηματιστές ενέργειας, πραγματοποιούνται ελεύθερα, λαμβάνοντας ωστόσο υπόψη τους «φυσικούς» περιορισμούς που τίθενται από το υφιστάμενο δίκτυο. Από την πλευρά της ζήτησης, οι τελικοί καταναλωτές είναι

ελεύθεροι να επιλέξουν τον προμηθευτή τους και να διαπραγματευθούν τα συμβόλαιά τους. Από την πλευρά της προσφοράς οι παραγωγοί μπορούν να πωλούν την παραγόμενη ενέργεια σε οποιονδήποτε παράγοντα της αγοράς. Το μοντέλο του ελεύθερου ανταγωνισμού εφαρμόστηκε αρχικά στην απελευθέρωση της αγοράς ηλεκτρικής ενέργειας στη Φινλανδία, στη Νορβηγία, στη Σουηδία, στην Ισπανία, στις ΗΠΑ, στη Μ.Βρετανία και με μερικές διαφοροποιήσεις στην Αυστραλία, στη Δανία, στη Γερμανία και στη Νέα Ζηλανδία. Στη συνέχεια ακολούθησαν και οι υπόλοιπες δυτικές χώρες συμπεριλαμβανομένης και της Ελλάδας.

Η αναδιάρθρωση της βιομηχανίας ηλεκτρισμού αναμένεται να αυξήσει την αποδοτικότητά της. Ωστόσο, οι πιο σημαντικές επιδράσεις της αναμένεται να φανούν μακροπρόθεσμα, κυρίως ως αποτέλεσμα ορθολογικότερων επενδυτικών αποφάσεων. Στις περισσότερες χώρες ο ανταγωνισμός ξεκίνησε μόλις πρόσφατα, ώστε να είναι πολύ νωρίς ακόμη για να αξιολογηθεί η απόδοση των νέων αγορών σε όρους κόστους, τιμών και κοινωνικής ωφέλειας συνολικά. Ως προς τη βραχυπρόθεσμη προοπτική τους οι αναδιρθρώσεις έχουν ήδη αποδώσει τα αναμενόμενα οφέλη. Μεγάλες αυξήσεις παραγωγικότητας έχουν ήδη παρατηρηθεί σε μια σειρά περιπτώσεων, κύρια λόγω των ιδιωτικοποιήσεων και του επιχειρηματικού προσανατολισμού των εταιρειών δημόσιας ωφέλειας του τομέα του ηλεκτρισμού. Στις περισσότερες περιπτώσεις, και όπου δεν παρατηρήθηκαν τάσεις χειραγώγησης της αγοράς, οι τελικές τιμές του ηλεκτρισμού έχουν μειωθεί ή παραμένει σταθερές, ενώ οι τιμές «χονδρικής» έχουν διατηρηθεί σε χαμηλά επίπεδα (σε σχέση με το κόστος παραγωγής από νέες μονάδες). Ωστόσο, ως επακόλουθο του χαμηλού κόστους παραγωγής, οι τελικές τιμές του ηλεκτρισμού μειώθηκαν και σε πολλές χώρες που μέχρι πρότινος δεν είχαν προβεί σε αναδιάρθρωση της αγοράς τους, κάνοντας έτσι δύσκολη την ποσοτικοποίηση των πραγματικών επιπτώσεων του ανταγωνισμού των τιμών.

Η δομή της αγοράς είναι καθοριστικός παράγοντας διαμόρφωσης των τιμών στις νέες αγορές ηλεκτρισμού. Υψηλός βαθμός συγκέντρωσης των μέσων παραγωγής επέφερε ατελή ανταγωνισμό στη χονδρική αγορά (πρώτα στάδια της αναδιάρθρωσης στη Βρετανία και στην Ισπανία), ενώ πιο έντονος ανταγωνισμός παρατηρήθηκε σε λιγότερο συγκεντρωτικές αγορές (π.χ. στη αγορά των Σκανδιναβικών χωρών και τη Γερμανία). Το συμπέρασμα ήταν ότι έπρεπε να δοθεί έμφαση στην πολιτική ενίσχυσης ανταγωνιστικών δομών με παράλληλη αποδυνάμωση ή/και άνοιγμα των εθνικών αγορών στο διεθνές (ή περιφερειακό) εμπόριο και ανταγωνισμό καθώς η ρυθμιστική αναδιάρθρωση δεν αρκεί από μόνη της για να δημιουργήσει συνθήκες ανταγωνισμού.

Θέμα μείζονος σημασίας για την πορεία της αναδιάρθρωσης φαίνεται επίσης να είναι η κατανομή του κόστους και του οφέλους στους τελικούς καταναλωτές, τους επενδυτές και τα άλλα εμπλεκόμενα μέρη, όπως για παράδειγμα είναι οι φορολογούμενοι ή οι εργαζόμενοι στη βιομηχανία του ηλεκτρισμού. Η κατανομή αυτή έχει τεράστια επίπτωση στην κοινωνική και πολιτική αποδοχή της αναδιάρθρωσης, πέραν της επίπτωσής της στην αποδοτικότητα. Για παράδειγμα, σε κάποιες χώρες οι μειώσεις του κόστους και των τιμών που παρατηρήθηκαν σε διάφορες στάδια της αλυσίδας παραγωγής δεν ενσωματώθηκαν στις τιμές του τελικού καταναλωτή. Το γεγονός αυτό δημιούργησε αμφιβολίες για την αποτελεσματικότητα της αναδιάρθρωσης και ενθάρρυνε την εισαγωγή του πλήρους ανταγωνισμού και άλλων μέτρων με στόχο τη μείωση των τιμών. Επίσης, οι ιδιωτικοποιήσεις όπου αυτές εφαρμόστηκαν, καθώς και ο τρόπος μεταχείρισης των επενδύσεων που πραγματοποιήθηκαν πριν την αναδιάρθρωση και δεν είχαν ακόμη αποσβεσθεί (*stranded costs*), είχαν σοβαρή επίπτωση στην κατανομή του οφέλους της αναδιάρθρωσης.

Η μετάβαση από το παλιό στο νέο ρυθμιστικό πλαίσιο αποτέλεσε μια μεγάλη πρόκληση για τους διαμορφωτές της ενεργειακής πολιτικής. Η αβεβαιότητα σχετικά με το ρυθμιστικό πλαίσιο κατά τη διάρκεια ή ακριβώς πριν τη μετάβαση είχε αρνητική επίπτωση στην επενδυτική δραστηριότητα καθώς οι επενδυτές καθυστερούσαν τις αποφάσεις τους έως ότου αποσαφηνισθεί πλήρως το νέο ρυθμιστικό πλαίσιο. Επιπλέον, η αξιοπιστία της τροφοδοσίας μπορεί να μειωθεί αν οι κανόνες και οι υποχρεώσεις των νέων παικτών δεν είναι σαφώς και με

συνεπώς καθορισμένοι. Η εμπειρία δείχνει ότι προβλήματα αξιοπιστίας εμφανίσθηκαν μόνο σποραδικά σε αναδιάρθρωμένες αγορές αλλά μόνο κατά ένα μέρος θα μπορούσαν να αποδοθούν στη μεταβατική περίοδο. Όπου πάντως ανέκυψαν τέτοιου είδους προβλήματα είχαν, τελικά, τεράστια αρνητική επίπτωση στην κοινή γνώμη αναφορικά με την αναδιάρθρωση. Έτσι, οι κυβερνήσεις κατά τη διάρκεια της μετάβασης παίζουν σημαντικό ρόλο στο να παρέχουν διαβεβαιώσεις σχετικά με τη δημιουργία των απαραίτητων ασφαλιστικών δικλείδων που εγγυώνται την αξιοπιστία του εγχειρήματος καθώς και την ελαχιστοποίηση των όποιων πιθανών κινδύνων.

Για να αναπτυχθεί ο ανταγωνισμός στις αγορές ηλεκτρικής ενέργειας, πρέπει οι μονοπωλιακές δραστηριότητες, όπως η λειτουργία του δικτύου μεταφοράς, να διαχωριστούν από τις δυνητικά ανταγωνιστικές δραστηριότητες (π.χ. παραγωγή). Ο κύριος στόχος του διαχωρισμού είναι η αποφυγή διακρίσεων στις δραστηριότητες εκείνες της βιομηχανίας ηλεκτρισμού που λειτουργούν ή μπορούν να λειτουργήσουν ανταγωνιστικά. Συνεπώς κάποιος βαθμός διαχωρισμού πρέπει να υφίσταται μεταξύ μεταφοράς και παραγωγής, παραγωγής και διανομής καθώς και μεταξύ διανομής και προμήθειας του τελικού καταναλωτή.

Ο διαχωρισμός του ιδιοκτησιακού καθεστώτος –διαφορετικοί ιδιοκτήτες σε διαφορετικές δραστηριότητες – παρέχει τη μεγαλύτερη δυνατότητα περιορισμού των διακρίσεων, καθώς εξαλείφει το κίνητρο που τις δημιουργεί. Ο λειτουργικός και ο λογιστικός διαχωρισμός έχουν περιορισμένη δυνατότητα στην αποτροπή των διακρίσεων καθώς το κίνητρο και μερικώς η δυνατότητα για διακρίσεις παραμένουν. Όταν εφαρμοσθούν, αυτού του είδους οι διαχωρισμοί απαιτείται σημαντική ρυθμιστική εποπτεία και σθεναρή εφαρμογή των κανόνων του ανταγωνισμού. Ο λειτουργικός διαχωρισμός της δραστηριότητας της μεταφοράς- διαχωρισμός της λειτουργίας του συστήματος από την κυριότητα του συστήματος – μπορεί να προσφέρει μια εναλλακτική λύση στο διαχωρισμό όταν η ιδιοκτησία του συστήματος μεταφοράς είναι μοιρασμένη μεταξύ διαφόρων μερών. Ωστόσο, η δημιουργία αποτελεσματικά ανεξάρτητων διαχειριστών του συστήματος μεταφοράς απαιτεί την ανάπτυξη σύνθετων και σε μεγάλο βαθμό μη δοκιμασμένων δομών διοίκησης.

Η προμήθεια του τελικού καταναλωτή έχει καθοριστική σημασία στην εφαρμογή του ανταγωνισμού προς όφελος των καταναλωτών, παρά τη μικρή της βαρύτητα στην αλυσίδα της βιομηχανίας ηλεκτρισμού. Η δυνατότητα των τελικών καταναλωτών να επιλέγουν προμηθευτή δημιουργεί μια σημαντική πίεση σε όλους τους παίκτες που μετέχουν στην αλυσίδα, γεγονός που είναι ουσιαστικά αδύνατο να υποκατασταθεί από οποιοδήποτε ρυθμιστική προσπάθεια. Η αξία της δυνατότητας επιλογής του καταναλωτή στο να πειθαρχεί του παίκτες της αγοράς έγκειται στο ότι τον εφοδιάζει με ένα διαπραγματευτικό εργαλείο. Το εργαλείο αυτό είναι αποτελεσματικό ακόμα και αν οι δυνατότητες που παρέχει (π.χ. αλλαγή προμηθευτή) δεν χρησιμοποιούνται συστηματικά. Ακόμα και αν η πίεση αυτή δεν είναι άμεσα ορατή (πολλοί καταναλωτές μπορεί να επιλέξουν να παραμείνουν στον ίδιο προμηθευτή), η έμμεση επίδραση στην τιμολογιακή διάρθρωση, στα επίπεδα των τιμών, στη διαφοροποίηση του τελικού προϊόντος και στις συνθήκες παροχής υπηρεσιών είναι δυνητικά ισχυρή.

Η εφαρμογή του ανταγωνισμού στην αγορά των τελικών καταναλωτών προϋποθέτει το διαχωρισμό της από τη διανομή, ένα κρίσιμο αριθμό προμηθευτών που αληθινά θα δώσει νόημα στη δυνατότητα επιλογής, καθώς και την ανάπτυξη του κατάλληλου τεχνικού πλαισίου που σχετίζεται με την καταμέτρηση της κατανάλωσης και την έκδοση λογαριασμών. Επιπρόσθετα, υπάρχουν τεχνικές, όπως η αποτύπωση του προφίλ κατανάλωσης (load profiling), που μειώνουν σημαντικά το κόστος της εφαρμογής του ανταγωνισμού στους μικρούς καταναλωτές.

Πέραν της οικονομικής αποδοτικότητας η ενεργειακή πολιτική προσπαθεί να επιτύχει και άλλους στόχους όπως η ασφάλεια του εφοδιασμού και η προστασία του περιβάλλοντος καθώς και να ικανοποιήσει διάφορους κοινωνικούς σκοπούς. Οι παλαιοί μηχανισμοί διοίκησης και ελέγχου, εφαρμοζόμενοι με τον παραδοσιακό τρόπο ούτε η καλύτερη προσέγγιση είναι, αλλά ούτε και είναι δυνατό να εφαρμοσθούν στο νέο αυτό πλαίσιο για την επίτευξη των στόχων αυτών. Σε μια αγορά ηλεκτρισμού που λειτουργεί με ανταγωνιστικό τρόπο, οι πολιτικές πρέπει να

εφαρμόζονται με ουδέτερους ως προς τον ανταγωνισμό τρόπους ώστε να μην γίνονται διακρίσεις μεταξύ των παικτών και να ελαχιστοποιούνται οι στρεβλώσεις της αγοράς. Η εύρεση και η εφαρμογή κατάλληλων μέσων για το σκοπό αυτό έχει αποδειχθεί ένα επίπονο και δύσκολο έργο.

Οι ρυθμιστικοί φορείς χρειάζεται να προσαρμοσθούν ώστε να ανταποκριθούν καλύτερα στις προκλήσεις που τίθενται από την αναδιάρθρωση των αγορών. Ειδικότερα, οι ρυθμιστές πρέπει να είναι ανεξάρτητοι από τους ρυθμιζόμενους, αλλιώς αλληλοσυγκρουόμενα συμφέροντα αναπόφευκτα προκύπτουν. Η ανεξαρτησία από τις κυβερνητικές αρχές ενδυναμώνει τη σταθερότητα της ρυθμιστικής πολιτικής, αποτρέπει τη χρησιμοποίηση του τομέα της ηλεκτρικής ενέργειας για την επίτευξη γενικότερων πολιτικών στόχων και, όπου το κράτος είναι ιδιοκτήτης των επιχειρήσεων κοινής ωφελείας, την αμερόληπτη μεταχείριση των παικτών της αγοράς. Ωστόσο, η ενδυνάμωση της υπευθυνότητας των ανεξάρτητων ρυθμιστικών φορέων είναι ένα δύσκολο ζήτημα και η επιλογή της προσέγγισής του μπορεί να εξαρτάται από ειδικότερα χαρακτηριστικά της χώρας- όπως για παράδειγμα, ο ρόλος της δικαστικής εξουσίας στη χώρα. Επιπλέον, ο ρόλος των αρχών ελέγχου του ανταγωνισμού είναι πραγματικά αναγκαίος από τη στιγμή που αναπτύσσεται ο ανταγωνισμός. Δύο τομείς όπου ο ανταγωνισμός επιδρά σημαντικά είναι ο έλεγχος των συγχωνεύσεων και η εξάλειψη των επιδοτήσεων.

Ταυτόχρονα με τα άλλα μεγάλα στρατηγικά ζητήματα, η ανάπτυξη των αγορών ηλεκτρικής ενέργειας απαιτεί την αντιμετώπιση και άλλων θεμάτων τεχνικής φύσεως που αφορούν στη ρύθμιση της μεταφοράς και της διανομής της ηλεκτρικής ενέργειας, καθώς και στην οργάνωση της αγοράς.

Μια σημαντική πλευρά της ρυθμιστικού πλαισίου που αφορά τα δίκτυα είναι η χρέωση μεταφοράς. Η κομβική χρέωση (nodal pricing) παρέχει κίνητρα για αποδοτική χρήση της ηλεκτροπαραγωγής και των εγκαταστάσεων μεταφοράς. Η εμπειρία δείχνει πως η κομβική χρέωση είναι αποδοτική αλλά πολυπλοκότερη στην εφαρμογή. Η ενιαία χρέωση (postage stamp pricing) γενικά δεν παρέχει επαρκή κίνητρα αποδοτικότητας. Ωστόσο, οι επακόλουθες ανεπάρκειες ενδέχεται να είναι περιορισμένες έκτασης σε συστήματα με ισχυρό δίκτυο και μεγάλα περιθώρια παραγωγικών αποθεμάτων καθώς η ενιαία χρέωση έχει το πλεονέκτημα της διαφάνειας και της ευκολίας εφαρμογής.

Δύο θέματα έχουν αποτελέσει αντικείμενο συζήτησης σε σχέση με την οργάνωση της «χονδρικής» αγοράς ηλεκτρικής ενέργειας. Το πρώτο ζήτημα αφορά στο ερώτημα για το αν η διάθεση της παραγωγής σε χρηματιστήρια συναλλαγών ηλεκτρικής ενέργειας (pools) θα πρέπει να είναι υποχρεωτική ή προαιρετική. Υπάρχει πλέον μια γενικότερη ομοφωνία στο ότι ο ανταγωνισμός μέσω διμερών εμπορικών συναλλαγών ηλεκτρικής ενέργειας πρέπει να αποτελεί ένα βασικό τμήμα μιας σύγχρονης και αποδοτικής αγοράς ηλεκτρισμού. Η μη υποχρεωτική συμμετοχή στα χρηματιστήρια συναλλαγών ηλεκτρικής ενέργειας όλο και περισσότερο φαίνεται να κυριαρχεί στο χώρο, ενώ η εφαρμογή του μοντέλου της υποχρεωτικής διάθεσης συνεχώς σταδιακά μειώνεται. Παρόλο που οι κανόνες τιμολόγησης και το σύστημα διαχείρισης της ηλεκτροπαραγωγής παρουσιάζουν σημαντικές διαφορές στα Ηλεκτρικά Συστήματα διαφόρων χωρών, αυτές οι διαφορές δε φαίνεται να έχουν κάποια σημαντική επίπτωση στη λειτουργία των χρηματιστηρίων συναλλαγών ηλεκτρισμού. Το δεύτερο θέμα, είναι το γεγονός ότι ενώ σε μερικές αγορές έχει καθιερωθεί η λεγόμενη χρέωση δυναμικότητας με στόχο την παροχή επιπλέον κινήτρων για επενδύσεις στην ηλεκτροπαραγωγή, σε κάποιες άλλες οι παραγωγοί πληρώνονται αποκλειστικά και μόνο για την ενέργεια που τελικώς παρέχουν στο σύστημα. Το γενικό συμπέρασμα είναι ότι στις περισσότερες των περιπτώσεων τα κίνητρα για επενδύσεις που παρέχει η αγορά επαρκούν στο να εξασφαλίσουν τις απαραίτητες επενδύσεις χωρίς να είναι αναγκαία η χρέωση δυναμικότητας.

Η αναδιάρθρωση της αγοράς ηλεκτρικής ενέργειας είναι στόχος που κινείται διαρκώς. Ένας σημαντικός παράγοντας ανάπτυξης είναι η ίδια η αγορά. Η βιομηχανία ηλεκτρισμού επεκτείνει τα όριά της με ταχείς ρυθμούς. Η ηλεκτροπαραγωγή όλο και περισσότερο ενοποιείται με εταιρείες φυσικού αερίου και πετρελαίου. Ταυτόχρονα, οι

δραστηριότητες της διανομής και της προμήθειας των τελικών καταναλωτών, σε τομείς όπου τα προϊόντα και οι υπηρεσίες όπως το φυσικό αέριο, ο ηλεκτρισμός, οι τηλεπικοινωνίες ή η ύδρευση που παρέχονται μέσω δικτύων, εμφανίζουν τάσεις ενοποίησης. Τα γεωγραφικά όρια της βιομηχανίας ηλεκτρισμού που κάποτε συνέπιπταν με τα γεωγραφικά όρια κρατών ή περιφερειών, μεταβάλλονται επίσης. Όλο και περισσότερο, τα ηλεκτρικά συστήματα ενοποιούνται μέσα σε μια ευρύτερη γεωγραφική περιοχή. Η απελευθέρωση επίσης ανοίγει το δρόμο για σημαντικές άμεσες επενδύσεις ξένων εταιρειών σε εθνικές αγορές. Το συμπέρασμα για την πολιτική της επέκτασης των ορίων της βιομηχανίας είναι ότι το ρυθμιστικό πλαίσιο θα πρέπει να εξετασθεί σε πολυτομεακό και πολυεθνικό επίπεδο. Αυτή η ροπή προς μεγαλύτερη εξάρτηση από το νόμο του ανταγωνισμού και τους κοινούς κανόνες του παγκόσμιου εμπορίου είναι μια διαδικασία η οποία έχει ήδη συντελεσθεί σε άλλους τομείς που έγινε άνοιγμα στον ανταγωνισμό.

Η αναδιάρθρωση των ενεργειακών αγορών που έχει συντελεσθεί στις περισσότερες δυτικές χώρες έγειρε μια σειρά θεμάτων τα οποία προέκυψαν ιδιαίτερα κατά τα πρώτα έτη της εφαρμογής του νέου πλαισίου. Το γεγονός αυτό κάνει πιο έντονη την ανάγκη για περαιτέρω ανάλυση των ενεργειακών αγορών και της συμπεριφοράς των συμμετεχόντων σε αυτές, στις συνθήκες απελευθέρωσης. Ιδιαίτερο ενδιαφέρον παρουσιάζουν θέματα που αφορούν τόσο τους υπάρχοντες ή μελλοντικούς παραγωγούς, (διαμόρφωση στρατηγικής, ελαχιστοποίηση ρίσκου) οι οποίοι καλούνται να δράσουν σε ένα νέο επιχειρηματικό περιβάλλον που χαρακτηρίζεται έντονα από ατελή πληροφορία και έλλειψη ιστορικών δεδομένων, όσο και τις ρυθμιστικές και διαχειριστικές αρχές (κανονιστικό πλαίσιο, μηχανισμοί διαμόρφωσης τιμών, εξασφάλιση υγιούς ανταγωνισμού) που καλούνται να διασφαλίσουν την ομαλή λειτουργία αυτής της νέας αγοράς αλλά και να καλύψουν τις ανάγκες της ζήτησης. Τα δεδομένα αυτά δημιουργούν έντονα την ανάγκη για περαιτέρω εξέταση και ανάλυση των μηχανισμών που διέπουν τη λειτουργία και τη διαμόρφωση συμπεριφοράς των συμμετεχόντων σε μια απελευθερωμένη αγορά ηλεκτρικής ενέργειας [1-4].

3.1.2. Χρηματιστηριακές Αγορές Ηλεκτρικής Ενέργειας (Spot Markets)

Η ανάπτυξη του ανταγωνισμού στη βιομηχανία ηλεκτρισμού επέφερε μεγάλη αύξηση των συναλλαγών ηλεκτρικής ενέργειας, καθώς και ανάπτυξη διαφόρων τύπων χρηματοοικονομικών συμβολαίων. Η φύση της ηλεκτρικής ενέργειας είναι τέτοια που δεν επιτρέπει την ανάπτυξη μια πραγματικής αγοράς τρέχουσας τιμής (spot market) συναλλαγών ηλεκτρικής ενέργειας, δηλαδή μιας αγοράς όπου πραγματοποιούνται άμεσες παραδόσεις ηλεκτρικής ενέργειας. Αντ'αυτού, οι συναλλαγές προγραμματίζονται από πριν όσον αφορά τη φυσική παράδοση (π.χ. μια μέρα, μια ώρα ή πέντε λεπτά πριν). Για το λόγο αυτό είναι πιθανό να προκύψουν κάποιες ανισορροπίες ανάμεσα στην προγραμματισμένη και την πραγματική προσφορά και ζήτηση, οι οποίες αντιμετωπίζονται σύμφωνα με ορισμένες προκαθορισμένες διαδικασίες που μπορεί να είναι ή όχι ανταγωνιστικές.

Αυτές οι αγορές ηλεκτρικής ενέργειας (τα λεγόμενα pool markets) αποτελούν υποκατάστατα μιας πραγματικής χρηματιστηριακής αγοράς. Στα περισσότερα υφιστάμενα pool, οι τιμές αγοράς και ο προγραμματισμένος εφοδιασμός καθορίζονται με διαδικασίες πλειστηριασμού, κάποια στιγμή πριν από την πραγματοποίηση της φυσικής παράδοσης. Οι τιμές πώλησης καθορίζονται προσθέτοντας στην τιμή αγοράς το κόστος της διευθέτησης αποκλίσεων προσφοράς-ζήτησης, τις επικουρικές υπηρεσίες, και ίσως κάποιες ακόμη χρεώσεις, όπως οι χρεώσεις διαθεσιμότητας ισχύος. Από τη στιγμή που οι τιμές καθορίζονται βάσει προγραμματισμένης προσφοράς και ζήτησης, η αγορά ονομάζεται *ex ante pool*, ενώ όταν οι τιμές καθορίζονται εκ των υστέρων βάσει των πραγματικών χρονοδιαγραμμάτων του παραγωγού και βάσει της ζήτησης ονομάζεται *ex post pool*. Σε ένα *ex post pool*, οι τιμές αγοράς και πώλησης συμπίπτουν. Η υφιστάμενη τεχνολογία μειώνει κατά πολύ το χρόνο που χρειάζεται για να επιτευχθεί η εξισορρόπηση του συστήματος – χωρίς όμως να εκμηδενίζει μια τέτοια ανάγκη.

Ο ιδανικός τρόπος λειτουργία μιας χρηματιστηριακής αγοράς ηλεκτρικής ενέργειας

Σε μία αγορά ηλεκτρικής ενέργειας, οι τιμές χρησιμοποιούνται για να συντονίσουν τις αποφάσεις τόσο των παραγωγών όσο και των αγοραστών ηλεκτρικής ενέργειας, ώστε η προσφορά να ισούται με τη ζήτηση, και για να εξασφαλιστεί ότι αυτές οι αποφάσεις είναι εφικτές δεδομένων των φυσικών περιορισμών του συστήματος.

Οι τιμές spot για την ηλεκτρική ενέργεια θα πρέπει να καθορίζονται για κάθε κόμβο του δικτύου, ενώ στην ανάλυση πρέπει να ληφθεί υπόψη ένας αριθμός περιπτώσεων με αύξοντα βαθμό πολυπλοκότητας. Στη συνέχεια παρατίθεται μια σειρά τέτοιων περιπτώσεων:

α) Απλή Περίπτωση

Στην απλούστερη των περιπτώσεων, όπου η ισχύς παραγωγής και μεταφοράς είναι αρκετή για να καλύψει τη ζήτηση και οι απώλειες μεταφοράς παραβλέπονται, θα ισχύει μία και μόνη χρέωση ηλεκτρικής ενέργειας για κάθε χρονική περίοδο:

Τιμή ενέργειας (T_1) = Οριακό κόστος της ακριβότερης εν λειτουργία μονάδας

β) Στην τιμή T_1 παρατηρείται έλλειμμα ισχύος.

Αν με τον καθορισμό της τιμής στο οριακό κόστος της ακριβότερης διαθέσιμης μονάδας διαπιστωθεί έλλειμμα ισχύος, ο κανόνας της προηγούμενης περίπτωσης δεν μπορεί να εφαρμοστεί. Η τιμή της ενέργειας πρέπει, στην περίπτωση αυτή, να αυξηθεί προκειμένου να μειωθεί η ζήτηση. Η αύξηση τιμής ΔT , που είναι απαραίτητη για να εξομοιωθεί η ζήτηση με τη διαθέσιμη ισχύ παραγωγής, είναι η διαφορά ανάμεσα στο οριακό κόστος παραγωγής και το οριακό κέρδος κατανάλωσης. Συνεπώς, η χρέωση της ενέργειας διαμορφώνεται ως εξής:

Τιμή ενέργειας (T_2) = $T_1 + \Delta T$

Το μέγεθος ΔT παρέχει κίνητρα για επενδύσεις στην παραγωγή.

γ) Ύπαρξη απωλειών μεταφοράς

Σε αυτή την περίπτωση η χρέωση της ενέργειας (είτε αυτή είναι η T_1 είτε η T_2) θα πρέπει να αυξηθεί με έναν συντελεστή ($1 + \text{Οριακή Απώλεια}$) σε κάθε κόμβο, αντικατοπτρίζοντας έτσι την ανάγκη να παραχθεί ($1 + \text{Οριακή Απώλεια}$) kWh, όταν μιλάμε για προμήθεια 1 kWh. Συνεπώς:

Τιμή ενέργειας (T_3) = $T_2 (1 + \text{Οριακές Απώλειες})$

δ) Ύπαρξη περιορισμών μεταφοράς

Η τιμή σε κόμβους όπου παρατηρείται συμφόρηση, πρέπει να αυξηθεί για να περιοριστεί η κατανάλωση. Η τιμή στα υπό συμφόρηση σημεία έγχυσης ηλεκτρικής ενέργειας του δικτύου πρέπει να μειωθεί για να περιοριστεί η κατανάλωση. Το μέγεθος της προσαρμογής της τιμής, γνωστό ως «σκιώδης τιμή» του περιορισμού, είναι τέτοιο που σε κανένα σημείο του δικτύου η προσφορά δεν ξεπερνά την ισχύ μεταφοράς. Οι τιμές που προκύπτουν είναι οι κομβικές χρεώσεις ηλεκτρικής ενέργειας συμπεριλαμβάνοντας όλες τις προηγούμενες περιπτώσεις:

Κομβική χρέωση = $T_3 + \text{Σκιώδης τιμή του περιορισμού μεταφοράς στον αντίστοιχο κόμβο}$

Εξαιτίας του τρόπου που δομούνται οι τιμές χονδρικής, η προσφορά είναι ίση με τη ζήτηση σε κάθε κόμβο και χρονική περίοδο εξασφαλίζοντας έτσι την ομαλή λειτουργία του Συστήματος. Γενικά ισχύει ότι οι παραγωγοί θα παράξουν ενέργεια εφόσον το οριακό τους κόστος δεν υπερβαίνει την τιμή της ενέργειας στο σημείο έγχυσης της ηλεκτρικής ενέργειας, αλλά όχι και το αντίστροφο. Οι καταναλωτές θα αγοράσουν ηλεκτρική ενέργεια μέχρι του σημείου όπου η τιμή της είναι ίση με το οριακό τους κέρδος [5-7].

Διεθνής Επισκόπηση των Χρηματιστηρίων Συναλλαγών Ηλεκτρικής Ενέργειας

Από μια γενική επισκόπηση της διεθνούς εμπειρίας, μπορούμε να διαπιστώσουμε ότι οι μηχανισμοί διαμόρφωσης των τιμών και προγραμματισμού λειτουργίας των μονάδων ποικίλουν σε μεγάλο βαθμό. , η διαδικασία υποβολής προσφορών των μονάδων παραγωγής μπορεί να είναι επαναληπτική ή ενός «κτυπήματος» ενώ μπορεί να συνυπάρχει ή όχι και με παράπλευρες προσφορές ζήτησης. Οι προσφορές μπορεί να είναι δεσμευτικές ή μεταβλητές, μπορεί επίσης να είναι απλές, να περιέχουν δηλαδή μόνο μία τιμή ανά kWh, ή μπορεί να περιλαμβάνουν διάφορους όρους. Οι τιμές μπορεί να καθορίζονται εκ των προτέρων ή εκ των υστέρων και να περιλαμβάνουν πληρωμές ισχύος στους παραγωγούς. Μπορεί να τεθούν ανώτατα όρια τιμών (price caps) ή άλλοι περιορισμοί κατά τη διαδικασία. Ακόμη, οι συναλλαγές μπορεί να διευθετηθούν με διάφορους τρόπους. Γενικά πάντως, τα χρηματιστήρια συναλλαγών ηλεκτρικής ενέργειας παρουσιάζουν διαφορές ως προς το βαθμό που επιτρέπεται η βελτιστοποίηση από πλευράς των παικτών – σε αντιδιαστολή με τη βελτιστοποίηση από πλευράς του χρηματιστηρίου συναλλαγών. Η εμπειρία ως προς τη σχετική αποτελεσματικότητα των διαφόρων κανόνων είναι ακόμη περιορισμένη, αναμένεται όμως να εμπλουτιστούν στο άμεσο μέλλον, καθώς αυξάνεται ραγδαία ο αριθμός των χρηματιστηρίων συναλλαγών ηλεκτρικής ενέργειας [8-14]. Υπάρχουν ωστόσο κάποια συμπεράσματα τα οποία με ασφάλεια μπορούν να διατυπωθούν ως κανόνες:

Έχει διαπιστωθεί ότι στις υποχρεωτικές spot αγορές ηλεκτρικής ενέργειας (mandatory pools) οι τιμές έχουν την τάση να εμφανίζονται πιο ευμετάβλητες από ότι στις αντίστοιχες προαιρετικές αγορές. Επίσης οι τιμές παρουσιάζουν μεγαλύτερη αστάθεια σε συστήματα που βασίζονται σε σταθμούς με συμβατικά καύσιμα (fossil fuel-based systems) από ότι σε συστήματα που βασίζονται σε υδροηλεκτρικούς σταθμούς (hydroelectric based systems). Επίσης, οι τιμές κινούνται σε χαμηλότερα επίπεδα, με πιο ασταθή ωστόσο τρόπο, όταν οι εταιρείες ηλεκτρικής ενέργειας είναι ιδιωτικές, απ' ό,τι όταν οι εταιρείες ανήκουν στο δημόσιο τομέα.

Στον επόμενο πίνακα παρουσιάζονται συνοπτικά τα κυριότερα χαρακτηριστικά για μερικά από τα παλαιότερα χρηματιστήρια συναλλαγών ηλεκτρικής ενέργειας. Ως «απλές προσφορές» νοούνται οι προσφορές που συνίστανται από απλά ζεύγη τιμής – ποσότητας χωρίς επιπρόσθετους όρους. Η τελευταία στήλη αναφέρεται στον αν κατά την κατανομή της παραγωγής λαμβάνοντας υπόψη τους περιορισμούς του δικτύου ή όχι.

Αγορά	Υποχρεωτική Συμμετοχή	Τιμολόγηση	Απλές Προσφορές	Προσφορές Ζήτησης	Πληρωμές Ισχύος	Κατανομή
Αυστραλίας	Ναι	Ex-post	Ναι	Ναι	Όχι	Μερικώς
ΗΠΑ (CalPX)	Όχι	Ex-ante	Ναι	Ναι	Όχι	Όχι
ΗΠΑ (PJM ISO)	Όχι	Ex-post	Όχι	Όχι	Ναι	Ναι
Ισπανίας	Όχι	Ex-ante	Όχι	Ναι	Ναι	Όχι
Μ.Βρετανίας	Όχι	Ex-ante	Ναι	Ναι	Όχι	Όχι
Ν.Ζηλανδίας	Όχι	Ex-post	Ναι	Ναι	Όχι	Ναι
NordPool	Όχι	Ex-ante	Ναι	Ναι	Όχι	Όχι

Πίνακας 1. Τα κυριότερα χαρακτηριστικά των πρώτων χρηματιστηρίων συναλλαγών ηλεκτρικής ενέργειας

3.1.3. Διμερείς Συναλλαγές Ηλεκτρικής Ενέργειας

Σε μια αναδιάρθρωμένη αγορά ηλεκτρικής ενέργειας, εμπορικές συναλλαγές μπορεί να λαμβάνουν χώρα και έξω από το πλαίσιο των οργανωμένων αγορών. Η σύναψη διμερών συμβάσεων αναμένεται να είναι αποτελεσματική εφόσον αποτελεί τον συνήθη – αν όχι μοναδικό – τρόπο με τον οποίο πραγματοποιούνται οι εμπορικές συναλλαγές σε πολλές αγορές. Οι διμερείς εμπορικές συναλλαγές είναι εκ των πραγμάτων πιο ευέλικτες από τις συγκεντρωτικές συναλλαγές στα οργανωμένα χρηματιστήρια ενέργειας, επειδή μπορούν και συνυπάρχουν με μια μη-υποχρεωτική οργανωμένη αγορά συναλλαγών. Μια τέτοια αγορά, με τη σειρά της, περιορίζει την ανησυχία για ενδεχόμενα φαινόμενα διακρίσεων και αποτελεί απαραίτητη προϋπόθεση για χρέωση και παροχή ασφάλειας και αξιοπιστίας της τροφοδοσίας, προσαρμοσμένων στις ανάγκες του κάθε καταναλωτή.

Στα συστήματα διμερών συναλλαγών, η λειτουργία της αγοράς και η διαχείριση του συστήματος συχνά ελέγχονται από ανεξάρτητους οργανισμούς. Ο διαχειριστής του συστήματος αναλαμβάνει ως επί το πλείστον το έργο του τεχνικού συντονισμού με στόχο την εξισορρόπηση του συστήματος, συμπεριλαμβανομένης και της κατανομής. Το χρηματιστήριο συναλλαγών ηλεκτρικής ενέργειας οργανώνει τις εμπορικές συναλλαγές μεταξύ των συμμετεχόντων. Αυτή η διπλή δομή μειώνει τον κίνδυνο διακρίσεων απέναντι στους εμπλεκόμενους στις διμερείς συναλλαγές.

Η σύναψη διμερών συμβάσεων στις αγορές ηλεκτρικής ενέργειας έχει εντούτοις δεχτεί έντονη κριτική: η σύναψη διμερών συμβάσεων δεν συμβαδίζει με μια συγκεντρωτική βελτιστοποίηση της κατανομής καθώς αυτή δε γίνεται βάσει αξιολογικής ιεράρχησης των προσφορών ή του κόστους. Ωστόσο, όπως και στις περισσότερες αγορές, η απουσία κάποιου κεντρικού μέσου βελτιστοποίησης δεν αποκλείει την πιθανότητα μια αγορά να είναι αποδοτική. Το επιχείρημα αυτό ίσως αποτέλεσε τον κυριότερο λόγο για την υιοθέτηση του μοντέλου της υποχρεωτικής αγοράς συναλλαγών στα πρώιμα μοντέλα αναδιάρθρωσης (Αγγλία και Ουαλία) όμως, από τη στιγμή που η εμπειρία στην αγορά NordPool έχει δείξει ότι η αποκεντρωμένη κατανομή μπορεί να λειτουργήσει επίσης αποτελεσματικά.

Υπάρχουν ανησυχίες ότι οι τιμές της ηλεκτρικής ενέργειας για τους τελικούς καταναλωτές ίσως να μην είναι διαφανείς και οι τιμές pool μπορεί να διαστρεβλωθούν στην περίπτωση που ένα μεγάλο μέρος συμμετεχόντων στην αγορά συνάψουν διμερείς συμβάσεις. Ο καθορισμός ρυθμιζόμενων τιμολογίων για τους τελικούς καταναλωτές, όταν η τιμή χονδρικής της ηλεκτρικής ενέργειας δεν είναι σαφής, μπορεί να αποδειχθεί δύσκολη υπόθεση. Στόχος της αναδιάρθρωσης είναι τις χρεώσεις τελικού καταναλωτή να καθορίζουν οι δυνάμεις της αγοράς και όχι κάποια ρυθμιστική αρχή. Ωστόσο, η μεγαλύτερη διαφάνεια μπορεί να διευκολύνει τη μετάβαση σε μία ανταγωνιστική αγορά και η επιβολή της μεταβατικής υποχρέωσης για τους μεγάλους προμηθευτές να αγοράζουν από το pool, μπορεί να είναι δικαιολογημένη. Ο προβληματισμός ότι οι διμερείς εμπορικές συναλλαγές μπορεί να οδηγήσουν σε χειραγώγηση της αγοράς έχει αντιμετωπισθεί στην Καλιφόρνια μέσω ορισμένων μεταβατικών διακανονισμών που περιορίζουν, χωρίς όμως να αναστέλλουν, τις διμερείς συναλλαγές. Οι εταιρείες κοινής ωφελείας που δραστηριοποιούνταν στη διανομή ηλεκτρικής ενέργειας, όφειλαν να χρησιμοποιούν το χρηματιστήριο συναλλαγών για μια μεταβατική περίοδο τεσσάρων ετών. Για τους υπόλοιπους αγοραστές και πωλητές ηλεκτρικής ενέργειας, η χρήση του οργανωμένου χρηματιστηρίου ηλεκτρικής ενέργειας είναι προαιρετική. Έτσι, δεν επιτρέπεται στους μεγαλύτερους παίκτες της αγοράς να εμπλακούν σε διμερείς συμβάσεις, χωρίς παράλληλα να επιβάλλονται περιορισμοί στους μικρότερους παίκτες.

Οι μακροπρόθεσμες διμερείς συμβάσεις μπορεί να διευκολύνουν τη χειραγώγηση της αγοράς, εφόσον οι παίκτες ήδη ελέγχουν την αγορά σε κάποιο βαθμό. Συγκεκριμένα, οι διμερείς συμβάσεις μπορεί να επιφέρουν μία λανθάνουσα μορφή κάθετης ολοκλήρωσης μεταξύ παραγωγών και προμηθευτών σε συστήματα όπου η κάθετη ολοκλήρωση δεν είναι επιτρεπτή. Σε μία όμως αγορά με πολλούς υποψήφιους αγοραστές και πωλητές, δεν θα υπήρχαν πολλά κίνητρα για να συνάψει κανείς διμερείς συμβάσεις σε μη ανταγωνιστικές τιμές. Αυτό σημαίνει ότι το πρόβλημα εντοπίζεται μάλλον στην ίδια τη δυνατότητα ελέγχου της αγοράς και ότι οι μη ανταγωνιστικές διμερείς συμβάσεις είναι απλώς το σύμπτωμα. Το διμερές εμπόριο ηλεκτρικής ενέργειας έχει επιτραπεί διαδοχικά στις

Σκανδιναβικές χώρες, τη Νέα Ζηλανδία, την Ισπανία, τις Η.Π.Α. και τη Γερμανία. Το Ηνωμένο Βασίλειο έχει προτείνει νέους διακανονισμούς συναλλαγών που θα επιτρέψουν την πραγματοποίηση διμερών συναλλαγών όπως ήδη αναφέρθηκε. Στην Καλιφόρνια και έμμεσα στην Ισπανία, υπάρχουν χρονικοί περιορισμοί στις διμερείς συναλλαγές, οι οποίοι επιβάλλονται στις μεγαλύτερες εταιρείες κοινής ωφελείας. Ωστόσο, η τάση αυτή δεν είναι παγκόσμια, καθώς στις περισσότερες χώρες τα χρηματιστήρια συναλλαγών ηλεκτρικής ενέργειας είναι ως επί το πλείστον υποχρεωτικά.

3.1.4. Χρέωση Ισχύος

Οι χρεώσεις ισχύος, γνωστές και ως χρεώσεις διαθεσιμότητας, είναι πληρωμές προς παραγωγούς ηλεκτρικής ενέργειας με αντάλλαγμα τη διαθεσιμότητα της παραγωγικής τους ισχύος. Με τον τρόπο αυτό η χρέωση της ηλεκτρικής ενέργειας από τους παραγωγούς έχει τουλάχιστον δύο συνιστώσες. Η μία έχει σχέση με την πραγματική παραγωγή ενέργειας ενώ η άλλη καθορίζεται από την παραγωγική ισχύ που διατίθεται από τους παραγωγούς. Οι χρεώσεις ισχύος άλλοτε καθορίζονται διοικητικά, όπως συμβαίνει σε Αγγλία- Ουαλία και Ισπανία, ή μέσω των μηχανισμών της αγοράς, όπως στην περίπτωση της PJM στις ΗΠΑ.

Ο απώτερος αντικειμενικός σκοπός της χρέωσης ισχύος είναι να αυξήσει την ασφάλεια τροφοδοσίας ηλεκτρικής ενέργειας περισσότερο από το βαθμό που αναμένεται να παρέχει η ίδια η αγορά. Στοχεύει τελικά στο να βελτιώσει την ασφάλεια ενθαρρύνοντας μεγαλύτερο περιθώριο εφεδρείας ή μειώνοντας τις διακυμάνσεις της διαθεσιμότητας στο βάθος του χρόνου, ή και τα δύο. Θα ανέμενε κανείς ότι οι χρεώσεις ισχύος συμβάλλουν σε μεγαλύτερες και σταθερότερες επενδύσεις, τηρουμένων βέβαια κάποιων προϋποθέσεων. Μια τέτοια προϋπόθεση για να οδηγήσουν οι πληρωμές ισχύος σε μακροπρόθεσμες επενδύσεις είναι η αξιοπιστία. Οι επενδυτές δεν πρόκειται να τροποποιήσουν τις επενδυτικές τους αποφάσεις εάν οι πληρωμές ισχύος θεωρηθούν ως ένα μεταβατικό μέτρο που οδεύει προς κατάργηση. Το γεγονός ότι οι χρεώσεις ισχύος δέχονται κριτική και έλεγχο από τις ρυθμιστικές αρχές, τόσο στη Μ.Βρετανία όσο και στην Ισπανία, φανερώνει ότι η επίδραση στις επενδυτικές προτάσεις είναι μάλλον περιορισμένη. Μια άλλη προϋπόθεση είναι η ανάγκη οι χρεώσεις ισχύος να αντικατοπτρίζουν το μακροχρόνιο κόστος της ισχύος. Αν υπάρχει στρέβλωση δεν παρέχεται το ζητούμενο κίνητρο για επενδύσεις.

Οι ρυθμιζόμενες χρεώσεις ισχύος, δηλαδή οι πληρωμές που επιβάλλει ο ρυθμιστής, εφαρμόζονται για να τροποποιήσουν την απόδοση των αγορών ηλεκτρικής ενέργειας. Η εφαρμογή τους έχει να κάνει με την πεποίθηση ότι είναι πιθανό να έχουμε μια μορφή ανεπάρκειας στην αγορά, η οποία θα έχει ως αποτέλεσμα ένα πολύ χαμηλό επίπεδο αξιοπιστίας. Τέσσερις είναι οι βασικές μορφές ενδεχόμενης ανεπάρκειας της αγοράς:

- *Επενδυτικοί κύκλοι*: Πολλές φορές έχει λεχθεί ότι οι επενδύσεις διαγράφουν κύκλους. Εάν, για παράδειγμα, οι επενδυτές έχουν βραχυπρόθεσμες βλέψεις τότε είναι πιθανό οι κύκλοι να είναι έντονοι. Εάν οι επενδυτές έχουν πιο μακροπρόθεσμους ορίζοντες, τότε οι επενδυτικοί κύκλοι είναι μικρότεροι ή αμελητέοι. Το ζήτημα θα μπορούσε να αντιμετωπιστεί με τα κατάλληλα κίνητρα, όπως η επιβολή χρηματικών προστίμων στις περιπτώσεις μη-παράδοσης. Στην πράξη, υπάρχουν διάφοροι παράγοντες που μπορούν να χρησιμεύσουν ως αντιστάθμισμα για κάθε είδους επενδυτική καθυστέρηση, όπως η νέα τεχνολογία που μειώνει κατά πολύ τους χρόνους κατασκευής, η προοπτική να τεθούν ξανά σε λειτουργία υφιστάμενα εργοστάσια ηλεκτρικής ενέργειας, και η δυνατότητα χρέωσης του φορτίου αιχμής η οποία εξομαλύνει τη ζήτηση.
- *Οι επενδύσεις σε ισχύ εφεδρείας είναι υψηλού κινδύνου*: ορισμένα συστήματα ηλεκτρικής ενέργειας, ιδιαίτερα όσα βασίζονται κατά κύριο λόγο σε υδροηλεκτρικούς σταθμούς, χρειάζονται μεγάλες επενδύσεις σε ισχύ εφεδρείας, η οποία χρησιμοποιείται απρόβλεπτα και μάλλον σπάνια. Οι επενδύσεις αυτές μπορεί να χαρακτηριστούν από τους επενδυτές ως υψηλού κινδύνου. Οι τιμές που θα έπρεπε να επιβληθούν για να είναι αυτές οι επενδύσεις επικερδείς, στις λίγες περιπτώσεις που οι εγκαταστάσεις αυτές χρησιμοποιούνται, θα έπρεπε να ήταν πολύ υψηλές. Ωστόσο, δεν έχει γίνει σαφές αν η καλύτερη δυνατή λύση απαιτεί πάντα και

ρυθμιστικές παρεμβάσεις. Για παράδειγμα, οι εταιρείες παραγωγής θα μπορούσαν να μειώσουν τον κίνδυνο αν οι σταθμοί παραγωγής που είχαν στην κατοχή τους παρουσίαζαν μεγαλύτερη διασπορά ως προς τον τύπο και τη χρησιμοποιούμενη τεχνολογία. Παράλληλα, τόσο οι παραγωγοί όσο και οι προμηθευτές θα μπορούσαν να συνάψουν χρηματοοικονομικά συμβόλαια με στόχο να αντισταθμίσουν τον κίνδυνο. Η αύξηση των εμπορικών συναλλαγών στις αγορές ηλεκτρικής ενέργειας διεθνώς, μπορούν επίσης να συμβάλουν σημαντικά στη διασπορά της παραγωγικής βάσης από την οποία εξαρτάται μία χώρα (αυτό γίνεται για παράδειγμα στο NordPool, λόγω της μεγάλης εξάρτησης της Νορβηγίας από την υδροηλεκτρική ενέργεια).

- *Το υψηλό κόστος του κεφαλαίου:* το κόστος του κεφαλαίου για τις παραγωγικές εγκαταστάσεις μπορεί να είναι δυσβάστακτα υψηλό σε μία ανταγωνιστική αγορά εξαιτίας του ότι οι επενδύσεις αυτές είναι εντάσεις κεφαλαίου και ότι έχουν μεγάλη διάρκεια ζωής. Συνέπεια αυτού μπορεί να είναι η συρρίκνωση των επενδύσεων και ο χαμηλός βαθμός ασφάλειας της τροφοδοσίας. Το κόστος του κεφαλαίου αναμφίβολα μπορεί να είναι υψηλότερο σε μια ανταγωνιστική αγορά. Κατά το παρελθόν, οι αποδόσεις των επενδύσεων στη βιομηχανία ηλεκτρικής ενέργειας ήταν εγγυημένες από τις ρυθμιστικές αρχές και οι επενδυτικοί κίνδυνοι ήταν σχετικά περιορισμένοι. Με τον ανταγωνισμό, ο επενδυτικός κίνδυνος μετατοπίζεται από τους καταναλωτές ηλεκτρικής ενέργειας στους ιδιοκτήτες των παραγωγικών εγκαταστάσεων και το κόστος του κεφαλαίου για τέτοιου είδους εγκαταστάσεις αναμένεται να αυξηθεί καθώς πλησιάζει σε φυσιολογικά επίπεδα αγοράς, αντικατοπτρίζοντας το κόστος του κεφαλαίου σε άλλες παρεμφερείς βιομηχανίες. Η αναπροσαρμογή του κόστους κεφαλαίου στα επίπεδα της αγοράς θα πρέπει να βελτιώνει την απόδοση. Επιπλέον, διάφορες χρηματοοικονομικά εργαλεία διαθέσιμα στις ανταγωνιστικές αγορές επιτρέπουν στις εταιρείες να περιορίζουν τους επενδυτικούς τους κινδύνους. Πάντως, η εμπειρία από άλλου είδους βιομηχανικούς τομείς που έχουν να αντιμετωπίσουν υψηλότερο κόστος κεφαλαίου, συνηγορεί στο ότι οι επενδύσεις δεν αποθαρρύνονται με το άνοιγμα των αγορών στον ανταγωνισμό.
- *Χαμηλό επίπεδο τιμών:* Λέγεται πολλές φορές ότι στις ανταγωνιστικές αγορές ηλεκτρικής ενέργειας οι τιμές εμφανίζουν την τάση να πέφτουν κάτω του κόστους, αποθαρρύνοντας έτσι τις επενδύσεις. Ο ισχυρισμός αυτός προέρχεται από μια παρανόηση σχετικά με το πώς λειτουργούν οι ανταγωνιστικές αγορές και δεν είναι τεκμηριωμένος. Οι ανταγωνιστικές τιμές αναμένεται να καλύψουν πλήρως το κόστος παραγωγής και ως τώρα οι χονδρικές τιμές ηλεκτρικής ενέργειας δεν έχουν γενικά πέσει σε τόσο χαμηλά επίπεδα. Σε ορισμένες περιπτώσεις μάλιστα, θεωρούνται μάλλον υψηλές.

Πέρα από την πιθανή επίδρασή τους στην επενδυτική δραστηριότητα, οι χρεώσεις ισχύος μπορεί να διαταράξουν την αποδοτικότητα της αγοράς με διάφορους άλλους τρόπους. Μπορεί να προκαλέσουν «στρατηγική» συμπεριφορά από την πλευρά των παραγωγών όπως για παράδειγμα συνέβη στο pool Αγγλίας-Ουαλίας, όπου παρατηρήθηκαν προσπάθειες εκ μέρους των παραγωγών να αυξήσουν τις χρεώσεις ισχύος μέσω των ανακοινωθέντων διαθεσιμότητων. Οι χρεώσεις ισχύος γενικά αυξάνουν τις χονδρικές και τελικές τιμές της ηλεκτρικής ενέργειας και γενικώς μπορεί να διαταράξουν τον ανταγωνισμό, ιδιαίτερα την είσοδο νέων ανταγωνιστών, επειδή παρέχουν έσοδα στους υφιστάμενους παραγωγούς ανεξάρτητα από το αν πραγματικά πωλούν ή όχι ηλεκτρική ενέργεια. Επιπλέον, αντιμετωπίζουν όλους, ή σχεδόν όλους, τους αγοραστές και πωλητές ηλεκτρικής ενέργειας με τον ίδιο τρόπο, ανεξάρτητα από την πραγματική τους ζήτηση για την επάρκεια της τροφοδοσίας. Οι καταναλωτές, ωστόσο, ποικίλουν ως προς τον τύπο - από μικρούς οικιακούς καταναλωτές σε μεγάλες βιομηχανίες - και πιθανότατα και ως προς τον τρόπο που αξιολογούν το θέμα της ασφάλειας τροφοδοσίας, γεγονός που δεν αντικατοπτρίζεται στον τρόπο χρέωσής τους. Οι χρεώσεις ισχύος ίσως να μην καταφέρουν να κάνουν τη διάκριση ανάμεσα στις επενδύσεις που συμβάλλουν σημαντικά στην ασφάλεια και στις επενδύσεις που δεν έχουν τέτοια συμβολή (π.χ. οι εγκαταστάσεις που λειτουργούν με αέριο για να αντιμετωπίσουν τις εποχιακές διακυμάνσεις σε ένα σύστημα υδροηλεκτρικής βάσης, σε αντίθεση με τις επιπρόσθετες υδροηλεκτρικές μονάδες)

και επιπλέον μπορεί να χρησιμεύσουν ως ένας βολικός τρόπος για την κάλυψη των απρόβλεπτων δαπανών που προκύπτουν από το προ της αναδιάρθρωσης σύστημα.

Υπάρχουν, ωστόσο, εναλλακτικές λύσεις αντί των χρεώσεων ισχύος που επιβάλλονται από τις ρυθμιστικές αρχές και περιλαμβάνουν:

- Υποχρεώσεις για την εξασφάλιση της παροχής, με πρόστιμα στις εταιρείες προμήθειας σε περιπτώσεις μη-παράδοσης.
- Κυριότητα των μονάδων παραγωγής, οι οποίες αντιμετωπίζουν εισοδηματική αβεβαιότητα, από τον διαχειριστή του συστήματος (π.χ. κάποια εργοστάσια αιχμής που χρησιμοποιούνται σπανίως, όπως στη Σουηδία).
- Παρακολούθηση των επενδύσεων από τον ρυθμιστή, με την δυνατότητα ρυθμιστικής παρέμβασης αν και όταν προβλέπονται προβλήματα.
- Μία προαιρετική οργανωμένη αγορά ισχύος που θα διασφαλίζει τη διαφάνεια και την ορθολογική χρέωση της διαθεσιμότητας ισχύος.
- Δυνατότητα σύναψης Συμβολαίων Διακοπτόμενης Παροχής για τους τελικούς καταναλωτές.

3.1.5. Χρηματοοικονομικές αγορές

Οι χρηματοοικονομικές συμβάσεις παίζουν αποφασιστικό ρόλο στην εξασφάλιση των παικτών στις αγορές ηλεκτρικής ενέργειας έναντι εντόνων διακυμάνσεων των τιμών. Οι τιμές στις ανταγωνιστικές αγορές ηλεκτρικής ενέργειας κινούνται ταχύτατα προκειμένου να ανταποκριθούν στις μεταβαλλόμενες συνθήκες προσφοράς και ζήτησης και, ως εκ τούτου είναι ιδιαίτερα ευμετάβλητες. Οι αυξομειώσεις των τιμών έχουν θετική επίδραση στην καλύτερη δυνατή εξομάλυνση της προσφοράς και της ζήτησης, αλλά μπορεί να έχουν αρνητική επίδραση στους παίκτες της αγοράς. Οι αγορές συμβολαίων ηλεκτρικής ενέργειας βοηθούν στην καλύτερη διαχείριση των κινδύνων που σχετίζονται με την τιμή ή τις παραδοτέες ποσότητες ηλεκτρισμού. Ενώ παλαιότερα οι κίνδυνοι που αφορούσαν στην βιομηχανία ηλεκτρισμού ήταν συνυφασμένοι με την ίδια την ηλεκτρική ενέργεια, τώρα ο κίνδυνος διαχωρίζεται από την προμήθεια του ίδιου του προϊόντος και μπορεί, ως ένα βαθμό, η διαχείριση και χρέωσή του να γίνονται ξεχωριστά και πιο ευέλικτα.

Οι αγορές συμβολαίων ηλεκτρικής ενέργειας είναι, κατά κύριο λόγο, χρηματοοικονομικές αγορές και δεν απαιτούν εξειδικευμένες ρυθμίσεις για το συγκεκριμένο προϊόν. Ωστόσο, όπως και σε άλλες προθεσμιακές αγορές, τα χρηματοοικονομικά συμβόλαια ηλεκτρικής ενέργειας ενδέχεται να χρειαστεί να συνοδευτούν από χρηματικά πρόστιμα και εγγυήσεις, προκειμένου να εξασφαλιστεί η τήρηση των συμβατικών όρων. Πολλοί είναι αυτοί που υποστηρίζουν ότι οι συμβάσεις βοηθούν στον περιορισμό της δυνατότητας ελέγχου της αγοράς από τους παραγωγούς καθώς τα χρηματοοικονομικά συμβόλαια μειώνουν τα κίνητρα των παραγωγών να θέσουν υψηλές τιμές στην χονδρική αγορά, επειδή η τιμή που εισπράττει ο παραγωγός καθορίζεται στο προθεσμιακό συμβόλαιο και όχι στη χονδρική αγορά. Ωστόσο, στοιχεία από το ροοί Αγγλίας και Ουαλίας δεν δείχνουν ότι τα προθεσμιακά συμβόλαια μετρίασαν σημαντικά την εμφάνιση τάσεων ελέγχου της αγοράς.

Τα χρηματοοικονομικά συμβόλαια ηλεκτρικής ενέργειας έχουν κυρίως τη μορφή προθεσμιακών συμβολαίων (forwards), συμβολαίων μελλοντικής εκπλήρωσης (futures), δικαιωμάτων προαίρεσης (options), ή Προγραμματικών Συμφωνιών (Power Purchase Agreements - PPA). Τα πρώτες τρία συνήθως χαρακτηρίζονται ως χρηματοοικονομικά συμβόλαια, επειδή δεν χρειάζεται να προσδιοριστεί ποια συγκεκριμένη μονάδα θα παρέχει την ηλεκτρική ενέργεια. Οι Προγραμματικές Συμφωνίες αποκαλούνται φυσικές συμβάσεις, επειδή ακριβώς προσδιορίζουν το εργοστάσιο που θα παρέχει την ηλεκτρική ενέργεια. Συνήθως χρησιμοποιούνται από

ανεξάρτητους παραγωγούς ηλεκτρικής ενέργειας που πωλούν σε κάποιο μονοπωλιακό αγοραστή, μπορούν όμως να χρησιμοποιηθούν και σε άλλες περιπτώσεις.

Τα **προθεσμιακά συμβόλαια** είναι διμερείς συμφωνίες για την παράδοση ενέργειας σε ορισμένη τιμή. Λειτουργούν σε συνδυασμό με μία ανταγωνιστική χονδρική αγορά και είναι ένα από τα απλούστερα εργαλεία παραγωγών που χρησιμεύουν για τη μεταβίβαση ή την αντιστάθμιση του κινδύνου που ενέχει η αυξομείωση των τιμών. Οι συμβαλλόμενοι συμφωνούν σε μία τιμή σήμερα (strike price) για την παράδοση συγκεκριμένης ποσότητας ενέργειας αργότερα. Ο διακανονισμός ενός προθεσμιακού συμβολαίου μπορεί να γίνει χωρίς να υπάρξει φυσική παράδοση. Αν η τιμή της αγοράς κατά την παράδοση είναι υψηλότερη από την τιμή του συμβολαίου, ο πωλητής της σύμβασης αποζημιώνει τον αγοραστή για τη διαφορά. Αν είναι χαμηλότερη, ο αγοραστής της σύμβασης αποζημιώνει τον πωλητή. Το προθεσμιακό συμβόλαιο που διευθετείται καθ' αυτό τον τρόπο, αποκαλείται συμβόλαιο διευθέτησης αποκλίσεων. Τα **συμβόλαια μελλοντικής εκπλήρωσης** είναι ανάλογα με τα προθεσμιακά με τη μόνη διαφορά ότι τυποποιούνται και εφαρμόζονται σε οργανωμένες αγορές. Τα **δικαιώματα προαίρεσης** δίνουν τη δυνατότητα, χωρίς να δημιουργούν την υποχρέωση, να αγοράζεται ή να πωλείται ηλεκτρική ενέργεια σε ορισμένη τιμή. Η τιμή έχει δύο συστατικά: μία πάγια αμοιβή ίση με τη χρέωση kW που καταβάλλεται με την υπογραφή της σύμβασης, και μία συμφωνημένη τιμή που καταβάλλεται για κάθε kWh που παραδίδεται στην πραγματικότητα. Το δικαίωμα προαίρεσης δεν απαιτεί για την εφαρμογή του μια τιμή spot ηλεκτρικής ενέργειας ως σημείο αναφοράς. Σε σύγκριση με τα συμβόλαια μελλοντικής εκπλήρωσης, τα δικαιώματα προαίρεσης δίνουν το πλεονέκτημα (στον πωλητή) να αντισταθμίζει εν μέρει τους κινδύνους που σχετίζονται με την ποσότητα, καθώς το πάγιο κόστος παραγωγής μπορεί να καλυφθεί από το σκέλος της πάγιας αμοιβής. Τα χρηματοοικονομικά συμβόλαια χρησιμοποιούνται σήμερα σε διάφορες αγορές ηλεκτρικής ενέργειας κυρίως για τη διευθέτηση των αποκλίσεων και την αντιστάθμιση του κινδύνου που σχετίζονται τόσο με τις τιμές όσο και με τις ποσότητες.

Οι **Προγραμματικές Συμφωνίες** (PPA) μοιάζουν με τις χρηματοοικονομικές συμβάσεις που περιγράψαμε αλλά, από τη στιγμή που προσδιορίζουν τη μονάδα που θα παρέχει την ηλεκτρική ενέργεια, απαιτούν τον αποκλεισμό του παραγωγού από όλα τα συγκεντρωτικά pool. Σε περιπτώσεις όπου υφίσταται υποχρεωτική pool αγορά ηλεκτρικής ενέργειας, η εφαρμογή αυτού του τύπου συμβάσεων είναι είτε περιορισμένη είτε απαγορευμένη.

3.1.6. Χρέωση Μεταφοράς και Διευθέτηση Συμφορήσεων

Τα δίκτυα μεταφοράς και διανομής και οι σχετικές με αυτά ρυθμιστικές διατάξεις και κανονισμοί, συνιστούν το κρίσιμο συνδυαστικό στοιχείο στις αγορές ηλεκτρικής ενέργειας για την ύπαρξη του ανταγωνισμού. Το δικαίωμα πρόσβασης στο δίκτυο και η κατάλληλη χρέωση είναι καθοριστικής σημασίας για την ανάπτυξη του ανταγωνισμού. Επίσης, στο βαθμό που κάθε αναδιάρθρωση στοχεύει ακόμη στο να ενοποιήσει μέχρι πρότινος ανεξάρτητες εθνικές ή περιφερειακές αγορές, οι ρυθμίσεις που αφορούν στη διασύνδεση των Ηλεκτρικών Συστημάτων, αποτελούν μείζον ζήτημα. Εξετάζοντας κανείς το ζήτημα της διαχείρισης του δικτύου το πρώτο και σημαντικότερο θέμα είναι ο ρόλος των τιμών στη βελτίωση της αποδοτικότητας. Βραχυπρόθεσμα, το κυρίαρχο ρυθμιστικό ζήτημα είναι ο αποτελεσματικός επιμερισμός της χρέωσης μεταφοράς (κυρίως των φορτίων αιχμής) και της κατανομής της ισχύος. Μακροπρόθεσμα, οι ρυθμίσεις πρέπει να παρέχουν τα κατάλληλα κίνητρα για αποδοτικές επενδύσεις, χωρίς να διαταράσσουν την οικονομική αντοχή των ρυθμιζόμενων εταιρειών. Ένα ακόμη σχετικό με τη χρέωση ζήτημα είναι ο επιμερισμός του κόστους των παγίων του δικτύου στους διάφορους χρήστες.

Ο τρόπος χρέωσης αποτελεί βασικό εργαλείο των αναδιρθρωτικών ρυθμίσεων που αφορούν στο δίκτυο και εξυπηρετούν ένα σύνολο συμπληρωματικών στόχων:

- την ανάκτηση των ήδη επενδεδυμένων κεφαλαίων (sunk costs) και την εξασφάλιση χρηματοοικονομικής επάρκειας.
- την παροχή κατάλληλων κινήτρων για μακροπρόθεσμες επενδύσεις (μακροπρόθεσμη αποδοτικότητα).
- την παροχή τα κατάλληλων ερεθισμάτων για την αποτελεσματική λειτουργία του δικτύου ή, με άλλα λόγια, για τον βέλτιστο επιμερισμό της διαθέσιμης δυναμικότητας μεταφοράς ώστε να αντιμετωπιστούν αποτελεσματικά φαινόμενα συμφόρησης (βραχυπρόθεσμη αποδοτικότητα).
- την αποτροπή φαινομένων διακρίσεων μεταξύ των χρηστών του Συστήματος μεταφοράς (ουδετερότητα) και,
- την απλούστευση και τη διαφάνεια των διαδικασιών.

Ο στόχος της βραχυπρόθεσμης αποτελεσματικότητας είναι εξαιρετικά σημαντικός αλλά, ταυτόχρονα, δεν είναι και εύκολο να επιτευχθεί. Παρ'όλο που το ζήτημα της βέλτιστης χρέωσης για τον καταμερισμό της δυναμικότητας μεταφοράς, με στόχο την αποφυγή φαινομένων συμφόρησης, μπορεί να φαίνεται ένα ζήτημα τεχνικής φύσης με περιορισμένες επιπλοκές, εντούτοις, αποτελεί κεντρικό ζήτημα στο συνολικό σχεδιασμό αποδοτικών αγορών ηλεκτρικής ενέργειας επειδή, εκτός από τον άμεσο ρόλο του στον καταμερισμό της δυναμικότητας μεταφοράς, επιδρά και στην κατανομή των μονάδων παραγωγής. Η επίδραση που ασκούν οι χρεώσεις μεταφοράς στην κατανομή εσόδων και κερδών μεταξύ των παραγωγών, μπορεί να είναι εξίσου σημαντική. Εξάλλου, η χρέωση της μεταφοράς μπορεί να επηρεάσει τον ανταγωνισμό σε επίπεδο παραγωγής, είτε με το να τον διευκολύνει είτε με το να τον δυσχεράνει.

Έχουν αναπτυχθεί πολλές οι μέθοδοι χρέωσης προκειμένου να επιτευχθούν οι διάφοροι στόχοι. Το μέσο επίπεδο τιμών ή το ανώτατο επιτρεπτό επίπεδο εσόδων μπορεί να καθοριστεί έτσι ώστε να καλύπτει το κόστος και ένα εύλογο κέρδος για το Διαχειριστή του Συστήματος ενώ παράλληλα να είναι τέτοιο ώστε να παρέχει κίνητρα αποδοτικότητας. Οι χρεώσεις συνήθως καθορίζονται μέσω σύνθετων τιμολογίων, τα οποία μπορεί να περιλαμβάνουν πάγιες χρεώσεις σύνδεσης, καθώς και χρεώσεις ισχύος και ενέργειας ανάλογα με το χρόνο χρήσης. Κάθε τμήμα του τιμολογίου μπορεί να προσδιοριστεί ξεχωριστά για κάθε τοποθεσία ή και για κάθε καταναλωτή. Με τον τρόπο αυτό, μπορούμε να έχουμε για ένα συγκεκριμένο μέσο επίπεδο τιμών, πολλούς διαφορετικούς συνδυασμούς χρεώσεων ανά τοποθεσία και ανά καταναλωτή. Για να επιτευχθούν οι διάφοροι στόχοι που αναφέρθηκαν πιο πάνω, οι χρεώσεις μεταφοράς πρέπει να συνδυάζουν ένα σύνολο μεθόδων χρέωσης. Στον Πίνακα 2 συνοψίζονται οι μέθοδοι χρεώσεων που έχουν υιοθετηθεί από διάφορες υπό αναδιάρθρωση χώρες.

Αγορά	Προσδιορισμός Μέσου Επιπέδου Τιμών	Χρέωση	Τρόπος Κατανομής	Τέλη ανάκτησης επενδυμένων κεφαλαίων (sunk costs)
Αυστραλία	CPI-X	Ζωνική	Ολοκληρωμένη Κατανομή (διαφοροποίηση ανά περιοχή)	Πάγιο τέλος σύνδεσης και ενιαίος τρόπος χρέωσης ενέργειας
Φινλανδία	Βάσει κόστους	Ενιαία	Εκκαθάριση αγοράς χωρίς περιορισμούς & επανακατανομή σε περίπτωση συμφορήσεων	Χρέωση Ενέργειας
Νέα Ζηλανδία	Ανώτατο όριο κέρδους	Κομβική	Πλήρως Ολοκληρωμένη Κατανομή	Τέλος σύνδεσης και χρέωση ισχύος
Νορβηγία	CPI-X	Ενιαία (ανά Περιφέρειες)	Εκκαθάριση Αγοράς χωρίς περιορισμούς & εφαρμογή τοπικής ή κομβικής χρέωση σε περίπτωση συμφορήσεων	Χρέωση ισχύος για φορτία αιχμής
Ισπανία	CPI-X	Ενιαία	Εκκαθάριση Αγοράς χωρίς περιορισμούς & επανακατανομή σε περίπτωση συμφορήσεων	Χρέωση Ισχύος και χρέωση ενέργειας
Σουηδία	Βάσει κόστους	Ενιαία (ανά Περιφέρειες)	Εκκαθάριση Αγοράς χωρίς περιορισμούς & επανακατανομή σε περίπτωση συμφορήσεων	Χρέωση ισχύος
ΗΠΑ (Καλιφόρνια)	Βάσει Ιστορικού κόστους	Ζωνική	Εκκαθάριση Αγοράς χωρίς περιορισμούς	Επιβάρυνση τελικών καταναλωτών
ΗΠΑ (PJM)	Βάσει Ιστορικού κόστους	Κομβική	Πλήρως Ολοκληρωμένη Κατανομή	Χρέωση ισχύος για φορτία αιχμής
Μ.Βρετανία	CPI-X	Ενιαία	Εκκαθάριση Αγοράς χωρίς περιορισμούς & επανακατανομή σε περίπτωση συμφορήσεων	Ζωνική Χρέωση Ισχύος

Πίνακας 2. Σύγκριση συστημάτων χρέωσης μεταφοράς σε διάφορες χώρες

Βραχυπρόθεσμη Χρέωση - Διαχείριση Συμφορήσεων

Υπάρχουν δύο βασικές προσεγγίσεις στο ζήτημα της χρέωσης των υπηρεσιών μεταφοράς. Υπάρχουν τιμολόγια που δεν βασίζονται στις συναλλαγές, τα λεγόμενα «σημειακά τιμολόγια» (point tariffs), τα οποία είναι ανεξάρτητα από τις εμπορικές συναλλαγές που απαιτούν τη μεταφορά της ηλεκτρικής ενέργειας. Τα τιμολόγια αυτά λαμβάνουν υπόψη μόνο την ποσότητα ενέργειας που εγχέεται ή παραλαμβάνεται σε κάθε κόμβο και σχεδιάζονται με τρόπο που να αντικατοπτρίζουν το κόστος χρήσης του δικτύου. Τα σημειακά τιμολόγια που λαμβάνουν υπόψη και την τοποθεσία – κομβική και ζωνική τιμολόγηση – στοχεύουν στην καλύτερη αντιμετώπιση των συμφορήσεων.

Η άλλη μέθοδος καθορισμού τιμολογίων βασίζεται στις πραγματοποιούμενες συναλλαγές και συνίσταται στη δημιουργία point-to-point τιμολογίων, τα οποία εξαρτώνται από την πηγή και την κατάληξη της εκάστοτε συναλλαγής. Τα «τιμολόγια διαδρομής» (contract path) και τα «τιμολόγια απόστασης» είναι δύο χαρακτηριστικά παραδείγματα αυτής της κατηγορίας. Τα τιμολόγια που βασίζονται στις συναλλαγές δεν αντικατοπτρίζουν σε γενικές γραμμές το κόστος και δεν είναι τόσο αποτελεσματικά στην αντιμετώπιση των συμφορήσεων. Εντούτοις, χρησιμοποιούνται ακόμη ευρέως όπως και παλαιότερα.

Κομβική χρέωση

Η κομβική χρέωση εξισώνει την προσφορά και τη ζήτηση ηλεκτρικής ενέργειας σε κάθε κόμβο του δικτύου μεταφοράς. Οι κομβικές χρεώσεις αναπροσαρμόζονται διαρκώς και καθορίζονται για την παραδοθείσα ενέργεια, συμπεριλαμβάνοντας τόσο την χρέωση ενέργειας όσο και της μεταφοράς. Συνεπώς, η τιμή που χρεώνεται για τη μεταφορά εμπεριέχεται στην κομβική χρέωση. Η κομβική χρέωση είναι γνωστή και ως τιμολόγηση τοπικής εκκαθάρισης της αγοράς, αλλά και ως μέθοδος διάσπασης της αγοράς (“split the market”). Μέχρι πρόσφατα, οι κομβικές τιμές δεν χρησιμοποιούνταν για τον επιμερισμό της δυναμικότητας μεταφοράς και για την αντιμετώπιση των συμφορήσεων. Αντιθέτως, τα περισσότερα συστήματα ηλεκτρικής ενέργειας βασίζονταν (και βασίζονται ακόμη) σε άλλες τεχνικές χρέωσης σε συνδυασμό με άλλους μηχανισμούς που δεν σχετίζονται με τη χρέωση. Οι κομβικές τιμές χρησιμοποιούνται πλέον σε ορισμένες αγορές των Η.Π.Α. και στη Νέα Ζηλανδία, συχνά σε συνάρτηση με το διαχειριστικό διαχωρισμό μεταφοράς και παραγωγής. Διάφορα κριτήρια έχουν άμεση σχέση με τη συμβολή της κομβικής χρέωσης στην επίτευξη των στόχων της χρέωσης μεταφοράς, που αναφέρθηκαν στην αρχή αυτού του κεφαλαίου, όπως η χρηματοοικονομική επάρκεια, η αποδοτικότητα, η δυνατότητα εφαρμογής και η ανταγωνιστική ουδετερότητα.

Ως προς τη χρηματοοικονομική επάρκεια, η κομβική χρέωση στην πράξη παράγει έσοδα πολύ κάτω του ιστορικού κόστους. Σε ένα βέλτιστο σύστημα, δεν επιτρέπεται να υπάρχει μια τέτοια αντίφαση. Η κομβική χρέωση θα απέφερε το κατάλληλο επίπεδο εσόδων αν τα πάγια της παραγωγής και της μεταφοράς ήταν βέλτιστα σχεδιασμένα και ως εκ τούτου δεν υπήρχε περίσσεια δυναμικότητας παραγωγής ή δυναμικότητας μεταφοράς, η χωροθέτηση των εγκαταστάσεων ήταν τέτοια που ικανοποιούσε το κριτήριο της ελαχιστοποίησης του συνολικού κόστους και τέλος η προσφορά ήταν ανταγωνιστική. Ωστόσο, οι υφιστάμενες εγκαταστάσεις μεταφοράς και παραγωγής είναι αποτέλεσμα συνεχώς αυξανόμενων επενδύσεων που έχουν γίνει κατά καιρούς υπό διαρκώς μεταβαλλόμενες τεχνικές και οικονομικές συνθήκες. Έτσι, στην πράξη, η κομβική χρέωση πρέπει να συνοδεύεται από μία πάγια χρέωση μεταφοράς για την είσπραξη συμπληρωματικών εσόδων.

Ως προς την αποδοτικότητα, η κομβική χρέωση αντικατοπτρίζει τη σχετική έλλειψη της δυναμικότητας μεταφοράς σε κάθε σημείο του δικτύου. Έτσι, εξασφαλίζονται κίνητρα αποδοτικότητας, τόσο βραχυπρόθεσμα όσο και μακροπρόθεσμα. Με την κομβική χρέωση, η μεταφορά είναι σχετικά ακριβή στους κόμβους εκείνους όπου δεν υπάρχει διαθέσιμη επαρκής ισχύς μεταφοράς για να καλύψει όλη την προγραμματισμένη παραγωγή. Οι υψηλότερες τιμές μειώνουν τη ζήτηση για ηλεκτρική ενέργεια, επιλύοντας έτσι το πρόβλημα της συμφόρησης (βραχυπρόθεσμη αποδοτικότητα). Οι υψηλότερες τιμές παρέχουν επίσης κίνητρα για επενδύσεις σε διασυνδέσεις,

σε περιοχές με υψηλές τιμές (μακροπρόθεσμη αποδοτικότητα). Το αν ωστόσο η κομβική χρέωση παρέχει τα ικανά κίνητρα για επενδύσεις που θα εξαλείψουν κάθε είδους αντικοινωνικών συμφορήσεων είναι ένα ανοικτό ζήτημα.

Υπάρχουν επίσης ζητήματα που αφορούν στον τρόπο εφαρμογής και την ανταγωνιστική ουδετερότητα. Η δημιουργία μιας αγοράς για υπηρεσίες μεταφοράς εγείρει ανησυχίες ως προς τη δυνατότητα ελέγχου της αγοράς από τους ιδιοκτήτες των δικτύων μεταφοράς και τον Διαχειριστή του συστήματος. Ο βασικός προβληματισμός είναι ότι ο ή οι ιδιοκτήτες των δικτύων μεταφοράς μπορούν να χειραγωγήσουν τις τιμές. Για παράδειγμα, η δέσμευση δυναμικότητας μεταφοράς μπορεί να δημιουργήσει πλασματική συμφόρηση προκαλώντας, έτσι, αύξηση των τιμών.

Ένας άλλος προβληματισμός είναι ότι ο καθορισμός των κομβικών χρεώσεων απαιτεί τη συγκέντρωση πληροφοριών για τον υπολογισμό των τιμών και την επανακατανομή της παραγόμενης ισχύος με αποδοτικό τρόπο. Έτσι, ο Διαχειριστής του συστήματος έχει τη δυνατότητα να διαμορφώσει τις αποφάσεις της αγοράς, γεγονός το οποίο ευνοεί τις μονοπωλιακές καταχρήσεις. Έχει λεχθεί ότι η αποκεντρωμένη εφαρμογή Διαπραγματεύσιμων Συμβολαίων Συμφόρησης (Tradable Congestion Contracts - TCC) μπορεί να εξασφαλίσει κάποια προστασία έναντι τέτοιου είδους καταχρήσεων. Σύμφωνα με αυτή την προσέγγιση, τα συμβόλαια αυτά πωλούνται στους συμμετέχοντες στην αγορά μέσω πλειστηριασμού και κατόπιν διακινούνται σε κάποια δευτερογενή αγορά. Τα Διαπραγματεύσιμα Συμβόλαια Συμφόρησης προστατεύουν τους παίκτες της αγοράς από μεταβολές των χρεώσεων μεταφοράς και οι δευτερογενείς συναλλαγές περιορίζουν το ρόλο του Διαχειριστή του συστήματος ως προς τη χειραγώγηση των τιμών. Ωστόσο, έχει αποδειχθεί ότι η μέθοδος αυτή μπορεί να επιτείνει τις στρεβλώσεις της αγοράς όταν υπάρχουν ήδη φαινόμενα ελέγχου της αγοράς, η αποδοτικότητά της δεν έχει αποδειχθεί και σε πραγματικές συνθήκες μπορεί να είναι ανεπαρκής. Η κομβική χρέωση επιτρέπει μεν την αποτελεσματική διαχείριση των συμφορήσεων, είναι όμως επιρρεπής στη χειραγώγηση όταν υπάρχει έλεγχος της αγοράς από κάποιους συμμετέχοντες. Σε πολλές περιπτώσεις, πάντως, η κομβική χρέωση πρέπει να συνοδεύεται και από άλλου είδους χρεώσεις προκειμένου να συγκεντρωθούν τα απαραίτητα έσοδα για την κάλυψη του κόστους.

Ζωνική Χρέωση

Στη ζωνική χρέωση η περιοχή ελέγχου ενός Διαχειριστή συστήματος διαιρείται σε ζώνες, και οι τιμές καθορίζονται για κάθε ζώνη εξάγοντας το μέσο όρο του κόστους συμφόρησης των κόμβων εντός της ζώνης αυτής. Βασίζεται στο επιχείρημα ότι τα φαινόμενα συμφόρησης παρατηρούνται σε λίγους μόνο κόμβους του δικτύου. Το πλεονέκτημά της είναι ότι εφαρμόζεται πιο εύκολα από την πλήρη εφαρμογή της κομβικής χρέωσης και στην ουσία αποτελεί μια απλουστευμένη μορφή της.

Ως προς τις υπόλοιπες πλευρές της απόδοσης, η ζωνική και η κομβική χρέωση είναι όμοιες. Η ζωνική χρέωση εξίσου αδυνατεί να παρέχει επαρκή έσοδα (καθιστώντας έτσι αναγκαίο να συνοδεύεται και από άλλες χρεώσεις), επιφέρει κινδύνους για τους συμμετέχοντες στην αγορά (που μπορούν να αντισταθμιστούν μόνο μέσω των TCC), και υπόκεινται σε χειραγώγηση όταν υπάρχει τέτοια δυνατότητα. Καθώς η εμπειρία όσον αφορά την κομβική και τη ζωνική χρέωση μεγαλώνει, η ζυγαριά κλίνει υπέρ της κομβικής χρέωσης. Η ζωνική χρέωση μπορεί να είναι αποδοτική υπό ορισμένες συνθήκες, αλλά η εμφανώς απλουστευμένη της μορφή μπορεί να οδηγήσει σε παρανοήσεις καθώς προϋποθέτει, όχι μόνο την ύπαρξη μηχανισμών τιμολόγησης προκειμένου να αντιμετωπιστεί η εντός της ζώνης συμφόρηση, αλλά και την ανάγκη καθορισμού τιμών για την αντιμετώπιση της συμφόρησης μεταξύ ζωνών, πράγμα το οποίο καθιστά το όλο ζήτημα πολύ περίπλοκο. Εξάλλου, με τη αύξηση της σχετικής τεχνολογίας, ο σκεπτικισμός ως προς την πολυπλοκότητα της κομβικής χρέωσης χάνει σταδιακά έδαφος. Σε ορισμένες περιπτώσεις η ζωνική χρέωση δεν έχει αποδώσει τα αναμενόμενα όπως για παράδειγμα στην αμερικάνικη αγορά (PJM). Η συμφόρηση υποτιμήθηκε και οι συμμετέχοντες στην αγορά προγραμματίσαν περισσότερες διμερείς συναλλαγές απ' όσες μπορούσε να καλύψει το δίκτυο. Έτσι, ο Διαχειριστής αναγκάστηκε να επέμβει διοικητικά για να διατηρήσει την ευστάθεια του συστήματος, περιορίζοντας το δικαίωμα επιλογής στην αγορά.

Ενιαία Χρέωση

Ο καθορισμός πάγιου τέλους είναι μία μέθοδος που ισχύει για προκαθορισμένες χρονικές περιόδους. Έτσι, δίδεται το δικαίωμα να διοχετεύεται ενέργεια σε οποιοδήποτε κόμβο του δικτύου και να λαμβάνεται απ' οποιοδήποτε άλλο. Αυτό σημαίνει ότι πρέπει να εφαρμοστούν άλλες μέθοδοι που δεν αφορούν σε χρέωση για να αντιμετωπιστούν οι όποιες συμφορήσεις που μπορεί να προκύψουν. Παρόλα αυτά, η ενιαία χρέωση έχει το πλεονέκτημα της απλότητας και της διαφάνειας: οι τιμές είναι γνωστές εκ των προτέρων και ελέγχονται εύκολα από τον ρυθμιστή, ενώ η αγορά της παραγωγής διαχωρίζεται από την αγορά της μεταφοράς. Η μέθοδος αυτή είναι πολύ διαδεδομένη στις χώρες της Ευρωπαϊκής Ένωσης.

Η χρέωση, αν και θεσμικά απλή και διαφανής, είναι σε γενικές γραμμές αναποτελεσματική. Οι άλλες μέθοδοι χρέωσης, όπως η κομβική, διαθέτουν ένα σαφές πλεονέκτημα έναντι της ενιαίας χρέωσης, στις περιπτώσεις που τα προβλήματα συμφόρησης είναι εκτεταμένα. Από την άλλη πλευρά, είναι μια λογική μέθοδος όταν τα προβλήματα συμφόρησης είναι περιορισμένα. Σε πολλές χώρες είναι συνηθισμένη η ύπαρξη ενός ισχυρού δικτύου παράλληλα με ένα μεγάλο απόθεμα δυναμικότητας παραγωγής. Σε τέτοιες περιπτώσεις, η συμφόρηση μπορεί να αποτελεί ένα σπάνιο φαινόμενο και τα οφέλη από μια κατάλληλη χρέωση σε συνάρτηση με κάποιο πάγιο τέλος μπορεί να είναι περιορισμένα και να μην αντισταθμίζουν το κόστος και την αυξημένη πολυπλοκότητα της καθολικής εφαρμογής της κομβικής χρέωσης. Για παράδειγμα, εκτιμάται ότι οι απώλειες από μια αναποτελεσματική χρέωση μεταφοράς στο Ηνωμένο Βασίλειο είναι μικρές (κάπου 0,6% επί του συνόλου των εσόδων των παραγωγών). Το επιχείρημα αυτό δικαιολογεί την εφαρμογή της ενιαίας χρέωσης σε πολλές χώρες της Ευρωπαϊκής Ένωσης.

Χρέωση βάσει συναλλαγών

Δύο κοινοί τύποι σημειακών τιμολογίων (point-to-point) είναι οι ακόλουθοι:

Τιμολόγια διαδρομής: Εδώ καθορίζονται τιμές για κάθε γραμμή μεταφοράς στο δίκτυο. Σε κάθε συναλλαγή αποδίδεται μία «συμβατική διαδρομή» κατά μήκος του δικτύου, που ενώνει την τοποθεσία του αγοραστή και του πωλητή. Η τιμή που χρεώνεται στη συναλλαγή είναι το άθροισμα των τιμών των γραμμών μεταφοράς που αποτελούν τη συμβατική διαδρομή. Ενώ μια τέτοια μέθοδος μπορεί να φαίνεται απλή και ίσως αυτονόητη, η συμβατική διαδρομή δεν αντιστοιχεί ούτε στην πραγματική ροή ηλεκτρικής ενέργειας σε όλη την έκταση του δικτύου ούτε στο κόστος της. Έτσι, δεν έχει ουσιαστικό αποτέλεσμα στην αποτελεσματική αντιμετώπιση της συμφόρησης.

Χρέωση βάσει απόστασης: Οι τιμές καθορίζονται ως παράμετρος της απόστασης μεταξύ αγοραστή και πωλητή. Η μέθοδος αυτή μοιάζει με τη χρέωση συμβατικής διαδρομής και έχει παρόμοια προβλήματα. Η χρέωση με βάση την απόσταση εφαρμόζεται στη Γερμανία.

Γενικά, οι προσεγγίσεις βάσει συναλλαγών δεν αποτελούν ικανοποιητική λύση επειδή παράγουν τιμές οι οποίες ούτε το κόστος αντικατοπτρίζουν αλλά ούτε και συμβάλλουν στην αποτελεσματική αντιμετώπιση της συμφόρησης. Επίσης, είναι πιθανό να έχουν επιδράσεις σε βάρος του ανταγωνισμού. Η εφαρμογή τους προϋποθέτει τη γνωστοποίηση πληροφοριών που αφορούν εμπορικές σχέσεις που μπορεί να είναι στρατηγικά ευαίσθητες. Μπορεί επίσης να ευνοήσουν φαινόμενα διακρίσεων εις βάρος κάποιων ανταγωνιστών. Για παράδειγμα, οι τιμές βάσει απόστασης τείνουν να επιβάλλουν μεγαλύτερο κόστος σε απομακρυσμένους και ξένους παραγωγούς, το οποίο δεν αντικατοπτρίζει απαραίτητα το κόστος μεταφοράς.

Άλλες μέθοδοι για την αντιμετώπιση των συμφορήσεων

Όταν απουσιάζει η καθολική εφαρμογή της κομβικής χρέωσης, θα πρέπει να αναπτυχθούν άλλοι μηχανισμοί για τον επιμερισμό της δυναμικότητας μεταφοράς. Ο επιμερισμός της δυναμικότητας μπορεί να γίνει βάσει κάποιων κανόνων προτεραιότητας, οι οποίοι θα δώσουν μία ιεραρχική σειρά μεταξύ παραγωγών και αγοραστών ηλεκτρικής ενέργειας. Επίσης, μπορεί να γίνει επιμερισμός του δικαιώματος πρόσβασης, μέσω μακροπρόθεσμων συμβάσεων.

Ωστόσο, αυτοί οι μηχανισμοί δεν είναι σε γενικές γραμμές πολύ αποδοτικοί και ενδεχομένως να ευνοήσουν φαινόμενα διακρίσεων.

Μπορεί να γίνει σχεδιασμός τέτοιων μηχανισμών που να μην ευνοούν τις διακρίσεις. Παράδειγμα τέτοιου μηχανισμού είναι η επονομαζόμενη προσέγγιση «εξισορροπητικής αγοράς» (countertrade), σύμφωνα με την οποία προβλέπεται η λειτουργία μιας παράλληλης αγοράς παραγωγής – της εξισορροπητικής αγοράς – για να αντιμετωπιστεί η συμφόρηση. Ο διαχειριστής του συστήματος εξισορροπεί απευθείας την αγορά χρησιμοποιώντας την εξισορροπητική αγορά. Αυτό επιτυγχάνεται οργανώνοντας εκ νέου την κατανομή των μονάδων παραγωγής που έχουν συμπεριληφθεί στις συμβάσεις της εξισορροπητικής αγοράς. Οι τιμές ενέργειας και μεταφοράς είναι αυτές που θα έχουν προκύψει απουσία συμφόρησης, ενώ πωλητές και αγοραστές πραγματοποιούν τις συναλλαγές τους σαν να μην υπήρχε καθόλου συμφόρηση. Ως αποτέλεσμα, η εξισορροπητική αγορά δεν μεταφέρει τα απαραίτητα μηνύματα στους παίκτες της αγοράς (που αγνοούν τους περιορισμούς μεταφοράς) και πιθανόν να προκύψουν κάποια βραχυπρόθεσμη μείωση της αποδοτικότητας. Από την άλλη πλευρά, η εξισορροπητική αγορά μπορεί να ενθαρρύνει τον ανταγωνισμό στην αγορά ηλεκτρικής ενέργειας επειδή, κατά τη γνώμη των παικτών, οι περιορισμοί μεταφοράς δεν διαιρούν την αγορά σε τμήματα, και κατά συνέπεια περιορίζεται το περιθώριο χειραγώγησής της.

Μία άλλη προσέγγιση που δεν ευνοεί τις διακρίσεις προβλέπει την διενέργεια πλειστηριασμού, στις περιπτώσεις που υπάρχει έλλειψη δυναμικότητας μεταφοράς. Η μέθοδος αυτή μπορεί να αποφέρει ικανοποιητικό αποτέλεσμα, αν ο πλειστηριασμός να σχεδιαστεί κατάλληλα αλλά δεν είναι πρακτική όταν το φαινόμενο της συμφόρησης είναι συχνό.

Καθορισμός επιπέδου τιμών για τη χρέωση μεταφοράς

Τα τιμολόγια του δικτύου σχεδιάζονται έτσι ώστε να εξασφαλίζουν ένα ορισμένο ποσό εσόδων για εκείνον που παρέχει τις υπηρεσίες χρήσης του δικτύου. Οι ρυθμιζόμενες επιχειρήσεις δικαιούνται να εισπράττουν τόσα έσοδα όσα χρειάζονται για να καλύψουν το ιστορικό τους κόστος, συμπεριλαμβανομένης και μιας απόδοσης επενδύσεων που αντιστοιχεί στο κόστος του κεφαλαίου. Οι ρυθμίσεις αυτές είναι γνωστές ως *κόστος παραχής υπηρεσιών* ή *βαθμός απόδοσης*. Βασίζονται δε, στη λογιστική αξία του ενεργητικού, επιτρέποντας στις εταιρείες να ανακτήσουν τα λογιστικά έξοδα και να εισπράξουν μία «λογική» απόδοση επί της επένδυσης. Οι πληθωριστικές αναπροσαρμογές και τα χρονοδιαγράμματα απόσβεσης είναι δύο μέτρα που μπορούν να εφαρμοστούν επιπροσθέτως. Το κόστος της ρύθμισης υπηρεσιών είναι λογικό, υπό την έννοια ότι δεν επιτρέπει στη ρυθμιζόμενη εταιρεία να έχει οποιοδήποτε ιδιαίτερα υψηλό οικονομικό κέρδος και είναι εξ' ορισμού οικονομικά ανεκτό. Ακόμη, δημιουργεί πολλά κίνητρα για επενδύσεις, καθώς οι τιμές αναπροσαρμόζονται ούτως ώστε οι επενδυτές να εξασφαλίσουν τόσο τη δυνατότητα ανάκτησης των επενδύσεών τους όσο και κάποιο κέρδος. Το κόστος της ρύθμισης υπηρεσιών έχει όμως και τα αρνητικά του. Η μέθοδος αυτή δεν δίνει κίνητρα για αποδοτικότητα κόστους ενώ, στην πραγματικότητα, δίνει κίνητρα για υπερβολικές επενδύσεις και υπερβολικά έξοδα.

Η ρύθμιση βάσει κινήτρων αποτελεί εναλλακτική της ρύθμισης κόστους υπηρεσιών, η οποία έχει στόχο να δώσει κίνητρα για αποδοτικότητα κόστους. Η ρύθμιση βάσει κινήτρων επιτρέπει στη ρυθμιζόμενη εταιρεία να διατηρήσει προσωρινά κάποια από τα οφέλη (ή και όλα) τα οποία προκύπτουν από τις τροποποιήσεις που έχουν στόχο να βελτιώσουν την αποδοτικότητα. Συνεπώς, δίνει στις ρυθμιζόμενες εταιρείες το κίνητρο να μειώσουν το κόστος, αλλά ταυτόχρονα επιτρέπει στις τιμές και τα έσοδα να το υπερβούν προσωρινά.

Η χρέωση βάσει κινήτρων στη μεταφορά και διανομή ηλεκτρικής ενέργειας, καθώς και σε άλλες ρυθμιζόμενες τομείς της οικονομίας, εφαρμόζεται συνήθως με τη θέσπιση ανωτάτων ορίων τιμών. Οι τιμές καθορίζονται με στόχο να καλύψουν το ιστορικό κόστος, συν μια απόδοση επί της επένδυσης, μείον ένα ορισμένο τμήμα X αυτού του κόστους. Η ρυθμιζόμενη εταιρεία μεταφοράς δικαιούται να κρατήσει όλα τα επιπρόσθετα κέρδη εάν το κόστος περισσότερο του X . Ωστόσο, είναι υποχρεωμένη να αναλάβει και όλες τις απώλειες εάν το κόστος δεν μειωθεί

περισσότερο από X . Τυπικά, η ρύθμιση κινήτρων εφαρμόζεται μέσω μίας φόρμουλας «RP1- X » η οποία επιτρέπει ετήσιες αυξήσεις των τιμών κατά $X\%$ μονάδες κάτω του πληθωρισμού (έτσι, αν ο πληθωρισμός ισούται με I , η τιμές μεταφοράς αυξάνονται κατά $I-X$). Η μέθοδος αυτή εφαρμόζεται σε αρκετές εταιρείες μεταφοράς.

Άλλοι μηχανισμοί χρέωσης για τη δημιουργία κινήτρων

Η χρέωση βάσει σημείου αναφοράς (benchmark pricing) και βάσει μέτρου σύγκρισης (yardstick pricing): Η τιμή που ορίζεται ισούται με το προβλεπόμενο κόστος παροχής της ίδιας υπηρεσίας από άλλες εταιρείες. Έχει ένα πρακτικό πλεονέκτημα σε σχέση με άλλες μεθόδους, όπως η ρύθμιση βάσει ανωτάτων ορίων όπου το X είναι πολύ δύσκολο να υπολογιστεί. Αυτή η μέθοδος (συγκριτικός ανταγωνισμός) στοχεύει στο να δώσει ένα σημείο αναφοράς το οποίο δεν θα επηρεάζεται από τη ρυθμιζόμενη εταιρεία. Μπορεί να εφαρμοστεί σε μεγαλύτερη κλίμακα στη διανομή ηλεκτρικής ενέργειας (ή και σε άλλες υπηρεσίες, όπως οι υπηρεσίες παροχής ύδρευσης), όπου λειτουργούν αρκετές συγκρίσιμες εταιρείες, απ' ό,τι στο χώρο της μεταφοράς. Σε ορισμένες όμως περιπτώσεις, μπορεί να εφαρμοστεί μία διεθνής (ή δια-περιφερειακή) καθιέρωση σημείου αναφοράς και για της εταιρείες μεταφοράς.

Άλλη μέθοδος είναι η ρύθμιση αναλογικής κλίμακας (sliding scale regulation), που είναι παρόμοια με τη ρύθμιση βάσει ανωτάτων ορίων, με τη διαφορά ότι η εταιρεία μπορεί να κρατήσει μόνο ένα τμήμα των κερδών που προέκυψαν από τα μέτρα βελτίωσης της αποδοτικότητας. Το μέρος των κερδών που μπορεί να κρατήσει μειώνεται με το ποσό των κερδών που αποκτά, τα οποία και μοιράζονται τυπικά μεταξύ ρυθμιζόμενης εταιρείας και καταναλωτών. Η μέθοδος αυτή αποσκοπεί στο να μειώσει τα κέρδη που έχουν αποκτήσει οι επιχειρήσεις με τη ρύθμιση βάσει ανωτάτων ορίων, και που πολλές φορές θεωρούνται υπέρογκα, δεν έχει όμως εφαρμοστεί στην πράξη.

Η εμπειρία με τις ρυθμίσεις κινήτρων έχει αποδείξει ότι υπάρχει μεγάλο περιθώριο για μείωση του κόστους στις δραστηριότητες μεταφοράς. Στο Ηνωμένο Βασίλειο, λόγω χάρη, η δημιουργία κινήτρων για την National Grid να μειώσει την προσαύξηση (uplift), των επικουρικών δηλαδή υπηρεσιών, έκανε το κόστος να μειωθεί, από 800 εκατομμύρια στερλίνες που ήταν την περίοδο 1994-95, σε 360 εκατομμύρια στερλίνες το 1998. Παρά την εμφανώς ικανοποιητική της απόδοση, η ρύθμιση κινήτρων έχει γίνει στόχος αρνητικής κριτικής από διάφορες πλευρές. Κατ' αρχήν, ορισμένες εταιρείες που υπόκεινται σε περιορισμούς ανωτάτου ορίου έχουν παρουσιάσει μεγάλα κέρδη, πράγμα που σημαίνει ότι το ανώτατο όριο θα μπορούσε να είναι μικρότερο. Ένας συστηματικότερος ρυθμιστικός έλεγχος στη διαμόρφωση των περιορισμών ανωτάτου ορίου και της αναλογικής κλίμακας θα επέτρεπε την ταχύτερη μετακίνηση της μείωσης κόστους στους καταναλωτές, σε βάρος των κινήτρων. Επιπρόσθετα, η ρύθμιση βάσει κινήτρων δέχεται την αρνητική κριτική ότι βασίζεται σε μεγάλο βαθμό στη ρυθμιστική παρέμβαση (π.χ. υπάρχει αναπόφευκτα ένας βαθμός παρέμβασης στον καθορισμό του συντελεστή X όσον αφορά μια ρήτρα τιμής). Η ρυθμιστική παρέμβαση μπορεί να λειτουργήσει σε βάρος των επενδυτών εάν οι συντελεστές X είναι υψηλοί ή πιθανότερα ακόμη και σε βάρος των καταναλωτών του δικτύου εάν οι πιέσεις που ασκούνται στους ρυθμιστές οδηγήσουν σε μικρούς συντελεστές X .

Ένα ακόμη ζήτημα, εκτός από τη ρύθμιση κινήτρων, είναι ότι η ρύθμιση του κόστους υπηρεσιών βασίζεται στο ιστορικό κόστος. Το ιστορικό κόστος είναι οι δαπάνες που έγιναν κατά το παρελθόν και οι οποίες, υπό τις παρούσες συνθήκες, μπορεί να είναι οικονομικά ασύμφωρες. Ως αποτέλεσμα, όταν οι τιμές βασίζονται στο ιστορικό κόστος, ίσως να μην μπορέσουν να δώσουν τα σωστά μηνύματα στους επενδυτές, καθώς και στους αγοραστές και πωλητές ηλεκτρικής ενέργειας. Το ζητούμενο είναι οι τιμές να βασίζονται στο οριακό κόστος παροχής της υπηρεσίας υπό τις παρούσες συνθήκες, και όχι βάσει του τι γινόταν στο παρελθόν και υπό άλλες συνθήκες [15-18].

Ο ρόλος του διαχειριστή του συστήματος στο σχεδιασμό και αναβάθμιση του δικτύου.

Ένα πολύ σημαντικό και, εν μέρει άλυτο ακόμη, ζήτημα είναι να προσδιορίσουμε το ρόλο που παίζει μακροπρόθεσμα ο διαχειριστής του συστήματος στη μεταφορά, δηλαδή στο σχεδιασμό και την υλοποίηση των επενδύσεων στο δίκτυο. Επικρατεί ευρέως η άποψη ότι ο διαχειριστής του συστήματος (ή οποιοδήποτε άλλο επίσημα αρμόδιο πρόσωπο) πρέπει να διατηρήσει ορισμένες αρμοδιότητες σε θέματα που αφορούν στο σχεδιασμό και την αναβάθμιση του δικτύου.

Τα κίνητρα για επενδύσεις στο δίκτυο είναι δυνατόν να διαστρεβλωθούν με διάφορους τρόπους. Η χειραγώγηση της αγοράς μπορεί να μειώσει τα κίνητρα για επενδύσεις (π.χ. οι συμφορήσεις συνεπάγονται επιπλέον κόστος για τους παραγωγούς). Ο κίνδυνος του φαινομένου που στις χρηματιστηριακές αγορές ονομάζεται *free riding*¹ μπορεί να αποθαρρύνει τις επενδύσεις. Για παράδειγμα, οποιαδήποτε μελλοντική επένδυση σε έργα αναβάθμισης μπορεί να μειώσει σημαντικά τις αποδόσεις παλαιότερων επενδύσεων, λόγω μεγάλων διακυμάνσεων στις τιμές, αποθαρρύνοντας έτσι την αναβάθμιση του δικτύου. Τα κίνητρα για συντήρηση και αντικατάσταση των εγκαταστάσεων μπορεί να είναι ανεπαρκή, αν δεν καθιερωθούν οι κατάλληλες επιβραβεύσεις και πρόστιμα αντίστοιχα για την ενίσχυση της ασφάλειας και της αξιοπιστίας. Οι αντιδράσεις σε σχέση με περιβαλλοντικές λόγους, οι χρονοβόρες διοικητικές διαδικασίες, και άλλοι εξω-οικονομικοί παράγοντες, μπορεί να προκαλέσουν επιπρόσθετες δαπάνες και καθυστερήσεις στην ανάπτυξη των εγκαταστάσεων μεταφοράς.

Αυτό σημαίνει ότι οι ορμώμενες εκ της αγοράς επενδύσεις μπορούν από μόνες τους να προκαλέσουν μικρή και ανεπαρκή επενδυτική δραστηριότητα για την επέκταση του δικτύου και ότι ο διαχειριστής του συστήματος πρέπει να διατηρήσει ορισμένες αρμοδιότητες στο σχεδιασμό και την αναβάθμιση του δικτύου. Ακατάλληλες ή άκαιρες επενδύσεις στο δίκτυο μπορεί να μειώσουν σημαντικά την αποδοτικότητα της βιομηχανίας ηλεκτρισμού και να έχουν αρνητικές επιπτώσεις στην ανάπτυξη του ανταγωνισμού. Η στρατηγική σημασία του δικτύου για την διευκόλυνση του ανταγωνισμού έγκειται στο ότι, για να είναι αποτελεσματική οποιαδήποτε προσέγγιση του ανταγωνισμού, πρέπει να δίνει προτεραιότητα στις σωστές χωρικά και χρονικά επενδύσεις σε εγκαταστάσεις μεταφοράς, και πρέπει ακόμη να διαμορφώνει τα κατάλληλα κίνητρα για την όσο το δυνατόν πιο άμεση εξάλειψη των συμφορήσεων στη μεταφορά. Με το ίδιο σκεπτικό, όλα τα απελευθερωμένα συστήματα αναθέτουν αρμοδιότητες για το σχεδιασμό της μεταφοράς στον διαχειριστή του δικτύου.

Οι προσεγγίσεις, εντούτοις, διαφέρουν ως προς τις υπόλοιπες αρμοδιότητες που ανατίθενται στο διαχειριστή του συστήματος. Ο ρόλος του διαχειριστή προσδιορίζεται βάσει δύο γενικότερων προσεγγίσεων:

Από τη μία πλευρά, ο διαχειριστής του συστήματος μπορεί να αποτελέσει ένα μονοπώλιο μεταφοράς που του ανήκει ολόκληρο το δίκτυο μεταφοράς και αναλαμβάνει την υποχρέωση να παρέχει απεριόριστες υπηρεσίες μεταφοράς, εκείνες δηλαδή τις υπηρεσίες που απαιτούνται για την αποτελεσματική λειτουργία του συστήματος. Η προσέγγιση αυτή έχει υιοθετηθεί από τα περισσότερα συστήματα ηλεκτρικής ενέργειας της Ευρώπης, μεταξύ των οποίων η Μεγάλη Βρετανία, η Νορβηγία, η Σουηδία, η Φινλανδία και η Ισπανία. Σύμφωνα με αυτήν την προσέγγιση, ο διαχειριστής του δικτύου είναι αρμόδιος για το σχεδιασμό της δικτυακής αναβάθμισης και της διαχείρισής της. Ουσιαστικά, η προσέγγιση αυτή αντιστοιχεί στο μοντέλο του ιδιοκτησιακού διαχωρισμού.

Από την άλλη πλευρά, ο διαχειριστής του συστήματος μπορεί να είναι ως ο αναπληρωματικός φορέας παροχής υπηρεσιών, που δεν είναι ιδιοκτήτης του δικτύου και επιτρέπει στους συμμετέχοντες στο χώρο της αγοράς να εμπορεύονται δικαιώματα μεταφοράς και να επενδύουν σε πάγιες εγκαταστάσεις μεταφοράς. Αυτή είναι η προσέγγιση που επικρατεί στις Η.Π.Α., σύμφωνα με την οποία ο διαχειριστής του συστήματος εξακολουθεί να έχει κάποιο ρόλο στο σχεδιασμό, αλλά δεν υλοποιεί τις επενδύσεις, ή τουλάχιστον το κάνει μόνο ως έσχατη λύση. Στην πράξη, η προσέγγιση αυτή αντιστοιχεί στο μοντέλο του διαχειριστικού διαχωρισμού.

¹ αγορά και άμεση πώληση μετοχών, στην ήδη αυξημένη τιμή τους, χωρίς να έχει ακόμη καταβληθεί η αξία της αρχικής αγοράς.

Επιμερισμός του κόστους στους καταναλωτές

Ένα ακόμη ζήτημα που αφορά στη χρέωση των υπηρεσιών του δικτύου είναι ο επιμερισμός των χρεώσεων στους καταναλωτές της μεταφοράς, ούτως ώστε να συγκεντρωθούν τα προβλεπόμενα για τη μεταφορά έσοδα. Οι χρεώσεις αυτές απαιτούνται συχνά για να καλύπτεται το έλλειμμα των εσόδων που προκύπτει από τα πάγια κόστη που δεν περιλαμβάνονται στο κόστος συμφόρησης. Στις περιπτώσεις όπου δεν υπάρχει κομβική χρέωση, η υποτιμολόγηση της συμφόρησης μπορεί επίσης να συμβάλει στη δημιουργία ελλείμματος.

Το κόστος, λίγο μπορεί να προσφέρει ως κριτήριο επιμερισμού αυτών των χρεώσεων στους καταναλωτές του δικτύου, εξαιτίας της πάγιας φύσης του. Άλλα είναι τα κριτήρια και οι κανόνες που πρέπει να εφαρμοστούν για να καθοριστούν οι τιμές για τις υπηρεσίες χρήσης του δικτύου. Τέτοιοι κανόνες μπορεί να είναι, για παράδειγμα, η καθιέρωση χρεώσεων ανάλογα με τη ζήτηση κάθε καταναλωτή (*ομοιόμορφη χρέωση*), με την ελαστικότητα ως προς την τιμή (*Χρέωση Ramsey-Boiteaux*), με το οριακό κόστος (*Κανόνας Allais*), ή με τα διαφυγόντα κέρδη ή τη δυσκολία πρόσβασης για κάθε καταναλωτή (*Κανόνας Oftel*). Ο συντελεστής αναλογίας προσαρμόζεται ώστε να αποφέρει τα επιθυμητά έσοδα. Για το λόγο αυτό, οι κανόνες αυτοί είναι γνωστοί και ως *χρέωση πλήρως κατανεμημένου κόστους*. Αν και οι κανόνες αυτοί είναι σχετικά απλοί στην εφαρμογή τους, η τεχνητή σύνδεση κόστους και χρεώσεων μπορεί να μην έχουν διάρκεια και να ευνοήσουν την εμφάνιση σταυροειδών επιδοτήσεων και ή άλλου είδους στρεβλώσεων. Παράλληλα, έχει προταθεί η εφαρμογή κάποιων ειδικών κανόνων που θα ισχύουν στις καθετοποιημένες επιχειρήσεις οι οποίες παρέχουν, εκτός από υπηρεσίες μεταφοράς, παραγωγής ή/και προμήθειας τελικού καταναλωτή. Στο πλαίσιο αυτό, η ρύθμιση των τιμών πρέπει να λαμβάνει υπόψη τα κίνητρα που παρέχονται στις κάθετα συνδεδεμένες αγορές (ιδιαίτερα στον τομέα της παραγωγής). Οι κανόνες αυτοί περιλαμβάνουν τα ακόλουθα: (α) *Κατάλληλη χρέωση των επί μέρους στοιχείων* (ή *κανόνας Baumol-Willig*): Ο κανόνας αυτός στοχεύει στην προώθηση της απόδοσης παραγωγής όσων επιθυμούν πρόσβαση στο δίκτυο. Ο ιδιοκτήτης της μεταφοράς μπορεί να πωλήσει πρόσβαση σε μια τιμή που, όχι μόνο ισοσκελίζει τα έξοδά του, αλλά τον αποζημιώνει και τα διαφυγόντα κέρδη, στις τελικές πωλήσεις του, λόγω του επιπλέον ανταγωνισμού στην αναζήτηση πρόσβασης. Ο κανόνας αυτός, αν εφαρμοστεί μεμονωμένα είναι το ίδιο σαν να επιτρέπει την απεριόριστη μονοπωλιακή χρέωση της πρόσβασης και πρέπει να εφαρμόζεται σε συνδυασμό με κάποιο ανώτατο όριο τιμής για την παραλαμβανόμενη ενέργεια. (β) *Ανώτατα Όρια Τιμής*: Έχει λεχθεί ότι η μεμονωμένη επιβολή ανώτατης τιμής για τη μεταφορά μπορεί να διαταράξει τη χρέωση της ενέργειας. Ένα συνολικό ανώτατο όριο τιμής της ενέργειας αναφέρεται στο σύνολο των υπηρεσιών που συνθέτουν την παροχή του αγαθού της ενέργειας, της μεταφοράς συμπεριλαμβανομένης. Ανατίθενται συντελεστές βαρύτητας για την κάθε επιμέρους υπηρεσία και επιπλέον μια οροφή στη μέση τιμή του συνόλου των υπηρεσιών που συνθέτουν το αγαθό.

3.2. Θεωρία Παιγνίων

Στις απελευθερωμένες αγορές ηλεκτρικής ενέργειας η εφαρμογή της θεωρίας παιγνίων φαίνεται πως όχι μόνο είναι ικανή αλλά ίσως και η πλέον κατάλληλη για να δώσει απαντήσεις σε μια σειρά ζητημάτων που έχουν να κάνουν με την αναδιάρθρωση τόσο από πλευράς ρυθμιστικών αρχών, όπως για παράδειγμα σε θέματα μετάβασης, σχεδιασμού και λειτουργίας του ανταγωνισμού, όσο και από την πλευρά των συμμετεχόντων στη νέα αυτή αγορά σε θέματα στήριξης αποφάσεων και στρατηγικής σε καθαρά επιχειρηματικό και λειτουργικό επίπεδο. Συγκεκριμένα, η Θεωρία Παιγνίων αφορά σε καταστάσεις όπου υπάρχουν πολλοί αποφασίζοντες των οποίων τα συμφέροντα είναι τουλάχιστον μερικώς αντικρουόμενα και οι αποφάσεις που λαμβάνει ο καθένας από τους συμμετέχοντες έχει επίπτωση στο τελικό αποτέλεσμα των υπολοίπων. Ωστόσο, οι αποφασίζοντες, ή παίκτες, δεν έχουν πάντοτε πλήρη έλεγχο στην έκβαση του παιγνίου καθώς μερικές φορές υπάρχουν αβεβαιότητες οι οποίες μπορεί να επηρεάσουν το παίγνιο με απρόβλεπτο τρόπο. Υπό αυτές τις συνθήκες το αποτέλεσμα βασίζεται εν μέρει σε δεδομένα που δεν είναι ακόμη γνωστά στους συμμετέχοντες και δεν διαμορφώνεται μόνο από τις επιλογές των υπολοίπων παικτών.

Υπάρχουν πολλές κατηγορίες και είδη παιγνίων, που έχουν να κάνουν με μια σειρά χαρακτηριστικών, κανόνων ή ιδιαιτεροτήτων, όπως ο τρόπος διεξαγωγής του παιγνίου (αν είναι επαναλαμβανόμενο ή στατικό), η συνεργασία μεταξύ των παικτών (cooperative ή non-cooperative), ο βαθμός πληροφορίας που έχουν οι παίκτες είτε για τα δεδομένα των αντιπάλων τους είτε για τις προηγούμενες επιλογές και αποτελέσματά τους (perfect ή imperfect information) κ.α. Στη συνέχεια ακολουθεί μια σύντομη αναφορά στους σημαντικότερες έννοιες και τύπους παιγνίων που χρησιμοποιούνται στην παρούσα διατριβή [19-32].

3.2.1. Σημείο Ισορροπίας κατά Nash και οι λύσεις Pareto

Μια από τις σημαντικότερες έννοιες στη Θεωρία Παιγνίων αλλά και γενικότερα στην οικονομική θεωρία είναι αυτή της ισορροπίας κατά Nash. Ένα παίγνιο λέμε ότι έχει λύση ισορροπίας κατά Nash (*Nash Equilibrium*) όταν υπάρχει λύση όπου κανείς από τους παίκτες δεν μπορεί να βελτιώσει το αποτέλεσμά του αλλάζοντας μονομερώς τη στρατηγική του. Με άλλα λόγια η θέση ισορροπίας κατά Nash αντιπροσωπεύει για κάθε παίκτη την καλύτερη δυνατή επιλογή του, ασχέτως των επιλογών των αντιπάλων του. Σε ένα παίγνιο είναι δυνατόν να υπάρχουν περισσότερα από ένα τέτοια σημεία ισορροπίας ή και να μην υπάρχουν καθόλου.

Συγκεκριμένα, σε ένα παίγνιο² με δύο συμμετέχοντες (παίκτης 1, παίκτης 2) με συναρτήσεις $f_1(x,y)$ και $f_2(x,y)$ αντίστοιχα όπου x είναι η μεταβλητή απόφασης του παίκτη 1 και y η μεταβλητή απόφασης του παίκτη 2, το σημείο ισορροπίας κατά Nash ορίζεται από το ζεύγος (x^*,y^*) για το οποίο ισχύουν οι ακόλουθες σχέσεις:

$$f_1(x^*,y^*) \leq f_1(x,y^*) \quad \forall x \in X \quad (1)$$

$$f_2(x^*,y^*) \leq f_2(x^*,y) \quad \forall y \in Y \quad (2)$$

όπου X και Y οι χώροι από όπου επιλέγονται τα x και y αντίστοιχα.

Το σύνολο των βέλτιστων επιλογών ενός παίκτη για κάθε πιθανή επιλογή του αντιπάλου ορίζουν την *Καμπύλη Αντίδρασης* (Reaction Curve) του παίκτη. Στο παίγνιο των δύο παικτών οι καμπύλες αντίδρασης \mathcal{R}_1 και \mathcal{R}_2 , εφόσον οι συναρτήσεις f_1 και f_2 είναι συνεχώς διαφορίσιμες, ορίζονται από τις σχέσεις:

$$\frac{\partial f_1(x,y)}{\partial x} = 0 \quad | \quad x(y) \in \mathcal{R}_1 \quad (3)$$

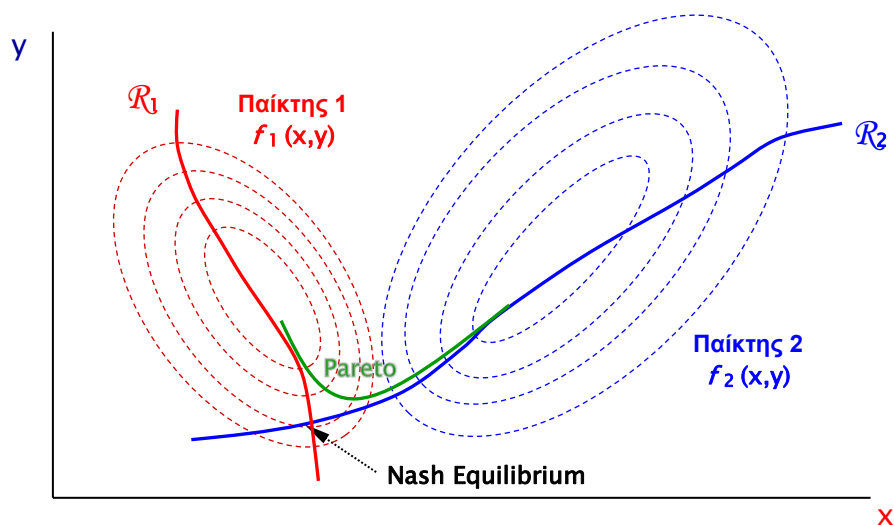
$$\text{και} \quad \frac{\partial f_2(x,y)}{\partial y} = 0 \quad | \quad y(x) \in \mathcal{R}_2 \quad (4)$$

Η τομή των δύο καμπυλών ορίζει και το σημείο της ισορροπίας κατά Nash (Σχ.1)

² Θεωρούμε ότι το παίγνιο είναι μη συνεργατικό (non cooperative) και ότι επιδίωξη των παικτών είναι η ελαχιστοποίηση του αποτελέσματός τους.

Σαν τέτοια παίγνια θα μπορούσαν να μοντελοποιηθούν οι αγορές όπου όλοι οι συμμετέχοντες «παίζουν» ανταγωνιστικά και ενδιαφέρονται πρωτίστως να εξασφαλίσουν ένα ελάχιστο όφελος, ανεξαρτήτως των επιλογών των αντιπάλων τους.

Υπάρχουν, ωστόσο, παίγνια όπου οι παίκτες θα μπορούσαν από κοινού να βελτιώσουν το αποτέλεσμα τους επιλέγοντας κάποια άλλη στρατηγική ο καθένας, με την προϋπόθεση βέβαια ότι υπάρχει εμπιστοσύνη μεταξύ τους και ότι συμφωνούν να προβούν σε κάποια μορφή συνεργασίας (cooperative games). Τα ζεύγη των στρατηγικών αυτών ονομάζονται λύσεις *Pareto* και έχουν σα χαρακτηριστικό ότι πέρα από αυτές καμία άλλη κοινή απόφαση των παικτών δεν μπορεί να βελτιώσει περαιτέρω το αποτέλεσμα του ενός χωρίς ταυτόχρονα να ζημιώσει τον άλλον. Οι λύσεις Pareto αν και έχουν καλύτερο αποτέλεσμα για τους συμμετέχοντες, δεν αποτελούν σημείο ισορροπίας στα μη συνεργατικά παίγνια καθώς αν κάποιος παίκτης ακολουθήσει μονομερώς μια άλλη στρατηγική προς όφελός του, κάποιοι από τους υπόλοιπους θα ζημιωθούν.

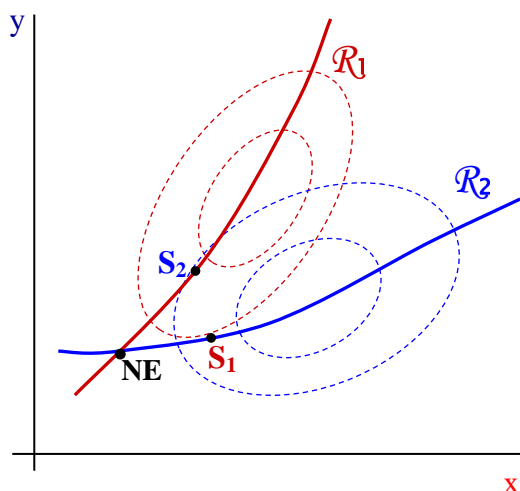


Σχήμα 1. Παράσταση των καμπυλών αντίδρασης δύο παικτών, του σημείου ισορροπίας κατά Nash και των λύσεων Pareto σε δισδιάστατο διάγραμμα καμπυλών ισοδύναμου αποτελέσματος.

Τέτοιες λύσεις μπορεί να εμφανισθούν σε αγορές όπου υπάρχει συνεργασία μεταξύ των συμμετεχόντων ή έστω μιας ομάδας εξ'αυτών με στόχο να πετύχουν καλύτερο αποτέλεσμα από αυτό που θα πετύχαιναν αν δρούσαν ανταγωνιστικά (π.χ. υψηλότερες τιμές, χειραγώγηση της αγοράς κτλ.). Τέτοια φαινόμενα είναι δυνατόν να παρατηρηθούν σε ολιγοπωλιακές αγορές όπως αυτές της ηλεκτρικής ενέργειας [33-35].

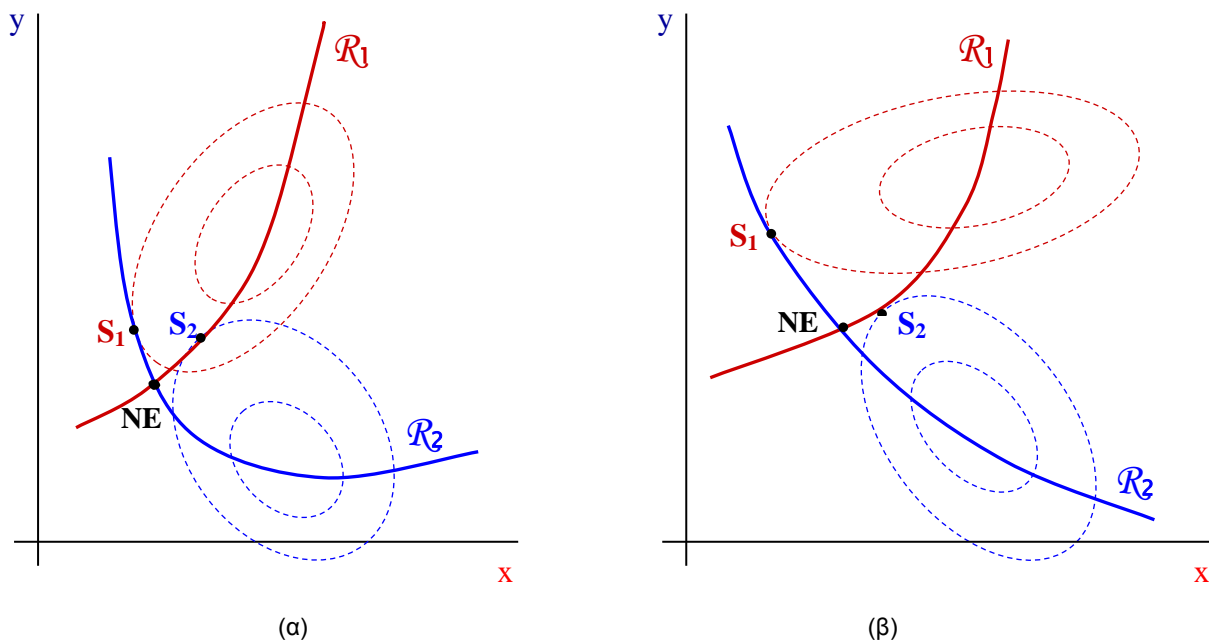
3.2.2. Παίγνιο Stackelberg

Τα παίγνια στα οποία αναφερθήκαμε προϋποθέτουν ότι η επιλογή των στρατηγικών γίνεται και ανακοινώνεται ταυτόχρονα για όλους τους παίκτες. Στην περίπτωση, όμως, που για κάποιο λόγο ένας παίκτης δηλώνει πρώτος τη στρατηγική του –*αρχηγός* (leader) - επιβάλλοντάς την ταυτόχρονα στους υπόλοιπους παίκτες – *ακόλουθοι* (followers) - λέμε ότι έχουμε ένα παίγνιο *Stackelberg*. Σε ένα τέτοιο παίγνιο μη μηδενικού αθροίσματος με δύο παίκτες, η λύση Stackelberg βρίσκεται στο σημείο εκείνο (S_1) της καμπύλης αντίδρασης του *ακόλουθου*, το οποίο έχει το καλύτερο αποτέλεσμα για τον *αρχηγό* (Παίκτης 1). Για τον *αρχηγό* η επιλογή της στρατηγικής που αντιστοιχεί στο σημείο S_1 είναι πάντοτε προτιμότερη από αυτή του σημείου ισορροπίας Nash, ενώ για τον *ακόλουθο*, η επιλογή στρατηγικής που του επιβάλλεται από τον *αρχηγό* δεν έχει απαραίτητα δυσμενέστερο αποτέλεσμα για αυτόν, από το σημείο ισορροπίας Nash [36].



Σχήμα 2. Παίγνιο Stackelberg όπου και οι δύο παίκτες έχουν συμφέρον να αναλάβουν το ρόλο του ακολούθου.

Σε μερικά παίγνια όπου υπάρχει δυνατότητα επιλογής από την πλευρά των παικτών για το ποιος θα είναι αρχηγός και ποιος ακόλουθος, τίθεται το ερώτημα για το ποιος ρόλος είναι επωφελέστερος, καθώς αυτό δεν είναι πάντα προφανές. Για παράδειγμα, στο παίγνιο που απεικονίζεται στο Σχήμα 2, και οι δύο παίκτες προτιμούν να αναλάβουν το ρόλο του ακολούθου (*follower*) καθώς το αποτέλεσμα τους βελτιώνεται όταν ο αντίπαλος αναλαμβάνει την πρωτοβουλία των κινήσεων. Υπάρχουν, ωστόσο, περιπτώσεις όπου συμφέρει και τους δύο παίκτες να αναλάβει το ρόλο του αρχηγού (*leader*) συγκεκριμένα ο ένας από τους δύο (π.χ. στο Σχήμα 3.α η ανάληψη της αρχηγίας από τον Παίκτη 2 συμφέρει και τους δύο), ενώ σε άλλα παίγνια όπου και οι δύο παίκτες επιδιώκουν να έχουν αυτοί το ρόλο του αρχηγού (Σχήμα 3.β).



Σχήμα 3. Παίγνιο Stackelberg όπου και οι δύο παίκτες έχουν συμφέρον (α) να αναλάβει συγκεκριμένος παίκτης το ρόλο του αρχηγού (β) να αναλάβουν οι ίδιοι το ρόλο του αρχηγού.

Το παίγνιο τύπου *Stackelberg* παρουσιάζει ιδιαίτερο ενδιαφέρον στην περίπτωση των αγορών της ηλεκτρικής ενέργειας, καθώς συναντώνται σε αυτές κάποιες επιχειρήσεις οι οποίες επειδή ακριβώς είναι αυτές που ασκούσαν τη δραστηριότητά τους μονοπωλιακά πριν την αναδιάρθρωση της αγοράς, διαθέτουν κάποια πλεονεκτήματα έναντι των υπολοίπων, και ως εκ τούτου μπορούν να εκληφθούν ως ο *αρχηγός* στο παίγνιο που ορίζεται από τους κανόνες ανταγωνισμού της αγοράς. Τα πλεονεκτήματα τέτοιων επιχειρήσεων μπορεί να είναι το μέγεθος, το ευρύτερο φάσμα επιλογών τεχνολογίας ηλεκτροπαραγωγής, η αναγνωρισιμότητα του εμπορικού σήματος και η ύπαρξη πελατολογίου, καθώς και η δυνατότητα επηρεασμού και άσκησης πίεσης προς όφελος τους, τουλάχιστον στα πρώτα και μεταβατικά στάδια της αναδιάρθρωσης.

4. Το Μοντέλο Προσομοίωσης της Αγοράς Ηλεκτρικής Ενέργειας

4.1. Το Μοντέλο ενός Συστήματος Ηλεκτρικής Ενέργειας.

4.1.1. Γενική Περιγραφή του Μοντέλου

Το μοντέλο που αναπτύχθηκε προσομοιώνει μια απελευθερωμένη αγορά ηλεκτρικής ενέργειας, αποτελούμενη από Παραγωγούς ηλεκτρικής ενέργειας και από έναν Ανεξάρτητο Διαχειριστή του Συστήματος, όπου μια διαδικασία υποβολής προσφορών από πλευράς παραγωγών διαμορφώνει την τιμή της ηλεκτρικής ενέργειας. Σε πρώτη φάση, θεωρούμε πως κάθε παραγωγός γνωρίζει μόνο το δικό του κόστος παραγωγής, τις προηγούμενες προσφορές του και τα αντίστοιχα αποτελέσματα που αυτές είχαν. Οι παραγωγοί χρησιμοποιούν ένα στοχαστικό προσαρμοστικό αλγόριθμο εκμάθησης (adaptive learning algorithm) με στόχο την μεγιστοποίηση του κέρδους τους. Συγκεκριμένα, κάθε παραγωγός προσαρμόζει την προσφορά του, μεταβάλλοντας τους συντελεστές της συνάρτησης προσφοράς του βάσει κάποιων τιμών που επιλέγονται από ένα στοχαστικό προφίλ συμπεριφοράς που αντιστοιχεί στον κάθε παραγωγό. Το προφίλ αυτό αναμορφώνεται συνεχώς και βαθμιαία εκτιμώντας την επίπτωση της τελευταίας μεταβολής της συνάρτησης προσφοράς, στο εισόδημα του παραγωγού. Στην πραγματικότητα πρόκειται για ένα παίγνιο Nash όπου οι παίκτες δεν γνωρίζουν τα κόστη, τις επιλογές και τα αποτελέσματα των αντιπάλων τους και για το λόγο αυτό χρησιμοποιούν ένα προσαρμοστικό στοχαστικό μαθησιακό μοντέλο για να αντισταθμίσουν αυτή την έλλειψη πληροφορίας.

Το ενδιαφέρον εδώ, επικεντρώνεται στο γεγονός ότι μπορούν να μελετηθούν, υπό συνθήκες βέβαια, διάφορες παράμετροι του Συστήματος όπως ο αριθμός των παραγωγών που συμμετέχουν στην αγορά και η επίδρασή τους στην διαμόρφωση της τιμής, ο τρόπος που οι παραγωγοί μεταβάλλουν τις προσφορές τους, η επίδραση πιθανών περιορισμών από την πλευρά του Διαχειριστή, η επιβολή διαφορετικού συστήματος τιμολόγησης, η επίδραση της αλλαγής του μίγματος των παικτών, διακρινόμενων ως προς διάφορα χαρακτηριστικά, (παραγωγική ικανότητα, κόστος παραγωγής, επιδιώξεις, ικανότητα προσαρμογής κ.α.).

Σε αυτή τη φάση ωστόσο επιδιώχθηκε περισσότερο η διαμόρφωση του μοντέλου και κάποιες πειραματικές εκτελέσεις, παρά η συστηματική εξέταση των διαφόρων παραμέτρων, που επηρεάζουν τη λειτουργία μιας αγοράς, προς εξαγωγή τεκμηριωμένων συμπερασμάτων. Ωστόσο, ελέγχθηκαν διεξοδικά κάποιες περιοχές έρευνας (αριθμός συμμετεχόντων και συμπεριφορά της τιμής, σύγκριση διαφορετικών μεθόδων τιμολόγησης, περιορισμός ανώτατου ορίου προσφοράς, τάσεις διαμόρφωσης των προσφορών σε παραγωγούς με τετραγωνικό και γραμμικό οριακό κόστος), με ομοιόμορφο όμως μίγμα παραγωγών και εξήχθησαν τα αντίστοιχα συμπεράσματα τα οποία παρατίθενται στη συνέχεια.

Το Σύστημα Ηλεκτρικής Ενέργειας που μοντελοποιήθηκε αποτελούν :

α) ένας **Ανεξάρτητος Διαχειριστής του Συστήματος** στον οποίο υποβάλλονται οι προσφορές και ο οποίος αγοράζει ποσότητες από τους παραγωγούς με στόχο την κάλυψη της ζήτησης D .

β) i **Παραγωγοί Ηλεκτρικής Ενέργειας** (παίκτες) που υποβάλλουν τις προσφορές τους υπό μορφή κάποιας συνάρτησης αλλά δεν γνωρίζουν τα κόστη και τις προσφορές των υπολοίπων.

Ο τρόπος με τον οποίο οι παραγωγοί πληρώνονται από τον Διαχειριστή αποτελεί καθοριστικό παράγοντα για τη συμπεριφορά του Συστήματος και στη συνέχεια εξετάζονται δύο διαφορετικές μεθοδολογίες:

α) βάσει της **Οριακής Τιμής του Συστήματος – ΟΤΣ (Uniform Pricing)**, όπου οι παραγωγοί πληρώνονται στην τιμή αυτή (η οποία ορίζεται ως το κόστος της τελευταίας και ακριβότερης kWh που αγοράστηκε από το Διαχειριστή) το σύνολο της ποσότητας που τελικά πωλούν στο Διαχειριστή.

β) βάσει της Προσφοράς τους (**Pay-As-Bid**), όπου οι παραγωγοί πληρώνονται για την ποσότητα που απορροφά από αυτούς ο Διαχειριστής ότι ακριβώς έχουν υποβάλλει ως συνάρτηση προσφοράς τους.

Χαρακτηριστικά των Παραγωγών

Σε κάθε παραγωγό i αντιστοιχεί :

α) μια συνάρτηση οριακού κόστους, για παράδειγμα, τετραγωνικής μορφής³:

$$MC_i(x) = a_i + b_i x + c_i x^2 \quad \text{όπου } a_i, b_i, c_i > 0, \text{ οι συντελεστές κόστους του κάθε παραγωγού και}$$

β) ένα δεδομένο εύρος δυναμικότητας παραγωγής $[x_{i \min}, x_{i \max}]$.

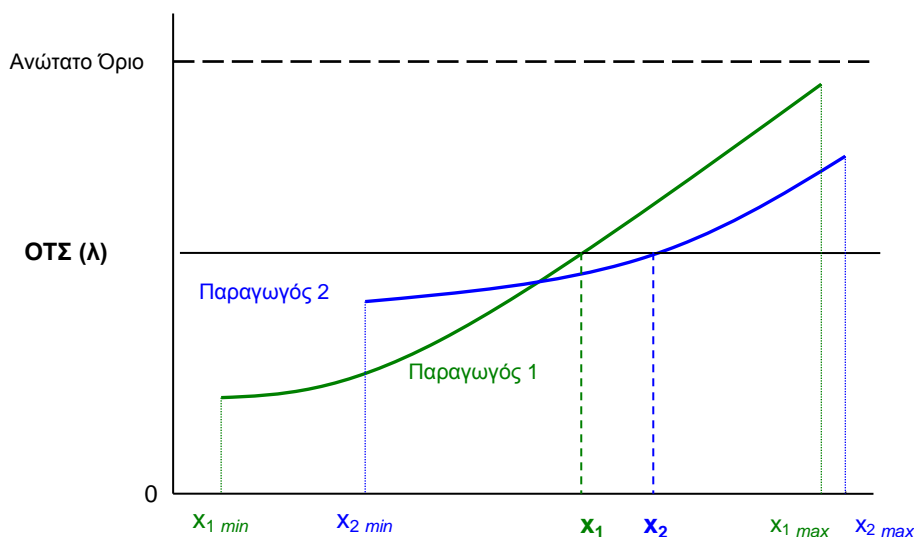
Το σύνολο της δυναμικότητας των παικτών πρέπει να υπερκαλύπτει τη ζήτηση :

$$\sum_{i=1}^i x_{\max_i} \geq D$$

Ανώτατο Όριο Τιμής Προσφοράς

Ο Διαχειριστής του Συστήματος δύναται να ορίζει ένα **Ανώτατο Όριο Τιμής Προσφοράς** (*Price Cap*) το οποίο υπολογίζεται ως κάποιο πολλαπλάσιο της τιμής που θα ισορροπούσε το σύστημα αν όλοι οι παίκτες είχαν υποβάλλει ως προσφορές το κόστος τους (Αρχική Τιμή Ισορροπίας). Σε αυτή την περίπτωση οι παραγωγοί υποβάλλουν προσφορές για το σύνολο της δυναμικότητάς τους με τρόπο τέτοιο ώστε για καμιά προσφερόμενη ποσότητα η τιμή να μην υπερβαίνει το ανώτατο όριο τιμής προσφοράς.

Στο Σχήμα 4 απεικονίζεται ένα Σύστημα με δύο παραγωγούς και σύστημα τιμολόγησης Uniform Pricing, που υποβάλλουν προσφορές για το σύνολο της δυναμικότητάς τους ($x_{1 \max}$ και $x_{2 \max}$) και ο Διαχειριστής καλύπτει τη ζήτηση D αγοράζοντας στην Οριακή Τιμή του Συστήματος, αντίστοιχα x_1 και x_2 , ώστε $x_1 + x_2 > D$.



Σχήμα 4. Προσδιορισμός της Οριακής Τιμής και των μεριδίων αγοράς σε ένα Σύστημα με δύο Παραγωγούς με τετραγωνικό Οριακό Κόστος και σύστημα τιμολόγησης Uniform Pricing.

Οι παραγωγοί υποβάλλουν προσφορές για n συνεχόμενους γύρους και παραμένουν στο παιχνίδι ακόμα και αν δεν καταφέρνουν να πάρουν μερίδιο αγοράς για μεγάλα χρονικά διαστήματα.

³ Αν και στο λογισμικό και στα περισσότερα παίγνια που διεξήχθησαν, χρησιμοποιήθηκαν **γραμμικές συναρτήσεις οριακού κόστους**, εδώ αναλύεται η γενικότερη και πολυπλοκότερη περίπτωση του τετραγωνικού οριακού κόστους για εποπτικότερη παρουσίαση του προσαρμοστικού μαθησιακού αλγορίθμου.

Έναρξη του Παιγνίου

Κατά την έναρξη του παιγνίου οι παραγωγοί υποβάλλουν για παράδειγμα ως προσφορές την καμπύλη κόστους των προσαυξημένη κατά ένα μικρό ποσοστό e (βήμα) ως προς τους συντελεστές κόστους a_i, b_i, c_i . Το βήμα e προσδιορίζει οποιαδήποτε μεταβολή των παραμέτρων της συνάρτησης προσφοράς των παικτών κατά τη διάρκεια του παιχνιδιού και οι τιμές του ορίζονται πριν την έναρξη του παιχνιδιού. Οι τιμές αυτές δύναται να είναι διαφορετικές ανά παραγωγό και συντελεστή εκφράζοντας έτσι τη διαφορετικότητα τόσο της αντίδρασης των παικτών όσο και της ταχύτητας με την οποία αυτή συντελείται.

Οι συντελεστές :

$$A_i = a_i \cdot (1 + e_i)$$

$$B_i = b_i \cdot (1 + e_i)$$

$$C_i = c_i \cdot (1 + e_i)$$

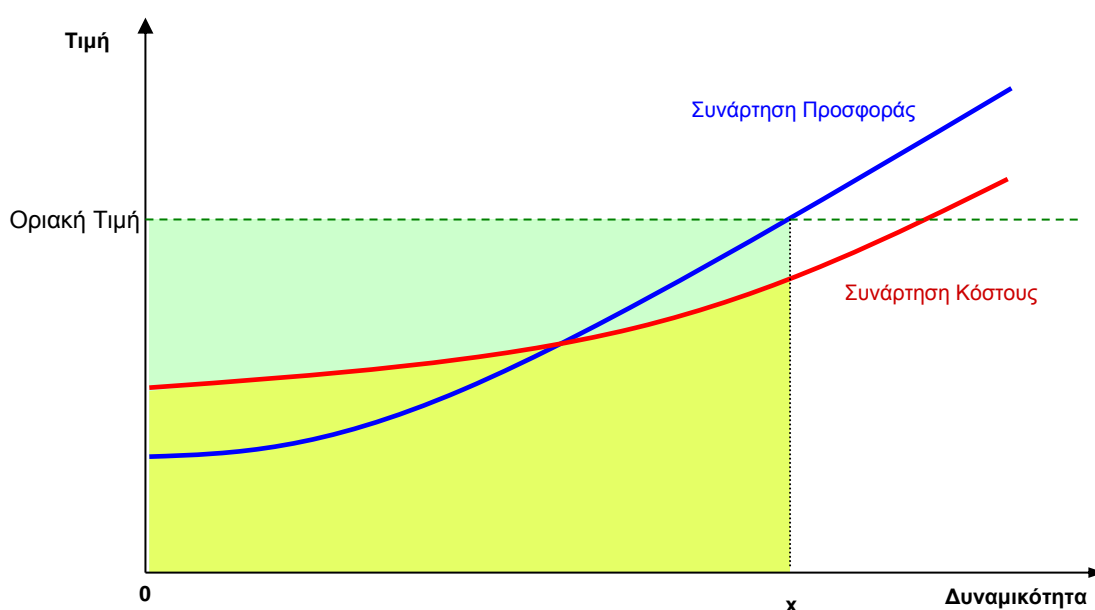
ορίζουν την πρώτη συνάρτηση προσφοράς του κάθε παραγωγού:

$$F_{i_1}(x) = A_{i_1} + B_{i_1} \cdot x + C_{i_1} \cdot x^2$$

Βάσει των προσφορών και των υπολοίπων παικτών προσδιορίζεται η Οριακή Τιμή του Συστήματος (λ_1), οι ποσότητες (x_{i_1}) που αγοράζει ο διαχειριστής από τον κάθε παίκτη, και στη συνέχεια το αντίστοιχο έσοδο (κέρδος) του κάθε παραγωγού (Σχήμα 5):

$$J_{i_1} = x_{i_1} \cdot \lambda_1(x_{i_1}) - \int_0^{x_{i_1}} MC_i(x) dx$$

Στο τέλος κάθε γύρου ο παραγωγός i γνωρίζει μόνο την τιμή ισορροπίας του συστήματος και την ποσότητα που αγόρασε από αυτόν ο διαχειριστής.



Σχήμα 5. Προσδιορισμός Κέρδους για τον Παραγωγό στο Σύστημα με Uniform Pricing Τιμολόγηση

Με το τέλος του πρώτου γύρου και πριν ο παραγωγός υποβάλλει την επόμενη προσφορά του, επιλέγει με τυχαίο τρόπο το είδος της συμπεριφοράς (δράση) που θα ακολουθήσει, δηλαδή αν θα αυξήσει, θα μειώσει ή θα διατηρήσει σταθερούς τους συντελεστές της συνάρτησης προσφοράς. Ο κάθε παραγωγός αλλάζει για έναν συγκεκριμένο αριθμό συνεχόμενων γύρων μόνο έναν εκ των τριών συντελεστών. Η διάρκεια των περιόδων των περιτροπής αλλαγών των συντελεστών είναι διαφορετική ανά παραγωγό και συντελεστή και ορίζεται στην αρχή του παιγνίου, βάσει τιμών που επιλέγονται από ορισμένο διάστημα με τυχαίο τρόπο.

Η τυχαία επιλογή της δράσης για τον κάθε συντελεστή γίνεται βάσει μιας κατανομής πιθανότητας των τριών πιθανών δράσεων (αύξηση, μείωση, σταθεροποίηση) που αντιστοιχούν στο συγκεκριμένο συντελεστή του παίκτη και οι οποίες μπορεί να έχουν αρχικά την ίδια ή και διαφορετική πιθανότητα να επιλεγούν.

Κατά αυτόν τον τρόπο, ανά παίκτη και συντελεστή αντιστοιχούν τρεις τιμές πιθανότητας P^{in} , P^{de} , P^{st} (αύξηση, μείωση και σταθεροποίηση αντίστοιχα) τέτοιες που κάθε φορά να ικανοποιούν τις σχέσεις:

$$P_{iA}^{in} + P_{iA}^{de} + P_{iA}^{st} = 1$$

$$P_{iB}^{in} + P_{iB}^{de} + P_{iB}^{st} = 1$$

$$P_{iC}^{in} + P_{iC}^{de} + P_{iC}^{st} = 1$$

Με την τυχαία επιλογή της δράσης διαμορφώνεται αντίστοιχα η νέα συνάρτηση προσφοράς του παίκτη i , η οποία υποβάλλεται στο διαχειριστή, με συντελεστές προσφοράς (ανάλογα με ποια περίοδο αλλαγής συντελεστή διανύεται στο συγκεκριμένο γύρο) :

περίοδος αλλαγής A_i

περίοδος αλλαγής B_i

περίοδος αλλαγής C_i

$$A_{i_n} = A_{i_{n-1}} \cdot (1 + e_i)$$

$$A_{i_n} = A_{i_{n-1}}$$

$$A_{i_n} = A_{i_{n-1}}$$

$$B_{i_n} = B_{i_{n-1}}$$

$$B_{i_n} = B_{i_{n-1}} \cdot (1 + e_i)$$

$$B_{i_n} = B_{i_{n-1}}$$

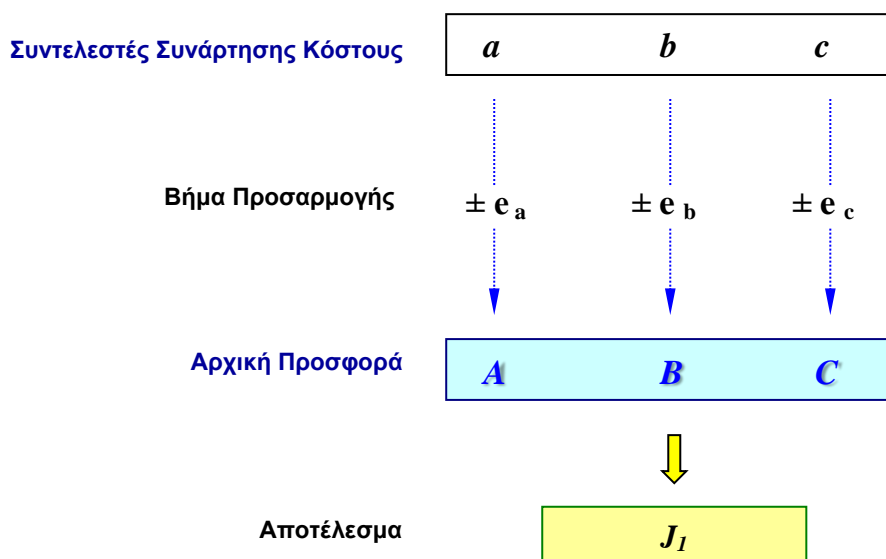
$$C_{i_n} = C_{i_{n-1}}$$

$$C_{i_n} = C_{i_{n-1}}$$

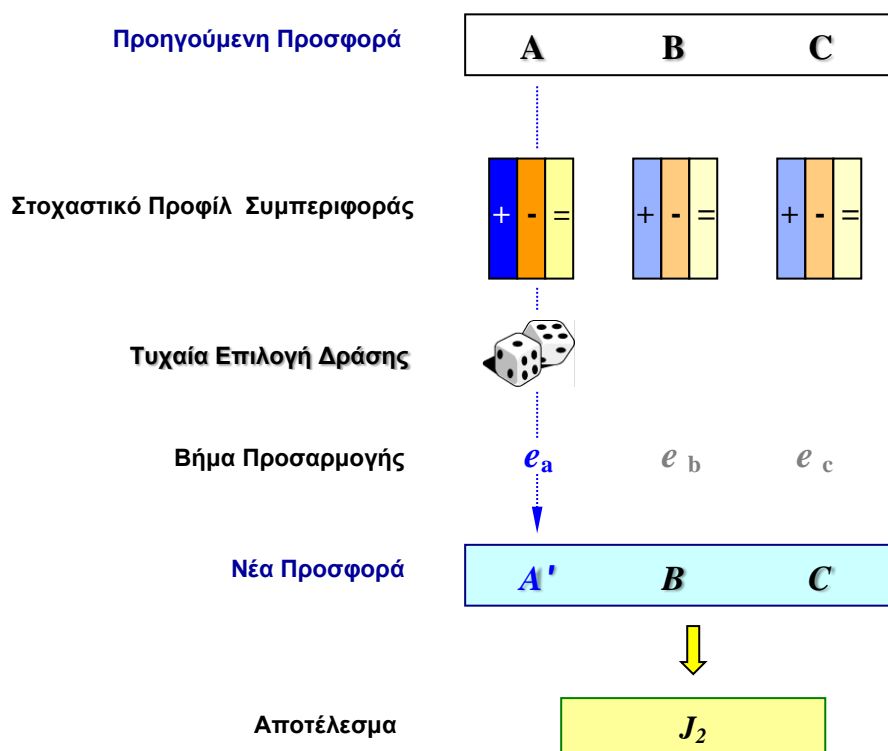
$$C_{i_n} = C_{i_{n-1}} \cdot (1 + e_i)$$

$$\text{όπου } e_i = \begin{cases} e_{A_i}, e_{B_i} \text{ ή } e_{C_i} & \text{αν η επιλεγείσα δράση είναι } \mathbf{αύξηση} \\ 0 & \text{αν η επιλεγείσα δράση είναι } \mathbf{σταθεροποίηση} \\ -e_{A_i}, -e_{B_i} \text{ ή } -e_{C_i} & \text{αν η επιλεγείσα δράση είναι } \mathbf{μείωση} \end{cases}$$

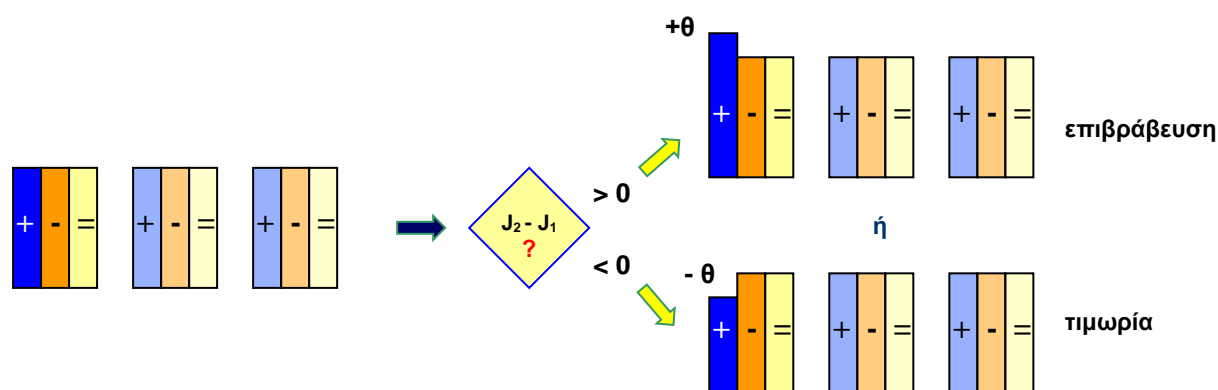
Το προκύπτον οικονομικό αποτέλεσμα J_{i_n} συγκρίνεται με το οικονομικό αποτέλεσμα του αμέσως προηγούμενου γύρου $J_{i_{n-1}}$ και βάσει αυτής της σύγκρισης αναδιαμορφώνεται η υπάρχουσα κατανομή πιθανοτήτων των τριών δράσεων. Συγκεκριμένα, αν η διαφορά $(J_{i_n} - J_{i_{n-1}})$ δύο συνεχόμενων γύρων είναι θετική τότε αυξάνεται η πιθανότητα της δράσης (επιβράβευση) που είχε επιλεγεί στον γύρο n , κατά ένα προκαθορισμένο βήμα θ , εκφρασμένο ως ποσοστό, και αντίστοιχα μειώνονται οι πιθανότητες των δύο άλλων δράσεων (βήμα προσαρμογής στοχαστικού προφίλ). Σε περίπτωση που το οικονομικό αποτέλεσμα είναι δυσμενέστερο από το αντίστοιχο του προηγούμενου γύρου μειώνεται κατά το ποσοστό θ η πιθανότητα της δράσης που είχε επιλεγεί (τιμωρία) και αντίστοιχα αυξάνονται οι πιθανότητες των δύο άλλων δράσεων. Η τιμή του βήματος θ δύναται να είναι διαφορετική για κάθε παίκτη εκφράζοντας έτσι τη διαφορετικότητα στην μαθησιακή ικανότητα κάθε παίκτη [49-54].



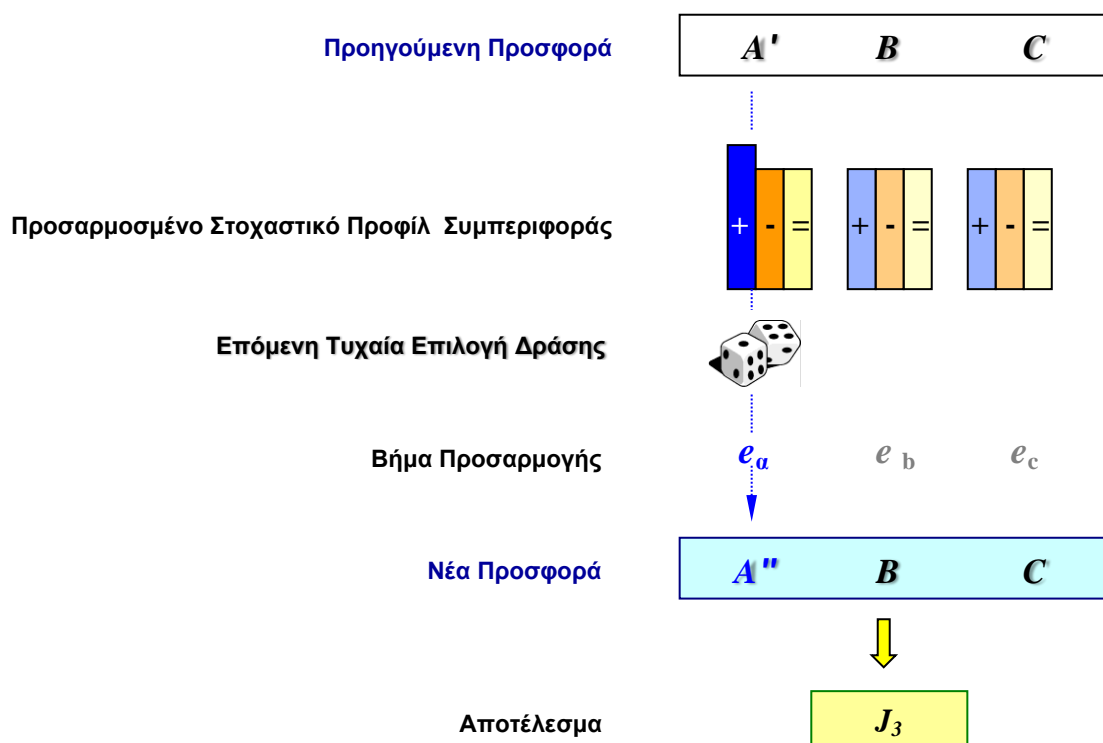
Σχήμα 6.α Η Μαθησιακή Διαδικασία: Η Πρώτη Προσφορά.



Σχήμα 6.β Η Μαθησιακή Διαδικασία: Τυχαία Επιλογή Δράσης για έναν από τους συντελεστές και διαμόρφωση της επόμενης προσφοράς.



Σχήμα 6.γ Η Μαθησιακή Διαδικασία: Σύγκριση των αποτελεσμάτων J_1 και J_2 , επαναπροσδιορισμός του στοχαστικού προφίλ συμπεριφοράς του συντελεστή με τη βοήθεια ενός βήματος θ και επιβράβευση ή τιμωρία της πιθανότητας της δράσης που επιλέχθηκε.



Σχήμα 6.δ Η Μαθησιακή Διαδικασία: Νέα Προσφορά βάσει του προσαρμοσμένου προφίλ συμπεριφοράς.

4.1.2. Μέθοδοι Τιμολόγησης και Προσδιορισμός της Τιμής

Οι μέθοδοι τιμολόγησης είναι δύο όπως αναφέρθηκε πιο πάνω και διαφέρουν μόνο ως προς τον τρόπο που ο Διαχειριστής πληρώνει τους παραγωγούς. Στη μέθοδο της Οριακής Τιμής του Συστήματος (Uniform Pricing) όλοι οι παραγωγοί πληρώνονται το σύνολο της ποσότητας που πωλούν στην ίδια τιμή ανά μονάδα ενέργειας (στην ΟΤΣ), ενώ στη μέθοδο Pay-As-Bid, οι παραγωγοί πληρώνονται βάσει της προσφοράς τους, δηλαδή η κάθε μονάδα ενέργειας πληρώνεται στην τιμή μονάδας όπως αυτή περιγράφεται στην προσφορά του παραγωγού. Αυτό σημαίνει πως στην μεν πρώτη περίπτωση οι προσφορές έχουν νόημα μόνο για τον καθορισμό των ποσοτήτων που θα αγοράσει ο Διαχειριστής από τον κάθε παραγωγό, ενώ στη δεύτερη περίπτωση χρησιμεύουν επιπλέον και για τον προσδιορισμό των πληρωμών στους παραγωγούς. Η σύγκριση των δύο συστημάτων παρουσιάζεται στη συνέχεια της παρούσης διατριβής.

Ο προσδιορισμός των ποσοτήτων γίνεται από το Διαχειριστή του Συστήματος αφού λάβει υπόψη του όλες τις προσφορές των παραγωγών και κάνει την οικονομική ιεράρχησή τους έως ότου ικανοποιηθεί η ζήτηση. Οι περιορισμοί, ωστόσο, που θέτουν οι δηλωμένες δυναμικότητες παραγωγής των παραγωγών τόσο ως προς τη μέγιστη δυναμικότητα παραγωγής αλλά κυρίως ως προς το τεχνικό τους ελάχιστο ενδέχεται να δημιουργήσουν κάποιες ασυνέχειες στη συνάρτηση συνολικής προσφερόμενης ισχύος, με αποτέλεσμα ο Διαχειριστής να είναι αναγκασμένος να αγοράσει τελικά ποσότητες που υπερβαίνουν τη ζήτηση. Για παράδειγμα, ενδέχεται να χρειασθεί να δώσει εντολή κατανομής για παραγωγό στο τεχνικό του ελάχιστο, προκειμένου να καλύψει το τμήμα εκείνο της ισχύος που υπολείπεται για την ικανοποίηση της ζήτησης και το οποίο είναι μικρότερο από το τεχνικό ελάχιστο του εν λόγω παραγωγού. Θα πρέπει να επισημανθεί εδώ, πως σε ένα πραγματικό Ηλεκτρικό Σύστημα υπεισέρχονται και άλλες λεπτομέρειες τεχνικής φύσεως, όπως η παραγωγή για επικουρικές υπηρεσίες, οι απώλειες μεταφοράς κ.α. τα οποία στην παρούσα προσομοίωση δεν λαμβάνονται υπόψη καθώς το εν λόγω έργο δεν αφορά προσομοίωση λειτουργίας του Συστήματος σε τεχνικό επίπεδο αλλά σε επίπεδο οικονομικών συναλλαγών. Στη συνέχεια παρουσιάζονται σχηματικά οι δύο μέθοδοι τιμολόγησης και οι σχέσεις μεταξύ οριακού και συνολικού κόστους παραγωγού με γραμμικό οριακό κόστος και σταθερό κόστος παραγωγής (FC). Η συνάρτηση του συνολικού του κόστους είναι⁴ :

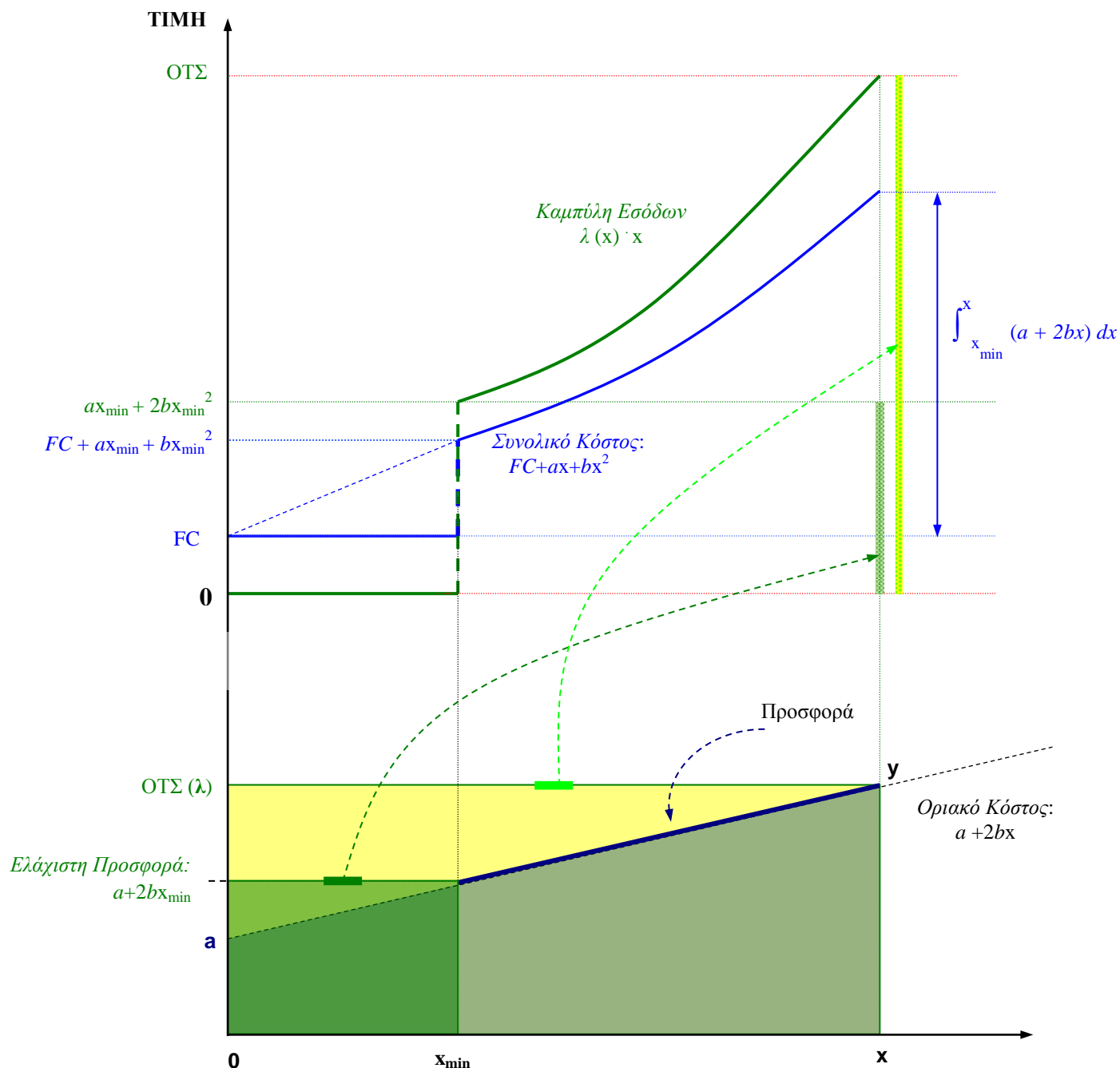
$$TC_i(x) = FC_i + a_i x + b_i x^2 \quad \text{όπου } a_i \text{ όπου } b_i \text{ οι συντελεστές κόστους}$$

το οριακό του κόστος περιγράφεται από τη σχέση:

$$MC_i(x) = a_i + 2b_i x$$

Σημειώνεται εδώ ότι αν και το λογισμικό που αναπτύχθηκε αρχικά ο χρήστης είχε κανείς τη δυνατότητα να επιλέγει τη μορφή της συνάρτησης οριακού κόστους των παραγωγών (γραμμική ή τετραγωνική), επιλέχθηκε τελικά η περίπτωση του γραμμικού οριακού κόστους, που πέραν της μεγαλύτερης απλότητας που προσφέρει, ανταποκρίνεται καλύτερα στην περίπτωση μοντελοποίησης παραγωγού ηλεκτρικής ενέργειας όπως αποδείχθηκε και από τα πειράματα που διεξήχθησαν. Ωστόσο η χρήση της τετραγωνικής μορφής και η διεξαγωγή πειραμάτων με τέτοια καμπύλη οριακού κόστους βοήθησε αρκετά στη μελέτη και την εξαγωγή χρήσιμων συμπερασμάτων για τη συμπεριφορά των παραγωγών σε σχέση με τον τρόπο μεταβολής των προσφορών τους. Οι προσφορές επίσης των παραγωγών είναι ίδια μορφή συνάρτησης με αυτή του οριακού τους κόστους. Στην πραγματικότητα βέβαια στις περισσότερες αγορές ηλεκτρικής ενέργειας οι συναρτήσεις προσφορών είναι κλιμακωτές συναρτήσεις περιορισμένων μάλιστα βαθμίδων (συνήθως τρεις βαθμίδες). Η θεωρητική όμως μελέτη των διαφόρων φαινομένων που εξετάστηκαν στο παρόν ερευνητικό έργο, επέβαλε τη χρήση συνεχών και όχι διακριτών συναρτήσεων προσφοράς.

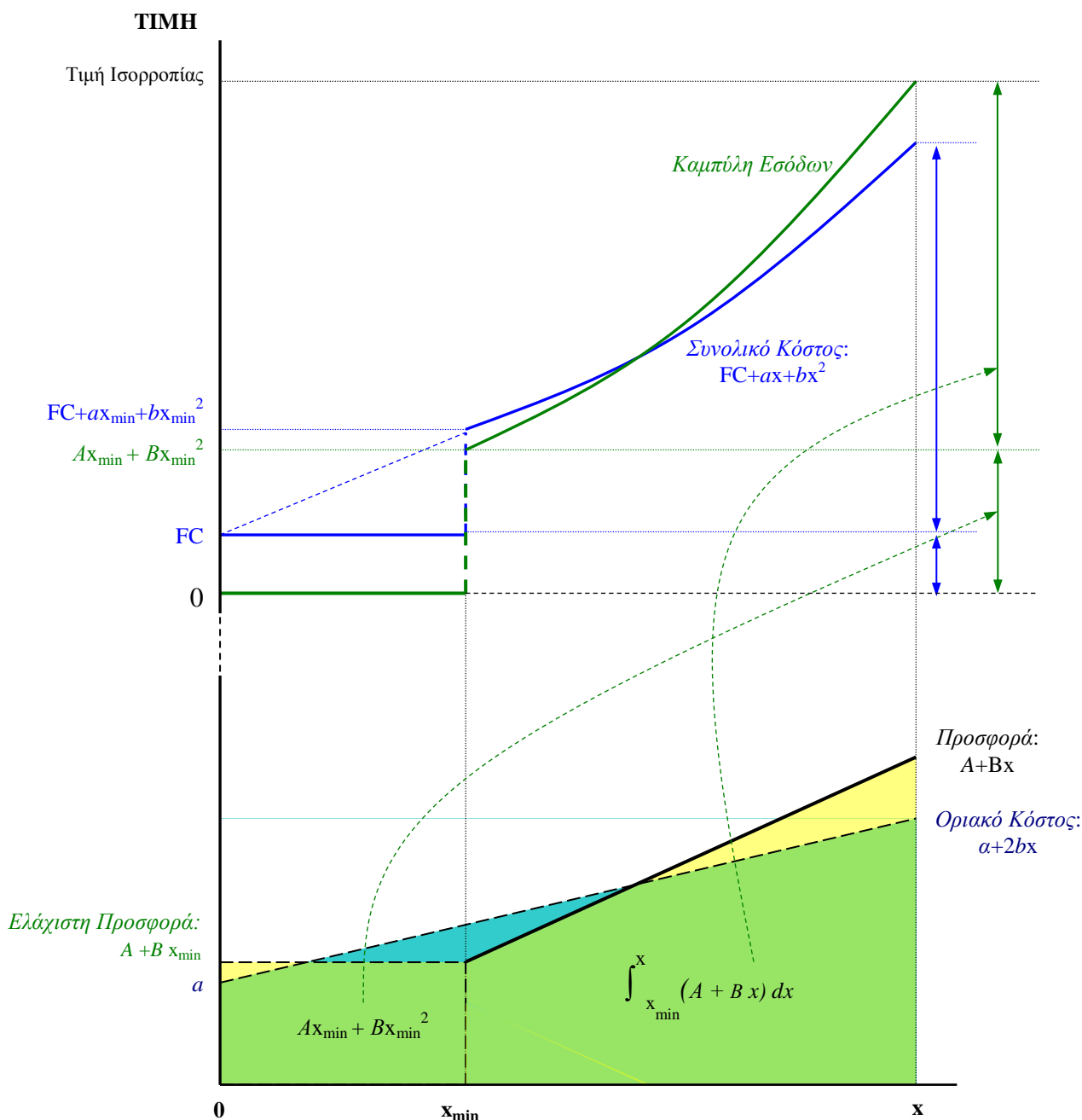
⁴ Χρησιμοποιείται η απλούστερη μορφή μοντέλου με τετραγωνικό συνολικό κόστος παραγωγών και αντίστοιχα γραμμικού οριακού κόστους.



Σχήμα 7. Σχέση Συνολικού και Οριακού κόστους, συνάρτησης προσφοράς και Οριακής Τιμής Συστήματος Παραγωγού με γραμμικό οριακό κόστος σε αγορά ηλεκτρικής ενέργειας με Uniform Pricing.

Ο παραγωγός του Σχήματος 7, υποβάλλει προσφορά ίση με το οριακό του κόστος. Το έσοδό του για παραγόμενη ισχύ ίση με x αναπαριστάται στο κάτω διάγραμμα με το εμβαδόν του παραλληλογράμμου **Oxyλ**, ενώ το αντίστοιχο κόστος από το εμβαδόν του τραπεζιού **Oxyα**. Οποιαδήποτε πώληση ισχύος που έχει τίμημα μεγαλύτερο από το σταθερό του κόστος είναι συμφέρουσα για τον παραγωγό και ο συγκεκριμένος έχει καλύψει το κόστος αυτό ήδη από επίπεδο ισχύος του τεχνικού του ελαχίστου. Είναι προφανές ότι η Uniform Pricing μέθοδος επιτρέπει στον παραγωγό να υποβάλλει και προσφορές που σε κάποια περιοχή του φάσματος δυναμικότητάς του μπορεί να βρίσκονται χαμηλότερα από το οριακό του κόστος, αφήνοντας του έτσι μεγαλύτερα περιθώρια τακτικών ελιγμών.

Αντίθετα, στην Pay-As-Bid τιμολόγηση (Σχήμα 8), ο παραγωγός δεσμεύεται από την προσφορά του, καθώς πληρώνεται βάσει αυτής, οπότε τα περιθώρια ευελιξίας που έχει είναι πολύ πιο περιορισμένα (στη συγκεκριμένη περίπτωση του Σχήματος 8, ο παραγωγός δεν υποβάλλει ως προφορά το οριακό του κόστος). Αν ο παραγωγός υποβάλλει προσφορά κάτω του κόστους του, για κάποια περιοχή της δυναμικότητάς του, ενδέχεται, ανάλογα την ποσότητα που θα απορροφήσει από αυτόν ο Διαχειριστής, να ζημιωθεί. Ως Τιμή Ισορροπίας εδώ ορίζεται απλά η ακριβότερη μονάδα ενέργειας που αγοράστηκε από το Διαχειριστή, χωρίς όμως η τιμή αυτή να αποτελεί τη μέση τιμή ηλεκτρικής ενέργειας του Συστήματος [59-67].



Σχήμα 8. Σχέση Συνολικού και Οριακού κόστους, συνάρτησης προσφοράς και Οριακής Τιμής Συστήματος Παραγωγού με γραμμικό οριακό κόστος σε αγορά ηλεκτρικής ενέργειας με Pay-As-Bid.

4.1.3. Το πρόβλημα κατανομής των παραγωγών και ο αλγόριθμος επίλυσής του

Το κυριότερο πρόβλημα που καλείται να λυθεί από την πλευρά του Διαχειριστή του Συστήματος είναι αυτό της μοντελοποίησης του μηχανισμού κατανομής των παραγωγών με βάση την προσφορά που αυτοί κάνουν με στόχο πάντα την ικανοποίηση της ζήτησης. Θεωρώντας μια αγορά ηλεκτρικής ενέργειας όπου n παραγωγοί, για την κάλυψη ζήτησης D , υποβάλλουν ως προσφορές γραμμικές συναρτήσεις τιμής-ισχύος της μορφής :

$$F_i(x) = A_i + B_i x$$

τότε θα πρέπει να υπάρχει μια τιμή λ για την οποία θα ισχύουν οι σχέσεις

$$\lambda = A_i + B_i x_i \quad \forall \text{ παίκτη } i$$

$$\sum_{i=1}^n x_i = D$$

Θεωρώντας ότι η ζήτηση είναι ανελαστική, η μαθηματική έκφραση του προβλήματος είναι η επίλυση ενός γραμμικού συστήματος $(n+1) \times (n+1)$ το οποίο έχει ως ακολούθως:

$$\left| \begin{array}{cccccc|c|c|c} B_1 & 0 & 0 & \dots & 0 & 1 & x_1 & -A_1 \\ 0 & B_2 & 0 & \dots & 0 & 1 & x_2 & -A_2 \\ 0 & 0 & B_3 & \dots & 0 & 1 & x_3 & -A_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & B_n & 1 & x_n & -A_n \\ 1 & 1 & 1 & \dots & 1 & 0 & -\lambda & D \end{array} \right| * =$$

Οι περιορισμοί που τίθενται αφορούν το λ ως προς το Ανώτατο Όριο Προσφορών :

$$0 < \lambda < Price\ Cap$$

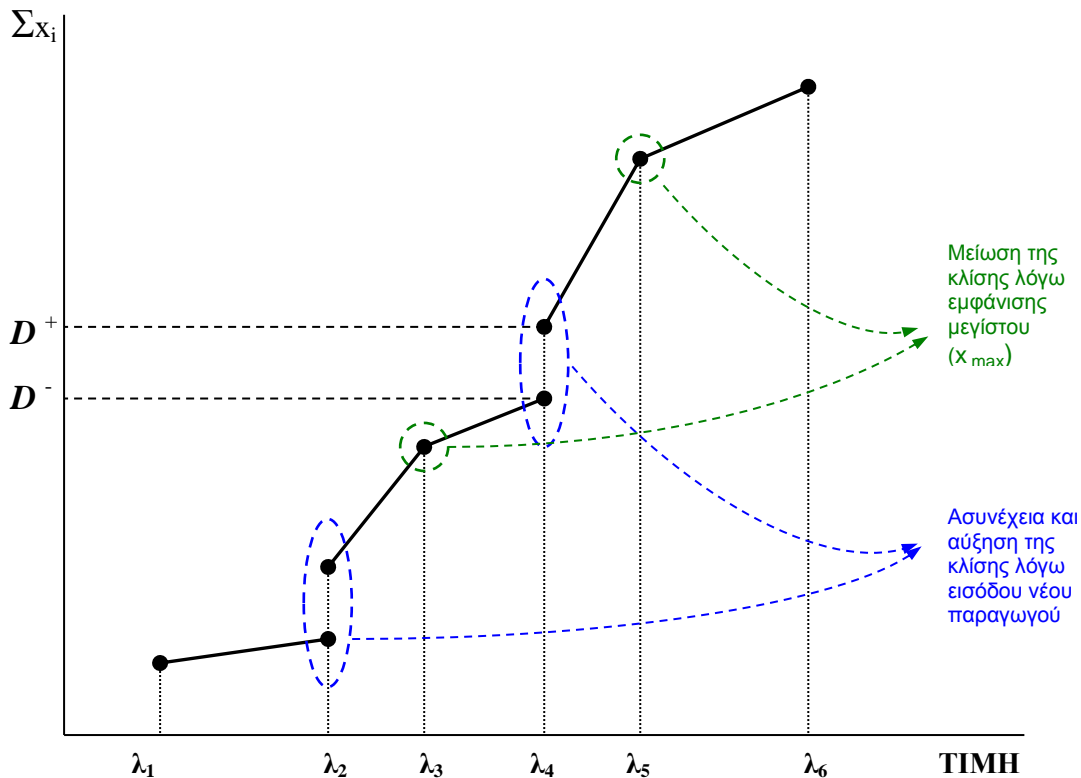
το πρόσημο των συντελεστών των συναρτήσεων προσφοράς ($A_i, B_i > 0$) και το εύρος τιμών που δύναται να κυμανθούν οι πωλούμενες ποσότητες ισχύος

$$x_{i\ min} < x_i < x_{i\ max} \quad \forall \text{ παίκτη } i.$$

Το πρόβλημα επιλύεται εύκολα όταν όλοι οι παραγωγοί δεν έχουν τεχνικό ελάχιστο, ισχύει δηλαδή $x_{i\ min} = 0$ και η αναλυτική λύση του συστήματος δίδεται από τον τύπο η ακόλουθη:

$$\lambda = \frac{D + \sum_{i=1}^n \frac{a_i}{b_i}}{\sum_{i=1}^n \frac{1}{b_i}}$$

Όταν όμως υπάρχουν τεχνικά ελάχιστα τότε παρουσιάζονται ασυνέχειες στη συνάρτηση συνολικής προσφοράς και το σύστημα δεν επιλύεται με ικανοποίηση όλων των περιορισμών. Η συνάρτηση της συνολική προσφοράς προκύπτει από αναστροφή της συνάρτησης τιμής – ισχύος που δίνουν οι παραγωγοί και στη συνέχεια άθροιση των ποσοτήτων.



Σχήμα 9. Το αντεστραμμένο διάγραμμα Προσφερόμενης Ισχύος-Τιμής όπου διακρίνονται οι περιοχές ασυνέχειας.

Καταρχήν θεωρούμε ότι η συνολική κατανομημένη ισχύς δεν πρέπει να ισούται της ζήτησης αλλά να είναι μεγαλύτερη ή ίση της ζήτησης. Υπολογίζουμε για κάθε x_{min} και x_{max} το αντίστοιχο λ_i και το αντίστοιχο D_i (δηλ. το σύνολο της διαθέσιμης παραγωγής στο επίπεδο τιμής λ_i). Βρίσκουμε τα D_i και D_{i+1} που ορίζουν το διάστημα μέσα στο οποίο βρίσκεται η ζήτηση D και εξετάζεται η κατάσταση των παικτών στα αντίστοιχα επίπεδα τιμών (λ_i και λ_{i+1}).

α) αν υπάρχει παίκτης για τον οποίο το $x(\lambda_i) = x(\lambda_{i+1})=0$ σημαίνει ότι στο διάστημα $[\lambda_i, \lambda_{i+1}]$ ο παίκτης δεν έχει φτάσει ακόμη το τεχνικό του ελάχιστο και ως εκ τούτου η προσφερόμενη ισχύς σε αυτό το διάστημα τιμών είναι ίση με το μηδέν. Έτσι ο τύπος της αναλυτικής λύσης του προηγούμενου εδαφίου εφαρμόζεται μόνο για τους υπόλοιπους παίκτες.

β) αν υπάρχει παίκτης για τον οποίο στο επίπεδο D_i έχει φτάσει στο τεχνικό του μέγιστο τότε του ανατίθεται

$x_i = x_{i \max}$ και στον αναλυτικό υπολογισμό του λ αφαιρούνται οι αντίστοιχοι συντελεστές του από τα αθροίσματα

$$\sum (a_i / b_i) \text{ και } \sum (1 / b_i).$$

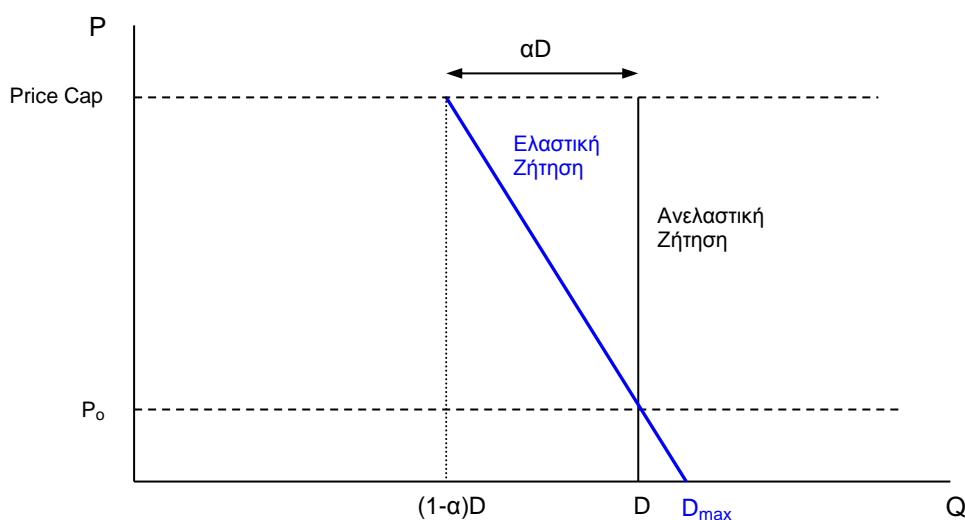
Επιπλέον η ζήτηση μειώνεται κατά $x_{i \max}$.

$$\lambda = \frac{(D - x_{\max}) + \sum_{i=1}^{n-1} \frac{a_i}{b_i}}{\sum_{i=1}^{n-1} \frac{1}{b_i}}$$

γ) αν υπάρχει παίκτης για τον οποίο στο επίπεδο D_i έχει $x(\lambda_i) = 0$ αλλά $x(\lambda_{i+1}) \neq 0$, γεγονός που σημαίνει ότι στο επίπεδο αυτό φθάνει στο τεχνικό του ελάχιστο και αρχίζει να προσφέρει την παραγωγή του, εξετάζεται αν η είσοδος στο σύστημα της ποσότητας $x_{i_{max}}$ κάνει το σύνολο της προσφερόμενης ποσότητας να ξεπερνά τη ζήτηση. Σε περίπτωση που η απάντηση είναι θετική τότε ορίζεται ως λ το λ_i και η κατανεμόμενη ισχύς ξεπερνά τη ζήτηση. Σε αντίθετη περίπτωση το σύστημα επιλύεται κανονικά.

4.1.4. Η Συνάρτηση της Ζήτησης

Στην προηγούμενη παράγραφο παρουσιάσθηκε η επίλυση του προβλήματος κατανομής θεωρώντας τη ζήτηση του Συστήματος ανελαστική (σταθερή). Κάτι τέτοιο ωστόσο δε συμβαίνει στην πραγματικότητα καθώς η ζήτηση του αγαθού της ηλεκτρικής ενέργειας παρουσιάζει μικρή μεν ελαστικότητα αλλά σε καμία περίπτωση δεν είναι ανελαστική ως προς την τιμή. Για το λόγο αυτό εισήχθη στο μοντέλο προσομοίωσης γραμμική συνάρτηση ζήτησης με στόχο να αποτυπώσει τις διακυμάνσεις της ζητούμενης ποσότητας ισχύος που οφείλονται σε αντίστοιχες διακυμάνσεις της τιμής.



Σχήμα 10. Η συνάρτηση Ζήτησης σε σχέση με την ανελαστική ζήτηση και το Price Cap.

Η συνάρτηση Τιμής-Ζήτησης δίδεται από την ακόλουθη σχέση:

$$P = A_D - B_D D$$

όπου A_D και B_D οι συντελεστές της συνάρτησης ζήτησης, P_o και D η τιμή που κλείνει το Σύστημα και η τιμή της ζήτησης όταν οι παραγωγοί υποβάλουν ως προσφορά το οριακό τους κόστος αντίστοιχα και P_{CAP} το Price Cap.

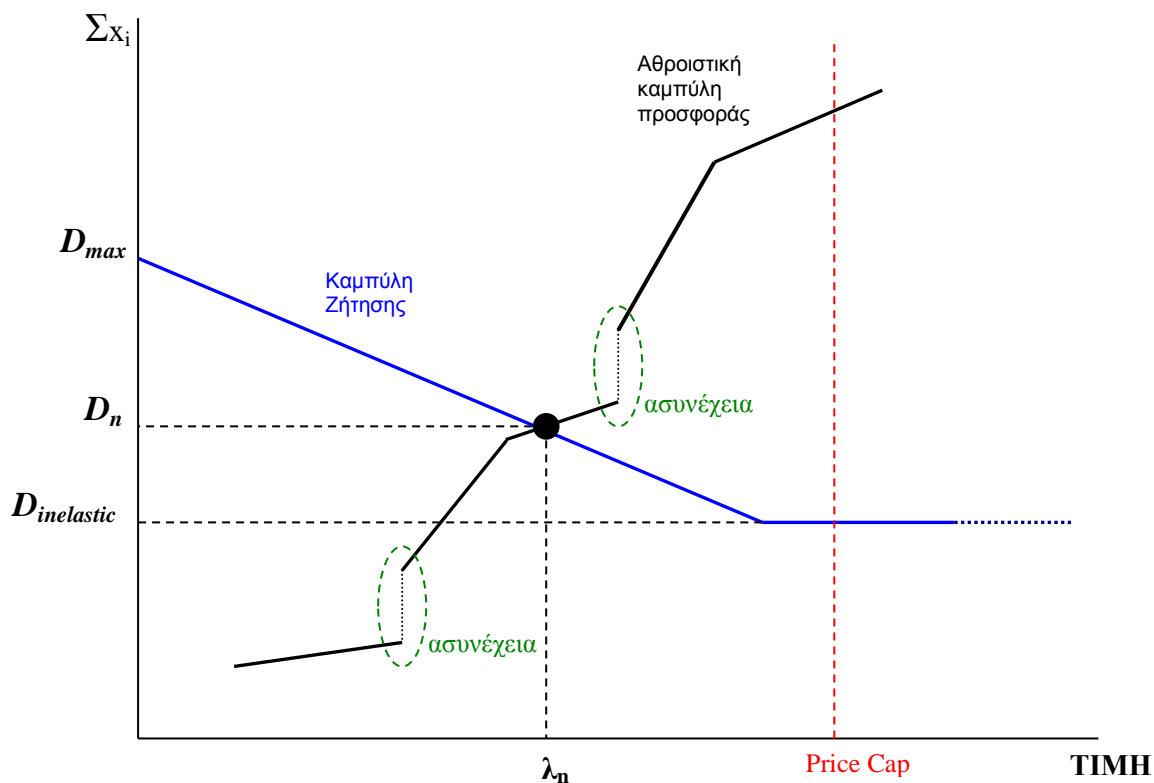
Οι συντελεστές της συνάρτησης ζήτησης ισούνται:

$$A_D = P_o + \frac{P_{CAP} - P_o}{\alpha} \quad \text{και} \quad B_D = \frac{P_{CAP} - P_o}{\alpha D}$$

Σε αυτή την περίπτωση, όπου η ζήτηση δεν είναι σταθερή, η τιμή του Συστήματος υπολογίζεται τελικά από το τύπο:

$$\lambda = \frac{D + \frac{P_o}{B_D} + \sum_{i=1}^n \frac{A_i}{B_i}}{\frac{1}{B_D} + \sum_{i=1}^n \frac{1}{B_i}}$$

Στην περίπτωση εισαγωγής γραμμικής συνάρτησης ζήτησης η επίλυση του προβλήματος της κατανομής των παραγωγών αναπαρίσταται γραφικά στο ακόλουθο σχήμα.



Σχήμα 11. Η επίλυση του προβλήματος της κατανομής των παραγωγών με γραμμική συνάρτηση ζήτησης.

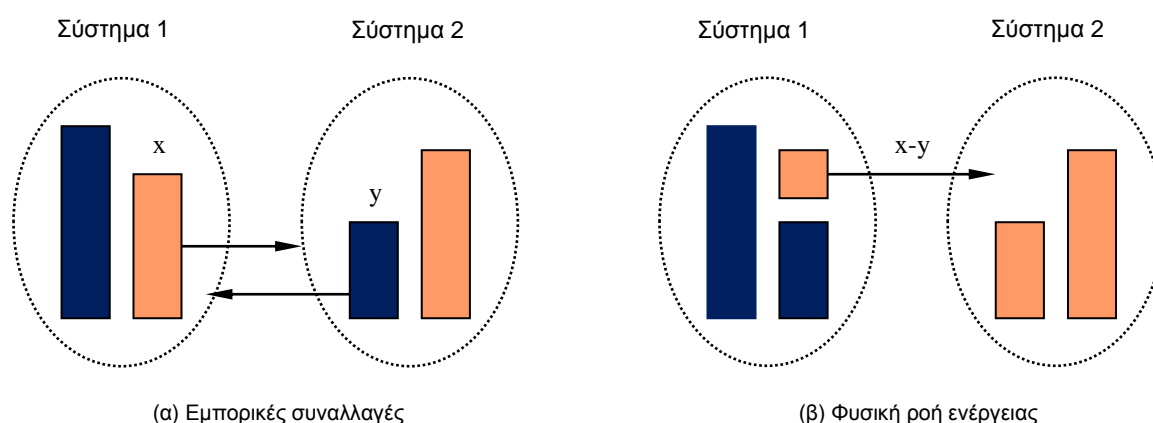
Στο Σχήμα 11, το σημείο (λ_n, D_n) απεικονίζει το σημείο τομής των καμπύλων ζήτησης-προσφοράς που καθορίζει το ύψος της τιμής κλεισίματος του Συστήματος και την τελικώς ζητούμενη ποσότητα. Παράλληλα στο σχήμα η τιμή D_{max} αντιστοιχεί στη μέγιστη τιμή ζήτησης που μπορεί να παρουσιασθεί λόγω υπερβολικής μείωσης της τιμής. Αντίστοιχα, η τιμή $D_{inelastic}$ είναι η ελάχιστη ποσότητα που μπορεί να ζητηθεί ή με άλλα λόγια το σημείο όπου η ζήτηση γίνεται ανελαστική [38-48].

4.2. Το Μοντέλο Δύο Διασυνδεδεμένων Αγορών Ηλεκτρικής Ενέργειας.

4.2.1. Γενική Περιγραφή του Μοντέλου

Το μοντέλο που αναπτύχθηκε έχει τη δυνατότητα προσομοίωσης μιας απελευθερωμένης αγοράς ηλεκτρικής ενέργειας, αποτελούμενη από δύο ανεξάρτητα ηλεκτρικά συστήματα διασυνδεδεμένα μεταξύ τους όπου στο καθένα από αυτά η τιμή της ηλεκτρικής ενέργειας διαμορφώνεται ξεχωριστά μέσω μιας διαδικασίας υποχρεωτικής υποβολής προσφορών (offers) από τους παραγωγούς ηλεκτρικής ενέργειας (γραμμική μορφής) στον αντίστοιχο Διαχειριστή του κάθε συστήματος. Καθώς όμως τα συστήματα είναι διασυνδεδεμένα μεταξύ τους οι παραγωγοί του ενός έχουν τη δυνατότητα να υποβάλλουν προσφορές και στο γειτονικό λειτουργώντας ως εισαγωγείς, επιμερίζοντας με όποιο τρόπο νομίζουν την παραγωγική τους δυναμικότητα στα δύο συστήματα μέσω των αντιστοίχων προσφορών. Επιπλέον, ως προς τη δυναμικότητα της διασύνδεσης μεταξύ των δύο συστημάτων, εξετάζονται συγκριτικά η περίπτωση της θεωρητικά απεριόριστης δυναμικότητας και η περίπτωση της διασύνδεσης που υπόκειται σε περιορισμό.

Στην περίπτωση των δυο ηλεκτρικών Συστημάτων το μοντέλο παραμένει το ίδιο στη βάση του με τη διαφορά ότι έχουμε δύο συστήματα που το καθένα από αυτά έχει τον δικό του Ανεξάρτητο Διαχειριστή και τους δικούς του εγκατεστημένους παραγωγούς ηλεκτρικής ενέργειας [58]. Τα δύο Συστήματα όμως είναι διασυνδεδεμένα μεταξύ τους και έτσι οι παραγωγοί μπορούν να υποβάλλουν προσφορές και στα δύο Συστήματα ανάλογα με το τι τους συμφέρει κάθε φορά. Υπάρχουν ωστόσο δύο διαφορετικές οπτικές πάνω στις οποίες μοντελοποιήθηκε και μελετήθηκε η περίπτωση των δύο Συστημάτων : α) η περίπτωση όπου οι παραγωγοί μπορούν να διαθέσουν ισχύ τους και στα δύο Συστήματα σε αναλογία που αυτοί κρίνουν συμφέρουσα και β) η περίπτωση όπου οι παραγωγοί είναι υποχρεωμένοι να διαθέσουν το σύνολο της παραγωγής τους αποκλειστικά είτε στο ένα είτε στο άλλο Σύστημα ανάλογα πάλι με το δικό τους προσδοκώμενο όφελος. Ανεξάρτητα ωστόσο από την πληθώρα και τη φορά των εμπορικών συναλλαγών μεταξύ των δύο συστημάτων μέσα στο χρονικό διάστημα που αυτές αφορούν, η φυσική ροή της ηλεκτρικής ενέργειας όπως προκύπτει μετά την οριστικοποίηση των συναλλαγών αυτών είναι πάντοτε μόνο προς το ένα σύστημα (Σχ.12).



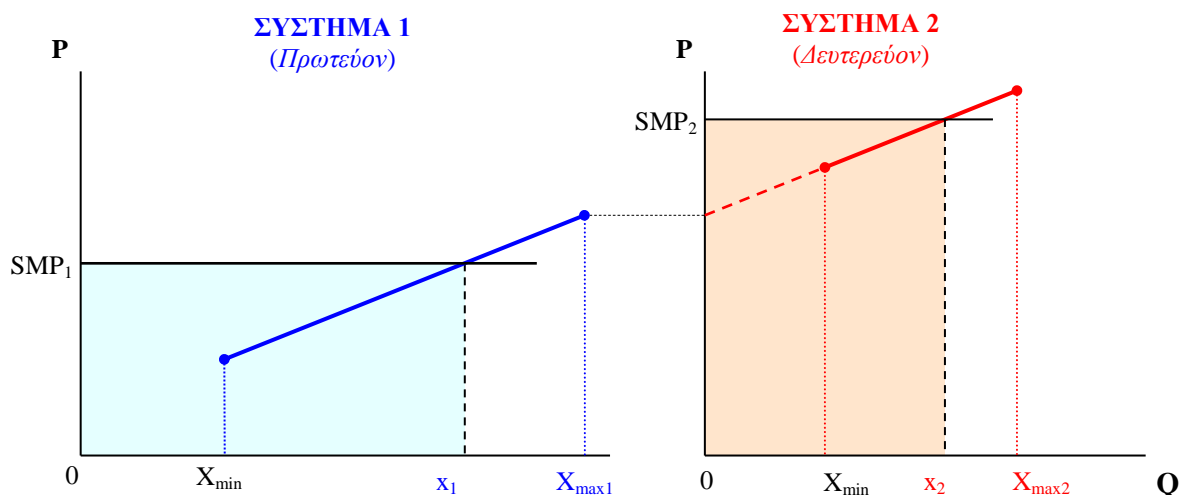
Σχήμα 12. Σχηματική απεικόνιση των εμπορικών συναλλαγών μεταξύ δύο διασυνδεδεμένων αγορών ηλεκτρικής ενέργειας σε σχέση με τις αντίστοιχες φυσικές ροές ηλεκτρικής ενέργειας.

Μία ακόμη παράμετρος που υπεισέρχεται στην περίπτωση των δύο Συστημάτων είναι η δυναμικότητα διασύνδεσης και η αντιμετώπιση των συμφορήσεων. Για λόγους συγκριτικής μελέτης μοντελοποιήθηκε και η θεωρητική περίπτωση της απεριόριστης δυναμικότητας διασύνδεσης (unconstrained problem) εκτός από την πραγματική περίπτωση της ύπαρξης περιορισμών στη μεταφορά ποσοτήτων ηλεκτρικής ενέργειας από το ένα Σύστημα στο άλλο (constrained problem). Για λόγους απλότητας στη συνέχεια θα περιγράψουμε τις περιπτώσεις συμμετοχής

στο ένα ή και στα δύο Συστήματα θεωρώντας απεριόριστη δυναμικότητα διασύνδεσης και στη συνέχεια θα δούμε πως τροποποιούνται οι περιπτώσεις αυτές όταν έχουμε περιορισμούς στη διασύνδεση.

4.2.2. Περίπτωση υποβολής προσφοράς και στα δύο Συστήματα

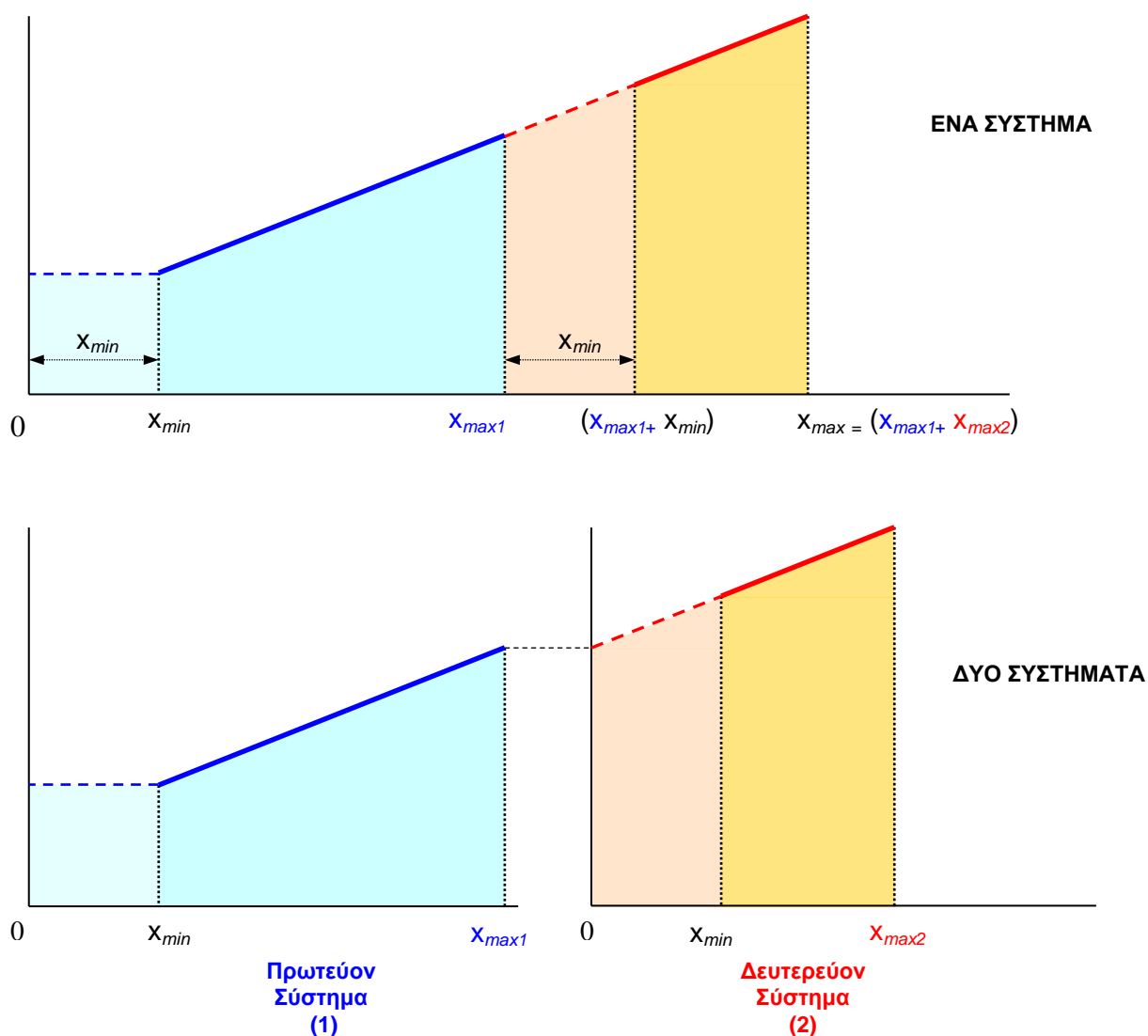
Στην περίπτωση αυτή ο κάθε παραγωγός έχει δικαίωμα να υποβάλλει προσφορά και στο γειτονικό Σύστημα για ένα τμήμα της παραγωγής του ή και για το σύνολο αυτής αν το κρίνει σκόπιμο (επιλογή “Split Offers”). Θεωρούμε ότι η καμπύλη προσφοράς του είναι ενιαία και απλά το ένα τμήμα της προσφοράς του υποβάλλεται στο γειτονικό για εκείνον Σύστημα (Δευτερεύον Σύστημα) ανάλογα με το ποιο Σύστημα θεωρείται για τον παραγωγό Πρωτεύον τη συγκεκριμένη χρονική στιγμή και ανεξάρτητα από το που βρίσκεται εγκατεστημένος. Ο τόπος εγκατάστασης του παραγωγού ορίζει απλά το ότι ο συγκεκριμένος παραγωγός θα ξεκινήσει το παίγνιο θεωρώντας ως Πρωτεύον Σύστημα είναι αυτό και στο οποίο είναι εγκατεστημένος. Ο καθορισμός του Πρωτεύοντος Συστήματος έχει σημασία για το παραγωγό και το παίγνιο με την έννοια ότι στο Πρωτεύον Σύστημα υποβάλλει το χαμηλό από πλευράς κόστους τμήμα της προσφοράς του. Ένας παραγωγός κατά τη διάρκεια του παιγνίου μπορεί να αλλάξει πολλές φορές Πρωτεύον Σύστημα ανάλογα με την εξέλιξη του παιγνίου.



Σχήμα 13. Απεικόνιση προσφοράς παραγωγού σε δύο συστήματα και αντιστοίχων πωλούμενων ποσοτήτων και εσόδων με Uniform Pricing τιμολόγηση.

Υπάρχει ωστόσο ένας επιπλέον περιορισμός που υπεισέρχεται στην περίπτωση υποβολής προσφοράς και στα δύο Συστήματα: ο παραγωγός πρέπει να προσφέρει τουλάχιστον το τεχνικό του ελάχιστο και στα δύο Συστήματα γιατί απλά υπάρχει περίπτωση να μην μπορέσει πωλήσει στο ένα από τα δύο οπότε θα πρέπει να παράξει τουλάχιστον το τεχνικό του ελάχιστο. Αυτό σημαίνει ότι ο παραγωγός τροποποιεί κάθε φορά την προσφορά του ως προς την αναλογία της προσφερόμενης ισχύος στα δύο Συστήματα έχοντας ένα εύρος μεταβολής που ορίζεται από τη διαφορά της δυναμικότητάς μείον το διπλάσιο του τεχνικού του ελαχίστου. Βέβαια, μπορεί να μην προσφέρει καθόλου στο ένα Σύστημα, οπότε δεν έχουν νόημα τέτοιου είδους περιορισμοί αφού διαθέτει το σύνολο της δυναμικότητάς του σε ένα Σύστημα.

Οι παραγωγοί χρησιμοποιούν τον ίδιο μαθησιακό αλγόριθμο θεωρώντας ως ενιαία την προσφορά τους αλλά και τα αποτελέσματα που προκύπτουν από αυτή. Προστίθεται ωστόσο μια ακόμη μεταβλητή απόφασης που δεν επηρεάζει τη διαμόρφωση της προσφοράς ως προς το ύψος της τιμής αλλά ως προς την κατανομή της ποσότητας στα δύο Συστήματα. Η μεταβλητή αυτή δεν χρησιμοποιεί προσαρμοστική μαθησιακή διαδικασία προκειμένου να συγκλίνει αλλά είναι τοπική μεταβλητή απόφασης. Συγκεκριμένα, ο παραγωγός εξετάζει την αποδοτικότητα των πωλήσεών του συγκρίνοντας το κέρδος ανά μονάδα ισχύος που πούλησε στο κάθε Σύστημα. Όταν λοιπόν η αποδοτικότητα των πωλήσεων στο ένα Σύστημα γίνει ελκυστικότερη από την αντίστοιχη του άλλου κατά ένα συγκεκριμένο μέγεθος (*Tolerance*) τότε ο παραγωγός αυξάνει την προσφερόμενη ποσότητα στο ελκυστικότερο Σύστημα κατά ένα μικρό ποσοστό (*Adaptivity*).



Σχήμα 14. Σύγκριση παραγωγού που πωλεί το σύνολο της δυναμικότητάς του σε ένα και σε δύο Συστήματα.

Στην περίπτωση που ο παραγωγός βρεθεί στην κατάσταση να πουλά στο λιγότερο ελκυστικό Σύστημα μόνο το τεχνικό του ελάχιστο, δεν μεταφέρει την ποσότητα αυτή στο άλλο Σύστημα, ασχέτως πόσο πιο ελκυστικό αυτό έχει γίνει, παρά μόνο στην περίπτωση που δεν καταφέρει να την πουλήσει. Τότε μόνο μπορεί να βρεθεί να πουλά μόνο στο ένα Σύστημα. Για να ξαναπροσφέρει στο Δευτερεύον πλέον γι' αυτόν Σύστημα, πρέπει να μην μπορέσει να πουλήσει στο Πρωτεύον Σύστημα τόση ποσότητα όσο είναι το τεχνικό του ελάχιστο και επιπλέον το Δευτερεύον Σύστημα να είναι τη συγκεκριμένη χρονική στιγμή ελκυστικότερο.

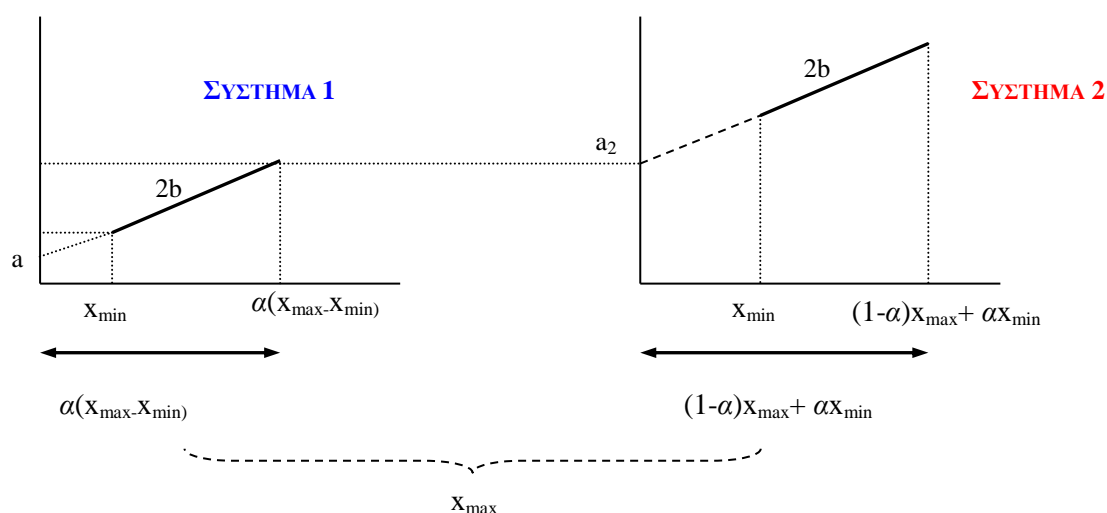
Ο τρόπος με τον οποίο μεταφέρει προσφερόμενες ποσότητες από το ένα Σύστημα στο άλλο κατά τη διάρκεια του παιγνίου περιγράφηκε ανωτέρω και η μόνη δυνατότητα παρέμβασης από πλευράς του μοντέλου μπορεί να γίνει με την τροποποίηση του δείκτη προσαρμοστικότητας (*Adaptivity*). Εκτός αυτού όμως μπορεί να καθορισθεί και η αρχική κατανομή της προσφερόμενης ποσότητας στα δύο Συστήματα κατά την έναρξη του παιγνίου. Αυτό γίνεται μέσω του δείκτη αρχικής κατανομής α (*Initial Index*) ο οποίος ορίζεται ως εξής:

$$a = \frac{x}{x_{\max} - x_{\min}}$$

Θέτοντας την κατάλληλη τιμή στο μοντέλο ορίζεται η αρχική κατανομή της προσφοράς του κάθε παίκτη. Συγκεκριμένα, η αντιστοίχιση των τιμών του δείκτη αρχικής κατανομής α και διαφόρων κατανομών είναι η ακόλουθη:

- $\alpha = 0$, ο παραγωγός προσφέρει το σύνολο της ποσότητάς του στο δευτερεύον Σύστημα.
- $\alpha = a_{\min} = \frac{x_{\min}}{x_{\max} - x_{\min}}$, ο παραγωγός προσφέρει μόνο το τεχνικό του ελάχιστο στο πρωτεύον Σύστημα.
- $a_{\min} < \alpha < 1$, ο παραγωγός προσφέρει και στα δύο Συστήματα ⁵.
- $\alpha = 1$, ο παραγωγός προσφέρει μόνο το τεχνικό του ελάχιστο στο Δευτερεύον Σύστημα.
- $\alpha > 1$, ο παραγωγός προσφέρει το σύνολο της ποσότητάς του στο Πρωτεύον Σύστημα

Στο επόμενο σχήμα απεικονίζεται η φυσική έννοια του δείκτη αρχικής κατανομής α και ο συσχετισμός των μεγεθών στην περίπτωση που ο παραγωγός προσφέρει και στα δύο Συστήματα.



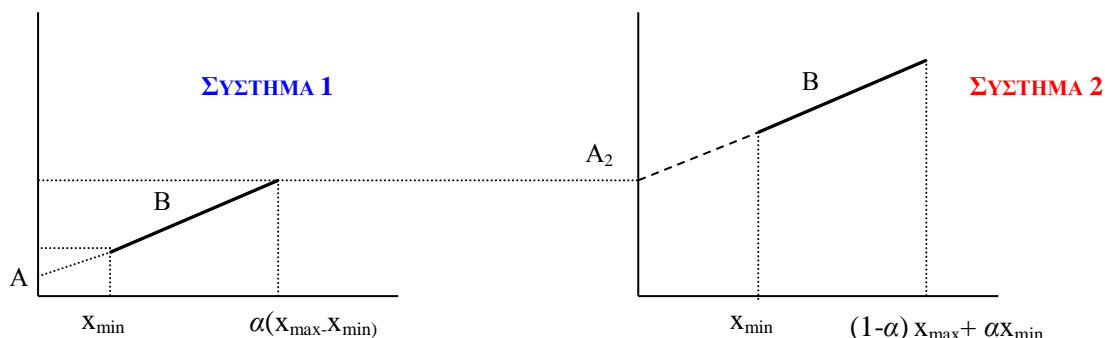
Σχήμα 15. Ο δείκτης αρχικής κατανομής α και ο συσχετισμός των μεγεθών σε παραγωγό που υποβάλλει προσφορά και στα δύο Συστήματα.

⁵ Ισοκατανομή της δυναμικότητας στα δύο Συστήματα έχουμε όταν $\alpha = 0.5 + a_{\min}$

Η διαίρεση της ενιαίας προσφοράς σε δύο Συστήματα απαιτεί τον κατάλληλο μαθηματικό μετασχηματισμό τόσο για τον υπολογισμό των συντελεστών της συνάρτησης οριακού κόστους όσο και για τον υπολογισμό των συντελεστών της συνάρτησης προσφοράς. Ο γραμμικός συντελεστής και των δύο συναρτήσεων παραμένει σταθερός και στα δύο Συστήματα, ενώ για τον σταθερό συντελεστή απαιτείται μετασχηματισμός.

Στην περίπτωση της συνάρτησης προσφοράς του παραγωγού ο μετασχηματισμός για το σταθερό συντελεστή είναι ο ακόλουθος :

$$A_2 = A + B \cdot \alpha (x_{\max} - x_{\min}) = A + Bx_1$$

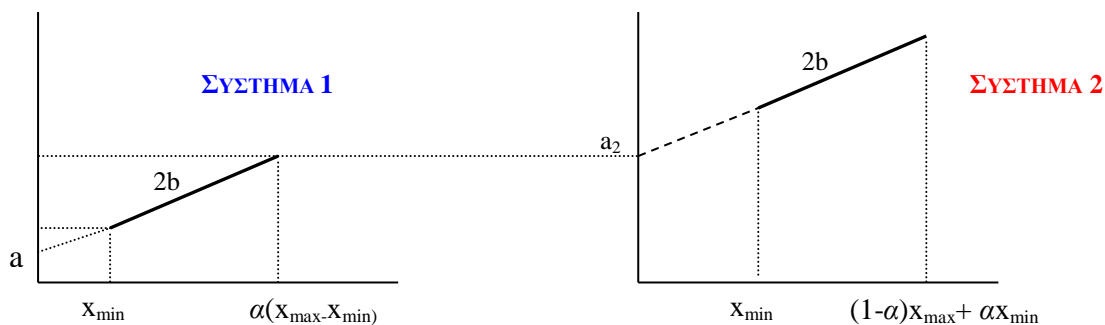


Σχήμα 16.α Ο μετασχηματισμός της συνάρτησης προσφοράς στο Δευτερεύον Σύστημα.

Στην περίπτωση της συνάρτησης του οριακού κόστους ο μετασχηματισμός για το σταθερό συντελεστή εξαρτάται από την πωλούμενη στο Πρωτεύον Σύστημα ποσότητα και ορίζεται ως ακολούθως :

- το κόστος στο Πρωτεύον Σύστημα είναι: $C_1 = a + 2bx_1$
- το κόστος στο Δευτερεύον Σύστημα είναι: $C_2 = a_2 + 2bx_2$, όπου $a_2 = a + 2bx_1$

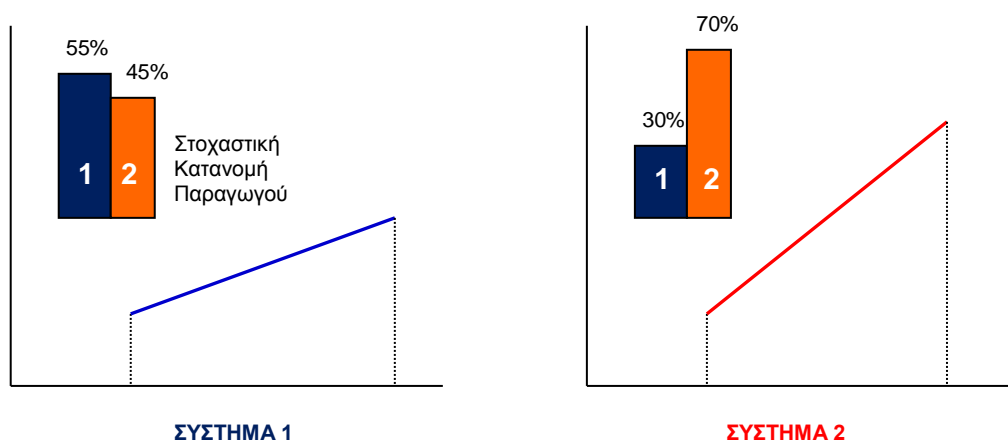
Αυτό σημαίνει ότι το κόστος του παραγωγού στο Δευτερεύον Σύστημα εξαρτάται από την ποσότητα που πωλείται στο πρωτεύον, π.χ. στην περίπτωση που $x_1=0 \Rightarrow a_2 = a$.



Σχήμα 16.β Ο μετασχηματισμός της συνάρτησης οριακού κόστους στο Δευτερεύον Σύστημα.

4.2.3. Περίπτωση υποβολής προσφοράς μόνο στο ένα Σύστημα

Στην περίπτωση αυτή ο παραγωγός έχει την ευχέρεια να προσφέρει σε όποιο από τα δύο Συστήματα επιθυμεί με μόνο περιορισμό ότι θα προσφέρει το σύνολο της δυναμικότητάς του στο Σύστημα που θα επιλέξει (επιλογή “Offer Only to One System”). Εδώ ωστόσο η μοντελοποίηση προσφέρει τη δυνατότητα επιλογής ως προς το κριτήριο απόφασης του παραγωγού. Μπορεί δηλαδή να επιλεγθεί οι παραγωγοί να χρησιμοποιούν τον αλγόριθμο που περιγράφεται στην περίπτωση *Split Offers* και ο οποίος λαμβάνει υπόψη την αποδοτικότητα των πωλήσεων στα δύο Συστήματα μόνο κατά την τελευταία προσφορά ή εναλλακτικά να γίνει χρήση ενός προσαρμοστικού μαθησιακού αλγορίθμου ανάλογου με εκείνου που ήδη χρησιμοποιεί για τον καθορισμό της μορφής της ενιαίας προσφοράς.



Σχήμα 17. Ο Προσαρμοστικός αλγόριθμος μάθησης για την επιλογή του Συστήματος που θα υποβληθεί η προσφορά.

Συγκεκριμένα, ο μαθησιακός αλγόριθμος που χρησιμοποιεί ο παραγωγός για να επιλέξει σε ποιο Σύστημα θα υποβάλλει την προσφορά του λαμβάνει υπόψη του μια κατανομή πιθανότητας η οποία αναφέρεται στις δύο πιθανές επιλογές του παραγωγού, δηλαδή υποβολή προσφοράς στο Σύστημα 1 ή στο Σύστημα 2. Η κατανομή αυτή προσαρμόζεται σταδιακά κατά ένα μικρό βήμα σύμφωνα με το κριτήριο που είναι η απόδοση της προσφοράς του παραγωγού στο ένα και στο άλλο Σύστημα κατά την τελευταία επανάληψη. Η απόδοση για το ένα Σύστημα, αυτό στο οποίο υπέβαλε προσφορά ο παραγωγός βασίζεται σε πραγματικά στοιχεία ενώ η απόδοση στο άλλο Σύστημα στην υποθετική απόδοση που θα είχε η προσφορά του αν την είχε υποβάλει στο άλλο Σύστημα βάσει της ποσότητας που θα πωλούσε και τιμής που έκλεισε το Σύστημα. Η επιλογή της δράσης του παραγωγού γίνεται τυχαία βάσει της στοχαστικής κατανομής όπως αυτή έχει διαμορφωθεί μέχρι εκείνη τη στιγμή.

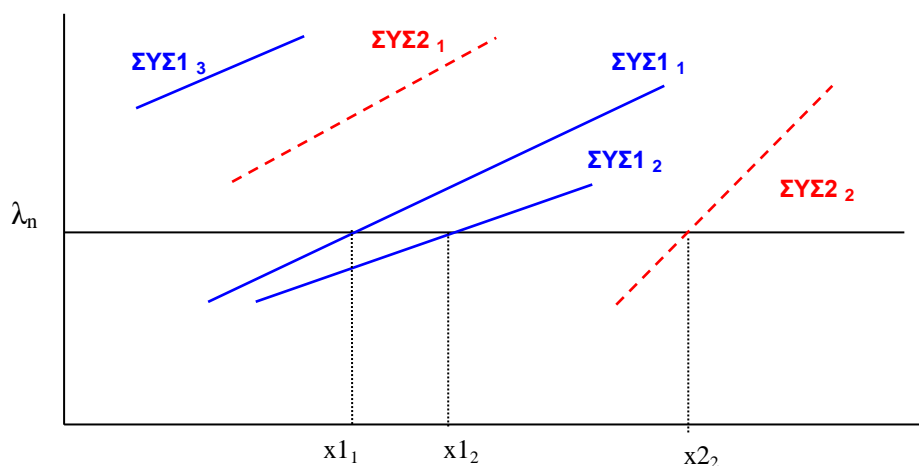
4.2.4. Περιορισμός στη δυναμικότητας διασύνδεσης

Στην περίπτωση ύπαρξης περιορισμού στη διασύνδεση των δύο ηλεκτρικών Συστημάτων ισχύουν ακριβώς τα ίδια που περιγράψαμε ανωτέρω με τη μόνη διαφορά ότι η μεταφορά ηλεκτρικής ενέργειας από το ένα στο άλλο Σύστημα υπόκειται σε φυσικούς περιορισμούς δυναμικότητας μεταφοράς. Σε τέτοια περίπτωση παρουσιάζεται το φαινόμενο της συμφόρησης (congestion) το οποίο αντιμετωπίζεται με διάφορους τρόπους τιμολόγησης, όπως αναπτύχθηκε σε προηγούμενη ενότητα.

Στο μοντέλο που αναπτύχθηκε για το ερευνητικό έργο, οι παραγωγοί του ενός Συστήματος κατά τη διάρκεια ενός συγκεκριμένου παιχνιδιού μπορούν να προσφέρουν ποσότητες στο γειτονικό Σύστημα και αντιστρόφως.

Ενδεχομένως κάποιες από αυτές τις προσφορές γίνονται δεκτές και η αντίστοιχη παραγωγή πρέπει να καταμεριστεί. Η συμφόρηση, λοιπόν, παρατηρείται όταν η απόλυτη τιμή της διαφοράς του συνόλου των πωλούμενων ποσοτήτων που πρέπει να μεταφερθεί προς τη μία κατεύθυνση και του συνόλου των πωλούμενων ποσοτήτων που πρέπει να μεταφερθεί προς την αντίθετη κατεύθυνση υπερβαίνει τη δυναμικότητα της διασύνδεσης. Κατ'αυτόν τον τρόπο η συμφόρηση νοείται μόνο κατά τη μία φορά, π.χ. από το Σύστημα 1 προς το Σύστημα 2, καθώς οι άλλες ποσότητες που θεωρητικά πωλούνται στο γειτονικό σύστημα δεν μεταφέρονται αλλά διευθετούνται λογιστικά κατά την εκκαθάριση των δύο Συστημάτων.

Η μεθοδολογία που μοντελοποιήθηκε η διευθέτηση των συμφορήσεων και ο αντίστοιχος αλγόριθμός παρουσιάζεται στη συνέχεια με τη χρήση ενός παραδείγματος. Ας υποθέσουμε ότι κατά την επανάληψη n του παιγνίου στο Σύστημα στο οποίο καταλήγει η φορά της συμφόρησης έχουν υποβάλλει προσφορές 3 αλλοδαποί (Sys1₁, Sys1₂ και Sys1₃) και δύο ημεδαποί παραγωγούς (Sys2₁ και Sys2₂). Το Σύστημα επιλύεται κανονικά σαν να μην υπήρχε περιορισμός δυναμικότητας στη διασύνδεση των δύο Συστημάτων και προκύπτει μία τιμή (λ_n) και ελέγχουμε αν το σύνολο των ξένων παικτών που πήραν μερίδιο αγοράς υπερέβησαν τον περιορισμό. Ας υποθέσουμε ότι στην επανάληψη n , δύο αλλοδαποί παίκτες (Sys1₁ και Sys1₂) παίρνουν μερίδια αγοράς x_{11} και x_{12} αντίστοιχα τα οποία στο σύνολό τους ξεπερνούν τον περιορισμό της διασύνδεσης IC , ενώ ο τρίτος αλλοδαπός παραγωγός δεν κατορθώνει να πουλήσει κάποια ποσότητα. Από τους εγχώριους παραγωγούς μόνο ο παραγωγός Sys2₂ πωλεί στο Σύστημα (x_{22}).



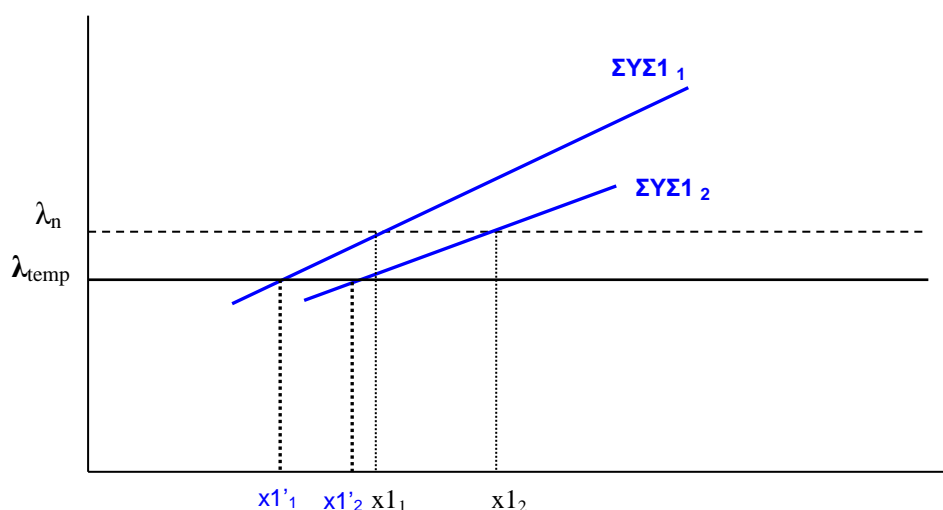
Σχήμα 18. Πρώτη επίλυση Συστήματος με συμμετοχή αλλοδαπών παραγωγών και υφιστάμενο περιορισμό δυναμικότητας διασύνδεσης.

Στη συνέχεια επιλύουμε ένα υποθετικό σύστημα στο οποίο συμμετέχουν μόνο οι δύο ξένοι παίκτες που πήραν μερίδιο αγοράς και βρίσκουμε το επίπεδο τιμής εκείνο (λ_{temp}) στο οποίο το άθροισμα των μεριδίων αγοράς x_{11} και x_{12} ισούται ή είναι ίσο ή μικρότερο από τον περιορισμό της διασύνδεσης, δηλαδή:

$$x_{11}' + x_{12}' \leq IC$$

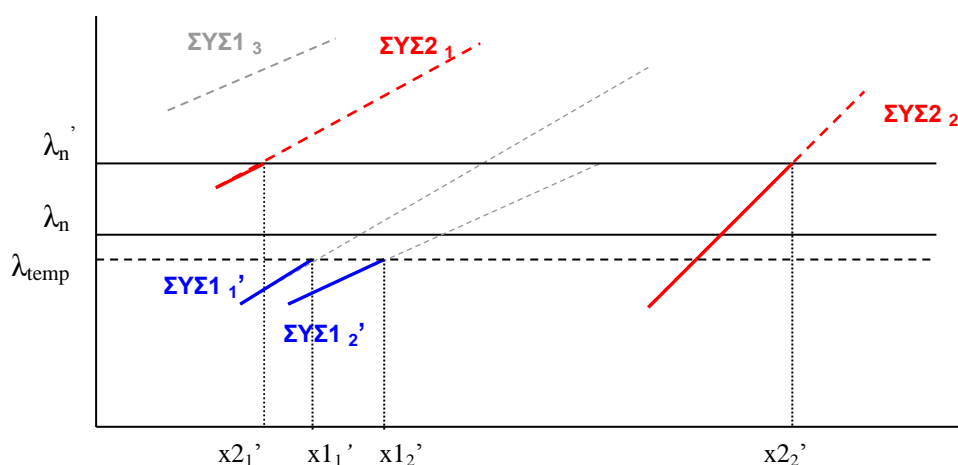
Τώρα το αρχικό Σύστημα επανεπιλύεται, μέσα στην ίδια επανάληψη n , με τις εξής τροποποιήσεις:

- α) όλοι οι αλλοδαποί παραγωγούς που συμμετείχαν πριν και δεν πήραν μερίδιο αγοράς τώρα δε συμμετέχουν.
- β) οι αλλοδαποί παίκτες που πήραν μερίδιο αγοράς προσφέρουν τροποποιημένες προσφορές όπου το μέγιστό τους ($x_{11 \max}$ και $x_{12 \max}$) είναι συμπίπτει με τα μερίδια αγοράς που προέκυψαν από την επίλυση του υποθετικού συστήματος (x_{11}' και x_{12}'),
- γ) οι εγχώριοι παραγωγούς προσφέρουν ότι είχαν προσφέρει και πριν.



Σχήμα 19. Καθορισμός των ποσοτήτων που πωλούν αλλοδαποί παραγωγοί λαμβάνοντας υπόψη τον περιορισμό της δυναμικότητας διασύνδεσης.

Η επίλυση του νέου αυτού Συστήματος είναι και η τελική από την οποία εξάγονται τα μεγέθη της τιμής του Συστήματος (λ_n) καθώς και των πωλήσεων. Η πρώτη εκκαθάριση θα είχε ως αποτέλεσμα ένα συγκεκριμένο κόστος για τον Διαχειριστή του Συστήματος, και κατ'επέκταση για τους καταναλωτές, το οποίο προέκυπτε από το άθροισμα των εσόδων όλων των παικτών. Η τελική εκκαθάριση έχει ως αντίστοιχο αποτέλεσμα ένα διαφορετικό μέγεθος υψηλότερο από το πρώτο. Η διαφορά αυτή ισούται στην ουσία με το λεγόμενο *Uplift Cost* που αντιστοιχεί στο «κόστος» του περιορισμού της διασύνδεσης. Το κόστος αυτό που αντιπροσωπεύει ένα επιπλέον κόστος για το Σύστημα λόγω της μη επαρκούς χωρητικότητας της διασύνδεσης επιμερίζεται αναλογικά στους εγχώριους παραγωγούς, οι οποίοι συμμετείχαν με προσφορά και κατανεμήθηκαν στο Σύστημα κατά την τρέχουσα επανάληψη.



Σχήμα 20. Τελική επίλυση Συστήματος με συμμετοχή αλλοδαπών παραγωγών και υφιστάμενο περιορισμό δυναμικότητας διασύνδεσης.

4.3. Το Μοντέλο Αγοράς με Προσφορές Ζήτησης και Κατανεμημένη Παραγωγή Ενέργειας

4.3.1. Γενική Περιγραφή του Μοντέλου

Στόχος της επέκτασης αυτής του μοντέλου είναι να αξιολογήσει την επίδραση από την εισαγωγή της **Κατανεμημένης Παραγωγής Ενέργειας** (*DER: Distributed Energy Resources Generation*) σε μια αγορά ηλεκτρικής ενέργειας, είτε από την πλευρά της προσφοράς ή της ζήτησης. Η Κατανεμημένη Παραγωγή Ενέργειας (*DER*) είναι η παραγωγή ενέργειας από μικρούς διάσπαρτους παραγωγούς ενέργειας οι οποίοι είναι εγκατεστημένοι κοντά στα κέντρα της ζήτησης και ως επί το πλείστον δεν χρησιμοποιούν το δίκτυο μεταφοράς αλλά εγχύουν την ενέργεια που παράγουν απευθείας στο δίκτυο διανομής ή ακόμη και στο ίδιο σημείο όπου βρίσκεται η κατανάλωση (π.χ. μεγάλος βιομηχανικός ή εμπορικός καταναλωτής, οικισμός κλπ.). Συνήθως πρόκειται για παραγωγή από Ανανεώσιμες Πηγές Ενέργειας (αιολικά ή φωτοβολταϊκά πάρκα, μικροί υδροηλεκτρικοί σταθμοί) ή από χρήση συμβατικών καυσίμων, κυρίως φυσικού αερίου, σε μονάδες Συμπαραγωγής ή Κυψέλες Καυσίμου (*Fuel Cells*). Η ένταξη στο μοντέλο αγοράς της Κατανεμημένης Παραγωγής δεν γίνεται για τον κάθε μικροπαραγωγό χωριστά αλλά μέσω των **Φορέων Σωρευτικής Εκπροσώπησης Παραγωγής** ή εν συντομία Εκπρόσωποι Παραγωγής (*Supply Aggregators* ή *DER Aggregators*), νομικά πρόσωπα δηλαδή που αντιπροσωπεύουν και διαχειρίζονται ένα «χαρτοφυλάκιο» παραγωγής από πολλούς μικρούς παραγωγούς με σκοπό την καλύτερη αξιοποίηση της. Με τον τρόπο αυτό ο Εκπρόσωπος Παραγωγής υποβάλλει μία μόνο σωρευτική προσφορά στην αγορά για το σύνολο των παραγωγών που εκπροσωπεί λαμβάνοντας υπόψη τη διαθεσιμότητα και τυχόν προβλέψεις σχετικά με την παραγωγική ισχύ που διαχειρίζεται [68-71].

Η ουσιαστική καινοτομία όμως που προστέθηκε στο μοντέλο της αγοράς είναι η δυνατότητα συμμετοχής της Κατανεμημένης Παραγωγής Ενέργειας και από την πλευρά της Ζήτησης. Αυτό γίνεται μέσω των Εκπροσώπων Παραγωγής οι οποίοι όμως δρουν από την άλλη πλευρά εκπροσωπώντας μικρούς παραγωγούς που είναι διατεθειμένοι να καλύψουν την ζήτηση προσφέροντας τοπικά την παραγωγή τους. Παράλληλα, εκπροσωπούν και καταναλωτές οι οποίοι είναι διατεθειμένοι να περικόψουν ένα μέρος του φορτίου τους από ένα ύψος τιμής και πάνω και να αποζημιωθούν γι' αυτό (Προσφορές Περικοπής Φορτίου - *Interruptible Load Bids*). Κατά τον τρόπο αυτό λειτουργούν στη χονδρεμπορική αγορά ως «μειωτές» της ζήτησης υποβάλλοντας και αυτοί σωρευτικές προσφορές μείωσης/εξυπηρέτησης της ζήτησης (*Demand Side Bidding*) μέσω των οποίων δηλώνουν κατά ποιά ποσότητα μπορούν να μειώσουν ή να αναλάβουν οι ίδιοι την εξυπηρέτηση της ζήτησης και σε ποιά τιμή. Οι φορείς αυτοί ονομάζονται **Φορείς Σωρευτικής Εκπροσώπησης Μείωσης/Κάλυψης της Ζήτησης** ή εν συντομία Εκπρόσωποι Ζήτησης (*Demand Side Aggregators*).

Ως βάση χρησιμοποιήθηκε το ίδιο μοντέλο της αγοράς ηλεκτρικής ενέργειας του ενός ανεξάρτητου διαχειριστή της αγοράς και πολλών παραγωγών ηλεκτρικής ενέργειας (παίκτες), οι οποίοι δρώντας σε ένα περιβάλλον ελλιπούς πληροφόρησης, χρησιμοποιούν έναν στοχαστικό προσαρμοστικό αλγόριθμο εκμάθησης, προκειμένου να μεγιστοποιήσουν τα κέρδη τους. Συγκεκριμένα το μοντέλο της αγοράς αποτελείται από:

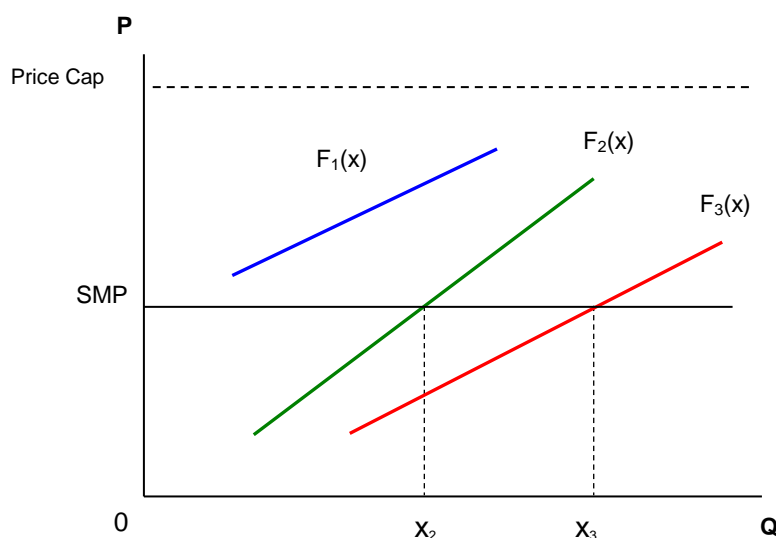
- 1) Έναν **Ανεξάρτητο Διαχειριστή του Συστήματος** στον οποίο υποβάλλονται οι προσφορές και ο οποίος αγοράζει ποσότητες από τους παραγωγούς με στόχο την κάλυψη της ζήτησης D .
- 2) **i Παραγωγούς Ηλεκτρικής Ενέργειας** (παίκτες) με ένα εύρος δυναμικότητας παραγωγής $[X_{i\min}, X_{i\max}]$ το οποίο καθορίζει αντίστοιχα: (α) το τεχνικό ελάχιστο κάτω από το οποίο η μονάδα του παραγωγού i δεν μπορεί να λειτουργήσει και (β) η μέγιστη δυναμικότητα που κάθε παραγωγός μπορεί να αποδώσει στο Σύστημα. Υπάρχουν δύο κύριες κατηγορίες παραγωγών που μπορούν να συμμετάσχουν στην αγορά: (α) οι *Συμβατικοί Παραγωγοί* ηλεκτρικής ενέργειας που διαθέτουν μονάδες ατμοστρόβιλων, αεριοστρόβιλων κλπ καθένας από τους οποίους έχει ένα σημαντικό μέγεθος και όλοι μαζί αντιπροσωπεύουν τη μεγάλη πλειονότητα του παραγωγικού δυναμικού του Συστήματος, και (β) ένας *Εκπρόσωπος Παραγωγής* που συγκεντρώνει και

διαχειρίζεται το παραγωγικό δυναμικό πολλών μικρών διάσπαρτων παραγωγών (Κατανεμημένη Παραγωγή - DER) και υποβάλλει μία συνολική προσφορά προς τον Διαχειριστή εκ μέρους των μικρών παραγωγών. Η συνολική δυναμικότητα του Εκπροσώπου Παραγωγής αντιστοιχεί μόνο σε ένα μικρό μέρος της συνολικής δυναμικότητας του Συστήματος, ενώ η συνολική εγκατεστημένη ισχύς όλων των συμμετεχόντων υπερβαίνει την αναμενόμενη ζήτηση.

Το κόστος παραγωγής του Παραγωγού i είναι μια τετραγωνική συνάρτηση της ακόλουθης μορφής:

$$TC_i(x) = FC_i + a_i x + b_i x^2$$

όπου FC το σταθερό κόστος και a_i, b_i οι συντελεστές κόστους ($a_i, b_i > 0$).

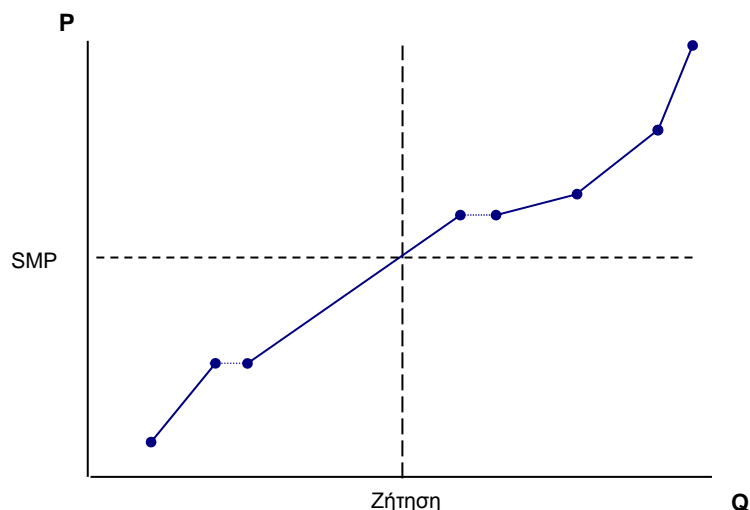


Σχήμα 21. Προσφορές των παραγωγών $F_i(x)$ και προσδιορισμός των αντίστοιχων ποσοτήτων παραγωγής (x_i) που κατανέμονται

Οι παραγωγοί υποβάλουν τις προσφορές τους στην ίδια μορφή όπως του οριακού τους κόστους⁶, δηλαδή μια αύξουσα γραμμική συνάρτηση (Σχήμα 21). Κάθε παραγωγός υποβάλλει προσφορά για το σύνολο της δυναμικότητάς του με τέτοιο τρόπο ώστε η προσφερόμενη τιμή, για οποιοδήποτε επίπεδο παραγωγής να μην υπερβαίνει το Ανώτατο Όριο Τιμής Προσφοράς (Price Cap) που έχει οριστεί από το Διαχειριστή. Το Ανώτατο Όριο καθορίζεται κατά προσέγγιση ως πολλαπλάσιο της τιμής όπου η αγορά ηλεκτρικής ενέργειας θα ισορροπούσε αν είχαν υποβληθεί ως προσφορές τα πραγματικά οριακά κόστη των παραγωγών.

Αν υποθέταμε ότι ζήτηση ήταν ανελαστική (σταθερή τιμή σε όλο το παίγνιο) ο Διαχειριστής λαμβάνοντας τις καμπύλες προσφοράς όλων των παραγωγών θα κατασκεύαζε τη συνολική καμπύλη συνολικής προσφοράς. Το σημείο όπου η καμπύλη προσφοράς τέμνει τη ζήτηση καθορίζει την τιμή της αγοράς-εκκαθάρισης (Σχήμα 22) και ο Διαχειριστής κατανέμει την παραγωγή με τον πιο αποδοτικό τρόπο. Στο σχήμα 21 απεικονίζεται ένα σύστημα με τρεις παραγωγούς ηλεκτρικής ενέργειας οι οποίοι υποβάλλουν προσφορές για το σύνολο της δυναμικότητάς τους και ο Διαχειριστής καλύπτει τη ζήτηση D , κατανέμοντας μόνο δύο από αυτούς (x_2 και x_3 αντίστοιχα, έτσι ώστε: $x_2 + x_3 = D$).

⁶ $MC_i(x) = a_i + 2b_i x$

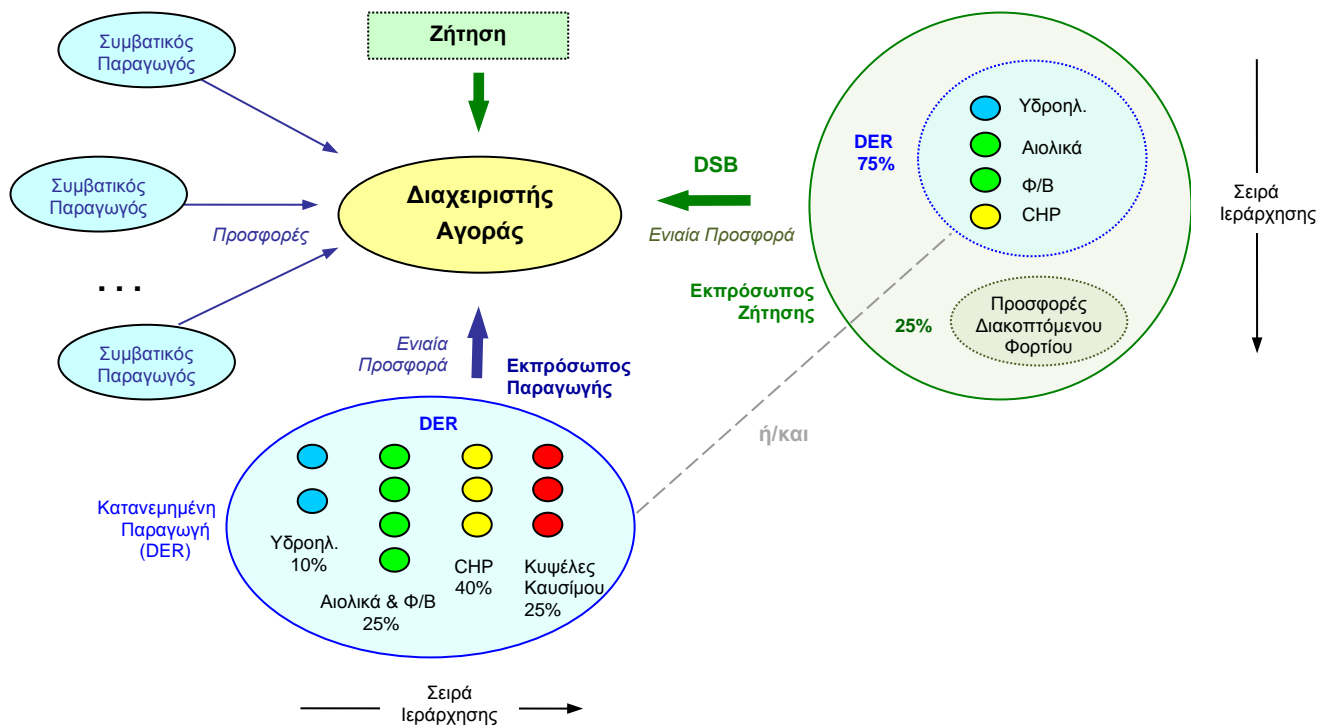


Σχήμα 22. Συνολική Καμπύλη Προσφοράς σύμφωνα με τις επιμέρους Προσφορές των Παραγωγών και Προσδιορισμός της Οριακή Τιμής Συστήματος.

Όπως και στα δύο προηγούμενα μοντέλα που περιγράφηκαν νωρίτερα, σε κάθε γύρο του παιγνίου, οι παραγωγοί μπορούν να τροποποιήσουν έναν μόνον από τους συντελεστές της καμπύλης προσφορά τους, αυξάνοντας, μειώνοντας ή διατηρώντας σταθερή την τιμή του. Η επιλογή γίνεται τυχαία γίνεται χρησιμοποιώντας ένα στοχαστικό προφίλ των πιθανών δράσεων, το οποίο σταδιακά και συνεχώς αναμορφώνεται, βάσει της αξιολόγησης των επιπτώσεων της τελευταίας αναπροσαρμογής του συντελεστή της καμπύλης προσφοράς στο εισόδημα του παραγωγού. Ένα παίγνιο που αποτελείται από ένα σύνολο μερικών εκατοντάδων χιλιάδων γύρων στην πράξη αντιπροσωπεύει ένα μικρό χρονικό διάστημα σε πραγματικό χρόνο με τη ζήτηση θεωρείται ως σταθερή, απομονώνοντας έτσι τη συμπεριφορά των παικτών από οποιαδήποτε επιρροή που θα προκαλούταν οι εποχιακές διακυμάνσεις της ζήτησης.

Το παίγνιο έχει πολλές παραμέτρους που μπορεί να ορισθούν, όπως το καθεστώς τιμολόγησης της αγοράς (Uniform Pricing ή Pay-as-Bid), η προσαρμοστικότητα και η ικανότητα αντίδρασης των παικτών, η αναλογία της ζήτησης ως προς τη συνολική εγκατεστημένη ισχύ και φυσικά το μίγμα των παραγωγών που συμμετέχουν στην αγορά όσον αφορά την τεχνολογία παραγωγής, το κόστος και τη δυναμικότητα παραγωγής. Ως αποτέλεσμα του παιγνίου θεωρούνται οι τιμές σύγκλιση της τιμής της ηλεκτρικής ενέργειας, των μεριδίων αγοράς και τα κέρδη αλλά και η συμπεριφορά των παικτών, όπως καταγράφεται μέσα από τις προσφορές τους.

Στο μοντέλο αυτό εισάγεται η συμμετοχή της Κατανεμημένης Παραγωγής Ενέργειας (DER) μέσω των Εκπροσώπων Παραγωγής (DER Aggregator) είτε από την πλευρά της προσφοράς είτε της ζήτησης όπως αναφέρθηκε ανωτέρω. Στην πρώτη περίπτωση ο Εκπρόσωπος Παραγωγής υποβάλλει προσφορές για την συνολική δυναμικότητα όλων των παραγωγών. Ο Εκπρόσωπος Παραγωγής που μοντελοποιείται εδώ λειτουργεί ως ένας ακόμη παίκτης με στοχαστικό κόστους παραγωγής, ανάλογα με τις διαφορετικές τεχνολογίες παραγωγής και της διαθεσιμότητας των μικροπαραγωγών που εκπροσωπεί. Το μίγμα των τεχνολογιών παραγωγής και ο αριθμός των παραγωγών DER είναι παραμετρική στο λογισμικό μοντελοποίησης του παιγνίου. Ο Εκπρόσωπος Ζήτησης (Demand Aggregator) ενεργεί ως ενδιάμεσος φορέας μεταξύ του Διαχειριστή και των παραγωγών DER όταν αυτοί συμμετέχουν από την πλευρά της ζήτησης μέσω των Προσφορών Ζήτησης. Οι Προσφορές Ζήτησης (Demand Side Bidding) δεν περιλαμβάνουν μόνο την δυναμικότητα παραγωγής DER που μπορούν να καλύψουν τοπικά τη Ζήτηση αλλά και προσφορές για μείωση φορτίου. Μια γενική επισκόπηση της αγοράς ηλεκτρικής ενέργειας διαμορφώθηκε από το EM Simulator απεικονίζεται στο Σχήμα 23.



Σχήμα 23. Γενική Επισκόπηση του Μοντέλου της Αγοράς με Προσφορές Ζήτησης και συμμετοχή Καταναμημένης Παραγωγής Ενέργειας (DER) .

4.3.2. Προσφορές Κατανεμημένης Παραγωγής Ενέργειας (DER generation)

Στο μοντέλο της αγοράς μελετήθηκε μια αγορά με σχετικά μικρό αριθμό συμβατικών παραγωγών και μεγαλύτερο αριθμό κατανεμημένης παραγωγής που συμμετέχουν στην χονδρεμπορική αγορά μέσω ενός Εκπροσώπου Παραγωγής (DER Aggregator). Η συνολική της δυναμικότητα παραγωγής που συγκεντρώνει και προσφέρει ο Εκπρόσωπος Παραγωγής (προεπιλεγμένη ρύθμιση) ετέθη περίπου ίση με την ικανότητα παραγωγής ενός συμβατικού παραγωγού (100 MW).

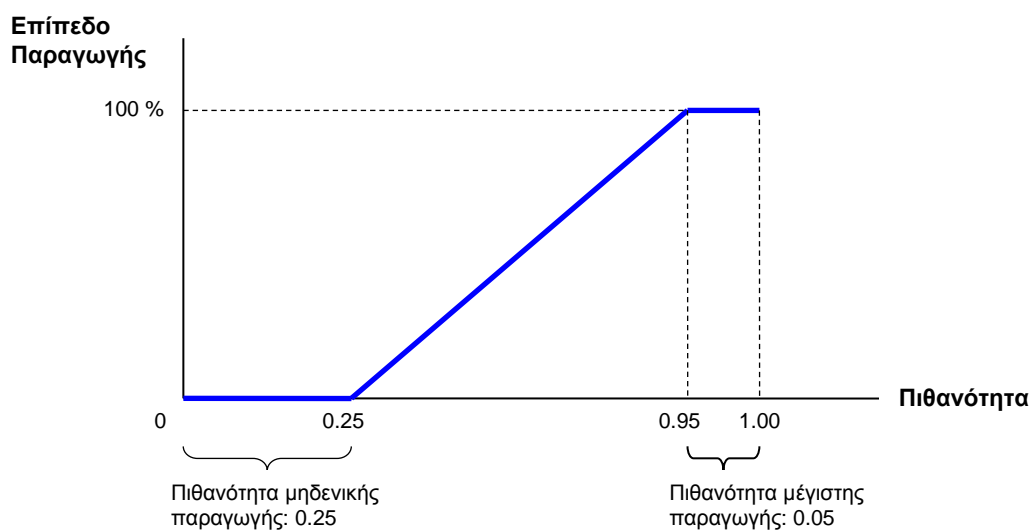
Το μίγμα τεχνολογίας της Κατανεμημένης Παραγωγής Ενέργειας (DER) του Εκπροσώπου Παραγωγής είναι η ακόλουθη:

1) Μικρές Μονάδες Συμπαγωγής (CHP):

Η συνάρτηση κόστους παραγωγής αυτής της τεχνολογίας DER θεωρείται ότι είναι του ίδιου τύπου με εκείνο ενός συμβατικού γεννήτρια αλλά λόγω της αυξημένης συνολικής απόδοσης της συμπαγωγής, υπάρχουν σημαντικές εξοικονομήσεις. Αυτό εκφράζεται ως ένα σταθερό ποσοστό που εφαρμόζεται για οποιοδήποτε επίπεδο δυναμικότητας παραγωγής και ορίζει την εξοικονόμηση κόστους σε σύγκριση με το κόστος του αντίστοιχου επιπέδου παραγωγής ενός συμβατικού παραγωγού. Η προκαθορισμένη τιμή της εξοικονόμησης κόστους της συμπαγωγής έχει οριστεί στο 40%. Η τεχνολογία της μικρής Συμπαγωγής θεωρούμε (προεπιλεγμένες τιμές του μοντέλου) ότι αντιπροσωπεύει το 40% της συνολικής παραγωγής που προσφέρει στην αγορά ο Εκπρόσωπος Παραγωγής (4 μικρές μονάδες συμπαγωγής που καθεμία από αυτές έχει δυναμικότητα παραγωγής 10 MW).

2) Σταθμοί ΑΠΕ μη Ελεγχόμενης Διαθεσιμότητας Παραγωγής

- i. **Μικρά Αιολικά Πάρκα / Ανεμογεννήτριες:** το μεταβλητό κόστος παραγωγής της ηλεκτροπαραγωγής από αιολική ενέργεια θεωρείται ίσο με το μηδέν ενώ παράλληλα εισάγουμε μια στοχαστική μεταβλητή που αντιστοιχεί στη διαθεσιμότητα παραγωγής. Η στοχαστική αυτή μεταβλητή παίρνει τιμές από μια συνάρτηση κατανομής πιθανότητας που ορίζεται παραμετρικά από το χρήστη (Σχήμα 4). Η λειτουργία αυτή μπορεί να καθορισθεί από τον χειριστή του μοντέλου ορίζοντας μια τιμή πιθανότητας για μηδενική διαθεσιμότητα παραγωγής (προκαθορισμένη τιμή: 0,25) και μια τιμή πιθανότητας για δυνατότητα εκμετάλλευσης του συνόλου της εγκατεστημένης δυναμικότητας παραγωγής (προκαθορισμένη τιμή: 0,05). Οι ενδιάμεσες τιμές πιθανότητας ακολουθούν γραμμική κατανομή.



Σχήμα 24. Συνάρτηση Κατανομής Πιθανότητας της Διαθεσιμότητας Παραγωγής από μικρά Αιολικά Πάρκα.

Σε κάθε επανάληψη (γύρο) του παιχνιδιού ο Εκπρόσωπος Παραγωγής DER λαμβάνει υπόψη για την προσφορά του τη μέση διαθεσιμότητα παραγωγής από τις ανεμογεννήτριες, όπως αυτή προκύπτει στοχαστικά από την συνάρτηση κατανομής πιθανότητας, παρόλο που τελικά μπορεί να παραχθεί λιγότερη ή περισσότερη ενέργεια από αυτές, ανάλογα με την πραγματική διαθεσιμότητα στη συγκεκριμένη επανάληψη. Η πραγματική παραγωγή υπολογίζεται τυχαία σε κάθε επανάληψη βάσει της συνάρτησης κατανομής πιθανότητας. Στην περίπτωση μεγαλύτερης διαθεσιμότητας, και ως εκ τούτου παραγωγής, ο εκπρόσωπος θα έχει χαμηλότερο συνολικό κόστος παραγωγής καθώς η αιολική ενέργεια έχει σχεδόν μηδενικό (μεταβλητό) κόστος παραγωγής. Στην περίπτωση μικρότερης πραγματικής παραγωγής θα είναι υποχρεωμένος είτε να χρησιμοποιήσει πιο ακριβές τεχνολογίες παραγωγής (όπως η συμπαραγωγή ή οι κυψέλες καυσίμου) για να καλύψει την ποσότητα που έχει δηλώσει στην προσφορά του, ή αν αυτό δεν είναι δυνατό (στην περίπτωση που όλη η προσφερόμενη παραγωγή του έχει ήδη καταναμεθεί) θα πρέπει να καταβάλει ένα ποσό για την ενέργεια που δεν μπορεί να παράξει. Το ποσό αυτό αντιπροσωπεύει είτε το «κόστος ασφάλισης» του Εκπροσώπου Παραγωγής ή στο πραγματικό κόστος που ο Διαχειριστής χρεώνει τον Εκπρόσωπο προκειμένου να καλύψει το έλλειμμα της ηλεκτρικής ενέργειας. Το κόστος αυτό εκφράζεται ως ένα ποσοστό επί της τρέχουσας Οριακής Τιμής και η προκαθορισμένη τιμή του έχει οριστεί σε 100%, δηλαδή διπλάσιο από την εκάστοτε Οριακή Τιμή Συστήματος.

- ii. **Φωτοβολταϊκοί Σταθμοί:** το ίδιο ακριβώς μοντέλο έχει εφαρμοστεί επίσης για αυτό το είδος της τεχνολογίας παραγωγής. Στην περίπτωση αυτή, οι προεπιλεγμένες τιμές για την συνάρτηση κατανομής πιθανότητας είναι 0,6 για την μη διαθεσιμότητα παραγωγής και 0,10 για μέγιστη διαθεσιμότητα. Το «κόστος ασφάλισης» που εφαρμόζεται είναι το ίδιο και για τις δύο τεχνολογίες ΑΠΕ μη ελεγχόμενης Παραγωγής.

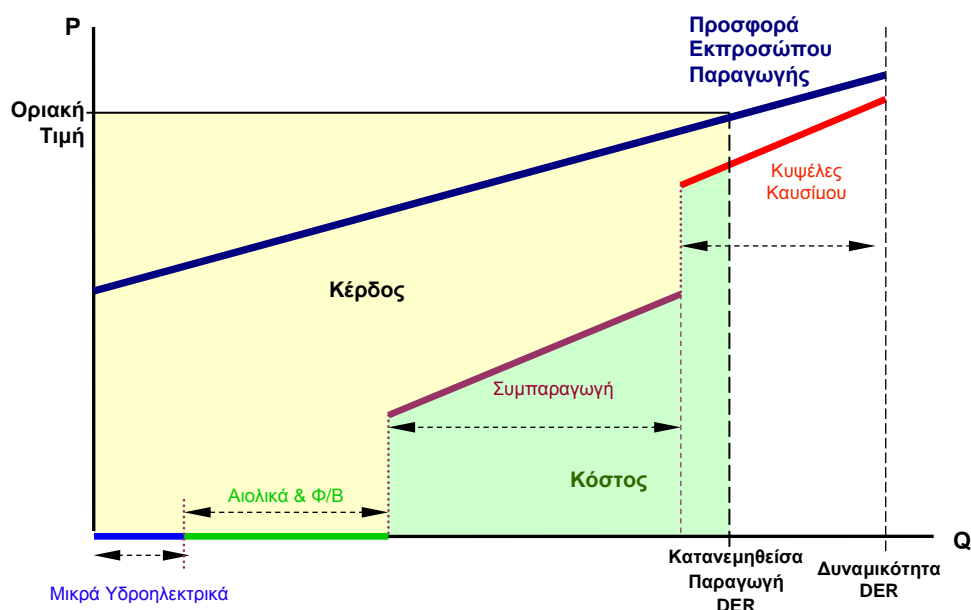
Στο μοντέλο ορίζεται ότι η δυναμικότητα παραγωγής που αντιστοιχεί στην δυναμικότητα των τεχνολογιών ΑΠΕ μη Ελεγχόμενης Παραγωγής αντιπροσωπεύει το 25% της συνολικής δυναμικότητας του χαρτοφυλακίου του (16% αιολική και 9% ηλιακή). Η εγκατεστημένη δυναμικότητα παραγωγής των αιολικών είναι 40 MW (8 αιολικά σταθμοί των 5 MW έκαστος), ενώ η μέση διαθέσιμη ισχύς είναι 16 MW (2 MW από κάθε αιολικό πάρκο). Αντίστοιχα, η εγκατεστημένη δυναμικότητα παραγωγής των φωτοβολταϊκών σταθμών είναι 36 MW (9 μονάδες των 4 MW η κάθε μία), ενώ η μέση εκτιμώμενη διαθέσιμη ισχύς είναι μόνο 9 MW (1 MW από κάθε μονάδα).

3) Μικροί Υδροηλεκτρικοί Σταθμοί

Η συγκεκριμένη τεχνολογία των μικρών υδροηλεκτρικών θεωρείται επίσης μηδενικού (μεταβλητού) κόστους και αντιπροσωπεύει το 10% της συνολικής δυναμικότητας του εκπροσώπου Παραγωγής (2 μονάδες μικρών υδροηλεκτρικών σταθμών των 5 MW η κάθε μία). Έχει οριστεί επίσης ο παράγοντας διαθεσιμότητα η οποία ωστόσο στο συγκεκριμένο μοντέλο έχει από προεπιλογή ρυθμιστεί στο 100% ενώ θα μπορούσε να πάρει και διαφορετικές τιμές, προκειμένου να προσομοιώσει τη διαθεσιμότητα παραγωγής σε ξηρά και κανονικά έτη.

4) Κυψέλες Καυσίμου (Fuel Cells)

Η συγκεκριμένη τεχνολογία παραγωγής θεωρείται η πιο ακριβή, από άποψης κόστους παραγωγής. Η συνάρτηση κόστους της συνδέεται με την συνάρτηση κόστους της Συμπαραγωγής, και στο μοντέλο για κάθε επίπεδο παραγωγής, το κόστος της υπολογίζεται ως ποσοστό επί του αντίστοιχου κόστους της συμπαραγωγής. Η προεπιλεγμένη τιμή που χρησιμοποιείται εδώ είναι 100%, δηλαδή ορίζεται να έχει διπλάσιο κόστος από το κόστος της συμπαραγωγής. Η προεπιλεγμένη συμμετοχή των κυψελών καυσίμου, στο χαρτοφυλάκιο του Εκπροσώπου Παραγωγής έχει οριστεί στο 25% (5 μονάδες των 5 MW η κάθε μία).

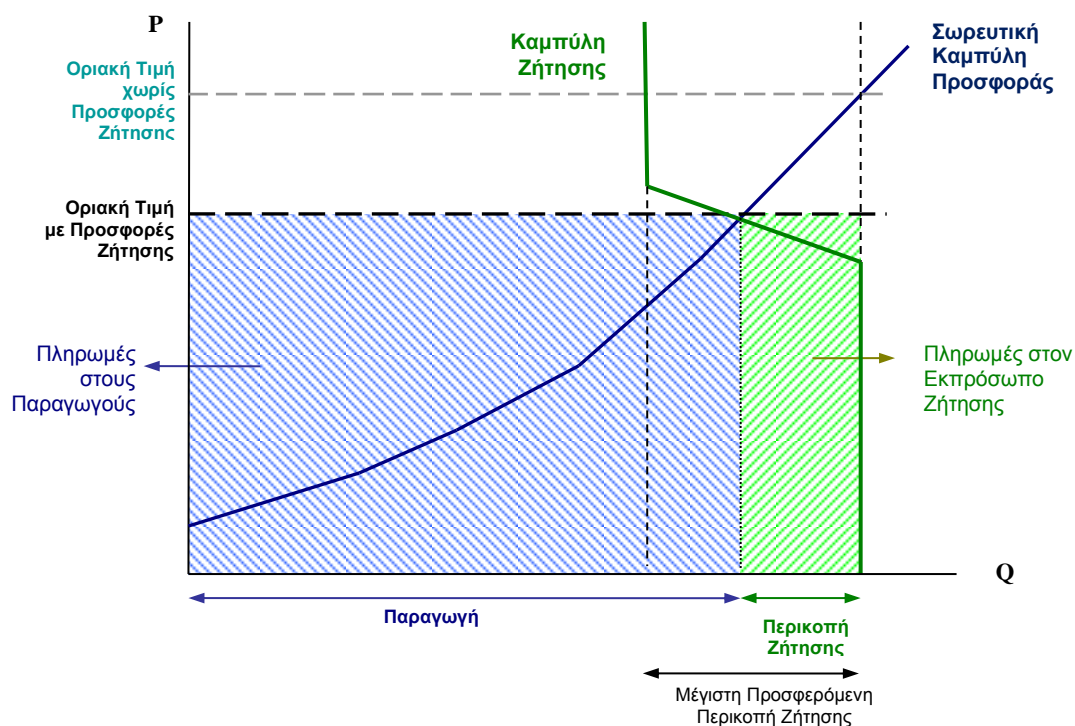


Σχήμα 25. Η συνάρτηση κόστους του Εκπροσώπου Παραγωγής, η μορφή της προσφοράς του και ο προσδιορισμός του κέρδους του.

Ο Εκπρόσωπος Παραγωγής υποβάλλει προσφορές πάντα για το σύνολο της δηλωμένης του δυναμικότητας παραγωγής. Η συνολική δηλούμενη περιλαμβάνει τη μέγιστη δυναμικότητα (εγκατεστημένη ισχύ) όλων των μονάδων του πλην των σταθμών ΑΠΕ μη Ελεγχόμενης Παραγωγής όπου εκεί λαμβάνεται υπόψη η μέση θεωρητικά διαθέσιμη δυναμικότητα όπως προκύπτει από τις συναρτήσεις κατανομής πιθανότητας. Στην πραγματικότητα, αυτό σημαίνει ότι σε κάθε επανάληψη η διαθέσιμη δυναμικότητά του θα μπορούσε να είναι μικρότερη ή μεγαλύτερη από τη δηλωμένη. Κατ'αυτόν τον τρόπο, ο Εκπρόσωπος Παραγωγής έχει μια σχετική ευελιξία στο να επιλέξει ποιες τεχνολογίες θα χρησιμοποιήσει για να παράξει την ποσότητα για την οποία έχει κατανεμηθεί. Ωστόσο έχει τον κίνδυνο να μην είναι σε θέση να παράξει την ποσότητα για την οποία κατανεμήθηκε και σε τέτοιες περιπτώσεις θα χρεωθεί από το Διαχειριστή το κόστος του ελλείμματός του. Στην πραγματικότητα, κάθε φορά που ο Εκπρόσωπος Παραγωγής κατανέμεται για μια συγκεκριμένη ποσότητα, εφαρμόζει μια εσωτερική σειρά ιεράρχησης (merit order) για τις διαθέσιμες ποσότητες του ανά τεχνολογία παραγωγής χρησιμοποιώντας ως κριτήριο το κόστος παραγωγής. Η σειρά ιεράρχησης των τεχνολογιών παραγωγής, τα αντίστοιχα κόστη και το προκύπτον κέρδος απεικονίζονται γραφικά στο Σχήμα 25.

4.3.3. Προσφορές Ζήτησης (Demand Side Bidding) με Περικοπή Φορτίου και χρήση Κατανεμημένης Παραγωγής Ενέργειας (DER)

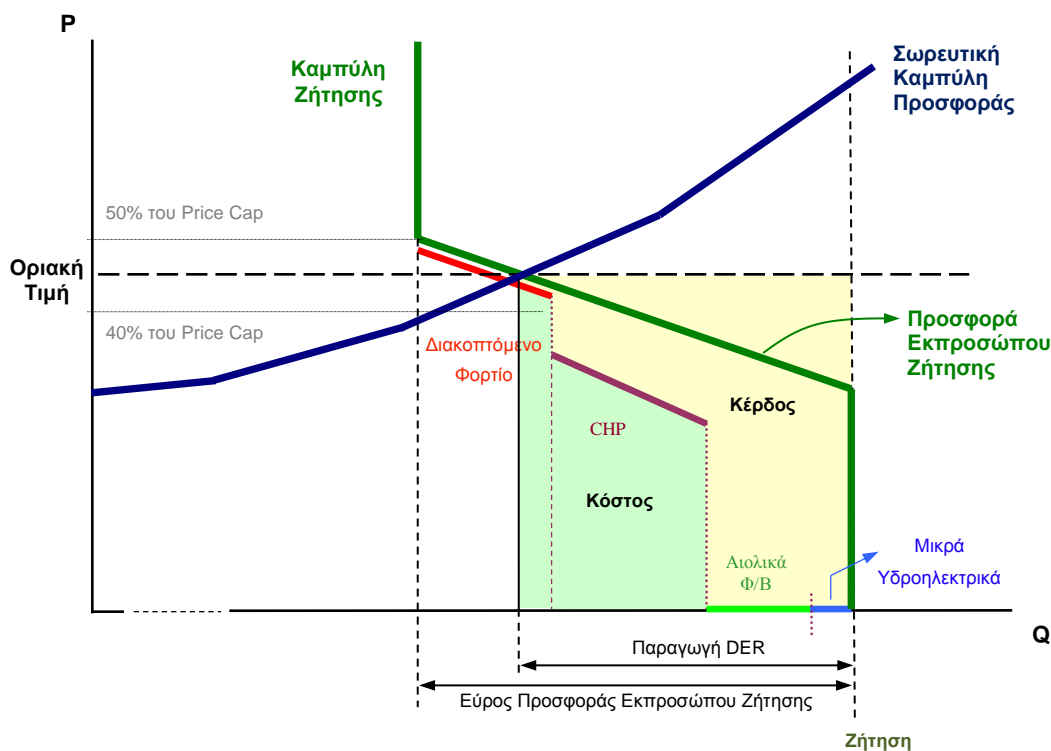
Στην περίπτωση που η Κατανεμημένη Παραγωγή Ενέργειας (DER) συμμετέχει στην αγορά από την πλευρά της Ζήτησης θεωρούμε ότι δεν υπάρχει Εκπρόσωπος Παραγωγής από την πλευρά της Προσφοράς που να χρησιμοποιεί τέτοιου είδους παραγωγούς, ενώ μια νέα επιχειρηματική οντότητα εισέρχεται στο παιχνίδι: ο Εκπρόσωπος Ζήτησης που υποβάλλει προσφορές μείωσης/κάλυψης του φορτίου στον Διαχειριστή για διαφορετικά επίπεδα τιμών. Ο Εκπρόσωπος Ζήτησης θα μειώσει πραγματικά μόνο ένα μέρος του φορτίου και το υπόλοιπο θα το καλύψει με τοπική Κατανεμημένη Παραγωγή Ενέργειας (DER). Έτσι, ο Εκπρόσωπος Ζήτησης δρα στην αγορά ως διαχειριστής του φορτίου για λογαριασμό των καταναλωτών και είτε καλύπτει το φορτίο με την παραγωγή DER σε χαμηλότερες τιμές ή επιτυγχάνει για λογαριασμό τους πληρωμές για μείωση (περικοπή) του φορτίου. Το κόστος του περιλαμβάνει το κόστος παραγωγής DER, αλλά και το κόστος για τη μείωση του φορτίου (την τιμή στην οποία οι πελάτες του συμφώνησαν να μειώσουν το φορτίο τους). Ο προσδιορισμός της Οριακής τιμής Συστήματος με Προσφορές Ζήτησης (*Demand Side Bidding*) απεικονίζεται στο Σχήμα 26.



Σχήμα 26. Προσδιορισμός της Οριακής Τιμής Συστήματος σε μια αγορά με Προσφορές Ζήτησης.

Στο μοντέλο έχει ορισθεί από τη συνολική προσφερόμενη δυναμικότητας κάλυψης/μείωσης της Ζήτησης (Προσφορά Ζήτησης), η δυναμικότητα παραγωγής από Κατανεμημένη Παραγωγή Ενέργειας (DER) να αντιπροσωπεύει το 75% ενώ το υπόλοιπο 25% να αντιστοιχεί σε μείωση του φορτίου. Το τμήμα της Προσφοράς που αντιστοιχεί στην Κατανεμημένη Παραγωγή Ενέργειας έχει ακριβώς την ίδια δομή και συναρτήσεις κόστους με τον αντίστοιχο καταμερισμό δυναμικότητας που είχε και προσφορά του Εκπρόσωπου Παραγωγής (DER Aggregator), δηλαδή το 10% μικρά υδροηλεκτρικά, 25% ΑΠΕ μη Ελεγχόμενης Διαθεσιμότητας Παραγωγής (16% αιολική και 9% ηλιακή) και 40% Συμπαραγωγή. Η μόνη διαφορά είναι ότι για το υπόλοιπο 25% ο Εκπρόσωπος Ζήτησης χρησιμοποιεί στην προσφορά του προσφορές Περικοπής Φορτίου αντί για παραγωγή από Κυψέλες Καυσίμου. Ωστόσο, δεδομένου ότι η μη ελεγχόμενη διαθεσιμότητα παραγωγής από ΑΠΕ είναι μέρος της προσφερόμενης δυναμικότητάς του, ο Εκπρόσωπος Ζήτησης έχει την ίδια αβεβαιότητα με τον Εκπρόσωπο Παραγωγής όσον αφορά την ικανότητά του να καλύψει την προσφορά του και ταυτόχρονα την ίδια ευελιξία να επιλέξει ανάμεσα στις υπόλοιπες πηγές του.

Ο Εκπρόσωπος Ζήτησης ξεκινά το παίγνιο με μια προσφορά η οποία είναι πανομοιότυπης μορφής με τη συνάρτηση κόστους του διακοπτόμενου φορτίου, αλλά επεκτείνεται στο σύνολο της δυναμικότητας της προσφοράς του. Το κόστος του διακοπτόμενου φορτίου είναι προφανές ότι αποτελεί το πάνω μέρος (υψηλότερη από την άποψη των τιμών) της προσφοράς του. Η ανώτερη και η κατώτερη τιμή της διακοπτόμενης καμπύλη φορτίου ενώ είναι παραμετρικές στο μοντέλο, στα συγκεκριμένα παίγνια έχουν οριστεί από προεπιλογή στο 50% και 40% της τιμής του Ανώτατου Ορίου Προσφοράς (Price Cap) αντίστοιχα. Αυτές οι δύο τιμές, μαζί με τη δυναμικότητα καθορίζουν την καμπύλη της Προσφορά Ζήτησης. Η άνω και κάτω τιμή μπορεί να εισαχθεί στο μοντέλο είτε ως συνάρτηση του Ανώτατου Ορίου Προσφοράς ή ως απόλυτες τιμές. Το Σχήμα 27 απεικονίζει τη συνάρτηση κόστους του Εκπροσώπου Ζήτησης, τη μορφή προσφοράς του, τον προσδιορισμό της Οριακής Τιμής καθώς και το τελικό κόστος και τα έσοδα του κατά την πρώτο κλείσιμο της αγοράς. Στη συνέχεια του παιγνίου ο Εκπρόσωπος Ζήτησης αναπροσαρμόζει της συνολική του προσφορά με τον ίδιο τρόπο που το κάνουν και οι λοιποί παίκτες, βάσει δηλαδή του προσαρμοστικού στοχαστικού αλγορίθμου απόφασης μεταβολής των συντελεστών προσφοράς, ο οποίος λαμβάνει υπόψη τα αποτελέσματα των προηγούμενων προσφορών και διαμορφώνει σταδιακά το στοχαστικό προφίλ των επιλογών απόφασης.



Σχήμα 27. Η συνάρτηση κόστους του Εκπροσώπου Ζήτησης, η μορφή της προσφοράς του και ο προσδιορισμός του κέρδους του.

Ο προσδιορισμός του κέρδους για τον κάθε μικροπαραγωγό (ή διακοπτόμενο καταναλωτή) εξαρτάται από τον τρόπο που ορίζουμε την οντότητα του Εκπροσώπου (Παραγωγής ή Ζήτησης). Μια επιλογή είναι να μοντελοποιήσουμε τον Εκπρόσωπο ως κοινοπραξία αυτών που εκπροσωπεί (παραγωγοί DER ή/και διακοπτόμενοι πελάτες), όπου το σύνολο των εσόδων και το συνολικό κόστος (συμπεριλαμβανομένων των εξόδων ασφάλισης της μη Ελεγχόμενης Παραγωγής ΑΠΕ) καθορίζει το κέρδος. Στη συνέχεια, το κέρδος μοιράζεται μεταξύ των συμμετεχόντων ανάλογα με τη συμβολή τους. Η άλλη επιλογή είναι να υπολογιστεί χωριστά το κέρδος του καθενός, λαμβάνοντας υπόψη το κόστος τους και τη συνεισφορά τους στα έσοδα. Το προεπιλεγμένο καθεστώς για τον Εκπρόσωπο Παραγωγής είναι το μοντέλο τύπου κοινοπραξίας ενώ για τον Εκπρόσωπο Ζήτησης ο δεύτερος τύπος μοντέλου. Το είδος του μοντέλου που επιλέγεται επηρεάζει μόνο τα εκτιμώμενα κέρδη της κάθε κατηγορίας παραγωγών DER (ή διακοπτόμενων πελατών) που συμμετέχουν στην προσφερόμενη δυναμικότητα του Εκπροσώπου.

5. Το Λογισμικό Προσομοίωσης

Το λογισμικό **EM Simulator** που αναπτύχθηκε για τη μοντελοποίηση των παιγνίων στις αγορές ηλεκτρικής ενέργειας που περιγράφηκαν ανωτέρω είναι μια αυτόνομη εφαρμογή που «τρέχει» σε λειτουργικό σύστημα Microsoft Windows. Έχει αναπτυχθεί σε προγραμματιστικό περιβάλλον Visual Fortran και αποτελείται από 29.000 γραμμές κώδικα προγραμματισμού.

Αποτελείται δε, από δύο κύρια μέρη: α) το τμήμα που προσομοιώνεται το θεωρητικό μοντέλο που αποτελεί την καρδιά του συστήματος και β) το τμήμα που διαχειρίζεται τις χρηστικές λειτουργίες του προγράμματος όπως το περιβάλλον επικοινωνίας με το χρήστη, τη δημιουργία διαγραμμάτων και τη δυναμική προσομοίωση των παιγνίων.

Η πρώτη ομάδα περιλαμβάνει τα ακόλουθα αρχεία (συνολικά 8.361 γραμμές κώδικα):

- <i>asyn_basic_m.f90</i>	660	γραμμές κώδικα
- <i>asyn_m.f90</i>	103	γραμμές κώδικα
- <i>asyn_one_sys_m.f90</i>	2.810	γραμμές κώδικα
- <i>asyn_two_sys_oneoffer_m.f90</i>	1.877	γραμμές κώδικα
- <i>asyn_two_sys_split_m.f90</i>	2.487	γραμμές κώδικα
- <i>main.f90</i>	424	γραμμές κώδικα

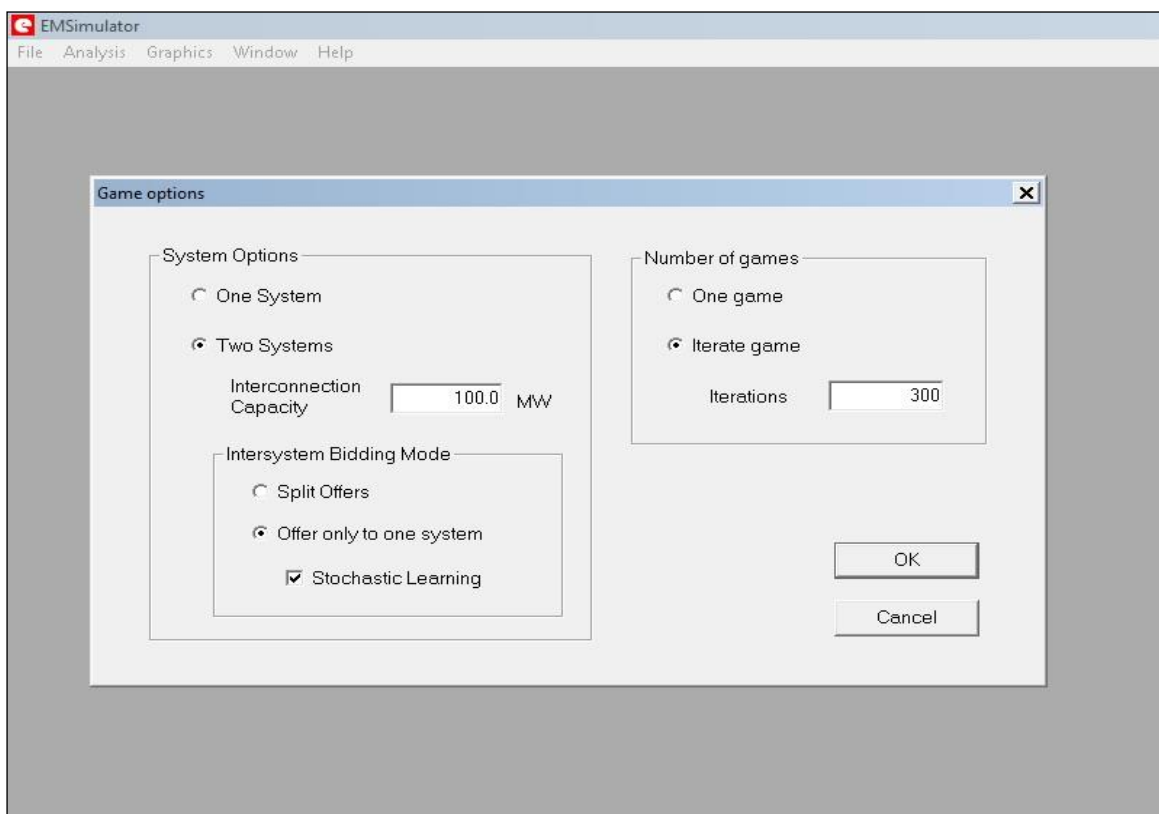
Η δεύτερη ομάδα περιλαμβάνει τα ακόλουθα αρχεία (20.639 γραμμές κώδικα συνολικά):

- <i>about.f90</i>	161	γραμμές κώδικα
- <i>analysis.f90</i>	1.547	γραμμές κώδικα
- <i>anim_m.f90</i>	201	γραμμές κώδικα
- <i>animation.f90</i>	4.079	γραμμές κώδικα
- <i>emsimulator_g_m.f90</i>	701	γραμμές κώδικα
- <i>emsimulator_m.f90</i>	137	γραμμές κώδικα
- <i>graph.f90</i>	4.286	γραμμές κώδικα
- <i>graph_m.f90</i>	29	γραμμές κώδικα
- <i>io.f90</i>	2.074	γραμμές κώδικα
- <i>MainWndProc.f90</i>	1.449	γραμμές κώδικα
- <i>MDIWndProc.f90</i>	737	γραμμές κώδικα
- <i>player_data_classical.f90</i>	33	γραμμές κώδικα
- <i>player_data_der.f90</i>	1.327	γραμμές κώδικα
- <i>player_data_dsb.f90</i>	1.425	γραμμές κώδικα
- <i>pleasewait.f90</i>	134	γραμμές κώδικα
- <i>TextWndProc.f90</i>	159	γραμμές κώδικα
- <i>util.f90</i>	64	γραμμές κώδικα
- <i>winmain.f90</i>	247	γραμμές κώδικα
- <i>xftgdi.f90</i>	1.799	γραμμές κώδικα

Στο Παράρτημα της παρούσης διατριβής για λόγους οικονομίας χώρου παρατίθεται μόνο ο κώδικας της πρώτης ομάδας αρχείων.

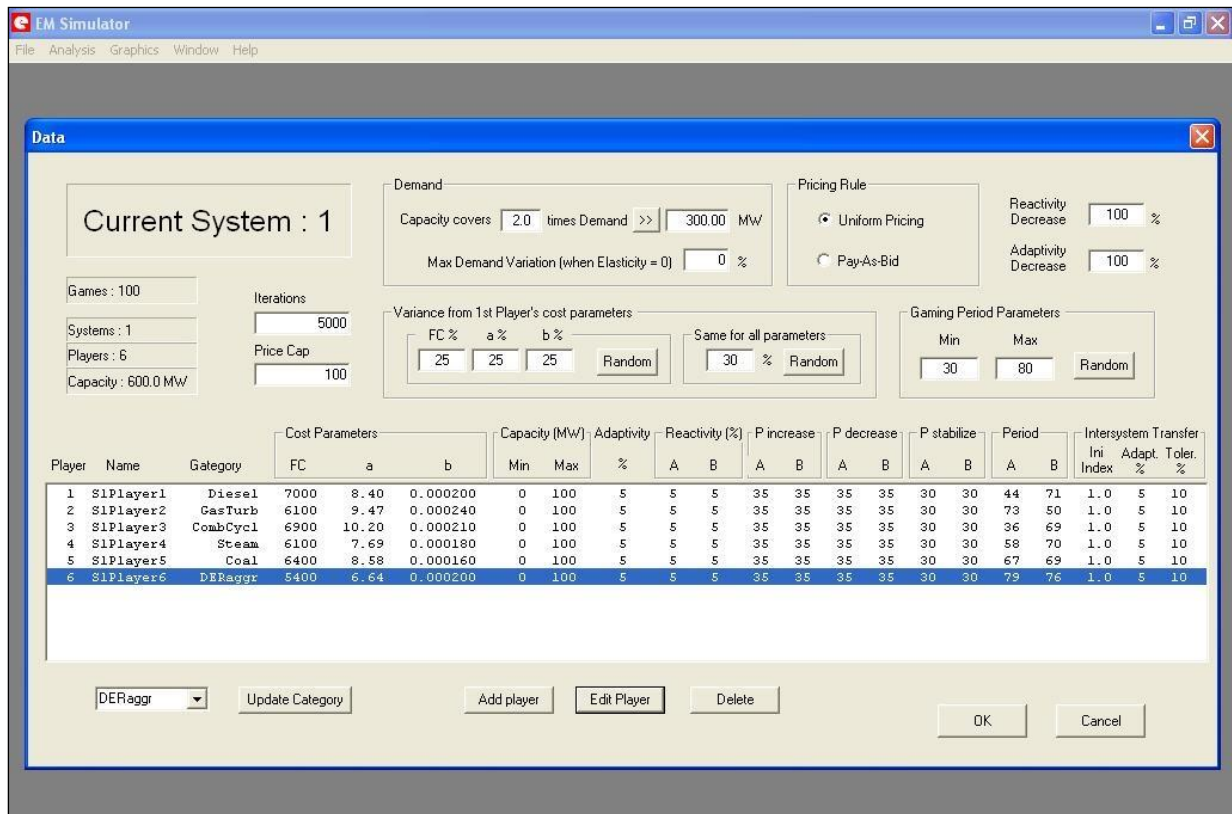
Το λογισμικό EM Simulator μπορεί να «τρέξει» αυτόνομα χωρίς δηλαδή να υπάρχει ανάγκη να έχει ο χρήστης εγκατεστημένη τη Visual Fortran καθώς η εφαρμογή δημιουργεί ένα εκτελέσιμο αρχείο που μπορεί να τρέξει οπουδήποτε και να εξαγει αποτελέσματα σε μορφή ASCII ή *.bmp, αν πρόκειται για γραφικά αποτελέσματα. Ωστόσο ο χρήστης μέσα από την εφαρμογή έχει τη δυνατότητα να δει στην οθόνη του όχι μόνο γραφήματα αλλά και δυναμικές προσομοιώσεις (animation) των παιγνίων.

Είναι εύκολο στη χρήση του και έχει τρεις κύριες λειτουργικές δραστηριότητες. Η πρώτη είναι η εισαγωγή δεδομένων όπου ο χρήστης ρυθμίζει τις παραμέτρους του παιχνιδιού, τον τύπο και τα χαρακτηριστικά των παικτών και τους γενικούς κανόνες της αγοράς. Η δεύτερη είναι η υπολογιστική διαδικασία η οποία διαρκεί από μερικά δευτερόλεπτα έως λίγα λεπτά, ανάλογα με τον αριθμό των επαναλήψεων και την πολυπλοκότητα της μοντελοποιημένης αγοράς (αριθμός παικτών, εκπροσώπων κ.λπ.). Η τελευταία αφορά στην παρουσίαση των αποτελεσμάτων (πίνακες, γραφήματα και δυναμική προσομοίωση). Ο χρήστης μπορεί να «τρέξει» σε επανάληψη ένα παιχνίδι βλέποντας τα λεπτομερή αποτελέσματα (γύρο προς γύρο) ή να επιλέξει να επαναλάβει το ίδιο παιχνίδι όσες φορές επιθυμεί και να πάρει στατιστικά συγκεντρωτικά αποτελέσματα. Λόγω των πολλών παραμέτρων εισόδου του μοντελοποιημένης αγοράς, το λογισμικό EM Simulator έχει τη δυνατότητα να αποθηκεύσει ένα παιχνίδι ή ένα σύνολο παιχνιδιών (εισροών και αποτελεσμάτων) σε μια συγκεκριμένη μορφή αρχείου, ώστε να μπορεί να φορτωθεί και πάλι για αναθεώρηση ή περαιτέρω προσομοιώσεις.

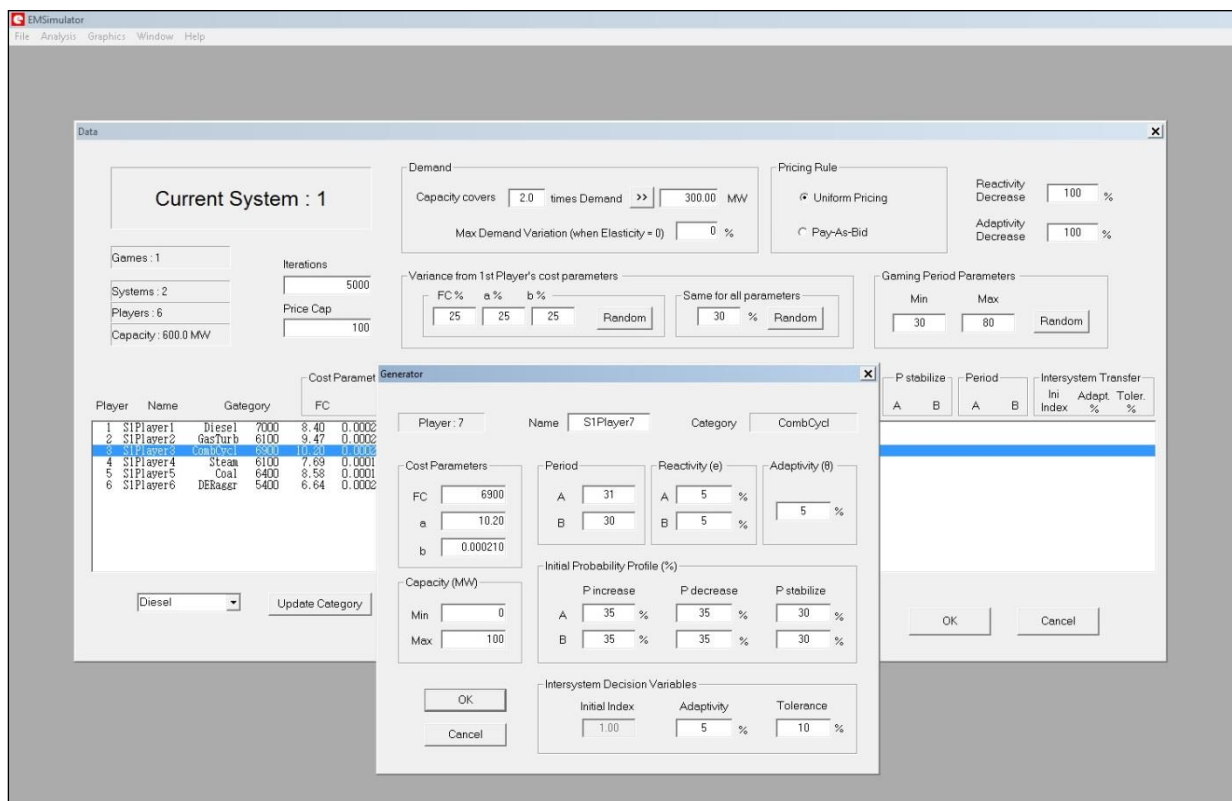


Σχήμα 28. EM Simulator: Η αρχική οθόνη γενικών επιλογών του παιχνιδιού.

Στο Σχήμα 29.α παρουσιάζεται ένα στιγμιότυπο από την κύρια φόρμα εισαγωγής δεδομένων, όπου ο χρήστης μπορεί να εισάγει τα δεδομένα σχετικά με την αγορά, τους παίκτες και τους βασικούς κανόνες του παιχνιδιού. Στο επόμενο Σχήμα 29.β Στο Σχήμα 30 φαίνεται η εισαγωγή δεδομένων και χαρακτηριστικών των Εκπροσώπων Παραγωγής και Ζήτησης. Η δομή του κάθε Εκπροσώπου το είδος και ο αριθμός των παραγωγών DER που συμμετέχουν στο παιχνίδι ορίζονται εδώ. Στο Σχήμα 31 απεικονίζονται (α) οι οθόνες δυναμικής προσομοίωσης του παιχνιδιού όπου ο χρήστης μπορεί να «τρέξει» το παιχνίδι γύρο προ γύρο και (β) επιλεγμένα γραφήματα με τα αποτελέσματα του παιχνιδιού. Πέρα από το γραφικό τύπο των αποτελεσμάτων το λογισμικό EM Simulator μπορεί να εξάγει λεπτομερή αποτελέσματα (raw data) που μπορούν να επεξεργαστούν και να αναλυθούν περαιτέρω με άλλα εργαλεία εξειδικευμένου λογισμικού.



Σχήμα 29.α EM Simulator: Η οθόνη εισαγωγής δεδομένων και γενικών χαρακτηριστικών του παιγνίου και της εποπτικής εικόνας παραγωγών που συμμετέχουν στο παίγνιο.



Σχήμα 29.β EM Simulator: Η οθόνη εισαγωγής δεδομένων και χαρακτηριστικών ενός παραγωγού.

DER Aggregator

Player: 6 Name: S1Player6 Category: DERaggr

DER Generators:

- CHP**
 - # 4 Capacity: 10 MW FC: 5400
 - Savings: 40 % a: 6.64 b: 0.000200
- Fuel Cells**
 - # 5 Capacity: 5 MW CHPcost +: 100 %
- Small Hydro**
 - # 2 Capacity: 5 MW Availability: 100 %
- Wind Parks**
 - # 8 Capacity: 5 MW No Production: 25 %
 - Av. Estimated: 2 MW Max Production: 5 %
- PV**
 - # 9 Capacity: 4 MW No Production: 60 %
 - Av. Estimated: 1 MW Max Production: 10 %

Intermittent Generation non Availability Cost on SMP: 100 %

Period

A	79
B	76

Reactivity (e)

A	5 %
B	5 %

Adaptivity (B)

	5 %
--	-----

Initial Probability Profile (%)

P increase	P decrease	P stabilize
A 35 %	35 %	30 %
B 35 %	35 %	30 %

Intersystem Decision Variables

Initial Index	Adaptivity	Tolerance
1.00	5 %	10 %

CHP	40 %	40 MW
Fuel Cells	25 %	25 MW
Small Hydro	10 %	10 MW
Intermittent	25 %	25 MW
DER Aggregator Total Capacity		100 MW

Aggregator acts as generators consortium

OK Cancel

Demand Aggregator

Player: 6 Name: Demand Aggr. Category: DSBaggr

Interruptible Load Bid

Capacity (MW): 25 Price Range: % of Price Cap High: 50 Absolute Value Low: 40

Curve Slope: >> 0.40

Demand Side Bidding DER

- CHP**
 - # 4 Capacity: 10 MW FC: 5400
 - Savings: 40 % a: 6.64 b: 0.000200
- Small Hydro**
 - # 2 Capacity: 5 MW Availability: 100 %
- Wind Parks**
 - # 8 Capacity: 5 MW No Production: 25 %
 - Av. Estimated: 2 MW Max Production: 5 %
- PV**
 - # 9 Capacity: 4 MW No Production: 60 %
 - Av. Estimated: 1 MW Max Production: 10 %

Intermittent Generation non Availability Cost on SMP: 100 %

Period

A	79
B	76

Reactivity (e)

A	5 %
B	5 %

Adaptivity (B)

	5 %
--	-----

Initial Probability Profile (%)

P increase	P decrease	P stabilize
A 35 %	35 %	30 %
B 35 %	35 %	30 %

Intersystem Decision Variables

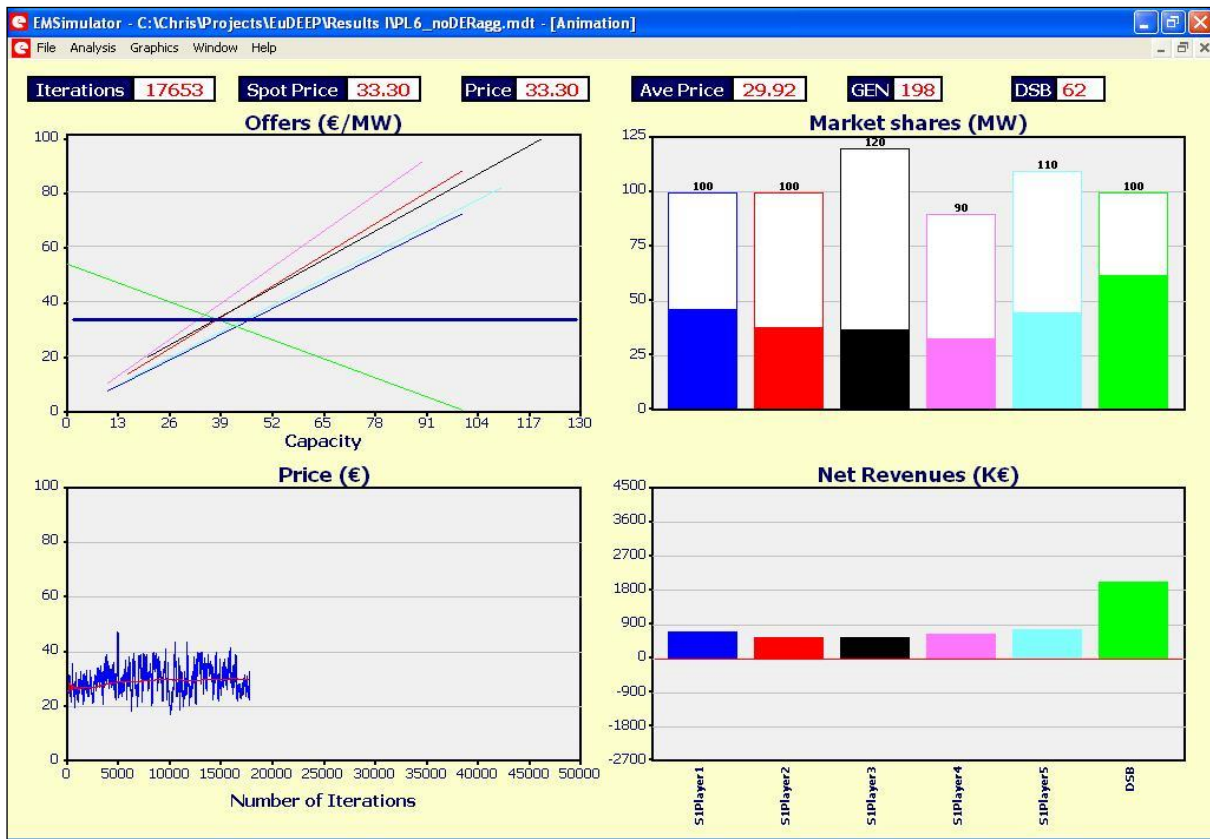
Initial Index	Adaptivity	Tolerance
1.00	5 %	10 %

CHP	40 %	40 MW
Small Hydro	10 %	10 MW
Intermittent	25 %	25 MW
DER generation	75 %	75 MW
Interruptible Load	25 %	25 MW
Demand Aggregator Total Capacity		100 MW

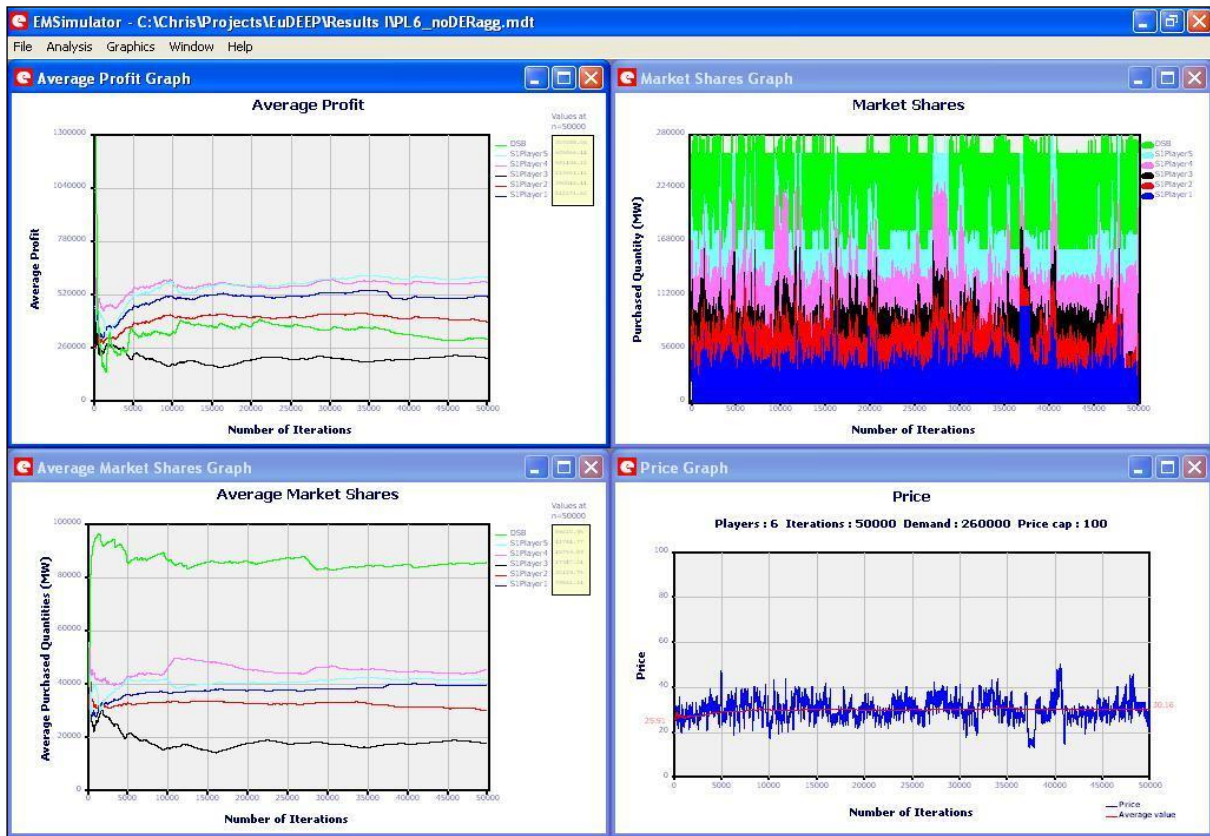
DER generators act as a consortium

OK Cancel

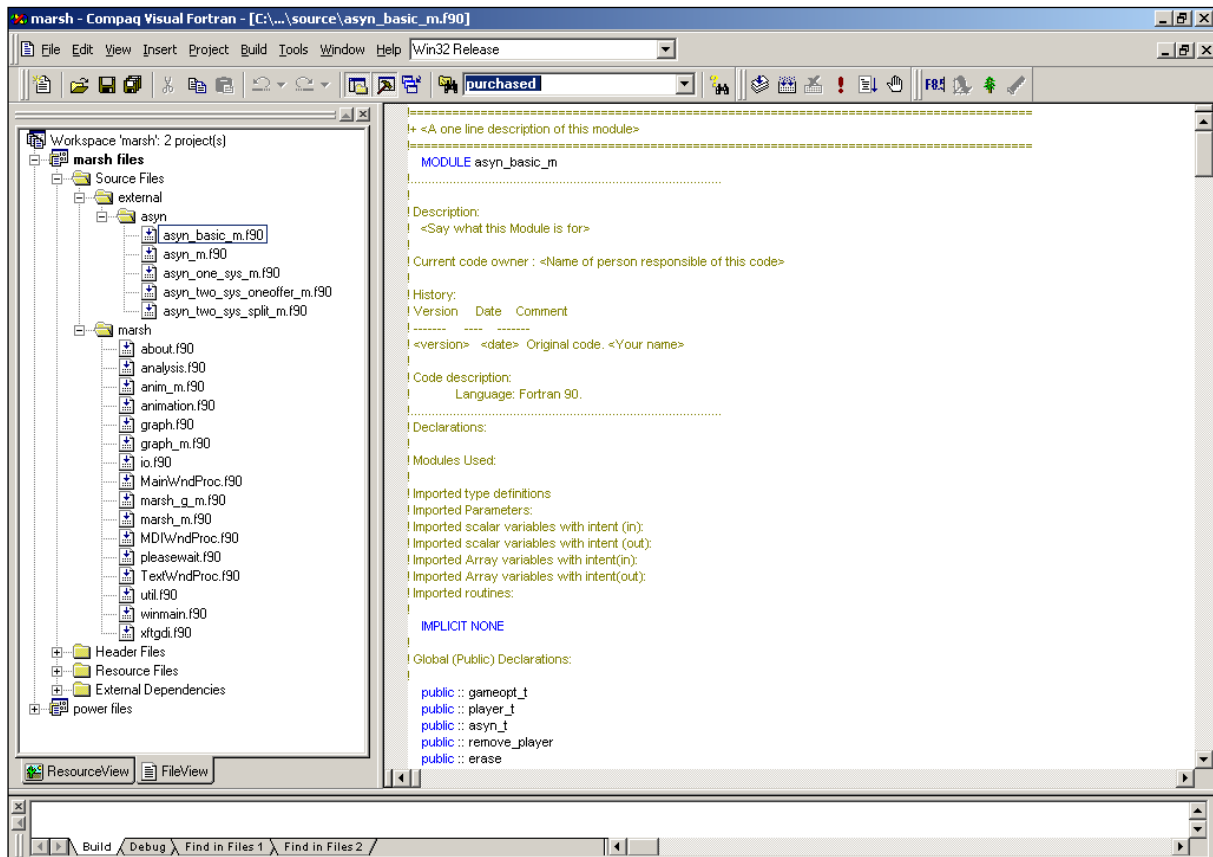
Σχήμα 30. EM Simulator: Οι οθόνες εισαγωγής δεδομένων και χαρακτηριστικών των Εκπροσώπων Παραγωγής/Ζήτησης που συμμετέχουν στο παιχνίδι.



Σχήμα 31. EM Simulator: Η Οθόνη δυναμικής προσομοίωσης παιγνίου σε ένα Σύστημα με Προσφορές Ζήτησης.



Σχήμα 32. EM Simulator: Εμφάνιση αποτελεσμάτων και διαγραμμάτων παιγνίου σε ένα Σύστημα με Προσφορές Ζήτησης.



Σχήμα 33. EM Simulator: Η δομή του προγράμματος όπως φαίνεται στο περιβάλλον της Visual Fortran .

6. Παιγνία και Αποτελέσματα

6.1. Παιγνία σε ένα Ηλεκτρικό Σύστημα

6.1.1. Παιγνίο 1: Οριακή Τιμή Συστήματος και Αριθμός Παικτών με τετραγωνική συνάρτηση οριακού κόστους σε Uniform Pricing τιμολόγηση

Με βάση το μοντέλο της αγοράς ηλεκτρικής ενέργειας και του αντίστοιχου παιγνίου όπως περιγράφηκε στα προηγούμενα, εφαρμόστηκε μια περιορισμένη μορφή του παιγνίου με συγκεκριμένα χαρακτηριστικά, που στόχο είχε την εξαγωγή γενικών συμπερασμάτων που αφορούσαν κυρίως στη σχέση της ΟΤΣ και του αριθμού των παραγωγών που συμμετέχουν στο παίγνιο, καθώς και στον τρόπο με τον οποίο οι παραγωγοί μεταβάλλουν τις καμπύλες προσφορών στην προσπάθειά τους να μεγιστοποιήσουν το κέρδος τους. Οι παράμετροι του παιγνίου είχαν στατικό χαρακτήρα ενώ παράλληλα η διαφορετικότητα των παικτών περιορίστηκε στο ελάχιστο.

Συγκεκριμένα, ως μέθοδος τιμολόγησης εφαρμόστηκε η Οριακή Τιμή του συστήματος (Uniform Pricing), θεωρήθηκε ότι οι παίκτες είχαν *τετραγωνικό*⁷ οριακό κόστος και ότι η ζήτηση ηλεκτρικής ενέργειας D παραμένει σταθερή καθ'όλη τη διάρκεια του παιγνίου. Οι παίκτες που συμμετείχαν σε αυτό είχαν την ίδια δυναμικότητα παραγωγής (χωρίς τεχνικό ελάχιστο παραγωγής) και το σύνολο της δυναμικότητάς τους είναι ίσο με το τριπλάσιο της ζήτησης. Οι παράμετροι των συναρτήσεων κόστους των παραγωγών έχουν τυχαία διάταξη εντός ενός εύρους $\pm 30\%$ γύρω από τις τιμές του πρώτου παίκτη-παραγωγού (κεντρικές τιμές) και οι τιμές των βημάτων e και θ θεωρήθηκαν ίσες για κάθε παίκτη και συντελεστή:

$$e_{A_i} = e_{B_i} = e_{C_i} = 2\% \quad \text{και} \quad \theta = 5\%$$

Οι αρχικές τιμές των P^{in} , P^{de} , P^{st} που αντιστοιχούν στην πιθανότητα επιλογής της κάθε δράσης λαμβάνονται επίσης ίσες για κάθε παίκτη και κάθε συντελεστή, ενώ διαφέρουν μόνο ανά είδος δράσης:

$$P_i^{in} = P_i^{de} = 35\% \quad \text{και} \quad P_i^{st} = 30\%$$

Η περίοδος με την οποία εναλλάσσονταν οι προσαρμογές των συντελεστών της καμπύλης προσφοράς για τον κάθε παίκτη επιλέχθηκαν τυχαία από ένα εύρος τιμών μεταξύ 30-80 επαναλήψεων. Οι ίδιες τυχαίες αυτές τιμές των περιόδων για κάθε συντελεστή και παίκτη που εφαρμόζονται στο πρώτο παίγνιο, εφαρμόζονται και σε όλα τα υπόλοιπα. Ως Ανώτατο Όριο Τιμής Προσφοράς ορίστηκε η τιμή 150, που αντιπροσωπεύει περίπου το δεκαπλάσιο της Αρχικής Τιμής Ισορροπίας. Πραγματοποιήθηκαν 11 διαφορετικά παίγνια με 3, 4, 5, 6, 7, 8, 9, 10, 12, 15 και 20 παίκτες αντίστοιχα, το καθένα από τα οποία επαναλήφθηκε 20 φορές. Το κάθε παίγνιο συνίσταται από 32.000 συνεχόμενες προσφορές κατά τη διάρκεια των οποίων οι παίκτες προσαρμόζουν τις καμπύλες προσφοράς τους με βάση την εμπειρία που αποκτούν από τις προηγούμενες προσφορές τους. Στον Πίνακα της επόμενης σελίδας δίδονται οι τιμές των μεγεθών που χαρακτηρίζουν κάθε παραγωγό. Για κάθε παίγνιο κατεγράφησαν οι τιμές των ακόλουθων μεγεθών:

- Η Οριακή Τιμή του Συστήματος (ΟΤΣ)
- Οι Πωλούμενες Ποσότητες (Μερίδια Αγοράς)
- Συντελεστές A , B , C της καμπύλης προσφοράς για τον κάθε παίκτη.
- Κινητοί Μέσοι Όροι όλων των προηγούμενων

Ιδιαίτερη σημασία για την εξέταση της συμπεριφοράς των παικτών και τη διενέργεια συγκρίσεων έχουν η πρώτη τιμή της ΟΤΣ, όταν δηλαδή οι καμπύλες προσφοράς των παικτών ελάχιστα διαφέρουν από τις πραγματικές τους καμπύλες κόστους, τα μερίδια αγοράς που αποκτούν οι παίκτες στον πρώτο γύρο καθώς και οι συντελεστές κόστους a_i , b_i , c_i του κάθε παίκτη.

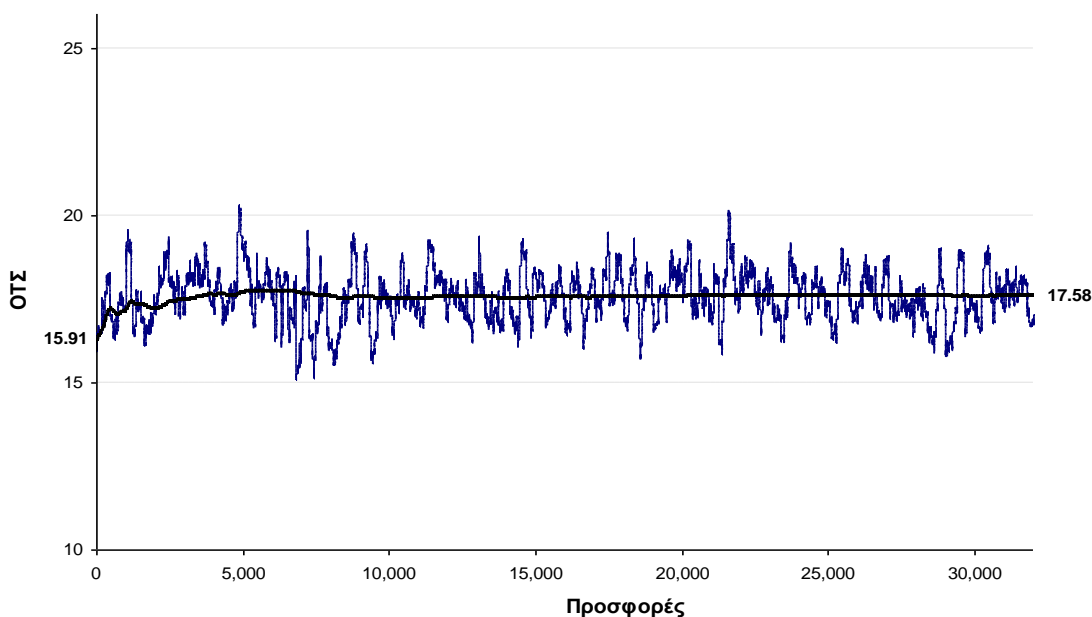
⁷ Για την απεικόνιση του οριακού κόστους ενός παραγωγού ηλεκτρικής ενέργειας αρκεί μια γραμμική συνάρτηση. Εδώ επιλέγεται τετραγωνική προκειμένου να δοθούν περισσότεροι βαθμοί ελευθερίας στους παραγωγούς για να μελετηθεί καλύτερα η συμπεριφορά τους στο παίγνιο.

Παραγωγός	Παράμετροι Συνάρτησης Κόστους			Δυναμικότητα Παραγωγής		Βήμα θ	Βήμα e (A, B, C)	Περίοδος Συντελεστών Προσφοράς		
	a	b	c	Ελάχιστο	Μέγιστο			A	B	C
1	10.00	0.0050	0.00055	0	300	5%	2%	44	71	48
2	11.54	0.0040	0.00050	0	300	5%	2%	73	69	30
3	8.97	0.0060	0.00067	0	300	5%	2%	36	69	69
4	10.23	0.0055	0.00067	0	300	5%	2%	58	70	69
5	11.94	0.0042	0.00052	0	300	5%	2%	67	69	72
6	7.92	0.0036	0.00058	0	300	5%	2%	79	76	62
7	11.76	0.0062	0.00045	0	300	5%	2%	70	59	66
8	9.57	0.0050	0.00070	0	300	5%	2%	58	48	66
9	10.26	0.0055	0.00047	0	300	5%	2%	69	52	73
10	7.43	0.0037	0.00039	0	300	5%	2%	52	78	48
11	7.34	0.0035	0.00049	0	300	5%	2%	79	59	53
12	12.01	0.0064	0.00058	0	300	5%	2%	34	43	61
13	12.91	0.0041	0.00051	0	300	5%	2%	39	61	57
14	7.12	0.0062	0.00060	0	300	5%	2%	32	41	56
15	10.94	0.0048	0.00063	0	300	5%	2%	58	44	74
16	12.73	0.0040	0.00062	0	300	5%	2%	50	64	77
17	7.72	0.0044	0.00045	0	300	5%	2%	39	68	43
18	9.31	0.0058	0.00040	0	300	5%	2%	46	65	42
19	9.96	0.0061	0.00047	0	300	5%	2%	60	33	63
20	12.40	0.0043	0.00045	0	300	5%	2%	69	64	73

Σχήμα 34. Πίνακας με τα χαρακτηριστικά μεγέθη των παραγωγών που συμμετέχουν στο παίγνιο.

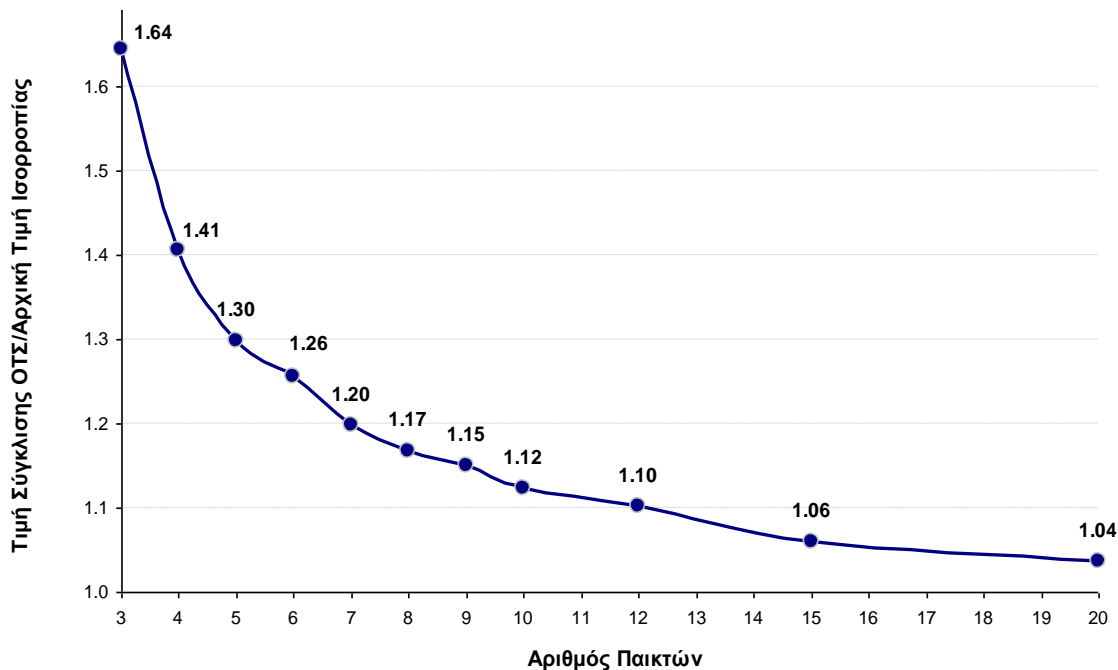
Οριακή Τιμή Συστήματος και αριθμός Παραγωγών

Παρά τις συνεχείς διακυμάνσεις που εμφάνισε η Οριακή Τιμή Συστήματος κατά τη διάρκεια των παιγνίων, οι μέσες τιμές της συγκλίνουν, σε όλα ανεξαιρέτως τα παίγνια που πραγματοποιήθηκαν, και μάλιστα σε τιμές μεγαλύτερες των αντίστοιχων Αρχικών Τιμών Ισορροπίας (ΑΤΙ). Στο Σχήμα 35 απεικονίζεται μια τυπική διακύμανση της ΟΤΣ και της κινητής της μέσης τιμής σε ένα παίγνιο με 10 παραγωγούς.



Σχήμα 35. Η εξέλιξη της Οριακής Τιμής του Συστήματος και η κινητή Μέση Τιμή της σε παίγνιο 10 Παραγωγών.

Ο αριθμός των συμμετεχόντων, στο παίγνιο, παραγωγών φάνηκε να επηρεάζει άμεσα την σύγκλιση αυτή τόσο ως προς την τιμή όσο και ως προς την ταχύτητά της. Σε παίγνια, δηλαδή, με λιγότερους παραγωγούς η σύγκλιση της μέσης τιμής της ΟΤΣ εμφανιζόταν ύστερα από μεγαλύτερο αριθμό συνεχόμενων προσφορών και σε επίπεδα τιμών σαφώς υψηλότερα. Στο Σχήμα 36, όπου παρουσιάζονται οι λόγοι των τιμών σύγκλισης των ΟΤΣ ως προς τις Αρχικές Τιμές Ισορροπίας για το σύνολο των παιγνίων που διεξήχθησαν, γίνεται αμέσως αντιληπτή η σχέση μεταξύ της ΟΤΣ και του αριθμού των παραγωγών.



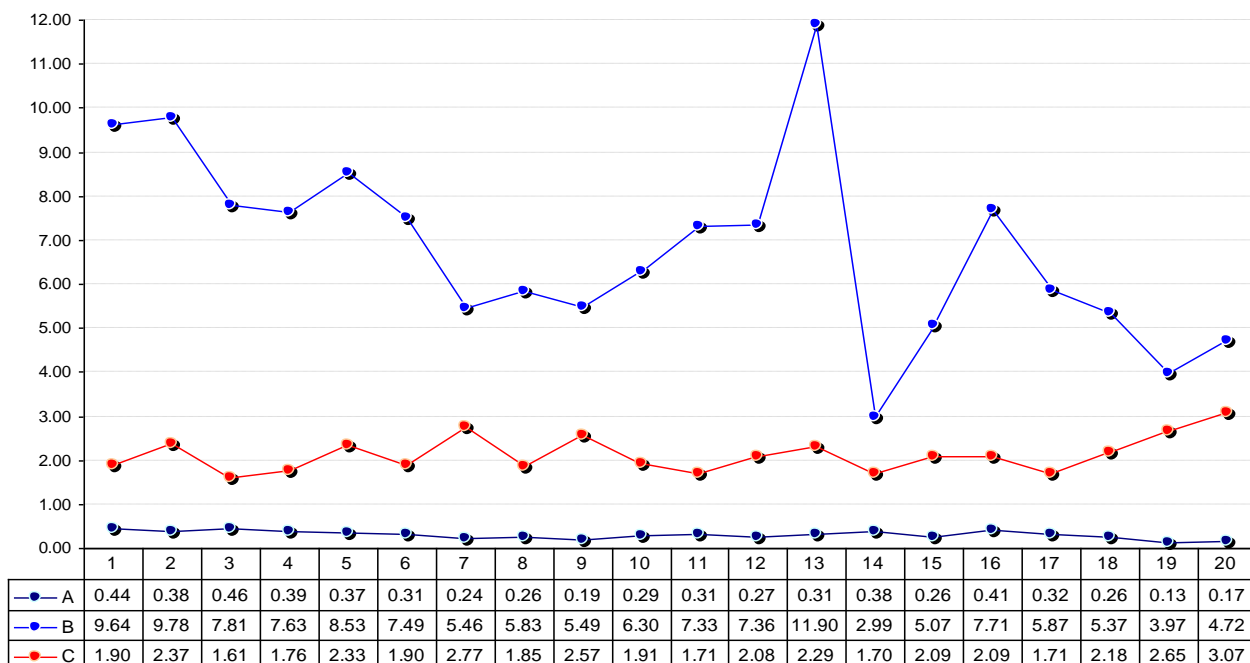
Σχήμα 36. Σχέση Αριθμού Παραγωγών-Παικτών και τιμής σύγκλισης της ΟΤΣ στα αντίστοιχα παίγνια.

Καμπύλες Προσφοράς

Η συμπεριφορά των παραγωγών κατά τη διάρκεια των παιγνίων όπως αυτή αποτυπώνεται στην πορεία της εξέλιξης των συναρτήσεων προσφοράς που αυτοί υποβάλλουν στον Ανεξάρτητο Διαχειριστή του Συστήματος, αν και χαρακτηρίζεται από διακυμάνσεις, φαίνεται τελικά να ακολουθεί κάποιο πρότυπο. Συγκεκριμένα παρατηρείται πως οι παραγωγοί τείνουν να μειώνουν τον συντελεστή A ενώ το αντίθετο κάνουν με τους συντελεστές B και C . Επίσης φαίνεται πως ο αριθμός των παραγωγών που συμμετέχουν στο παίγνιο επηρεάζει τον βαθμό στον οποίο μεταβάλλουν τον καθένα από αυτούς τους συντελεστές σε σχέση με τους συντελεστές κόστους των.

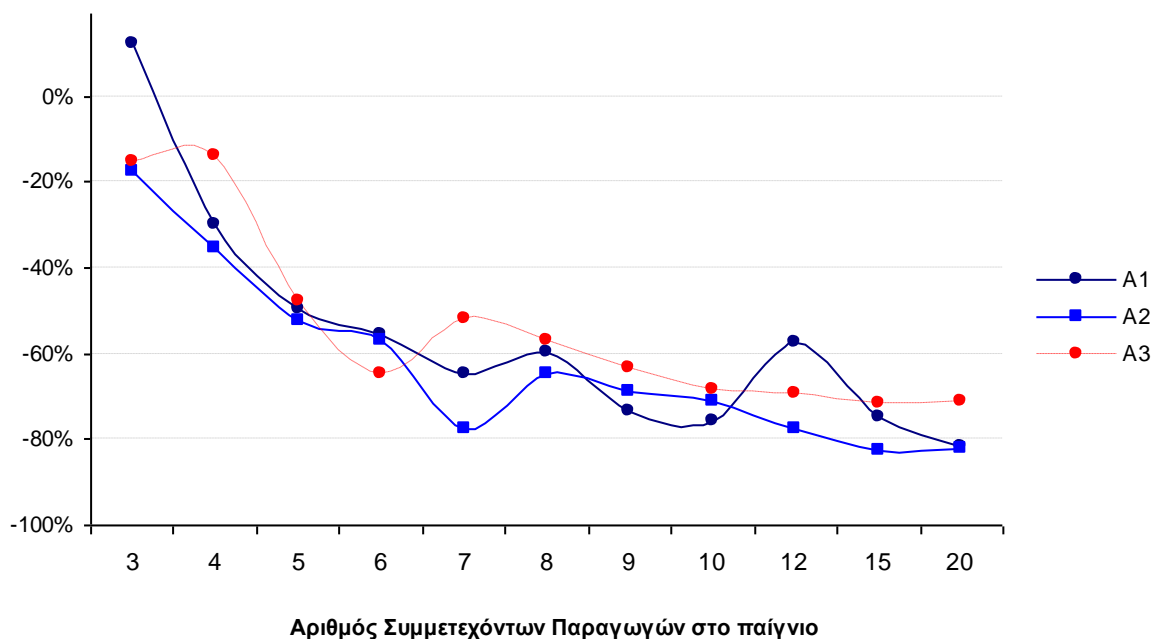
Στο Σχήμα 37, όπου παρουσιάζεται η σχέση της κινητής μέσης τιμής στην οποία συγκλίνει ο καθένας από τους συντελεστές A , B και C του κάθε Παραγωγού προς τους αντίστοιχους συντελεστές κόστους του (a , b , c), διακρίνεται καθαρά η ομοιομορφία της συμπεριφοράς των Παραγωγών, καθώς για όλους ανεξαιρέτως τους Παραγωγούς

- η τιμή του Συντελεστού A συγκλίνει σε χαμηλότερα επίπεδα (~ 70% χαμηλότερα κατά μ.ο. για όλους τους παραγωγούς) από αυτά των αντιστοίχων συντελεστών κόστους a .
- η τιμή του Συντελεστού B συγκλίνει σε επίπεδα κατά πολύ υψηλότερα (~ 580% κατά μ.ο. για όλους τους παραγωγούς) από τις τιμές των αντιστοίχων συντελεστών κόστους b .
- η τιμή του Συντελεστού C επίσης συγκλίνει σε τιμές υψηλότερες (~ 113% κατά μ.ο. για όλους τους παραγωγούς) από τις τιμές των αντιστοίχων συντελεστών κόστους c .

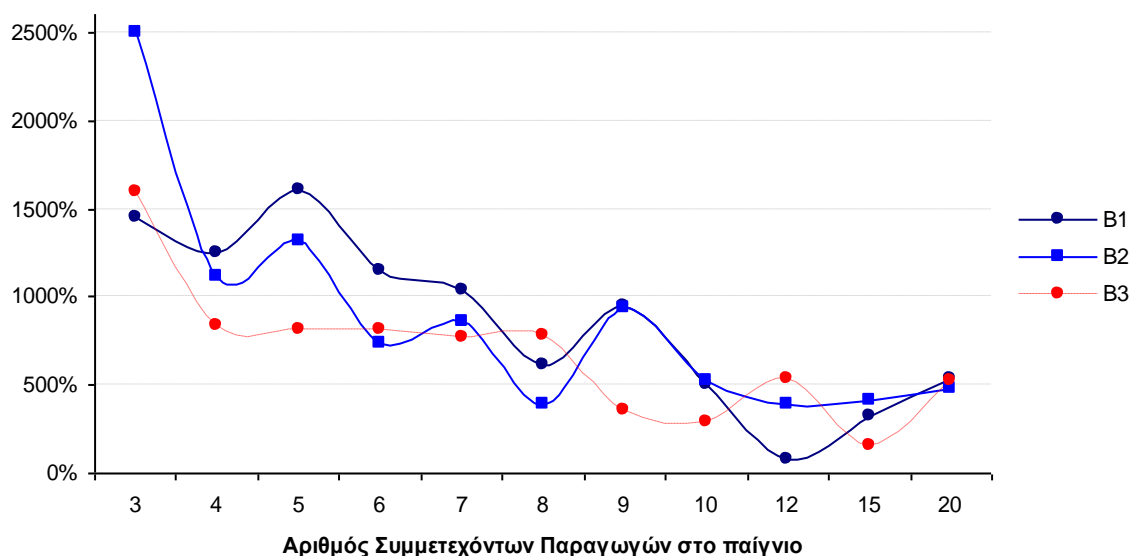


Σχήμα 37. Σχέση τιμών σύγκλισης της Μ.Τ. των Συντελεστών A , B και C και των αντιστοίχων συντελεστών κόστους a , b , c για κάθε παραγωγό.

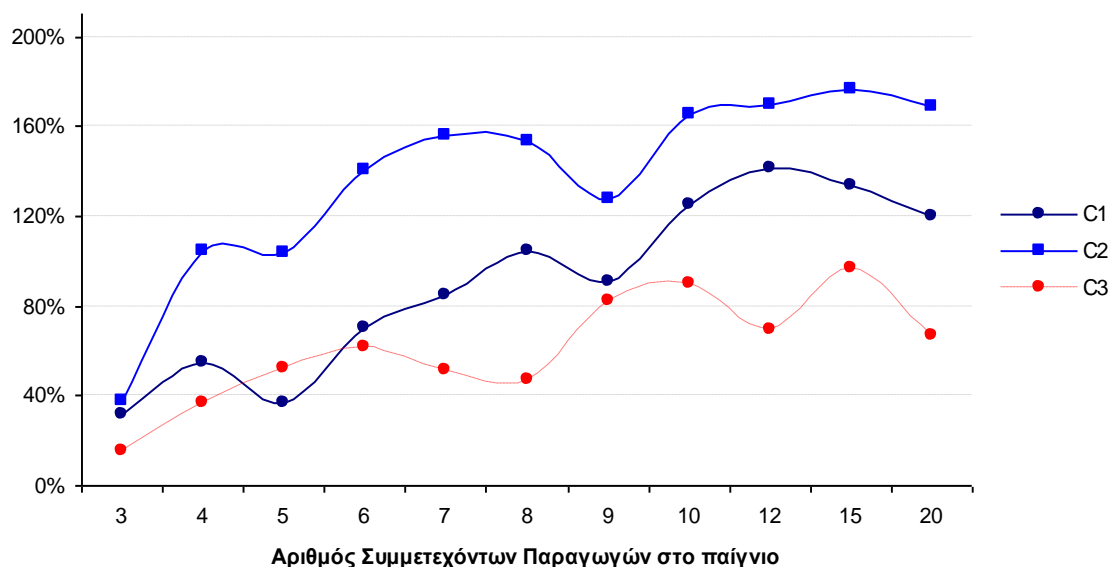
Ενδιαφέρον ωστόσο παρουσιάζει η εξέταση της έντασης της συμπεριφοράς αυτής των συντελεστών σε συνάρτηση με τον αριθμό των παικτών που συμμετέχουν στο παίγνιο. Παρατηρείται λοιπόν, ότι η τάση της συμπεριφοράς του συντελεστή *B* αμβλύνεται (μικρότερη αύξηση) καθώς αυξάνει ο ανταγωνισμός, ενώ η τάση της συμπεριφοράς των συντελεστών *A* και *C* εντείνεται, δηλαδή ο συντελεστής *A* τείνει να **μειωθεί** ακόμη περισσότερο και ο συντελεστής *C* να **αυξηθεί** περισσότερο (*hockey stick bidding*) όταν αυξάνει ο αριθμός των συμμετεχόντων παραγωγών. Ενδεικτικά είναι τα ακόλουθα διαγράμματα (Σχήμα 38) που αφορούν στους παίκτες 1, 2 και 3 οι οποίοι συμμετείχαν σε όλα τα παίγνια που διεξήχθησαν.



Σχήμα 38.α Σχέση της τιμής σύγκλισης της Μ.Τ. του Συντελεστή *A* ως προς την τιμή του αντιστοίχου συντελεστή κόστους *a*, για τους Παραγωγούς 1,2 και 3 ανά είδος παιγνίου.



Σχήμα 38.β Σχέση της τιμής σύγκλισης της Μ.Τ. του Συντελεστή *B* ως προς την τιμή του αντιστοίχου συντελεστή κόστους *b*, για τους Παραγωγούς 1,2 και 3 ανά είδος παιγνίου.



Σχήμα 38.γ Σχέση της τιμής σύγκλισης της Μ.Τ. του Συντελεστή C ως προς την τιμή του αντιστοίχου συντελεστή κόστους c, για τους Παραγωγούς 1,2 και 3 ανά είδος παιγνίου.

Είναι προφανές από τα αποτελέσματα των παιγνίων πως αν και αρχικώς οι παίκτες δείχνουν μια διάθεση κερδοσκοπίας χωρίς όριο, ο ανταγωνισμός τους κάνει να μετριάζουν τις απαιτήσεις τους και να τείνουν - όταν ο ανταγωνισμός είναι πλέον έντονος (μεγάλος αριθμός παικτών) - να προσφέρουν σε τιμές παραπλήσιες του οριακού τους κόστους. Ωστόσο, ακόμη και σε συνθήκες έντονου ανταγωνισμού οι παίκτες τείνουν να υποβάλλουν υψηλές προσφορές για το τελευταίο τμήμα της δυναμικότητάς τους γεγονός που μπορεί να οδηγήσει σε ιδιαίτερα υψηλές τιμές αγοράς αν η ζήτηση αυξηθεί αρκετά. Ας σημειωθεί ότι στα παίγνια που διεξήχθησαν εδώ, η αναλογία της συνολικής δυναμικότητα ως προς τη ζήτηση παραμένει σταθερή, ανεξαρτήτως του αριθμού των παραγωγών. Σε διαφορετική περίπτωση η ένταση των φαινομένων αυτών μεταβάλλεται αντίστοιχα, όπως είναι λογικό και αναμενόμενο.

6.1.2. Παιγνίο 2: Οριακή Τιμή Συστήματος και Αριθμός Παικτών με γραμμική συνάρτηση οριακού κόστους και διαφορετικό βήμα προσαρμογής σε Uniform Pricing τιμολόγηση

Τα χαρακτηριστικά και οι τροποποιήσεις του παιγνίου

Στο παίγνιο όπου οι παραγωγοί θεωρούμε ότι έχουν **γραμμικό** οριακό κόστος και υποβάλλουν προσφορά υπό μορφή γραμμικής επίσης συνάρτησης, οι μεταβλητές του παιγνίου μειώνονται καθώς οι συντελεστές των συναρτήσεων και όλες οι μεταβλητές (πιθανότητες, βήμα προσαρμογής) που σχετίζονται με αυτούς μειώνονται δραστικά. Συγκεκριμένα, οι παίκτες που τώρα έχουν μια συνάρτηση συνολικού κόστους της μορφής:

$$TC_i(x) = FC_i + a_i x + b_i x^2$$

και αντίστοιχα υποβάλλουν προσφορές της μορφής :

$$F_{i_1}(x) = A_{i_1} + B_{i_1} x$$

όπου για την πρώτη προσφορά⁸ ισχύει: $A_{i_1} = a_i$ και $B_{i_1} = 2b_i$

Ως προς τα υπόλοιπα το παίγνιο έχει ακριβώς την ίδια φιλοσοφία με αυτό των τριών συντελεστών με τη διαφορά ότι δεν υπάρχει ο τετραγωνικός συντελεστής C και όλες οι σχετιζόμενες με αυτόν μεταβλητές. Επιπλέον θεωρούμε ότι κάθε παίκτη αντιστοιχεί ένα σταθερό κόστος FC_i .

Η ζήτηση ηλεκτρικής ενέργειας D παραμένει σταθερή καθ'όλη τη διάρκεια του παιγνίου και το σύνολο της μέγιστης δυναμικότητάς τους είναι ίσο με το *διπλάσιο* της ζήτησης. Οι παραγωγοί είχαν το ίδιο τεχνικό ελάχιστο και εύρος δυναμικότητας παραγωγής (7 MW - 15 MW), χρησιμοποιούν δηλαδή όλοι το ίδιο καύσιμο και την ίδια τεχνολογία παραγωγής (π.χ. μικροί πετρελαϊκοί θερμοηλεκτρικοί σταθμοί). Οι τιμές των παραμέτρων κόστους και του σταθερού κόστους έχουν τυχαία διάταξη εντός ενός εύρους $\pm 25\%$ γύρω από τις τιμές του πρώτου παίκτη-παραγωγού (κεντρικές τιμές) οι οποίες είναι οι ακόλουθες:

$$FC_1 = 7.000 \quad a_1 = 8,40 \quad b_1 = 0,00020$$

Όλοι οι παίκτες θεωρούνται ισοδύναμοι σε σχέση με την προσαρμοστική και μαθησιακή τους ικανότητα, και για το λόγο αυτό οι τιμές των βημάτων e και θ θεωρήθηκαν ίσες για κάθε παίκτη και συντελεστή. Αντίθετα υπήρξε μια μικρή ανισοκατανομή στις πιθανότητες των δράσεων, αλλά η ίδια για κάθε παίκτη:

$$P_i^{in} = P_i^{de} = 35\% \quad \text{και} \quad P_i^{st} = 30\%$$

Η περίοδοι εναλλαγών για την προσαρμογή των συντελεστών της καμπύλης προσφοράς του κάθε παίκτη επιλέχθηκαν τυχαία από το ίδιο εύρος τιμών (30-80 επαναλήψεων) και παρέμειναν οι ίδιες για όλη τη διάρκεια του παιγνίου. Το Ανώτατο Όριο Προσφοράς ορίσθηκε ίσο με 100, περίπου δεκαπλάσιο της αρχικής Οριακής Τιμής του Συστήματος, της τιμής δηλαδή που προκύπτει από την υποβολή ως προσφοράς του οριακού κόστους.

Κάθε παίγνιο αποτελείται από 32.000 συνεχόμενες προσφορές (επαναλήψεις) και η εμπειρία που αποκτάται από τις προσφορές αυτές χρησιμοποιείται μόνο στο τρέχον παίγνιο, καθώς κατά την διεξαγωγή του επόμενου παιγνίου θεωρείται ότι οι παίκτες έχουν μηδενική εμπειρία. Προσομοιώνονται δέκα διαφορετικοί τύποι παιγνίων, με 2, 4, 6, 8, 10, 12, 14, 16, 18 και 20 παίκτες αντίστοιχα. Σε όλους τους τύπους παιγνίων εφαρμόζονται πειραματικά οι τιμές 2%, 5%, 10%, 15% και 20% στο βήμα e . Αντιθέτως το βήμα θ θεωρείται ίσο με 5% για όλους τους τύπους παιγνίων και τις επαναλήψεις τους. Κάθε τύπος παιγνίου εκτελέσθηκε 100 φορές, οπότε ο συνολικός αριθμός παιγνίων που εκτελέσθηκαν είναι 5.000 και ο αντίστοιχος αριθμός επαναλήψεων (προσφορές) 160.000.000.

⁸ ουσιαστικά το οριακό τους κόστος $MC_i(x) = a_i + 2b_i x$

Παίκτης (#)	Σταθερό Κόστος	Παράμετροι Συνάρτησης Κόστους		Δυναμικότητα Παραγωγής (MW)		Βήματα Προσαρμογής Συντελεστών Προσφοράς	Βήμα	Περίοδος Προσαρμογής Συντελεστών Προσφοράς	
		<i>FC</i>	<i>a</i>	<i>b</i>	Ελάχιστο			Μέγιστο	e_A, e_B (%)
1	7,000	8.40	0.00020	7	15	2, 5, 10, 15, 20	5%	44	71
2	6,100	9.47	0.00024	7	15	2, 5, 10, 15, 20	5%	73	50
3	6,900	10.20	0.00021	7	15	2, 5, 10, 15, 20	5%	36	69
4	6,100	7.69	0.00018	7	15	2, 5, 10, 15, 20	5%	58	70
5	6,400	8.58	0.00016	7	15	2, 5, 10, 15, 20	5%	67	69
6	5,400	6.64	0.00020	7	15	2, 5, 10, 15, 20	5%	79	76
7	6,900	8.96	0.00017	7	15	2, 5, 10, 15, 20	5%	70	59
8	6,000	8.02	0.00025	7	15	2, 5, 10, 15, 20	5%	58	48
9	8,300	10.33	0.00016	7	15	2, 5, 10, 15, 20	5%	69	52
10	7,300	6.78	0.00015	7	15	2, 5, 10, 15, 20	5%	52	78
11	7,800	10.43	0.00016	7	15	2, 5, 10, 15, 20	5%	79	59
12	6,200	8.96	0.00022	7	15	2, 5, 10, 15, 20	5%	34	43
13	7,700	8.82	0.00024	7	15	2, 5, 10, 15, 20	5%	39	61
14	6,700	10.09	0.00017	7	15	2, 5, 10, 15, 20	5%	32	41
15	8,100	8.71	0.00024	7	15	2, 5, 10, 15, 20	5%	58	44
16	7,800	7.33	0.00023	7	15	2, 5, 10, 15, 20	5%	50	64
17	6,800	9.91	0.00019	7	15	2, 5, 10, 15, 20	5%	39	68
18	5,500	6.61	0.00020	7	15	2, 5, 10, 15, 20	5%	46	65
19	6,700	8.14	0.00023	7	15	2, 5, 10, 15, 20	5%	60	33
20	6,400	9.49	0.00023	7	15	2, 5, 10, 15, 20	5%	69	64

Σχήμα 39. Πίνακας με τα χαρακτηριστικά μεγέθη των Παραγωγών που συμμετέχουν στο παίγνιο

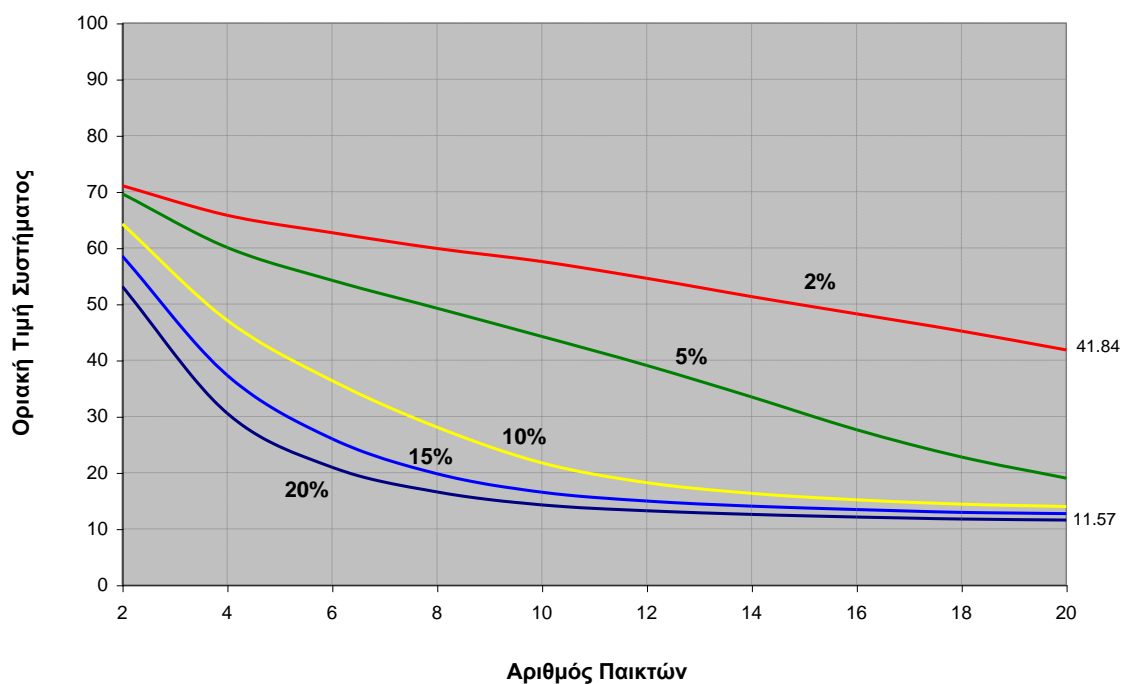
Στον τύπο παιγνίου με 2 παίκτες συμμετέχουν μόνο οι δύο πρώτοι παίκτες (#1 και #2) και κάθε φορά που προχωρούμε στον επόμενο τύπο παιγνίου (π.χ. 4 παίκτες), προσθέτουμε τους δύο επόμενους παίκτες (δηλ. #3 και #4). Κατά συνέπεια, οι δύο πρώτοι παίκτες συμμετείχαν σε όλα τα παίγνια που διεξήχθησαν ενώ οι δύο τελευταίοι (#19 και #20) μόνο στον τύπο παιγνίου με 20 παίκτες.

Αποτελέσματα

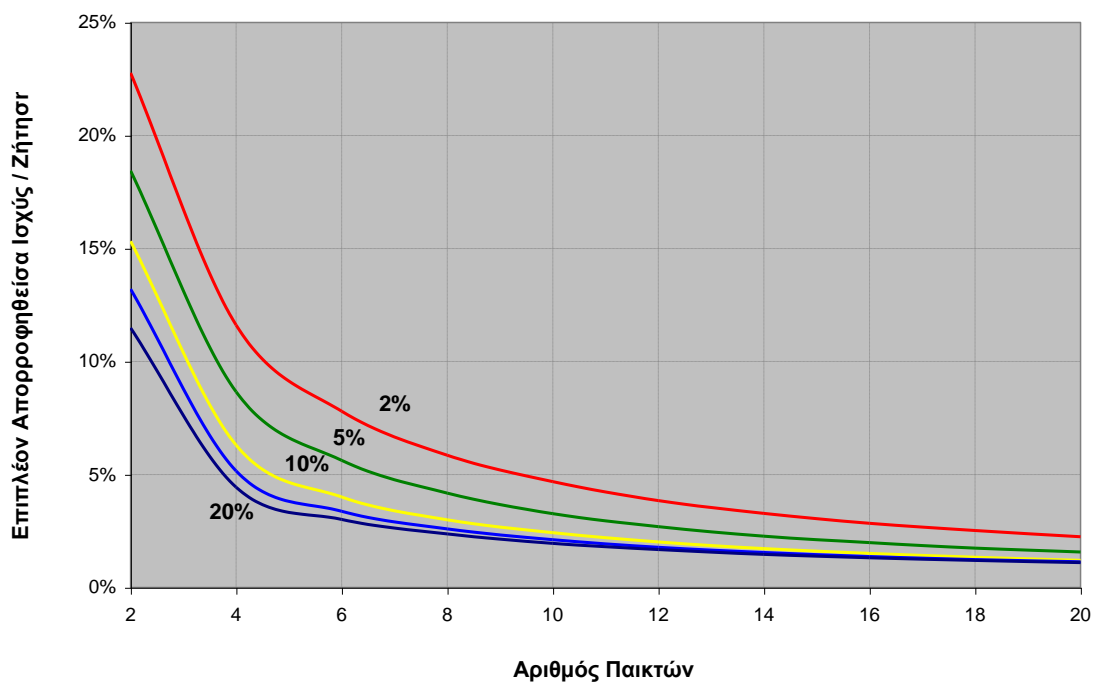
Παρά τις έντονες διακυμάνσεις που παρουσίασε η Οριακή Τιμή του Συστήματος (ΟΤΣ) κατά τη διάρκεια των παιγνίων, ο κινητός μέσος όρος της συγκλίνει και μάλιστα συγκλίνει σε υψηλότερη τιμή από την αρχική που αντιστοιχεί στις προσφορές του οριακού κόστους (Σχ. 40). Δύο είναι οι κυριότεροι παράγοντες που επηρεάζουν τη συμπεριφορά της:

- Ο αριθμός των παραγωγών που συμμετέχει στο παίγνιο επηρεάζει την τιμή σύγκλισης της ΟΤΣ καθώς και την ταχύτητα σύγκλισης. Συγκεκριμένα, η ΟΤΣ συγκλίνει σε υψηλότερα επίπεδα και με πιο αργό ρυθμό όσο λιγότεροι είναι οι παραγωγοί που απαρτίζουν την αγορά.
- Η προσαρμοστική ικανότητα των παικτών επηρεάζει έντονα την τιμή σύγκλισης της ΟΤΣ, με παρόμοιο τρόπο που την επηρεάζει ο αριθμός παραγωγών (δηλ. όσο υψηλότερη είναι η τιμή του βήματος e , τόσο πιο γρήγορα και σε χαμηλότερη τιμή συγκλίνει η ΟΤΣ). Ωστόσο, μεγαλύτερες τιμές της τιμής του e , προσδίδουν μεγαλύτερη μεταβλητότητα στην ΟΤΣ και κατά συνέπεια δημιουργούν μια πιο ασταθή αγορά.

Λόγω των ασυνεχειών στη συνολική καμπύλη προσφοράς ο Διαχειριστής του Συστήματος αναγκάζεται σε αρκετές περιπτώσεις να αγοράσει επιπλέον ισχύ από αυτή που αντιστοιχεί στη ζήτηση, όπως αναφέρθηκε σε προηγούμενη παράγραφο. Οι ποσότητες αυτές συναρτώνται σε μεγάλο βαθμό τόσο από τον αριθμό των παραγωγών που συμμετέχουν στην αγορά όσο και από την τιμή του e (Σχ.41).



Σχήμα 40. Σχέση της Οριακής Τιμής του Συστήματος σε σχέση με τον αριθμό των παραγωγών που συμμετέχουν στην αγορά για διαφορετικές τιμές του βήματος προσαρμογής ϵ .



Σχήμα 41. Σχέση της επιπλέον ισχύος που αναγκάζεται να αγοράσει ο Διαχειριστής του Συστήματος για να καλύψει τη ζήτηση, και του αριθμού των παραγωγών που συμμετέχουν στην αγορά για διαφορετικές τιμές του βήματος προσαρμογής ϵ .

6.1.3. Παιγνίο 3: Οριακή Τιμή Συστήματος για διαφορετικό Ανώτατο Όριο Προσφορών (Price Cap) και μεταβλητό βήμα προσαρμογής e

Χαρακτηριστικά του παιγνίου

Πρόκειται στην ουσία για το ίδιο παίγνιο με το προηγούμενο με τη διαφορά ότι οι εκτελέσεις γίνονται:

- και με τις δύο μεθόδους (Uniform Pricing και Pay-As-Bid)
- μόνο για τον τύπο παιγνίου με 6 παίκτες
- με δύο διαφορετικά Ανώτατα Όρια Προσφορών (Price Cap): 100 και 500.

Η πιο σημαντική όμως διαφορά είναι ότι τα παίγνια εκτελούνται με τρεις διαφορετικούς τρόπους ορισμού του βήματος e_A και e_B :

1. τα βήματα e_A και e_B είναι σταθερά κατά τη διάρκεια του παιγνίου και ίσα με 10%.
2. τα βήματα e_A και e_B είναι βαθμιαίως μειούμενα από το $\frac{1}{4}$ του αριθμού των συνολικών επαναλήψεων και μετά, με βάση το τύπο :

$$e_A = (\alpha / x) \quad \text{όπου } \alpha = 8.000 \text{ δηλ. το } \frac{1}{4} \text{ του } 32.000 \text{ (σύνολο επαναλήψεων).}$$

Η τελική τιμή του βήματος στο τέλος του παιγνίου είναι ίση με 2,5%. (το $\frac{1}{4}$ της αρχικής).

3. τα βήματα e_A και e_B βαθμιαίως μειώνονται από το $\frac{1}{10}$ του αριθμού των συνολικών επαναλήψεων και μετά, με βάση το τύπο :

$$e_A = (\alpha / x) \quad \text{όπου } \alpha = 3.200 \text{ δηλ. το } \frac{1}{10} \text{ του } 32.000 \text{ (σύνολο επαναλήψεων).}$$

Η τελική τιμή του βήματος στο τέλος του παιγνίου είναι ίση με 1%. (το $\frac{1}{10}$ της αρχικής).

Τιμή Ηλεκτρικής Ενέργειας

Στα Pay-As-Bid παίγνια καταγράφεται η μέση τιμή της ηλεκτρικής ενέργειας καθώς και η Οριακή Τιμή (Τιμή Ισορροπίας), ενώ στο Uniform Pricing οι δύο τιμές συμπίπτουν. Κατά κανόνα στα Pay-As-Bid παίγνια η τιμή της ηλεκτρικής ενέργειας⁹ είναι υψηλότερη. Όταν η τιμή του βήματος μειώνεται βαθμιαία κατά τη διάρκεια του παιγνίου, στο Pay-As-Bid παίγνιο η τιμή μειώνεται, ενώ στο Uniform Pricing παίγνιο αυξάνεται. Και στα δύο παίγνια η διακύμανση της τιμής μειώνεται παράλληλα με τη μείωση του βήματος (Σχ.42 και 43). Η εξήγηση φαίνεται να είναι πως όταν έχουμε σταθερό βήμα καθ'όλη τη διάρκεια του παιχνιδιού οι παίκτες δεν «μαθαίνουν» αλλά στην ουσία κυμαίνονται στο επιτρεπτό εύρος που ορίζεται από το μηδέν και το Ανώτατο Όριο Προσφορών (Price Cap), με αποτέλεσμα η τιμή να παρουσιάζει μια μεγάλη διακύμανση και η τιμή σύγκλισης να μην είναι τίποτε περισσότερο από τη μέση τιμή αυτής της διακύμανσης (φαινομενική σύγκλιση). Όταν το βήμα μειώνεται βαθμιαία τότε διαφαίνεται η πραγματική τάση του παιχνιδιού, που στο Uniform Pricing είναι να ανεβαίνει συνεχώς μέχρι του σημείου εκείνου που δεν μπορεί πλέον να ανέλθει (λόγω του Price Cap) ενώ στο Pay-As-Bid η τιμή τείνει (μετά από μια περίοδο αναζήτησης) προς το οριακό κόστος των παραγωγών.

Μερίδια Αγοράς και Επιπλέον Ισχύς

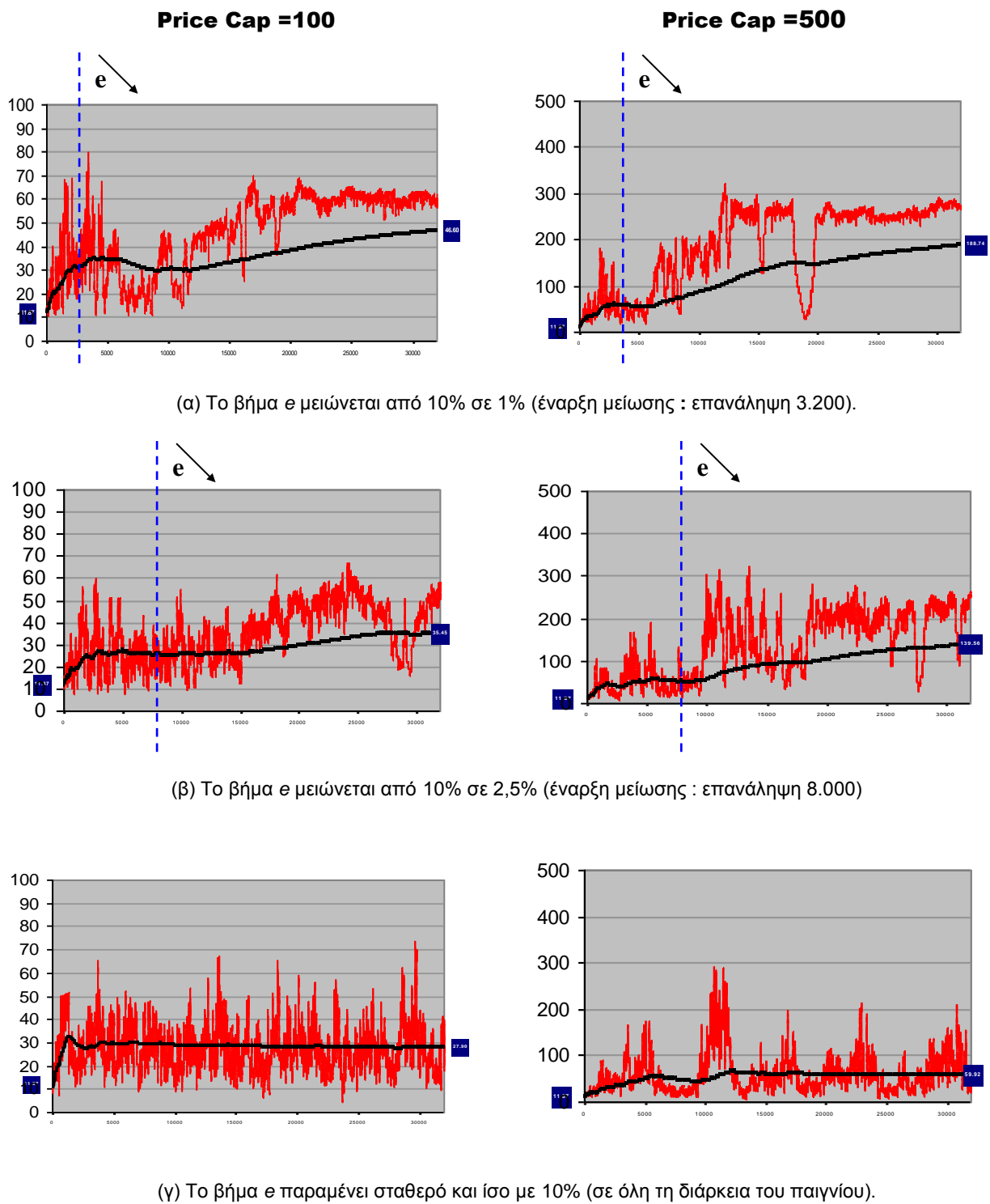
Μια άλλη ενδιαφέρουσα παρατήρηση είναι πως στα Pay-As-Bid παίγνια τα μερίδια αγοράς σχεδόν ισοκατανέμονται ενώ στο Uniform Pricing παρατηρούνται έντονες ανισοροπίες (Σχ.44). Η ανισοκατανομή των μεριδίων εντείνεται στο Uniform Pricing όταν το βήμα μειώνεται βαθμιαία, ενώ στο Pay-As-Bid η επίδραση είναι μικρή. Στο Pay-As-Bid οι επιπλέον ποσότητες είναι της τάξης του 1,5% της ζήτησης ενώ στο Uniform Pricing κυμαίνεται από 2,5 έως 2,9%.

Συντελεστές της προσφοράς $A+Bx$

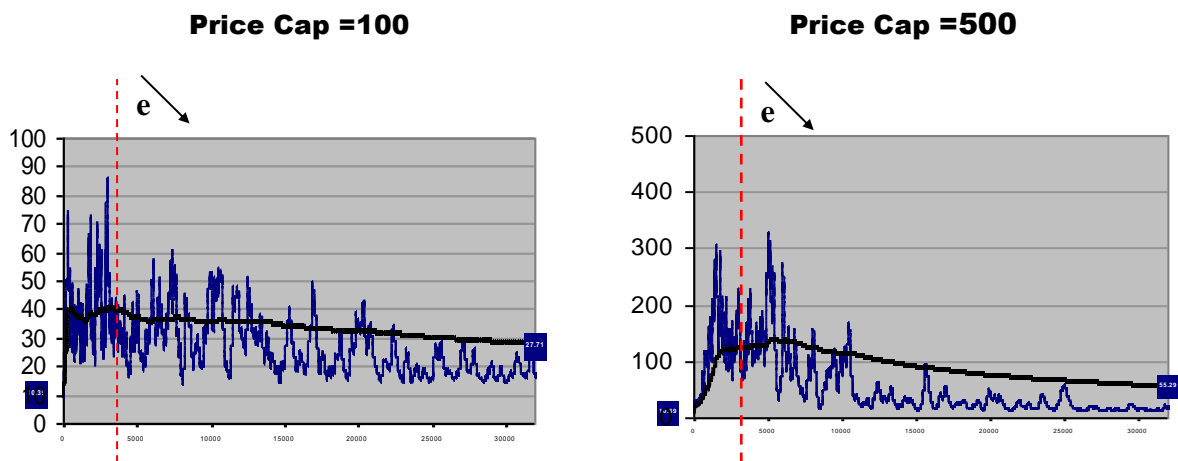
Ο συντελεστής A στο Pay-As-Bid συγκλίνει ανεξαρτήτως συμπεριφοράς του βήματος σε τιμές 4πλάσιες της αρχικής. Στο Uniform Pricing ο συντελεστής A φαίνεται να μειώνεται προς το τέλος του παιχνιδιού και να συγκλίνει σε τιμές από διπλάσιες ως και ίδιες με αυτές της αρχικής προσφοράς (οριακό κόστος παραγωγού).

⁹ Σαν κόστος για τον Διαχειριστή του Συστήματος

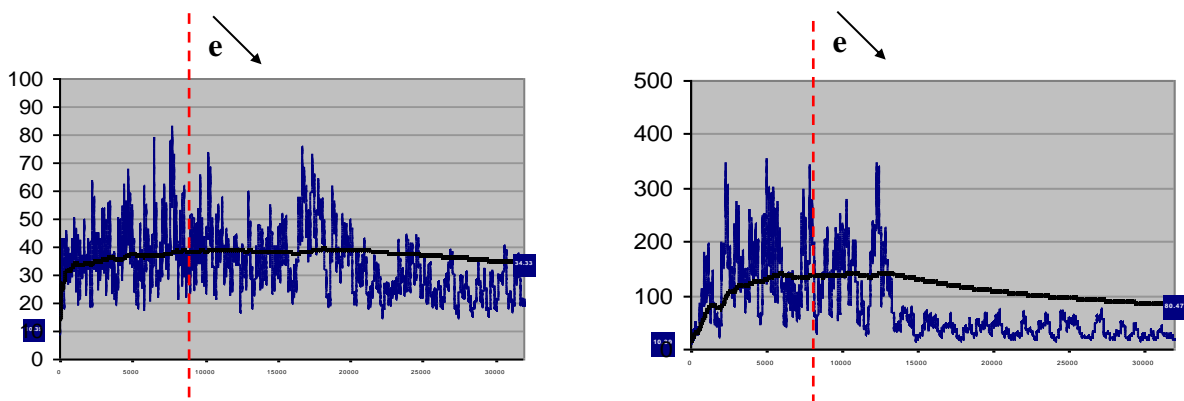
Ο Συντελεστής Β στο Pay-As-Bid με σταθερό βήμα συγκλίνει με τον ίδιο τρόπο και στα ίδια επίπεδα (λίγο υψηλότερα) με το Uniform Pricing. Με τη βαθμιαία μείωση του βήματος, στο Pay-As-Bid η σύγκλιση είναι πιο έντονη (μικρή διαφοροποίηση μεταξύ των παικτών) και σε χαμηλότερα επίπεδα ενώ στο Uniform Pricing συμβαίνει ακριβώς το αντίθετο. Αυτό τελικά αντανακλάται και στη τελική διαμόρφωση της τιμής.



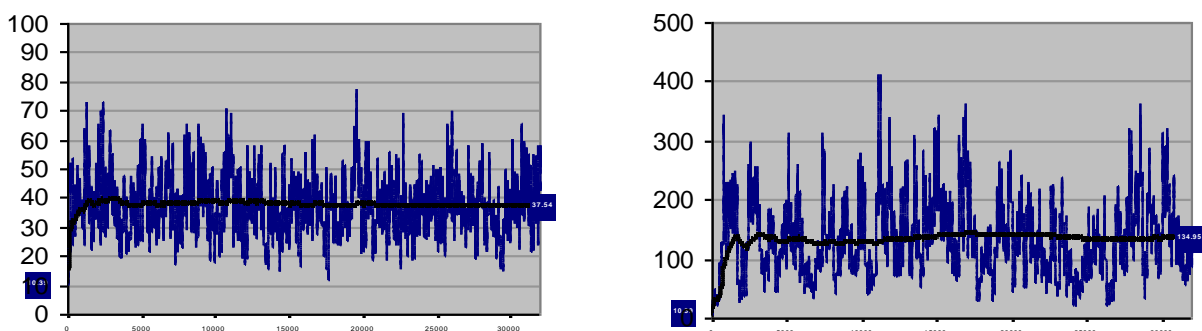
Σχήμα 42. Διαγράμματα ΟΤΣ σε Uniform Pricing παίγνια με διαφορετικό Price Cap και διαφορετική συμπεριφορά του βήματος προσαρμογής e .



(α) Το βήμα e μειώνεται από 10% σε 1% (έναρξη μείωσης: επανάληψη 3.200).

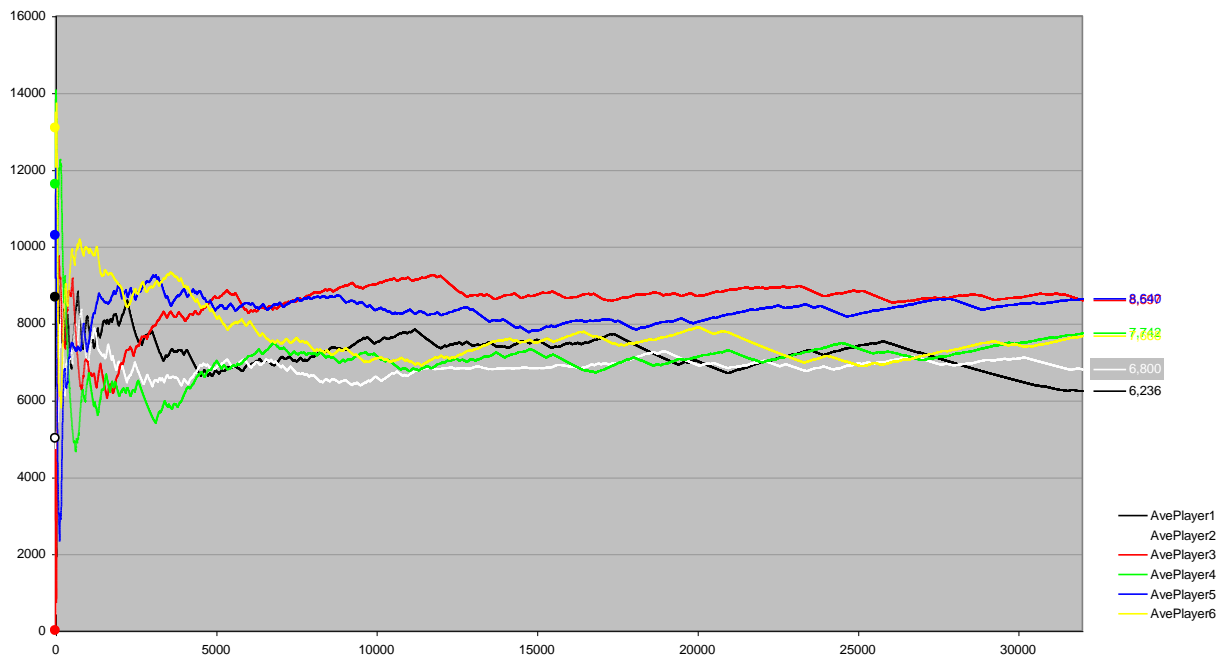


(β) Το βήμα e μειώνεται από 10% σε 2,5% (έναρξη μείωσης: επανάληψη 8.000)

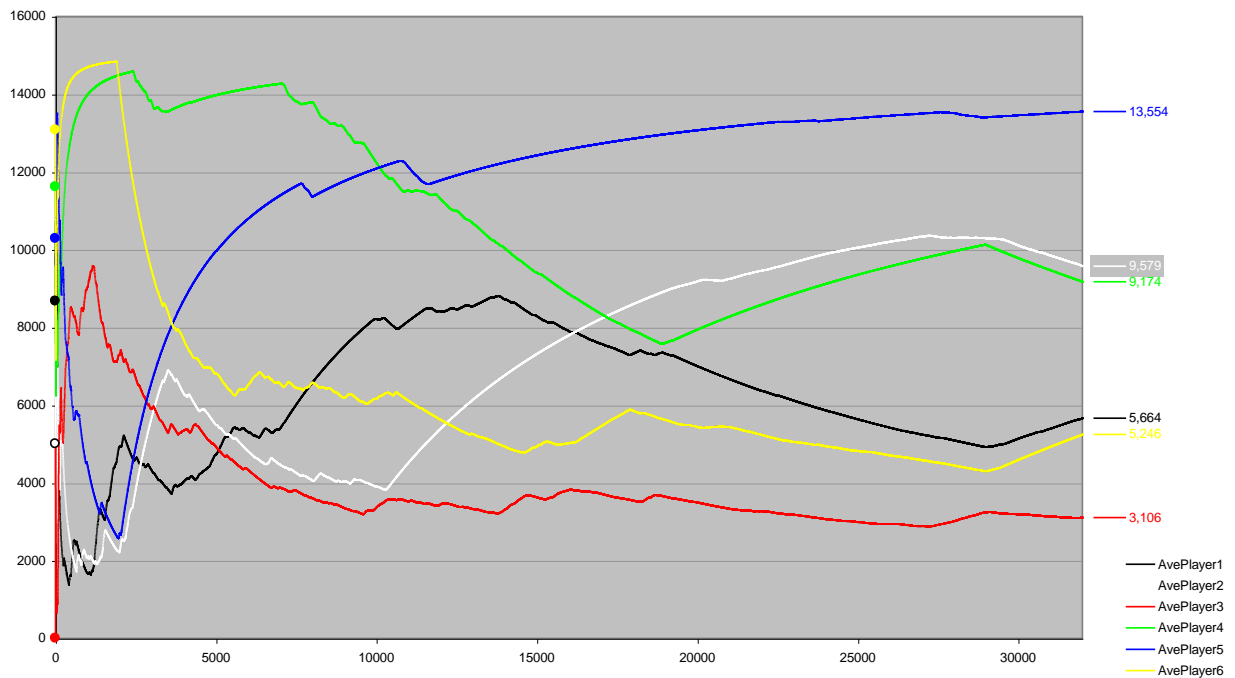


(γ) Το βήμα e παραμένει σταθερό και ίσο με 10% (σε όλη τη διάρκεια του παιγνίου).

Σχήμα 43. Διαγράμματα Τιμής Ηλεκτρικής Ενέργειας σε Pay-As-Bid παίγνια με διαφορετικό Price Cap και διαφορετική συμπεριφορά του βήματος προσαρμογής e .



(α) Pay-as-Bid



(β) Uniform Pricing

Σχήμα 44. Διαγράμματα μέσω μεριδίων αγοράς των παικτών σε τύπο παιχνιδιού έξι παικτών και με τις δύο μεθόδους τιμολόγησης και μειούμενο βήμα προσαρμογής ϵ .

6.1.4. Παιγνίο 4: Σύγκριση Uniform Pricing και Pay-As-Bid τιμολόγησης

Μια άλλη ενδιαφέρουσα μελέτη και σύγκριση που διενεργήθηκε ήταν αυτή που αφορά στη συμπεριφορά ενός συστήματος με τους ίδιους παραγωγούς και τα ίδια χαρακτηριστικά υπό διαφορετική μέθοδο τιμολόγησης. Η μέθοδος τιμολόγησης επηρεάζει καθοριστικά όπως φαίνεται και στα επόμενα διαγράμματα όλα τα σχετικά μεγέθη, ποσοτικά και ποιοτικά. Βασιζόμενοι στο μοντέλο της αγοράς στα παίγνια που αναφέρθηκαν ανωτέρω, χρησιμοποιήθηκε μια πιο περιορισμένη έκδοση του παιγνίου με συγκεκριμένα σταθερά χαρακτηριστικά με σκοπό να δοκιμασθεί το μοντέλο και να εξαχθούν κάποια γενικά συμπεράσματα σχετικά με τη διαφοροποίηση της συμπεριφορά της αγοράς όταν το καθεστώς τιμολόγησης αλλάζει. Για το λόγο αυτό κάποιες παράμετροι του παιγνίου θεωρήθηκαν σταθεροί ενώ παράλληλα οι διαφορές μεταξύ των παικτών περιορίστηκαν στο ελάχιστο δυνατό [57].

Συγκεκριμένα, θεωρήσαμε ότι η ζήτηση D παραμένει σταθερή κατά τη διάρκεια του παιγνίου και σε όλα τα παίγνια που διενεργήθηκαν, και είναι πάντοτε ίση με το ήμισυ της αθροίσματος της εγκατεστημένης ισχύος των παραγωγών που συμμετέχουν κάθε φορά στο παίγνιο. Οι παραγωγοί βρίσκονται πάντα μέσα στο ίδιο εύρος δυναμικότητας παραγωγής (5 MW - 15 MW) και χρησιμοποιούν όλοι την ίδια τεχνολογία παραγωγής (π.χ. μικρά πετρελαϊκά θερμικά εργοστάσια παραγωγής). Οι παραγωγοί είχαν το ίδιο τεχνικό ελάχιστο και εύρος δυναμικότητας παραγωγής (5 MW - 15 MW), χρησιμοποιούν δηλαδή όλοι το ίδιο καύσιμο και την ίδια τεχνολογία παραγωγής (π.χ. μικροί πετρελαϊκοί θερμοηλεκτρικοί σταθμοί). Οι τιμές των παραμέτρων κόστους και του σταθερού κόστους έχουν τυχαία διάταξη εντός ενός εύρους $\pm 25\%$ γύρω από τις τιμές του πρώτου παίκτη-παραγωγού (κεντρικές τιμές) οι οποίες είναι οι ακόλουθες:

$$FC_1 = 7,000 \quad a_1 = 8.40 \quad b_1 = 0.00020$$

Όλοι οι παίκτες θεωρούνται ισοδύναμοι σε σχέση με την προσαρμοστική και μαθησιακή τους ικανότητα, και για το λόγο αυτό οι τιμές των βημάτων e και θ θεωρήθηκαν ίσες για κάθε παίκτη και συντελεστή.

Οι αρχικές τιμές των P^{in} , P^{de} , P^{st} που αντιστοιχούν στην πιθανότητα επιλογής της κάθε δράσης (αύξηση, μείωση, σταθεροποίηση) και διαμορφώνουν το στοχαστικό προφίλ λαμβάνονται επίσης ίσες για κάθε παίκτη και κάθε συντελεστή, ενώ διαφέρουν μόνο ανά είδος δράσης:

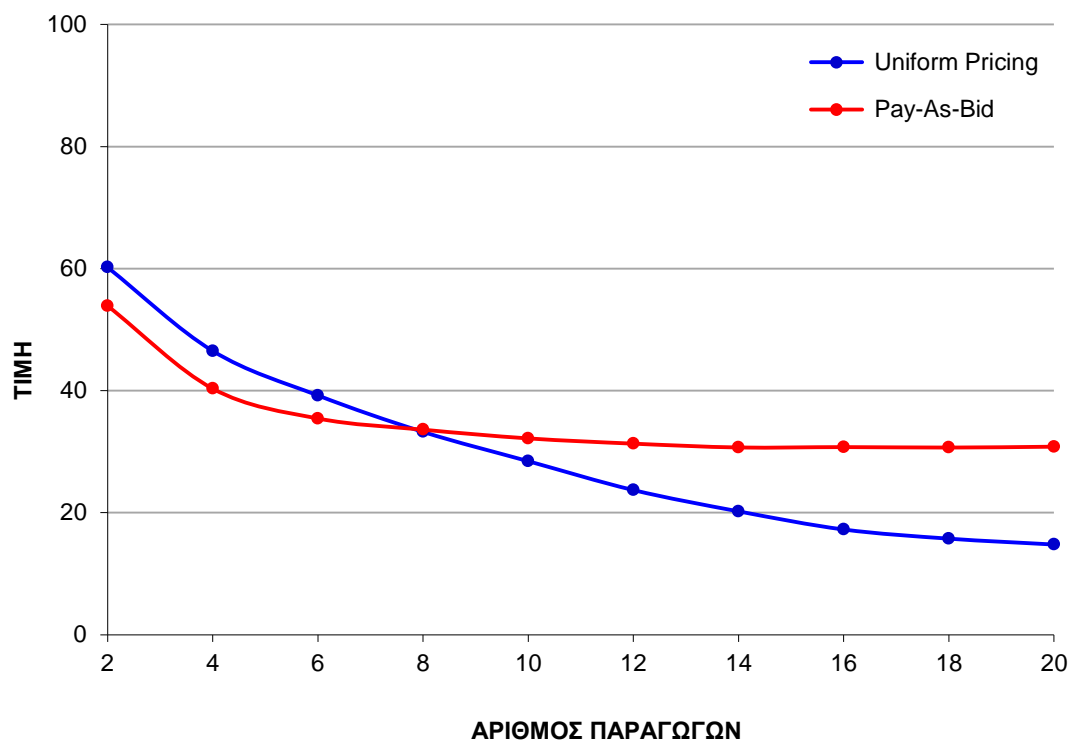
$$P_i^{in} = P_i^{de} = 35\% \quad \text{και} \quad P_i^{st} = 30\%$$

Οι περίοδοι εναλλαγών για την προσαρμογή των συντελεστών της καμπύλης προσφοράς του κάθε παίκτη επιλέχθηκαν τυχαία από το ίδιο εύρος τιμών (30-80 επαναλήψεων) και παρέμειναν οι ίδιες για όλη τη διάρκεια του παιγνίου. Το Ανώτατο Όριο Προσφοράς ορίστηκε ίσο με 100, περίπου δεκαπλάσιο της αρχικής Οριακής Τιμής του Συστήματος, της τιμής δηλαδή που προκύπτει από την υποβολή ως προσφοράς του οριακού κόστους. Κάθε παίγνιο αποτελείται από 32.000 συνεχόμενες προσφορές (επαναλήψεις) και η εμπειρία που αποκτάται από τις προσφορές αυτές χρησιμοποιείται μόνο στο τρέχον παίγνιο, καθώς κατά την διεξαγωγή του επόμενου παιγνίου θεωρείται ότι οι παίκτες έχουν μηδενική εμπειρία. Προσομοιώνονται δέκα διαφορετικοί τύποι παιγνίων, με 2, 4, 6, 8, 10, 12, 14, 16, 18 και 20 παίκτες αντίστοιχα. Σε όλους τους τύπους παιγνίων και τις επαναλήψεις τους εφαρμόζεται η τιμή 5% στο βήμα e και στο βήμα θ . Κάθε τύπος παιγνίου εκτελέστηκε 100 φορές, οπότε ο συνολικός αριθμός παιγνίων που εκτελέστηκαν είναι 2.000 και ο αντίστοιχος αριθμός επαναλήψεων (γύρω προσφορών) 64.000.000. Στον τύπο παιγνίου με 2 παίκτες συμμετέχουν μόνο οι δύο πρώτοι παίκτες (#1 και #2) και κάθε φορά που προχωρούμε στον επόμενο τύπο παιγνίου (π.χ. 4 παίκτες), προσθέτουμε τους δύο επόμενους παίκτες (δηλ. #3 και #4). Κατά συνέπεια, οι δύο πρώτοι παίκτες συμμετείχαν σε όλα τα παίγνια που διεξήχθησαν ενώ οι δύο τελευταίοι (#19 και #20) μόνο στον τύπο παιγνίου με 20 παίκτες.

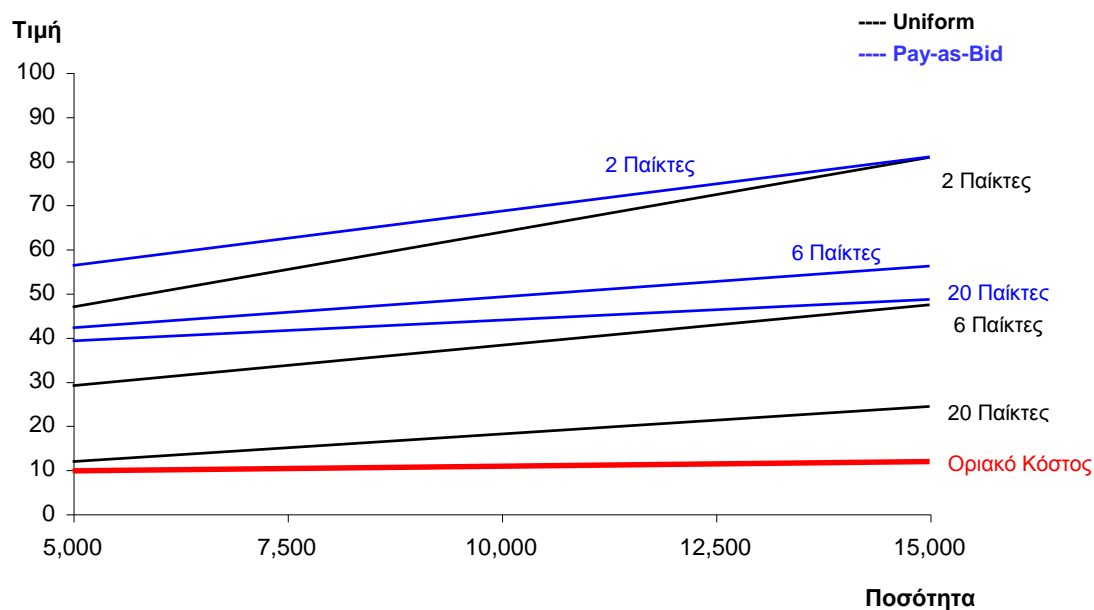
Για κάθε παίγνιο και επανάληψη καταγράφονται οι τιμές και ο κινητός Μέσος Όρος για τα ακόλουθα:

1. Τιμή Ηλεκτρικής Ενέργειας
2. Μεριδία Αγοράς x_i και καθαρά έσοδα για κάθε παραγωγό
3. Τιμές των Συντελεστών Προσφοράς A_i , B_i

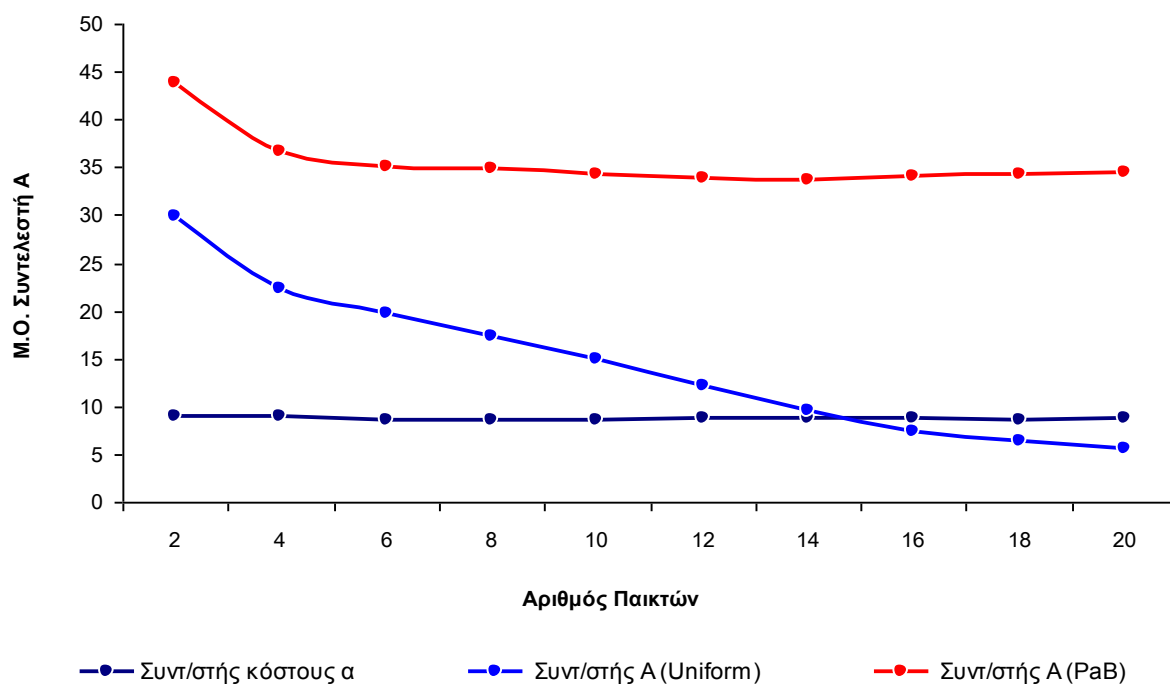
Τα αποτελέσματα του πρώτου γύρου του κάθε παιχνιδιού, όπου οι προσφορές ταυτίζονται με το οριακό κόστος αποτελούν τις τιμές αναφοράς για όλες τις απαραίτητες συγκρίσεις.



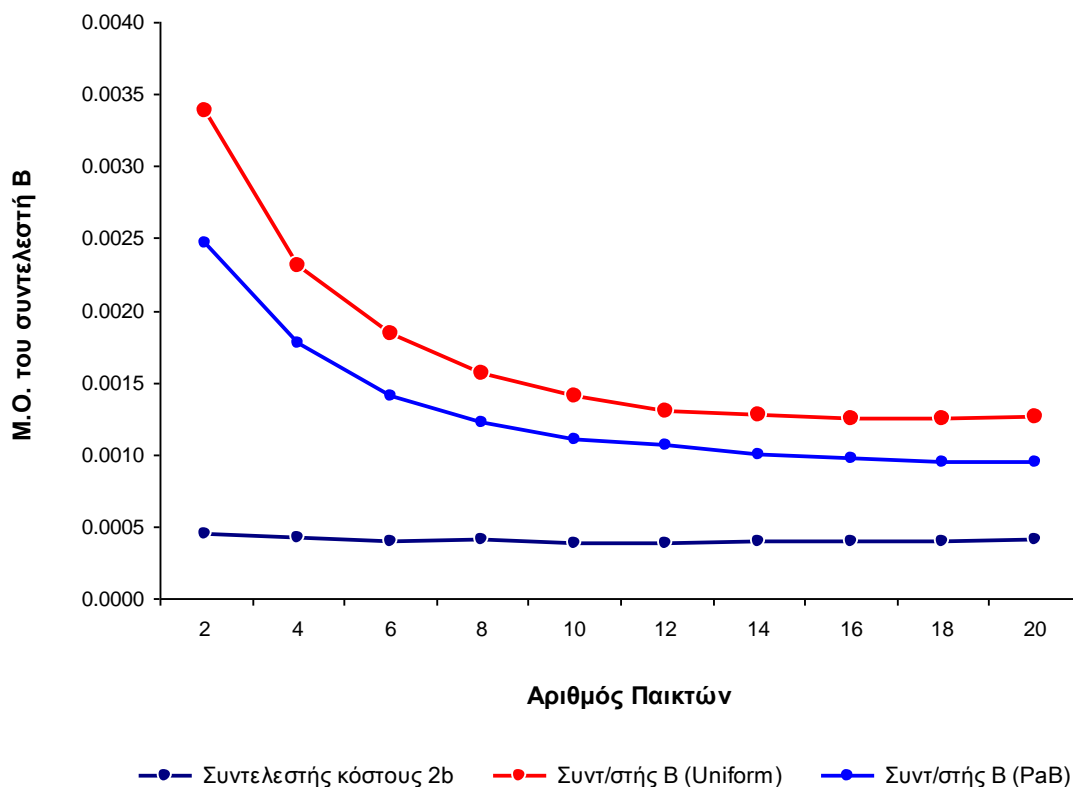
Σχήμα 45. Διάγραμμα σύγκρισης της τιμής της ηλεκτρικής ενέργειας συναρτήσει του αριθμού των παραγωγών σε Συστήματα με διαφορετική μέθοδο τιμολόγησης (Uniform Pricing και Pay-As-Bid).



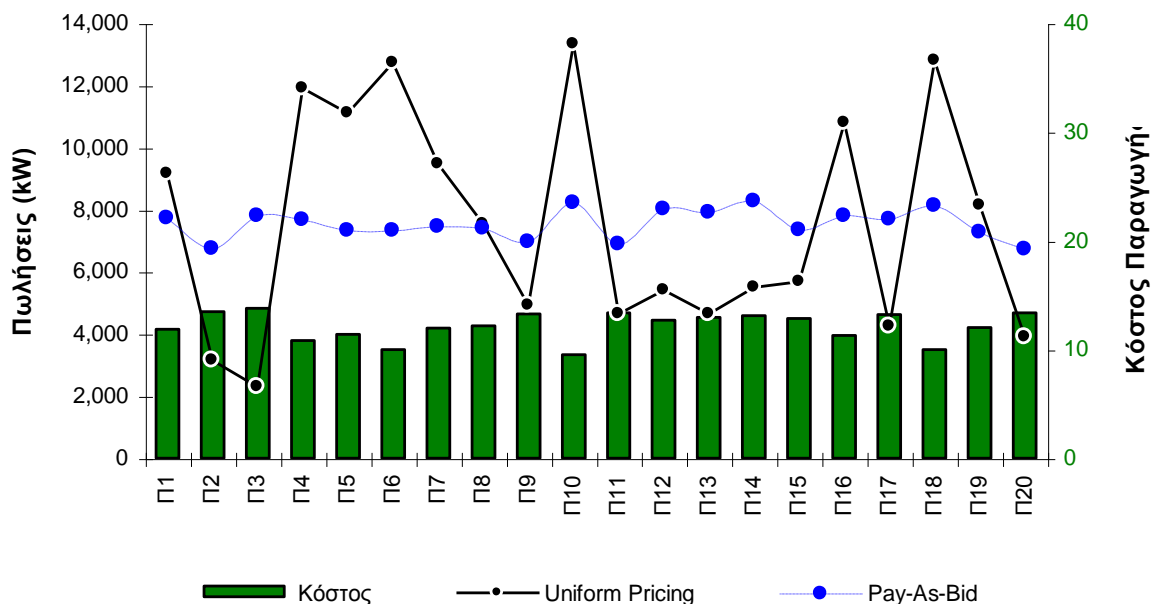
Σχήμα 46. Διάγραμμα σύγκρισης του μέσου προφίλ προσφορών σε Συστήματα με 2, 6 και 20 παραγωγούς και με διαφορετική μέθοδο τιμολόγησης: Uniform Pricing και Pay-As-Bid.



Σχήμα 47. Διάγραμμα σύγκρισης της μέσης τιμής του σταθερού συντελεστή Α της προσφοράς των παραγωγών σε Συστήματα με διαφορετική μέθοδο τιμολόγησης και σε συνάρτηση με τον αριθμό των συμμετεχόντων παραγωγών.



Σχήμα 48. Διάγραμμα σύγκρισης της μέσης τιμής του γραμμικού συντελεστή Β της προσφοράς των παραγωγών σε Συστήματα με διαφορετική μέθοδο τιμολόγησης και σε συνάρτηση με τον αριθμό των συμμετεχόντων παραγωγών.



Σχήμα 49. Μέσος όρος πωλήσεων για κάθε παίκτη σε παίγνιο 20 παραγωγών και σε Συστήματα με διαφορετική μέθοδο τιμολόγησης.

Η αύξηση του ανταγωνισμού φαίνεται να προκαλεί πτωτικές τάσεις για την τιμή της ηλεκτρικής ενέργειας και στα δύο συστήματα τιμολόγησης αλλά με διαφορετικό τρόπο. Στο Uniform Pricing το φαινόμενο είναι πολύ πιο έντονο καθώς η τιμή μειώνεται στο ένα τρίτο καθώς η αγορά από ολιγοπώλιο τείνει στον πλήρη ανταγωνισμό. Σε ολιγοπωλιακή αγορά το σύστημα Pay-As-Bid είναι πιο αποτελεσματικό στη συγκράτηση των τιμών αλλά στη συνέχεια η τιμή δεν μειώνεται με τον ίδιο ραγδαίο ρυθμό όπως στο Uniform Pricing καθώς η αγορά τείνει να γίνει ανταγωνιστική, και μάλιστα η τιμή σταθεροποιείται από ένα σημείο και έπειτα ανεξάρτητα από το πόσοι νέοι παραγωγοί εισέρχονται στην αγορά (Σχήμα 45).

Η μορφή των προσφορών των παραγωγών (Σχήμα 46, 47 και 48) φαίνεται να βρίσκεται πίσω από αυτή τη διαφοροποίηση στη συμπεριφορά της αγοράς στα δύο συστήματα τιμολόγησης. Στο σύστημα Uniform Pricing οι παραγωγοί τείνουν σε πιο ανταγωνιστικές και επιθετικές προσφορές που χαρακτηρίζονται από χαμηλές αρχικές τιμές (σταθερός συντελεστής A) και αυξημένη κλίση (γραμμικός συντελεστής B), ενώ στο άλλο σύστημα οι καμπύλες προσφορών παρουσιάζονται σαφώς πιο επίπεδες, γεγονός που από κάποιο σημείο και πέρα δεν εξυπηρετούν τον ανταγωνισμό και την αποτελεσματικότητα της αγοράς.

Επιπρόσθετα και σαν αποτέλεσμα αυτής της διαφοροποίησης στη συμπεριφορά των παραγωγών στα δύο συστήματα τιμολόγησης, η κατανομή των πωλήσεων μεταξύ των παραγωγών του ίδιου συστήματος διαφέρει και μάλιστα έντονα. Η σχεδόν ομοιόμορφη κατανομή της πωλούμενης παραγωγής από τους παραγωγούς στο σύστημα Pay-As-Bid έρχεται σε αντίθεση με τις έντονες ανισότητες ως προς τον όγκο των πωλήσεων που επιτυγχάνουν οι παραγωγοί στο σύστημα Uniform Pricing, οι οποίες αντανακλούν, και μάλιστα μεγεθυντικά, τις όποιες διαφορές εμφανίζουν στο κόστος παραγωγής τους (Σχήμα 49).

6.2. Παιγνία σε δύο Διασυνδεδεμένα Ηλεκτρικά Συστήματα

Η διεξαγωγή συγκρίσεων βάσει αποτελεσμάτων παιγνίων που διενεργήθηκαν σε δύο ηλεκτρικά συστήματα αναπόφευκτα χρησιμοποιεί τα όποια αποτελέσματα και συμπεράσματα είχαν εξαχθεί από τα αντίστοιχα παίγνια σε ένα ηλεκτρικό σύστημα. Η πολυπλοκότητα όμως αυξάνει αν θελήσει κανείς να επεκτείνει τις συγκρίσεις σε όλο το φάσμα των διαθέσιμων στοιχείων και μοιραία σε κάποια από αυτά οι συγκρίσεις και τα όποια συμπεράσματα είναι επισφαλής. Υπάρχουν ωστόσο κάποιες καίριες περιοχές ενδιαφέροντος όπως η σύγκριση της τιμής στα δύο συστήματα για όλους τους πιθανούς συνδυασμούς μεθόδων τιμολόγησης που εφαρμόζονται σε αυτά, που οι συγκρίσεις είναι εφικτές αν ληφθούν υπόψη μόνο κάποιοι περιορισμοί συγκρισιμότητας.

Στο πλαίσιο της παρούσης διατριβής για λόγους που είχαν να κάνουν με τη δοκιμασία των επιμέρους μοντέλων που αναπτύχθηκαν έγιναν τέτοιες συγκρίσεις που αφορούν στην τιμή της ηλεκτρικής ενέργειας σε συστήματα που έχουν ίδια ή διαφορετική μέθοδο τιμολόγησης για όλο το φάσμα αριθμού συμμετεχόντων σε αυτά, με τον περιορισμό όμως ότι τα συστήματα είχαν ίδιο λόγο ζήτησης προς εγκατεστημένη ισχύ και οι παραγωγοί που ήταν εγκατεστημένοι σε αυτά ήταν ίδιοι ως προς τον αριθμό και ως προς τα τεχνικά και κοστολογικά τους χαρακτηριστικά. Επιπλέον τέθηκε και περιορισμός στη χωρητικότητα της διασύνδεσης προκειμένου να συγκριθούν με αντίστοιχα αποτελέσματα που εξήχθησαν χωρίς την ύπαρξη τέτοιου περιορισμού.

Συγκεκριμένα, διεξήχθησαν παίγνια σε δύο συστήματα που έχουν και τα δύο το ίδιο σύστημα τιμολόγησης (και για τις δύο μεθόδους), καθώς και διαφορετική μέθοδο τιμολόγησης το καθένα. Καθώς οι παίκτες σε μια τέτοια αγορά ηλεκτρικής ενέργειας έχουν το δικαίωμα να προσφέρουν και στα δύο συστήματα τα παίγνια διεξήχθησαν αρχίζοντας από 4 παραγωγούς στο κάθε σύστημα και φτάνοντας ως τους 20. Χρησιμοποιήθηκαν και οι δύο τρόποι λειτουργίας της διασυνδεδεμένης αγοράς, δηλαδή οι παραγωγοί να μπορούν να διασπούν τη προσφερόμενη δυναμικότητα και στα δύο συστήματα ή να προσφέρουν μόνο σε ένα από τα δύο συστήματα κάθε φορά το σύνολο της δυναμικότητάς τους. Επίσης χρησιμοποιήθηκαν και οι δύο τρόποι επιλογής συστήματος και βελτίωσης των προσφορών, δηλαδή ο βραχυπρόθεσμος που λαμβάνει υπόψη μόνο την τελευταία εικόνα αλλά και ο μακροπρόθεσμος που συγκλίνει με τη βοήθεια προσαρμοστικού μαθησιακού αλγορίθμου. Τέλος, εκτελέστηκαν παίγνια με και χωρίς περιορισμό στη δυναμικότητα διασύνδεσης. Τα παίγνια με τον περιορισμό αφορούσαν σε δυναμικότητα διασύνδεση ίση με το 30% της εγκατεστημένης ισχύος του ενός συστήματος.

Στη σειρά αυτή των παιγνίων μελετήθηκε η συμπεριφορά δύο διασυνδεδεμένων συστημάτων για όλους τους συνδυασμούς μεθόδων τιμολόγησης και για διαφορετικό πλήθος παραγωγών κάθε φορά. Συγκεκριμένα, μελετήθηκαν οι περιπτώσεις που οι Διαχειριστές των δύο συστημάτων χρησιμοποιούν την ίδια μέθοδο τιμολόγησης - είτε Uniform Pricing (*Uhl*) είτε Pay-As-Bid (*PaB*) - καθώς και η περίπτωση που εφαρμόζουν διαφορετική μέθοδο τιμολόγησης. Για λόγους άμεσης συγκρισιμότητας θεωρήσαμε ότι στα συγκεκριμένα παίγνια τα δύο συστήματα έχουν πάντα τον ίδιο αριθμό παραγωγών τα χαρακτηριστικά των οποίων (τεχνολογία παραγωγής, κόστος, δυναμικότητα, προσαρμοστικότητα) είναι παραπλήσια και ότι η ζήτηση του ενός συστήματος είναι ίση με τη ζήτηση του άλλου και επιπλέον ανελαστική. Για όλους τους συνδυασμούς ανωτέρω διενεργήθηκαν παίγνια με 4, 6, 8, 10, 12, 14, 16, 18 και 20 παραγωγούς στο κάθε σύστημα.

Για την εκτίμηση των αποτελεσμάτων των παιγνίων σε δύο ηλεκτρικά συστήματα διενεργήθηκαν αντίστοιχα παίγνια και σε ένα μόνο ηλεκτρικό σύστημα υπό τις ίδιες συνθήκες και χρησιμοποιήθηκαν τα αποτελέσματα που προέκυψαν από αυτά. Ωστόσο η σύγκριση των παιγνίων σε ένα και σε δύο συστήματα, ιδιαίτερα για μεγέθη όπως η τιμή της ηλεκτρικής ενέργειας έχει περισσότερο ποιοτικό παρά ποσοτικό χαρακτήρα. Τέλος, αναφορικά με τη δυναμικότητα διασύνδεσης όλα τα παίγνια που αφορούσαν σε δύο συστήματα διεξήχθησαν α) με απεριόριστη δυναμικότητα διασύνδεσης και β) με δυναμικότητα διασύνδεσης ίση με το 30% της εγκατεστημένης ισχύος του ενός συστήματος. Το κάθε παίγνιο αποτελείται από 500.000 επαναλήψεις υποβολής προσφορών-εκκαθάρισης και

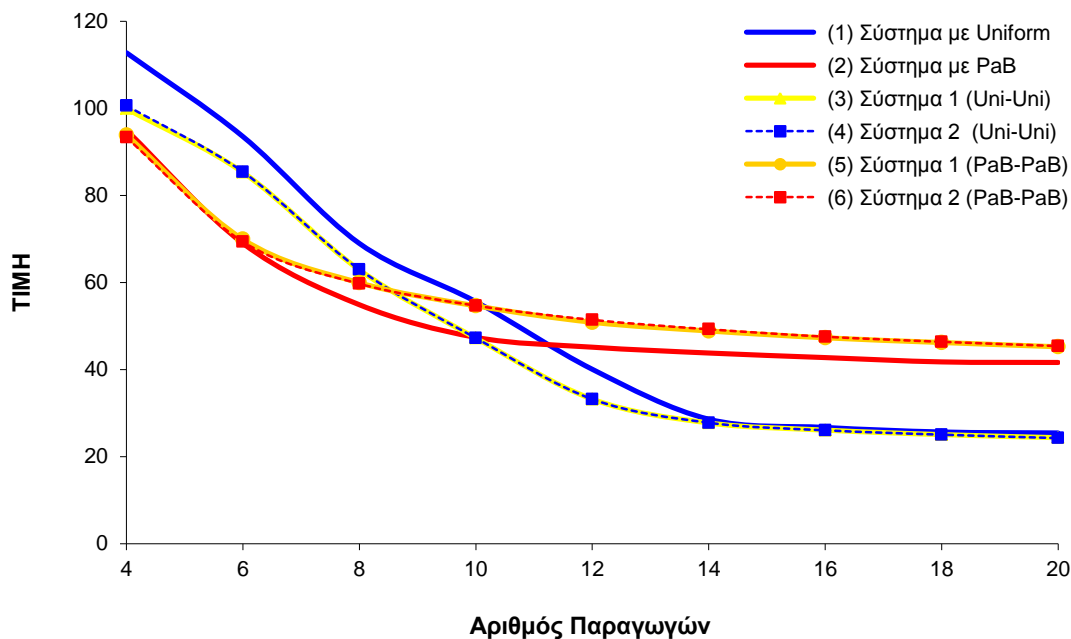
διενεργείται επαναλαμβανόμενα 100 φορές, ενώ ως αποτέλεσμα του παιγνίου λαμβάνεται η μέση τιμή των αποτελεσμάτων που προκύπτουν κατά τις 100 επαναλήψεις του. Ως εκ τούτου έγιναν συνολικά 3,6 δις εκκαθαρίσεις της αγοράς σε 7.200 επαναλήψεις 72 διαφορετικών παιγνίων. Στο Σχήμα 50 παρουσιάζεται πίνακας με τα συνοπτικά αποτελέσματα των παιγνίων που αφορούν στην τιμή σύγκλισης της αγοράς καθώς και οι τιμές της αγοράς που θα προέκυπταν αν οι παραγωγοί είχαν υποβάλλει ως προσφορά το οριακό τους κόστος.

Παραγωγοί	1 ΣΥΣΤΗΜΑ		2 ΣΥΣΤΗΜΑΤΑ (χωρίς περιορισμό διασύνδεσης)					
	Uni	PaB	Uni-Uni		PaB-PaB		Uni-PaB	
			Σ1	Σ2	Σ1	Σ2	Σ1	Σ2
4	112.8	95.0	99.9	100.7	93.9	93.3	101.2	96.1
6	93.5	69.0	85.3	85.4	69.9	69.4	78.8	76.5
8	69.0	54.9	62.9	63.0	59.9	59.7	58.0	59.1
10	55.6	47.4	47.2	47.2	54.6	54.7	50.2	50.6
12	40.0	45.1	33.2	33.2	50.8	51.4	42.4	44.0
14	28.6	43.8	27.7	27.8	48.9	49.2	38.8	40.7
16	26.7	42.7	26.1	26.0	47.3	47.5	36.9	38.8
18	25.6	41.7	25.0	25.0	46.2	46.4	35.4	37.7
20	25.4	41.6	24.3	24.2	45.2	45.4	33.8	36.4
Ορ. Κόστος	22.0	20.5	19.7	18.4	22.0	18.4	22.0	19.7

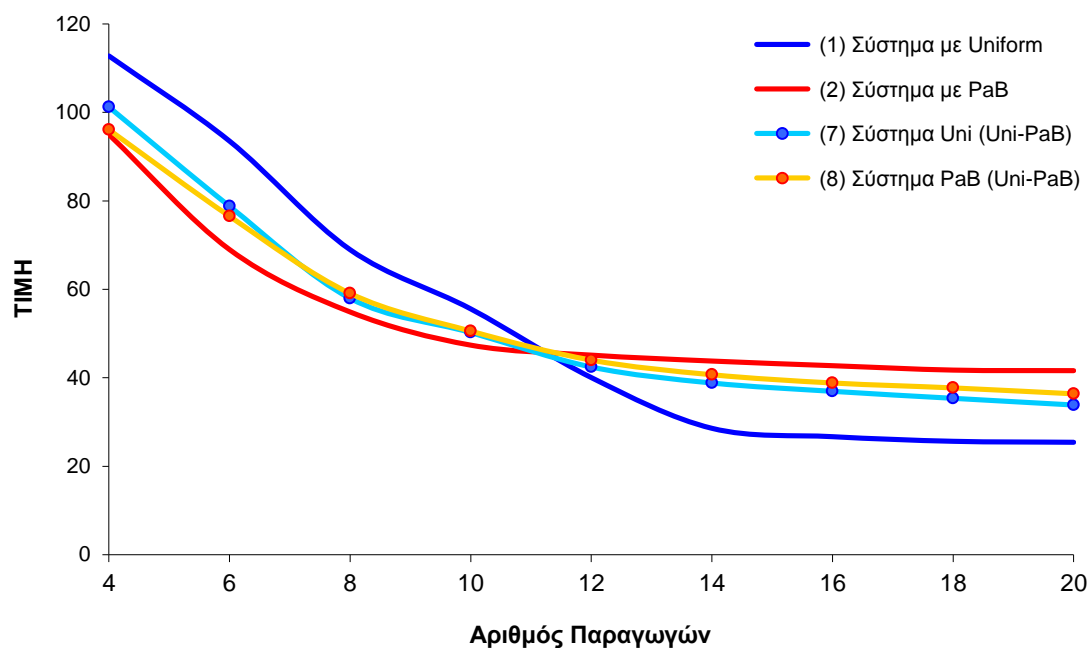
Σχήμα 50. Συγκριτικός πίνακας της τιμής της ηλεκτρικής ενέργειας σε παίγνια με διαφορετικό αριθμό παραγωγών σε ένα και σε δύο ηλεκτρικά συστήματα χωρίς περιορισμό στη δυναμικότητα διασύνδεσής.

Αυτό που προκύπτει καταρχήν για την περίπτωση της απεριόριστης δυναμικότητας διασύνδεσης και όταν τα δύο συστήματα εφαρμόζουν την ίδια μέθοδο τιμολόγησης (Σχήμα 51) είναι πως οι τιμές και των δύο συστημάτων συγκλίνουν πάντα στην ίδια τιμή (καμπύλες 3-4 και 5-6). Στην περίπτωση της τιμολόγησης Uniform Pricing η τιμή σε δύο διασυνδεδεμένα συστήματα (καμπύλες 3-4) σε σύγκριση με την τιμή που προκύπτει σε απομονωμένο σύστημα (καμπύλη 1) ξεκινά από χαμηλότερα επίπεδα και συγκλίνει τελικώς στο ίδιο επίπεδο καθώς η αγορά κινείται από το ολιγοπώλιο προς τον τέλειο ανταγωνισμό. Αντίθετα για την περίπτωση της τιμολόγησης Pay-As-Bid η τιμή ξεκινά από το ίδιο επίπεδο και συγκλίνει σε υψηλότερο (καμπύλες 2 και 5-6) καθώς η αγορά μεταβαίνει από το ολιγοπώλιο στον τέλειο ανταγωνισμό.

Στην περίπτωση των δύο συστημάτων με διαφορετική μέθοδο τιμολόγησης (Σχήμα 52) τα δύο συστήματα συγκλίνουν στο ίδιο επίπεδο τιμών για τον ίδιο αριθμό παραγωγών. Επιπλέον οι έντονες διαφορές των αποτελεσμάτων λόγω διαφορετικής μεθόδου τιμολόγησης που παρατηρούνται σε απομονωμένα συστήματα (καμπύλες 1-2) αμβλύνονται και τείνουν να συμπέσουν σε ενδιάμεσα επίπεδα τιμών (καμπύλες 7-8).

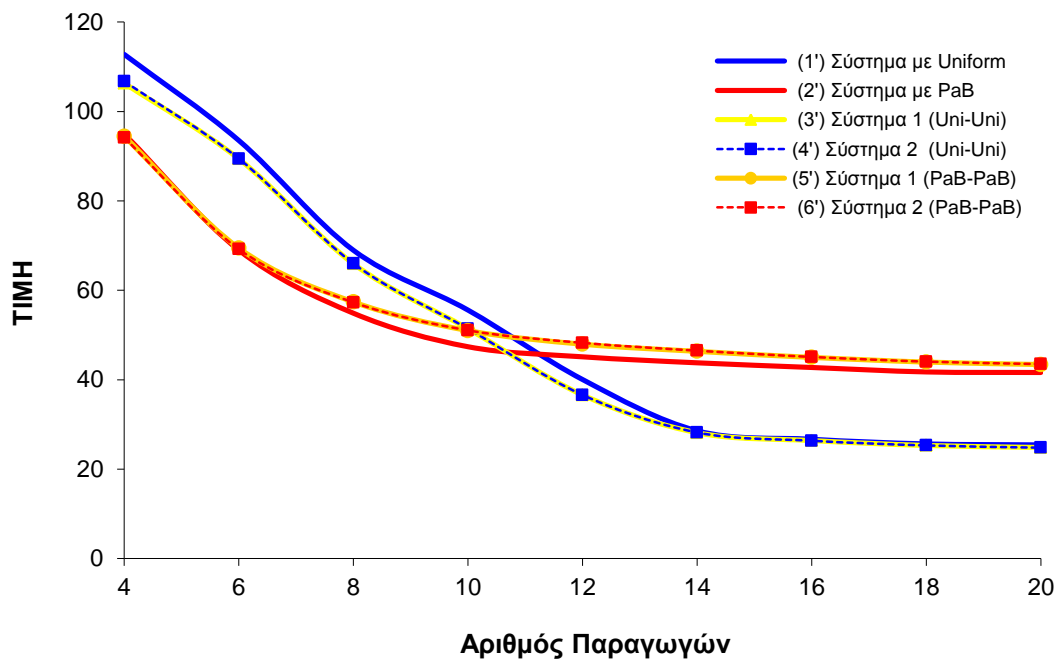


Σχήμα 51. Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με κοινή μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και δυνατότητα διάσπασης της προσφοράς.

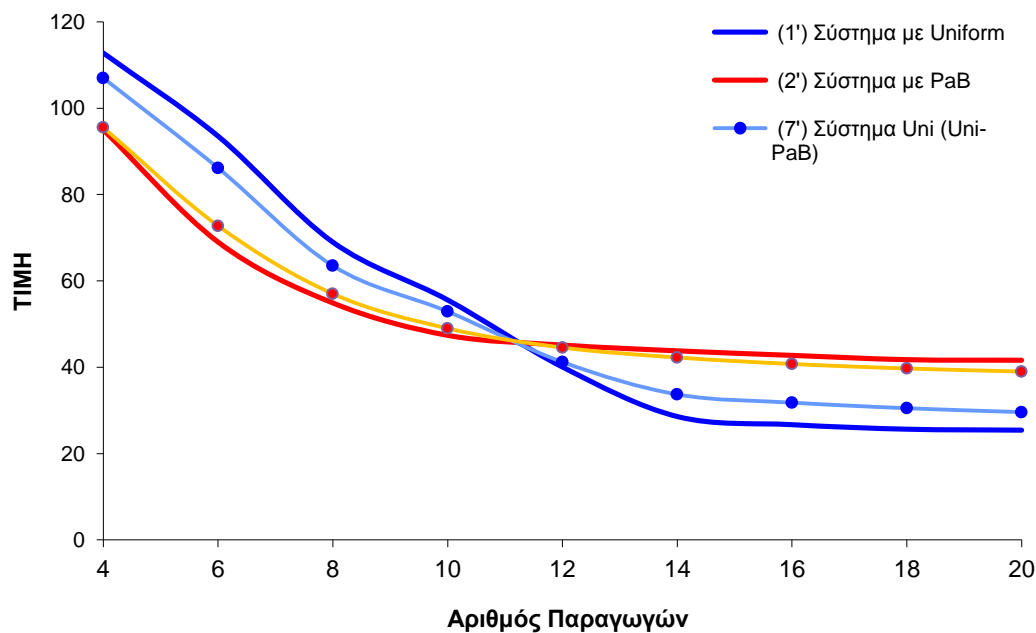


Σχήμα 52. Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με διαφορετική μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και δυνατότητα διάσπασης της προσφοράς.

Στα αντίστοιχα παίγνια με δύο διασυνδεδεμένα συστήματα όπου δεν υπάρχει δυνατότητα διάσπασης της προσφοράς και οι παίκτες προσφέρουν όλη τους τη δυναμικότητα είτε στο ένα είτε στο άλλο σύστημα (ανάλογα με το τι θεωρούν πιο συμφέρον), για παίγνια με κοινή μέθοδο τιμολόγησης (Σχήμα 53) και σε δύο συστήματα με διαφορετική μέθοδο τιμολόγησης (Σχήμα 54) παρουσιάζεται το ίδιο μοτίβο συμπεριφοράς αλλά τα αποτελέσματα των τιμών τείνουν περισσότερο προς τα αποτελέσματα των παιγνίων ενός συστήματος παρά σε αυτά των δύο συνδεδεμένων συστημάτων.



Σχήμα 53. Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με κοινή μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και χωρίς δυνατότητα διάσπασης της προσφοράς.



Σχήμα 54. Η σχέση τιμής και αριθμού παραγωγών σε παίγνια ενός και δύο συστημάτων με διαφορετική μέθοδο τιμολόγησης, απεριόριστη δυναμικότητα διασύνδεσης και χωρίς δυνατότητα διάσπασης της προσφοράς.

Τέλος, από τα αποτελέσματα που προκύπτουν από τα αντίστοιχα παίγνια με υφιστάμενο τον περιορισμό της δυναμικότητας διασύνδεσης φαίνεται πως ο περιορισμός της δυναμικότητας έχει αυξητική επίδραση στην τιμή της ηλεκτρικής ενέργειας και στα δύο συστήματα και ανεξάρτητα από το σύστημα τιμολόγησης που εφαρμόζεται. Αυτό προφανώς εξηγείται από το ότι ο περιορισμός παρεμποδίζει τη συμμετοχή περισσότερων παραγωγών στην τελική κατανομή γεγονός το οποίο θα είχε σαν αποτέλεσμα την περαιτέρω μείωση της τιμής.

6.3. Παιγνία σε Αγορά με Προσφορές Ζήτησης και Κατανεμημένη Παραγωγή Ενέργειας (DER)

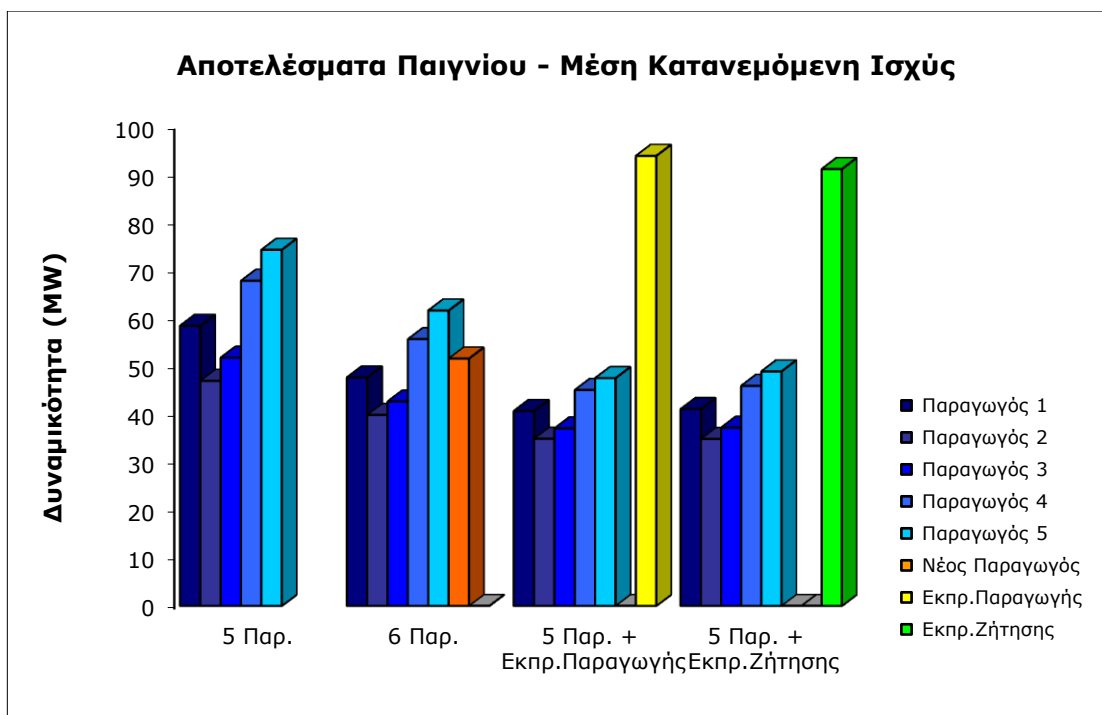
Προκειμένου να αξιολογηθεί η επίδραση της εισαγωγής της Κατανεμημένης Παραγωγής Ενέργειας (DER) σε μια αγορά ηλεκτρικής ενέργειας, είτε από την πλευρά της προσφοράς ή της ζήτησης προσομοιώνουμε μια μικρή αγορά ενέργειας αποτελούμενη από πέντε παραγωγούς με την ίδια ικανότητα παραγωγής (100 MW έκαστος). Η ζήτηση είναι ίση με 300 MW δηλαδή περισσότερο από το ήμισυ της συνολικής εγκατεστημένης ισχύος των παραγωγών που συμμετέχουν σε αυτή. Αφού πρώτα να προσδιορίζεται η οριακή τιμή του Συστήματος στην οποία συγκλίνει η αγορά αυτή, στη συνέχεια υποθέτουμε ότι ένας έκτος παραγωγός εισέρχεται στο παίγνιο, ενώ η ζήτηση παραμένει η ίδια. Ο νέος παραγωγός (παίκτης) έχει την ίδια δυναμικότητα παραγωγής με τους υπόλοιπους παραγωγούς, δηλαδή 100 MW. Εξετάζουμε τις περιπτώσεις όπου ο νέος παίκτης είναι (α) ένας συμβατικός παραγωγός και (β) ένας Εκπρόσωπος Κατανεμημένης Παραγωγής Ενέργειας (DER) και (γ) ένας Εκπρόσωπος Ζήτησης που χρησιμοποιεί Κατανεμημένη Παραγωγή (DER και προσφορές Διακοπτόμενου Φορτίου. Προκειμένου να διευκολυνθούν οι συγκρίσεις, θεωρήθηκε ότι όλοι οι παραγωγοί έχουν ίδιες μαθησιακές δεξιότητες και χαρακτηριστικά συμπεριφοράς κατά τη διάρκεια των παιγνίων.

Κάθε ένα από τα τέσσερα παίγνια που περιγράφονται παραπάνω αποτελείται από 500.000 επαναλήψεις (επαρκής αριθμός για τη σύγκλιση του παιγνίου) και έχει εκτελεσθεί 200 φορές, προκειμένου να προσδιορισθεί η μέση τιμή σύγκλισης για την Οριακή Τιμή Συστήματος και τα μερίδια αγοράς των Παραγωγών (κατανεμόμενη ισχύς), μεγέθη τα οποία θεωρούνται ως το αποτέλεσμα του παιγνίου (payoff). Έτσι, για κάθε παίγνιο έχουν εκτελεστεί αθροιστικά 100.000.000 επαναλήψεις, ενώ ο συνολικός αριθμός των επαναλήψεων και για τα τέσσερα παίγνια ανέρχεται σε 400.000.000. Τα αποτελέσματα που αφορούν στην Οριακή Τιμή τους του Συστήματος συνοψίζονται κατωτέρω:

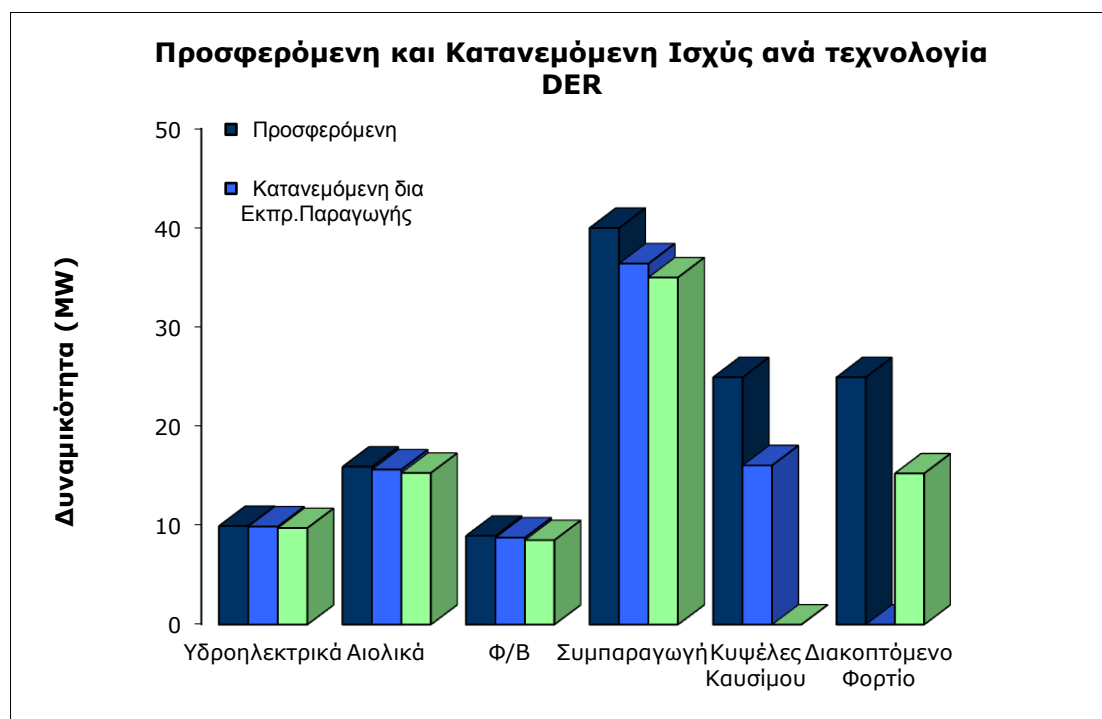
Περιγραφή της Αγοράς	Τιμή	Μείωση (%)
5 Συμβατικοί Παραγωγοί (βασικό παίγνιο)	42.21	-
5 + 1 Συμβατικοί Παραγωγοί	34.88	- 17.37 %
5 Συμβατικοί Παραγωγοί + 1 Εκπρόσωπος Παραγωγής	32.33	- 23.41 %
5 Συμβατικοί Παραγωγοί + 1 Εκπρόσωπος Ζήτησης	32.34	- 23.37 %

Είναι σαφές ότι η εισαγωγή του Εκπρόσωπου Παραγωγής έχει επιπτώσεις στην τιμή και αυτό αντικατοπτρίζεται και στο μερίδιο αγοράς που οι δύο τύποι Εκπροσώπων αποσπούν από την αγορά. Τα μέσα μερίδια αγοράς όλων των συμμετεχόντων και στους τέσσερις τύπους παιγνίων απεικονίζονται στο Σχήμα 55. Μια άλλη ενδιαφέρουσα πτυχή των αποτελεσμάτων των παιγνίων αφορά στο μέσο μερίδιο της αγοράς που αποσπά κάθε τεχνολογία παραγωγής DER σε σύγκριση με την παραγωγή προσφορά χωρητικότητας του (Σχήμα 56).

Το ανωτέρω παράδειγμα μοντέλου αγοράς που προσομοιώθηκε της για την εκτέλεση των παιγνίων και τα αντίστοιχα αποτελέσματα αφορά σε μια θεωρητική αγορά. Τα δε αποτελέσματα έχουν περισσότερο ποιοτικό και συγκριτικό χαρακτήρα, και δεν είναι επαρκή για να οδηγήσουν σε ασφαλείς ποσοτικές εκτιμήσεις. Αυτό θα μπορούσε να γίνει μόνο αν το μοντέλο προσομοίωνε μια ρεαλιστική αγορά ηλεκτρικής ενέργειας όπου θα υπήρχαν διαθέσιμα ακριβή στοιχεία σχετικά με το κόστος παραγωγής καθώς και την απόκριση του φορτίου έναντι των διακυμάνσεων της τιμής.



Σχήμα 55. Μέση Κατανεμόμενη Ισχύς για τους συμμετέχοντες και στους τέσσερις τύπους παιχνιδιών.



Σχήμα 56. Εγκατεστημένη (Προσφερόμενη) Ισχύς και Μέση Κατανεμόμενη για όλες τις τεχνολογίες παραγωγής της Καταμεμημένης Παραγωγής Ενέργειας (DER generation) και της Αποκοπής Φορτίου.

6.4. Σύγκριση της Θεωρητικής Λύσης του Παιγνίου Nash και των Αποτελεσμάτων του Μοντέλου

Στο παίγνιο του μοντέλου της αγοράς ηλεκτρικής ενέργειας που περιγράφηκε στα προηγούμενα κεφάλαια συμμετέχουν n παίκτες ο καθένας εκ των οποίων έχει μια συνάρτηση προσδιορισμού του κέρδους του σε κάθε επανάληψη του παιγνίου η τιμή της οποίας αποτελεί και το αποτέλεσμα του παιγνίου (payoff) για τον κάθε παίκτη. Η Συνάρτηση Κέρδους του κάθε παίκτη εμπεριέχει τις μεταβλητές απόφασης του παιγνίου για τον συγκεκριμένο παίκτη που δεν είναι άλλοι από τους Συντελεστές της Προσφοράς του. Οι μεταβλητές αυτές σε συνάρτηση και με τις τιμές που έχουν θέσει και οι υπόλοιποι παίκτες στις αντίστοιχες μεταβλητές τους καθορίζουν την ποσότητα παραγωγής ανά παίκτη/παραγωγό και φυσικά την Οριακή Τιμή του Συστήματος. Τα δύο αυτά μεγέθη προσδιορίζουν τα έσοδα των παικτών ενώ το πρώτο μέγεθος καθορίζει το αντίστοιχο συνολικό κόστος παραγωγής. Από την αφαίρεση των εξόδων από τα έσοδα προκύπτει το κέρδος του κάθε παίκτη. Παρουσιάζεται στη συνέχεια η θεωρητική επίλυση ενός απλού παιγνίου δύο παικτών με σταθερή ζήτηση και συγκρίνεται με τα αποτελέσματα που προκύπτουν από τις πειραματικές προσομοιώσεις που διεξήχθησαν για το ίδιο παίγνιο.

6.4.1. Παιγνιο με Uniform Pricing

Προσδιορισμός των Συναρτήσεων Κέρδους

Οι συναρτήσεις κέρδους των παικτών είναι στην ουσία αυτές που παράγουν το αποτέλεσμα (payoff) του παιγνίου και είναι απαραίτητο να προσδιορισθούν προκειμένου να επιλυθεί αναλυτικά το παίγνιο. Στο απλό παίγνιο με δύο παίκτες χωρίς τεχνικό ελάχιστο παραγωγής, σε αγορά ηλεκτρικής ενέργειας με Uniform Pricing τιμολόγηση και σταθερή ζήτηση D , οι συναρτήσεις κέρδους f_1 και f_2 για τους παίκτες 1 και 2 αντίστοιχα, είναι οι ακόλουθες:

$$f_1(x_1) = \lambda x_1 - f_{c_1}(x_1) = \lambda x_1 - (FC_1 + \alpha_1 x_1 + b_1 x_1^2) \quad (6.4.1.α)$$

και

$$f_2(x_2) = \lambda x_2 - f_{c_2}(x_2) = \lambda x_2 - (FC_2 + \alpha_2 x_2 + b_2 x_2^2) \quad (6.4.1.β)$$

όπου:

λ η Οριακή Τιμή του Συστήματος

x_1, x_2 οι ποσότητες ηλεκτρικής ενέργειας για τις οποίες κατανέμονται οι δύο παίκτες

FC_1, α_1, b_1 οι συντελεστές της συνάρτησης κόστους $f_{c_1}(x)$ του παίκτη 1

FC_2, α_2, b_2 οι συντελεστές της συνάρτησης κόστους $f_{c_2}(x)$ του παίκτη 2

Για τον προσδιορισμό της Οριακής Τιμής του Συστήματος λ και των ποσοτήτων x_1, x_2 για τις οποίες κατανέμονται οι δύο συμμετέχοντες, ισχύουν οι ακόλουθες σχέσεις:

$$x_1 + x_2 = D \quad (6.4.1.γ)$$

$$\lambda = A_1 + B_1 x_1 = A_2 + B_2 x_2 \Rightarrow x_1 = \frac{A_2 - A_1 + B_2 D}{B_1 + B_2} \quad \text{και} \quad x_2 = \frac{A_1 - A_2 + B_1 D}{B_1 + B_2} \quad (6.4.1.δ)$$

Γενικότερα ισχύουν οι περιορισμοί:

$$B_1, B_2 \geq 0$$

$$0 \leq x_1 \leq x_{1 \max} \quad \text{και} \quad 0 \leq x_2 \leq x_{2 \max}$$

$$A_1 + B_1 x_{1 \max} \leq \text{Price Cap} \quad (6.4.1.ε)$$

$$A_2 + B_2 x_{2 \max} \leq \text{Price Cap}$$

Βάσει των σχέσεων (6.4.1.α-δ) η συνάρτηση κέρδους του κάθε παίκτη μπορεί να μετασχηματισθεί ως συνάρτηση με μεταβλητές τους δύο Συντελεστές Προσφοράς του αλλά και τους Συντελεστές Προσφοράς του/των αντιπάλου/ων του. Το αποτέλεσμα δηλαδή του παιγνίου δεν εξαρτάται μόνο από τις επιλογές του αλλά από τις επιλογές του σε σχέση με τις επιλογές που έχει κάνει ο αντίπαλος ή οι αντίπαλοί του.

Τελικά οι συναρτήσεις κέρδους των δύο παικτών με μεταβλητές τους Συντελεστές Προσφορών τους διαμορφώνονται ως ακολούθως:

$$f_1(A_1, B_1, A_2, B_2) = \{[(2b_1 + B_2)(A_2 + B_2D) + \alpha_1 B_2]A_1 + (A_2 + B_2D)(A_2 + B_2D - \alpha_1)B_1 - (b_1 + B_2)A_1^2 - (A_2 + B_2D - \alpha_1)A_1B_1 - (A_2 + B_2D)[\alpha_1 B_2 + b_1(A_2 + B_2D)]\} \cdot \frac{1}{(B_1 + B_2)^2} - FC_1$$

$$f_2(A_1, B_1, A_2, B_2) = \{[(2b_2 + B_1)(A_1 + B_1D) + \alpha_2 B_1]A_2 + (A_1 + B_1D)(A_1 + B_1D - \alpha_2)B_2 - (b_2 + B_1)A_2^2 - (A_1 + B_1D - \alpha_2)A_2B_2 - (A_1 + B_1D)[\alpha_2 B_1 + b_2(A_1 + B_1D)]\} \cdot \frac{1}{(B_1 + B_2)^2} - FC_2$$

Συνθήκες Σημείου Ισορροπίας κατά Nash

Οι παίκτες προσπαθούν να μεγιστοποιήσουν τη συνάρτηση κέρδους τους αγνοώντας τις μεταβλητές απόφασης των αντιπάλων τους (μηδενισμός της μερικής πρώτης παραγώγου της συνάρτησης κέρδους ως προς τις δικές τους μεταβλητές απόφασης). Οπότε οι συνθήκες για τον προσδιορισμό του Σημείου Ισορροπίας κατά Nash είναι οι ακόλουθες:

$$\frac{\partial f_1(A_1, B_1, A_2, B_2)}{\partial A_1} = 0$$

$$\frac{\partial f_1(A_1, B_1, A_2, B_2)}{\partial A_2} = 0$$

$$\frac{\partial f_2(A_1, B_1, A_2, B_2)}{\partial B_1} = 0$$

$$\frac{\partial f_2(A_1, B_1, A_2, B_2)}{\partial B_2} = 0$$

όπου:

f_1, f_2 είναι οι συναρτήσεις κέρδους για τους παίκτες 1 και 2 αντίστοιχα,

A_1, B_1 οι συντελεστές της προσφοράς του παίκτη 1 και

A_2, B_2 οι συντελεστές της προσφοράς του παίκτη 2 (μεταβλητές απόφασης).

Η αναλυτική λύση του Παιγνίου προκύπτει επιλύοντας τις ανωτέρω εξισώσεις και είναι της μορφής:

$$\begin{bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{bmatrix} = \begin{bmatrix} f(B_1, B_2) \\ B_1 \\ f(B_1, B_2) \\ B_2 \end{bmatrix}$$

Υπάρχουν δηλαδή πολλαπλές λύσεις και κατά συνέπεια πολλαπλά Σημεία Ισορροπίας Nash.

Αν μετασχηματίσουμε τις συναρτήσεις κέρδους ως προς τις κατανεμόμενες ποσότητες ενέργειας, δηλαδή (α) για τον Παίκτη 1 η συνάρτηση κέρδους του να περιέχει ως μεταβλητές την ποσότητα ενέργειας x_1 και τους Συντελεστές

Προσφοράς του αντιπάλου του A_2, B_2 , ενώ (β) για τον Παίκτη 2 η συνάρτηση κέρδους του να περιέχει ως μεταβλητές την αντίστοιχη ποσότητα ενέργειας x_2 και τους Συντελεστές Προσφοράς του αντιπάλου του A_1, B_1 έχουμε:

$$f_1(x_1, A_2, B_2) = \lambda x_1 - (FC_1 + \alpha_1 x_1 + b_1 x_1^2) = [A_2 + B_2 (D - x_1)]x_1 - FC_1 - \alpha_1 x_1 - b_1 x_1^2$$

$$f_2(x_2, A_1, B_1) = \lambda x_2 - (FC_2 + \alpha_2 x_2 + b_2 x_2^2) = [A_1 + B_1 (D - x_2)]x_2 - FC_2 - \alpha_2 x_2 - b_2 x_2^2$$

Για κάθε καμπύλη προσφοράς του Παίκτη 2 (A_2, B_2) υπάρχει ένα σημείο (λ, x_1) για τον Παίκτη 1 (το οποίο μπορεί να δοθεί με άπειρους συνδυασμούς των Συντελεστών Προσφοράς A_1 και B_1) στο οποίο η συνάρτηση κέρδους του Παίκτη 1 μεγιστοποιείται (Σχήμα 57). Το σύνολο των σημείων αυτών (λ^*, x_1^*) που προκύπτουν για κάθε πιθανό συνδυασμό των Συντελεστών Προσφοράς του Παίκτη 2 ορίζουν την Καμπύλη Αντίδρασης του Παίκτη 1. Αντίστοιχα ορίζεται και η Καμπύλη Αντίδρασης του Παίκτη 2. Το σημείο (ή τα σημεία) τομής των δύο καμπυλών ορίζουν το σημείο (ή τα σημεία) ισορροπίας κατά Nash.

Οι συνθήκες σημείου ισορροπίας κατά Nash χρησιμοποιώντας ως μεταβλητή την κατανεμόμενη ποσότητα ενέργειας x_i αντί των Συντελεστών Προσφοράς (A_i, B_i) για τον κάθε παίκτη προσδιορίζονται ως ακολούθως:

$$\frac{\partial f_1(x_1, A_2, B_2)}{\partial x_1} = 0$$

$$\Rightarrow \frac{\partial f_1}{\partial x_1} = A_2 + B_2 D - \alpha_1 - 2(b_1 + B_2)x_1 = 0 \Rightarrow x_1^* = \frac{A_2 + B_2 D - \alpha_1}{2(b_1 + B_2)}$$

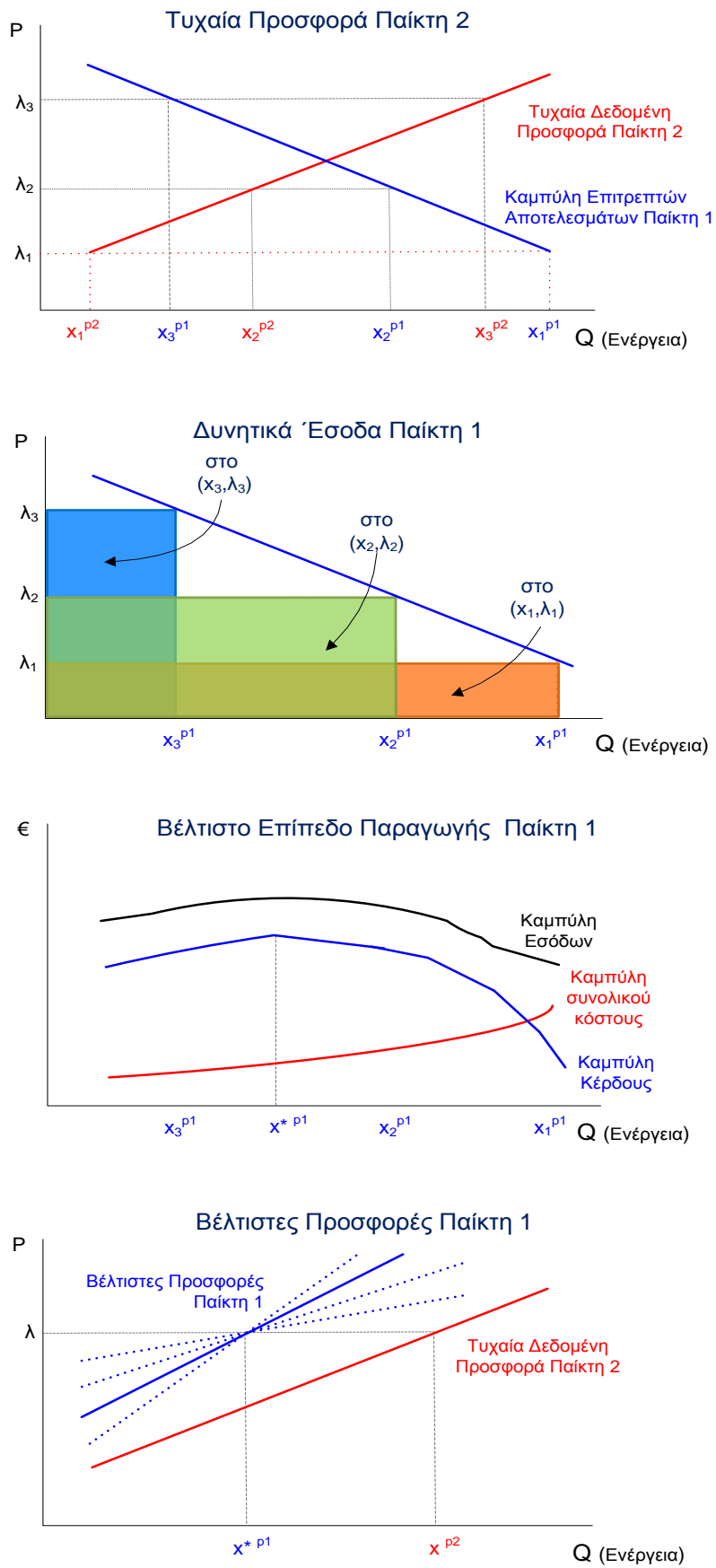
και

$$\frac{\partial f_2(x_2, A_1, B_1)}{\partial x_2} = 0$$

$$\Rightarrow \frac{\partial f_2}{\partial x_2} = A_1 + B_1 D - \alpha_2 - 2(b_2 + B_1)x_2 = 0 \Rightarrow x_2^* = \frac{A_1 + B_1 D - \alpha_2}{2(b_2 + B_1)}$$

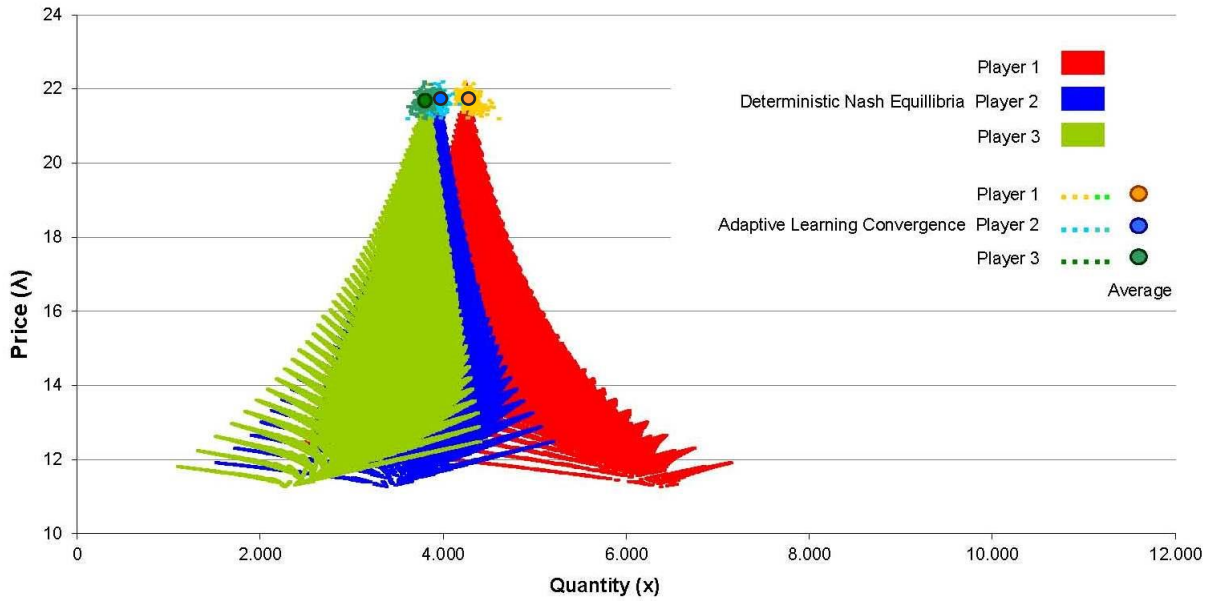
Για να απεικονισθεί η συγκέντρωση των λύσεων Nash (x_1^*, x_2^*) που προκύπτουν, υπολογίζονται τα A_1, A_2 για κάθε $B_1, B_2 \in [0, k)$, όπου k ένας πραγματικός αριθμός. Τα αποτελέσματα της απεικόνισης των αποτελεσμάτων του Παιγνίου σε ένα διάγραμμα Τιμής-Ποσότητας δείχνουν ότι όντως υπάρχει μια περιοχή αρκετά εκτεταμένη όπου συσσωρεύονται τα σημεία ισορροπίας κατά Nash. Η περιοχή αυτή είναι «φραγμένη» από τους περιορισμούς του προβλήματος δηλαδή το Ανώτατο Επιτρεπτό Όριο Προσφοράς (*Price Cap*), το ύψος της Ζήτησης (D), τη μέγιστη δυναμικότητα (x_{max}) και το κόστος παραγωγής των παικτών. Η αλλαγή των τιμών των περιορισμών, π.χ. διπλασιασμός του *Price Cap*, μεταβάλλει αντίστοιχα τη γεωγραφία της περιοχής συσσώρευσης των σημείων ισορροπίας Nash.

Αυτό που παρουσιάζει ωστόσο μεγαλύτερο ενδιαφέρον είναι η θέση των πειραματικών τιμών σύγκλισης των παιγνίων που διενεργούνται με τον μαθησιακό αλγόριθμο σε σχέση με τις τιμές που προκύπτουν από την αναλυτική λύση του ίδιου παιγνίου. Η σύγκριση έγινε για απλά παίγνια με δύο αλλά και με τρεις παίκτες μόνο, καθώς η υπολογιστική πολυπλοκότητα δεν επέτρεψε την εξαγωγή αναλυτικής λύσης σε παίγνιο με πάνω από τρεις παίκτες. Εξαντλητικές προσομοιώσεις για διάφορες συνθήκες της αγοράς που διεξήχθησαν με το λογισμικό *EM Simulator* έδειξαν ότι οι πειραματικές λύσεις (τιμές σύγκλισης των παιγνίων) βρίσκονται όλες εντός των περιοχών των σημείων ισορροπίας κατά Nash και μάλιστα στην περιοχή των πιο ελκυστικών λύσεων για τους παίκτες μεταξύ όλων των σημείων ισορροπίας κατά Nash (Σχήμα 58).

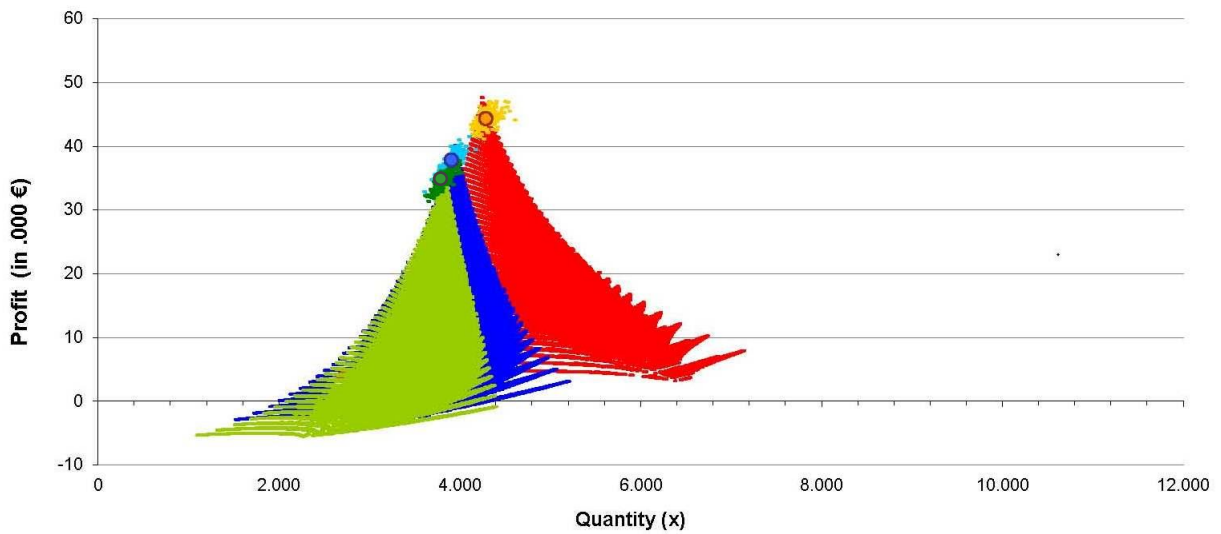


Σχήμα 57. Προσδιορισμός του βέλτιστου επιπέδου παραγωγής και αντιστοίχων προσφορών του Παίκτη 1 για δεδομένη Προσφορά του Παίκτη 2

Comparison of Deterministic Nash Equilibria and Adaptive Learning Solutions in the Uniform Pricing Game for 3 Players



Values of Objective Function (Profit) for the NE Area & the Adaptive Learning Solutions in the Uniform Pricing Game for 3 Players



Game Parameters

	Player 1	Player 2	Player 3
α_i	8,40	9,47	10,20
β_i	0,00020	0,00024	0,00021
FC_i	7.000	6.100	6.900
X_{min}	0	0	0
X_{max}	12.000	12.000	12.000
D	12.000	3	

Deterministic Solution

3.375.000 B1,B2,B3 Combinations

	λ	x_1	x_2	x_3
Ave	14,99	4.746	3.805	3.449
max	21,92	4.238	3.940	3.822

	J1	J2	J3
Ave J	19	12	8
max J	47	39	35

Adaptive Learning Solution

200 Games with 500.000 runs each

	λ	x_1	x_2	x_3
Ave	21,68	4.290	3.907	3.804

	J1	J2	J3
Ave	44	38	34

Σχήμα 58. Σύγκριση των σημείων ισορροπίας κατά Nash (αναλυτικά υπολογισμένων) και των σημείων σύγκλισης του προσαρμοστικού μαθησιακού μοντέλου στο Uniform Pricing Παίγνιο με 3 παίκτες .

6.4.2. Παίγνιο με Pay-As-Bid

Προσδιορισμός των Συναρτήσεων Κέρδους

Στο ίδιο παίγνιο με δύο παίκτες σε αγορά ηλεκτρικής ενέργειας με Pay-As-Bid τιμολόγηση και σταθερή ζήτηση D , οι συναρτήσεις κέρδους f_1 και f_2 για τους παίκτες 1 και 2 αντίστοιχα, είναι οι ακόλουθες:

$$f_1(x_1) = fb_1(x_1) - fc_1(x_1) = A_1x_1 + \frac{1}{2}B_1x_1^2 - (FC_1 + \alpha_1x_1 + b_1x_1^2) \quad (6.4.2.α)$$

$$f_2(x_2) = fb_2(x_2) - fc_2(x_2) = A_2x_2 + \frac{1}{2}B_2x_2^2 - (FC_2 + \alpha_2x_2 + b_2x_2^2) \quad (6.4.2.β)$$

όπου:

x_1, x_2 οι ποσότητες ηλεκτρικής ενέργειας για τις οποίες κατανέμονται οι δύο παίκτες

A_1, B_1 οι συντελεστές της συνάρτησης Προσφοράς $fb_1(x)$ του παίκτη 1

A_2, B_2 οι συντελεστές της συνάρτησης Προσφοράς $fb_2(x)$ του παίκτη 2

FC_1, α_1, b_1 οι συντελεστές της συνάρτησης κόστους $fc_1(x)$ του παίκτη 1

FC_2, α_2, b_2 οι συντελεστές της συνάρτησης κόστους $fc_2(x)$ του παίκτη 2

Με βάση τις ανωτέρω εξισώσεις και λαμβάνοντας υπόψη τις εξισώσεις (6.4.1.γ) και (6.4.1.δ) που ισχύουν και σε αυτή την περίπτωση προκύπτουν και τους γενικούς περιορισμούς (6.4.1.ε), η συνάρτηση κέρδους του κάθε παίκτη μετασχηματίζεται ως συνάρτηση με μεταβλητές τους δύο συντελεστές της προσφοράς του αλλά και τους συντελεστές Προσφοράς του/των αντιπάλου/ων του :

$$f_1(A_1, B_1, A_2, B_2) = \left\{ [(2b_1 + B_2)(A_2 + B_2D) + \alpha_1 B_2] A_1 + (A_2 + B_2D) \left[\frac{1}{2}(A_2 + B_2D) - \alpha_1 \right] B_1 + \alpha_1 A_1 B_1 - \frac{1}{2} A_1^2 B_1 - (b_1 + B_2) A_1^2 - (A_2 + B_2D)[\alpha_1 B_2 + b_1(A_2 + B_2D)] \right\} \cdot \frac{1}{(B_1 + B_2)^2} - FC_1$$

$$f_2(A_1, B_1, A_2, B_2) = \left\{ [(2b_2 + B_1)(A_1 + B_1D) + \alpha_2 B_1] A_2 + (A_1 + B_1D) \left[\frac{1}{2}(A_1 + B_1D) - \alpha_2 \right] B_2 + \alpha_2 A_2 B_2 - \frac{1}{2} A_2^2 B_2 - (b_2 + B_1) A_2^2 - (A_1 + B_1D)[\alpha_2 B_1 + b_2(A_1 + B_1D)] \right\} \cdot \frac{1}{(B_1 + B_2)^2} - FC_2$$

Ισχύουν και σε αυτή την περίπτωση οι ακόλουθες σχέσεις:

$$x_1 + x_2 = D \quad (6.4.2.γ)$$

$$\lambda = A_1 + B_1x_1 = A_2 + B_2x_2 \Rightarrow x_1 = \frac{A_2 - A_1 + B_2D}{B_1 + B_2} \quad \text{και} \quad x_2 = \frac{A_1 - A_2 + B_1D}{B_1 + B_2} \quad (6.4.2.δ)$$

και οι περιορισμοί:

$$B_1, B_2 \geq 0$$

$$0 \leq x_1 \leq x_{1 \max} \quad \text{και} \quad 0 \leq x_2 \leq x_{2 \max}$$

$$A_1 + B_1x_{1 \max} \leq Price \ Cap \quad (6.4.2.ε)$$

$$A_2 + B_2x_{2 \max} \leq Price \ Cap$$

Συνθήκες Σημείου Ισορροπίας κατά Nash

Οι αντίστοιχες συνθήκες για τον προσδιορισμό του Σημείου Ισορροπίας κατά Nash είναι οι ακόλουθες:

$$\frac{\partial f_1(A_1, B_1, A_2, B_2)}{\partial A_1} = 0$$

$$\frac{\partial f_1(A_1, B_1, A_2, B_2)}{\partial A_2} = 0$$

$$\frac{\partial f_2(A_1, B_1, A_2, B_2)}{\partial B_1} = 0$$

$$\frac{\partial f_2(A_1, B_1, A_2, B_2)}{\partial B_2} = 0$$

όπου:

f_1, f_2 είναι οι συναρτήσεις κέρδους για τους παίκτες 1 και 2 αντίστοιχα,

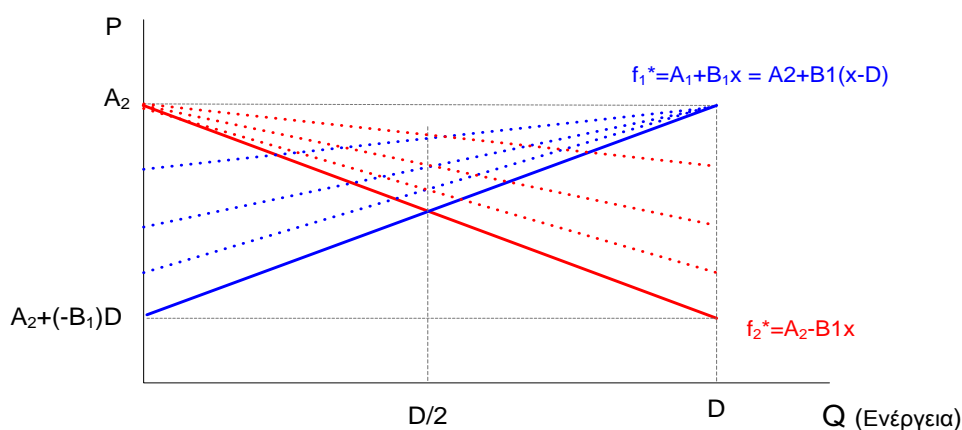
A_1, B_1 οι συντελεστές της προσφοράς του παίκτη 1 και

A_2, B_2 οι συντελεστές της προσφοράς του παίκτη 2 (μεταβλητές απόφασης).

Η αναλυτική επίλυση του Παιγνίου δίνει δύο λύσεις που είναι οι ακόλουθες:

$$\begin{bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{bmatrix} = \begin{bmatrix} A_2 + (-B_1)D \\ B_1 \\ A_2 \\ -B_1 \end{bmatrix} \quad \text{και} \quad \begin{bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{bmatrix} = \begin{bmatrix} \alpha_2 + B_2D \\ -B_2 \\ \alpha_2 \\ B_2 \end{bmatrix}$$

Κατωτέρω απεικονίζονται οι θεωρητικές καμπύλες (άπειρες) που αντιπροσωπεύει η πρώτη από τις δύο λύσεις για μια δεδομένη τιμή της Παραμέτρου Προσφοράς A_2 .



Σχήμα 59. Σχηματική απεικόνιση των καμπυλών της πρώτης αναλυτικής λύσης του παιγνίου δύο παικτών με Pay-As-Bid τιμολόγηση.

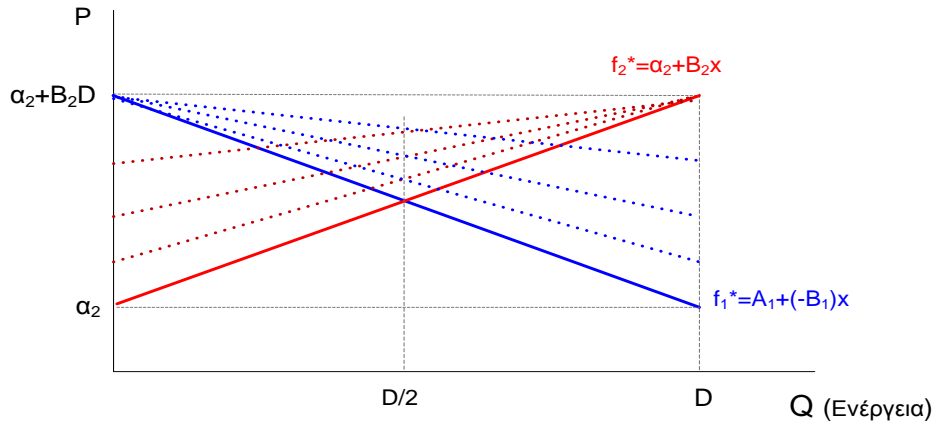
Επειδή ωστόσο οι μόνες τιμές που ικανοποιούν τις συνθήκες:

$$\left. \begin{array}{l} B_1, B_2 \geq 0 \\ B_1 = B_1 \\ B_2 = -B_1 \end{array} \right\} \Rightarrow B_1 = B_2 = 0$$

η λύση ουσιαστικά μετασχηματίζεται ως εξής:

$$\begin{bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{bmatrix} = \begin{bmatrix} A_2 \\ 0 \\ A_2 \\ 0 \end{bmatrix}$$

Οι καμπύλες που αντιστοιχούν στη δεύτερη λύση για μια συγκεκριμένη τιμή του συντελεστή Προσφοράς A_2 απεικονίζονται στο κατωτέρω σχήμα.



Σχήμα 60. Σχηματική απεικόνιση των καμπυλών της δεύτερης αναλυτικής λύσης του παιγνίου δύο παικτών με Pay-As-Bid τιμολόγηση.

Ομοίως οι μόνες τιμές που ικανοποιούν ταυτόχρονα τον περιορισμό $B_1, B_2 \geq 0$

και τις συνθήκες $B_1 = -B_2$ και $B_2 = B_2$ είναι οι ακόλουθες:

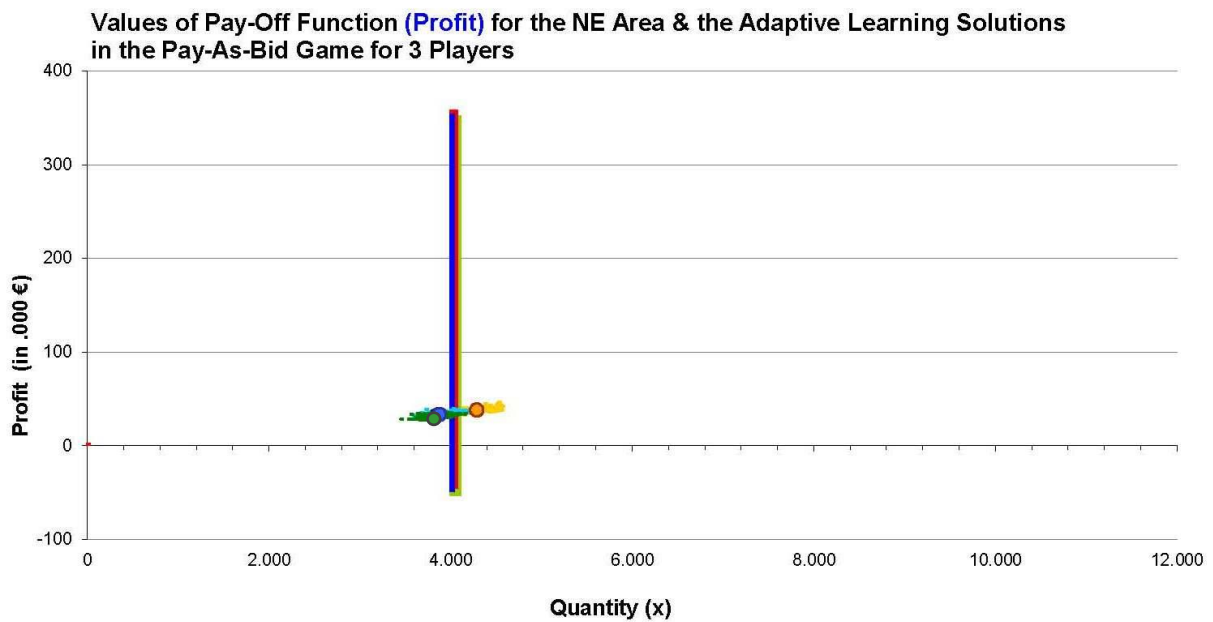
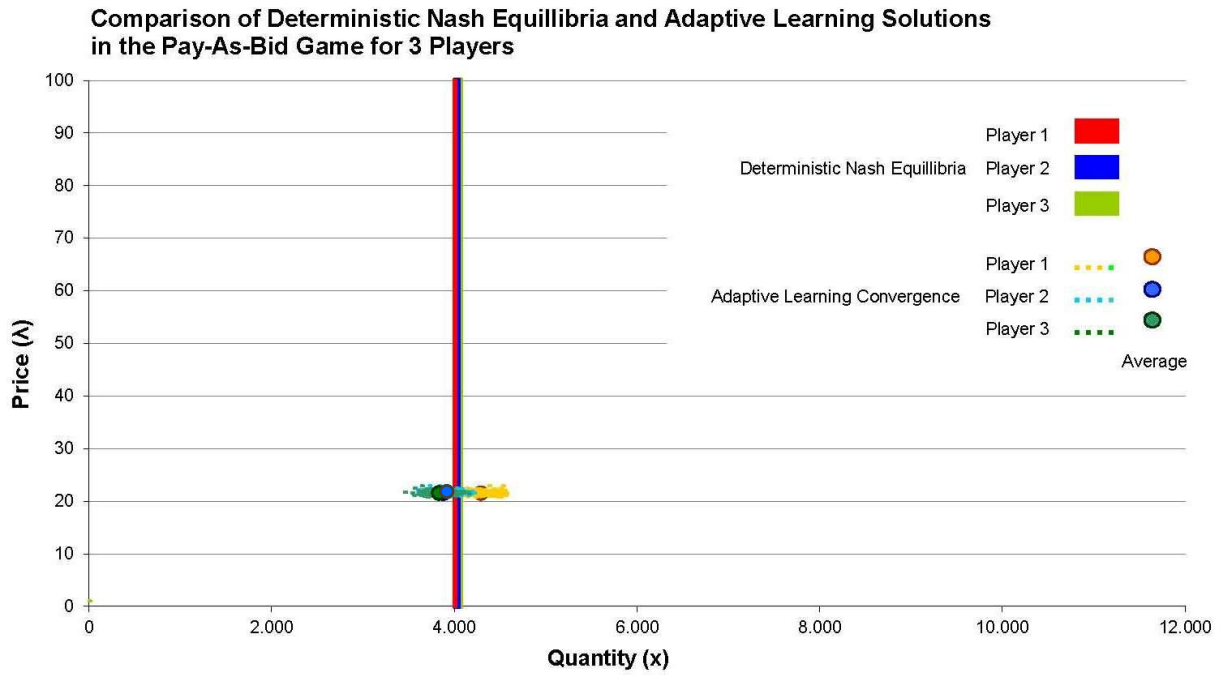
$$B_1 = B_2 = 0$$

Έτσι η λύση ουσιαστικά μετασχηματίζεται ως εξής:

$$\begin{bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{bmatrix} = \begin{bmatrix} \alpha_2 \\ 0 \\ \alpha_2 \\ 0 \end{bmatrix}$$

η οποία υπερκαλύπτεται από την πρώτη λύση.

Συμπερασματικά, τα σημεία ισορροπίας κατά Nash σε παίγνιο με Pay-As-Bid τιμολόγηση προκύπτουν από προσφορές με $B_1=B_2=0$ και $A_1=A_2$, δηλαδή ταυτόσημες προσφορές. Έχουμε δηλαδή πολλαπλά σημεία ισορροπίας κατά Nash τα οποία αντιστοιχούν σε ισόποση κατανομή των ζήτησης ανάμεσα στους δύο παίκτες για όλα τα επιτρεπτά επίπεδα της Οριακής Τιμής του Συστήματος. Το ίδιο αποτέλεσμα προκύπτει και για το παίγνιο με τρεις παίκτες: ίσες ποσότητες παραγωγής για όλα τα επίπεδα Οριακής Τιμής. Η σύγκριση της αναλυτικής λύσης με τα πειραματικά αποτελέσματα που προέκυψαν από τις προσομοιώσεις που διεξήχθησαν για ένα παίγνιο με τρεις παίκτες απεικονίζονται στο Σχήμα 61. Τα αποτελέσματα του μοντέλου δείχνουν μια συγκέντρωση των τιμών σύγκλισης πολύ κοντά στα θεωρητικά Nash Equilibria (ισοκατανομή της ζήτησης) σε χαμηλό σχετικά επίπεδο Οριακής Τιμής (διπλάσιο περίπου από αυτή που θα προέκυπτε με προσφορές τα οριακά κόστη και στο ένα πέμπτο του επιπέδου του Price Cap).



Game Parameters

	Player 1	Player 2	Player 3
α_i	8,40	9,47	10,20
β_i	0,00020	0,00024	0,00021
FC_i	7.000	6.100	6.900
X_{min}	0	0	0
X_{max}	12.000	12.000	12.000
D	12.000	3	

Deterministic Solution

3.375.000 B1,B2,B3 Combinations

	λ	x1	x2	x3
Ave	50,00	4.000	4.000	4.000
max	100,00	4.000	4.000	4.000

	J1	J2	J3
Ave J	156	152	149
max J	356	352	349

Adaptive Learning Solution

200 Games with 500.000 runs each

	λ	x1	x2	x3
Ave	21,54	4.290	3.878	3.832

	J1	J2	J3
Ave	38	33	31

Σχήμα 61. Σύγκριση των σημείων ισορροπίας κατά Nash (αναλυτικά υπολογισμένων) και των σημείων σύγκλισης του προσαρμοστικού μαθησιακού μοντέλου στο Pay-As-Bid Παιγνίο με 3 παίκτες .

7. Επισκόπηση του Έργου

Είναι σκόπιμο στο τέλος της διατριβής να γίνει μια αξιολόγηση των αποτελεσμάτων της έρευνας σε σχέση με αυτό που αρχικά είχε σχεδιασθεί, της φύσης και της αξιοπιστίας του μοντέλου που αναπτύχθηκε, των βασικών συμπερασμάτων που προέκυψαν καθώς και αναφορά στα κυριότερα προβλήματα που αντιμετωπίστηκαν κατά τη διάρκεια τα εκπόνησής της. Τέλος γίνεται μια εκτίμηση για τη δυνατότητα αξιοποίησής της και κάποιες κατευθύνσεις σχετικά με την περαιτέρω επέκτασή της.

7.1. Συμπεράσματα

Η πρώτη δέσμη συμπερασμάτων αφορά στον τρόπο που επιλέχθηκε για να γίνει η μοντελοποίηση της λειτουργίας μιας χονδρεμπορικής αγοράς ηλεκτρικής ενέργειας, δηλαδή ως προσαρμοστικό στοχαστικό μαθησιακό παίγνιο με ατελή πληροφόρηση των συμμετεχόντων ως προς τις επιλογές και τα αποτελέσματα των αντιπάλων. Από τη σύγκριση της αναλυτικής επίλυσης του ίδιου θεωρητικού παιγνίου και των πειραματικών αποτελεσμάτων του μοντέλου προσομοίωσης που αναπτύχθηκε προέκυψε ότι τα αποτελέσματα της προσομοίωσης βρίσκονται εντός της περιοχής συγκέντρωσης των σημείων ισορροπίας Nash του παιγνίου. Όταν δε μάλιστα εφαρμόζεται στην αγορά Uniform τιμολόγηση τα σημεία της πειραματικής σύγκλισης του παιγνίου παρατηρούνται στην άνω πλευρά της εν λόγω περιοχής δηλαδή ταυτίζονται με τις πιο ευνοϊκές λύσεις (σημεία ισορροπίας κατά Nash) που προκύπτουν από την αναλυτική επίλυση του παιγνίου. Στην μέθοδο Pay-As-Bid τα πειραματικά αποτελέσματα συγκεντρώνονται σε πιο μέτρια (ως προς το επίπεδο τιμών και εσόδων) περιοχή των αντίστοιχων σημείων ισορροπίας Nash.

Η επίδραση της επιλογής της μεθόδου τιμολόγησης αποδείχθηκε κρίσιμη για τη λειτουργία της αγοράς επηρεάζοντας δραστικά το επίπεδο της τιμής, τον τρόπο συμμετοχής των παικτών καθώς και τα έσοδά/κέρδη τους. Η αγορά με τη μέθοδο τιμολόγησης Pay-As-Bid, όπως αναφέρθηκε ανωτέρω, λειτουργεί οικονομικότερα όταν αυτή συνιστά ολιγοπώλιο, επιτυγχάνοντας ταυτόχρονα χαλιναγώγηση τυχόν τάσεων κερδοσκοπίας από πλευράς των παραγωγών. Στην περίπτωση ωστόσο που η αγορά τείνει προς τον τέλειο ανταγωνισμό η μέθοδος Uniform Pricing φαίνεται να είναι η πλέον ενδεδειγμένη. Ο αριθμός των συμμετεχόντων, όπως άλλωστε αναμενόταν, αποδείχθηκε καθοριστικός για την τιμή της ηλεκτρικής ενέργειας καθώς όσο η αγορά βαίνει από ολιγοπώλιο στον τέλειο ανταγωνισμό τόσο μειώνεται το επίπεδο στο οποίο συγκλίνει η τιμή και για τις δύο μεθόδους τιμολόγησης. Επιπλέον σημαντικό ρόλο φαίνεται να παίζει και η ύπαρξη ενός ανωτάτου όριο προσφοράς (Price Cap) το οποίο λειτουργεί ως ανασταλτικός παράγοντας σε τυχόν κερδοσκοπικές διαθέσεις των παραγωγών.

Στην περίπτωση δύο διασυνδεδεμένων αγορών ηλεκτρικής ενέργειας που εφαρμόζουν κοινό σύστημα τιμολόγησης προκύπτουν τιμές χαμηλότερες με Uniform Pricing σε συνθήκες ολιγοπωλίου, υψηλότερες με Pay-As-Bid στον τέλειο ανταγωνισμό και παρόμοιες σε όλες τις υπόλοιπες περιπτώσεις, σε σχέση με τις τιμές που θα είχαμε σε δύο αντίστοιχες ανεξάρτητες αγορές. Στην περίπτωση που οι δύο διασυνδεδεμένες αγορές εφαρμόζουν διαφορετικά συστήματα τιμολόγησης οι τιμές που προκύπτουν βρίσκονται σε ενδιάμεσα επίπεδα από αυτές που θα προέκυπταν αν οι αγορές ήταν ανεξάρτητες. Τυχόν δε, ύπαρξη φυσικού περιορισμού στη δυναμικότητα διασύνδεσης των δύο αγορών αμβλύνει απλώς την ένταση των ανωτέρω παρατηρούμενων φαινομένων.

Στο μοντέλο της αγοράς, τέλος, όπου υφίσταται κατανομημένη τοπική παραγωγή η οποία αντιπροσωπεύεται συνολικά είτε από Εκπροσώπους Προσφοράς στην μία περίπτωση είτε από Εκπροσώπους Ζήτησης στην άλλη, συνοδευόμενη και από προσφορές διακοπτόμενου φορτίου, προκύπτει ότι η προσθήκη όποιας από τις δύο προαναφερόμενες ισοδύναμες εκδοχές είναι σαφώς ελκυστικότερη από άποψη διαμορφούμενης τιμής σε σχέση με την περίπτωση εισαγωγής ενός ακόμη συμβατικού παραγωγού.

7.2. Κυριότερα προβλήματα κατά την εκπόνηση της διατριβής

7.2.1. Το πρόβλημα της κατανομής των παραγωγών

Το κυριότερο αλγοριθμικό πρόβλημα που προέκυψε ήταν αυτό της μοντελοποίησης του μηχανισμού κατανομής των παραγωγών με βάση την προσφορά που αυτοί κάνουν και με στόχο πάντα την ικανοποίηση της ζήτησης. Η ύπαρξη τεχνικών ελαχίστων στην διαθέσιμη ισχύ από την πλευρά των παραγωγών δημιουργούσε ασυνέχειες στην αθροιστική καμπύλη προσφοράς και κατά συνέπεια δυσχέρειες στην επίλυση του προβλήματος κατανομής οι οποίες αρχικά δεν είχαν προβλεφθεί. Έτσι ήταν αδύνατο να λυθεί το πρόβλημα ως απλό πρόβλημα γραμμικού προγραμματισμού και έπρεπε να εισαχθούν τροποποιήσεις στη μοντελοποίησή του. Τελικώς το πρόβλημα επιλύθηκε με σταδιακή επίλυση λαμβάνοντας κομβικά σημεία στην αθροιστική συνάρτηση προσφοράς τα οποία αντιστοιχούσαν στα τεχνικά ελάχιστα και μέγιστα των προσφορών των παραγωγών και τα οποία όριζαν διαστήματα με διαφορετικές συνθήκες και περιορισμούς. Έτσι εντοπίζεται αρχικά το διάστημα μέσα στο οποίο βρίσκεται η λύση και ακολούθως το πρόβλημα επιλύεται τοπικά ως ένα απλό γραμμικό πρόβλημα με περιορισμούς, οι οποίοι φυσικά διαφοροποιούνται από διάστημα σε διάστημα. Εκτενέστερα η επίλυση του προβλήματος παρουσιάστηκε στο αντίστοιχο κεφάλαιο της παρούσης διατριβής (4.1.3. Το πρόβλημα της κατανομής των παραγωγών και η επίλυσή του).

7.2.2. Η Συνάρτησης Ζήτησης

Η εισαγωγή συνάρτησης ζήτησης με την κατασκευή ενός πιο πολύπλοκου module από το χρησιμοποιούμενο το οποίο μπορούσε μόνο να παράγει τυχαίες τιμές για την αναμενόμενη ζήτηση γύρω από μια δεδομένη κεντρική τιμή και μια διακύμανση γύρω από αυτή, ήταν από τους αρχικούς στόχους της μοντελοποίησης (4.1.4 Η Συνάρτηση της Ζήτησης). Αρχικά όμως είχε σχεδιασθεί και η ενσωμάτωση προτύπων (profiles) ζήτησης που θα εμπειρεύσαν στοιχεία εποχικότητας και τάσης της ζήτησης ηλεκτρικής ενέργειας. Διαπιστώθηκε στην πορεία ότι η εισαγωγή τέτοιων προτύπων η οποία έγινε αρχικά δίνοντας ως συνάρτηση ζήτησης μια γραμμική συνάρτηση η οποία με την πάροδο του χρόνου υπόκειντο σε μια παλινδρομική μετατόπιση ημιτονοειδούς χαρακτήρα προκάλεσε διαφοροποιήσεις στα αποτελέσματα του μοντέλου μόνο ως προς την ταχύτητα σύγκλισής του και συγκεκριμένα την επιβράδυνε. Αυτό εξηγείται καθώς το μοντέλο είναι προσαρμοστικό και οποιαδήποτε αστάθεια ή συστηματική μεταβολή στη συμπεριφορά των παραμέτρων που το απαρτίζουν, πέραν εκείνων που βελτιστοποιούνται με την προσαρμογή, μπορεί μεν θεωρητικά να μοντελοποιεί καλύτερα την πραγματικότητα, έχει δε μόνον αρνητικές συνέπειες στην ταχύτητα εξαγωγής συμπερασμάτων ή ακόμη και στη ίδια τη δυνατότητα εξαγωγής τους (μη σύγκλιση). Εξάλλου, ο προσαρμοστικός χαρακτήρας των παιγνίων δεν έχει σα στόχο την πιστή προσομοίωση μιας πραγματικής κατάστασης αλλά την απομόνωση κάποιων μεταβλητών απόφασης και στη συνέχεια τη βελτιστοποίησή τους μέσα από τη διαδικασία της προσαρμοστικής μάθησης διατηρώντας όμως τις υπόλοιπες παραμέτρους του προσομοιούμενου περιβάλλοντος σταθερές. Έτσι με την εισαγωγή της συνάρτησης ελαστικότητας της ζήτησης μοντελοποιήθηκε η δυνατότητα μεταβολής της ζήτησης αποκλειστικά και μόνο λόγω της τιμής της προσφερόμενης ηλεκτρικής ενέργειας, απομονώνοντας τυχόν εποχιακές ή κλιματολογικές επιδράσεις αφού κάθε παίγνιο αντιπροσωπεύει μια πολύ μικρή χρονική περίοδο (π.χ. μία ώρα) όπου η ζήτηση δεν μεταβάλλεται ουσιαστικά για άλλους λόγους πέραν της τιμής.

7.2.3. Ο Προσαρμοστικός Μαθησιακός Αλγόριθμος

Κατά τη διάρκεια εκπόνησης της διατριβής εξετάσθηκε η βελτιστοποίηση του προσαρμοστικού αλγορίθμου εκμάθησης που χρησιμοποιούν οι παίκτες για να υποβάλλουν τις προσφορές τους. Συγκεκριμένα, δοκιμάσθηκε η περίπτωση να ορίζονται τα e_A και e_B ως μέγιστες τιμές, η τιμή τους σε ένα παίγνιο να μην είναι σταθερή ή με προκαθορισμένο τρόπο αυξανόμενη/μειούμενη, αλλά να παίρνει τιμές μέσα από δεδομένο εύρος, με βάση όμως το αποτέλεσμα του προηγούμενου γύρου ή κάποιων άλλων κριτηρίων που θα τεθούν. Η προσαρμοστικότητα δηλαδή

των παικτών επεκτάθηκε εκτός της περιοχής επιλογής των δράσεων και στο βαθμό που αυτές (οι δράσεις) συνέβαιναν.

Για παράδειγμα, η τιμή που λάμβανε το e_A ενός παίκτη στον γύρο k μπορούσε να δίνεται από τη σχέση:

$$e_{Ai}^k = (J_i^{k-1} / J_i^{\max}) * e_{Ai}^{\max}$$

όπου

e_{Ai}^k : η τιμή του e_{Ai} στον γύρο k

J_i^{k-1} : το έσοδο του παίκτη i στον προηγούμενο γύρο $k-1$

J_i^{\max} : το μέγιστο έσοδο του παίκτη i που έχει καταγραφεί μέχρι τον γύρο $k-1$

e_{Ai}^{\max} : η μέγιστη τιμή του e_{Ai} που έχει ορισθεί κατά την έναρξη του παιχνιδιού.

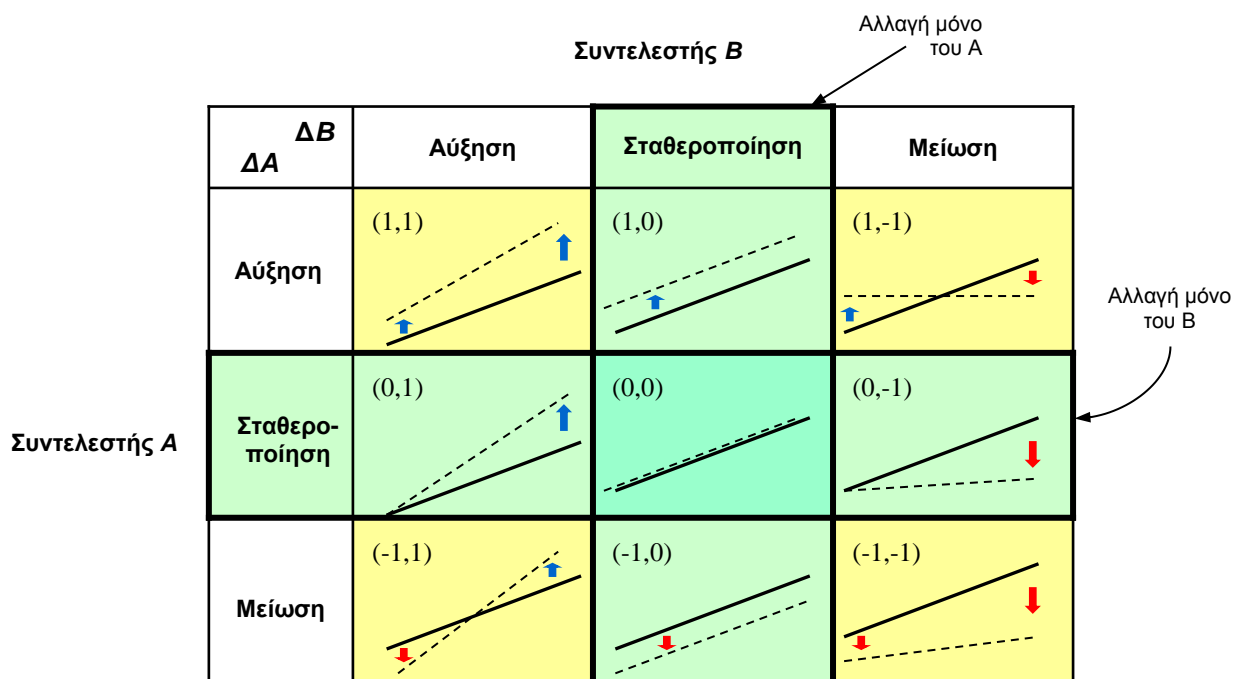
Επίσης εξετάστηκε το αν το κριτήριο του εσόδου είναι το πιο κατάλληλο για μεταβλητή απόφασης, καθώς το έσοδο εξαρτάται και από τα επίπεδα στα οποία κινείται η τιμή λ κατά τη διάρκεια των τελευταίων γύρων και ως εναλλακτικά κριτήρια απόφασης εισήχθησαν το μερίδιο αγοράς (S_i) του παίκτη καθώς και μια γραμμική συνάρτηση του εσόδου και του μεριδίου αγοράς της μορφής:

$$f_i(J, S) = a (J_i / J_{i \max}) + b S_i \quad (\text{όπου } a, b: \text{ συντελεστές βαρύτητας})$$

Επίσης εξετάστηκε και η πιθανή ουσιαστική βελτίωση στο αποτέλεσμα του κάθε παίκτη που θα επέφερε η εισαγωγή δυνατότητας μαθησιακής ικανότητας, όχι μόνο για καθένα συντελεστή A, B ξεχωριστά, αλλά για συνδυασμό κινήσεων των συντελεστών. Έτσι, αντί για τις δύο τρισδιάστατες στοχαστικές κατανομές

$$A(P^{in}, P^{de}, P^{st}) \text{ και } B(P^{in}, P^{de}, P^{st})$$

που αντιστοιχούσαν στους δύο συντελεστές γραμμικής προσφοράς, δημιουργήθηκε μια στοχαστική κατανομή που αναφερόταν στο συνδυασμό δράσεων των δύο συντελεστών με εννέα διαστάσεις όπως παρουσιάζεται στο Σχήμα 62.



Σχήμα 62. Οι δράσεις της στοχαστικής κατανομής απόφασης για την προσαρμογή της γραμμικής συνάρτησης προσφοράς των παικτών με ταυτόχρονη μεταβολή και των δύο συντελεστών της συνάρτησης.

Στη πράξη όμως διαπιστώθηκε ότι όλες οι προαναφερόμενες αλλαγές αύξησαν μεν την πολυπλοκότητα του μοντέλου και την πιστότητα της προσομοίωσης, αλλά δημιούργησαν σοβαρά προβλήματα στην ταχύτητα σύγκλισης των εξαγομένων τιμών σε όλα τα παίγνια που διεξήχθησαν. Παράλληλα τα αποτελέσματα που προέκυψαν συγκρινόμενα με αυτά της προτέρας κατάστασης δεν παρουσίαζαν διαφορές ως προς τις τελικές τιμές σύγκλισης του μοντέλου αλλά ούτε και ως προς τις γενικές γραμμές του τρόπου συμπεριφοράς των παικτών. Η μόνη διαφοροποίηση αφορούσε στο χρόνο στο οποίο αυτές καθίστατο εμφανείς και αναγνωρίσιμες, ο οποίος ήταν σαφώς μεγαλύτερος καθώς δινόταν στους παίκτες μεγαλύτερη δυνατότητα επιλογών με αποτέλεσμα το μοντέλο να καθυστερεί σημαντικά να συγκλίνει. Με άλλα λόγια το σύστημα έφτανε στον ίδιο προορισμό με τον ίδιο ποιοτικό τρόπο αλλά μέσω μιας πολύ μακρύτερης διαδρομής. Για το λόγο αυτό οι αλλαγές που δοκιμάστηκαν δεν ενσωματώθηκαν στο τελικό λογισμικό καθώς αυτό θα επιβαρυνόταν με τη χρήση ενός πολύ πιο πολύπλοκου μοντέλου χωρίς να υπάρχει άμεση συνέπεια την εξαγωγή αποτελεσμάτων μεγαλύτερης ακρίβειας και αξιοπιστίας.

7.3. Αξιοποίηση και Μελλοντικές Επεκτάσεις της Έρευνας

Το μοντέλο που αναπτύχθηκε έχει περισσότερο τον χαρακτήρα του θεωρητικού εργαλείου και όχι ενός μοντέλου ακριβούς προσομοίωσης της λειτουργίας μιας αγοράς ηλεκτρικής ενέργειας. Με άλλα λόγια μέσα από το συγκεκριμένο μοντέλο και το λογισμικό που αναπτύχθηκε μπορεί κανείς να μελετήσει την επίπτωση των διαφόρων μεταβολών στο σχεδιασμό, στους κανόνες, στον τρόπο λειτουργίας ή στην σύνθεση τέτοιων αγορών, τόσο πάνω στην τιμή όσο και στην κατανομή των πωλήσεων και στον τρόπο συμπεριφοράς των συμμετεχόντων και λιγότερο στο να εισάγει πραγματικά στοιχεία κόστους συγκεκριμένων παραγωγών προκειμένου να προβλέψει την τιμή του Συστήματος.

Το μοντέλο στη μορφή που αναπτύχθηκε θα μπορούσε να αποτελέσει ένα χρήσιμο εργαλείο προσομοίωσης και ελέγχου κανόνων λειτουργίας και ανάλυσης απελευθερωμένων αγορών ηλεκτρικής ενέργειας και να χρησιμοποιηθεί από κυβερνητικούς οργανισμούς ή αρμόδια υπουργεία, ρυθμιστικές αρχές για σχεδιασμό ή ανασχεδιασμό της αγοράς καθώς και από επιχειρήσεις παραγωγής ή/και προμήθειας ηλεκτρικής ενέργειας ώστε να μπορέσουν να μελετήσουν προσαρμόσουν τη στρατηγική τους στο νέο ανταγωνιστικό περιβάλλον που δημιουργείται.

Μια περαιτέρω επέκτασή του θα μπορούσε κάλλιστα να είναι η μοντελοποίηση μιας συγκεκριμένης αγοράς ενέργειας με ρεαλιστική ακρίβεια ή η παραμετροποίηση του λογισμικού ώστε να μπορεί να μοντελοποιήσει οποιαδήποτε αγορά ενέργειας με τρόπο που θα υπήρχε η δυνατότητα για εισαγωγή πραγματικών δεδομένων των συμμετεχόντων, όπως για παράδειγμα συναρτήσεις κόστους παραγωγών, τεχνικά ελάχιστα, περιορισμούς σχετικά με το χρόνο δυνατότητας λειτουργίας των μονάδων ανά τεχνολογία παραγωγής αλλά και συνθήκες και τεχνικά χαρακτηριστικά μιας αγοράς ηλεκτρικής ενέργειας όπως ανάγκες για πρωτεύουσα ή δευτερεύουσα εφεδρεία, περιορισμούς ευστάθειας ή δυναμικότητας του συστήματος μεταφοράς και των διασυνδέσεων με άλλα συστήματα.

Η δυνατότητας επίσης αλλαγών των κανόνων της αγοράς θα μπορούσε να αποτελέσει μια περαιτέρω εξέλιξη του μοντέλου. Για παράδειγμα η εισαγωγή δυνατότητα επιλογής ενός υβριδικού συστήματος τιμολόγησης όπου οι προσφορές των παικτών μέχρι ενός επιπέδου τιμών (Soft Cap) να εκκαθαρίζονται με την μέθοδο Uniform Pricing ενώ από το σημείο αυτό και πάνω με τη μέθοδο Pay-As-Bid. Η επίδραση της μεθόδου αυτής στις τιμές θα μπορούσε να αξιολογηθεί για διαφορετικό τύπο αγοράς (ολιγοπώλιο ως τέλειο ανταγωνισμό) σε σχέση με τις άλλες δύο και επιπλέον θα μπορούσε να προσδιορισθεί και το βέλτιστο επίπεδο της τιμής στο οποίο αλλάζει η μέθοδος τιμολόγησης.

Τέλος η εξεύρεση και η σύγκρισή των θεωρητικών (αναλυτικών) λύσεων παιγνίων με περισσότερους από 3 παίκτες (μέχρι είκοσι) με τα αντίστοιχα πειραματικά αποτελέσματα που προέκυψαν εδώ, θα είχε ενδιαφέρον προκειμένου να διαπιστωθεί όχι μόνο αν οι πειραματικές λύσεις παραμένουν εντός της περιοχής των σημείων ισορροπίας Nash (που είναι το πιθανότερο) αλλά κυρίως για το αν μετατοπίζονται εντός αυτής και με ποιόν τρόπο.

7.4. Επιστημονικές Δημοσιεύσεις και Ανακοινώσεις

Οι μεθοδολογίες που αναπτύχθηκαν και συμπεράσματα που προέκυψαν κατά τη διάρκεια εκπόνησης της παρούσας διατριβής αποτέλεσαν το υλικό για δημοσιεύσεις σε διεθνή επιστημονικά περιοδικά και ανακοινώσεις σε συνέδρια.

Δημοσιεύσεις σε διεθνή επιστημονικά περιοδικά κατόπιν κρίσης.

1. Christos C. Skoulidas, Costas D. Vournas, George P. Papavasilopoulos, "[An Adaptive Learning Game Model for Interacting Electric Power Markets](#)", **INFOR: Information Systems and Operational Research**, vol. 48, pp. 261-266, University of Toronto Press, 2010.
2. Skoulidas,C., Vournas,C., Papavasilopoulos,G., "[Adaptive Game Modelling of Deregulated Power Markets](#)", PES Letters, **IEEE Power Engineering Review**, Vol. 22, Issue 9, pp. 11-11, Sept. 2002.

Ανακοινώσεις σε επιστημονικά συνέδρια:

3. Skoulidas,C., Vournas,C., Papavasilopoulos,G., "[Adaptive Learning Games in Interconnected Power Markets](#)", 1st Scientific Congress of the Electrical and Mechanical Engineers Association, Athens, March 2005.
4. Skoulidas.C., Vournas,C., Papavasilopoulos,G., "[An Adaptive Game for Pay-as-Bid and Uniform Pricing Power Pools Comparison](#)", MED POWER 2002 Conference, IEE Greece, Cyprus and Israel, Athens, Nov.2002.
5. Skoulidas,C., Vournas,C., Papavasilopoulos,G., "[An Adaptive Game for Electricity Markets](#)", National Conference KTISIVIOS, Santorini 2001.

7.5. Βιβλιογραφία

- [1] Hogan H.William, "Competitive Electricity Market Design: A Wholesale Primer", Center for Business and Government John F. Kennedy School of Government, Harvard University, Cambridge, 1998.
- [2] Foley, A. M., et al. "A strategic review of electricity systems models." *Energy* 35.12 (2010): 4522-4530.
- [3] Glachant, Jean-Michel, ed. *Electricity reform in Europe: Towards a single energy market*. Edward Elgar Publishing, 2009.
- [4] Helm, Dieter. "The European framework for energy and climate policies." *Energy Policy* 64, 2014, pp 29-35.
- [5] M. Ilic, F. Galiana, L. Fink, "Power System Restructuring", Kluwer AP, 1998
- [6] L. H. Fink, C. D. Vournas (editors), "Bulk Power Systems Dynamics and Control IV – Restructuring", Symposium Proceedings, Santorini, Greece, Aug. 1998.
- [7] F. C. Schweppe, *Spot pricing of electricity*", Kluwer AP, 1988
- [8] Wolak, F.A., Patrick, R.H., 1996. "The Impact of Market Rules and Market Structure on the Price Determination Process in the England and Wales Electricity Market". Stanford University, Palo Alto.
- [9] Nielsen, Steffen, Peter Sorknæs, and Poul Alberg Østergaard. "Electricity market auction settings in a future Danish electricity system with a high penetration of renewable energy sources—A comparison of marginal pricing and pay-as-bid." *Energy* 36.7 (2011): 4434-4444.
- [10] Dahl, Carol. *International Energy Markets: Understanding Pricing, Policies, & Profits*. PennWell Books, 2015.
- [11] Inigo Herguera, "Bilateral contracts and the spot market for electricity: some observations on the British and the NordPool experiences", *Utilities Policy* 9 (2000), pp.73–80.
- [12] Wolfram, C., "Strategic Bidding in a Multi-Unit Auction: An Empirical Analysis of Bids to Supply Electricity in England and Wales" *RAND Journal of Economics*, 29(4), 703-725, 1998.
- [13] Borenstein, S., J. Bushnell, C. Knittel, and C. Wolfram, "Price Convergence in California's Deregulated Wholesale Electricity Market", *Proceedings of the University of California Energy Institute Conference*, March, 2000.
- [14] Joskow, P. and E. Kahn, "A Quantitative Analysis of Pricing Behavior in California's Wholesale Electricity Market During Summer 2000," NBER Working Paper 8157, March 2001.
- [15] A.L. Motto, F.D. Galiana, "Equilibrium of auction markets with unit commitment: the need for augmented pricing", *IEEE Transactions on Power Systems*, 2002, 17, 3, 798
- [16] Conejo, A.J., Nogales, F.J. and Arroyo, J.M., "Price-taker bidding strategy under price uncertainty", *IEEE Transactions on Power Systems*, 17(4), 2002, pp.1081-1088.
- [17] Veit, Daniel J., Anke Weidlich, and Jacob A. Krafft. "An agent-based analysis of the German electricity market with transmission capacity constraints." *Energy Policy* 37.10 (2009): 4132-4144.
- [18] Baillo, A., Ventosa, M., Rivier, M. and Ramos, A., "Optimal offering strategies for generation companies operating in electricity spot markets", *IEEE Transactions on Power Systems*, 19 (2), 2004, pp.745-753.
- [19] Singh, Harry. "Introduction to game theory and its application in electric power markets." *IEEE Computer Applications in Power* 12, no. 4 (1999): 18-20.
- [20] Xiaomin Bai, Shahidehpour, S.M. Ramesh, V.C and Erkeng Yu, "Transactions Analysis by Nash Game Method" *Ibid* Vol. 12, No 3, Aug. 1997, pp 1046- 1052
- [21] Krishna, V. and Ramesh, V.C., "Intelligent Agents for Negotiations in Market Games, Parts I and II", *IEEE Transactions on Power Systems*, Vol 13, No 3 August 1998, pp 1103-1114.
- [22] Ferrero, R.W., Rivera, J.F. and Shahidehpour, S.M. "Application of Games with Incomplete Information for Pricing Electricity in Deregulated Power Pools" *IEEE Transactions on Power Systems*, Vol 13, No 1, Aug 1998, pp 184-189.
- [23] Torre S., Contreras J., Conejo A.J., "Finding Multiperiod Nash Equilibria in Pool-Based Electricity Markets", *IEEE Transactions on Power Systems*, Vol 19, No 1, February 2004, pp 643-651.
- [24] Ferrero, R.W. Shahidehpour, S.M. and Ramesh, V.C. "Transactions Analysis in Deregulated Power Systems Using Game Theory" *Ibid* Vol 12, No 13, Aug 1997, p 1340-1357.
- [25] Singh, Harry, Shangyou Hao, and Alex Papalexopoulos. "Transmission congestion management in competitive electricity markets." *IEEE Transactions on power systems* 13, no. 2 (1998): 672-680.

- [26] Park, Jong-Bae, Balho H. Kim, Jin-Ho Kim, Man-Ho Jung, and Jong-Keun Park. "A continuous strategy game for power transactions analysis in competitive electricity markets." *IEEE transactions on power systems* 16, no. 4 (2001): 847-855.
- [27] Villar, Jorge, and Hugh Rudnick. "Hydrothermal market simulator using game theory: assessment of market power." *IEEE Transactions on Power Systems* 18, no. 1 (2003): 91-98.
- [28] Stoft, Steven. "Using game theory to study market power in simple networks." *IEEE Tutorial on Game Theory in Electric Power Markets* (1999): 33-40.
- [29] Blake, Michael. "Game theory and electricity markets." *Drayton Analytics research paper series* (2003).
- [30] Wang, Jianhui, Zhi Zhou, and Audun Botterud. "An evolutionary game approach to analyzing bidding strategies in electricity markets with elastic demand." *Energy* 36.5 (2011): 3459-3467.
- [31] Wang, Yunpeng, et al. "A game-theoretic approach to energy trading in the smart grid." *IEEE Transactions on Smart Grid* 5.3 (2014): 1439-1450.
- [32] Sueyoshi, Toshiyuki. "An agent-based approach equipped with game theory: strategic collaboration among learning agents during a dynamic market change in the California electricity crisis." *Energy Economics* 32.5 (2010): 1009-1024.
- [33] Basar, T. and Olsder G.J. *Dynamic Noncooperative Game Theory*, Academic Press New York, New York 1982.
- [34] Nash, John. "Non-cooperative games." *Annals of mathematics* (1951): 286-295.
- [35] Fudenberg, D and Tirole, J, *Game Theory*, MIT Press, 1998
- [36] Von Stackelberg, Heinrich. *The theory of the market economy*. Oxford University Press, 1952.
- [37] Li, Gong, Jing Shi, and Xiuli Qu. "Modeling methods for GenCo bidding strategy optimization in the liberalized electricity spot market—A state-of-the-art review." *Energy* 36.8 (2011): 4686-4700.
- [38] Kwon, Roy H., and Daniel Frances. "Optimization-based bidding in day-ahead electricity auction markets: A review of models for power producers" *Handbook of Networks in Power Systems I*. Springer Berlin Heidelberg, 2012. 41-59.
- [39] J. D. Weber, T. J. Overbye, P. W. Sauer, "Simulation of Electricity Markets with Player Bidding", in [1] pp. 331-339
- [40] Gabriel, Steven A., et al. *Complementarity modeling in energy markets*. Vol. 180. Springer Science & Business Media, 2012.
- [41] Wolfram, Catherine D. *Strategic bidding in a multi-unit auction: An empirical analysis of bids to supply electricity*. No. w6269. National Bureau of Economic Research, 1997.
- [42] Soleymani, S. "Bidding strategy of generation companies using PSO combined with SA method in the pay as bid markets." *International Journal of Electrical Power & Energy Systems* 33.7 (2011): 1272-1278.
- [43] Banal-Estanol, Albert, and Augusto Rupérez Micola. "Behavioural simulations in spot electricity markets." *European Journal of Operational Research* 214.1 (2011), pp.147-159.
- [44] de la Torre S., A.J.Conejo, J.Contreras, "Simulating oligopolistic pool-based electricity markets: a multiperiod approach", *IEEE Transactions on Power Systems*, Vol. 18, Issue 4, pp. 1547 – 1555, 2003.
- [45] de la Torre, Sebastián, José M. Arroyo, Antonio J. Conejo, and Javier Contreras. "Price maker self-scheduling in a pool-based electricity market: a mixed-integer LP approach." *IEEE Transactions on Power Systems* 17, no. 4 (2002): 1037-1042.
- [46] J.H.Grobman, J.M.Carey, "Price caps and investment: long-run effects in the electric generation industry", *Energy Policy* 29 (2001), pp.545-552.
- [47] Borenstein, S., 1999. "Understanding Competitive Pricing and Market Power in Wholesale Electricity Markets", University of California Energy Institute, Berkeley.
- [48] Conejo, A.J., Contreras, J., Arroyo, J.M. and De la Torre, S., "Optimal response of an oligopolistic generating company to a competitive pool-based electric power market", *IEEE transactions on power systems*, 17 (2), 2002, pp.424-430.
- [49] Lakshmivarahan, S. *Learning Algorithms: Theory and Applications*, Springer Verlag, New York, New York, 1981
- [50] Papavassilopoulos, G.P. *Learning Algorithms for Repeated Bitmatrix Games with Incomplete Information JOTA*, Vol. 62, No. 3, September 1989, pp. 467-488.

- [51] Papavassilopoulos, G.P. "Iterative Techniques for the Nash Solution in Quadratic Games with Unknown Parameters," *SIAM Journal on Optimization and Control*, Vol. 24, No. 4, July 1986, pp. 821-834.
- [52] Papavassilopoulos, G.P. "Adaptive Games," in *Stochastic Processes in Physics and Engineering*, S. Albererio, Ph. Blanchard, L. Streit and M. Hazenwinkel, eds., Reidal Publishing Co., 1987, pp. 223-236.
- [53] Pinto, Tiago, et al. "Adaptive learning in agents behaviour: A framework for electricity markets simulation." *Integrated Computer-Aided Engineering* 21.4 (2014): 399-415.
- [54] Rahimiyan, Morteza, and Habib Rajabi Mashhadi. "An Adaptive-Learning Algorithm Developed for Agent-Based Computational Modeling of Electricity Market." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.5 (2010): 547-556.
- [55] Skoulidas, Ch., Vournas, C., Papavassilopoulos, G.P., "An Adaptive Games for Electricity Markets" in *National Conference KTISIVIOS, Santorini, Greece, June 2001*.
- [56] Skoulidas C.C., Vournas C.D., Papavassilopoulos G.P., (2002), "Adaptive Game Modeling of Deregulated Power Markets", *IEEE Power Engineering Review, PES Letters*, pp. 42-45.
- [57] Skoulidas C.C., Vournas C.D., Papavassilopoulos G.P., (2002), "An Adaptive Game for Pay-As-Bid and Uniform Pricing Power Pools Comparison", *MED POWER 2002 IEE Conference*, Athens, Greece.
- [58] Christos C. Skoulidas, Costas D. Vournas, George P. Papavassilopoulos, "An Adaptive Learning Game Model for Interacting Electric Power Markets", *INFOR: Information Systems and Operational Research*, vol. 48, pp. 261-266, University of Toronto Press, 2010.
- [59] Haghghat, Hossein, Hossein Seifi, and Ashkan Rahimi Kian. "Pay-as-bid versus marginal pricing: the role of suppliers strategic behavior." *International Journal of Electrical Power & Energy Systems* 42.1 (2012): 350-358.
- [60] Xiong, Gaofeng, Shigeru Okuma, and Hideki Fujita. "Multi-agent based experiments on uniform price and pay-as-bid electricity auction markets." In *Electric Utility Deregulation, Restructuring and Power Technologies, 2004.(DRPT 2004)*. Proceedings of the 2004 IEEE International Conference on, vol. 1, pp. 72-76. IEEE, 2004.
- [61] Ausubel, Lawrence M., and Peter Cramton. "Demand reduction and inefficiency in multi-unit auctions." (2002).
- [62] Nielsen, Steffen, Peter Sorknæs, and Poul Alberg Østergaard. "Electricity market auction settings in a future Danish electricity system with a high penetration of renewable energy sources—A comparison of marginal pricing and pay-as-bid." *Energy* 36.7 (2011): 4434-4444.
- [63] Son, Y.S., Baldick, R., Lee, K.H. and Siddiqi, S., 2004. Short-term electricity market auction game analysis: uniform and pay-as-bid pricing. *IEEE Transactions on Power Systems*, 19(4), pp.1990-1998.
- [64] Federico, Giulio, and David Rahman. "Bidding in an electricity pay-as-bid auction." *Journal of Regulatory Economics* 24, no. 2 (2003): 175-211.
- [65] Cramton, Peter, Alfred E. Kahn, Robert H. Porter, and Richard D. Tabors. "Pricing in the California power exchange electricity market: Should California switch from uniform pricing to pay-as-bid pricing?." (2001).
- [66] David, A. Kumar, and Fushuan Wen. "Strategic bidding in competitive electricity markets: a literature survey." In *Power Engineering Society Summer Meeting, 2000*. IEEE, vol. 4, pp. 2168-2173. IEEE, 2000.
- [67] L.Chen, Y.Kinoshita, Ge-Ril, R. Yokoyama, "Pricing Structure of Market-Based Power systems by Game Theoretic Analysis", in [1], pp. 495-501.
- [68] Tsikalakis, A.G. and Hatziaargyriou, N.D., 2011. Operation of microgrids with demand side bidding and continuity of supply for critical loads. *European Transactions on Electrical Power*, 21(2), pp.1238-1254.
- [69] Borghetti, A., Gross, G. and Nucci, C.A., "Auctions with explicit demand-side bidding in competitive electricity markets", *The Next Generation of Electric Power Unit Commitment Models* (pp. 53-74), 2002, Springer US.
- [70] David, A. Kumar, and Fushuan Wen. "Strategic bidding in competitive electricity markets: a literature survey.", *Power Engineering Society Summer Meeting, 2000*. IEEE, vol. 4, pp. 2168-2173. IEEE, 2000.
- [71] Wen, Fushuan, and A. Kumar David. "Optimal bidding strategies and modeling of imperfect information among competitive generators." *IEEE transactions on power systems* 16.1 (2001): 15-21.

7.6. Citations

Για την δημοσίευση [1] της παραγράφου 7.4 ανωτέρω, υπάρχουν οι ακόλουθες παραπομπές:

- [1] G.Conzelmann, G.Boyd, V.Koritarov, T.Veselka, "Multi-agent power market simulation using EMCAS", IEEE Power Engineering Society General Meeting, 2005.
- [2] G.Conzelmann, M.North, G.Boyd, R.Cirillo, V.Koritarov, C.Macal, Prakash Thimmapuram, T.Veselka, "Simulating strategic market behavior using an agent-based modeling approach", 6th IAEE European Energy Conference on Modeling in Energy Economics and Policy, Zurich 2004.
- [3] Miltiadis Alamaniotis, Rong Gao, Lefteri H.Tsoukalas, "Towards an Energy Internet: A Game-Theoretic Approach to Price-Directed Energy Utilization", Series Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecom-munications Engineering, Vol. 54, pp 3-11, Springer Berlin Heidelberg, ISBN 978-3-642-19321-7, 2011.
- [4] S.F.Ghaderi, A.Azadeh, B.Pourvalikhan Nokhandan, E.Fathi, "Behavioral simulation and optimization of generation companies in electricity markets by fuzzy cognitive map", Expert Systems with Applications, Vol.39, Issue 5, pp. 4635–4646 ELSEVIER, April 2012.
- [5] Allen G.Morinec ; F.Eugenio Villaseca, "Optimal generator bidding strategies for power and ancillary services using game theory", Browse Conferences, 40th North American Power Symposium, 2008.
- [6] A. Azadeh, S.F.Ghadrei, B.Pourvalikhan Nokhandan, "GENCO behavior model and simulation in electricity market by FCM-approach", 2009 IEEE Workshop on Hybrid Intelligent Models and Applications.
- [7] Zhao, Liang, "Multi-area network analysis", PhD Dissertation, Texas A&M University, 2004.
- [8] N.Kakogiannis, P.Kontogiorgos, E.Sarri, G.P.Papavassilopoulos, "Multicriteria energy policy investments and energy market clearance via integer programming", Central European Journal of Operations Research, Vol 24, Issue 3, pp 515–534. September 2016, ISSN 1435-246X.
- [9] Skoulidas, Christos C., Costas D. Vournas, and George P. Papavassilopoulos. "An adaptive learning game model for interacting electric power markets." Infor 48.4 (2010): 261.
- [10] D.D.P.Pacaba and A.C.Nerves, "Multiagent-based market simulator for the wholesale electricity spot market" TENCON 2012, IEEE Region 10 Conference, Cebu, 2012, pp. 1-6.
- [11] D.Kiose, V.Voudouris, "The ACEWEM framework: An integrated agent-based and statistical modelling laboratory for repeated power auctions", Expert Systems with Applications, Volume 42, Issue 5, Elsevier, 1 April 2015, Pages 2731–2748, ISSN 09574174.
- [12] Sandra M. Londoño H., Carlos A.Lozano, Gladys Caicedo Delgado, "Estabilidad del precio en el mercado de electricidad colombiano", INGENIERÍA & DESARROLLO, Revista de la División de Ingenierías de la Universidad del Norte, No 21, June 2007, ISSN: 0122-3461.
- [13] Sandra M. Londoño H., Carlos A.Lozano, Gladys Caicedo Delgado, "Pronóstico de precio en el mercado de electricidad colombiano usando redes neuronales", Universidad del Valle, Energía y Computación Vol. 13, No.1, 2005.
- [14] Shahrjerdi, Reza; Anuar, Mohd Khairol; Mustapha, F; Ismail, N; Esmaili, "Equilibrium and non-equilibrium models of the power markets", M. African Journal of Business Management 6.4 (Feb 1, 2012): 1614-162.
- [15] I.Kordonis, G.P.Papavassilopoulos, "Cheating in adaptive games motivated by electricity markets", 6th International Symposium on Communications, Control and Signal Processing (ISCCSP) 2014.
- [16] Ritchie, Adam Caine. "Viability of an Automated Meter Reading (AMR) System for full Scale Implementation within Western Australia's (WA's) South West Interconnected Network (SWIN)", 2003.
- [17] California Energy Commission, Lawrence Berkeley National Laboratory, "WECC CAISO Specific needs for loop flow monitoring management, near term prediction and probabilistic assessment, and prototype monitoring system design", PIER Consultant Report, Nov.2005, CEC-500-2005-168.
- [18] Lascano Farak, Sheila Katherine."Tesis de Doctorados Repositorio Dspace/Manakin" (2007).
- [19] GRIGORAȘ, Gheorghe, and Gheorghe CÂRȚINĂ. "DEREGULATED MARKET SIMULATION IN EDUCATIONAL PROCESS SIMULAREA PIEȚII DEREGLEMENTATE ÎN PROCESUL DE EDUCAȚIE."
- [20] 謝宗煌. "對局理論應用於解制電力市場之獨立發電業者機組維修策略." Journal of China Institute of Technology 34 (2006).
- [21] Londoño, S.M., Londoño, M., Lozano, C.A. and Delgado, G.C., 2007. "Price stability in the Colombian electricity market". Ingeniería y Desarrollo, (21).

- [22] Ηλιόπουλος, Ανδρέας "Προσαρμοστικά Παιγνία Σε Απελευθερωμένη Αγορά Ενέργειας Με Χρήση Μαθησιακού Αλγορίθμου." (2010), Διπλωματική εργασία, Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, ΕΜΠ.

Για την δημοσίευση [2] οι ακόλουθες παραπομπές:

- [23] Xin, Baogui, and Yuting Li. "Bifurcation and chaos in a price game of irrigation water in a coastal irrigation district." *Discrete Dynamics in Nature and Society* 2013 (2013).
- [24] Kakogiannis, N., et al. "Multicriteria energy policy investments and energy market clearance via integer programming." *Central European Journal of Operations Research* 24.3 (2016), pp.515-534.

Για την ανακοίνωση σε συνέδριο [4] οι ακόλουθες παραπομπές:

- [25] Alamaniotis, Miltiadis, Rong Gao, and Lefteri H. Tsoukalas. "Towards an energy internet: a game-theoretic approach to price-directed energy utilization." *International Conference on Energy-Efficient Computing and Networking*. Springer Berlin Heidelberg, 2010.
- [26] Mohammadi, Mohammad, and Hossein Nasiraghdam. "Particle swarm optimization of wind farm due to non-greenhouse gas emission under power market considering uncertainty of wind speed using Monte Carlo method." *Tehnički vjesnik* 22.1 (2015), pp. 79-85.
- [27] Skoulidas, Christos C., Costas D. Vournas, and George P. Papavassilopoulos. "An adaptive learning game model for interacting electric power markets." *Infor* 48.4 (2010), pp. 261.
- [28] Mohammadi, Mohammad. "GA-based optimal sizing of PV under pay as bid and uniform power market pricing considering uncertainty of solar radiation." *Tehnički vjesnik* 20.3 (2013), pp. 511-518.
- [29] Mousavi, Seyed Hosein, Ali Nazemi, and Ashkan Hafezalkotob. "Nash equilibrium strategy in the deregulated power industry and comparing its lost welfare with Iran wholesale electricity market." *Journal of Industrial Engineering International* 12.4 (2016), pp. 421-435.
- [30] Garavand, Saman Abasi, A. Mohammadi Rozbahani, and M. Khodaei. "Techno-economic analysis of utilization photovoltaic cells in distribution system with considering effects of carbon price variation for environmental protection purpose." *Advances in Environmental Biology* (2014): 704-710.
- [31] Mohammadi, M., et al. "Micro grid optimization as grid connected in pool based power market under pay as bid and uniform pricing." *International Review of Electrical Engineering (IREE)* 7.2 (2012).

Παράρτημα

Ο κώδικας του Λογισμικού EM Simulator

(Α' μέρος: το παίγνιο του μοντέλου της αγοράς)

- *asyn_basic_m.f90*
- *asyn_m.f90*
- *asyn_one_sys_m.f90*
- *asyn_two_sys_oneoffer_m.f90*
- *asyn_two_sys_split_m.f90*
- *main.f90*

```

1  !=====
2  !+ <A one line description of this module>
3  !=====
4  MODULE asyn_basic_m
5  !.....
6  !
7  ! Description:
8  !   <Say what this Module is for>
9  !
10 ! Current code owner : C.Skoulidas
11 !
12 ! History:
13 ! Version      Date      Comment
14 ! -----      ----      -
15 ! <18.1>      <2011>    Original code  <C.Skoulidas>
16 !
17 ! Code description:
18 ! Language: Fortran 90.
19 !.....
20 ! Declarations:
21 !
22 ! Modules Used:
23 !
24 ! Imported type definitions
25 ! Imported Parameters:
26 ! Imported scalar variables with intent (in):
27 ! Imported scalar variables with intent (out):
28 ! Imported Array variables with intent(in):
29 ! Imported Array variables with intent(out):
30 ! Imported routines:
31 !
32     IMPLICIT NONE
33 !
34 ! Global (Public) Declarations:
35 !
36     public :: gameopt_t
37     public :: player_t
38     public :: asyn_t
39     public :: remove_player
40     public :: erase
41     public :: copy
42     public :: realloc
43     public :: CATEGORY
44     public :: ASYN_SUCCESS
45     public :: ASYN_NO_MEM
46     public :: ASYN_IO_ERROR
47     public :: ERR_DEMAND
48     public :: ERR_DEMAND_CON
49     public :: ERR_SOLVE_LOW
50     public :: ERR_SOLVE_UP
51     public :: NGAMES_LB
52     public :: NGAMES_UB
53     public :: NITERATIONS_LB
54     public :: NITERATIONS_UB
55     public :: INID_LB
56     public :: INID_UB
57     public :: INID_FACTOR_LB
58     public :: INID_FACTOR_UB
59     public :: DD_LB
60     public :: DD_UB
61     public :: PRICECAP_LB
62     public :: PRICECAP_UB
63     public :: INTERCONNE_LB
64     public :: INTERCONNE_UB
65     public :: PERIOD_MIN_LB
66     public :: PERIOD_MIN_UB
67     public :: PERIOD_MAX_LB
68     public :: PERIOD_MAX_UB
69     public :: ESTEP_DECR_LB
70     public :: ESTEP_DECR_UB
71     public :: THSTEP_DECR_LB
72     public :: THSTEP_DECR_UB
73     public :: RND_LB
74     public :: RND_UB
75     public :: NCONST
76     public :: NCATEG
77     public :: UB_LEN_NAME
78     public :: UB_LEN_CATEG
79     public :: FC_UB
80     public :: FC_LB

```

```

81  public :: A_UB
82  public :: A_LB
83  public :: B_UB
84  public :: B_LB
85  public :: XMIN_UB
86  public :: XMIN_LB
87  public :: XMAX_UB
88  public :: XMAX_LB
89  public :: THETA_INI_LB
90  public :: THETA_INI_UB
91  public :: ALPHA_INI_LB
92  public :: ALPHA_INI_UB
93  public :: ALPHA_TOLERANCE_LB
94  public :: ALPHA_TOLERANCE_UB
95  public :: E_INIT_UB
96  public :: E_INIT_LB
97  public :: INIPI_UB
98  public :: INIPI_LB
99  public :: INIPD_UB
100 public :: INIPD_LB
101 public :: INIPC_LB
102 public :: INIPC_UB
103 public :: PERIOD_LB
104 public :: PERIOD_UB
105 public :: OFFER_ALL
106 public :: OFFER_SPLIT
107
108 !
109 public :: CHP_CAPACITY_LB, CHP_CAPACITY_UB
110 public :: CHP_COUNT_LB, CHP_COUNT_UB
111 public :: CHP_SAVINGS_LB, CHP_SAVINGS_UB
112 !
113 public :: FCELLS_CAPACITY_LB, FCELLS_CAPACITY_UB
114 public :: FCELLS_COUNT_LB, FCELLS_COUNT_UB
115 public :: FCELLS_PLUSCOST_LB, FCELLS_PLUSCOST_UB
116 !
117 public :: SMHYDRO_CAPACITY_LB, SMHYDRO_CAPACITY_UB
118 public :: SMHYDRO_COUNT_LB, SMHYDRO_COUNT_UB
119 public :: SMHYDRO_AVAILABILITY_LB, SMHYDRO_AVAILABILITY_UB
120 !
121 public :: WIND_CAPACITY_LB, WIND_CAPACITY_UB
122 public :: WIND_COUNT_LB, WIND_COUNT_UB
123 public :: WIND_NO_PRODUCTION_LB, WIND_NO_PRODUCTION_UB
124 public :: WIND_MAX_PRODUCTION_LB, WIND_MAX_PRODUCTION_UB
125 !
126 public :: PV_CAPACITY_LB, PV_CAPACITY_UB
127 public :: PV_COUNT_LB, PV_COUNT_UB
128 public :: PV_NO_PRODUCTION_LB, PV_NO_PRODUCTION_UB
129 public :: PV_MAX_PRODUCTION_LB, PV_MAX_PRODUCTION_UB
130
131 !
132 public :: ASSURANCE_COST_LB, ASSURANCE_COST_UB
133 !
134 public :: TOTAL_CHP_CAPACITY_LB, TOTAL_CHP_CAPACITY_UB
135 public :: TOTAL_FCELLS_CAPACITY_LB, TOTAL_FCELLS_CAPACITY_UB
136 public :: TOTAL_SMHYDRO_CAPACITY_LB, TOTAL_SMHYDRO_CAPACITY_UB
137 public :: TOTAL_INTERMITTENT_CAPACITY_LB, TOTAL_INTERMITTENT_CAPACITY_UB
138 public :: TOTAL_DER_CAPACITY_LB, TOTAL_DER_CAPACITY_UB
139 !
140 public :: INTERRUPT_CAPACITY_LB, INTERRUPT_CAPACITY_UB
141 public :: CURVE_SLOPE_LB, CURVE_SLOPE_UB
142 public :: INTERRUPT_HIGH_PRICE_LB, INTERRUPT_HIGH_PRICE_UB
143 public :: INTERRUPT_LOW_PRICE_LB, INTERRUPT_LOW_PRICE_UB
144 public :: total_dsb_dergen_capacity_LB, total_dsb_dergen_capacity_UB
145 !
146 integer(4), parameter :: NCONST = 2
147 integer(4), parameter :: UB_LEN_NAME = 10
148 integer(4), parameter :: UB_LEN_CATEG = 9
149 integer(4), parameter :: NCATEG = 11
150 !
151 ! Global Type Definitions:
152 !
153 type player_t
154 !   If someone add pointers, he/she should care for realloc function
155 integer(4) :: fc
156 integer(4) :: nextit
157 real(8) :: a
158 real(8) :: b
159 real(8) :: xmin
160 real(8) :: xmax
161 real(8) :: theta_ini

```



```

162     character(len=UB_LEN_NAME) :: name
163     character(len=UB_LEN_CATEG) :: category
164     integer(4), dimension (NCONST) :: period
165     real(8), dimension (NCONST) :: e_init
166     real(8), dimension (NCONST) :: inipi
167     real(8), dimension (NCONST) :: inipd
168     real(8), dimension (NCONST) :: inipc
169     real(8) :: sys2_a
170     real(8) :: alpha_ini
171     real(8) :: alpha_tolerance
172     real(8) :: chp_capacity
173     integer(4) :: chp_count
174     real(8) :: chp_savings
175     real(8) :: fcells_capacity
176     integer(4) :: fcells_count
177     real(8) :: fcells_pluscost
178 !
179     real(8) :: smhydro_capacity
180     integer(4) :: smhydro_count
181     real(8) :: smhydro_availability
182 !
183     real(8) :: wind_capacity
184     integer(4) :: wind_count
185     real(8) :: wind_no_production
186     real(8) :: wind_max_production
187     real(8) :: wind_estim_capacity
188 !
189     real(8) :: pv_capacity
190     integer(4) :: pv_count
191     real(8) :: pv_no_production
192     real(8) :: pv_max_production
193     real(8) :: pv_estim_capacity
194     integer(4) :: share_profit
195 !
196     real(8) :: assurance_cost
197 !
198     real(8) :: total_chp_capacity
199     real(8) :: total_fcells_capacity
200     real(8) :: total_smhydro_capacity
201     real(8) :: total_intermittent_capacity
202     real(8) :: total_der_capacity
203 !
204         real(8) :: interrup_capacity
205         real(8) :: curve_slope
206         integer(4) :: price_range
207         real(8) :: interrup_high_price
208         real(8) :: interrup_low_price
209         real(8) :: total_dsb_dergen_capacity
210     real(8) :: total_dsb_capacity
211 !
212 end type
213 !
214 type asyn_t
215     integer(4) :: nplayers
216     integer(4) :: niterations
217     integer(4) :: period_min
218     integer(4) :: period_max
219     integer(4) :: p_flag
220     integer(4) :: pr_method
221     integer(4) :: ngames
222     integer(4) :: p_linear
223     integer(4) :: version
224     integer(4) :: nplayers_der
225     integer(4) :: nplayers_dsb
226     real(8) :: inid
227     real(8) :: inid_factor
228     real(8) :: dd
229     real(8) :: pricecap
230     real(8) :: estep_decr
231     real(8) :: thstep_decr
232     real(8) :: rnd_all
233     real(8) :: rnd_fc
234     real(8) :: rnd_a
235     real(8) :: rnd_b
236     type (player_t), pointer, dimension (:) :: player
237     real(4), pointer, dimension (:,:) :: out_one_game
238     real(4), pointer, dimension (:,:) :: out_many_games
239 end type
240 !
241 type gameopt_t
242     integer(4) :: ngames

```

```

243     integer(4) :: nsystems
244     integer(4) :: offer_type
245     real(8)    :: interconne
246     integer(4) :: currentsystem
247     logical(4) :: probability_on
248 end type
249 !
250 ! Global parameters:
251 !
252 ! This Parameters Must be justified !
253
254 character(len = UB_LEN_CATEG), &
255     parameter, &
256 ! dimension(NCATEG) :: CATEGORY = ("Diesel", "GasTurb", "Diesel P" &
257 ! "Steam", "CombCycl", "Cogen", "Hydro",
258 &
259 ! "Wind", "PhV"/)
260     dimension(NCATEG) :: CATEGORY = ("Diesel", "GasTurb", &
261 "CombCycl", "Steam", &
262 "Coal", "DieselP", &
263 "Cogen", "Nuclear", &
264 "RES", "DERaggr", "DSBaggr"/)
265 !
266 integer(4), parameter :: OFFER_ALL = 1
267 integer(4), parameter :: OFFER_SPLIT = 2
268 integer(4), parameter :: ASYN_NO_MEM = -1
269 integer(4), parameter :: ASYN_SUCCESS = 0
270 integer(4), parameter :: ASYN_BAD_DATA = 1
271 integer(4), parameter :: ASYN_IO_ERROR = 2
272 integer(4), parameter :: ERR_DEMAND = -1
273 integer(4), parameter :: ERR_DEMAND_CON = -2
274 integer(4), parameter :: ERR_SOLVE_UP = -3
275 integer(4), parameter :: ERR_SOLVE_LOW = -4
276 integer(4), parameter :: FC_LB = 0
277 integer(4), parameter :: FC_UB = 99999
278 integer(4), parameter :: NEXIT_LB = 0
279 integer(4), parameter :: NEXIT_UB = 20000
280 integer(4), parameter :: NITERATIONS_LB = 1
281 integer(4), parameter :: NITERATIONS_UB = 600000
282 integer(4), parameter :: PERIOD_MIN_LB = 0
283 integer(4), parameter :: PERIOD_MIN_UB = 999
284 integer(4), parameter :: PERIOD_MAX_LB = 0
285 integer(4), parameter :: PERIOD_MAX_UB = 999
286 integer(4), parameter :: NGAMES_LB = 1, NGAMES_UB = 1000
287 real(8), parameter :: A_LB = 0.0
288 real(8), parameter :: A_UB = 9999.0
289 real(8), parameter :: B_LB = 0.0
290 real(8), parameter :: B_UB = 9999.0
291 real(8), parameter :: XMIN_LB = 0.0
292 real(8), parameter :: XMIN_UB = 9999.0
293 real(8), parameter :: XMAX_LB = 0.0
294 real(8), parameter :: XMAX_UB = 9999.0
295 real(8), parameter :: THETA_INI_LB = 0.0
296 real(8), parameter :: THETA_INI_UB = 100.0
297 real(8), parameter :: ALPHA_INI_LB = 0.0
298 real(8), parameter :: ALPHA_INI_UB = 100.0
299 real(8), parameter :: ALPHA_TOLERANCE_LB = 0.0
300 real(8), parameter :: ALPHA_TOLERANCE_UB = 100.0
301 real(8), parameter :: INID_LB = 0.0
302 real(8), parameter :: INID_UB = 1000000.0
303 real(8), parameter :: INID_FACTOR_LB = 0.0
304 real(8), parameter :: INID_FACTOR_UB = 100.0
305 real(8), parameter :: DD_LB = 0.0
306 real(8), parameter :: DD_UB = 1.0
307 real(8), parameter :: PRICECAP_LB = 1.0
308 real(8), parameter :: PRICECAP_UB = 1000.0
309 real(8), parameter :: INTERCONNE_LB = 1.0
310 real(8), parameter :: INTERCONNE_UB = 1000.0
311 real(8), parameter :: ESTEP_DECR_LB = 0.0
312 real(8), parameter :: ESTEP_DECR_UB = 1.0
313 real(8), parameter :: THSTEP_DECR_LB = 0.0
314 real(8), parameter :: THSTEP_DECR_UB = 1.0
315 real(8), parameter :: RND_LB = 0.0
316 real(8), parameter :: RND_UB = 100.0
317 !
318 integer(4), parameter, dimension (NCONST) :: PERIOD_LB = (/0, 0/)
319 integer(4), parameter, dimension (NCONST) :: PERIOD_UB = (/999,999/)
320 real(8), parameter, dimension (NCONST) :: E_INIT_LB = (/0.0, 0.0/)
321 real(8), parameter, dimension (NCONST) :: E_INIT_UB = (/100.0, 100.0/)
322 real(8), parameter, dimension (NCONST) :: INIPI_LB = (/0.0, 0.0/)
323 real(8), parameter, dimension (NCONST) :: INIPI_UB = (/100.0, 100.0/)

```

```

324 real(8), parameter, dimension (NCONST) :: INIPD_LB = (/0.0, 0.0/)
325 real(8), parameter, dimension (NCONST) :: INIPD_UB = (/100.0, 100.0/)
326 real(8), parameter, dimension (NCONST) :: INIPC_LB = (/0.0, 0.0/)
327 real(8), parameter, dimension (NCONST) :: INIPC_UB = (/100.0, 100.0/)
328 !
329 REAL(8), PARAMETER :: CHP_CAPACITY_LB = 0, CHP_CAPACITY_UB = 1000
330 INTEGER(4), PARAMETER :: CHP_COUNT_LB = 0, CHP_COUNT_UB = 1000
331 REAL(8), PARAMETER :: CHP_SAVINGS_LB = 0, CHP_SAVINGS_UB = 100.
332 !
333 REAL(8), PARAMETER :: FCELLS_CAPACITY_LB = 0, FCELLS_CAPACITY_UB = 1000
334 INTEGER(4), PARAMETER :: FCELLS_COUNT_LB = 0, FCELLS_COUNT_UB = 1000
335 REAL(8), PARAMETER :: FCELLS_PLUSCOST_LB = 0, FCELLS_PLUSCOST_UB = 100.
336 !
337 REAL(8), PARAMETER :: SMHYDRO_CAPACITY_LB = 0, SMHYDRO_CAPACITY_UB = 1000
338 INTEGER(4), PARAMETER :: SMHYDRO_COUNT_LB = 0, SMHYDRO_COUNT_UB = 1000
339 REAL(8), PARAMETER :: SMHYDRO_AVAILABILITY_LB = 0, SMHYDRO_AVAILABILITY_UB = 100.
340 !
341 REAL(8), PARAMETER :: WIND_CAPACITY_LB = 0, WIND_CAPACITY_UB = 1000
342 INTEGER(4), PARAMETER :: WIND_COUNT_LB = 0, WIND_COUNT_UB = 1000
343 REAL(8), PARAMETER :: WIND_NO_PRODUCTION_LB = 0, WIND_NO_PRODUCTION_UB = 100.
344 REAL(8), PARAMETER :: WIND_MAX_PRODUCTION_LB = 0, WIND_MAX_PRODUCTION_UB = 100.
345
346 !
347 REAL(8), PARAMETER :: PV_CAPACITY_LB = 0, PV_CAPACITY_UB = 1000
348 INTEGER(4), PARAMETER :: PV_COUNT_LB = 0, PV_COUNT_UB = 1000
349 REAL(8), PARAMETER :: PV_NO_PRODUCTION_LB = 0, PV_NO_PRODUCTION_UB = 100.
350 REAL(8), PARAMETER :: PV_MAX_PRODUCTION_LB = 0, PV_MAX_PRODUCTION_UB = 100.
351
352 !
353 REAL(8), PARAMETER :: ASSURANCE_COST_LB = 0, ASSURANCE_COST_UB = 100.
354 !
355 REAL(8), PARAMETER :: TOTAL_CHP_CAPACITY_LB = 0, TOTAL_CHP_CAPACITY_UB = 1000
356 REAL(8), PARAMETER :: TOTAL_FCELLS_CAPACITY_LB = 0, TOTAL_FCELLS_CAPACITY_UB = 1000
357 REAL(8), PARAMETER :: TOTAL_SMHYDRO_CAPACITY_LB = 0, TOTAL_SMHYDRO_CAPACITY_UB = 1000
358 REAL(8), PARAMETER :: TOTAL_INTERMITTENT_CAPACITY_LB = 0, TOTAL_INTERMITTENT_CAPACITY_UB = 1000
359 REAL(8), PARAMETER :: TOTAL_DER_CAPACITY_LB = 0, TOTAL_DER_CAPACITY_UB = 1000
360 !
361 REAL(8), PARAMETER :: INTERRUP_CAPACITY_LB = 0, INTERRUP_CAPACITY_UB = 1000
362 REAL(8), PARAMETER :: CURVE_SLOPE_LB = 0, CURVE_SLOPE_UB = 1000
363 REAL(8), PARAMETER :: INTERRUP_HIGH_PRICE_LB = 0, INTERRUP_HIGH_PRICE_UB = 1000
364 REAL(8), PARAMETER :: INTERRUP_LOW_PRICE_LB = 0, INTERRUP_LOW_PRICE_UB = 1000
365 REAL(8), PARAMETER :: total_dsb_dergen_capacity_LB = 0, total_dsb_dergen_capacity_UB = 1000
366
367
368
369 !
370 ! Global scalars:
371 ! Global arrays:
372 ! Local (Private) Declarations:
373     private
374 ! Local Type Definitions:
375 ! Local parameters:
376 ! Local scalars:
377 ! Local arrays:
378 ! Operator definitions:
379     interface realloc
380         module procedure realloc_players
381     end interface
382     interface copy
383         module procedure copy_asyn
384     end interface
385     interface erase
386         module procedure erase_asyn
387     end interface
388 !.....
389     CONTAINS
390 !-----
391 ! List Of Internal Procedures:
392 ! Subroutines:
393 !
394 !
395 ! Functions:
396 !
397 !
398 !-----
399 !~~~~~
400 ! Cleans up memory for results
401 !~~~~~
402     SUBROUTINE remove_player (tgameopt, asyn, ipl, ierr)
403 !.....
404 !

```

```

405 ! Description:
406 !   <Say what this subroutine does>
407 !
408 ! Code description:
409 !       Language: Fortran 90.
410 !.....
411 ! Declarations:
412 !
413 ! Subroutine Arguments
414 ! Scalar arguments with intent(in):
415 !     integer(4) :: ipl, ierr
416 !     TYPE (asyn_t) :: asyn
417 !     TYPE (gameopt_t) :: tgameopt
418 ! Array arguments with intent(in):
419 ! Scalar arguments with intent(inout):
420 ! Array arguments with intent(inout):
421 ! Scalar arguments with intent(out):
422 ! Array arguments with intent(out):
423 ! Local parameters:
424 ! Local scalars:
425 !     TYPE (asyn_t) :: dupl
426 ! Local arrays:
427 !     integer(4) :: k, iplayer
428 !.....
429 ! Listing
430 !
431 !     IF (ipl .GE. LBOUND(asyn%player, 1) .AND. ipl .LE. UBOUND(asyn%player, 1)) THEN
432 !         dupl=asyn
433 !         dupl%nplayers=asyn%nplayers-1
434 !         ALLOCATE (dupl%player(dupl%nplayers), STAT=ierr)
435 !         IF (ierr.NE.0) THEN
436 !             ierr = ASYN_NO_MEM
437 !             RETURN
438 !         ENDIF
439 !         k=0
440 !         DO iplayer=LBOUND(asyn%player, 1), UBOUND(asyn%player, 1)
441 !             IF (ipl .NE. iplayer) THEN
442 !                 k=k+1
443 !                 dupl%player(k)=asyn%player(iplayer)
444 !             ENDIF
445 !         ENDDO
446 !         NULLIFY (dupl%out_one_game)
447 !         NULLIFY (dupl%out_many_games)
448 !         CALL erase (asyn, ierr)
449 !         IF (ierr .NE. ASYN_SUCCESS) RETURN
450 !         CALL copy (tgameopt, dupl, asyn, ierr)
451 !         IF (ierr .NE. ASYN_SUCCESS) RETURN
452 !     ENDIF
453 !
454 ! End Listing
455 !.....
456 !     END SUBROUTINE remove_player
457 !~~~~~
458 !+ <A one line description of this internal subroutine>
459 !~~~~~
460 !     SUBROUTINE erase_asyn (obj, ierr)
461 !.....
462 !
463 ! Description:
464 !   <Say what this subroutine does>
465 !
466 ! Code description:
467 !       Language: Fortran 90.
468 !.....
469 ! Declarations:
470 !
471 ! Subroutine Arguments
472 ! Scalar arguments with intent(in):
473 !     integer(4) :: ierr
474 !     TYPE (asyn_t) :: obj
475 ! Array arguments with intent(in):
476 ! Scalar arguments with intent(inout):
477 ! Array arguments with intent(inout):
478 ! Scalar arguments with intent(out):
479 ! Array arguments with intent(out):
480 ! Local parameters:
481 ! Local scalars:
482 ! Local arrays:
483 !.....
484 ! Listing
485 !

```

```

486     IF (ASSOCIATED(obj%player)) THEN
487         DEALLOCATE (obj%player, STAT=ierr)
488         IF( ierr.NE.0) THEN
489             ierr=ASYN_NO_MEM
490             RETURN
491         ENDIF
492         NULLIFY(obj%player)
493     ENDIF
494     IF (ASSOCIATED(obj%out_one_game)) THEN
495         DEALLOCATE (obj%out_one_game, STAT=ierr)
496         IF( ierr.NE.0) THEN
497             ierr=ASYN_NO_MEM
498             RETURN
499         ENDIF
500         NULLIFY(obj%out_one_game)
501     ENDIF
502     IF (ASSOCIATED(obj%out_many_games)) THEN
503         DEALLOCATE (obj%out_many_games, STAT=ierr)
504         IF( ierr.NE.0) THEN
505             ierr=ASYN_NO_MEM
506             RETURN
507         ENDIF
508         NULLIFY(obj%out_many_games)
509     ENDIF
510 !
511     ierr = ASYN_SUCCESS
512 !
513 ! End Listing
514 !.....
515     END SUBROUTINE erase_asyn
516 !~~~~~
517 !+ <A one line description of this internal subroutine>
518 !~~~~~
519     SUBROUTINE copy_asyn (tgameopt, orig, dupl, ierr)
520 !.....
521 !
522 ! Description:
523 !   <Say what this subroutine does>
524 !
525 ! Code description:
526 !       Language: Fortran 90.
527 !.....
528 ! Declarations:
529 !
530 ! Subroutine Arguments
531 ! Scalar arguments with intent(in):
532     integer(4) :: ierr
533     TYPE (asyn_t) :: orig, dupl
534     TYPE (gameopt_t) :: tgameopt
535 ! Array arguments with intent(in):
536 ! Scalar arguments with intent(inout):
537 ! Array arguments with intent(inout):
538 ! Scalar arguments with intent(out):
539 ! Array arguments with intent(out):
540 ! Local parameters:
541 ! Local scalars:
542     integer(4) :: iplayer
543 ! Local arrays:
544 !.....
545 ! Listing
546 !
547     dupl=orig
548 !
549     IF (ASSOCIATED(dupl%player)) THEN
550         ALLOCATE (dupl%player(dupl%nplayers), STAT=ierr)
551         IF (ierr.NE.0) THEN
552             ierr = ASYN_NO_MEM
553             RETURN
554         ENDIF
555 !
556         DO iplayer=1, dupl%nplayers
557             dupl%player(iplayer) = orig%player(iplayer)
558         ENDDO
559     ENDIF
560 !
561     IF (ASSOCIATED(orig%out_one_game)) THEN
562         ALLOCATE (dupl%out_one_game(UBOUND(orig%out_one_game, 1), dupl%niterations), STAT=ierr)
563         IF (ierr.NE.0) THEN
564             ierr = ASYN_NO_MEM
565             RETURN
566         ENDIF

```

```

567     dupl%out_one_game=orig%out_one_game
568   ENDF
569   IF (ASSOCIATED(orig%out_many_games)) THEN
570     ALLOCATE (dupl%out_many_games(UBOUND(orig%out_many_games, 1), tgameopt%ngames), STAT=ierr)
571     IF (ierr.NE.0) THEN
572       ierr = ASYN_NO_MEM
573       RETURN
574     ENDF
575     dupl%out_many_games=orig%out_many_games
576   ENDF
577 !
578   ierr = ASYN_SUCCESS
579 !
580 ! End Listing
581 !.....
582   END SUBROUTINE copy_asyn
583 !~~~~~
584 !+ <A one line description of this internal subroutine>
585 !~~~~~
586   SUBROUTINE realloc_players ( player, newup, ierr)
587 !.....
588 !
589 ! Description:
590 !   <Say what this subroutine does>
591 !
592 ! Code description:
593 !       Language: Fortran 90.
594 !.....
595 ! Declarations:
596 ! Subroutine Arguments
597 ! Scalar arguments with intent(in):
598   integer(4), INTENT(IN) :: newup
599 ! Array arguments with intent(in):
600 ! Scalar arguments with intent(inout):
601 ! Array arguments with intent(inout):
602   TYPE (player_t), POINTER, dimension (:) :: player
603 ! Scalar arguments with intent(out):
604   integer(4), INTENT(OUT) :: ierr
605 ! Array arguments with intent(out):
606 ! Local parameters:
607 ! Local scalars:
608   integer(4) :: lb, ub, i
609 ! Local arrays:
610   TYPE (player_t), POINTER, dimension (:) :: pltemp
611 !.....
612 ! Listing
613 !
614   lb=LBOUND (player, 1)
615   ub=UBOUND (player, 1)
616 !
617   IF (newup<lb) THEN
618     ierr= ASYN_BAD_DATA
619     RETURN
620   ENDF
621 !
622   ALLOCATE (pltemp(lb:ub), STAT=ierr)
623   IF (ierr.NE.0) THEN
624     ierr= ASYN_NO_MEM
625     RETURN
626   ENDF
627 !
628   DO i = lb, ub
629     pltemp(i)=player(i)
630   ENDDO
631 !
632   DEALLOCATE (player)
633 !
634   ALLOCATE (player(lb:newup), STAT=ierr)
635   IF (ierr.NE.0) THEN
636     ierr= ASYN_NO_MEM
637     RETURN
638   ENDF
639 !
640   IF (ub>newup) THEN
641     DO i = lb, newup
642       player(i)=pltemp(i)
643     ENDDO
644   ELSE
645     DO i = lb, ub
646       player(i)=pltemp(i)
647     ENDDO

```

```
648     ENDIF
649 !
650     DEALLOCATE (pltemp, STAT=ierr)
651 !
652     ierr= ASYN_SUCCESS
653 !
654 ! End Listing
655 !.....
656     END SUBROUTINE realloc_players
657 !~~~~~
658     END MODULE asyn_basic_m
659 !=====
660 !=====
```

```

1  !=====
2  !+ <A one line description of this module>
3  !=====
4  MODULE asyn_m
5  !.....
6  !
7  ! Description:
8  !   <Say what this Module is for>
9  !
10 ! Current code owner : C.Skoulidas
11 !
12 ! History:
13 ! Version      Date      Comment
14 ! -----      -
15 ! <18.1>      <2011>   Original code  <C.Skoulidas>
16 !
17 ! Code description:
18 ! Language: Fortran 90.
19 !.....
20 ! Declarations:
21 !
22 ! Modules Used:
23 !   USE asyn_two_sys_oneoffer_m
24 !
25 ! Imported type definitions
26 ! Imported Parameters:
27 ! Imported scalar variables with intent (in):
28 ! Imported scalar variables with intent (out):
29 ! Imported Array variables with intent(in):
30 ! Imported Array variables with intent(out):
31 ! Imported routines:
32 !
33 !   IMPLICIT NONE
34 !
35 ! Global (Public) Declarations:
36 !
37 !
38 ! Global Type Definitions:
39 !
40 !.....
41 !   CONTAINS
42 !~~~~~
43 !+ <A one line description of this internal subroutine>
44 !~~~~~
45 !   SUBROUTINE calc_asyn (gameopt, asyn, kerror)
46 !.....
47 !
48 ! Description:
49 !   <Say what this subroutine does>
50 !
51 ! Code description:
52 !       Language: Fortran 90.
53 !.....
54 ! Declarations:
55 !
56 ! Subroutine Arguments
57 ! Scalar arguments with intent(in):
58 !   TYPE (gameopt_t) :: gameopt
59 !   TYPE (asyn_t)   :: asyn(gameopt%nsystems)
60 !   integer(4)     :: kerror(gameopt%nsystems)
61 ! Array arguments with intent(in):
62 ! Scalar arguments with intent(inout):
63 ! Array arguments with intent(inout):
64 ! Scalar arguments with intent(out):
65 ! Array arguments with intent(out):
66 ! Local parameters:
67 ! Local scalars:
68 ! Local arrays:
69 !.....
70 ! Listing
71 !
72 !   IF (gameopt%nsystems .EQ. 1) THEN
73 !
74 !       CALL calc_asyn_1sys (gameopt, asyn(gameopt%nsystems), kerror(gameopt%nsystems))
75 !
76 !   ELSE IF (gameopt%nsystems .EQ. 2 .AND. gameopt%offer_type .EQ. OFFER_SPLIT) THEN
77 !
78 !       CALL calc_asyn_2sys_ofsplit (gameopt, asyn, kerror)

```



```

79 !
80     ELSE IF (gameopt%nsystems .EQ. 2 .AND. gameopt%offer_type .EQ. OFFER_ALL) THEN
81 !
82         CALL calc_asyn_2sys_ofall (gameopt, asyn, kerror)
83 !
84     END IF
85 !
86 ! End Listing
87 !.....
88     END SUBROUTINE calc_asyn
89 !-----
90 ! List Of Internal Procedures:
91 ! Subroutines:
92 !
93 !
94 ! Functions:
95 !
96 !
97 !-----
98 !~~~~~
99 !~~~~~
100     END MODULE asyn_m
101 !=====
102 !
103 !=====

```

```

1  !=====
2  !+ <A one line description of this module>
3  !=====
4  MODULE asyn_one_sys_m
5  !.....
6  !
7  ! Description:
8  !   <Say what this Module is for>
9  !
10 ! Current code owner: C.Skoulidas
11 !
12 ! History:
13 ! Version      Date      Comment
14 ! -----      -
15 ! <18.1>      <2011>   Original code   <C.Skoulidas>
16 !
17 ! Code description:
18 ! Language: Fortran 90.
19 !.....
20 ! Declarations:
21 !
22 ! Modules Used:
23 !   USE asyn_basic_m
24 !   use numerical_libraries
25 !
26 ! Imported type definitions
27 ! Imported Parameters:
28 ! Imported scalar variables with intent (in):
29 ! Imported scalar variables with intent (out):
30 ! Imported Array variables with intent(in):
31 ! Imported Array variables with intent(out):
32 ! Imported routines:
33 !
34 !   IMPLICIT NONE
35 !
36 ! Global (Public) Declarations:
37 ! Global parameters:
38 ! Global scalars:
39 ! Global arrays:
40 ! Local (Private) Declarations:
41 ! Local Type Definitions:
42 ! Local parameters:
43 ! Local scalars:
44 ! Local arrays:
45 ! Operator definitions:
46 !
47 !.....
48 !   CONTAINS
49 !-----
50 ! List Of Internal Procedures:
51 ! Subroutines:
52 !
53 !
54 ! Functions:
55 !
56 !-----
57 !=====
58 !+ <A one line description of this external subroutine>
59 !=====
60 !   SUBROUTINE calc_asyn_lsys (gameopt, asyn, kerr)
61 !.....
62 !
63 ! Description:
64 !   ASYN      KEEP ON BIDDING - Bids Above Price Cap NOT ALLOWED
65 !             LINEAR Marginal Cost
66 !             Optional Pricing Method (Uniform OR Pay-As-Bid)
67 !             NO use of GCOMP      Routine
68 !             LAST REVISION: 28 OCTOBER 2005
69 !             ALLOWS Excess Capacity - ALLOWS Loosing at zero sales
70 !             REPEATED runs of the same game
71 !             StDeviation and Variance of SMP added
72 !
73 ! Method:
74 !   <Say how it does it: include references to external documentation>
75 !   <If this routine is divided in to sections, be brief here,
76 !       and put Method comments at the start of each section>
77 !
78 ! Current code owner : <Name of person responsible of this code>
79 !
80 ! History:

```

```

81 ! Version      Date      Comment
82 ! -----      ----      -
83 ! <18.1>      <2011>      Original code <C.Skoulidas>
84 !
85 ! Code description:
86 !           Language: Fortran 90.
87 !.....
88 ! Declarations:
89 !
90 ! Modules Used:
91 ! include statements
92 ! Subroutine Arguments
93 ! Scalar arguments with intent(in):
94     TYPE (asyn_t) :: asyn
95     TYPE (gameopt_t) :: gameopt
96     integer(4) :: kerr
97 ! Array arguments with intent(in):
98 ! Scalar arguments with intent(inout):
99 ! Array arguments with intent(inout):
100 ! Scalar arguments with intent(out):
101 ! Array arguments with intent(out):
102 ! Local parameters:
103     real(8), parameter :: eps_d = 1.0d-08, BIG = 1.0d+30
104     real(8), parameter :: bmin = 1.0d-05
105     real(8), parameter :: zr = 1.d-07
106     integer(4), parameter :: ikind4 = 4, PLAYER_DER = 1, PLAYER_DSB = 2, PLAYER_TYPICAL = 3
107 ! Local scalars:
108     integer(4) :: igames, iplayers, iconst, counter, jplayers, up, low,
109 excess, k, iterations, act
110     REAL(8) :: y, sumab, sumlb, sum_sp, sumsp_sq, sum_purchases,
111 sum_extra, sum_isocost, d, zpos, zneg, difd, l, &
112 demx, spot, avspot, purchases, isocost, extra, kwh, avekwh, price,
113 &
114 aveprice, sumper1, sumper2, rnd, p1, p2, djt, dpi, dpl, dp2,
115 sp_stdevp, sumabs_pr, sp_var, availpower
116     REAL(8) :: player_cost, x_rest, x_wind_pv, total_pv_cap,
117 total_wind_cap, x_shortage
118     REAL(8) :: wind_prod_coef, pv_prod_coef
119     REAL(8) :: wind_production, pv_production,
120 total_wind_production, total_pv_production, xavail
121     INTEGER(4) :: kplayers_der, player_type, kplayers_dsb
122 ! Local arrays:
123
124     integer(4), dimension (asyn%nplayers) :: player_count, coef_flag, expl
125     integer(4), dimension (asyn%nplayers, NCONST) :: action
126     REAL(8), dimension (asyn%nplayers, NCONST) :: pi, pd, pc, lp, eps
127     REAL(8), dimension (asyn%nplayers) :: ba, bb, xx, x, avex, rev, jo, avej, aveba,
128 avebb, theta, da, bamax, bbmax, j, dj
129     REAL(8), dimension (asyn%nplayers*NCONST) :: lpoint, qpoint
130     REAL(8), dimension (asyn%nplayers*NCONST, asyn%nplayers) :: qx
131     REAL(8), ALLOCATABLE, dimension(:) :: varpr
132 !
133     REAL(8), dimension (asyn%nplayers_der) :: x_der_smhydro, x_der_wind, x_der_pv,
134 x_der_chp, x_der_fcells, x_der_shortage_pv, x_der_shortage_wind
135     REAL(8), dimension (asyn%nplayers_der) :: cost_der_smhydro, cost_der_wind,
136 cost_der_pv, cost_der_chp, cost_der_fcells
137     REAL(8), dimension (asyn%nplayers_der) :: j_der_smhydro, j_der_wind, j_der_pv,
138 j_der_chp, j_der_fcells
139
140     REAL(8), dimension (asyn%nplayers_der) :: avex_der_smhydro, avex_der_wind,
141 avex_der_pv, avex_der_chp, avex_der_fcells
142     REAL(8), dimension (asyn%nplayers_der) :: avej_der_smhydro, avej_der_wind,
143 avej_der_pv, avej_der_chp, avej_der_fcells
144 !
145     REAL(8), dimension (asyn%nplayers_dsb) :: x_dsb_smhydro, x_dsb_wind, x_dsb_pv,
146 x_dsb_chp, x_dsb_interrup, x_dsb_shortage_pv, x_dsb_shortage_wind
147     REAL(8), dimension (asyn%nplayers_dsb) :: cost_dsb_smhydro, cost_dsb_wind,
148 cost_dsb_pv, cost_dsb_chp, cost_dsb_interrup
149     REAL(8), dimension (asyn%nplayers_dsb) :: j_dsb_smhydro, j_dsb_wind, j_dsb_pv,
150 j_dsb_chp, j_dsb_interrup
151
152     REAL(8), dimension (asyn%nplayers_dsb) :: avex_dsb_smhydro, avex_dsb_wind,
153 avex_dsb_pv, avex_dsb_chp, avex_dsb_interrup
154     REAL(8), dimension (asyn%nplayers_dsb) :: avej_dsb_smhydro, avej_dsb_wind,
155 avej_dsb_pv, avej_dsb_chp, avej_dsb_interrup
156
157
158     REAL(8), dimension (asyn%nplayers*NCONST) :: dpoint, diffcheck, lpoint_sorted,
159 qpoint_sorted, dpoint_sorted !new
160     integer(4) :: knees, jknee, iknee
161     REAL(8) :: q1, q2, l1, l2, d1, d2, q, diff1, diff2, ba_dsb, slope, dsb_capacity

```

```

162     logical :: check
163
164 ! .....
165 ! Listing
166 !
167     ALLOCATE (varpr(asyn%niterations) )
168 !
169 ! initialize error index
170     kerr = ASYN_SUCCESS
171 !
172 ! Start the game iterations
173 !
174     DO igames = 1, gameopt%ngames
175 !
176         CALL random_seed ()
177 !
178 ! asyn%p_flag .EQ. 0 ! the following condition is always FALSE
179         IF (asyn%p_flag .EQ. 1) THEN
180 !
181             DO iplayers = 1, asyn%nplayers
182 !
183                 DO iconst = 1, NCONST
184 !
185                     CALL random_number (y)
186                     asyn%player(iplayers)%period(iconst) = INT(asyn%period_min + (asyn%period_max -
187 asyn%period_min ) * y)
188 !
189                     ENDDO
190                 END DO
191             END IF
192 !
193 !         Initial bid parameters values and sums related to them
194 !
195 !         Assign/Reassign initial values (=zero) to Sums related with the
196 !         Bid Parameters - 1st Round
197 !
198         sum_sp= 0.0d+00
199         sumsp_sq = 0.0d+00
200 !
201         sum_purchases = 0.0d+00
202         sum_extra = 0.0d+00
203         sum_isocost = 0.0d+00
204 !
205 !         Initial Demand Assignment/Reassignment
206         CALL random_seed()
207         d = asyn%inid
208 !
209 !         Assign/Reassign initial values to Behaviour Probabilities
210 !
211         DO iplayers=1, asyn%nplayers
212             pi(iplayers, :) = asyn%player(iplayers)%inipi(:)
213             pd(iplayers, :) = asyn%player(iplayers)%inipd(:)
214             pc(iplayers, :) = asyn%player(iplayers)%inipc(:)
215         ENDDO
216 !
217 !         Assing/Reassign values to bidding parameters and Calculate
218 !         the sums related with them - 1st Round
219 !
220         DO iplayers=1, asyn%nplayers
221 ! dsb-start
222             IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
223                 ba(iplayers) = asyn%player(iplayers)%a
224                 bb(iplayers) = 2.0d+00 * asyn%player(iplayers)%b
225             ELSE
226                 ba(iplayers) = asyn%player(iplayers)%interrup_high_price -
227 &
228                     &
229                         asyn%player(iplayers)%curve_slope *
230                             asyn%player(iplayers)%total_dsb_capacity
231             IF (ba(iplayers) .LT. 0.0) ba(iplayers) = 0.0
232                 bb(iplayers) = asyn%player(iplayers)%curve_slope
233                 ba_dsb = ba(iplayers)
234                 slope = asyn%player(iplayers)%curve_slope
235                 dsb_capacity = asyn%player(iplayers)%total_dsb_capacity
236
237             END IF
238 ! dsb-end
239         END DO
240 !
241         sumab = 0.0d+00
242         sum1b = 0.0d+00

```

```

243      DO iplayers = 1, asyn%nplayers
244 ! dsb-start
245           IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
246               sumab = sumab + (ba(iplayers) / bb(iplayers))
247               sumlb = sumlb + (1.0d+00 / bb(iplayers))
248           ENDIF
249 ! dsb-end
250       END DO
251 !
252       DO iplayers = 1, asyn%nplayers
253           lp(iplayers, 1) = ba(iplayers) + bb(iplayers) * asyn%player(iplayers)%xmin
254           lp(iplayers, 2) = ba(iplayers) + bb(iplayers) * asyn%player(iplayers)%xmax
255       ENDDO
256 !
257 ! Check for available power with respect to the demand
258       availpower = 0.0d+00
259       DO iplayers = 1, asyn%nplayers
260 ! dsb-start
261           IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
262               availpower = availpower + asyn%player(iplayers)%xmax
263           END IF
264 ! dsb-start
265       END DO
266 !
267       IF (availpower .LT. d ) THEN
268 !
269           kerr = ERR_DEMAND
270           RETURN
271 !
272       END IF
273 !
274 !       Calculation of the Spot Price
275 !
276       counter = 0
277 !
278 !       Definition of the 2xPLAYERS nodal points of the aggregate Supply Curve
279 !       and calculation of the available quantity (Qx)for each player at every point
280 !
281       DO iconst = 1, NCONST
282           DO iplayers = 1, asyn%nplayers
283               IF (asyn%player(iplayers)%category .eq. CATEGORY(11) ) cycle
284               counter = counter + 1
285               lpoint(counter) = lp(iplayers, iconst) !assignment of all values
286               qpoint(counter) = 0
287               DO jplayers = 1, asyn%nplayers
288                   IF (asyn%player(jplayers)%category .eq. CATEGORY(11) ) cycle
289                   IF (lpoint(counter) .LE. lp(jplayers, 1)) THEN
290 !                       If the point is below player's min
291                       qx(counter, jplayers) = 0.0d+00
292                   ELSE IF (lpoint(counter) .GE. lp(jplayers, 2)) THEN
293 !                       If the point is above player's max
294                       qx(counter, jplayers) = asyn%player(jplayers)%xmax
295                   ELSE
296 !                       The point between player's min and max
297                       qx(counter, jplayers) = (lpoint(counter) - ba(jplayers)) / bb(jplayers)
298                   END IF
299 !                       Also calculation of the total available quantity (QPoint)
300 !                       at each nodal point
301                       qpoint(counter) = qpoint(counter) + qx(counter, jplayers)
302                   ENDDO
303               ENDDO
304           ENDDO
305           knees = counter
306 !
307       if (asyn%nplayers_dsb .ne. 0 ) then
308 !
309           DO iknee= 1, knees
310 !
311               dpoint(iknee) = ba_dsb + slope * (asyn%inid - qpoint(iknee))
312 !
313           END DO
314 !
315           CALL DSVRGN (knees, qpoint, qpoint_sorted)
316 !
317           DO iknee= 1, knees
318               DO jknee= 1, knees
319                   IF ( ABS( qpoint_sorted(iknee)-qpoint(jknee) ) .LE. eps_d )
320 THEN
321 !
322                     lpoint_sorted(iknee) = lpoint(jknee)
323                     dpoint_sorted(iknee) = dpoint(jknee)

```

```

324                               EXIT
325 !
326                               END IF
327
328                               END DO
329                               END DO
330 !
331                               DO iknee= 1, knees-1
332 !
333                                   diff1 = dpoint_sorted(iknee) - lpoint_sorted(iknee)
334                                   diff2 = dpoint_sorted(iknee+1) - lpoint_sorted(iknee+1)
335
336                                   IF ( abs(diff1) .LE. eps_d .OR. abs(diff2) .LE. eps_d ) THEN
337
338                                       IF ( abs(diff1) .LE. eps_d ) THEN
339                                           counter = iknee
340                                       ELSE
341                                           counter = iknee+1
342                                       ENDDIF
343 !
344                                       l = lpoint(counter)
345
346                                       DO jplayers = 1, asyn%nplayers
347
348                                           IF (asyn%player(jplayers)%category .NE. CATEGORY(11) )
349 THEN
350
351                                               xx(jplayers) = qx(counter, jplayers)
352                                               IF ( xx(jplayers) .LE.
353 asyn%player(jplayers)%xmin ) xx(jplayers) = 0.0
354                                               IF ( xx(jplayers) .GE.
355 asyn%player(jplayers)%xmax ) xx(jplayers) = asyn%player(jplayers)%xmax
356
357                                               ELSE
358
359                                                   xx(jplayers) = asyn%inid - q
360                                                   xx(jplayers) = asyn%inid - (1 - ba_dsb) /slope
361                                               IF ( xx(jplayers) .LE.
362 asyn%player(jplayers)%xmin ) xx(jplayers) = 0.0
363                                               IF ( xx(jplayers) .GE.
364 asyn%player(jplayers)%xmax ) xx(jplayers) = asyn%player(jplayers)%xmax
365
366                                               END IF
367
368                                       END DO
369
370                                       GOTO 444
371
372                               END IF
373
374                               IF (diff1*diff2 .LT. 0.0) THEN
375                                   jknee =iknee
376                                   EXIT ! exoume tomi
377 !
378                               END DO
379
380                                   q1 = qpoint_sorted(jknee)
381                                   q2 = qpoint_sorted(jknee+1)
382 !
383                                   l1 = lpoint_sorted(jknee)
384                                   l2 = lpoint_sorted(jknee+1)
385 !
386                                   d1 = dpoint_sorted(jknee)
387                                   d2 = dpoint_sorted(jknee+1)
388 !
389                                   q = ((d2-l2)*q1 - (d1-l1)*q2) / (l1-l2-(d1-d2))
390 !
391                                   IF (q .LE. d .AND. q .GE. d-dsb_capacity) THEN
392
393                                       l = (l1-l2) * q / (q1-q2) + (l2*q1-l1*q2) / (q1-q2)
394
395                                   ELSE
396                                       IF (q .GT. d) THEN
397                                           q = d
398                                       ELSE
399                                           q = d-dsb_capacity
400                                       END IF
401 !
402                                   DO iknee= 1, knees-1
403 !
404                                       diff1 = q - dpoint_sorted(iknee)

```

```

405         diff2 = q - dpoint_sorted(iknee+1)
406         IF (diff1*diff2 .LT. 0.0) THEN
407             jknee = iknee
408             EXIT ! exoume tomi
409         END IF
410     END DO
411
412     q1 = qpoint_sorted(jknee)
413     q2 = qpoint_sorted(jknee+1)
414 !
415     l1 = lpoint_sorted(jknee)
416     l2 = lpoint_sorted(jknee+1)
417 !
418     l = (l1-l2) * q / (q1-q2) + (l2*q1-l1*q2) / (q1-q2)
419 END IF
420 !
421 DO iplayers = 1, asyn%nplayers
422     IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
423         xx(iplayers) = (1 - ba(iplayers)) / bb(iplayers)
424         IF ( xx(iplayers) .LE. asyn%player(iplayers)%xmin )
425     xx(iplayers) = 0.0
426         IF ( xx(iplayers) .GE. asyn%player(iplayers)%xmax )
427     xx(iplayers) = asyn%player(iplayers)%xmax
428     ELSE
429         xx(iplayers) = asyn%inid - q
430 !         xx(iplayers) = asyn%inid - (1 - ba_dsb) /slope
431         IF ( xx(iplayers) .LE. asyn%player(iplayers)%xmin )
432     xx(iplayers) = 0.0
433         IF ( xx(iplayers) .GE. asyn%player(iplayers)%xmax )
434     xx(iplayers) = asyn%player(iplayers)%xmax
435         END IF
436     end do
437 !
438     444 continue
439
440 else ! NO DSB AGGREGATOR
441 !
442     zpos = REAL(asyn%nplayers) * d + 1.0d+00
443     zneg = -zpos
444 !
445     up = 0
446     low = 0
447 !
448     DO counter = 2 * asyn%nplayers, 1, -1
449         difd = qpoint(counter) - d
450         IF ( ABS(dofd) .LE. eps_d ) THEN
451             DO iplayers = 1, asyn%nplayers
452                 expl(iplayers) = iplayers
453                 xx(iplayers) = qx(counter,
454 iplayers)
455             END DO
456             l = lpoint(counter)
457             GOTO 530
458         ELSEIF (dofd .GT. 0) THEN
459             !
460             IF (zpos .GT. difd) THEN
461                 zpos = difd
462                 up = counter
463             ELSE IF (ABS(zpos - difd) .LE. eps_d)
464 THEN
465                 IF (up .EQ. 0 ) THEN
466                     kerr = ERR_SOLVE_UP
467                     RETURN
468                 END IF
469                 IF (lpoint(up) .GT.
470 lpoint(counter)) up = counter
471             END IF
472         ELSE
473             IF (zneg .LT. difd) THEN
474                 zneg = difd
475                 low = counter
476             ELSE IF (ABS(zneg - difd) .LE. eps_d)
477 THEN
478                 IF (low .EQ. 0 ) THEN
479                     kerr = ERR_SOLVE_LOW
480                     RETURN
481                 END IF
482                 IF (lpoint(low) .LT.
483 lpoint(counter)) low = counter
484             END IF
485

```

```

486                                     END IF
487     END DO
488     !
489     demx = d
490     excess = 0
491     !
492     DO iplayers = 1, asyn%nplayers
493     !
494         expl(iplayers) = 0
495     !
496         IF (low .EQ. 0 ) THEN
497             kerr = ERR_SOLVE_LOW
498             RETURN
499         END IF
500
501         IF (up .EQ. 0 ) THEN
502             kerr = ERR_SOLVE_UP
503             RETURN
504         END IF
505     !
506     IF ( ABS(qx(low, iplayers) -
507 asyn%player(iplayers)%xmax) .LE. eps_d ) THEN
508     !
509         expl(iplayers) = iplayers
510         xx(iplayers) =
511 asyn%player(iplayers)%xmax
512     !
513         demx = demx - xx(iplayers)
514     !
515     ! dsb-start
516         IF (asyn%player(iplayers)%category .NE.
517 CATEGORY(11) ) THEN
518             sumab = sumab - (ba(iplayers) /
519 bb(iplayers))
520             sumlb = sumlb - (1.0d+00 /
521 bb(iplayers))
522         END IF
523     ! dsb-start
524     !
525     ELSE IF ( ABS(qx(up,iplayers)) .LE. eps_d )
526 THEN
527     !
528         expl(iplayers) = iplayers
529         xx(iplayers) = 0.0d+00
530     !
531     ! dsb-start
532         IF (asyn%player(iplayers)%category .NE.
533 CATEGORY(11) ) THEN
534             sumab = sumab - (ba(iplayers) /
535 bb(iplayers))
536             sumlb = sumlb - (1.0d+00 /
537 bb(iplayers))
538         ENDIF
539     ! dsb-end
540     !
541     ELSE IF ( ABS(qx(low,iplayers)) .LE. eps_d )
542 THEN
543     !
544         IF ( qpoint(low) +
545 asyn%player(iplayers)%xmin .GT. d ) THEN
546     !
547         excess = iplayers
548         EXIT
549     !
550         END IF
551     !
552     END IF
553     !
554     END DO
555     !
556     ! Calculation of Price's first value [SPOT]
557     ! Assign initial value to Price's Moving Average [AveSPOT]
558     IF (excess .GT. 0) THEN
559     !
560         IF (low .EQ. 0 ) THEN
561             kerr = ERR_SOLVE_LOW
562             RETURN
563         END IF
564
565         l = lpoint(low)
566     !

```



```

567 DO iplayers = 1, asyn%nplayers
568     expl(iplayers) = iplayers
569     xx(iplayers) = qx(low, iplayers)
570 END DO
571 !
572     xx(excess) = asyn%player(excess)%xmin
573 !
574     ELSE
575         l = (demx + sumab) / sumlb
576     END IF
577 !
578     530 CONTINUE
579 !
580
581 ! Assign to Players of the first quantities
582 [X(iiterations)] bought by the ISO
583 ! Assign the same values to quantities' moving averages
584 [AveX(iiterations)]
585 ! Calculation of players' first income [Jo(iiterations)]
586 ! Assign the same values to incomes' moving averages
587 [AveJ(iiterations)]
588 !
589 DO iplayers = 1, asyn%nplayers
590     IF (expl(iplayers) .NE. iplayers)
591 xx(iplayers) = (1 - ba(iplayers)) / bb(iplayers)
592     END DO
593
594 end if
595
596     spot = 1
597     avespot = spot
598 !
599     varpr(1) = spot
600     sum_sp = spot
601     sumsp_sq = spot * spot
602 !
603     purchases = 0.0d+00
604     isocost = 0.0d+00
605     kplayers_der = 0
606     kplayers_dsb = 0
607 !
608 ! Calculation of costs, profits and moving averages for each player
609 !
610     DO iplayers = 1, asyn%nplayers
611         IF (asyn%player(iplayers)%category .EQ. CATEGORY(10)) THEN
612 !
613             kplayers_der = kplayers_der + 1
614             player_type = PLAYER_DER
615 ! dsb-start
616             ELSE IF (asyn%player(iplayers)%category .EQ. CATEGORY(11) ) THEN
617 !
618                 kplayers_dsb = kplayers_dsb + 1
619                 player_type = PLAYER_DSB
620 ! dsb-end
621             ELSE
622 !
623                 player_type = PLAYER_TYPICAL
624 !
625             ENDIF
626 !
627             x(iplayers) = xx(iplayers)
628             purchases = purchases + x(iplayers)
629             avex(iplayers) = x(iplayers)
630 !
631 !
632 ! If player is of DER Aggregator type extra calculations
633 !
634 ! dsb-start
635             IF (player_type .EQ. PLAYER_DER .OR. player_type .EQ. PLAYER_DSB) THEN ! DERaggr
636 ! dsb-end
637 !
638             Random for wind
639 !
640             CALL random_number (y)
641 !
642             IF (y .LE. asyn%player(iplayers)%wind_no_production) THEN
643 !
644                 wind_prod_coef = 0.0
645 !
646             ELSEIF (y.GE.asyn%player(iplayers)%wind_max_production) THEN
647 !

```

```

648             wind_prod_coef = 1.0
649 !
650             ELSE
651 !
652             wind_prod_coef = (y-asyn%player(iplayers)%wind_no_production)
653 / &
654             (1.0 - asyn%player(iplayers)%wind_no_production -
655 asyn%player(iplayers)%wind_max_production)
656             END IF
657 !
658             wind_production = asyn%player(iplayers)%wind_capacity * wind_prod_coef
659             total_wind_production = wind_production *
660 asyn%player(iplayers)%wind_count
661 !
662             total_wind_cap = asyn%player(iplayers)%wind_capacity *
663 asyn%player(iplayers)%wind_count
664 !
665 !           Random for PV
666 !
667             CALL random_number (y)
668 !
669             IF (y .LE. asyn%player(iplayers)%pv_no_production) THEN
670 !
671             pv_prod_coef = 0.0
672 !
673             ELSEIF (y.GE.asyn%player(iplayers)%pv_max_production) THEN
674 !
675             pv_prod_coef = 1.0
676 !
677             ELSE
678 !
679             pv_prod_coef = (y-asyn%player(iplayers)%pv_no_production)/ &
680 (1.0 - asyn%player(iplayers)%pv_no_production -
681 asyn%player(iplayers)%pv_max_production)
682 !
683             END IF
684 !
685             pv_production = asyn%player(iplayers)%pv_capacity * pv_prod_coef
686             total_pv_production = pv_production * asyn%player(iplayers)%pv_count
687 !
688             total_pv_cap = asyn%player(iplayers)%pv_capacity *
689 asyn%player(iplayers)%pv_count
690 !
691             END IF
692 !
693             IF (player_type .EQ. PLAYER_DER) THEN ! DERaggr
694 !
695             DER Internal Merit Order
696 !
697             x_der_smhydro(kplayers_der) = 0.0
698             x_der_wind(kplayers_der) = 0.0
699             x_der_pv(kplayers_der) = 0.0
700             x_der_chp(kplayers_der) = 0.0
701             x_der_fcells(kplayers_der) = 0.0
702 !
703             x_rest = x(iplayers)
704 !
705             Small Hydro
706             x_der_smhydro(kplayers_der) = MIN(x_rest,
707 asyn%player(iplayers)%total_smhydro_capacity)
708             x_rest = x_rest - x_der_smhydro(kplayers_der)
709             IF (x_rest .LE. 0.0d+00) GOTO 440
710 !
711             Wind & PV
712             x_wind_pv = MIN(x_rest, (total_wind_production + total_pv_production))
713 !
714             IF (x_wind_pv .EQ. (total_wind_production + total_pv_production)) THEN
715 !
716             x_der_wind(kplayers_der) = total_wind_production
717             x_der_pv(kplayers_der) = total_pv_production
718 !
719             ELSE
720 !
721             x_der_wind(kplayers_der) = MIN(x_wind_pv / 2.0, total_wind_production)
722             x_der_pv(kplayers_der) = MIN(x_wind_pv / 2.0, total_pv_production)
723 !
724             IF (x_der_pv(kplayers_der) .EQ. total_pv_production) THEN
725 !
726             x_der_wind(kplayers_der) = x_wind_pv -
727 x_der_pv(kplayers_der) !MIN(x_wind_pv-x_pv(kplayers_der), total_wind_cap)
728

```

```

729             END IF
730 !
731             IF (x_der_wind(kplayers_der) .EQ. total_wind_production) THEN
732 !
733                 x_der_pv(kplayers_der) = x_wind_pv -
734 x_der_wind(kplayers_der) !MIN(x_wind_pv-x_wind(kplayers_der), total_pv_cap)
735 !
736             END IF
737 !
738         ENDIF
739 !
740             x_rest = x_rest - x_der_wind(kplayers_der) - x_der_pv(kplayers_der)
741             IF (x_rest .LE. 0.0d+00) GOTO 440
742 !
743 !   CHP
744             x_der_chp(kplayers_der) = MIN(x_rest,
745 asyn%player(iplayers)%total_chp_capacity)
746             x_rest = x_rest - x_der_chp(kplayers_der)
747             IF (x_rest .LE. 0.0d+00) GOTO 440
748 !
749 !   Fuel Cells
750             x_der_fcells(kplayers_der) = MIN(x_rest,
751 asyn%player(iplayers)%total_fcells_capacity)
752             x_rest = x_rest - x_der_fcells(kplayers_der)
753             440 CONTINUE
754 !
755 ! Average Quantities for DER generators
756 !
757             avex_der_smhydro(kplayers_der) = x_der_smhydro(kplayers_der)
758             avex_der_wind(kplayers_der) = x_der_wind(kplayers_der)
759             avex_der_pv(kplayers_der) = x_der_pv(kplayers_der)
760             avex_der_chp(kplayers_der) = x_der_chp(kplayers_der)
761             avex_der_fcells(kplayers_der) = x_der_fcells(kplayers_der)
762 !
763 ! Shortage
764             IF (x(iplayers).GT. (asyn%player(iplayers)%total_der_capacity -
765 asyn%player(iplayers)%total_intermittent_capacity + &
766             total_wind_production + total_pv_production)) THEN
767 !
768                 x_shortage = x(iplayers) -
769 (asyn%player(iplayers)%total_der_capacity - asyn%player(iplayers)%total_intermittent_capacity + &
770             total_wind_production + total_pv_production)
771 !
772                 IF ((total_wind_production .LT.
773 asyn%player(iplayers)%wind_estim_capacity) &
774             .AND. (total_pv_production .GE.
775 asyn%player(iplayers)%pv_estim_capacity) ) THEN
776 !
777                     x_der_shortage_wind(kplayers_der) = x_shortage
778                     x_der_shortage_pv(kplayers_der) = 0.0
779 !
780                     ELSE IF (total_wind_production .GE.
781 asyn%player(iplayers)%wind_estim_capacity &
782             .AND. total_pv_production .LT.
783 asyn%player(iplayers)%pv_estim_capacity) THEN
784 !
785                         x_der_shortage_wind(kplayers_der) = 0.0
786                         x_der_shortage_pv(kplayers_der) = x_shortage
787 !
788                     ELSE
789 !
790                         x_der_shortage_wind(kplayers_der) =
791 asyn%player(iplayers)%wind_estim_capacity - total_wind_production
792                         x_der_shortage_pv(kplayers_der) = x_shortage -
793 x_der_shortage_wind(kplayers_der)
794 !
795                     END IF
796 !
797                 ELSE
798                     x_shortage = 0.0
799                     x_der_shortage_pv(kplayers_der) = 0.0
800                     x_der_shortage_wind(kplayers_der) = 0.0
801                 END IF
802 !
803 ! DER Costs Calculation
804 !
805             cost_der_smhydro(kplayers_der) = 0.0
806 !
807             cost_der_wind(kplayers_der) = x_der_shortage_wind(kplayers_der) * spot * (1.0d+00 +
808 asyn%player(iplayers)%assurance_cost)
809 !

```

```

810         cost_der_pv(kplayers_der) = x_der_shortage_pv(kplayers_der) * spot * (1.0d+00 +
811 asyn%player(iplayers)%assurance_cost)
812 !
813         cost_der_chp(kplayers_der) = (asyn%player(iplayers)%fc +
814 asyn%player(iplayers)%a * x_der_chp(kplayers_der) + &
815                                     asyn%player(iplayers)%b *
816 x_der_chp(kplayers_der) * x_der_chp(kplayers_der)) *
817                                     &
818 (1.0 - asyn%player(iplayers)%chp_savings)
819 !
820         cost_der_fcells(kplayers_der) = (asyn%player(iplayers)%fc +
821 asyn%player(iplayers)%a * x_der_fcells(kplayers_der) + &
822                                     asyn%player(iplayers)%b * x_der_fcells(kplayers_der)) * &
823                                     (1.0 -
824 asyn%player(iplayers)%chp_savings) * (1.0 + asyn%player(iplayers)%fcells_pluscost)
825 !
826         player_cost = cost_der_smhydro(kplayers_der) + cost_der_wind(kplayers_der) +
827 cost_der_pv(kplayers_der) + cost_der_chp(kplayers_der) + cost_der_fcells(kplayers_der)
828 !
829 !
830 !
831 ! dsb-start
832         ELSE IF (player_type .EQ. PLAYER_DSB) THEN ! DSBaggr
833 !
834 !     DER Internal Merit Order
835 !
836         x_dsb_smhydro(kplayers_dsb) = 0.0
837         x_dsb_wind(kplayers_dsb) = 0.0
838         x_dsb_pv(kplayers_dsb) = 0.0
839         x_dsb_chp(kplayers_dsb) = 0.0
840         x_dsb_interrup(kplayers_dsb) = 0.0
841 !
842         x_rest = x(iplayers)
843 !
844 !     Small Hydro
845         x_dsb_smhydro(kplayers_dsb) = MIN(x_rest,
846 asyn%player(iplayers)%total_smhydro_capacity)
847         x_rest = x_rest - x_dsb_smhydro(kplayers_dsb)
848         IF (x_rest .LE. 0.0d+00) GOTO 450
849 !
850 !     Wind & PV
851         x_wind_pv = MIN(x_rest, (total_wind_production + total_pv_production))
852 !
853         IF (x_wind_pv .EQ. (total_wind_production + total_pv_production)) THEN
854 !
855             x_dsb_wind(kplayers_dsb) = total_wind_production
856             x_dsb_pv(kplayers_dsb) = total_pv_production
857 !
858         ELSE
859 !
860             x_dsb_wind(kplayers_dsb) = MIN(x_wind_pv / 2.0, total_wind_production)
861             x_dsb_pv(kplayers_dsb) = MIN(x_wind_pv / 2.0, total_pv_production)
862 !
863             IF (x_dsb_pv(kplayers_dsb) .EQ. total_pv_production) THEN
864 !
865                 x_dsb_wind(kplayers_dsb) = x_wind_pv -
866 x_dsb_pv(kplayers_dsb) !MIN(x_wind_pv-x_pv(kplayers_dsb), total_wind_cap)
867 !
868             END IF
869 !
870             IF (x_dsb_wind(kplayers_dsb) .EQ. total_wind_production) THEN
871 !
872                 x_dsb_pv(kplayers_dsb) = x_wind_pv -
873 x_dsb_wind(kplayers_dsb) !MIN(x_wind_pv-x_wind(kplayers_dsb), total_pv_cap)
874 !
875             END IF
876 !
877         ENDIF
878 !
879         x_rest = x_rest - x_dsb_wind(kplayers_dsb) - x_dsb_pv(kplayers_dsb)
880         IF (x_rest .LE. 0.0d+00) GOTO 450
881 !
882 !     CHP
883         x_dsb_chp(kplayers_dsb) = MIN(x_rest,
884 asyn%player(iplayers)%total_chp_capacity)
885         x_rest = x_rest - x_dsb_chp(kplayers_dsb)
886         IF (x_rest .LE. 0.0d+00) GOTO 450
887 !
888 !     Fuel Cells
889         x_dsb_interrup(kplayers_dsb) = MIN(x_rest,
890 asyn%player(iplayers)%interrup_capacity)

```

```

891             x_rest = x_rest - x_dsb_interrup(kplayers_dsb)
892             450 CONTINUE
893 !
894 ! Average Quantities for dsb generators
895 !
896             avex_dsb_smhydro(kplayers_dsb)= x_dsb_smhydro(kplayers_dsb)
897             avex_dsb_wind(kplayers_dsb)= x_dsb_wind(kplayers_dsb)
898             avex_dsb_pv(kplayers_dsb)= x_dsb_pv(kplayers_dsb)
899             avex_dsb_chp(kplayers_dsb)= x_dsb_chp(kplayers_dsb)
900             avex_dsb_interrup(kplayers_dsb)= x_dsb_interrup(kplayers_dsb)
901 !
902 ! Shortage
903
904             IF (x(iplayers).GT. (asyn%player(iplayers)%total_dsb_dergen_capacity -
905 asyn%player(iplayers)%total_intermittent_capacity + &
906             total_wind_production + total_pv_production)) THEN
907 !
908             x_shortage = x(iplayers) -
909 (asyn%player(iplayers)%total_dsb_dergen_capacity -
910 asyn%player(iplayers)%total_intermittent_capacity + &
911             total_wind_production + total_pv_production)
912 !
913             IF (x_dsb_interrup(kplayers_dsb) .LT.
914 asyn%player(iplayers)%interrup_capacity) THEN
915 !
916             xavail = asyn%player(iplayers)%interrup_capacity -
917 x_dsb_interrup(kplayers_dsb)
918 !
919             IF (x_shortage .LE. xavail) THEN
920 !
921             x_dsb_interrup(kplayers_dsb) =
922 x_dsb_interrup(kplayers_dsb) + x_shortage
923             x_shortage = 0.0
924
925             ELSE
926             x_dsb_interrup(kplayers_dsb) =
927 asyn%player(iplayers)%interrup_capacity
928             x_shortage = x_shortage - xavail
929             ENDIF
930             END IF
931 !
932             IF (x_shortage .GT. 0.0) THEN
933 !
934             IF ((total_wind_production .LT.
935 asyn%player(iplayers)%wind_estim_capacity) &
936             .AND. (total_pv_production .GE.
937 asyn%player(iplayers)%pv_estim_capacity) ) THEN
938 !
939             x_dsb_shortage_wind(kplayers_dsb) = x_shortage
940             x_dsb_shortage_pv(kplayers_dsb) = 0.0
941 !
942             ELSE IF (total_wind_production .GE.
943 asyn%player(iplayers)%wind_estim_capacity &
944             .AND. total_pv_production .LT.
945 asyn%player(iplayers)%pv_estim_capacity) THEN
946 !
947             x_dsb_shortage_wind(kplayers_dsb) = 0.0
948             x_dsb_shortage_pv(kplayers_dsb) = x_shortage
949 !
950             ELSE
951 !
952             x_dsb_shortage_wind(kplayers_dsb) =
953 asyn%player(iplayers)%wind_estim_capacity - total_wind_production
954             x_dsb_shortage_pv(kplayers_dsb) = x_shortage -
955 x_dsb_shortage_wind(kplayers_dsb)
956 !
957             END IF
958 !
959             END IF
960 !
961             ELSE
962             x_shortage = 0.0
963             x_dsb_shortage_pv(kplayers_dsb) = 0.0
964             x_dsb_shortage_wind(kplayers_dsb) = 0.0
965             END IF
966 !
967 ! dsb Costs Calculation
968 !
969             cost_dsb_smhydro(kplayers_dsb) = 0.0
970 !
971             cost_dsb_wind(kplayers_dsb) = x_dsb_shortage_wind(kplayers_dsb) * spot * (1.0d+00 +

```

```

972 asyn%player(iplayers)%assurance_cost)
973 !
974         cost_dsb_pv(kplayers_dsb) = x_dsb_shortage_pv(kplayers_dsb) * spot * (1.0d+00 +
975 asyn%player(iplayers)%assurance_cost)
976 !
977         cost_dsb_chp(kplayers_dsb) = (asyn%player(iplayers)%fc +
978 asyn%player(iplayers)%a * x_dsb_chp(kplayers_dsb) + &
979                                     asyn%player(iplayers)%b *
980 x_dsb_chp(kplayers_dsb) * x_dsb_chp(kplayers_dsb)) *
981                                     &
982 (1.0 - asyn%player(iplayers)%chp_savings)
983 !
984         cost_dsb_interrup(kplayers_dsb) =
985 asyn%player(iplayers)%interrup_low_price * x_dsb_interrup(kplayers_dsb) +
986                                     0.5d+00 *
987 asyn%player(iplayers)%curve_slope * x_dsb_interrup(kplayers_dsb)
988 !
989         player_cost = cost_dsb_smhydro(kplayers_dsb) + cost_dsb_wind(kplayers_dsb) +
990 cost_dsb_pv(kplayers_dsb) +
991                                     &
992                                     cost_dsb_chp(kplayers_dsb) +
993 cost_dsb_interrup(kplayers_dsb)
994 ! dsb-end
995 !
996         ELSE ! Common Player
997 !
998         player_cost = asyn%player(iplayers)%fc +
999
1000         &
1001                                     asyn%player(iplayers)%a * x(iplayers) +
1002         &
1003                                     asyn%player(iplayers)%b * x(iplayers) *
1004 x(iplayers)
1005         END IF
1006 ! Pricing Method
1007 !
1008         IF (asyn%pr_method .EQ. 0) THEN
1009             ! Uniform Pricing
1010             rev(iplayers) = spot * x(iplayers)
1011 !
1012             ELSE
1013                 ! Pay-as-Bid
1014                 IF ( ABS(x(iplayers)) .LE. eps_d ) THEN
1015                     rev(iplayers) = 0.0d+00
1016                 ELSE
1017                     rev(iplayers) = ba(iplayers) * x(iplayers) + (bb(iplayers) / 2.0d+00) *
1018 x(iplayers) * x(iplayers) +
1019                                     &
1020                                     (bb(iplayers) / 2.0d+00) * asyn%player(iplayers)%xmin *
1021 asyn%player(iplayers)%xmin
1022                     END IF
1023                 END IF
1024 !
1025                 jo(iplayers) = rev(iplayers) - player_cost
1026                 isocost = isocost + rev(iplayers)
1027                 avej(iplayers) = jo(iplayers)
1028             END DO
1029 !
1030             extra = purchases - d
1031             kwh = isocost / purchases
1032             avekwh = kwh
1033 !
1034             IF (asyn%pr_method .EQ. 0) THEN
1035 !
1036                 price = spot
1037                 aveprice = avspot
1038 !
1039             ELSE
1040 !
1041                 price = kwh
1042                 aveprice = avekwh
1043 !
1044             END IF
1045 !
1046             sum_purchases = purchases
1047             sum_extra = extra
1048             sum_isocost = isocost
1049 !
1050             DER Generators Profit Calculation
1051 !
1052             kplayers_der = 0

```

```

1053         DO iplayers = 1, asyn%nplayers
1054     !
1055         IF (asyn%player(iplayers)%category .EQ. CATEGORY(10)) THEN ! DERaggr
1056     !
1057         kplayers_der = kplayers_der + 1
1058     !
1059         IF (asyn%player(iplayers)%share_profit .EQ. 0) THEN
1060     !
1061             IF ( ABS(x(iplayers)) .GT. eps_d) THEN
1062     !
1063                 j_der_smhydro(kplayers_der)= jo(iplayers) *
1064     x_der_smhydro(kplayers_der) / x(iplayers)
1065                 j_der_wind(kplayers_der)= jo(iplayers) *
1066     x_der_wind(kplayers_der) / x(iplayers)
1067                 j_der_pv(kplayers_der)= jo(iplayers)*
1068     x_der_pv(kplayers_der) / x(iplayers)
1069                 j_der_chp(kplayers_der)= jo(iplayers)*
1070     x_der_chp(kplayers_der) / x(iplayers)
1071                 j_der_fcells(kplayers_der)= jo(iplayers)*
1072     x_der_fcells(kplayers_der) / x(iplayers)
1073     !
1074             ELSE
1075     !
1076                 j_der_smhydro(kplayers_der)= 0.0d+00
1077                 j_der_wind(kplayers_der)= 0.0d+00
1078                 j_der_pv(kplayers_der)= 0.0d+00
1079                 j_der_chp(kplayers_der)= 0.0d+00
1080                 j_der_fcells(kplayers_der)= 0.0d+00
1081     !
1082             END IF
1083     !
1084         ELSE
1085     !
1086             j_der_smhydro(kplayers_der)= x_der_smhydro(kplayers_der) *
1087     spot - cost_der_smhydro(kplayers_der)
1088             j_der_wind(kplayers_der)= (x_der_wind(kplayers_der)+
1089     x_der_shortage_wind(kplayers_der)) * spot - cost_der_wind(kplayers_der)
1090             j_der_pv(kplayers_der)= (x_der_pv(kplayers_der)+
1091     x_der_shortage_pv(kplayers_der)) * spot - cost_der_pv(kplayers_der)
1092             j_der_chp(kplayers_der)= x_der_chp(kplayers_der) * spot -
1093     cost_der_chp(kplayers_der)
1094             j_der_fcells(kplayers_der)= x_der_fcells(kplayers_der) * spot
1095     - cost_der_fcells(kplayers_der)
1096     !
1097             END IF
1098     !
1099     ! Average Profit for DER generators
1100             avej_der_smhydro(kplayers_der)= j_der_smhydro(kplayers_der)
1101             avej_der_wind(kplayers_der)= j_der_wind(kplayers_der)
1102             avej_der_pv(kplayers_der)= j_der_pv(kplayers_der)
1103             avej_der_chp(kplayers_der)= j_der_chp(kplayers_der)
1104             avej_der_fcells(kplayers_der)= j_der_fcells(kplayers_der)
1105         END IF
1106     !
1107     END DO
1108     !
1109     ! DSB Generators Profit Calculation
1110     !
1111     kplayers_dsb = 0
1112     DO iplayers = 1, asyn%nplayers
1113     !
1114         IF (asyn%player(iplayers)%category .EQ. CATEGORY(11)) THEN ! dsbaggr
1115     !
1116         kplayers_dsb = kplayers_dsb + 1
1117     !
1118         IF (asyn%player(iplayers)%share_profit .EQ. 0) THEN
1119     !
1120             IF ( ABS(x(iplayers)) .GT. eps_d) THEN
1121     !
1122                 j_dsb_smhydro(kplayers_dsb)= jo(iplayers) *
1123     x_dsb_smhydro(kplayers_dsb) / x(iplayers)
1124                 j_dsb_wind(kplayers_dsb)= jo(iplayers) *
1125     x_dsb_wind(kplayers_dsb) / x(iplayers)
1126                 j_dsb_pv(kplayers_dsb)= jo(iplayers)*
1127     x_dsb_pv(kplayers_dsb) / x(iplayers)
1128                 j_dsb_chp(kplayers_dsb)= jo(iplayers)*
1129     x_dsb_chp(kplayers_dsb) / x(iplayers)
1130                 j_dsb_interrup(kplayers_dsb)=
1131     x_dsb_interrup(kplayers_dsb) * spot - cost_dsb_interrup(kplayers_dsb)
1132     !
1133             ELSE

```

```

1134 !
1135 !                                     j_dsb_smhydro(kplayers_dsb)= 0.0d+00
1136 !                                     j_dsb_wind(kplayers_dsb)= 0.0d+00
1137 !                                     j_dsb_pv(kplayers_dsb)= 0.0d+00
1138 !                                     j_dsb_chp(kplayers_dsb)= 0.0d+00
1139 !                                     j_dsb_interrup(kplayers_dsb)= 0.0d+00
1140 !
1141 !                                     END IF
1142 !
1143 !                                     ELSE
1144 !
1145 !                                     j_dsb_smhydro(kplayers_dsb)= x_dsb_smhydro(kplayers_dsb) *
1146 spot - cost_dsb_smhydro(kplayers_dsb)
1147 !                                     j_dsb_wind(kplayers_dsb)= (x_dsb_wind(kplayers_dsb)+
1148 x_dsb_shortage_wind(kplayers_dsb)) * spot - cost_dsb_wind(kplayers_dsb)
1149 !                                     j_dsb_pv(kplayers_dsb)= (x_dsb_pv(kplayers_dsb)+
1150 x_dsb_shortage_pv(kplayers_dsb)) * spot - cost_dsb_pv(kplayers_dsb)
1151 !                                     j_dsb_chp(kplayers_dsb)= x_dsb_chp(kplayers_dsb) * spot -
1152 cost_dsb_chp(kplayers_dsb)
1153 !                                     j_dsb_interrup(kplayers_dsb)= x_dsb_interrup(kplayers_dsb) *
1154 spot - cost_dsb_interrup(kplayers_dsb)
1155 !
1156 !                                     END IF
1157 !
1158 !
1159 !     Average Profit for dsb generators
1160 !                                     avej_dsb_smhydro(kplayers_dsb)= j_dsb_smhydro(kplayers_dsb)
1161 !                                     avej_dsb_wind(kplayers_dsb)= j_dsb_wind(kplayers_dsb)
1162 !                                     avej_dsb_pv(kplayers_dsb)= j_dsb_pv(kplayers_dsb)
1163 !                                     avej_dsb_chp(kplayers_dsb)= j_dsb_chp(kplayers_dsb)
1164 !                                     avej_dsb_interrup(kplayers_dsb)= j_dsb_interrup(kplayers_dsb)
1165 !                                     END IF
1166 !
1167 !     END DO
1168 !
1169 !     Assign initial values to Bidding Parameters' moving averages
1170 !
1171 !     aveba(:) = ba(:)
1172 !     avebb(:) = bb(:)
1173 !
1174 !     Detailed output of initial state
1175 !
1176 !     IF (gameopt%ngames .EQ. 1) THEN
1177 !
1178 !         asyn%out_one_game(1, 1) = REAL(price, ikind4)
1179 !         asyn%out_one_game(2, 1) = REAL(aveprice, ikind4)
1180 !
1181 !         DO iplayers = 1, asyn%nplayers
1182 !             k = iplayers + 2
1183 !             asyn%out_one_game(k, 1) = REAL(x(iplayers), ikind4)
1184 !         END DO
1185 !
1186 !         DO iplayers = 1, asyn%nplayers
1187 !             k = asyn%nplayers + 2 + iplayers
1188 !             asyn%out_one_game(k, 1) = REAL(avex(iplayers), ikind4)
1189 !         END DO
1190 !
1191 !         DO iplayers = 1, asyn%nplayers
1192 !             k = 2 * asyn%nplayers + 2 + iplayers
1193 !             asyn%out_one_game(k, 1) = REAL(jo(iplayers), ikind4)
1194 !         END DO
1195 !
1196 !         DO iplayers = 1, asyn%nplayers
1197 !             k = 3 * asyn%nplayers + 2 + iplayers
1198 !             asyn%out_one_game(k, 1) = REAL(avej(iplayers), ikind4)
1199 !         END DO
1200 !
1201 !         DO iplayers = 1, asyn%nplayers
1202 !             k = 4 * asyn%nplayers + 2
1203 !             k = k + iplayers + (iplayers - 1)
1204 !             asyn%out_one_game(k, 1) = REAL(ba(iplayers), ikind4)
1205 !             asyn%out_one_game(k+1, 1) = REAL(bb(iplayers), ikind4)
1206 !         END DO
1207 !
1208 !         DO iplayers = 1, asyn%nplayers
1209 !             k = 6 * asyn%nplayers + 2
1210 !             k = k + iplayers + (iplayers - 1)
1211 !             asyn%out_one_game(k, 1) = REAL(aveba(iplayers), ikind4)
1212 !             asyn%out_one_game(k+1, 1) = REAL(avebb(iplayers), ikind4)
1213 !         END DO
1214 !

```



```

1215      k = 8 * asyn%nplayers + 3
1216      asyn%out_one_game(k, 1) = REAL(purchases, ikind4)
1217      asyn%out_one_game(k+1, 1) = REAL(isocost, ikind4)
1218      asyn%out_one_game(k+2, 1) = REAL(extra, ikind4)
1219      asyn%out_one_game(k+3, 1) = REAL(spot, ikind4)
1220      asyn%out_one_game(k+4, 1) = REAL(avespot, ikind4)
1221 !
1222 !
1223      k = 8 * asyn%nplayers + 7
1224 !
1225      DO iplayers = 1, asyn%nplayers_der
1226 !
1227          k = k + (iplayers-1) * 20
1228          asyn%out_one_game(k+1, 1) = REAL(x_der_smhydro(iplayers), ikind4)
1229          asyn%out_one_game(k+2, 1) = REAL(x_der_wind(iplayers), ikind4)
1230          asyn%out_one_game(k+3, 1) = REAL(x_der_pv(iplayers), ikind4)
1231          asyn%out_one_game(k+4, 1) = REAL(x_der_chp(iplayers), ikind4)
1232          asyn%out_one_game(k+5, 1) = REAL(x_der_fcells(iplayers), ikind4)
1233 !
1234          asyn%out_one_game(k+6, 1) = REAL(avex_der_smhydro(iplayers), ikind4)
1235          asyn%out_one_game(k+7, 1) = REAL(avex_der_wind(iplayers), ikind4)
1236          asyn%out_one_game(k+8, 1) = REAL(avex_der_pv(iplayers), ikind4)
1237          asyn%out_one_game(k+9, 1) = REAL(avex_der_chp(iplayers), ikind4)
1238          asyn%out_one_game(k+10, 1) = REAL(avex_der_fcells(iplayers), ikind4)
1239 !
1240          asyn%out_one_game(k+11, 1) = REAL(j_der_smhydro(iplayers), ikind4)
1241          asyn%out_one_game(k+12, 1) = REAL(j_der_wind(iplayers), ikind4)
1242          asyn%out_one_game(k+13, 1) = REAL(j_der_pv(iplayers), ikind4)
1243          asyn%out_one_game(k+14, 1) = REAL(j_der_chp(iplayers), ikind4)
1244          asyn%out_one_game(k+15, 1) = REAL(j_der_fcells(iplayers), ikind4)
1245 !
1246          asyn%out_one_game(k+16, 1) = REAL(avej_der_smhydro(iplayers), ikind4)
1247          asyn%out_one_game(k+17, 1) = REAL(avej_der_wind(iplayers), ikind4)
1248          asyn%out_one_game(k+18, 1) = REAL(avej_der_pv(iplayers), ikind4)
1249          asyn%out_one_game(k+19, 1) = REAL(avej_der_chp(iplayers), ikind4)
1250          asyn%out_one_game(k+20, 1) = REAL(avej_der_fcells(iplayers), ikind4)
1251      ENDDO
1252 !
1253      k = 8 * asyn%nplayers + 7 + 20 * asyn%nplayers_der
1254 !
1255      DO iplayers = 1, asyn%nplayers_dsb
1256 !
1257          k = k + (iplayers-1) * 20
1258          asyn%out_one_game(k+1, 1) = REAL(x_dsb_smhydro(iplayers), ikind4)
1259          asyn%out_one_game(k+2, 1) = REAL(x_dsb_wind(iplayers), ikind4)
1260          asyn%out_one_game(k+3, 1) = REAL(x_dsb_pv(iplayers), ikind4)
1261          asyn%out_one_game(k+4, 1) = REAL(x_dsb_chp(iplayers), ikind4)
1262          asyn%out_one_game(k+5, 1) = REAL(x_dsb_interrup(iplayers), ikind4)
1263 !
1264          asyn%out_one_game(k+6, 1) = REAL(avex_dsb_smhydro(iplayers), ikind4)
1265          asyn%out_one_game(k+7, 1) = REAL(avex_dsb_wind(iplayers), ikind4)
1266          asyn%out_one_game(k+8, 1) = REAL(avex_dsb_pv(iplayers), ikind4)
1267          asyn%out_one_game(k+9, 1) = REAL(avex_dsb_chp(iplayers), ikind4)
1268          asyn%out_one_game(k+10, 1) = REAL(avex_dsb_interrup(iplayers), ikind4)
1269 !
1270          asyn%out_one_game(k+11, 1) = REAL(j_dsb_smhydro(iplayers), ikind4)
1271          asyn%out_one_game(k+12, 1) = REAL(j_dsb_wind(iplayers), ikind4)
1272          asyn%out_one_game(k+13, 1) = REAL(j_dsb_pv(iplayers), ikind4)
1273          asyn%out_one_game(k+14, 1) = REAL(j_dsb_chp(iplayers), ikind4)
1274          asyn%out_one_game(k+15, 1) = REAL(j_dsb_interrup(iplayers), ikind4)
1275 !
1276          asyn%out_one_game(k+16, 1) = REAL(avej_dsb_smhydro(iplayers), ikind4)
1277          asyn%out_one_game(k+17, 1) = REAL(avej_dsb_wind(iplayers), ikind4)
1278          asyn%out_one_game(k+18, 1) = REAL(avej_dsb_pv(iplayers), ikind4)
1279          asyn%out_one_game(k+19, 1) = REAL(avej_dsb_chp(iplayers), ikind4)
1280          asyn%out_one_game(k+20, 1) = REAL(avej_dsb_interrup(iplayers), ikind4)
1281      ENDDO
1282 !
1283      END IF
1284 !
1285 !-----
1286 !
1287      player_count(:) = 0
1288 !
1289      DO iterations = 1, asyn%niterations-1
1290 !
1291          CALL random_number (y)
1292 !
1293          d = asyn%inid * (1.0d+00 - asyn%dd + 2.0d+00 * asyn%dd * y)
1294 !
1295 ! Check for available power with respect to the demand

```

```

1296         availpower = 0.0d+00
1297 !
1298         DO iplayers = 1, asyn%nplayers
1299 !
1300 ! dsb-start
1301             IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
1302                 availpower = availpower + asyn%player(iplayers)%xmax
1303             ENDIF
1304 ! dsb-start
1305
1306         END DO
1307 !
1308         IF (availpower .LT. d ) THEN
1309 !
1310             kerr = ERR_DEMAND
1311             RETURN
1312 !
1313         END IF
1314 !
1315 ! Random desicion & optimization
1316 !
1317 ! Assign initial values (=zero) to Sums related with the Bid Parameters
1318 !
1319         sumab = 0.0
1320         sumlb = 0.0
1321 !
1322 ! Start iterative procedure for each player
1323 !
1324         DO iplayers = 1, asyn%nplayers
1325 !
1326             sumper1 = asyn%player(iplayers)%period(1)
1327             sumper2 = asyn%player(iplayers)%period(1) + asyn%player(iplayers)%period(2)
1328 !
1329             IF (player_count(iplayers) .GE. sumper2) THEN
1330                 player_count(iplayers) = 0
1331             END IF
1332 !
1333             player_count(iplayers) = player_count(iplayers) + 1
1334 !
1335 ! Decision block that assigns value only to one of the BA,BB parameter's step
1336 ! (and keep the others zero) depending on the current parameter adaption period
1337 !
1338             IF (player_count(iplayers) .LE. sumper1) THEN
1339
1340                 coef_flag(iplayers) = 1
1341 !
1342                 IF (iiterations .GT. INT(asyn%estep_decr * (asyn%niterations))) THEN !IF THEN
1343 Block that
1344
1345 !
1346                     eps(iplayers, 1) = asyn%player(iplayers)%e_init(1) * &
1347                         (asyn%estep_decr * REAL(asyn%niterations)) / REAL(iiterations + 1)
1348 !gradually decreases
1349 !
1350                     ELSE !the step size of BA
1351 !
1352                         eps(iplayers, 1) = asyn%player(iplayers)%e_init(1)
1353 !
1354                     END IF
1355 !
1356                     eps(iplayers, 2) = 0.0d+00
1357 !
1358                 ELSE IF (player_count(iplayers) .LE. sumper2) THEN
1359                     coef_flag(iplayers) = 2
1360                     eps(iplayers, 1) = 0.0d+00
1361 !
1362                     IF (iiterations .GT. INT(asyn%estep_decr * (asyn%niterations))) THEN !IF THEN
1363 Block that
1364 !
1365                         eps(iplayers, 2) = asyn%player(iplayers)%e_init(2) * &
1366                             (asyn%estep_decr * REAL(asyn%niterations)) / REAL(iiterations + 1)
1367 !gradually decreases
1368 !
1369                         ELSE !the step size of BB
1370 !
1371                             eps(iplayers, 2) = asyn%player(iplayers)%e_init(2)
1372 !
1373                         END IF
1374 !
1375                     END IF
1376 !

```

```

1377 !      Decision block that assigns value to theta
1378 !
1379 !      IF (iterations .GT. INT(asyn%thstep_decr * (asyn%niterations))) THEN !IF THEN
1380 Block that
1381 !
1382 !      theta(iplayers) = asyn%player(iplayers)%theta_ini * (asyn%thstep_decr *
1383 &
1384 !      REAL(asyn%niterations)) / REAL(iterations + 1) !gradualy
1385 decreases
1386 !
1387 !      ELSE ! theta step size
1388 !
1389 !      theta(iplayers) = asyn%player(iplayers)%theta_ini
1390 !
1391 !      END IF
1392 !
1393 !      Choose a random value from the probabilistic action profile
1394 !      for the in turn parameter (BA or BB)and
1395 !      adjust the in turn parameter (BA or BB) in compliance with
1396 !      the randomly selected value
1397 !
1398 !      CALL random_number (y)
1399 !      rnd = y
1400 !      p1 = pi(iplayers, coef_flag(iplayers))
1401 !      p2 = pi(iplayers, coef_flag(iplayers)) +
1402 !      pd(iplayers, coef_flag(iplayers)) &
1403 !
1404 !      IF (rnd .LE. p1) THEN ! increase
1405 !
1406 !      action(iplayers, coef_flag(iplayers)) = 1
1407 !      ba(iplayers) = ba(iplayers) + eps(iplayers, 1) * ba(iplayers)
1408 !      bb(iplayers) = bb(iplayers) + eps(iplayers, 2) * bb(iplayers)
1409 !
1410 !      ELSE IF (rnd .LE. p2) THEN ! decrease
1411 !
1412 !      action(iplayers, coef_flag(iplayers)) = 2
1413 !      ba(iplayers) = ba(iplayers) - eps(iplayers, 1) * ba(iplayers)
1414 !      bb(iplayers) = bb(iplayers) - eps(iplayers, 2) * bb(iplayers)
1415 !
1416 !      ELSE ! stabilize
1417 !
1418 !      action(iplayers, coef_flag(iplayers)) = 3
1419 !      da(iplayers) = 0.0d+00
1420 !      ba(iplayers) = ba(iplayers) + da(iplayers)
1421 !      bb(iplayers) = bb(iplayers) + da(iplayers)
1422 !
1423 !      END IF
1424 !      IF (asyn%player(iplayers)%category .EQ. CATEGORY(11) .AND. ba(iplayers)
1425 .LT.0) ba(iplayers) = 0.0
1426 !
1427 !
1428 !      Definition of the current maximum and minimum allowed values for the BA, BB parameters
1429 !
1430 !      bamax(iplayers) = asyn%pricecap - (bb(iplayers) * asyn%player(iplayers)%xmax)
1431 !      bbmax(iplayers) = (asyn%pricecap - ba(iplayers)) / asyn%player(iplayers)%xmax
1432 !
1433 !      Limit the bid curve up to the Price Cap
1434 !
1435 !      IF (coef_flag(iplayers) .EQ. 1) THEN
1436 !      IF (ba(iplayers) .GT. bamax(iplayers)) THEN
1437 !      ba(iplayers) = bamax(iplayers)
1438 !      END IF
1439 !      IF (ba(iplayers) .LT. bmin) ba(iplayers)=bmin
1440 !      ELSE IF (coef_flag(iplayers) .EQ. 2) THEN
1441 !      IF (bb(iplayers) .GT. bbmax(iplayers)) bb(iplayers) = bbmax(iplayers)
1442 !      IF (bb(iplayers) .LT. bmin) bb(iplayers) = bmin
1443 !      END IF
1444 !
1445 !      Calculate the sums related with the bidding parameters
1446 !
1447 !      dsb-start
1448 !      IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
1449 !      sumab = sumab + (ba(iplayers) / bb(iplayers))
1450 !      sumlb = sumlb + (1.0d+00 / bb(iplayers))
1451 !      END IF
1452 !      dsb-start
1453 !
1454 !      lp(iplayers, 1)= ba(iplayers) + bb(iplayers) * asyn%player(iplayers)%xmin
1455 !      lp(iplayers, 2)= ba(iplayers) + bb(iplayers) * asyn%player(iplayers)%xmax
1456 !
1457 !      Calculation of bidding Parameters' Moving Averages

```

```

1458 !
1459     aveba(iplayers) = aveba(iplayers) + (ba(iplayers) - &
1460         aveba(iplayers)) / REAL(iiterations + 1)
1461     avebb(iplayers) = avebb(iplayers) + (bb(iplayers) - &
1462         avebb(iplayers)) / REAL(iiterations+1)
1463 !
1464     ENDDO
1465 !
1466 !     Calculation of the Spot Price
1467 !
1468     counter = 0
1469 !
1470 !     Definition of the 2xPLAYERS nodal points of the aggregate Supply Curve
1471 !     and calculation of the available quantity (Qx)for each player at every point
1472 !
1473     DO iconst = 1, NCONST
1474 !
1475         DO iplayers = 1, asyn%nplayers
1476             IF (asyn%player(iplayers)%category .eq. CATEGORY(11) ) cycle
1477             counter = counter + 1
1478             lpoint(counter) = lp(iplayers, iconst) !assignment of all vallues
1479             qpoint(counter) = 0
1480             DO jplayers = 1, asyn%nplayers
1481                 IF (asyn%player(jplayers)%category .eq. CATEGORY(11) ) cycle
1482                 IF ( lpoint(counter) .LE. lp(jplayers, 1)) THEN !If the point is below
1483 player's min
1484                     qx(counter, jplayers) = 0.0d+00
1485                 ELSE IF (lpoint(counter) .GE. lp(jplayers, 2)) THEN !If the point is above
1486 player's max
1487                     qx(counter, jplayers) = asyn%player(jplayers)%xmax
1488                 ELSE
1489                     qx(counter,jplayers) = (lpoint(counter) - ba(jplayers)) / &
1490                         bb(jplayers) !The point between player's min
1491 and max
1492                 END IF
1493 !
1494 !                 Also calculation of the total available
1495 quantity (QPoint) at each nodal point
1496                     qpoint(counter) = qpoint(counter) + qx(counter, jplayers)
1497                 END DO
1498             END DO
1499         END DO
1500         knees = counter
1501         if (asyn%nplayers_dsb .ne. 0 ) then
1502             DO iknee= 1, knees
1503 !
1504                 dpoint(iknee) = ba_dsb + slope * (asyn%inid - qpoint(iknee))
1505 !
1506             END DO
1507
1508
1509             CALL DSVRGN (knees, qpoint, qpoint_sorted)
1510 !
1511             DO iknee= 1, knees
1512                 DO jknee= 1, knees
1513                     IF ( ABS( qpoint_sorted(iknee)-qpoint(jknee) ) .LE. eps_d )
1514 THEN
1515 !
1516                         lpoint_sorted(iknee) = lpoint(jknee)
1517                         dpoint_sorted(iknee) = dpoint(jknee)
1518                         EXIT
1519 !
1520                     END IF
1521                 END DO
1522             END DO
1523         END DO
1524 !
1525         DO iknee= 1, knees-1
1526 !
1527             diff1 = dpoint_sorted(iknee) - lpoint_sorted(iknee)
1528             diff2 = dpoint_sorted(iknee+1) - lpoint_sorted(iknee+1)
1529
1530             IF ( abs(diff1) .LE. eps_d .OR. abs(diff2) .LE. eps_d ) THEN
1531
1532                 IF ( abs(diff1) .LE. eps_d ) THEN
1533                     counter = iknee
1534                 ELSE
1535                     counter = iknee+1
1536                 ENDIF
1537 !
1538             l = lpoint(counter)

```

```

1539
1540         DO jplayers = 1, asyn%nplayers
1541
1542             IF (asyn%player(jplayers)%category .NE. CATEGORY(11) )
1543 THEN
1544                 xx(jplayers) = qx(counter, jplayers)
1545                 IF ( xx(jplayers) .LE.
1546 asyn%player(jplayers)%xmin ) xx(jplayers) = 0.0
1547                 IF ( xx(jplayers) .GE.
1548 asyn%player(jplayers)%xmax ) xx(jplayers) = asyn%player(jplayers)%xmax
1549
1550                 ELSE
1551                     xx(jplayers) = asyn%inid-q
1552 !                     xx(jplayers) = asyn%inid - (1 - ba_dsb) /slope
1553
1554                     IF ( xx(jplayers) .LE.
1555 asyn%player(jplayers)%xmin ) xx(jplayers) = 0.0
1556                     IF ( xx(jplayers) .GE.
1557 asyn%player(jplayers)%xmax ) xx(jplayers) = asyn%player(jplayers)%xmax
1558                     END IF
1559                 END DO
1560 !
1561                 GOTO 555
1562
1563             END IF
1564
1565             IF (diff1*diff2 .LT. 0.0) THEN
1566                 jknee =iknee
1567                 EXIT ! exoume tomi
1568             END IF
1569 !
1570         END DO
1571
1572         q1 = qpoint_sorted(jknee)
1573         q2 = qpoint_sorted(jknee+1)
1574 !
1575         l1 = lpoint_sorted(jknee)
1576         l2 = lpoint_sorted(jknee+1)
1577 !
1578         d1 = dpoint_sorted(jknee)
1579         d2 = dpoint_sorted(jknee+1)
1580 !
1581         q = ((d2-l2)*q1 - (d1-l1)*q2) / (l1-l2-(d1-d2))
1582 !
1583         IF (q .LE. d .AND. q .GE. d-dsb_capacity) THEN
1584
1585             l = (l1-l2) * q / (q1-q2) + (l2*q1-l1*q2) / (q1-q2)
1586
1587         ELSE
1588             IF (q .GT. d) THEN
1589                 q = d
1590             ELSE
1591                 q = d-dsb_capacity
1592             END IF
1593 !
1594         DO iknee= 1, knees-1
1595 !
1596             diff1 = q - dpoint_sorted(iknee)
1597             diff2 = q - dpoint_sorted(iknee+1)
1598             IF (diff1*diff2 .LT. 0.0) THEN
1599                 jknee =iknee
1600                 EXIT ! exoume tomi
1601             END IF
1602         END DO
1603
1604         q1 = qpoint_sorted(jknee)
1605         q2 = qpoint_sorted(jknee+1)
1606 !
1607         l1 = lpoint_sorted(jknee)
1608         l2 = lpoint_sorted(jknee+1)
1609 !
1610         l = (l1-l2) * q / (q1-q2) + (l2*q1-l1*q2) / (q1-q2)
1611     END IF
1612
1613
1614
1615
1616     DO iplayers = 1, asyn%nplayers
1617
1618         IF (asyn%player(iplayers)%category .NE. CATEGORY(11) ) THEN
1619

```

```

1620             xx(iplayers) = (1 - ba(iplayers)) / bb(iplayers)
1621             IF ( xx(iplayers) .LE. asyn%player(iplayers)%xmin )
1622 xx(iplayers) = 0.0
1623             IF ( xx(iplayers) .GE. asyn%player(iplayers)%xmax )
1624 xx(iplayers) = asyn%player(iplayers)%xmax
1625
1626             ELSE
1627 !
1628             xx(iplayers) = asyn%inid-q
1629 !             xx(iplayers) = asyn%inid - (1 - ba_dsb) /slope
1630 !
1631             IF ( xx(iplayers) .LE. asyn%player(iplayers)%xmin )
1632 xx(iplayers) = 0.0
1633             IF ( xx(iplayers) .GE. asyn%player(iplayers)%xmax )
1634 xx(iplayers) = asyn%player(iplayers)%xmax
1635
1636             END IF
1637             end do
1638 !
1639             555 continue
1640 !
1641 else
1642 !
1643             zpos = REAL(asyn%nplayers) * d + 1.0D+00
1644             zneg = -zpos
1645 !
1646             up = 0
1647             low = 0
1648 !
1649             DO iplayers = 2 * asyn%nplayers, 1, -1
1650                 difd = qpoint(iplayers) - d
1651                 IF (dofd .EQ. 0) THEN
1652                     DO jplayers = 1, asyn%nplayers
1653                         expl(jplayers) = jplayers
1654                         xx(jplayers) = qx(iplayers,
1655 jplayers)
1656                     END DO
1657                     l = lpoint(iplayers)
1658                     GOTO 630
1659                 !
1660                 ELSE IF (dofd .GT. 0.0d+00 ) THEN
1661                     IF (zpos .GT. difd) THEN
1662                         zpos = difd
1663                         up = iplayers
1664                     ELSE IF (ABS(zpos - difd) .LT. eps_d)
1665 THEN
1666                         IF (up .EQ. 0 ) THEN
1667                             kerr = ERR_SOLVE_UP
1668                             RETURN
1669                         END IF
1670                         IF (lpoint(up) .GT.
1671 lpoint(iplayers)) up = iplayers
1672                     END IF
1673                 !
1674                 ELSE
1675                     IF (zneg .LT. difd) THEN
1676                         zneg = difd
1677                         low = iplayers
1678                     ELSE IF (ABS(zneg - difd) .LT. eps_d)
1679 THEN
1680                         IF (low .EQ. 0 ) THEN
1681                             kerr = ERR_SOLVE_LOW
1682                             RETURN
1683                         END IF
1684                         IF (lpoint(low) .LT.
1685 lpoint(iplayers)) low = iplayers
1686                     END IF
1687                 END IF
1688             END DO
1689 !
1690             demx = d
1691             excess = 0
1692 !
1693             DO iplayers = 1, asyn%nplayers
1694                 !
1695                 expl(iplayers) = 0
1696                 !
1697                 IF (low .EQ. 0 ) THEN
1698                     kerr = ERR_SOLVE_LOW
1699                     RETURN
1700                 END IF

```

```

1701          !
1702          IF (up .EQ. 0 ) THEN
1703              kerr = ERR_SOLVE_UP
1704              RETURN
1705          END IF
1706          !
1707          IF (ABS(qx(low, iplayers) -
1708 asyn%player(iplayers)%xmax) .LT. eps_d) THEN
1709              expl(iplayers) = iplayers
1710              xx(iplayers) =
1711 asyn%player(iplayers)%xmax
1712              demx = demx - xx(iplayers)
1713              ! dsb-start
1714              IF (asyn%player(iplayers)%category .NE.
1715 CATEGORY(11) ) THEN
1716                  sumab = sumab - (ba(iplayers) /
1717 bb(iplayers))
1718                  sumlb = sumlb - (1.0d+00 /
1719 bb(iplayers))
1720              END IF
1721              ! dsb-end
1722          ELSE IF (ABS(qx(up, iplayers)) .LT. eps_d) THEN
1723              expl(iplayers) = iplayers
1724              xx(iplayers) = 0.0d+00
1725              ! dsb-start
1726              IF (asyn%player(iplayers)%category .NE.
1727 CATEGORY(11) ) THEN
1728                  sumab = sumab - (ba(iplayers) /
1729 bb(iplayers))
1730                  sumlb = sumlb - (1.0d+00 /
1731 bb(iplayers))
1732              END IF
1733              ! dsb-end
1734              !
1735          ELSE IF ( ABS(qx(low, iplayers)) .LT. eps_d)
1736 THEN
1737          !
1738          IF (qpoint(low) +
1739 asyn%player(iplayers)%xmin .GT. d) THEN
1740          !
1741              excess = iplayers
1742              EXIT
1743          !
1744          END IF
1745          !
1746          END IF
1747          !
1748          END DO
1749          !
1750          IF (excess .GT. 0) THEN
1751              IF (low .EQ. 0 ) THEN
1752                  kerr = ERR_SOLVE_LOW
1753                  RETURN
1754              END IF
1755              l = lpoint(low)
1756              DO iplayers = 1, asyn%nplayers
1757                  expl(iplayers) = iplayers
1758                  xx(iplayers) = qx(low,iplayers)
1759              END DO
1760          !
1761              xx(excess) = asyn%player(excess)%xmin
1762          !
1763          ELSE
1764          !
1765              l = (demx + sumab) / sumlb
1766          !
1767          END IF
1768          !
1769
1770          630 CONTINUE
1771
1772
1773          !      Assign to Players the corresponding quantities [X(I1)] bought
1774 by the ISO
1775          !      Calculation of the quantities' moving averages [AveX(I1)]
1776          !      Calculation of each player's profit [J(I1)] and its moving
1777 average [AveJ(I1)]
1778          !      Calculation of each player's profit variation [DJ(I1)]
1779 compared with
1780          !      previous turn's profit
1781          !      Assign the current profit's value to the Magnitude "Previous

```

```

1782 Turn's Profit" [Jo(I1)]
1783 !
1784 !
1785 ! DO iplayers = 1, asyn%nplayers
1786 ! IF (expl(iplayers) .NE. iplayers)
1787 ! xx(iplayers) = (spot - ba(iplayers))/bb(iplayers)
1788 ! END DO
1789 !
1790 endif
1791 !
1792 ! spot = 1
1793 !
1794 ! Calculation of the Spot Price's Moving Average [AveSPOT]
1795 !
1796 ! avespot = avespot + (spot - avespot) / REAL(iiterations + 1)
1797 !
1798 ! varpr(iiterations + 1) = spot
1799 ! sum_sp = sum_sp + spot
1800 ! sumsp_sq = sumsp_sq + spot**2
1801 !
1802 !
1803 ! purchases= 0
1804 ! isocost = 0
1805 !
1806 ! kplayers_der = 0
1807 ! kplayers_dsb = 0
1808 !
1809 ! Calculation of costs, profits and mving averages for each player
1810 !
1811 ! DO iplayers = 1, asyn%nplayers
1812 !
1813 ! IF (asyn%player(iplayers)%category .EQ. CATEGORY(10)) THEN
1814 !
1815 ! kplayers_der = kplayers_der + 1
1816 ! player_type = PLAYER_DER
1817 !
1818 ! dsb-start
1819 ! ELSE IF (asyn%player(iplayers)%category .EQ. CATEGORY(11) ) THEN
1820 !
1821 ! kplayers_dsb = kplayers_dsb + 1
1822 ! player_type = PLAYER_DSB
1823 ! dsb-end
1824 ! ELSE
1825 !
1826 ! player_type = PLAYER_TYPICAL
1827 !
1828 ! ENDDIF
1829 !
1830 ! x(iplayers) = xx(iplayers)
1831 ! purchases = purchases + x(iplayers)
1832 ! avex(iplayers) = avex(iplayers) + (x(iplayers) - avex(iplayers))/ REAL(iiterations
1833 + 1)
1834 !
1835 ! if player is of DER Aggregator type extra calculations
1836 !
1837 ! dsb-start
1838 ! IF (player_type .EQ. PLAYER_DER .OR. player_type .EQ. PLAYER_DSB)
1839 THEN ! DERaggr
1840 ! dsb-end
1841 !
1842 ! Random for wind
1843 !
1844 ! CALL random_number (y)
1845 !
1846 ! IF (y .LE. asyn%player(iplayers)%wind_no_production) THEN
1847 !
1848 ! wind_prod_coef = 0.0
1849 !
1850 ! ELSEIF (y.GE.asyn%player(iplayers)%wind_max_production) THEN
1851 !
1852 ! wind_prod_coef = 1.0
1853 !
1854 ! ELSE
1855 !
1856 ! wind_prod_coef = (y-
1857 asyn%player(iplayers)%wind_no_production)/ &
1858 (1.0 - asyn%player(iplayers)%wind_no_production -
1859 asyn%player(iplayers)%wind_max_production)
1860 !
1861 ! END IF
1862 !

```



```

1863 wind_production = asyn%player(iplayers)%wind_capacity *
1864 wind_prod_coef
1865
1866 asyn%player(iplayers)%wind_count
1867 !
1868
1869 asyn%player(iplayers)%wind_count
1870 !
1871 ! Random for PV
1872 !
1873 CALL random_number (y)
1874 !
1875 IF (y .LE. asyn%player(iplayers)%pv_no_production) THEN
1876 !
1877     pv_prod_coef = 0.0
1878 !
1879 ELSEIF (y.GE.asyn%player(iplayers)%pv_max_production) THEN
1880 !
1881     pv_prod_coef = 1.0
1882 !
1883 ELSE
1884 !
1885     pv_prod_coef = (y-
1886 asyn%player(iplayers)%pv_no_production)/ &
1887 (1.0 - asyn%player(iplayers)%pv_no_production -
1888 asyn%player(iplayers)%pv_max_production)
1889 !
1890     END IF
1891 !
1892 pv_production = asyn%player(iplayers)%pv_capacity *
1893 pv_prod_coef
1894 total_pv_production = pv_production *
1895 asyn%player(iplayers)%pv_count
1896 !
1897 total_pv_cap = asyn%player(iplayers)%pv_capacity *
1898 asyn%player(iplayers)%pv_count
1899 !
1900     END IF
1901 !
1902     IF (player_type .EQ. PLAYER_DER) THEN ! DERaggr
1903 !
1904 ! DER Internal Merit Order
1905 !
1906     x_der_smhydro(kplayers_der) = 0.0
1907     x_der_wind(kplayers_der) = 0.0
1908     x_der_pv(kplayers_der) = 0.0
1909     x_der_chp(kplayers_der) = 0.0
1910     x_der_fcells(kplayers_der) = 0.0
1911 !
1912     x_rest = x(iplayers)
1913 !
1914 ! Small Hydro
1915     x_der_smhydro(kplayers_der) = MIN(x_rest,
1916 asyn%player(iplayers)%total_smhydro_capacity)
1917     x_rest = x_rest - x_der_smhydro(kplayers_der)
1918     IF (x_rest .LE. 0.0d+00) GOTO 540
1919 !
1920 ! Wind & PV
1921     x_wind_pv = MIN(x_rest, (total_wind_production +
1922 total_pv_production))
1923 !
1924     IF (x_wind_pv .EQ. (total_wind_production +
1925 total_pv_production)) THEN
1926 !
1927         x_der_wind(kplayers_der) = total_wind_production
1928         x_der_pv(kplayers_der) = total_pv_production
1929 !
1930     ELSE
1931 !
1932         x_der_wind(kplayers_der) = MIN(x_wind_pv / 2.0,
1933 total_wind_production)
1934         x_der_pv(kplayers_der) = MIN(x_wind_pv / 2.0,
1935 total_pv_production)
1936 !
1937         IF (x_der_pv(kplayers_der) .EQ. total_pv_production)
1938 THEN
1939 !
1940             x_der_wind(kplayers_der) = x_wind_pv -
1941 x_der_pv(kplayers_der) !MIN(x_wind_pv-x_pv(kplayers_der), total_wind_cap)
1942 !
1943     END IF

```

```

1944 !
1945 IF (x_der_wind(kplayers_der) .EQ. total_wind_production) THEN
1946 x_der_pv(kplayers_der) = x_wind_pv - x_der_wind(kplayers_der)
1947 !MIN(x_wind_pv-x_wind(kplayers_der), total_pv_cap)
1948     END IF
1949     ENDIF
1950 !
1951     x_rest = x_rest - x_der_wind(kplayers_der) -
1952 x_der_pv(kplayers_der)
1953     IF (x_rest .LE. 0.0d+00) GOTO 540
1954 ! CHP
1955     x_der_chp(kplayers_der) = MIN(x_rest,
1956 asyn%player(iplayers)%total_chp_capacity)
1957     x_rest = x_rest - x_der_chp(kplayers_der)
1958     IF (x_rest .LE. 0.0d+00) GOTO 540
1959 ! Fuel Cells
1960     x_der_fcells(kplayers_der) = MIN(x_rest,
1961 asyn%player(iplayers)%total_fcells_capacity)
1962     x_rest = x_rest - x_der_fcells(kplayers_der)
1963     540 CONTINUE
1964 !
1965 ! Average Quantities for DER generators
1966 !
1967     avex_der_smhydro(kplayers_der)= avex_der_smhydro(kplayers_der)
1968 + (x_der_smhydro(kplayers_der) - avex_der_smhydro(kplayers_der))/ REAL(iiterations + 1)
1969     avex_der_wind(kplayers_der)= avex_der_wind(kplayers_der) +
1970 (x_der_wind(kplayers_der) - avex_der_wind(kplayers_der))/ REAL(iiterations + 1)
1971     avex_der_pv(kplayers_der)= avex_der_pv(kplayers_der) +
1972 (x_der_pv(kplayers_der) - avex_der_pv(kplayers_der))/ REAL(iiterations + 1)
1973     avex_der_chp(kplayers_der)= avex_der_chp(kplayers_der) +
1974 (x_der_chp(kplayers_der) - avex_der_chp(kplayers_der))/ REAL(iiterations + 1)
1975     avex_der_fcells(kplayers_der)= avex_der_fcells(kplayers_der) +
1976 (x_der_fcells(kplayers_der) - avex_der_fcells(kplayers_der))/ REAL(iiterations + 1)
1977
1978 ! Shortage
1979     IF (x(iplayers).GT. (asyn%player(iplayers)%total_der_capacity
1980 - asyn%player(iplayers)%total_intermittent_capacity + &
1981     total_wind_production + total_pv_production))
1982 THEN
1983 !
1984     x_shortage = x(iplayers) -
1985 (asyn%player(iplayers)%total_der_capacity - asyn%player(iplayers)%total_intermittent_capacity + &
1986     total_wind_production + total_pv_production)
1987 !
1988     IF (total_wind_production .LT.
1989 asyn%player(iplayers)%wind_estim_capacity &
1990     .AND. total_pv_production .GE.
1991 asyn%player(iplayers)%pv_estim_capacity ) THEN
1992 !
1993     x_der_shortage_wind(kplayers_der) =
1994 x_shortage
1995     x_der_shortage_pv(kplayers_der) = 0.0
1996 !
1997     ELSE IF (total_wind_production .GE.
1998 asyn%player(iplayers)%wind_estim_capacity &
1999     .AND. total_pv_production .LT.
2000 asyn%player(iplayers)%pv_estim_capacity) THEN
2001 !
2002     x_der_shortage_wind(kplayers_der) = 0.0
2003     x_der_shortage_pv(kplayers_der) =
2004 x_shortage
2005 !
2006     ELSE
2007 !
2008     x_der_shortage_wind(kplayers_der) =
2009 asyn%player(iplayers)%wind_estim_capacity - total_wind_production
2010     x_der_shortage_pv(kplayers_der) =
2011 x_shortage - x_der_shortage_wind(kplayers_der)
2012 !
2013     END IF
2014 !
2015     ELSE
2016 !
2017     x_shortage = 0.0
2018     x_der_shortage_pv(kplayers_der) = 0.0
2019     x_der_shortage_wind(kplayers_der) = 0.0
2020     END IF
2021 !
2022 ! DER Costs Calculation
2023 !
2024     cost_der_smhydro(kplayers_der) = 0.0

```

```

2025 !
2026 cost_der_wind(kplayers_der) =
2027 x_der_shortage_wind(kplayers_der) * spot * (1.0d+00 + asyn%player(iplayers)%assurance_cost)
2028 !
2029 cost_der_pv(kplayers_der) = x_der_shortage_pv(kplayers_der) *
2030 spot * (1.0d+00 + asyn%player(iplayers)%assurance_cost)
2031 !
2032 cost_der_chp(kplayers_der) = (asyn%player(iplayers)%fc +
2033 asyn%player(iplayers)%a * x_der_chp(kplayers_der) + &
2034
2035 asyn%player(iplayers)%b * x_der_chp(kplayers_der) * x_der_chp(kplayers_der)) * (1.0 -
2036 asyn%player(iplayers)%chp_savings)
2037 !
2038 cost_der_fcells(kplayers_der) = (asyn%player(iplayers)%fc +
2039 asyn%player(iplayers)%a * x_der_fcells(kplayers_der) + &
2040
2041 asyn%player(iplayers)%b * x_der_fcells(kplayers_der) * x_der_fcells(kplayers_der)) &
2042 * (1.0 -
2043 asyn%player(iplayers)%chp_savings) * (1.0 + asyn%player(iplayers)%fcells_pluscost)
2044 !
2045 player_cost = cost_der_smhydro(kplayers_der) +
2046 cost_der_wind(kplayers_der) + cost_der_pv(kplayers_der) + cost_der_chp(kplayers_der) +
2047 cost_der_fcells(kplayers_der)
2048 !
2049 ELSE IF (player_type .EQ. PLAYER_DSB) THEN ! DSBaggr
2050 ! DER Internal Merit Order
2051 x_dsb_smhydro(kplayers_dsb) = 0.0
2052 x_dsb_wind(kplayers_dsb) = 0.0
2053 x_dsb_pv(kplayers_dsb) = 0.0
2054 x_dsb_chp(kplayers_dsb) = 0.0
2055 x_dsb_interrup(kplayers_dsb) = 0.0
2056 !
2057 x_rest = x(iplayers)
2058 !
2059 ! Small Hydro
2060 x_dsb_smhydro(kplayers_dsb) = MIN(x_rest,
2061 asyn%player(iplayers)%total_smhydro_capacity)
2062 x_rest = x_rest - x_dsb_smhydro(kplayers_dsb)
2063 IF (x_rest .LE. 0.0d+00) GOTO 550
2064 ! Wind & PV
2065 x_wind_pv = MIN(x_rest, (total_wind_production + total_pv_production))
2066 !
2067 IF (x_wind_pv .EQ. (total_wind_production + total_pv_production)) THEN
2068 !
2069 x_dsb_wind(kplayers_dsb) = total_wind_production
2070 x_dsb_pv(kplayers_dsb) = total_pv_production
2071 ELSE
2072 !
2073 x_dsb_wind(kplayers_dsb) = MIN(x_wind_pv / 2.0,
2074 total_wind_production)
2075 x_dsb_pv(kplayers_dsb) = MIN(x_wind_pv / 2.0,
2076 total_pv_production)
2077 !
2078 IF (x_dsb_pv(kplayers_dsb) .EQ. total_pv_production)
2079 THEN
2080 !
2081 x_dsb_wind(kplayers_dsb) = x_wind_pv -
2082 x_dsb_pv(kplayers_dsb) !MIN(x_wind_pv-x_pv(kplayers_dsb), total_wind_cap)
2083
2084 END IF
2085 !
2086 IF (x_dsb_wind(kplayers_dsb) .EQ. total_wind_production) THEN
2087 !
2088 x_dsb_pv(kplayers_dsb) = x_wind_pv -
2089 x_dsb_wind(kplayers_dsb) !MIN(x_wind_pv-x_wind(kplayers_dsb), total_pv_cap)
2090 !
2091 END IF
2092 !
2093 ENDF
2094 !
2095 x_rest = x_rest - x_dsb_wind(kplayers_dsb) -
2096 x_dsb_pv(kplayers_dsb)
2097 IF (x_rest .LE. 0.0d+00) GOTO 550
2098 !
2099 ! CHP
2100 x_dsb_chp(kplayers_dsb) = MIN(x_rest,
2101 asyn%player(iplayers)%total_chp_capacity)
2102 x_rest = x_rest - x_dsb_chp(kplayers_dsb)
2103 IF (x_rest .LE. 0.0d+00) GOTO 550
2104 !
2105 ! Fuel Cells

```

```

2106             x_dsb_interrup(kplayers_dsb) = MIN(x_rest,
2107 asyn%player(iplayers)%interrup_capacity)
2108             x_rest = x_rest - x_dsb_interrup(kplayers_dsb)
2109             550 CONTINUE
2110 !
2111 ! Average Quantities for dsb generators
2112 !
2113             avex_dsb_smhydro(kplayers_dsb)= avex_dsb_smhydro(kplayers_dsb)
2114 + (x_dsb_smhydro(kplayers_dsb) - avex_dsb_smhydro(kplayers_dsb))/ REAL(iiterations + 1)
2115             avex_dsb_wind(kplayers_dsb)= avex_dsb_wind(kplayers_dsb) +
2116 (x_dsb_wind(kplayers_dsb) - avex_dsb_wind(kplayers_dsb))/ REAL(iiterations + 1)
2117             avex_dsb_pv(kplayers_dsb)= avex_dsb_pv(kplayers_dsb) +
2118 (x_dsb_pv(kplayers_dsb) - avex_dsb_pv(kplayers_dsb))/ REAL(iiterations + 1)
2119             avex_dsb_chp(kplayers_dsb)= avex_dsb_chp(kplayers_dsb) +
2120 (x_dsb_chp(kplayers_dsb) - avex_dsb_chp(kplayers_dsb))/ REAL(iiterations + 1)
2121             avex_dsb_interrup(kplayers_dsb)=
2122 avex_dsb_interrup(kplayers_dsb) + (x_dsb_interrup(kplayers_dsb) - avex_dsb_interrup(kplayers_dsb))/
2123 REAL(iiterations + 1)
2124 !
2125 ! Shortage
2126             IF (x(iplayers).GT.
2127 (asyn%player(iplayers)%total_dsb_dergen_capacity -
2128 asyn%player(iplayers)%total_intermittent_capacity + &
2129             total_wind_production + total_pv_production)) THEN
2130 !
2131             x_shortage = x(iplayers) -
2132 (asyn%player(iplayers)%total_dsb_dergen_capacity -
2133 asyn%player(iplayers)%total_intermittent_capacity + &
2134             total_wind_production + total_pv_production)
2135 !
2136             IF (x_dsb_interrup(kplayers_dsb) .LT.
2137 asyn%player(iplayers)%interrup_capacity) THEN
2138 !
2139             xavail =
2140 asyn%player(iplayers)%interrup_capacity - x_dsb_interrup(kplayers_dsb)
2141 !
2142             IF (x_shortage .LE. xavail) THEN
2143 !
2144             x_dsb_interrup(kplayers_dsb) =
2145 x_dsb_interrup(kplayers_dsb) + x_shortage
2146             x_shortage = 0.0
2147
2148             ELSE
2149             x_dsb_interrup(kplayers_dsb) =
2150 asyn%player(iplayers)%interrup_capacity
2151             x_shortage = x_shortage - xavail
2152             ENDIF
2153             END IF
2154 !
2155             IF (x_shortage .GT. 0.0) THEN
2156 !
2157             IF ((total_wind_production .LT.
2158 asyn%player(iplayers)%wind_estim_capacity) &
2159             .AND. (total_pv_production .GE.
2160 asyn%player(iplayers)%pv_estim_capacity) ) THEN
2161 !
2162             x_dsb_shortage_wind(kplayers_dsb) =
2163 x_shortage
2164             x_dsb_shortage_pv(kplayers_dsb) = 0.0
2165             ELSE IF (total_wind_production .GE.
2166 asyn%player(iplayers)%wind_estim_capacity &
2167             .AND. total_pv_production .LT.
2168 asyn%player(iplayers)%pv_estim_capacity) THEN
2169 !
2170             x_dsb_shortage_wind(kplayers_dsb) = 0.0
2171             x_dsb_shortage_pv(kplayers_dsb) =
2172 x_shortage
2173 !
2174             ELSE
2175             x_dsb_shortage_wind(kplayers_dsb) =
2176 asyn%player(iplayers)%wind_estim_capacity - total_wind_production
2177             x_dsb_shortage_pv(kplayers_dsb) =
2178 x_shortage - x_dsb_shortage_wind(kplayers_dsb)
2179 !
2180             END IF
2181 !
2182             END IF
2183 !
2184             ELSE
2185             x_shortage = 0.0
2186             x_dsb_shortage_pv(kplayers_dsb) = 0.0

```

```

2187         x_dsb_shortage_wind(kplayers_dsb) = 0.0
2188     END IF
2189 !
2190 ! dsb Costs Calculation
2191         cost_dsb_smhydro(kplayers_dsb) = 0.0
2192 !
2193         cost_dsb_wind(kplayers_dsb) = x_dsb_shortage_wind(kplayers_dsb) *
2194 spot * (1.0d+00 + asyn%player(iplayers)%assurance_cost)
2195 !
2196         cost_dsb_pv(kplayers_dsb) = x_dsb_shortage_pv(kplayers_dsb) *
2197 spot * (1.0d+00 + asyn%player(iplayers)%assurance_cost)
2198 !
2199         cost_dsb_chp(kplayers_dsb) = (asyn%player(iplayers)%fc +
2200 asyn%player(iplayers)%a * x_dsb_chp(kplayers_dsb) + &
2201         asyn%player(iplayers)%b *
2202 x_dsb_chp(kplayers_dsb) * x_dsb_chp(kplayers_dsb)) *
2203         &
2204 (1.0 - asyn%player(iplayers)%chp_savings)
2205 !
2206         cost_dsb_interrup(kplayers_dsb) =
2207 asyn%player(iplayers)%interrup_low_price * x_dsb_interrup(kplayers_dsb) + &
2208         0.5d+00 *
2209 asyn%player(iplayers)%curve_slope * x_dsb_interrup(kplayers_dsb)
2210 !
2211         player_cost = cost_dsb_smhydro(kplayers_dsb) +
2212 cost_dsb_wind(kplayers_dsb) + cost_dsb_pv(kplayers_dsb) + &
2213         cost_dsb_chp(kplayers_dsb) +
2214 cost_dsb_interrup(kplayers_dsb)
2215 !
2216 ! dsb-end
2217 !
2218     ELSE ! Common Player
2219 !
2220         player_cost = asyn%player(iplayers)%fc +
2221 asyn%player(iplayers)%a * x(iplayers) + &
2222         asyn%player(iplayers)%b * x(iplayers)
2223 * x(iplayers)
2224     END IF
2225 !
2226 ! Pricing Method, Revenues and Profits
2227     IF (asyn%pr_method .EQ. 0) THEN
2228 ! Uniform Pricing
2229         rev(iplayers) = spot * x(iplayers)
2230     ELSE
2231 ! Pay-as-Bid
2232 !
2233         IF ( ABS(x(iplayers)) .LE. eps_d ) THEN
2234             rev(iplayers) = 0.0d+00
2235         ELSE
2236 !
2237             rev(iplayers) = ba(iplayers) * x(iplayers) + (bb(iplayers) /
2238 2.0d+00) * x(iplayers) * x(iplayers) +
2239             &
2240             (bb(iplayers) / 2.0d+00) *
2241 asyn%player(iplayers)%xmin * asyn%player(iplayers)%xmin
2242 !
2243         END IF
2244 !
2245         j(iplayers) = rev(iplayers) - player_cost
2246         isocost = isocost + rev(iplayers)
2247         avej(iplayers) = avej(iplayers) + (j(iplayers) - avej(iplayers)) /
2248 REAL(iiterations + 1)
2249         dj(iplayers) = j(iplayers) - jo(iplayers)
2250         jo(iplayers) = j(iplayers)
2251     END DO
2252 !
2253     extra = purchases - d
2254     kwh = isocost / purchases
2255     avekwh = avekwh + (kwh - avekwh) / REAL(iiterations + 1)
2256 !
2257 ! Spot Price & Price of electricity
2258 !
2259     IF (asyn%pr_method .EQ. 0) THEN ! Uniform Pricing
2260         price = spot
2261         aveprice = avespot
2262     ELSE
2263 ! Pay-as-Bid
2264         price = kwh
2265         aveprice = avekwh
2266 !
2267     END IF

```

```

2268 !
2269     sum_purchases = sum_purchases + purchases
2270     sum_extra = sum_extra + extra
2271     sum_isocost = sum_isocost + isocost
2272 !
2273 !!     DER Generators Profit Calculation
2274 !
2275         kplayers_der = 0
2276         DO iplayers = 1, asyn%nplayers
2277 !
2278             IF (asyn%player(iplayers)%category .EQ. CATEGORY(10)) THEN ! DERaggr
2279 !
2280                 kplayers_der = kplayers_der + 1
2281 !
2282                 IF (asyn%player(iplayers)%share_profit .EQ. 0) THEN
2283 !
2284                     IF ( ABS(x(iplayers)) .GT. eps_d) THEN
2285                         j_der_smhydro(kplayers_der)= j(iplayers) *
2286 x_der_smhydro(kplayers_der) / x(iplayers)
2287                         j_der_wind(kplayers_der)= j(iplayers) *
2288 x_der_wind(kplayers_der) / x(iplayers)
2289                         j_der_pv(kplayers_der)= j(iplayers)*
2290 x_der_pv(kplayers_der) / x(iplayers)
2291                         j_der_chp(kplayers_der)= j(iplayers)*
2292 x_der_chp(kplayers_der) / x(iplayers)
2293                         j_der_fcells(kplayers_der)= j(iplayers)*
2294 x_der_fcells(kplayers_der) / x(iplayers)
2295                     ELSE
2296                         j_der_smhydro(kplayers_der)= 0.0d+00
2297                         j_der_wind(kplayers_der)= 0.0d+00
2298                         j_der_pv(kplayers_der)= 0.0d+00
2299                         j_der_chp(kplayers_der)= 0.0d+00
2300                         j_der_fcells(kplayers_der)= 0.0d+00
2301                     END IF
2302                 ELSE
2303 !
2304                     j_der_smhydro(kplayers_der)=
2305 x_der_smhydro(kplayers_der)* spot - cost_der_smhydro(kplayers_der)
2306                     j_der_wind(kplayers_der)= (x_der_wind(kplayers_der)+
2307 x_der_shortage_wind(kplayers_der))* spot - cost_der_wind(kplayers_der)
2308                     j_der_pv(kplayers_der)= (x_der_pv(kplayers_der)+
2309 x_der_shortage_pv(kplayers_der))* spot - cost_der_pv(kplayers_der)
2310                     j_der_chp(kplayers_der)= x_der_chp(kplayers_der)* spot
2311 - cost_der_chp(kplayers_der)
2312                     j_der_fcells(kplayers_der)=
2313 x_der_fcells(kplayers_der)* spot - cost_der_fcells(kplayers_der)
2314                     END IF
2315 !
2316 !     Average Profit for DER generators
2317                     avej_der_smhydro(kplayers_der)= avej_der_smhydro(kplayers_der)
2318 + (j_der_smhydro(kplayers_der) - avej_der_smhydro(kplayers_der))/ REAL(iiterations + 1)
2319                     avej_der_wind(kplayers_der)= avej_der_wind(kplayers_der) +
2320 (j_der_wind(kplayers_der) - avej_der_wind(kplayers_der))/ REAL(iiterations + 1)
2321                     avej_der_pv(kplayers_der)= avej_der_pv(kplayers_der) +
2322 (j_der_pv(kplayers_der) - avej_der_pv(kplayers_der))/ REAL(iiterations + 1)
2323                     avej_der_chp(kplayers_der)= avej_der_chp(kplayers_der) +
2324 (j_der_chp(kplayers_der) - avej_der_chp(kplayers_der))/ REAL(iiterations + 1)
2325                     avej_der_fcells(kplayers_der)= avej_der_fcells(kplayers_der) +
2326 (j_der_fcells(kplayers_der) - avej_der_fcells(kplayers_der))/ REAL(iiterations + 1)
2327 !
2328                     END IF
2329                 END DO
2330 !
2331 !!     DSB Generators Profit Calculation
2332 !
2333         kplayers_dsb = 0
2334         DO iplayers = 1, asyn%nplayers
2335 !
2336             IF (asyn%player(iplayers)%category .EQ. CATEGORY(11)) THEN ! dsbaggr
2337 !
2338                 kplayers_dsb = kplayers_dsb + 1
2339 !
2340                 IF (asyn%player(iplayers)%share_profit .EQ. 0) THEN
2341 !
2342                     IF ( ABS(x(iplayers)) .GT. eps_d) THEN
2343 !
2344                         j_dsb_smhydro(kplayers_dsb)= jo(iplayers) *
2345 x_dsb_smhydro(kplayers_dsb) / x(iplayers)
2346                         j_dsb_wind(kplayers_dsb)= jo(iplayers) *
2347 x_dsb_wind(kplayers_dsb) / x(iplayers)
2348                         j_dsb_pv(kplayers_dsb)= jo(iplayers)*

```

```

2349 x_dsb_pv(kplayers_dsb) / x(iplayers)
2350                                     j_dsb_chp(kplayers_dsb)= jo(iplayers)*
2351 x_dsb_chp(kplayers_dsb) / x(iplayers)
2352                                     j_dsb_interrup(kplayers_dsb)=
2353 x_dsb_interrup(kplayers_dsb) * spot - cost_dsb_interrup(kplayers_dsb)
2354 !
2355                                     ELSE
2356 !
2357                                     j_dsb_smhydro(kplayers_dsb)= 0.0d+00
2358                                     j_dsb_wind(kplayers_dsb)= 0.0d+00
2359                                     j_dsb_pv(kplayers_dsb)= 0.0d+00
2360                                     j_dsb_chp(kplayers_dsb)= 0.0d+00
2361                                     j_dsb_interrup(kplayers_dsb)= 0.0d+00
2362 !
2363                                     END IF
2364     ELSE
2365         j_dsb_smhydro(kplayers_dsb)=
2366 x_dsb_smhydro(kplayers_dsb) * spot - cost_dsb_smhydro(kplayers_dsb)
2367         j_dsb_wind(kplayers_dsb)= (x_dsb_wind(kplayers_dsb)+
2368 x_dsb_shortage_wind(kplayers_dsb)) * spot - cost_dsb_wind(kplayers_dsb)
2369         j_dsb_pv(kplayers_dsb)= (x_dsb_pv(kplayers_dsb)+
2370 x_dsb_shortage_pv(kplayers_dsb)) * spot - cost_dsb_pv(kplayers_dsb)
2371         j_dsb_chp(kplayers_dsb)= x_dsb_chp(kplayers_dsb) *
2372 spot - cost_dsb_chp(kplayers_dsb)
2373         j_dsb_interrup(kplayers_dsb)=
2374 x_dsb_interrup(kplayers_dsb) * spot - cost_dsb_interrup(kplayers_dsb)
2375 !
2376     END IF
2377 !
2378 !     Average Profit for dsb generators
2379         avej_dsb_smhydro(kplayers_dsb)= avej_dsb_smhydro(kplayers_dsb)
2380 + (j_dsb_smhydro(kplayers_dsb) - avej_dsb_smhydro(kplayers_dsb))/ REAL(iiterations + 1)
2381         avej_dsb_wind(kplayers_dsb)= avej_dsb_wind(kplayers_dsb) +
2382 (j_dsb_wind(kplayers_dsb) - avej_dsb_wind(kplayers_dsb))/ REAL(iiterations + 1)
2383         avej_dsb_pv(kplayers_dsb)= avej_dsb_pv(kplayers_dsb) +
2384 (j_dsb_pv(kplayers_dsb) - avej_dsb_pv(kplayers_dsb))/ REAL(iiterations + 1)
2385         avej_dsb_chp(kplayers_dsb)= avej_dsb_chp(kplayers_dsb) +
2386 (j_dsb_chp(kplayers_dsb) - avej_dsb_chp(kplayers_dsb))/ REAL(iiterations + 1)
2387         avej_dsb_interrup(kplayers_dsb)=
2388 avej_dsb_interrup(kplayers_dsb) + (j_dsb_interrup(kplayers_dsb) - avej_dsb_interrup(kplayers_dsb))/
2389 REAL(iiterations + 1)
2390
2391     END IF
2392 END DO
2393 !
2394 !     Propability correction decision block
2395 !
2396     DO iplayers = 1, asyn%nplayers
2397 !
2398         djt = dj(iplayers)
2399         act = action(iplayers, coef_flag(iplayers))
2400 !
2401         IF (djt .GT. 0.0d+00 .AND. ABS(djt) .GT. eps_d) THEN
2402 !
2403             IF (act .EQ. 1) THEN
2404 !
2405                 IF ( pd(iplayers, coef_flag(iplayers)) .LE. zr .OR.                &
2406 pc(iplayers, coef_flag(iplayers)) .LE. zr ) THEN
2407 !
2408                     IF (pd(iplayers, coef_flag(iplayers)) .LE. zr) THEN
2409 !
2410                         dpi = theta(iplayers) * pc(iplayers, coef_flag(iplayers))
2411 !
2412                         pc(iplayers, coef_flag(iplayers)) =                &
2413 pc(iplayers, coef_flag(iplayers)) - dpi
2414 !
2415                     ELSE
2416 !
2417                         dpi = theta(iplayers) * pd(iplayers,coef_flag(iplayers))
2418 !
2419                         pd(iplayers,coef_flag(iplayers)) =                &
2420 pd(iplayers,coef_flag(iplayers)) - dpi
2421 !
2422                     END IF
2423 !
2424                         pi(iplayers, coef_flag(iplayers)) =                &
2425 pi(iplayers, coef_flag(iplayers)) + dpi
2426 !
2427                     ELSE
2428 !
2429                         dpi = theta(iplayers) * (1.0d+00 -                &

```

```

2430         pi(iplayers, coef_flag(iplayers))) / 2.0d+00
2431 !
2432         dp1 = pd(iplayers, coef_flag(iplayers)) - dpi
2433         dp2 = pc(iplayers, coef_flag(iplayers)) - dpi
2434 !
2435         IF (dp1 .LT. zr .OR. dp2 .LT. zr) THEN
2436 !
2437             dpi = MIN( pd(iplayers, coef_flag(iplayers)),      &
2438                     pc(iplayers,coef_flag(iplayers)) )
2439 !
2440         END IF
2441 !
2442         pd(iplayers, coef_flag(iplayers)) =                &
2443         pd(iplayers, coef_flag(iplayers)) - dpi
2444 !
2445         pc(iplayers, coef_flag(iplayers)) =                &
2446         pc(iplayers, coef_flag(iplayers)) - dpi
2447         pi(iplayers, coef_flag(iplayers)) =                &
2448         pi(iplayers, coef_flag(iplayers)) + 2.0d+00 * dpi
2449 !
2450     END IF
2451 !
2452     ELSE IF (ACT .EQ. 2) THEN
2453 !
2454         IF (pi(iplayers, coef_flag(iplayers)) .LE. zr .OR.    &
2455             pc(iplayers, coef_flag(iplayers)) .LE. zr        ) THEN
2456 !
2457             IF (pi(iplayers, coef_flag(iplayers)) .LE. zr) THEN
2458 !
2459                 dpi = theta(iplayers) * pc(iplayers,coef_flag(iplayers))
2460 !
2461                 pc(iplayers, coef_flag(iplayers)) =          &
2462                 pc(iplayers, coef_flag(iplayers)) - dpi
2463 !
2464             ELSE
2465 !
2466                 dpi = theta(iplayers) * pi(iplayers, coef_flag(iplayers))
2467 !
2468                 pi(iplayers, coef_flag(iplayers)) =          &
2469                 pi(iplayers, coef_flag(iplayers)) - dpi
2470 !
2471             END IF
2472 !
2473             pd(iplayers, coef_flag(iplayers)) =              &
2474             pd(iplayers, coef_flag(iplayers)) + dpi
2475         ELSE
2476             dpi = theta(iplayers) * (1.0d+00 -                &
2477                 pd(iplayers, coef_flag(iplayers))) / 2.0d+00
2478 !
2479             dp1 = pi(iplayers, coef_flag(iplayers)) - dpi
2480             dp2 = pc(iplayers, coef_flag(iplayers)) - dpi
2481 !
2482             IF (dp1 .LT. zr .OR. DP2 .LT. ZR) THEN
2483 !
2484                 dpi = MIN( pi(iplayers, coef_flag(iplayers)), &
2485                           pc(iplayers,coef_flag(iplayers)) )
2486             END IF
2487 !
2488             pi(iplayers, coef_flag(iplayers)) =              &
2489             pi(iplayers, coef_flag(iplayers)) - dpi
2490 !
2491             pc(iplayers, coef_flag(iplayers)) =              &
2492             pc(iplayers, coef_flag(iplayers)) - dpi
2493 !
2494             pd(iplayers, coef_flag(iplayers)) =              &
2495             pd(iplayers, coef_flag(iplayers)) + 2.0 * dpi
2496         END IF
2497     ELSE
2498         IF (pi(iplayers, coef_flag(iplayers)) .LE. zr .OR.    &
2499             pd(iplayers, coef_flag(iplayers)) .LE. zr        ) THEN
2500 !
2501             IF (pi(iplayers, coef_flag(iplayers)) .LE. zr) THEN
2502 !
2503                 dpi = theta(iplayers) * pd(iplayers,coef_flag(iplayers))
2504                 pd(iplayers,coef_flag(iplayers)) =          &
2505                 pd(iplayers,coef_flag(iplayers)) - dpi
2506 !
2507             ELSE
2508 !
2509                 dpi = theta(iplayers) * pi(iplayers, coef_flag(iplayers))
2510 !

```



```

2511         pi(iplayers, coef_flag(iplayers)) = &
2512         pi(iplayers, coef_flag(iplayers)) - dpi
2513 !
2514         END IF
2515 !
2516         pc(iplayers,coef_flag(iplayers)) = &
2517         pc(iplayers,coef_flag(iplayers)) + dpi
2518 !
2519         ELSE
2520 !
2521         dpi = theta(iplayers) * (1.0d+00 - &
2522         pc(iplayers, coef_flag(iplayers))) / 2.0d+00
2523 !
2524         dp1 = pi(iplayers, coef_flag(iplayers)) - dpi
2525         dp2 = pd(iplayers, coef_flag(iplayers)) - dpi
2526 !
2527         IF (dp1 .LT. zr .OR. dp2 .LT. zr) THEN
2528 !
2529         dpi = MIN( pi(iplayers, coef_flag(iplayers)), &
2530         pd(iplayers,coef_flag(iplayers)) )
2531 !
2532         END IF
2533 !
2534         pi(iplayers, coef_flag(iplayers)) = &
2535         pi(iplayers, coef_flag(iplayers)) - dpi
2536 !
2537         pd(iplayers, coef_flag(iplayers)) = &
2538         pd(iplayers, coef_flag(iplayers)) - dpi
2539 !
2540         pc(iplayers, coef_flag(iplayers)) = &
2541         pc(iplayers, coef_flag(iplayers)) + 2.0d+00 * dpi
2542 !
2543         END IF
2544 !
2545         END IF
2546 !
2547         ELSE
2548 !
2549         IF (ACT .EQ. 1) THEN
2550 !
2551         dpi = theta(iplayers) * pi(iplayers,coef_flag(iplayers))
2552 !
2553         pi(iplayers, coef_flag(iplayers)) = &
2554         pi(iplayers, coef_flag(iplayers)) - dpi
2555 !
2556         pd(iplayers, coef_flag(iplayers)) = &
2557         pd(iplayers, coef_flag(iplayers)) + dpi / 2.0d+00
2558 !
2559         pc(iplayers, coef_flag(iplayers)) = &
2560         pc(iplayers, coef_flag(iplayers)) + dpi / 2.0d+00
2561 !
2562         ELSE IF (ACT .EQ. 2) THEN
2563 !
2564         dpi = theta(iplayers) * pd(iplayers, coef_flag(iplayers))
2565 !
2566         pd(iplayers, coef_flag(iplayers)) = &
2567         pd(iplayers, coef_flag(iplayers)) - dpi
2568 !
2569         pi(iplayers, coef_flag(iplayers)) = &
2570         pi(iplayers, coef_flag(iplayers)) + dpi / 2.0d+00
2571 !
2572         pc(iplayers, coef_flag(iplayers)) = &
2573         pc(iplayers, coef_flag(iplayers)) + dpi / 2.0d+00
2574 !
2575         ELSE
2576 !
2577         dpi = theta(iplayers) * pc(iplayers, coef_flag(iplayers))
2578 !
2579         pc(iplayers, coef_flag(iplayers)) = &
2580         pc(iplayers, coef_flag(iplayers)) - dpi
2581 !
2582         pd(iplayers, coef_flag(iplayers)) = &
2583         pd(iplayers, coef_flag(iplayers)) + dpi / 2.0d+00
2584         pi(iplayers, coef_flag(iplayers)) = &
2585         pi(iplayers, coef_flag(iplayers)) + dpi / 2.0d+00
2586 !
2587         END IF
2588 !
2589         END IF
2590 !
2591         ENDDO

```

```

2592 !
2593 !      Output of results
2594 !
2595 !      IF (gameopt%ngames .EQ. 1) THEN
2596 !
2597 !          asyn%out_one_game(1, iterations+1) = REAL(price, ikind4)
2598 !          asyn%out_one_game(2, iterations+1) = REAL(aveprice, ikind4)
2599 !
2600 !          DO iplayers = 1, asyn%nplayers
2601 !              k = iplayers + 2
2602 !              asyn%out_one_game(k, iterations+1) = REAL(x(iplayers), ikind4)
2603 !          END DO
2604 !
2605 !          DO iplayers = 1, asyn%nplayers
2606 !              k = asyn%nplayers + 2 + iplayers
2607 !              asyn%out_one_game(k, iterations+1) = REAL(avex(iplayers), ikind4)
2608 !          END DO
2609 !
2610 !          DO iplayers = 1, asyn%nplayers
2611 !              k = 2 * asyn%nplayers + 2 + iplayers
2612 !              asyn%out_one_game(k, iterations+1) = REAL(j(iplayers), ikind4)
2613 !          END DO
2614 !
2615 !          DO iplayers = 1, asyn%nplayers
2616 !              k = 3 * asyn%nplayers + 2 + iplayers
2617 !              asyn%out_one_game(k, iterations+1) = REAL(avej(iplayers), ikind4)
2618 !          END DO
2619 !
2620 !          DO iplayers = 1, asyn%nplayers
2621 !              k = 4 * asyn%nplayers + 2
2622 !              k = k + iplayers + (iplayers - 1)
2623 !
2624 !              asyn%out_one_game(k, iterations+1) = REAL(ba(iplayers), ikind4)
2625 !              asyn%out_one_game(k+1, iterations+1) = REAL(bb(iplayers), ikind4)
2626 !          END DO
2627 !
2628 !          DO iplayers = 1, asyn%nplayers
2629 !              k = 6 * asyn%nplayers + 2
2630 !              k = k + iplayers + (iplayers - 1)
2631 !              asyn%out_one_game(k, iterations+1) = REAL(aveba(iplayers), ikind4)
2632 !              asyn%out_one_game(k+1, iterations+1) = REAL(avebb(iplayers), ikind4)
2633 !          END DO
2634 !
2635 !          k = 8 * asyn%nplayers + 3
2636 !          asyn%out_one_game(k, iterations+1) = REAL(purchases, ikind4)
2637 !          asyn%out_one_game(k+1, iterations+1) = REAL(isocost, ikind4)
2638 !          asyn%out_one_game(k+2, iterations+1) = REAL(extra, ikind4)
2639 !          asyn%out_one_game(k+3, iterations+1) = REAL(spot, ikind4)
2640 !          asyn%out_one_game(k+4, iterations+1) = REAL(avespot, ikind4)
2641 !
2642 !          k = 8 * asyn%nplayers + 7
2643 !
2644 !          DO iplayers = 1, asyn%nplayers_der
2645 !
2646 !              k = k + (iplayers-1) * 20
2647 !              asyn%out_one_game(k+1, iterations+1) = REAL(x_der_smhydro(iplayers), ikind4)
2648 !              asyn%out_one_game(k+2, iterations+1) = REAL(x_der_wind(iplayers), ikind4)
2649 !              asyn%out_one_game(k+3, iterations+1) = REAL(x_der_pv(iplayers), ikind4)
2650 !              asyn%out_one_game(k+4, iterations+1) = REAL(x_der_chp(iplayers), ikind4)
2651 !              asyn%out_one_game(k+5, iterations+1) = REAL(x_der_fcells(iplayers), ikind4)
2652 !
2653 !              asyn%out_one_game(k+6, iterations+1) = REAL(avex_der_smhydro(iplayers),
2654 ! ikind4)
2655 !
2656 !              asyn%out_one_game(k+7, iterations+1) = REAL(avex_der_wind(iplayers), ikind4)
2657 !              asyn%out_one_game(k+8, iterations+1) = REAL(avex_der_pv(iplayers), ikind4)
2658 !              asyn%out_one_game(k+9, iterations+1) = REAL(avex_der_chp(iplayers), ikind4)
2659 !              asyn%out_one_game(k+10, iterations+1) = REAL(avex_der_fcells(iplayers),
2660 ! ikind4)
2661 !
2662 !              asyn%out_one_game(k+11, iterations+1) = REAL(j_der_smhydro(iplayers), ikind4)
2663 !              asyn%out_one_game(k+12, iterations+1) = REAL(j_der_wind(iplayers), ikind4)
2664 !              asyn%out_one_game(k+13, iterations+1) = REAL(j_der_pv(iplayers), ikind4)
2665 !              asyn%out_one_game(k+14, iterations+1) = REAL(j_der_chp(iplayers), ikind4)
2666 !              asyn%out_one_game(k+15, iterations+1) = REAL(j_der_fcells(iplayers), ikind4)
2667 !
2668 !              asyn%out_one_game(k+16, iterations+1) = REAL(avej_der_smhydro(iplayers),
2669 ! ikind4)
2670 !              asyn%out_one_game(k+17, iterations+1) = REAL(avej_der_wind(iplayers), ikind4)
2671 !              asyn%out_one_game(k+18, iterations+1) = REAL(avej_der_pv(iplayers), ikind4)
2672 !              asyn%out_one_game(k+19, iterations+1) = REAL(avej_der_chp(iplayers), ikind4)

```

```

2673         asyn%out_one_game(k+20, iterations+1) = REAL(avej_der_fcells(iplayers),
2674 ikind4)
2675         ENDDO
2676 !
2677         k = 8 * asyn%nplayers + 7 + 20 * asyn%nplayers_der
2678 !
2679         DO iplayers = 1, asyn%nplayers_dsb
2680
2681             k = k + (iplayers-1) * 20
2682             asyn%out_one_game(k+1, iterations+1) = REAL(x_dsb_smhydro(iplayers), ikind4)
2683             asyn%out_one_game(k+2, iterations+1) = REAL(x_dsb_wind(iplayers), ikind4)
2684             asyn%out_one_game(k+3, iterations+1) = REAL(x_dsb_pv(iplayers), ikind4)
2685             asyn%out_one_game(k+4, iterations+1) = REAL(x_dsb_chp(iplayers), ikind4)
2686             asyn%out_one_game(k+5, iterations+1) = REAL(x_dsb_interrup(iplayers), ikind4)
2687 !
2688             asyn%out_one_game(k+6, iterations+1) = REAL(avex_dsb_smhydro(iplayers),
2689 ikind4)
2690             asyn%out_one_game(k+7, iterations+1) = REAL(avex_dsb_wind(iplayers), ikind4)
2691             asyn%out_one_game(k+8, iterations+1) = REAL(avex_dsb_pv(iplayers), ikind4)
2692             asyn%out_one_game(k+9, iterations+1) = REAL(avex_dsb_chp(iplayers), ikind4)
2693             asyn%out_one_game(k+10, iterations+1) = REAL(avex_dsb_interrup(iplayers),
2694 ikind4)
2695 !
2696             asyn%out_one_game(k+11, iterations+1) = REAL(j_dsb_smhydro(iplayers), ikind4)
2697             asyn%out_one_game(k+12, iterations+1) = REAL(j_dsb_wind(iplayers), ikind4)
2698             asyn%out_one_game(k+13, iterations+1) = REAL(j_dsb_pv(iplayers), ikind4)
2699             asyn%out_one_game(k+14, iterations+1) = REAL(j_dsb_chp(iplayers), ikind4)
2700             asyn%out_one_game(k+15, iterations+1) = REAL(j_dsb_interrup(iplayers), ikind4)
2701 !
2702             asyn%out_one_game(k+16, iterations+1) = REAL(avej_dsb_smhydro(iplayers),
2703 ikind4)
2704             asyn%out_one_game(k+17, iterations+1) = REAL(avej_dsb_wind(iplayers), ikind4)
2705             asyn%out_one_game(k+18, iterations+1) = REAL(avej_dsb_pv(iplayers), ikind4)
2706             asyn%out_one_game(k+19, iterations+1) = REAL(avej_dsb_chp(iplayers), ikind4)
2707             asyn%out_one_game(k+20, iterations+1) = REAL(avej_dsb_interrup(iplayers),
2708 ikind4)
2709         ENDDO
2710 !
2711         END IF
2712 !
2713         ENDDO ! iterations
2714 !
2715         IF (sumsp_sq .LT. BIG .AND. sum_sp .LT. BIG) THEN
2716             sp_stdevp = SQRT( ABS(REAL(asyn%niterations) * sumsp_sq - sum_sp**2.0d+00) ) /
2717 &
2718                 REAL(asyn%niterations)
2719         ELSE
2720             sp_stdevp = BIG
2721         ENDIF
2722 !
2723         sumabs_pr = 0.0d+00
2724 !
2725         DO iterations = 1, asyn%niterations
2726 !
2727             sumabs_pr= sumabs_pr + ABS(varpr(iterations) - aveprice)
2728 !
2729         END DO
2730 !
2731         sp_var = sumabs_pr / REAL(asyn%niterations)
2732 !
2733         asyn%out_many_games(1, igames) = REAL(igames, ikind4)
2734         asyn%out_many_games(2, igames) = REAL(aveprice, ikind4)
2735         asyn%out_many_games(3, igames) = REAL(sp_stdevp, ikind4)
2736         asyn%out_many_games(4, igames) = REAL(sp_var, ikind4)
2737 !
2738         DO iplayers = 1, asyn%nplayers
2739             k = iplayers + 4
2740             asyn%out_many_games(k, igames) = REAL(avex(iplayers), ikind4)
2741         END DO
2742 !
2743         DO iplayers = 1, asyn%nplayers
2744             k = asyn%nplayers + 4 + iplayers
2745             asyn%out_many_games(k, igames) = REAL(avej(iplayers), ikind4)
2746         END DO
2747 !
2748         DO iplayers = 1, asyn%nplayers
2749             k = 2 * asyn%nplayers + 4
2750             k = k + iplayers + (iplayers - 1)
2751             asyn%out_many_games(k, igames) = REAL(aveba(iplayers), ikind4)
2752             asyn%out_many_games(k+1, igames) = REAL(avebb(iplayers), ikind4)
2753         END DO

```

```

2754 !
2755     k = 4 * asyn%nplayers + 5
2756     asyn%out_many_games(k, igames) = sum_purchases
2757     asyn%out_many_games(k+1, igames) = REAL(sum_isocost, ikind4)
2758     asyn%out_many_games(k+2, igames) = REAL(sum_extra, ikind4)
2759     asyn%out_many_games(k+3, igames) = REAL(avespot, ikind4)
2760 !
2761     k = 4 * asyn%nplayers + 8
2762 !
2763     DO iplayers = 1, asyn%nplayers_der
2764         k = k + (iplayers-1) * 10
2765         asyn%out_many_games(k+1, igames) = REAL(avex_der_smhydro(iplayers), ikind4)
2766         asyn%out_many_games(k+2, igames) = REAL(avex_der_wind(iplayers), ikind4)
2767         asyn%out_many_games(k+3, igames) = REAL(avex_der_pv(iplayers), ikind4)
2768         asyn%out_many_games(k+4, igames) = REAL(avex_der_chp(iplayers), ikind4)
2769         asyn%out_many_games(k+5, igames) = REAL(avex_der_fcells(iplayers), ikind4)
2770
2771         asyn%out_many_games(k+6, igames) = REAL(avej_der_smhydro(iplayers) , ikind4)
2772         asyn%out_many_games(k+7, igames) = REAL(avej_der_wind(iplayers), ikind4)
2773         asyn%out_many_games(k+8, igames) = REAL(avej_der_pv(iplayers), ikind4)
2774         asyn%out_many_games(k+9, igames) = REAL(avej_der_chp(iplayers), ikind4)
2775         asyn%out_many_games(k+10, igames) = REAL(avej_der_fcells(iplayers), ikind4)
2776     END DO
2777 !
2778     k = 4 * asyn%nplayers + 8 + 10* asyn%nplayers_der
2779 !
2780     DO iplayers = 1, asyn%nplayers_dsb
2781         k = k + (iplayers-1) * 10
2782         asyn%out_many_games(k+1, igames) = REAL(avex_dsb_smhydro(iplayers), ikind4)
2783         asyn%out_many_games(k+2, igames) = REAL(avex_dsb_wind(iplayers), ikind4)
2784         asyn%out_many_games(k+3, igames) = REAL(avex_dsb_pv(iplayers), ikind4)
2785         asyn%out_many_games(k+4, igames) = REAL(avex_dsb_chp(iplayers), ikind4)
2786         asyn%out_many_games(k+5, igames) = REAL(avex_dsb_interrup(iplayers), ikind4)
2787
2788         asyn%out_many_games(k+6, igames) = REAL(avej_dsb_smhydro(iplayers) , ikind4)
2789         asyn%out_many_games(k+7, igames) = REAL(avej_dsb_wind(iplayers), ikind4)
2790         asyn%out_many_games(k+8, igames) = REAL(avej_dsb_pv(iplayers), ikind4)
2791         asyn%out_many_games(k+9, igames) = REAL(avej_dsb_chp(iplayers), ikind4)
2792         asyn%out_many_games(k+10, igames) = REAL(avej_dsb_interrup(iplayers), ikind4)
2793     END DO
2794
2795     END DO ! games
2796
2797     900 FORMAT(I15, 1000(1X, E14.7))
2798 !
2799     DEALLOCATE (varpr)
2800 !
2801     kerr = ASYN_SUCCESS
2802 !
2803 ! End
2804 Listing.....
2805 !~~~~~
2806     END SUBROUTINE calc_asyn_1sys
2807 !~~~~~
2808     END MODULE asyn_one_sys_m
2809 !=====
2810 !=====

```

```

1  !=====
2  !+ <A one line description of this module>
3  !=====
4  MODULE asyn_two_sys_oneoffer_m
5  !.....
6  !
7  ! Description:
8  !   <Say what this Module is for>
9  !
10 ! Current code owner : C.Skoulidas
11 !
12 ! History:
13 ! Version      Date      Comment
14 ! -----      -
15 ! <18.1>      <2011>   Original code  <C.Skoulidas>
16 !
17 ! Code description:
18 ! Language: Fortran 90.
19 !.....
20 ! Declarations:
21 !
22 ! Modules Used:
23 !
24 !     USE asyn_two_sys_split_m
25 !
26 ! Imported type definitions
27 ! Imported Parameters:
28 ! Imported scalar variables with intent (in):
29 ! Imported scalar variables with intent (out):
30 ! Imported Array variables with intent(in):
31 ! Imported Array variables with intent(out):
32 ! Imported routines:
33 !
34 !     IMPLICIT NONE
35 !
36 ! Global (Public) Declarations:
37 !
38 !
39 !
40 ! Global Type Definitions:
41 !
42 !
43 !     Global parameters:
44 !
45 !     This Parameters Must be justified !
46 ! Global scalars:
47 ! Global arrays:
48 ! Local (Private) Declarations:
49 ! Local Type Definitions:
50 ! Local parameters:
51 ! Local scalars:
52 ! Local arrays:
53 ! Operator definitions:
54 !.....
55 !     CONTAINS
56 !-----
57 ! List Of Internal Procedures:
58 ! Subroutines:
59 !
60 !
61 ! Functions:
62 !
63 !
64 !-----
65 !=====
66 !+ <A one line description of this external subroutine>
67 !=====
68 !     SUBROUTINE calc_asyn_2sys_ofall (gameopt, asyn, kerr)
69 !.....
70 !
71 ! Description:
72 ! ASYN      KEEP ON BIDDING - Bids Above Price Cap NOT ALLOWED
73 !          LINEAR Marginal Cost
74 !          Optional Pricing Method (Uniform OR Pay-As-Bid)
75 !          NO use of GCOMP      Routine
76 !          LAST REVISION: 25 OCTOBER 2002
77 !          ALLOWS Excess Capacity - ALLOWS Loosing at zero sales
78 !          REPEATED runs of the same game
79 !          StDeviation and Variance of SMP added
80 !

```

```

81 ! Method:
82 !   <Say how it does it: include references to external documentation>
83 !   <If this routine is divided in to sections, be brief here,
84 !       and put Method comments at the start of each section>
85 !
86 ! Current code owner : <Name of person responsible of this code>
87 !
88 ! History:
89 ! Version      Date      Comment
90 ! -----      ----      -
91 ! <version>    <date>    Original code. <Your name>
92 !
93 ! Code description:
94 !       Language: Fortran 90.
95 ! .....
96 ! Declarations:
97 !
98 ! Modules Used:
99 ! include statements
100 ! Subroutine Arguments
101 ! Scalar arguments with intent(in):
102 !   TYPE (gameopt_t) :: gameopt
103 !   TYPE (asyn_t)   :: asyn(gameopt%nsystems)
104 !   integer(4)      :: kerr(gameopt%nsystems)
105 ! Array arguments with intent(in):
106 ! Scalar arguments with intent(inout):
107 ! Array arguments with intent(inout):
108 ! Scalar arguments with intent(out):
109 ! Array arguments with intent(out):
110 ! Local parameters:
111 !   real(8), parameter      :: eps_d = 1.0d-8, BIG = 1.0d+30
112 !   real(8), parameter      :: BBMIN = 1.0d-05, BAMIN = 1.0d-05
113 !   real(8), parameter      :: zr = 1.d-07
114 !   integer(4), parameter   :: ikind4 = 4
115 !   integer(4), parameter   :: PRIMARY = 1
116 !   integer(4), parameter   :: SECONDARY = 2
117 ! Local scalars:
118 !   integer(4)              :: igrames, igrplayers, iconst,
119 ! &
120 !                           jplayers, k,
121 ! &
122 !                           iterations, act
123 !   integer(4)              :: ksystem, jsystem, opsystem
124 !   real(8)                 :: y, sumper1, sumper2, rnd, p1, p2, djt,
125 ! &
126 !                           dpi, dp1, dp2, sp_stdevp, sumabs_pr, sp_var
127 !
128 !   real(8)                 :: temp
129 !   real(8)                 :: bamax, bbmax, availpower, &
130 !                           xcurr, revtemp, xtot
131
132 ! Local arrays:
133 !
134 !   integer(4) , dimension (gameopt%nsystems, asyn(1)%nplayers+asyn(2)%nplayers, 2) :: conne
135 ! conne - (:,:,1) = system the player belongs - (:,:,2) = the id numbering of the player
136 !
137 !   integer(4), dimension (gameopt%nsystems) :: nplayers, errcounter, kcol
138 !   integer(4), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers))
139 ! :: &
140 !                                               player_count,
141 coef_flag
142 !   integer(4), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers), NCONST)
143 ! :: &
144
145 action
146 !
147 !   real(8), dimension (gameopt%nsystems)      :: sum_purchases, sum_extra, sum_isocost, extra, kwh,
148 ! &
149 !                                               avekwh, price, aveprice, spot, avspot, sum_sp,
150 ! &
151 !                                               sumsp_sq, purchases, isocost, d, isocost1, uplift
152 !
153 !   real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers), NCONST) :: pi,
154 pd, pc
155 !   real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers))      :: pprim
156 !   real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers),
157 gameopt%nsystems) :: &
158 !                                               x, prof, avex
159 !   real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers)) :: bb, ba,
160 ! &
161 !                                               theta, j, dj, avej, jo, aveba,

```

```

162 avebb
163   real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers), NCONST) :: eps
164 !
165   integer(4), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers)) :: alpha
166 !
167   integer(4), allocatable, dimension (:) :: expl
168 !
169   real(8), allocatable, dimension (:,:) :: cur_ba, cur_bb, xmin, xmax
170   real(8), allocatable, dimension (:,:,) :: lp
171 !
172   real(8), allocatable, dimension (:) :: lpoint, qpoint
173   real(8), allocatable, dimension (:,:) :: qx
174   real(8), allocatable, dimension (:) :: xx, uplar
175   real(8), allocatable, dimension (:,:) :: varpr
176
177 !.....
178 ! Listing
179 !
180   DO ksystem = 1, gameopt%nsystems
181 !
182       IF (gameopt%ngames .EQ. 1) asyn(ksystem)%out_one_game = 0.0d+00
183       asyn(ksystem)%out_many_games = 0.0d+00
184 !
185       END DO
186 ! initialize error index
187   kerr = 0
188   errcounter = 0
189 !
190   asyn(2)%niterations = asyn(1)%niterations
191 !
192   ALLOCATE ( varpr(gameopt%nsystems, asyn(1)%niterations) )
193 !
194 ! Start the game iterations
195 !
196   CALL random_seed()
197 !
198   DO igames = 1, gameopt%ngames
199 !
200 ! asyn%p_flag .EQ. 0 ! the following condition is always FALSE
201
202       DO ksystem = 1, gameopt%nsystems
203 !
204           IF (asyn(ksystem)%p_flag .EQ. 1) THEN
205 !
206               DO iplayers = 1, asyn(ksystem)%nplayers
207 !
208                   DO iconst = 1, NCONST
209 !
210                       CALL random_number (y)
211                       asyn(ksystem)%player(iplayers)%period(iconst) =
212 INT(asyn(ksystem)%period_min + &
213                                     (asyn(ksystem)%period_max -asyn(ksystem)%period_min )
214 * y)
215 !
216                   ENDDO
217 !
218               END DO
219 !
220           END IF
221 !
222       END DO
223 !
224 ! Assign/Reassign initial values
225 !
226   DO ksystem = 1, gameopt%nsystems
227 !
228       DO iplayers = 1, asyn(ksystem)%nplayers
229 !
230           bb(ksystem, iplayers) = 2.0d+00 * asyn(ksystem)%player(iplayers)%b
231           ba(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%a
232           alpha(ksystem, iplayers) = PRIMARY
233 !
234       END DO
235 !
236   END DO
237 !
238   aveba(:, :) = ba(:, :)
239   avebb(:, :) = bb(:, :)
240 !
241 ! Initial bid parameters values and sums related to them
242 !

```

```

243 !           Assign/Reassign initial values (=zero) to Sums related with the
244 ! Bid Parameters - 1st Round
245 !
246 !           sum_sp = 0.0d+00
247 !           sumsp_sq = 0.0d+00
248 !
249 !           sum_purchases = 0.0d+00
250 !           sum_extra = 0.0d+00
251 !           sum_isocost = 0.0d+00
252 !
253 !           x(:, :, :) = 0.0d+00; prof(:, :, :) = 0.0d+00; avex(:, :, :) = 0.0d+00
254 !
255 ! Find the active players
256 !
257 !           nplayers(:) = 0
258 !
259 !           DO ksystem = 1, gameopt%nsystems
260 !
261 !               IF (ksystem .EQ. 1) jsystem = 2
262 !               IF (ksystem .EQ. 2) jsystem = 1
263 !
264 !               DO iplayers = 1, asyn(ksystem)%nplayers
265 !
266 !                   IF (alpha(ksystem, iplayers) .EQ. SECONDARY) THEN ! pazei sto secondary
267 !
268 !                       nplayers(jsystem) = nplayers(jsystem) + 1
269 !                       conne(jsystem, nplayers(jsystem), 1) = iplayers
270 !                       conne(jsystem, nplayers(jsystem), 2) = ksystem
271 !
272 !                   ELSE IF (alpha(ksystem, iplayers) .EQ. PRIMARY) THEN ! pazei sto primary
273 !
274 !                       nplayers(ksystem) = nplayers(ksystem) + 1
275 !                       conne(ksystem, nplayers(ksystem), 1) = iplayers
276 !                       conne(ksystem, nplayers(ksystem), 2) = ksystem
277 !
278 !                   END IF
279 !
280 !               END DO
281 !
282 !           END DO
283 !
284 !           ALLOCATE (cur_ba(gameopt%nsystems, MAX(nplayers(1), nplayers(2))),
285 ! &
286 !                   cur_bb(gameopt%nsystems, MAX(nplayers(1), nplayers(2))),
287 ! &
288 !                   xmins(gameopt%nsystems, MAX(nplayers(1), nplayers(2))),
289 ! &
290 !                   xmaxs(gameopt%nsystems, MAX(nplayers(1), nplayers(2))))
291 !
292 !           Assing/Reassign values to bidding parameters and Calculate
293 !           the sums related with them - 1st Round
294 !
295 !           DO ksystem = 1, gameopt%nsystems
296 !
297 !               DO iplayers = 1, nplayers(ksystem)
298 !
299 !                   jplayers = conne(ksystem, iplayers, 1)
300 !                   jsystem = conne(ksystem, iplayers, 2)
301 !
302 !                   cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
303 !                   cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
304 !
305 !                   xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
306 !                   xmaxs(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmax
307 !
308 !               END DO
309 !
310 !           END DO
311 !
312 !           Initial Demand Assignment/Reassignment
313 !
314 !           DO ksystem = 1, gameopt%nsystems
315 !
316 !               ALLOCATE (xx(nplayers(ksystem)), uplar(nplayers(ksystem)) )
317 !
318 !               d(ksystem) = asyn(ksystem)%inid
319 !
320 !           END DO
321 ! Check for available power with respect to the demand
322 !           availpower = 0.0d+00
323 !

```



```

324         DO iplayers = 1, nplayers(ksystem)
325     !
326         availpower = availpower + xmaxs(ksystem, iplayers)
327
328     END DO
329     !
330     IF (availpower .LT. d(ksystem)) THEN
331     !
332         kerr(ksystem) = ERR_DEMAND
333     !
334         RETURN
335     !
336     END IF
337     !
338     CALL spot_price(nplayers(ksystem), cur_ba(ksystem, 1:nplayers(ksystem)),
339 &
340         cur_bb(ksystem, 1:nplayers(ksystem)), xmin(ksystem,
341 1:nplayers(ksystem)), &
342         xmaxs(ksystem, 1:nplayers(ksystem)), d(ksystem), kerr(ksystem),
343 &
344         spot(ksystem), xx)
345
346     IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
347     !
348     !Calculate the isocost (isocost1) from the first cleaning
349     !
350     isocost1(ksystem) = 0.0d+00
351     !
352     DO iplayers = 1, nplayers(ksystem)
353     !
354         IF (asyn(ksystem)%pr_method .EQ. 0) THEN
355
356             revtemp = spot(ksystem) * xx(iplayers)
357         !
358         ELSE
359         !
360             revtemp = 0.0d+00
361             IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
362         !
363                 revtemp = cur_ba(ksystem, iplayers) * xx(iplayers) +
364 &
365                     (cur_bb(ksystem, iplayers) / 2.0d+00) * xx(iplayers) *
366 xx(iplayers) + &
367                     (cur_bb(ksystem, iplayers) / 2.0d+00) *
368 &
369                     xmin(ksystem, iplayers) * xmin(ksystem, iplayers)
370             END IF
371         !
372         END IF
373     !
374     isocost1(ksystem) = isocost1(ksystem) + revtemp
375     !
376     END DO
377     !
378     CALL capacity_constraint (gameopt, asyn, nplayers(ksystem),
379 &
380         cur_ba(ksystem, 1:nplayers(ksystem)), cur_bb(ksystem, 1:nplayers(ksystem)),
381 &
382         xmin(ksystem, 1:nplayers(ksystem)), xmaxs(ksystem, 1:nplayers(ksystem)),
383 d(ksystem), &
384         kerr(ksystem), xx, ksystem, conne(ksystem, 1:nplayers(ksystem), :))
385
386     IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
387     !
388     isocost(ksystem) = 0.0d+00
389     !
390     DO iplayers = 1, nplayers(ksystem)
391     !
392         IF (asyn(ksystem)%pr_method .EQ. 0) THEN
393
394             revtemp = spot(ksystem) * xx(iplayers)
395         !
396         ELSE
397         !
398             revtemp = 0.0d+00
399             IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
400         !
401                 revtemp = cur_ba(ksystem, iplayers) * xx(iplayers) +
402 &
403                     (cur_bb(ksystem, iplayers) / 2.0d+00) * xx(iplayers) *
404 xx(iplayers) + &

```

```

405                                     (cur_bb(kssystem, iplayers) / 2.0d+00) *
406 &
407                                     xmins(kssystem, iplayers) * xmins(kssystem, iplayers)
408     END IF
409 !
410     END IF
411 !
412     isocost(kssystem) = isocost(kssystem) + revtemp
413 !
414     END DO
415     uplift(kssystem) = isocost(kssystem) - isocost1(kssystem)
416
417 !
418     avespot(kssystem) = spot(kssystem)
419 !
420     varpr(kssystem, 1) = spot(kssystem)
421 !
422     sum_sp(kssystem) = spot(kssystem)
423     sumsp_sq(kssystem) = spot(kssystem) * spot(kssystem)
424 !
425     purchases(kssystem) = 0.0d+00
426 !
427 ! Calculate the uplift allocation ratio
428 !
429     xtot = 0.0d+00
430     DO iplayers = 1, nplayers(kssystem)
431 !
432         jsystem = conne(kssystem, iplayers, 2)
433 !
434         IF (kssystem .EQ. jsystem) xtot = xtot + xx(iplayers)
435
436     END DO
437 !
438     DO iplayers = 1, nplayers(kssystem)
439 !
440         jsystem = conne(kssystem, iplayers, 2)
441 !
442         IF (kssystem .EQ. jsystem .and. xtot .GT. eps_d) THEN
443             uplar(iplayers) = xx(iplayers) / xtot
444         ELSE
445             uplar(iplayers) = 0.0d+00
446         END IF
447
448     END DO
449 !
450     DO iplayers = 1, nplayers(kssystem)
451 !
452         purchases(kssystem) = purchases(kssystem) + xx(iplayers)
453 !
454         jplayers = conne(kssystem, iplayers, 1)
455         jsystem = conne(kssystem, iplayers, 2)
456 !
457         x(jsystem, jplayers, kssystem) = xx(iplayers)
458         avex(jsystem, jplayers, kssystem) = x(jsystem, jplayers, kssystem)
459 !
460         IF (asyn(kssystem)%pr_method .EQ. 0) THEN
461 !
462             prof(jsystem, jplayers, kssystem) = spot(kssystem) * xx(iplayers)
463 !
464         ELSE
465 !
466             IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
467 !
468                 prof(jsystem, jplayers, kssystem) =
469 &
470                     cur_ba(kssystem, iplayers) * xx(iplayers) +
471 &
472                     (cur_bb(kssystem, iplayers) / 2.0d+00) * xx(iplayers) *
473 xx(iplayers) + &
474                     (cur_bb(kssystem, iplayers) / 2.0d+00) *
475 &
476                     xmins(kssystem, iplayers) * xmins(kssystem, iplayers)
477 !
478             ELSE
479                 prof(jsystem, jplayers, kssystem) = 0.0
480             END IF
481 !
482         END IF
483 !
484 !
485 ! substarct the marginal cost

```

```

486
487     prof(jsystem, jplayers, ksystem) = prof(jsystem, jplayers, ksystem) -
488 &
489         (asyn(jsystem)%player(jplayers)%fc +
490 &
491     asyn(jsystem)%player(jplayers)%a * x(jsystem, jplayers, ksystem) +
492 &
493     asyn(jsystem)%player(jplayers)%b * x(jsystem, jplayers, ksystem)**2.0d+00)
494 !
495     IF (jsystem .EQ. ksystem) THEN
496         prof(jsystem, jplayers, ksystem) = prof(jsystem, jplayers, ksystem) -
497 &
498             uplift(ksystem) * uplar(iplayers)
499     END IF
500 !
501     jo(jsystem, jplayers) = prof(jsystem, jplayers, ksystem)
502 !
503     avej(jsystem, jplayers) = jo(jsystem, jplayers)
504 !
505 END DO ! player
506 !
507     extra(ksystem) = purchases(ksystem) - d(ksystem)
508     kwh(ksystem) = isocost(ksystem) / purchases(ksystem)
509     avekwh(ksystem) = kwh(ksystem)
510 !
511     IF (asyn(ksystem)%pr_method .EQ. 0) THEN
512 !
513         price(ksystem) = spot(ksystem)
514         aveprice(ksystem) = avespot(ksystem)
515 !
516     ELSE
517 !
518         price(ksystem) = kwh(ksystem)
519         aveprice(ksystem) = avekwh(ksystem)
520 !
521     END IF
522 !
523     sum_purchases(ksystem) = purchases(ksystem)
524     sum_extra(ksystem) = extra(ksystem)
525     sum_isocost(ksystem) = isocost(ksystem)
526 !
527     DEALLOCATE (xx, uplar)
528 !
529 END DO ! ksystem
530 !
531     Detailed output of initial state
532 !
533 IF (gameopt%ngames .EQ. 1) THEN
534 !
535     DO ksystem = 1, gameopt%nsystems
536 !
537         asyn(ksystem)%out_one_game(1, 1) = REAL(price(ksystem), ikind4)
538         asyn(ksystem)%out_one_game(2, 1) = REAL(aveprice(ksystem), ikind4)
539         asyn(ksystem)%out_one_game(3, 1) = REAL(spot(ksystem), ikind4)
540         asyn(ksystem)%out_one_game(4, 1) = REAL(avespot(ksystem), ikind4)
541 !
542         DO iplayers = 1, asyn(ksystem)%nplayers
543 !
544             kcol = 4 + iplayers
545             asyn(ksystem)%out_one_game(kcol, 1) = x(ksystem, iplayers, 1)
546 !
547         END DO
548 !
549         DO iplayers = 1, asyn(ksystem)%nplayers
550 !
551             kcol = 4 + asyn(ksystem)%nplayers + iplayers
552             asyn(ksystem)%out_one_game(kcol, 1) = x(ksystem, iplayers, 2)
553 !
554         END DO
555 !
556     END DO
557 !
558     ksystem = 1
559     DO iplayers = 1, nplayers(ksystem)
560 !
561         jplayers = conne(ksystem, iplayers, 1)
562         jsystem = conne(ksystem, iplayers, 2)
563
564         kcol = 4 + 2*asyn(jsystem)%nplayers + jplayers
565         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers)
566 !

```

```

567         END DO
568     !
569     ksystem = 2
570     DO iplayers = 1, nplayers(ksystem)
571     !
572         jplayers = conne(ksystem, iplayers, 1)
573         jsystem = conne(ksystem, iplayers, 2)
574
575         kcol = 4 + 3*asyn(jsystem)%nplayers + jplayers
576         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers)
577     !
578     END DO
579     !
580     ksystem = 1
581     DO iplayers = 1, nplayers(ksystem)
582     !
583         jplayers = conne(ksystem, iplayers, 1)
584         jsystem = conne(ksystem, iplayers, 2)
585
586         kcol = 4 + 2*asyn(jsystem)%nplayers + jplayers
587         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
588     &
589         cur_bb(ksystem, iplayers) * xmins(ksystem, iplayers)
590     !
591     END DO
592     !
593     ksystem = 2
594     DO iplayers = 1, nplayers(ksystem)
595     !
596         jplayers = conne(ksystem, iplayers, 1)
597         jsystem = conne(ksystem, iplayers, 2)
598
599         kcol = 4 + 3*asyn(jsystem)%nplayers + jplayers
600         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
601     &
602         cur_bb(ksystem, iplayers) * xmins(ksystem, iplayers)
603     !
604     END DO
605     !
606     ksystem = 1
607     DO iplayers = 1, nplayers(ksystem)
608     !
609         jplayers = conne(ksystem, iplayers, 1)
610         jsystem = conne(ksystem, iplayers, 2)
611
612         kcol = 4 + 4*asyn(jsystem)%nplayers + jplayers
613         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
614     &
615         cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers)
616     !
617     END DO
618     !
619     ksystem = 2
620     DO iplayers = 1, nplayers(ksystem)
621     !
622         jplayers = conne(ksystem, iplayers, 1)
623         jsystem = conne(ksystem, iplayers, 2)
624
625         kcol = 4 + 5*asyn(jsystem)%nplayers + jplayers
626         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
627     &
628         cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers)
629     !
630     END DO
631     ksystem = 1
632     DO iplayers = 1, nplayers(ksystem)
633     !
634         jplayers = conne(ksystem, iplayers, 1)
635         jsystem = conne(ksystem, iplayers, 2)
636
637         kcol = 4 + 6*asyn(jsystem)%nplayers + jplayers
638         asyn(jsystem)%out_one_game(kcol, 1) = xmins(ksystem, iplayers)
639     !
640     END DO
641     !
642     ksystem = 2
643     DO iplayers = 1, nplayers(ksystem)
644     !
645         jplayers = conne(ksystem, iplayers, 1)
646         jsystem = conne(ksystem, iplayers, 2)
647

```

```

648         kcol = 4 + 7*asyn(jsystem)%nplayers + jplayers
649         asyn(jsystem)%out_one_game(kcol, 1) = xmins(ksystem, iplayers)
650 !
651     END DO
652 !
653     ksystem = 1
654     DO iplayers = 1, nplayers(ksystem)
655 !
656         jplayers = conne(ksystem, iplayers, 1)
657         jsystem = conne(ksystem, iplayers, 2)
658
659         kcol = 4 + 8*asyn(jsystem)%nplayers + jplayers
660         asyn(jsystem)%out_one_game(kcol, 1) = xmaxs(ksystem, iplayers)
661 !
662     END DO
663 !
664     ksystem = 2
665     DO iplayers = 1, nplayers(ksystem)
666 !
667         jplayers = conne(ksystem, iplayers, 1)
668         jsystem = conne(ksystem, iplayers, 2)
669
670         kcol = 4 + 9*asyn(jsystem)%nplayers + jplayers
671         asyn(jsystem)%out_one_game(kcol, 1) = xmaxs(ksystem, iplayers)
672 !
673     END DO
674 !
675 END IF
676 !
677     DEALLOCATE (cur_ba, cur_bb, xmins, xmaxs)
678 !
679 -----Iterations-----
680 ----
681 !
682 !     Assign/Reassign initial values to Behaviour Probabilities
683 !
684     DO ksystem = 1, gameopt%nsystems
685 !
686         DO iplayers = 1, asyn(ksystem)%nplayers
687 !
688             pi(ksystem, iplayers, :) = asyn(ksystem)%player(iplayers)%inipi(:)
689             pd(ksystem, iplayers, :) = asyn(ksystem)%player(iplayers)%inipd(:)
690             pc(ksystem, iplayers, :) = asyn(ksystem)%player(iplayers)%inipc(:)
691             pprim(ksystem, iplayers) = 1.0 - asyn(ksystem)%player(iplayers)%alpha_tolerance
692 !
693             IF (asyn(ksystem)%player(iplayers)%xmin .GT. gameopt%interconne*1000.0d+00)
694 pprim(ksystem, iplayers) = 1.0 ! an xmin> interconne den mporei na metaferai sto secondary
695 !
696         END DO
697 !
698     END DO
699 !
700     player_count(:, :) = 0
701 !
702     DO iterations = 1, asyn(1)%niterations-1
703 !
704         20 CONTINUE
705 !
706 !     Random desicion & optimization
707 !
708 !
709 !     Start iterative procedure for each player
710 !
711         DO ksystem = 1, gameopt%nsystems
712 !
713             DO iplayers = 1, asyn(ksystem)%nplayers
714 !
715                 sumper1 = asyn(ksystem)%player(iplayers)%period(1)
716                 sumper2 = asyn(ksystem)%player(iplayers)%period(1) +
717 &
718                     asyn(ksystem)%player(iplayers)%period(2)
719 !
720                 IF (player_count(ksystem, iplayers) .GE. sumper2) player_count(ksystem,
721 iplayers) = 0
722 !
723                 player_count(ksystem, iplayers) = player_count(ksystem, iplayers) + 1
724 !
725 !     Decision block that assigns value only to one of the BA, BB parameter's step
726 !     (and keep the others zero) depending on the current parameter adaption period
727 !
728                 IF (player_count(ksystem, iplayers) .LE. sumper1) THEN

```

```

729
730         coef_flag(ksystem, iplayers) = 1
731 !
732         IF (iiterations .GT. INT(asyn(ksystem)%estep_decr *
733 (asyn(ksystem)%niterations))) THEN !IF THEN Block that
734 !
735             eps(ksystem, iplayers, 1) = asyn(ksystem)%player(iplayers)%e_init(1) *
736 &
737                 (asyn(ksystem)%estep_decr *
738 &
739                 REAL(asyn(ksystem)%niterations)) / REAL(iiterations
740 + 1) ! gradually decreases
741 !
742             ELSE !the step size of BA
743 !
744                 eps(ksystem, iplayers, 1) = asyn(ksystem)%player(iplayers)%e_init(1)
745 !
746             END IF
747 !
748             eps(ksystem, iplayers, 2) = 0.0d+00
749 !
750         ELSE IF (player_count(ksystem, iplayers) .LE. sumper2) THEN
751 !
752             coef_flag(ksystem, iplayers) = 2
753             eps(ksystem, iplayers, 1) = 0.0d+00
754 !
755             IF (iiterations .GT. INT(asyn(ksystem)%estep_decr *
756 (asyn(ksystem)%niterations))) THEN !IF THEN Block that
757 !
758                 eps(ksystem, iplayers, 2) = asyn(ksystem)%player(iplayers)%e_init(2) *
759 &
760                     (asyn(ksystem)%estep_decr *
761                     REAL(asyn(ksystem)%niterations)) / REAL(iiterations + 1) !gradually
762 decreases
763 !
764             ELSE !the step size of BB
765 !
766                 eps(ksystem, iplayers, 2) = asyn(ksystem)%player(iplayers)%e_init(2)
767 !
768             END IF
769 !
770         END IF
771 !
772         Decision block that assigns value to  $\theta$ 
773 !
774         IF (iiterations .GT. INT(asyn(ksystem)%thstep_decr *
775 (asyn(ksystem)%niterations))) THEN !IF THEN Block that
776 !
777             theta(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%theta_ini *
778 &
779                 (asyn(ksystem)%thstep_decr *
780                 REAL(asyn(ksystem)%niterations)) / REAL(iiterations + 1) !gradually
781 decreases
782 !
783             ELSE !  $\theta$  step size
784 !
785                 theta(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%theta_ini
786 !
787             END IF
788 !
789         Choose a random value from the probabilistic action profile
790         for the in turn parameter (BA or BB)and
791         adjust the in turn parameter (BA or BB) in compliance with
792         the randomly selected value
793 !
794         CALL random_number (y)
795 !
796         rnd = y
797 !
798         p1 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers))
799         p2 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) +
800 &
801             pd(ksystem, iplayers, coef_flag(ksystem, iplayers))
802 !
803         IF (rnd .LE. p1) THEN ! increase
804 !
805             action(ksystem, iplayers, coef_flag(ksystem, iplayers)) = 1
806             ba(ksystem, iplayers) = ba(ksystem, iplayers) +
807 &
808                 eps(ksystem, iplayers, 1) * ba(ksystem, iplayers)
809             bb(ksystem, iplayers) = bb(ksystem, iplayers) +

```

```

810 &
811         eps(ksystem, iplayers, 2) * bb(ksystem, iplayers)
812 !
813     ELSE IF (rnd .LE. p2) THEN ! decrease
814
815 !
816         action(ksystem, iplayers, coef_flag(ksystem, iplayers)) = 2
817         ba(ksystem, iplayers) = ba(ksystem, iplayers) -
818 &
819         eps(ksystem, iplayers, 1) * ba(ksystem, iplayers)
820 !
821         bb(ksystem, iplayers) = bb(ksystem, iplayers) -
822 &
823         eps(ksystem, iplayers, 2) * bb(ksystem, iplayers)
824 !
825     ELSE ! stabilize
826
827 !
828         action(ksystem, iplayers, coef_flag(ksystem, iplayers)) = 3
829         ba(ksystem, iplayers) = ba(ksystem, iplayers)
830         bb(ksystem, iplayers) = bb(ksystem, iplayers)
831 !
832     END IF
833 !
834     IF (bb(ksystem, iplayers) .LT. BBMIN) bb(ksystem, iplayers) = BBMIN
835     IF (ba(ksystem, iplayers) .LT. BAMIN) ba(ksystem, iplayers) = BAMIN
836 !
837 ! Choose a random value from the probabilistic action profile for the transfer parameter
838 alpha
839 !
840     IF (gameopt%probability_on) THEN
841
842 !
843         CALL random_number (y)
844 !
845         rnd = y
846 !
847         IF (rnd .LE. pprim(ksystem, iplayers)) THEN ! na paei sto primary
848 !
849             alpha(ksystem, iplayers) = PRIMARY
850 !
851         ELSE ! na paei sto secondary
852 !
853             alpha(ksystem, iplayers) = SECONDARY
854 !
855         END IF
856 !
857     END IF
858 !
859     ENDDO ! players
860 !
861     ENDDO ! systems
862 !
863 ! Find the active players
864 !
865     nplayers(:) = 0
866     x(:, :, :) = 0.0d+00; prof(:, :, :) = 0.0d+00
867 !
868     DO ksystem = 1, gameopt%nssystems
869 !
870         IF (ksystem .EQ. 1) jsystem = 2
871         IF (ksystem .EQ. 2) jsystem = 1
872 !
873         DO iplayers = 1, asyn(ksystem)%nplayers
874 !
875             IF (alpha(ksystem, iplayers) .EQ. SECONDARY) THEN ! pazei sto secondary
876 !
877                 nplayers(jsystem) = nplayers(jsystem) + 1
878                 conne(jsystem, nplayers(jsystem), 1) = iplayers
879                 conne(jsystem, nplayers(jsystem), 2) = ksystem
880 !
881             ELSE IF (alpha(ksystem, iplayers) .EQ. PRIMARY) THEN ! pazei mono sto primary
882 !
883                 nplayers(ksystem) = nplayers(ksystem) + 1
884                 conne(ksystem, nplayers(ksystem), 1) = iplayers
885                 conne(ksystem, nplayers(ksystem), 2) = ksystem
886 !
887             END IF
888 !
889         END DO
890 !

```

```

891         END DO
892 !
893 ! Check for available power with respect to the demand
894         DO ksystem = 1, gameopt%nsystems
895 !
896             CALL random_number (y)
897 !
898             d(ksystem) = asyn(ksystem)%inid * (1.0d+00 - asyn(ksystem)%dd + 2.0d+00 *
899 asyn(ksystem)%dd * y)
900
901 !
902             availpower = 0.0d+00
903             DO iplayers = 1, nplayers(ksystem)
904 !
905                 jplayers = conne(ksystem, iplayers, 1)
906                 jsystem = conne(ksystem, iplayers, 2)
907 !
908                 availpower = availpower + asyn(jsystem)%player(jplayers)%xmax
909 !
910             END DO
911             IF (availpower .LT. d(ksystem)) THEN
912 !
913                 errcounter(ksystem) = errcounter(ksystem) + 1
914
915                 GO TO 20
916 !
917             ENDIF
918 !
919         END DO
920 !
921         ALLOCATE ( cur_ba(gameopt%nsystems, MAX(nplayers(1),nplayers(2))),
922 &
923                 cur_bb(gameopt%nsystems, MAX(nplayers(1),nplayers(2))),
924 &
925                 xmins(gameopt%nsystems, MAX(nplayers(1),nplayers(2))),
926 &
927                 xmaxs(gameopt%nsystems, MAX(nplayers(1),nplayers(2))) )
928 !
929         DO ksystem = 1, gameopt%nsystems
930 !
931             DO iplayers = 1, nplayers(ksystem)
932 !
933                 jplayers = conne(ksystem, iplayers, 1)
934                 jsystem = conne(ksystem, iplayers, 2)
935 !
936                 cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
937                 cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
938 !
939                 xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
940                 xmaxs(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmax
941 !
942                 bamax = asyn(ksystem)%pricecap -
943 &
944                     (cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers))
945 !
946                 IF (coef_flag(jsystem, jplayers) .EQ. 1 .AND. cur_ba(ksystem, iplayers) .GT.
947 bamax) THEN
948
949                     IF (bamax .LT. BAMIN) THEN
950
951                         bbmax = (asyn(ksystem)%pricecap - BAMIN) / xmaxs(ksystem, iplayers)
952 !
953                         ba(jsystem, jplayers) = BAMIN
954                         bb(jsystem, jplayers) = bbmax
955 !
956                         cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
957                         cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
958 !
959                     ELSE
960
961                         ba(jsystem, jplayers) = bamax
962                         cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
963
964                     END IF
965 !
966                 END IF
967 !
968                 bbmax = (asyn(ksystem)%pricecap - cur_ba(ksystem, iplayers)) / xmaxs(ksystem,
969 iplayers)
970 !
971                 IF (coef_flag(jsystem, jplayers) .EQ. 2 .AND. cur_bb(ksystem, iplayers) .GT.

```



```

972 bbmax) THEN
973 !
974         IF (bbmax .LT. BAMIN) THEN
975
976             bamax = asyn(kssystem)%pricecap - (BBMIN * xmaxs(kssystem, iplayers))
977
978             bb(jsystem, jplayers) = BBMIN
979             ba(jsystem, jplayers) = bamax
980 !
981             cur_bb(kssystem, iplayers) = bb(jsystem, jplayers)
982             cur_ba(kssystem, iplayers) = ba(jsystem, jplayers)
983 !
984         ELSE
985 !
986             bb(jsystem, jplayers) = bbmax
987             cur_bb(kssystem, iplayers) = bb(jsystem, jplayers)
988 !
989         END IF
990 !
991     END IF
992 !
993 END DO
994 END DO
995 !
996 ! Calculation of bidding Parameters' Moving Averages
997 !
998 DO ksystem = 1, gameopt%nsystems
999     DO iplayers = 1, asyn(kssystem)%nplayers
1000 !
1001         aveba(kssystem, iplayers) = aveba(kssystem, iplayers) + (ba(kssystem, iplayers) -
1002 &
1003             aveba(kssystem, iplayers)) / REAL(iiterations + 1)
1004 !
1005         avebb(kssystem, iplayers) = avebb(kssystem, iplayers) + (bb(kssystem, iplayers) -
1006 &
1007             avebb(kssystem, iplayers)) / REAL(iiterations+1)
1008 !
1009     END DO
1010 END DO
1011 !
1012 ! Initial Demand Assignment/Reassignment
1013 !
1014 ! DO ksystem = 1, gameopt%nsystems
1015 !
1016 !     ALLOCATE (xx(nplayers(kssystem)), uplar(nplayers(kssystem)) )
1017 !
1018 !     CALL spot_price(nplayers(kssystem), cur_ba(kssystem, 1:nplayers(kssystem)),
1019 cur_bb(kssystem, 1:nplayers(kssystem)),
1020 &
1021         xmin(kssystem, 1:nplayers(kssystem)), xmaxs(kssystem,
1022 1:nplayers(kssystem)), d(kssystem), kerr(kssystem), spot(kssystem), xx)
1023 !
1024     IF (kerr(kssystem) .NE. ASYN_SUCCESS) RETURN
1025 !
1026     isocost1(kssystem) = 0.0d+00
1027 !
1028     DO iplayers = 1, nplayers(kssystem)
1029 !
1030         IF (asyn(kssystem)%pr_method .EQ. 0) THEN
1031
1032             revtemp = spot(kssystem) * xx(iplayers)
1033 !
1034         ELSE
1035 !
1036             revtemp = 0.0d+00
1037             IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
1038 !
1039                 revtemp = cur_ba(kssystem, iplayers) * xx(iplayers) +
1040 &
1041                     (cur_bb(kssystem, iplayers) / 2.0d+00) * xx(iplayers) *
1042 xx(iplayers) + &
1043                     (cur_bb(kssystem, iplayers) / 2.0d+00) *
1044 &
1045                         xmin(kssystem, iplayers) * xmin(kssystem, iplayers)
1046             END IF
1047 !
1048         END IF
1049 !
1050     isocost1(kssystem) = isocost1(kssystem) + revtemp
1051 !
1052     END DO

```

```

1053 !
1054         CALL capacity_constraint (gameopt, asyn, nplayers(ksystem), cur_ba(ksystem,
1055 1:nplayers(ksystem)), &
1056         cur_bb(ksystem, 1:nplayers(ksystem)), xmins(ksystem,
1057 1:nplayers(ksystem)),
1058         &
1059         &
1060         ksystem, conne(ksystem, 1:nplayers(ksystem), :))
1061 !
1062         IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
1063 !
1064
1065         isocost(ksystem) = 0.0d+00
1066 !
1067         DO iplayers = 1, nplayers(ksystem)
1068 !
1069             IF (asyn(ksystem)%pr_method .EQ. 0) THEN
1070
1071                 revtemp = spot(ksystem) * xx(iplayers)
1072 !
1073             ELSE
1074 !
1075                 revtemp = 0.0d+00
1076                 IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
1077 !
1078                     revtemp = cur_ba(ksystem, iplayers) * xx(iplayers) +
1079 &
1080                     (cur_bb(ksystem, iplayers) / 2.0d+00) * xx(iplayers) *
1081 xx(iplayers) + &
1082                     (cur_bb(ksystem, iplayers) / 2.0d+00) *
1083 &
1084                     xmins(ksystem, iplayers) * xmins(ksystem, iplayers)
1085                 END IF
1086 !
1087             END IF
1088 !
1089             isocost(ksystem) = isocost(ksystem) + revtemp
1090 !
1091         END DO
1092         uplift(ksystem) = isocost(ksystem) - isocost1(ksystem)
1093 !
1094 ! Calculate the uplift allocation ratio
1095 !
1096         xtot = 0.0d+00
1097         DO iplayers = 1, nplayers(ksystem)
1098 !
1099             jsystem = conne(ksystem, iplayers, 2)
1100 !
1101             IF (ksystem .EQ. jsystem) xtot = xtot + xx(iplayers)
1102
1103         END DO
1104 !
1105         DO iplayers = 1, nplayers(ksystem)
1106 !
1107             jsystem = conne(ksystem, iplayers, 2)
1108 !
1109             IF (ksystem .EQ. jsystem .and. xtot .GT. eps_d) THEN
1110                 uplar(iplayers) = xx(iplayers) / xtot
1111             ELSE
1112                 uplar(iplayers) = 0.0d+00
1113             END IF
1114
1115         END DO
1116 !
1117 !
1118 ! Calculation of the Spot Price's Moving Average [AveSPOT]
1119 !
1120         avespot(ksystem) = avespot(ksystem) +
1121 &
1122         (spot(ksystem) - avespot(ksystem)) / REAL(iterations + 1)
1123 !
1124         varpr(ksystem, iterations + 1) = spot(ksystem)
1125 !
1126         sum_sp(ksystem) = sum_sp(ksystem) + spot(ksystem)
1127         sumsp_sq(ksystem) = sumsp_sq(ksystem) + spot(ksystem)**2.0D+00
1128 !
1129         purchases(ksystem) = 0.0d+00
1130 !
1131         DO iplayers = 1, nplayers(ksystem)
1132 !
1133             purchases(ksystem) = purchases(ksystem) + xx(iplayers)

```

```

1134 !
1135       jplayers = conne(ksystem, iplayers, 1)
1136       jsystem = conne(ksystem, iplayers, 2)
1137 !
1138       x(jsystem, jplayers, ksystem) = xx(iplayers)
1139       avex(jsystem, jplayers, ksystem) = avex(jsystem, jplayers, ksystem) +
1140 &
1141           (x(jsystem, jplayers, ksystem) - avex(jsystem, jplayers, ksystem))
1142 /   &
1143           REAL(iiterations + 1)
1144 !
1145       IF (asyn(ksystem)%pr_method .EQ. 0) THEN
1146 !
1147           prof(jsystem, jplayers, ksystem) = spot(ksystem) * xx(iplayers)
1148 !
1149       ELSE
1150 !
1151           IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
1152 !
1153               prof(jsystem, jplayers, ksystem) =
1154 &
1155                   cur_ba(ksystem, iplayers) * xx(iplayers) +
1156 &
1157                   (cur_bb(ksystem, iplayers) / 2.0d+00) *
1158 &
1159                   xx(iplayers) * xx(iplayers) +
1160 &
1161                   (cur_bb(ksystem, iplayers) / 2.0d+00) *
1162 &
1163                   xmins(ksystem, iplayers) * xmins(ksystem, iplayers)
1164 !
1165           ELSE
1166 !
1167               prof(jsystem, jplayers, ksystem) = 0.0
1168 !
1169           END IF
1170 !
1171       END IF
1172 !
1173 ! subtract the marginal cost
1174       prof(jsystem, jplayers, ksystem) = prof(jsystem, jplayers, ksystem) -
1175 &
1176           (asyn(jsystem)%player(jplayers)%fc +
1177 &
1178           asyn(jsystem)%player(jplayers)%a * x(jsystem, jplayers, ksystem) +
1179 &
1180           asyn(jsystem)%player(jplayers)%b * x(jsystem, jplayers, ksystem)**2.0d+00)
1181 !
1182       IF (jsystem .EQ. ksystem) THEN
1183           prof(jsystem, jplayers, ksystem) = prof(jsystem, jplayers, ksystem) -
1184 &
1185           uplift(ksystem) * uplar(iplayers)
1186 !
1187       END IF
1188 !
1189 !
1190 ! Calc the profit in the ficticius case the player was at the opposite system
1191 !
1192       IF (ksystem .EQ. 1) opsystem = 2
1193       IF (ksystem .EQ. 2) opsystem = 1
1194 !
1195       IF (asyn(opsystem)%pr_method .EQ. 0) THEN
1196 !
1197           xcurr = xx(iplayers)
1198           prof(jsystem, jplayers, opsystem) = spot(opsystem) * xx(iplayers) !
1199 ficticius from the oposite system
1200
1201       ELSE
1202 !
1203           xcurr = (spot(opsystem) - cur_ba(ksystem, iplayers)) / cur_bb(ksystem,
1204 iplayers)
1205
1206           IF (xcurr .GT. xmaxs(ksystem, iplayers)) xcurr = xmaxs(ksystem, iplayers)
1207           IF (xcurr .LT. xmin(s(ksystem, iplayers)) xcurr = 0.0
1208
1209           IF ( ABS(xcurr) .GT. eps_d ) THEN
1210
1211               prof(jsystem, jplayers, opsystem) =
1212 &
1213                   cur_ba(ksystem, iplayers) * xcurr +
1214 &

```

```

1215             (cur_bb(kssystem, iplayers) / 2.0d+00) * xcurr * xcurr +
1216 &
1217             (cur_bb(kssystem, iplayers) / 2.0d+00) *
1218 &
1219             xmins(kssystem, iplayers) * xmins(kssystem, iplayers)
1220
1221             ELSE
1222
1223             prof(jssystem, jplayers, opsystem) = 0.0
1224
1225             END IF
1226 !
1227             END IF
1228 !
1229 ! subtract the marginal cost
1230             prof(jssystem, jplayers, opsystem) = prof(jssystem, jplayers, opsystem) -
1231 &
1232             (asyn(jssystem)%player(jplayers)%fc +
1233 &
1234             asyn(jssystem)%player(jplayers)%a * xcurr +
1235 &
1236             asyn(jssystem)%player(jplayers)%b * xcurr**2.0d+00)
1237
1238 !
1239             j(jssystem, jplayers) = prof(jssystem, jplayers, kssystem)
1240
1241             avej(jssystem, jplayers) = avej(jssystem, jplayers) +
1242 &
1243             (j(jssystem, jplayers) - avej(jssystem, jplayers)) / REAL(iiterations
1244 + 1)
1245 !
1246             dj(jssystem, jplayers) = j(jssystem, jplayers) - jo(jssystem, jplayers)
1247 !
1248             jo(jssystem, jplayers) = j(jssystem, jplayers)
1249
1250
1251             END DO ! player
1252
1253             extra(kssystem) = purchases(kssystem) - d(kssystem)
1254             kwh(kssystem) = isocost(kssystem) / purchases(kssystem)
1255             avekwh(kssystem) = avekwh(kssystem) + (kwh(kssystem) - avekwh(kssystem)) /
1256 REAL(iiterations + 1)
1257 !
1258             IF (asyn(kssystem)%pr_method .EQ. 0) THEN
1259 !
1260             price(kssystem) = spot(kssystem)
1261             aveprice(kssystem) = avespot(kssystem)
1262 !
1263             ELSE
1264 !
1265             price(kssystem) = kwh(kssystem)
1266             aveprice(kssystem) = avekwh(kssystem)
1267 !
1268             END IF
1269
1270             sum_purchases(kssystem) = sum_purchases(kssystem) + purchases(kssystem)
1271             sum_extra(kssystem) = sum_extra(kssystem) + extra(kssystem)
1272             sum_isocost(kssystem) = sum_isocost(kssystem) + isocost(kssystem)
1273 !
1274             DEALLOCATE (xx, uplar)
1275 !
1276             END DO ! kssystem
1277 !
1278 !
1279             IF (gameopt%probability_on) THEN
1280
1281             Propability correction decision block
1282 !
1283             DO kssystem = 1, gameopt%nsystems
1284             DO iplayers = 1, asyn(kssystem)%nplayers
1285 !
1286             IF (kssystem .EQ. 1) opsystem = 2
1287             IF (kssystem .EQ. 2) opsystem = 1
1288 !
1289             IF (asyn(kssystem)%pr_method .EQ. 1 .AND. asyn(opsystem)%pr_method .EQ. 1
1290 .AND. &
1291             x(kssystem, iplayers, kssystem) .EQ. asyn(kssystem)%player(iplayers)%xmax
1292 .AND. &
1293             x(kssystem, iplayers, opsystem) .EQ. asyn(kssystem)%player(iplayers)%xmax)
1294 THEN
1295

```

```

1296 ! Special case where the two systems Pay as Bid and the x(1)=x(2)=xmax
1297 !
1298 !
1299           temp = spot(ksystem) * asyn(ksystem)%player(iplayers)%alpha_tolerance
1300 !
1301           IF ((spot(ksystem) - spot(opsystem)) .GT. temp) THEN
1302
1303               pprim(ksystem, iplayers) = pprim(ksystem, iplayers) +
1304 &
1305                   asyn(ksystem)%player(iplayers)%alpha_ini
1306
1307           ELSEIF ((spot(ksystem) - spot(opsystem)) .LT. temp) THEN
1308
1309               pprim(ksystem, iplayers) = pprim(ksystem, iplayers) -
1310 &
1311                   asyn(ksystem)%player(iplayers)%alpha_ini
1312
1313           ELSE
1314
1315               pprim(ksystem, iplayers) = pprim(ksystem, iplayers)
1316
1317           ENDIF
1318
1319           ELSE
1320 !
1321           temp = prof(ksystem, iplayers, ksystem) *
1322 asyn(ksystem)%player(iplayers)%alpha_tolerance
1323 !
1324           IF ( (prof(ksystem, iplayers, ksystem) - prof(ksystem, iplayers, opsystem))
1325 .GT. temp) THEN
1326 !
1327               pprim(ksystem, iplayers) = pprim(ksystem, iplayers) +
1328 &
1329                   asyn(ksystem)%player(iplayers)%alpha_ini
1330 !
1331           ELSE IF ( (prof(ksystem, iplayers, opsystem) - prof(ksystem, iplayers,
1332 ksystem)) .GT. temp) THEN
1333 !
1334               pprim(ksystem, iplayers) = pprim(ksystem, iplayers) -
1335 &
1336                   asyn(ksystem)%player(iplayers)%alpha_ini
1337 !
1338           ELSE
1339 !
1340               pprim(ksystem, iplayers) = pprim(ksystem, iplayers)
1341 !
1342           END IF
1343
1344           END IF
1345 !
1346           IF (pprim(ksystem, iplayers) .GT. 1.0) pprim(ksystem, iplayers) = 1.0
1347           IF (pprim(ksystem, iplayers) .LT. 0.0) pprim(ksystem, iplayers) = 0.0
1348           IF (asyn(ksystem)%player(iplayers)%xmin .GT. gameopt%interconne*1000.0d+00)
1349 pprim(ksystem, iplayers) = 1.0 ! an xmin> interconne den mporei na metaferi sto secondary
1350 !
1351           END DO
1352           END DO
1353 !
1354           ELSE
1355 !
1356 !           No Propability correction decision block
1357 !
1358           DO ksystem = 1, gameopt%nsystems
1359           DO iplayers = 1, asyn(ksystem)%nplayers
1360 !
1361               temp = 1.0 + asyn(ksystem)%player(iplayers)%alpha_tolerance
1362 !
1363               IF (ksystem .EQ. 1) opsystem = 2
1364               IF (ksystem .EQ. 2) opsystem = 1
1365 !
1366               IF (alpha(ksystem, iplayers) .EQ. PRIMARY ) THEN
1367 !
1368                   IF ( ABS(x(ksystem, iplayers, ksystem)) .LE. eps_d )
1369 THEN
1370 !
1371                       alpha(ksystem, iplayers) = SECONDARY
1372 !
1373                   END IF
1374
1375               ELSE
1376 !

```

```

1377         IF ( ABS(x(ksystem, iplayers, opsystem)) .LE. eps_d )
1378 THEN
1379 !
1380         alpha(ksystem, iplayers) = PRIMARY
1381     END IF
1382 !
1383     END IF
1384 !
1385     IF ( asyn(ksystem)%player(iplayers)%xmin .GT. gameopt%interconne*1000.0d+00)
1386 &
1387         alpha(ksystem, iplayers) = PRIMARY ! an xmin> interconne den
1388 mporei na metaferai sto secondary
1389 !
1390         END DO
1391     END DO
1392 !
1393     ENDIF
1394 !
1395     DO ksystem = 1, gameopt%nsystems
1396     DO iplayers = 1, asyn(ksystem)%nplayers
1397
1398         djt = dj(ksystem, iplayers)
1399         act = action(ksystem, iplayers, coef_flag(ksystem, iplayers))
1400 !
1401         IF (djt .GT. 0.0d+00 .AND. ABS(djt) .GT. eps_d) THEN
1402 !
1403             IF (act .EQ. 1) THEN
1404 !
1405                 IF ( pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr .OR.
1406 &
1407                     pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr )
1408 THEN
1409 !
1410                 IF (pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr)
1411 THEN
1412 !
1413                     dpi = theta(ksystem, iplayers) *
1414 &
1415                         pc(ksystem, iplayers, coef_flag(ksystem, iplayers))
1416 !
1417                     pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1418 &
1419                         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1420 !
1421                     ELSE
1422 !
1423                         dpi = theta(ksystem, iplayers) *
1424 &
1425                             pd(ksystem, iplayers,coef_flag(ksystem, iplayers))
1426 !
1427                             pd(ksystem, iplayers,coef_flag(ksystem, iplayers)) =
1428 &
1429                                 pd(ksystem, iplayers,coef_flag(ksystem, iplayers)) - dpi
1430 !
1431                     END IF
1432 !
1433                     pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1434 &
1435                         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi
1436 !
1437                     ELSE
1438 !
1439                         dpi = theta(ksystem, iplayers) * (1.0d+00 -
1440 &
1441                             pi(ksystem, iplayers, coef_flag(ksystem, iplayers))) /
1442 2.0d+00
1443 !
1444                         dp1 = pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1445                         dp2 = pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1446 !
1447                         IF (dp1 .LT. zr .OR. dp2 .LT. zr) THEN
1448 !
1449                             dpi = MIN( pd(ksystem, iplayers, coef_flag(ksystem, iplayers)),
1450 &
1451                                 pc(ksystem, iplayers,coef_flag(ksystem, iplayers)) )
1452 !
1453                         END IF
1454 !
1455                         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1456 &
1457                             pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi

```

```

1458 !
1459         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1460 &
1461         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1462         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1463 &
1464         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) + 2.0d+00 * dpi
1465 !
1466         END IF
1467 !
1468     ELSE IF (ACT .EQ. 2) THEN
1469 !
1470         IF (pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr .OR.
1471 &
1472         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr      )
1473 THEN
1474 !
1475         IF (pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr)
1476 THEN
1477 !
1478             dpi = theta(ksystem, iplayers) *
1479 &
1480             pc(ksystem, iplayers,coef_flag(ksystem, iplayers))
1481 !
1482             pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1483 &
1484             pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1485 !
1486         ELSE
1487 !
1488             dpi = theta(ksystem, iplayers) *
1489 &
1490             pi(ksystem, iplayers, coef_flag(ksystem, iplayers))
1491 !
1492             pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1493 &
1494             pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1495 !
1496         END IF
1497 !
1498             pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1499 &
1500             pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi
1501 !
1502         ELSE
1503 !
1504             dpi = theta(ksystem, iplayers) * (1.0d+00 -
1505 &
1506             pd(ksystem, iplayers, coef_flag(ksystem, iplayers))) /
1507 2.0d+00
1508 !
1509             dp1 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1510             dp2 = pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1511 !
1512             IF (dp1 .LT. zr .OR. DP2 .LT. ZR) THEN
1513 !
1514                 dpi = MIN( pi(ksystem, iplayers, coef_flag(ksystem, iplayers)),
1515 &
1516                 pc(ksystem, iplayers,coef_flag(ksystem, iplayers)) )
1517 !
1518             END IF
1519 !
1520             pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1521 &
1522             pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1523 !
1524             pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1525 &
1526             pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1527 !
1528             pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1529 &
1530             pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) + 2.0 * dpi
1531 !
1532         END IF
1533 !
1534     ELSE
1535 !
1536         IF (pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr .OR.
1537 &
1538         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr      )

```

```

1539 THEN
1540 !
1541         IF (pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) .LE. zr)
1542 THEN
1543 !
1544         dpi = theta(ksystem, iplayers) *
1545 &
1546             pd(ksystem, iplayers, coef_flag(ksystem, iplayers))
1547 !
1548         pd(ksystem, iplayers,coef_flag(ksystem, iplayers)) =
1549 &
1550         pd(ksystem, iplayers,coef_flag(ksystem, iplayers)) - dpi
1551 !
1552 ELSE
1553 !
1554         dpi = theta(ksystem, iplayers) *
1555 &
1556             pi(ksystem, iplayers, coef_flag(ksystem, iplayers))
1557 !
1558         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1559 &
1560         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1561 !
1562 END IF
1563 !
1564         pc(ksystem, iplayers,coef_flag(ksystem, iplayers)) =
1565 &
1566         pc(ksystem, iplayers,coef_flag(ksystem, iplayers)) + dpi
1567 !
1568 ELSE
1569 !
1570         dpi = theta(ksystem, iplayers) * (1.0d+00 -
1571 &
1572             pc(ksystem, iplayers, coef_flag(ksystem, iplayers))) /
1573 2.0d+00
1574 !
1575         dp1 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1576         dp2 = pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1577 !
1578         IF (dp1 .LT. zr .OR. dp2 .LT. zr) THEN
1579 !
1580             dpi = MIN( pi(ksystem, iplayers, coef_flag(ksystem, iplayers)),
1581 &
1582                 pd(ksystem, iplayers,coef_flag(ksystem, iplayers)) )
1583 !
1584         END IF
1585 !
1586         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1587 &
1588         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1589 !
1590         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1591 &
1592         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1593 !
1594         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1595 &
1596         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) + 2.0d+00 * dpi
1597 !
1598         END IF
1599 !
1600     END IF
1601 !
1602 ELSE
1603 !
1604     IF (ACT .EQ. 1) THEN
1605 !
1606         dpi = theta(ksystem, iplayers) *
1607 &
1608             pi(ksystem, iplayers,coef_flag(ksystem, iplayers))
1609 !
1610         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1611 &
1612         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1613 !
1614         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1615 &
1616         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1617 !
1618         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1619 &

```



```

1620         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi / 2.0d+00
1621 !
1622     ELSE IF (ACT .EQ. 2) THEN
1623 !
1624         dpi = theta(kssystem, iplayers) *
1625 &
1626         pd(kssystem, iplayers, coef_flag(kssystem, iplayers))
1627 !
1628         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1629 &
1630         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1631 !
1632         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1633 &
1634         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi / 2.0d+00
1635 !
1636         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1637 &
1638         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi / 2.0d+00
1639 !
1640     ELSE
1641 !
1642         dpi = theta(kssystem, iplayers) *
1643 &
1644         pc(kssystem, iplayers, coef_flag(kssystem, iplayers))
1645 !
1646         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1647 &
1648         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1649 !
1650         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1651 &
1652         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi / 2.0d+00
1653         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1654 &
1655         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi / 2.0d+00
1656 !
1657     END IF
1658 !
1659     END IF
1660 !
1661     ENDDO !players
1662 !
1663     ENDDO !ksystem
1664 !
1665     Detailed output of initial state
1666 !
1667     IF (gameopt%ngames .EQ. 1) THEN
1668         DO kssystem = 1, gameopt%nsystems
1669 !
1670         asyn(kssystem)%out_one_game(1, iterations+1) = REAL(price(kssystem), ikind4)
1671         asyn(kssystem)%out_one_game(2, iterations+1) = REAL(aveprice(kssystem), ikind4)
1672         asyn(kssystem)%out_one_game(3, iterations+1) = REAL(spot(kssystem), ikind4)
1673         asyn(kssystem)%out_one_game(4, iterations+1) = REAL(avespot(kssystem), ikind4)
1674 !
1675         DO iplayers = 1, asyn(kssystem)%nplayers
1676 !
1677         kcol = 4 + iplayers
1678         asyn(kssystem)%out_one_game(kcol, iterations+1) = x(kssystem, iplayers, 1)
1679 !
1680         END DO
1681 !
1682         DO iplayers = 1, asyn(kssystem)%nplayers
1683 !
1684         kcol = 4 + asyn(kssystem)%nplayers + iplayers
1685         asyn(kssystem)%out_one_game(kcol, iterations+1) = x(kssystem, iplayers, 2)
1686 !
1687         END DO
1688 !
1689     END DO
1690 !
1691     ENDDO
1692     kssystem = 1
1693     DO iplayers = 1, nplayers(kssystem)
1694 !
1695         jplayers = conne(kssystem, iplayers, 1)
1696         jsystem = conne(kssystem, iplayers, 2)
1697 !
1698         kcol = 4 + 2*asyn(jsystem)%nplayers + jplayers
1699         asyn(jsystem)%out_one_game(kcol, iterations+1) = cur_ba(kssystem, iplayers) +
1700 &

```

```

1701         cur_bb(kssystem, iplayers) * xmins(kssystem, iplayers)
1702 !
1703     END DO
1704 !
1705     kssystem = 2
1706     DO iplayers = 1, nplayers(kssystem)
1707 !
1708         jplayers = conne(kssystem, iplayers, 1)
1709         jsystem = conne(kssystem, iplayers, 2)
1710
1711         kcol = 4 + 3*asyn(jsystem)%nplayers + jplayers
1712         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(kssystem, iplayers) +
1713 &
1714         cur_bb(kssystem, iplayers) * xmins(kssystem, iplayers)
1715 !
1716     END DO
1717 !
1718     kssystem = 1
1719     DO iplayers = 1, nplayers(kssystem)
1720 !
1721         jplayers = conne(kssystem, iplayers, 1)
1722         jsystem = conne(kssystem, iplayers, 2)
1723
1724         kcol = 4 + 4*asyn(jsystem)%nplayers + jplayers
1725         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(kssystem, iplayers) +
1726 &
1727         cur_bb(kssystem, iplayers) * xmaxs(kssystem, iplayers)
1728 !
1729     END DO
1730 !
1731     kssystem = 2
1732     DO iplayers = 1, nplayers(kssystem)
1733 !
1734         jplayers = conne(kssystem, iplayers, 1)
1735         jsystem = conne(kssystem, iplayers, 2)
1736
1737         kcol = 4 + 5*asyn(jsystem)%nplayers + jplayers
1738         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(kssystem, iplayers) +
1739 &
1740         cur_bb(kssystem, iplayers) * xmaxs(kssystem, iplayers)
1741 !
1742     END DO
1743     kssystem = 1
1744     DO iplayers = 1, nplayers(kssystem)
1745 !
1746         jplayers = conne(kssystem, iplayers, 1)
1747         jsystem = conne(kssystem, iplayers, 2)
1748
1749         kcol = 4 + 6*asyn(jsystem)%nplayers + jplayers
1750         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmins(kssystem, iplayers)
1751 !
1752     END DO
1753 !
1754     kssystem = 2
1755     DO iplayers = 1, nplayers(kssystem)
1756 !
1757         jplayers = conne(kssystem, iplayers, 1)
1758         jsystem = conne(kssystem, iplayers, 2)
1759
1760         kcol = 4 + 7*asyn(jsystem)%nplayers + jplayers
1761         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmins(kssystem, iplayers)
1762 !
1763     END DO
1764 !
1765     kssystem = 1
1766     DO iplayers = 1, nplayers(kssystem)
1767 !
1768         jplayers = conne(kssystem, iplayers, 1)
1769         jsystem = conne(kssystem, iplayers, 2)
1770
1771         kcol = 4 + 8*asyn(jsystem)%nplayers + jplayers
1772         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmaxs(kssystem, iplayers)
1773 !
1774     END DO
1775 !
1776     kssystem = 2
1777     DO iplayers = 1, nplayers(kssystem)
1778 !
1779         jplayers = conne(kssystem, iplayers, 1)
1780         jsystem = conne(kssystem, iplayers, 2)
1781

```

```

1782         kcol = 4 + 9*asyn(jssystem)%nplayers + jplayers
1783         asyn(jssystem)%out_one_game(kcol, iterations+1) = xmaxs(kssystem, iplayers)
1784 !
1785         END DO
1786 !
1787         END IF
1788 !
1789         DEALLOCATE (cur_ba, cur_bb, xmin, xmax)
1790 !
1791     END DO ! iterations
1792 !
1793     DO ksystem = 1, gameopt%nsystems
1794 !
1795         IF (sumsp_sq(kssystem) .LT. BIG .AND. sum_sp(kssystem) .LT. BIG) THEN
1796             sp_stdevp = SQRT( ABS(REAL(asyn(kssystem)%niterations) * sumsp_sq(kssystem) -
1797 &
1798                 sum_sp(kssystem)**2.0d+00) ) /
1799 REAL(asyn(kssystem)%niterations)
1800             ELSE
1801 !
1802                 sp_stdevp = BIG
1803 !
1804             ENDIF
1805 !
1806             sumabs_pr = 0.0d+00
1807 !
1808             DO iterations = 1, asyn(kssystem)%niterations
1809 !
1810                 sumabs_pr = sumabs_pr + ABS(varpr(kssystem, iterations) - aveprice(kssystem))
1811 !
1812             END DO
1813 !
1814             sp_var = sumabs_pr / REAL(asyn(kssystem)%niterations)
1815 !
1816             asyn(kssystem)%out_many_games(1, igames) = REAL(aveprice(kssystem), ikind4)
1817             asyn(kssystem)%out_many_games(2, igames) = REAL(sp_stdevp, ikind4)
1818             asyn(kssystem)%out_many_games(3, igames) = REAL(sp_var, ikind4)
1819 !
1820             DO iplayers = 1, asyn(kssystem)%nplayers
1821                 k = iplayers + 3
1822                 asyn(kssystem)%out_many_games(k, igames) = REAL(avex(kssystem, iplayers, 1), ikind4)
1823             END DO
1824 !
1825             DO iplayers = 1, asyn(kssystem)%nplayers
1826                 k = asyn(kssystem)%nplayers + 3 + iplayers
1827                 asyn(kssystem)%out_many_games(k, igames) = REAL(avex(kssystem, iplayers, 2), ikind4)
1828             END DO
1829 !
1830             DO iplayers = 1, asyn(kssystem)%nplayers
1831                 k = 2 * asyn(kssystem)%nplayers + 3 + iplayers
1832                 asyn(kssystem)%out_many_games(k, igames) = REAL(avej(kssystem, iplayers), ikind4)
1833             END DO
1834 !
1835             DO iplayers = 1, asyn(kssystem)%nplayers
1836                 k = 3 * asyn(kssystem)%nplayers + 3
1837                 k = k + iplayers + (iplayers - 1)
1838 !
1839                 asyn(kssystem)%out_many_games(k, igames) = REAL(aveba(kssystem, iplayers), ikind4)
1840                 asyn(kssystem)%out_many_games(k+1, igames) = REAL(avebb(kssystem, iplayers), ikind4)
1841 !
1842             END DO
1843 !
1844             k = 5 * asyn(kssystem)%nplayers + 4
1845 !
1846             asyn(kssystem)%out_many_games(k, igames) = sum_purchases(kssystem)
1847             asyn(kssystem)%out_many_games(k+1, igames) = REAL(sum_isocost(kssystem), ikind4)
1848             asyn(kssystem)%out_many_games(k+2, igames) = REAL(sum_extra(kssystem), ikind4)
1849             asyn(kssystem)%out_many_games(k+3, igames) = REAL(avespot(kssystem), ikind4)
1850 !
1851             END DO
1852 !
1853         END DO ! games
1854 !
1855         DEALLOCATE (varpr)
1856 !
1857     DO ksystem = 1, gameopt%nsystems
1858 !
1859         IF ( errcounter(kssystem) .NE. 0) THEN
1860             kerr(kssystem) = errcounter(kssystem)
1861         ELSE
1862

```

```
1863             kerr(ksystem) = ASYN_SUCCESS
1864             END IF
1865 !
1866             END DO
1867 !
1868 !
1869 ! End Listing
1870 !.....
1871 !~~~~~
1872     END SUBROUTINE calc_asyn_2sys_ofall
1873 !~~~~~
1874     END MODULE asyn_two_sys_oneoffer_m
1875 !=====
1876 !
1877 !=====
```

```

1  !=====
2  !+ <A one line description of this module>
3  !=====
4  MODULE asyn_two_sys_split_m
5  !.....
6  !
7  ! Description:
8  !   <Say what this Module is for>
9  !
10 ! Current code owner : C.Skoulidas
11 !
12 ! History:
13 ! Version      Date      Comment
14 ! -----      -
15 ! <18.1>      <2011>   Original code  <C.Skoulidas>
16 !
17 ! Code description:
18 ! Language: Fortran 90.
19 !.....
20 ! Declarations:
21 !
22 ! Modules Used:
23 !   USE asyn_one_sys_m
24 !
25 ! Imported type definitions
26 ! Imported Parameters:
27 ! Imported scalar variables with intent (in):
28 ! Imported scalar variables with intent (out):
29 ! Imported Array variables with intent(in):
30 ! Imported Array variables with intent(out):
31 ! Imported routines:
32 !
33 !   IMPLICIT NONE
34 !
35 ! Global (Public) Declarations:
36 !
37 ! Global Type Definitions:
38 !
39 ! Global scalars:
40 ! Global arrays:
41 ! Local (Private) Declarations:
42 ! Local Type Definitions:
43 ! Local parameters:
44 ! Local scalars:
45 ! Local arrays:
46 ! Operator definitions:
47 !.....
48 !   CONTAINS
49 !-----
50 ! List Of Internal Procedures:
51 ! Subroutines:
52 !
53 !
54 ! Functions:
55 !
56 !
57 !-----
58 !=====
59 !+ <A one line description of this external subroutine>
60 !=====
61 !   SUBROUTINE calc_asyn_2sys_ofsplit (gameopt, asyn, kerr)
62 !.....
63 !
64 ! Description:
65 !   ASYN      KEEP ON BIDDING - Bids Above Price Cap NOT ALLOWED
66 !             LINEAR Marginal Cost
67 !             Optional Pricing Method (Uniform OR Pay-As-Bid)
68 !             NO use of GCOMP Routine
69 !             LAST REVISION: 25 OCTOBER 2011
70 !             ALLOWS Excess Capacity - ALLOWS Loosing at zero sales
71 !             REPEATED runs of the same game
72 !             StDeviation and Variance of SMP added
73 !
74 ! Method:
75 !   <Say how it does it: include references to external documentation>
76 !   <If this routine is divided in to sections, be brief here,
77 !       and put Method comments at the start of each section>
78 !
79 ! Current code owner : C.Skoulidas
80 !

```

```

81 ! History:
82 ! Version      Date      Comment
83 ! -----      ----      -
84 ! <18.1>      <2011> Original code <C.Skoulidas>
85 !
86 ! Code description:
87 !               Language: Fortran 90.
88 ! .....
89 ! Declarations:
90 !
91 ! Modules Used:
92 ! include statements
93 ! Subroutine Arguments
94 ! Scalar arguments with intent(in):
95     TYPE (gameopt_t) :: gameopt
96     TYPE (asyn_t)   :: asyn(gameopt%nsystems)
97     integer(4)      :: kerr(gameopt%nsystems)
98 ! Array arguments with intent(in):
99 ! Scalar arguments with intent(inout):
100 ! Array arguments with intent(inout):
101 ! Scalar arguments with intent(out):
102 ! Array arguments with intent(out):
103 ! Local parameters:
104     real(8), parameter      :: EPS_D = 1.0d-20, BIG = 1.0d+30, eps_s = 1.0e-8
105     real(8), parameter      :: BBMIN = 1.0d-05, BAMIN = 1.0d-05
106     real(8), parameter      :: zr = 1.d-020
107     integer(4), parameter    :: ikind4 = 4
108 ! Local scalars:
109     integer(4)               ::  igames,  iplayers,  iconst,  counter,
110 &
111                               jplayers,  k,
112 &
113                               iterations, act
114     integer(4)               ::  ksystem, jsystem, backcount, kcol
115     real(8)                  ::  y, d, sumper1, sumper2, rnd, pl, p2, djt,
116 &
117                               dpi, dp1, dp2, sp_stdevp, sumabs_pr, sp_var
118     real(8)                  ::  amax, amin, revtemp
119     real(8)                  ::  xtot, temp
120     real(8)                  ::  bamax, bbmax, availpower, alpha_prev
121
122 ! Local arrays:
123 !
124     integer(4) , dimension (gameopt%nsystems, asyn(1)%nplayers+asyn(2)%nplayers, 2) :: conne
125 ! conne - (:,:,1) = system the player belongs - (:,:,2) = the id numbering of the player
126 !
127     integer(4), dimension (gameopt%nsystems) :: nplayers
128     integer(4), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers))
129 :: &
130                                             player_count,
131 coef_flag
132     integer(4), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers), NCONST)
133 :: &
134
135 action
136 !
137     real(8), dimension (gameopt%nsystems)      :: sum_purchases, sum_extra, sum_isocost, extra, kwh,
138 &
139                                             avekwh, price, aveprice, spot, avespot, sum_sp,
140 &
141                                             sumsp_sq, purchases, isocost, averatio, isocost1,
142 uplift
143 !
144     real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers), NCONST) :: pi,
145 pd, pc
146     real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers),
147 gameopt%nsystems) :: &
148                                             x, rev, ratio,
149 avex
150     real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers)) :: alpha, bb,
151 ba, &
152                                             theta, j, dj, avej, jo, aveba,
153 &
154                                             avebb, uplar
155     real(8), dimension (gameopt%nsystems, MAX(asyn(1)%nplayers, asyn(2)%nplayers), NCONST) :: eps
156 !
157     real(8), allocatable, dimension (:,:) :: cur_ba, cur_bb, xmin, xmax
158 !
159     real(8), allocatable, dimension (:)   :: xx
160     logical(4)                            :: back
161     real(8), allocatable, dimension (:,:) :: varpr

```

```

162 ! .....
163 ! Listing
164 !
165     asyn(2)%niterations = asyn(1)%niterations
166 !
167     DO ksystem = 1, gameopt%nsystems
168 !
169         IF (gameopt%ngames .EQ. 1) asyn(ksystem)%out_one_game = 0.0d+00
170         asyn(ksystem)%out_many_games = 0.0d+00
171 !
172     END DO
173 !
174 ! initialize error index
175     kerr = ASYN_SUCCESS
176 !
177     ALLOCATE ( varpr(gameopt%nsystems, asyn(1)%niterations) )
178 !
179 ! Start game iterations
180 !
181     CALL random_seed()
182 !
183     DO igames = 1, gameopt%ngames
184 !
185 ! asyn%p_flag .EQ. 0 ! the following condition is always FALSE
186
187         DO ksystem = 1, gameopt%nsystems
188 !
189             IF (asyn(ksystem)%p_flag .EQ. 1) THEN
190 !
191                 DO iplayers = 1, asyn(ksystem)%nplayers
192 !
193                     DO iconst = 1, NCONST
194 !
195                         CALL random_number (y)
196                         asyn(ksystem)%player(iplayers)%period(iconst) =
197 INT(asyn(ksystem)%period_min + &
198                                     (asyn(ksystem)%period_max -asyn(ksystem)%period_min )
199 * y)
200 !
201                             ENDDO
202 !
203                     END DO
204 !
205                 END IF
206 !
207             END DO
208 !
209             Assign/Reassign initial values
210 !
211             DO ksystem = 1, gameopt%nsystems
212 !
213                 DO iplayers = 1, asyn(ksystem)%nplayers
214 !
215                     alpha(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%sys2_a
216                     bb(ksystem, iplayers) = 2.0d+00 * asyn(ksystem)%player(iplayers)%b
217                     ba(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%a
218 !
219 ! Correct alpha gaps due to the xmin at the secondary
220 !
221                     amin = asyn(ksystem)%player(iplayers)%xmin / (asyn(ksystem)%player(iplayers)%xmax -
222 &
223                                     asyn(ksystem)%player(iplayers)%xmin)
224
225                     amax = asyn(ksystem)%player(iplayers)%xmax / (asyn(ksystem)%player(iplayers)%xmax -
226 &
227                                     asyn(ksystem)%player(iplayers)%xmin)
228 !
229                     IF (alpha(ksystem, iplayers) .LT. amin) alpha(ksystem, iplayers) = 0.0d+00
230                     IF (alpha(ksystem, iplayers) .GT. 1.0d+00) alpha(ksystem, iplayers) = amax
231                     IF (alpha(ksystem, iplayers) .EQ. 1.0d+00) alpha(ksystem, iplayers) = 1.0d+00 -
232 eps_s ! the algorithm of spot does not work for xmin=xmax
233                     IF (alpha(ksystem, iplayers) .EQ. amin) alpha(ksystem, iplayers) = amin + eps_s !
234 the algorithm of spot does not work for xmin=xmax
235
236                     IF (amin .GT. 1.0d+00) alpha(ksystem, iplayers) = amax ! an amin> 1 -> xmax-xmin <
237 xmin ara sto secondary system den exei na metaferi pote energiea
238                     IF (asyn(ksystem)%player(iplayers)%xmin .GT. gameopt%interconne*1000.0d+00)
239 &
240                         alpha(ksystem, iplayers) = amax ! an xmin> interconne den mporei na
241 metaferi sto secondary
242 !

```

```

243         END DO
244 !
245     END DO
246 !
247     aveba(:, :) = ba(:, :)
248     avebb(:, :) = bb(:, :)
249 !
250     Initial bid parameters values and sums related to them
251 !
252     Assign/Reassign initial values (=zero) to Sums related with the
253     Bid Parameters - 1st Round
254 !
255     sum_sp = 0.0d+00
256     sumsp_sq = 0.0d+00
257 !
258     sum_purchases = 0.0d+00
259     sum_extra = 0.0d+00
260     sum_isocost = 0.0d+00
261 !
262     x(:, :, :) = 0.0d+00; rev(:, :, :) = 0.0d+00; ratio(:, :, :) = 0.0d+00; avex(:, :, :) = 0.0d+00
263 !
264 ! Find the active players
265 !
266     nplayers(:) = 0
267 !
268     DO ksystem = 1, gameopt%nssystems
269 !
270         IF (ksystem .EQ. 1) jsystem = 2
271         IF (ksystem .EQ. 2) jsystem = 1
272 !
273         DO iplayers = 1, asyn(ksystem)%nplayers
274 !
275             amin = asyn(ksystem)%player(iplayers)%xmin / (asyn(ksystem)%player(iplayers)%xmax -
276 &
277                 asyn(ksystem)%player(iplayers)%xmin)
278
279             amax = asyn(ksystem)%player(iplayers)%xmax / (asyn(ksystem)%player(iplayers)%xmax -
280 &
281                 asyn(ksystem)%player(iplayers)%xmin)
282 !
283             IF (alpha(ksystem, iplayers) .LE. 0.0d+00) THEN ! pazei mono sto secondary
284 !
285                 nplayers(jsystem) = nplayers(jsystem) + 1
286                 conne(jsystem, nplayers(jsystem), 1) = iplayers
287                 conne(jsystem, nplayers(jsystem), 2) = ksystem
288 !
289             ELSE IF (alpha(ksystem, iplayers) .GE. amin .AND. alpha(ksystem, iplayers) .LE.
290 1.0d+00) THEN ! kai sta duo
291 !
292                 nplayers(ksystem) = nplayers(ksystem) + 1
293                 conne(ksystem, nplayers(ksystem), 1) = iplayers
294                 conne(ksystem, nplayers(ksystem), 2) = ksystem
295 !
296                 nplayers(jsystem) = nplayers(jsystem) + 1
297                 conne(jsystem, nplayers(jsystem), 1) = iplayers
298                 conne(jsystem, nplayers(jsystem), 2) = ksystem
299 !
300             ELSE IF (alpha(ksystem, iplayers) .GE. amax) THEN ! pazei mono sto primary
301 !
302                 nplayers(ksystem) = nplayers(ksystem) + 1
303                 conne(ksystem, nplayers(ksystem), 1) = iplayers
304                 conne(ksystem, nplayers(ksystem), 2) = ksystem
305 !
306             END IF
307 !
308         END DO
309 !
310     END DO
311 !
312     ALLOCATE (cur_ba(gameopt%nssystems, MAX(nplayers(1), nplayers(2))),
313 &
314             cur_bb(gameopt%nssystems, MAX(nplayers(1), nplayers(2))),
315 &
316             xmin(gameopt%nssystems, MAX(nplayers(1), nplayers(2))),
317 &
318             xmax(gameopt%nssystems, MAX(nplayers(1), nplayers(2))))
319 !
320
321     Assing/Reassign values to bidding parameters and Calculate
322     the sums related with them - 1st Round
323 !

```



```

324      DO ksystem = 1, gameopt%nsystems
325      !
326          DO iplayers = 1, nplayers(ksystem)
327          !
328              jplayers = conne(ksystem, iplayers, 1)
329              jsystem = conne(ksystem, iplayers, 2)
330          !
331              amin = asyn(jsystem)%player(jplayers)%xmin / (asyn(jsystem)%player(jplayers)%xmax -
332      &
333                  asyn(jsystem)%player(jplayers)%xmin)
334              amax = asyn(jsystem)%player(jplayers)%xmax / (asyn(jsystem)%player(jplayers)%xmax -
335      &
336                  asyn(jsystem)%player(jplayers)%xmin)
337          !
338          !
339          IF (ksystem .EQ. jsystem) THEN !the player has the current system as primary
340          !
341              cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
342              cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
343          !
344              IF (alpha(jsystem, jplayers) .GE. amax) THEN ! mono primary
345          !
346                  xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
347                  xmaxs(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmax
348          !
349              ELSE ! kai sta duo
350          !
351                  xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
352                  xmaxs(ksystem, iplayers) = (asyn(jsystem)%player(jplayers)%xmax -
353      &
354                      asyn(jsystem)%player(jplayers)%xmin ) *
355      &
356                      alpha(jsystem, jplayers)
357          !
358              END IF
359          !
360          ELSE !the player has the current system as secondary
361          !
362              IF (alpha(jsystem, jplayers) .LE. 0.0d+00) THEN ! mono secondary
363          !
364                  cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
365                  cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
366          !
367                  xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
368                  xmaxs(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmax
369          !
370              ELSE ! kai sta duo
371          !
372                  cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
373          !
374                  cur_ba(ksystem, iplayers) = ba(jsystem, jplayers) +
375      &
376                      alpha(jsystem, jplayers) * cur_bb(ksystem, iplayers) *
377      &
378                      asyn(jsystem)%player(jplayers)%xmax -
379      &
380                      (1.0d+00 + alpha(jsystem, jplayers)) * cur_bb(ksystem, iplayers) *
381      &
382                      asyn(jsystem)%player(jplayers)%xmin
383          !
384                  xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
385                  xmaxs(ksystem, iplayers) = (1.0d+00 - alpha(jsystem, jplayers)) *
386      &
387                      asyn(jsystem)%player(jplayers)%xmax +
388      &
389                      asyn(jsystem)%player(jplayers)%xmin *
390      &
391                      alpha(jsystem, jplayers)
392          !
393              END IF
394          !
395          END IF
396          !
397          END DO
398          !
399      END DO
400      !
401      DO ksystem = 1, gameopt%nsystems
402      !
403          d = asyn(ksystem)%inid
404      !

```

```

405 ! Check for available power with respect to the demand
406     availpower = 0.0d+00
407 !
408     DO iplayers = 1, nplayers(ksystem)
409 !
410         availpower = availpower + xmaxs(ksystem, iplayers)
411
412     END DO
413 !
414     IF (availpower .LT. d) THEN
415 !
416         kerr(ksystem) = ERR_DEMAND
417 !
418         RETURN
419 !
420     END IF
421 !
422     ALLOCATE ( xx(nplayers(ksystem)) )
423 !
424     CALL spot_price(nplayers(ksystem), cur_ba(ksystem, 1:nplayers(ksystem)),
425 &
426                 cur_bb(ksystem, 1:nplayers(ksystem)), xmin(ksystem,
427 1:nplayers(ksystem)), &
428                 xmaxs(ksystem, 1:nplayers(ksystem)), d, kerr(ksystem), spot(ksystem),
429 xx)
430 !
431     IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
432 !
433 ! Calculate the isocost (isocost1) from the first cleaning
434 !
435     isocost1(ksystem) = 0.0d+00
436 !
437     DO iplayers = 1, nplayers(ksystem)
438 !
439         IF (asyn(ksystem)%pr_method .EQ. 0) THEN
440
441             revtemp = spot(ksystem) * xx(iplayers)
442 !
443         ELSE
444 !
445             revtemp = 0.0d+00
446             IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
447 !
448                 revtemp = cur_ba(ksystem, iplayers) * xx(iplayers) +
449 &
450                         (cur_bb(ksystem, iplayers) / 2.0d+00) * xx(iplayers) *
451 xx(iplayers) + &
452                         (cur_bb(ksystem, iplayers) / 2.0d+00) *
453 &
454                         xmin(ksystem, iplayers) * xmin(ksystem, iplayers)
455
456             END IF
457         END IF
458 !
459         isocost1(ksystem) = isocost1(ksystem) + revtemp
460 !
461     END DO
462 !
463     CALL capacity_constraint (gameopt, asyn, nplayers(ksystem),
464 &
465                 cur_ba(ksystem, 1:nplayers(ksystem)), cur_bb(ksystem,
466 1:nplayers(ksystem)), &
467                 xmin(ksystem, 1:nplayers(ksystem)), xmaxs(ksystem,
468 1:nplayers(ksystem)), &
469                 d, kerr(ksystem), xx, ksystem, conne(ksystem, 1:nplayers(ksystem), :))
470 !
471     IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
472 !
473     avspot(ksystem) = spot(ksystem)
474 !
475     varpr(ksystem, 1) = spot(ksystem)
476 !
477     sum_sp(ksystem) = spot(ksystem)
478     sumsp_sq(ksystem) = spot(ksystem) * spot(ksystem)
479 !
480     purchases(ksystem) = 0.0d+00
481     isocost(ksystem) = 0.0d+00
482     averatio(ksystem) = 0.0d+00
483     counter = 0
484 !
485     DO iplayers = 1, nplayers(ksystem)

```

```

486 !
487     purchases(kssystem) = purchases(kssystem) + xx(iplayers)
488 !
489     jplayers = conne(kssystem, iplayers, 1)
490     jssystem = conne(kssystem, iplayers, 2)
491 !
492     x(jssystem, jplayers, kssystem) = xx(iplayers)
493     avex(jssystem, jplayers, kssystem) = x(jssystem, jplayers, kssystem)
494 !
495     IF (asyn(kssystem)%pr_method .EQ. 0) THEN
496 !
497         revtemp = spot(kssystem) * xx(iplayers)
498 !
499         rev(jssystem, jplayers, kssystem) = revtemp
500 !
501     ELSE
502 !
503         revtemp = 0.0d+00
504         IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
505 !
506             revtemp = cur_ba(kssystem, iplayers) * xx(iplayers) +
507 &
508                 (cur_bb(kssystem, iplayers) / 2.0d+00) * xx(iplayers) *
509 xx(iplayers) + &
510                 (cur_bb(kssystem, iplayers) / 2.0d+00) *
511 &
512                 xmins(kssystem, iplayers) * xmins(kssystem, iplayers)
513 !
514             END IF
515 !
516             rev(jssystem, jplayers, kssystem) = revtemp
517 !
518         END IF
519 !
520         isocost(kssystem) = isocost(kssystem) + revtemp
521 !
522         IF ( ABS(x(jssystem, jplayers, kssystem)) .GT. eps_d ) THEN
523 !
524             ratio(jssystem, jplayers, kssystem) = revtemp / x(jssystem, jplayers, kssystem)
525             averatio(kssystem) = averatio(kssystem) + ratio(jssystem, jplayers, kssystem)
526             counter = counter + 1
527 !
528         ELSE
529 !
530             ratio(jssystem, jplayers, kssystem) = 0.0d+00
531 !
532         ENDIF
533 !
534     END DO
535 !
536     uplift(kssystem) = isocost(kssystem) - isocost1(kssystem)
537 !
538     averatio(kssystem) = averatio(kssystem) / REAL(counter)
539 !
540     extra(kssystem) = purchases(kssystem) - d
541     kwh(kssystem) = isocost(kssystem) / purchases(kssystem)
542     avekwh(kssystem) = kwh(kssystem)
543 !
544     IF (asyn(kssystem)%pr_method .EQ. 0) THEN
545 !
546         price(kssystem) = spot(kssystem)
547         aveprice(kssystem) = avespot(kssystem)
548 !
549     ELSE
550 !
551         price(kssystem) = kwh(kssystem)
552         aveprice(kssystem) = avekwh(kssystem)
553 !
554     END IF
555 !
556     sum_purchases(kssystem) = purchases(kssystem)
557     sum_extra(kssystem) = extra(kssystem)
558     sum_isocost(kssystem) = isocost(kssystem)
559 !
560     DEALLOCATE (xx)
561 !
562     END DO ! kssystem
563 !
564 ! Calculate the uplift allocation ratio
565     DO ksystem = 1, gameopt%nsystems
566 !

```

```

567         xtot = 0.0d+00
568         DO iplayers = 1, asyn(kssystem)%nplayers
569             xtot = xtot + x(kssystem, iplayers, kssystem)
570         END DO
571 !
572         DO iplayers = 1, asyn(kssystem)%nplayers
573 !
574             IF (xtot .GT. eps_d) THEN
575                 uplar(kssystem, iplayers) = x(kssystem, iplayers, kssystem) / xtot
576             ELSE
577                 uplar(kssystem, iplayers) = 0.0d+00
578             END IF
579         END DO
580 !
581     END DO
582 !
583     DO ksystem = 1, gameopt%nsystems
584         DO iplayers = 1, asyn(kssystem)%nplayers
585 !
586             jo(kssystem, iplayers) = 0.0d+00
587 !
588             xtot = 0.0d+00
589             DO jsystem = 1, gameopt%nsystems
590 !
591                 jo(kssystem, iplayers) = jo(kssystem, iplayers) + rev(kssystem, iplayers, jsystem)
592                 xtot = xtot + x(kssystem, iplayers, jsystem)
593 !
594             END DO
595
596             jo(kssystem, iplayers) = jo(kssystem, iplayers) - (asyn(kssystem)%player(iplayers)%fc
597 + &
598                 asyn(kssystem)%player(iplayers)%a * xtot +
599 &
600                 asyn(kssystem)%player(iplayers)%b * xtot * xtot ) -
601 &
602                 uplift(kssystem) * uplar(kssystem, iplayers)
603 !
604             avej(kssystem, iplayers) = jo(kssystem, iplayers)
605 !
606         END DO
607     ENDDO
608 !
609     Detailed output of initial state
610 !
611     IF (gameopt%ngames .EQ. 1) THEN
612 !
613         DO ksystem = 1, gameopt%nsystems
614 !
615             asyn(kssystem)%out_one_game(1, 1) = REAL(price(kssystem), ikind4)
616             asyn(kssystem)%out_one_game(2, 1) = REAL(aveprice(kssystem), ikind4)
617             asyn(kssystem)%out_one_game(3, 1) = REAL(spot(kssystem), ikind4)
618             asyn(kssystem)%out_one_game(4, 1) = REAL(avespot(kssystem), ikind4)
619 !
620             DO iplayers = 1, asyn(kssystem)%nplayers
621 !
622                 kcol = 4 + iplayers
623                 asyn(kssystem)%out_one_game(kcol, 1) = x(kssystem, iplayers, 1)
624 !
625             END DO
626 !
627             DO iplayers = 1, asyn(kssystem)%nplayers
628 !
629                 kcol = 4 + asyn(kssystem)%nplayers + iplayers
630                 asyn(kssystem)%out_one_game(kcol, 1) = x(kssystem, iplayers, 2)
631 !
632             END DO
633 !
634         END DO
635 !
636         ksystem = 1
637         DO iplayers = 1, nplayers(kssystem)
638 !
639             jplayers = conne(kssystem, iplayers, 1)
640             jsystem = conne(kssystem, iplayers, 2)
641
642             kcol = 4 + 2*asyn(jsystem)%nplayers + jplayers
643             asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(kssystem, iplayers) +
644 &
645             cur_bb(kssystem, iplayers) * xmin(kssystem, iplayers)
646 !
647         END DO

```

```

648 !
649     ksystem = 2
650     DO iplayers = 1, nplayers(ksystem)
651 !
652         jplayers = conne(ksystem, iplayers, 1)
653         jsystem = conne(ksystem, iplayers, 2)
654
655         kcol = 4 + 3*asyn(jsystem)%nplayers + jplayers
656         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
657 &
658         cur_bb(ksystem, iplayers) * xmin(ksystem, iplayers)
659 !
660     END DO
661 !
662     ksystem = 1
663     DO iplayers = 1, nplayers(ksystem)
664 !
665         jplayers = conne(ksystem, iplayers, 1)
666         jsystem = conne(ksystem, iplayers, 2)
667
668         kcol = 4 + 4*asyn(jsystem)%nplayers + jplayers
669         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
670 &
671         cur_bb(ksystem, iplayers) * xmax(ksystem, iplayers)
672 !
673     END DO
674 !
675     ksystem = 2
676     DO iplayers = 1, nplayers(ksystem)
677 !
678         jplayers = conne(ksystem, iplayers, 1)
679         jsystem = conne(ksystem, iplayers, 2)
680
681         kcol = 4 + 5*asyn(jsystem)%nplayers + jplayers
682         asyn(jsystem)%out_one_game(kcol, 1) = cur_ba(ksystem, iplayers) +
683 &
684         cur_bb(ksystem, iplayers) * xmax(ksystem, iplayers)
685 !
686     END DO
687 !
688     ksystem = 1
689     DO iplayers = 1, nplayers(ksystem)
690 !
691         jplayers = conne(ksystem, iplayers, 1)
692         jsystem = conne(ksystem, iplayers, 2)
693
694         kcol = 4 + 6*asyn(jsystem)%nplayers + jplayers
695         asyn(jsystem)%out_one_game(kcol, 1) = xmin(ksystem, iplayers)
696 !
697     END DO
698 !
699     ksystem = 2
700     DO iplayers = 1, nplayers(ksystem)
701 !
702         jplayers = conne(ksystem, iplayers, 1)
703         jsystem = conne(ksystem, iplayers, 2)
704
705         kcol = 4 + 7*asyn(jsystem)%nplayers + jplayers
706         asyn(jsystem)%out_one_game(kcol, 1) = xmin(ksystem, iplayers)
707 !
708     END DO
709 !
710     ksystem = 1
711     DO iplayers = 1, nplayers(ksystem)
712 !
713         jplayers = conne(ksystem, iplayers, 1)
714         jsystem = conne(ksystem, iplayers, 2)
715
716         kcol = 4 + 8*asyn(jsystem)%nplayers + jplayers
717         asyn(jsystem)%out_one_game(kcol, 1) = xmax(ksystem, iplayers)
718 !
719     END DO
720 !
721     ksystem = 2
722     DO iplayers = 1, nplayers(ksystem)
723 !
724         jplayers = conne(ksystem, iplayers, 1)
725         jsystem = conne(ksystem, iplayers, 2)
726
727         kcol = 4 + 9*asyn(jsystem)%nplayers + jplayers
728         asyn(jsystem)%out_one_game(kcol, 1) = xmax(ksystem, iplayers)

```

```

729 !
730         END DO
731 !
732     END IF
733 !
734         DEALLOCATE (cur_ba, cur_bb, xmin, xmax)
735 !
736 !-----Iterations-----
737 ----
738 !
739         Assign/Reassign initial values to Behaviour Probabilities
740 !
741     DO ksystem = 1, gameopt%nsystems
742 !
743         DO iplayers = 1, asyn(ksystem)%nplayers
744 !
745             pi(ksystem, iplayers, :) = asyn(ksystem)%player(iplayers)%inipi(:)
746             pd(ksystem, iplayers, :) = asyn(ksystem)%player(iplayers)%inipd(:)
747             pc(ksystem, iplayers, :) = asyn(ksystem)%player(iplayers)%inipc(:)
748 !
749         END DO
750 !
751     END DO
752 !
753     player_count(:, :) = 0
754 !
755     DO iiterations = 1, asyn(1)%niterations-1
756 !
757         Random desicion & optimization
758 !
759         Start iterative procedure for each player
760 !
761         DO ksystem = 1, gameopt%nsystems
762 !
763             DO iplayers = 1, asyn(ksystem)%nplayers
764 !
765                 sumper1 = asyn(ksystem)%player(iplayers)%period(1)
766                 sumper2 = asyn(ksystem)%player(iplayers)%period(1) +
767 &
768                 asyn(ksystem)%player(iplayers)%period(2)
769 !
770                 IF (player_count(ksystem, iplayers) .GE. sumper2) player_count(ksystem,
771 iplayers) = 0
772 !
773                 player_count(ksystem, iplayers) = player_count(ksystem, iplayers) + 1
774 !
775                 Decision block that assigns value only to one of the BA, BB parameter's step
776                 (and keep the others zero) depending on the current parameter adaption period
777 !
778                 IF (player_count(ksystem, iplayers) .LE. sumper1) THEN
779 !
780                     coef_flag(ksystem, iplayers) = 1
781 !
782                     IF (iiterations .GT. INT(asyn(ksystem)%estep_decr *
783 (asyn(ksystem)%niterations))) THEN !IF THEN Block that
784 !
785                         eps(ksystem, iplayers, 1) = asyn(ksystem)%player(iplayers)%e_init(1) *
786 &
787                         (asyn(ksystem)%estep_decr *
788 &
789                         REAL(asyn(ksystem)%niterations)) / REAL(iiterations + 1) !
790 gradually decreases
791 !
792                     ELSE !the step size of BA
793 !
794                         eps(ksystem, iplayers, 1) = asyn(ksystem)%player(iplayers)%e_init(1)
795 !
796                     END IF
797 !
798                     eps(ksystem, iplayers, 2) = 0.0d+00
799 !
800                 ELSE IF (player_count(ksystem, iplayers) .LE. sumper2) THEN
801 !
802                     coef_flag(ksystem, iplayers) = 2
803                     eps(ksystem, iplayers, 1) = 0.0d+00
804 !
805                     IF (iiterations .GT. INT(asyn(ksystem)%estep_decr *
806 (asyn(ksystem)%niterations))) THEN !IF THEN Block that
807 !
808 
```

```

810             eps(ksystem, iplayers, 2) = asyn(ksystem)%player(iplayers)%e_init(2) *
811 &
812             (asyn(ksystem)%estep_decr *
813 &
814             REAL(asyn(ksystem)%niterations)) / REAL(iiterations + 1)
815 !gradually decreases
816 !
817             ELSE !the step size of BB
818 !
819             eps(ksystem, iplayers, 2) = asyn(ksystem)%player(iplayers)%e_init(2)
820 !
821             END IF
822 !
823             END IF
824 !
825 !         Decision block that assigns value to  $\theta$ 
826 !
827             IF (iiterations .GT. INT(asyn(ksystem)%thstep_decr *
828 (asyn(ksystem)%niterations))) THEN !IF THEN Block that
829 !
830             theta(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%theta_ini *
831 &
832             (asyn(ksystem)%thstep_decr *
833 &
834             REAL(asyn(ksystem)%niterations)) / REAL(iiterations + 1)
835 !gradually decreases
836 !
837             ELSE !  $\theta$  step size
838 !
839             theta(ksystem, iplayers) = asyn(ksystem)%player(iplayers)%theta_ini
840 !
841             END IF
842 !
843 !         Choose a random value from the probabilistic action profile
844 !         for the in turn parameter (BA or BB) and
845 !         adjust the in turn parameter (BA or BB) in compliance with
846 !         the randomly selected value
847 !
848             CALL random_number (y)
849             rnd = y
850 !
851             p1 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers))
852             p2 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) +
853 &
854             pd(ksystem, iplayers, coef_flag(ksystem, iplayers))
855 !
856             IF (rnd .LE. p1) THEN ! increase
857 !
858             action(ksystem, iplayers, coef_flag(ksystem, iplayers)) = 1
859             ba(ksystem, iplayers) = ba(ksystem, iplayers) +
860 &
861             eps(ksystem, iplayers, 1) * ba(ksystem, iplayers)
862             bb(ksystem, iplayers) = bb(ksystem, iplayers) +
863 &
864             eps(ksystem, iplayers, 2) * bb(ksystem, iplayers)
865 !
866             ELSE IF (rnd .LE. p2) THEN ! decrease
867 !
868             action(ksystem, iplayers, coef_flag(ksystem, iplayers)) = 2
869             ba(ksystem, iplayers) = ba(ksystem, iplayers) -
870 &
871             eps(ksystem, iplayers, 1) * ba(ksystem, iplayers)
872             bb(ksystem, iplayers) = bb(ksystem, iplayers) -
873 &
874             eps(ksystem, iplayers, 2) * bb(ksystem, iplayers)
875 &
876             eps(ksystem, iplayers, 2) * bb(ksystem, iplayers)
877 !
878             ELSE ! stabilize
879 !
880 !
881             action(ksystem, iplayers, coef_flag(ksystem, iplayers)) = 3
882             ba(ksystem, iplayers) = ba(ksystem, iplayers)
883             bb(ksystem, iplayers) = bb(ksystem, iplayers)
884 !
885             END IF
886 !
887             IF (bb(ksystem, iplayers) .LT. BBMIN) bb(ksystem, iplayers) = BBMIN
888             IF (ba(ksystem, iplayers) .LT. BAMIN) ba(ksystem, iplayers) = BAMIN
889 !
890 !         adjust the in turn parameter alpha in compliance with the previous step results

```

```

891 !
892         IF (ksystem .EQ. 1) jsystem = 2
893         IF (ksystem .EQ. 2) jsystem = 1
894 !
895         amin = asyn(ksystem)%player(iplayers)%xmin /
896 (asyn(ksystem)%player(iplayers)%xmax - &
897 asyn(ksystem)%player(iplayers)%xmin)
898 !
899         amax = asyn(ksystem)%player(iplayers)%xmax /
900 (asyn(ksystem)%player(iplayers)%xmax - &
901 asyn(ksystem)%player(iplayers)%xmin)
902 !
903         IF (alpha(ksystem, iplayers) .GE. amin .AND. alpha(ksystem, iplayers) .LE.
904 1.0D+00) THEN ! paizei kai sta duo
905 !
906         IF (ABS(ratio(ksystem, iplayers, jsystem)) .GT. eps_d) THEN ! sto secundary
907 poulaei kati
908 !
909         temp = ratio(ksystem, iplayers, ksystem) / ratio(ksystem, iplayers,
910 jsystem)
911 !
912         ELSE ! sto secondary epaikse kai den poulise tipote ara metaferi sto
913 primary
914 !
915         temp = 1.0d+00 + 2.0d+00 *
916 asyn(ksystem)%player(iplayers)%alpha_tolerance
917 !
918         END IF
919 !
920         IF (temp .GE. 1.0d+00 + asyn(ksystem)%player(iplayers)%alpha_tolerance)
921 THEN ! metefere sto primary
922 !
923         IF ( alpha(ksystem, iplayers) .EQ. 1.0d+00-eps_s .AND. x(ksystem,
924 iplayers, jsystem) .GT. eps_d) GO TO 30
925 ! (an exei xmin sto secondary, metaferi mono otan den poulaei)
926 !
927         alpha_prev = alpha(ksystem, iplayers)
928 alpha(ksystem, iplayers) = alpha(ksystem, iplayers) +
929 &
930         alpha(ksystem, iplayers) *
931 asyn(ksystem)%player(iplayers)%alpha_ini
932 !
933         IF (alpha_prev .LT. 1.0d+00-eps_s .AND. alpha(ksystem, iplayers) .GT.
934 1.0d+00) &
935         alpha(ksystem, iplayers) =
936 1.0d+00-eps_s
937 ! kata ti metavasi apo to secondary sto primary aneksartitos vimatos to teleutaio instance einai,
938 primary: xmin - (xmax-xmin), second: xmin-xmin
939 !
940         ELSE IF (temp .LE. 1.0d+00 -
941 asyn(ksystem)%player(iplayers)%alpha_tolerance) THEN ! metefere sto secondary
942 !
943         IF ( alpha(ksystem, iplayers) .EQ. amin+eps_s .AND. x(ksystem,
944 iplayers, ksystem) .GT. eps_d) GO TO 30
945 ! (an exei xmin sto primary, metaferi mono otan den poulaei)
946 !
947         alpha_prev = alpha(ksystem, iplayers)
948 !
949         alpha(ksystem, iplayers) = alpha(ksystem, iplayers) -
950 &
951         alpha(ksystem, iplayers) *
952 asyn(ksystem)%player(iplayers)%alpha_ini
953 !
954         IF (alpha_prev .GT. amin+eps_s .AND. alpha(ksystem, iplayers) .LT.
955 amin) &
956         alpha(ksystem, iplayers) =
957 amin+eps_s
958 ! kata ti metavasi apo to primary sto secondary, aneksartitos vimatos to teleutaio instance einai,
959 secondary: xmin - (xmax-xmin), primary: xmin-xmin
960 !
961         END IF
962 !
963         ELSE ! paizei se ena apo ta duo
964 !
965         temp = 1.0d+00 + asyn(ksystem)%player(iplayers)%alpha_tolerance
966 !
967         IF (alpha(ksystem, iplayers) .GE. amax) THEN ! epaikse mono sto primary

```



```

972 !
973             IF (averatio(jsystem) .GT. temp*ratio(ksystem, iplayers, ksystem) .AND.
974 &
975                 x(ksystem, iplayers, ksystem) .LE.
976 (asyn(ksystem)%player(iplayers)%xmax - &
977                                     asyn(ksystem)%player(iplayers)%xmin
978 ) ) THEN
979 !
980 ! metefere na paiksei kai sto secondary (an poulaei oli tin posotita na min metferei)
981 !
982             alpha(ksystem, iplayers) = 1.0d+00-eps_s
983 !
984             END IF
985 !
986             ELSE IF (alpha(ksystem, iplayers) .LE. 0.0) THEN !epaikse mono sto
987 secondary
988 !
989             IF (averatio(ksystem) .GT. temp*ratio(ksystem, iplayers, jsystem).AND.
990 &
991                 x(ksystem, iplayers, jsystem) .LE.
992 (asyn(ksystem)%player(iplayers)%xmax - &
993 asyn(ksystem)%player(iplayers)%xmin) ) THEN
994 ! metefere na paiksei kai sto primary (an poulaei oli tin posotita na min metferei)
995 !
996             alpha(ksystem, iplayers) = amin+eps_s
997 !
998             END IF
999
1000 !
1001             END IF
1002 !
1003             END IF
1004 !
1005             30 CONTINUE
1006 !
1007 ! check the maximum values for alpha
1008
1009
1010             IF (alpha(ksystem, iplayers) .LT. amin) alpha(ksystem, iplayers) = 0.0d+00
1011             IF (alpha(ksystem, iplayers) .GT. 1.0d+00) alpha(ksystem, iplayers) = amax
1012             IF (alpha(ksystem, iplayers) .EQ. 1.0d+00)
1013 &
1014                 alpha(ksystem, iplayers) = 1.0d+00 - eps_s ! the algorithm of
1015 spit does not work for xmin=xmax
1016 !
1017             IF (alpha(ksystem, iplayers) .EQ. amin) alpha(ksystem, iplayers) = amin + eps_s
1018 ! the algorithm of spit does not work for xmin=xmax
1019
1020             IF (amin .GT. 1.0d+00) alpha(ksystem, iplayers) = amax ! an amin> 1 -> xmax-
1021 xmin < xmin ara sto secondary system den exei na metaferi pote energieia
1022             IF (asyn(ksystem)%player(iplayers)%xmin .GT. gameopt%interconne*1000.0d+00)
1023 &
1024                 alpha(ksystem, iplayers) = amax ! an xmin> interconne den
1025 mporei na metaferi sto secondary
1026 !
1027             ENDDO ! players
1028 !
1029             ENDDO ! systems
1030 !
1031 ! Find the active players
1032 !
1033             nplayers(:) = 0
1034 !
1035             x(:, :, :) = 0.0d+00; rev(:, :, :) = 0.0d+00; ratio(:, :, :) = 0.0d+00
1036 !
1037             DO ksystem = 1, gameopt%nsystems
1038 !
1039                 IF (ksystem .EQ. 1) jsystem = 2
1040                 IF (ksystem .EQ. 2) jsystem = 1
1041 !
1042                 DO iplayers = 1, asyn(ksystem)%nplayers
1043 !
1044                     amin = asyn(ksystem)%player(iplayers)%xmin /
1045 (asyn(ksystem)%player(iplayers)%xmax - &
1046 asyn(ksystem)%player(iplayers)%xmin)
1047 !
1048                     amax = asyn(ksystem)%player(iplayers)%xmax /
1049 (asyn(ksystem)%player(iplayers)%xmax - &
1050 asyn(ksystem)%player(iplayers)%xmin)
1051
1052

```

```

1053 !
1054         IF (alpha(ksystem, iplayers) .LE. 0.0d+00) THEN ! pazei mono sto secondary
1055 !
1056             nplayers(jsystem) = nplayers(jsystem) + 1
1057             conne(jsystem, nplayers(jsystem), 1) = iplayers
1058             conne(jsystem, nplayers(jsystem), 2) = ksystem
1059 !
1060         ELSE IF (alpha(ksystem, iplayers) .GE. amin .AND. alpha(ksystem, iplayers) .LE.
1061 1.0d+00) THEN ! kai sta duo
1062 !
1063             nplayers(ksystem) = nplayers(ksystem) + 1
1064             conne(ksystem, nplayers(ksystem), 1) = iplayers
1065             conne(ksystem, nplayers(ksystem), 2) = ksystem
1066 !
1067             nplayers(jsystem) = nplayers(jsystem) + 1
1068             conne(jsystem, nplayers(jsystem), 1) = iplayers
1069             conne(jsystem, nplayers(jsystem), 2) = ksystem
1070 !
1071         ELSE IF (alpha(ksystem, iplayers) .GE. amax) THEN ! pazei mono sto primary
1072 !
1073             nplayers(ksystem) = nplayers(ksystem) + 1
1074             conne(ksystem, nplayers(ksystem), 1) = iplayers
1075             conne(ksystem, nplayers(ksystem), 2) = ksystem
1076 !
1077         END IF
1078 !
1079     END DO
1080 !
1081     END DO
1082 !
1083     ALLOCATE ( cur_ba(gameopt%nsystems, MAX(nplayers(1),nplayers(2))),
1084 &
1085             cur_bb(gameopt%nsystems, MAX(nplayers(1),nplayers(2))),
1086 &
1087             xmins(gameopt%nsystems, MAX(nplayers(1),nplayers(2))),
1088 &
1089             xmaxs(gameopt%nsystems, MAX(nplayers(1),nplayers(2))) )
1090 !
1091     backcount = 0
1092     CONTINUE
1093     back = .FALSE.
1094 !
1095     DO ksystem = 1, gameopt%nsystems
1096 !
1097         DO iplayers = 1, nplayers(ksystem)
1098 !
1099             jplayers = conne(ksystem, iplayers, 1)
1100             jsystem = conne(ksystem, iplayers, 2)
1101 !
1102             amin = asyn(jsystem)%player(jplayers)%xmin /
1103 (asyn(jsystem)%player(jplayers)%xmax - &
1104 asyn(jsystem)%player(jplayers)%xmin)
1105 !
1106             amax = asyn(jsystem)%player(jplayers)%xmax /
1107 (asyn(jsystem)%player(jplayers)%xmax - &
1108 asyn(jsystem)%player(jplayers)%xmin)
1109 !
1110             IF (ksystem .EQ. jsystem) THEN !the player has the current system as primary
1111 !
1112                 IF (alpha(jsystem, jplayers) .GE. amax) THEN ! mono primary
1113 !
1114                     xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
1115                     xmaxs(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmax
1116 !
1117                 ELSE ! kai sta duo
1118 !
1119                     xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
1120                     xmaxs(ksystem, iplayers) = (asyn(jsystem)%player(jplayers)%xmax -
1121 &
1122 asyn(jsystem)%player(jplayers)%xmin ) *
1123 &
1124 alpha(jsystem, jplayers)
1125 !
1126                 END IF
1127 !
1128             cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
1129             cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
1130 !
1131 !
1132 !
1133 !

```

```

1134      bamax = asyn(ksystem)%pricecap -
1135 &
1136      (cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers))
1137 !
1138      IF (coef_flag(jsystem, jplayers) .EQ. 1 .AND. cur_ba(ksystem, iplayers)
1139 .GT. bamax) THEN
1140 !
1141      IF (bamax .LT. BAMIN) THEN
1142 !
1143      bbmax = (asyn(ksystem)%pricecap - BAMIN) / xmaxs(ksystem, iplayers)
1144 !
1145      ba(jsystem, jplayers) = BAMIN
1146      bb(jsystem, jplayers) = bbmax
1147 !
1148      ELSE
1149 !
1150      ba(jsystem, jplayers) = bamax
1151 !
1152      END IF
1153 !
1154      back = .TRUE.
1155 !
1156      END IF
1157 !
1158      bbmax = (asyn(ksystem)%pricecap - cur_ba(ksystem, iplayers)) /
1159 xmaxs(ksystem, iplayers)
1160 !
1161      IF (coef_flag(jsystem, jplayers) .EQ. 2 .AND. cur_bb(ksystem, iplayers)
1162 .GT. bbmax) THEN
1163 !
1164      IF (bbmax .LT. BAMIN) THEN
1165 !
1166      bb(jsystem, jplayers) = BBMIN
1167      bamax = asyn(ksystem)%pricecap - (BBMIN * xmaxs(ksystem, iplayers))
1168      ba(jsystem, jplayers) = bamax
1169 !
1170      ELSE
1171 !
1172      bb(jsystem, jplayers) = bbmax
1173 !
1174      END IF
1175 !
1176      back = .TRUE.
1177 !
1178      END IF
1179 !
1180      ELSE !the player has the current system as secondary
1181 !
1182      IF (alpha(jsystem, jplayers) .LE. 0.0) THEN ! mono secondary
1183 !
1184      cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
1185      cur_ba(ksystem, iplayers) = ba(jsystem, jplayers)
1186 !
1187      xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
1188      xmaxs(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmax
1189 !
1190      ELSE ! kai sta duo
1191 !
1192      cur_bb(ksystem, iplayers) = bb(jsystem, jplayers)
1193 !
1194      cur_ba(ksystem, iplayers) = ba(jsystem, jplayers) +
1195 &
1196      alpha(jsystem, jplayers) * cur_bb(ksystem, iplayers) *
1197 &
1198      asyn(jsystem)%player(jplayers)%xmax -
1199 &
1200      (1.0d+00 + alpha(jsystem, jplayers)) * cur_bb(ksystem,
1201 iplayers) * &
1202      asyn(jsystem)%player(jplayers)%xmin
1203 !
1204      xmins(ksystem, iplayers) = asyn(jsystem)%player(jplayers)%xmin
1205 !
1206      xmaxs(ksystem, iplayers) = (1.0d+00 - alpha(jsystem, jplayers)) *
1207 &
1208      asyn(jsystem)%player(jplayers)%xmax +
1209 &
1210      asyn(jsystem)%player(jplayers)%xmin *
1211 &
1212      alpha(jsystem, jplayers)
1213      END IF
1214 !

```

```

1215          bamax = asyn(ksystem)%pricecap -
1216 &
1217          (cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers))
1218 !
1219          IF (coef_flag(jsystem, jplayers) .EQ. 1 .AND. cur_ba(ksystem, iplayers)
1220 .GT. bamax) THEN
1221 !
1222          IF (alpha(jsystem, jplayers) .LE. 0.0d+00) THEN ! mono secondary
1223 !
1224          IF (bamax .LT. BAMIN) THEN
1225 !
1226          bbmax = (asyn(ksystem)%pricecap - BAMIN) / xmaxs(ksystem, iplayers)
1227 !
1228          ba(jsystem, jplayers) = BAMIN
1229          bb(jsystem, jplayers) = bbmax
1230 !
1231          ELSE
1232 !
1233          ba(jsystem, jplayers) = bamax
1234 !
1235          END IF
1236
1237          ELSE
1238 !
1239          ba(jsystem, jplayers) = bamax -
1240 &
1241          bb(jsystem, jplayers) * alpha(jsystem, jplayers) *
1242 &
1243          asyn(jsystem)%player(jplayers)%xmax + bb(jsystem,
1244 jplayers) * &
1245          (1.0d+00 + alpha(jsystem, jplayers)) *
1246 &
1247          asyn(jsystem)%player(jplayers)%xmin
1248
1249          IF (ba(jsystem, jplayers) .LT. bamax) ba(jsystem, jplayers) = bamax
1250 !
1251          END IF
1252 !
1253          back = .TRUE.
1254 !
1255          END IF
1256 !
1257          bbmax = (asyn(ksystem)%pricecap - cur_ba(ksystem, iplayers)) /
1258 xmaxs(ksystem, iplayers)
1259
1260          IF (coef_flag(jsystem, jplayers) .EQ. 2 .AND. cur_bb(ksystem, iplayers)
1261 .GT. bbmax) THEN
1262 !
1263          IF (bbmax .LT. BAMIN) THEN
1264 !
1265          bb(jsystem, jplayers) = BBMIN
1266          bamax = asyn(ksystem)%pricecap - (BBMIN * xmaxs(ksystem, iplayers))
1267          ba(jsystem, jplayers) = bamax
1268 !
1269          ELSE
1270 !
1271          bb(jsystem, jplayers) = bbmax
1272 !
1273          END IF
1274          back = .TRUE.
1275 !
1276          END IF
1277 !
1278          END IF
1279 !
1280          END DO
1281          END DO
1282 !
1283          backcount = backcount + 1
1284          IF (back .AND. backcount .LE. 2) GO TO 10
1285 !
1286          Calculation of bidding Parameters' Moving Averages
1287 !
1288          DO ksystem = 1, gameopt%nsystems
1289          DO iplayers = 1, asyn(ksystem)%nplayers
1290 !
1291          aveba(ksystem, iplayers) = aveba(ksystem, iplayers) + (ba(ksystem, iplayers) -
1292 &
1293          aveba(ksystem, iplayers)) / REAL(iiterations + 1)
1294 !
1295          avebb(ksystem, iplayers) = avebb(ksystem, iplayers) + (bb(ksystem, iplayers) -

```

```

1296 &
1297         avebb(ksystem, iplayers)) / REAL(iiterations+1)
1298 !
1299         END DO
1300     END DO
1301 !
1302     DO ksystem = 1, gameopt%nsystems
1303 !
1304         ALLOCATE (xx(nplayers(ksystem)))
1305 !
1306         CALL random_number (y)
1307 !
1308         d = asyn(ksystem)%inid * (1.0d+00 - asyn(ksystem)%dd + 2.0d+00 * asyn(ksystem)%dd *
1309 y)
1310 !
1311 ! Check for available power with respect to the demand
1312         availpower = 0.0d+00
1313 !
1314         DO iplayers = 1, nplayers(ksystem)
1315 !
1316             availpower = availpower + xmaxs(ksystem, iplayers)
1317 !
1318         END DO
1319 !
1320         IF (availpower .LT. d ) THEN
1321 !
1322             kerr(ksystem) = ERR_DEMAND
1323 !
1324             RETURN
1325 !
1326         END IF
1327 !
1328         CALL spot_price(nplayers(ksystem), cur_ba(ksystem, 1:nplayers(ksystem)),
1329 &
1330             cur_bb(ksystem, 1:nplayers(ksystem)), xmin(s(ksystem,
1331 1:nplayers(ksystem))), &
1332             xmaxs(ksystem, 1:nplayers(ksystem)), d, kerr(ksystem), spot(ksystem),
1333 xx)
1334 !
1335         IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
1336 !
1337 !Calculate the isocost (isocost1) from the first cleaning
1338 !
1339         isocost1(ksystem) = 0.0d+00
1340 !
1341         DO iplayers = 1, nplayers(ksystem)
1342 !
1343             IF (asyn(ksystem)%pr_method .EQ. 0) THEN
1344
1345                 revtemp = spot(ksystem) * xx(iplayers)
1346 !
1347             ELSE
1348 !
1349                 revtemp = 0.0d+00
1350                 IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
1351 !
1352                     revtemp = cur_ba(ksystem, iplayers) * xx(iplayers) +
1353 &
1354                         (cur_bb(ksystem, iplayers) / 2.0d+00) * xx(iplayers) *
1355 xx(iplayers) + &
1356                         (cur_bb(ksystem, iplayers) / 2.0d+00) *
1357 &
1358                             xmin(s(ksystem, iplayers)) * xmin(s(ksystem, iplayers))
1359                 END IF
1360 !
1361             END IF
1362 !
1363             isocost1(ksystem) = isocost1(ksystem) + revtemp
1364 !
1365         END DO
1366 !
1367         CALL capacity_constraint (gameopt, asyn, nplayers(ksystem),
1368 &
1369             cur_ba(ksystem, 1:nplayers(ksystem)), cur_bb(ksystem,
1370 1:nplayers(ksystem)), &
1371             xmin(s(ksystem, 1:nplayers(ksystem))), xmaxs(ksystem, 1:nplayers(ksystem)),
1372 &
1373             d, kerr(ksystem), xx, ksystem, conne(ksystem, 1:nplayers(ksystem), :))
1374 !
1375         IF (kerr(ksystem) .NE. ASYN_SUCCESS) RETURN
1376 !

```

```

1377 !      Calculation of the Spot Price's Moving Average [AveSPOT]
1378 !
1379         avespot(ksystem) = avespot(ksystem) +
1380 &
1381             (spot(ksystem) - avespot(ksystem)) / REAL(iiterations + 1)
1382 !
1383         varpr(ksystem, iiterations + 1) = spot(ksystem)
1384 !
1385         sum_sp(ksystem) = sum_sp(ksystem) + spot(ksystem)
1386         sumsp_sq(ksystem) = sumsp_sq(ksystem) + spot(ksystem)**2.0d+00
1387 !
1388 !
1389         purchases(ksystem) = 0.0d+00; isocost(ksystem) = 0.0d+00; averatio(ksystem) =
1390 0.0d+00
1391             counter = 0
1392 !
1393         DO iplayers = 1, nplayers(ksystem)
1394 !
1395             purchases(ksystem) = purchases(ksystem) + xx(iplayers)
1396 !
1397             jplayers = conne(ksystem, iplayers, 1)
1398             jsystem = conne(ksystem, iplayers, 2)
1399 !
1400             x(jsystem, jplayers, ksystem) = xx(iplayers)
1401 !
1402             avex(jsystem, jplayers, ksystem) = avex(jsystem, jplayers, ksystem) +
1403 &
1404                 (x(jsystem, jplayers, ksystem) - avex(jsystem, jplayers, ksystem))
1405 /      &
1406                 REAL(iiterations + 1)
1407 !
1408             IF (asyn(ksystem)%pr_method .EQ. 0) THEN
1409 !
1410                 revtemp = spot(ksystem) * xx(iplayers)
1411 !
1412                 rev(jsystem, jplayers, ksystem) = revtemp
1413 !
1414             ELSE
1415 !
1416                 revtemp = 0.0d+00
1417                 IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
1418 !
1419                     revtemp = cur_ba(ksystem, iplayers) * xx(iplayers) +
1420 &
1421                         (cur_bb(ksystem, iplayers) / 2.0d+00) *
1422 &
1423                             xx(iplayers) * xx(iplayers) +
1424 &
1425                                 (cur_bb(ksystem, iplayers) / 2.0d+00) *
1426 &
1427                                     xmins(ksystem, iplayers) * xmins(ksystem, iplayers)
1428 !
1429                     END IF
1430 !
1431                     rev(jsystem, jplayers, ksystem) = revtemp
1432 !
1433                 END IF
1434 !
1435                 isocost(ksystem) = isocost(ksystem) + revtemp
1436 !
1437                 IF ( ABS(xx(iplayers)) .GT. eps_d ) THEN
1438 !
1439                     ratio(jsystem, jplayers, ksystem) = revtemp / x(jsystem, jplayers,
1440 ksystem)
1441                     averatio(ksystem) = averatio(ksystem) + ratio(jsystem, jplayers, ksystem)
1442                     counter = counter + 1
1443 !
1444                 ELSE
1445 !
1446                     ratio(jsystem, jplayers, ksystem) = 0.0d+00
1447 !
1448                 ENDIF
1449 !
1450             END DO
1451 !
1452             uplift(ksystem) = isocost(ksystem) - isocost1(ksystem)
1453 !
1454             averatio(ksystem) = averatio(ksystem) / REAL(counter)
1455             extra(ksystem) = purchases(ksystem) - d
1456             kwh(ksystem) = isocost(ksystem) / purchases(ksystem)
1457             avekwh(ksystem) = avekwh(ksystem) + (kwh(ksystem) - avekwh(ksystem)) /

```

```

1458 REAL(iiterations + 1)
1459 !
1460         IF (asyn(ksystem)%pr_method .EQ. 0) THEN
1461 !
1462             price(ksystem) = spot(ksystem)
1463             aveprice(ksystem) = avespot(ksystem)
1464 !
1465         ELSE
1466 !
1467             price(ksystem) = kwh(ksystem)
1468             aveprice(ksystem) = avekwh(ksystem)
1469 !
1470         END IF
1471
1472         sum_purchases(ksystem) = sum_purchases(ksystem) + purchases(ksystem)
1473         sum_extra(ksystem) = sum_extra(ksystem) + extra(ksystem)
1474         sum_isocost(ksystem) = sum_isocost(ksystem) + isocost(ksystem)
1475 !
1476         DEALLOCATE (xx)
1477 !
1478     END DO ! ksystem
1479 !
1480 ! Calculate the uplift allocation ratio
1481     DO ksystem = 1, gameopt%nsystems
1482 !
1483         xtot = 0.0d+00
1484         DO iplayers = 1, asyn(ksystem)%nplayers
1485             xtot = xtot + x(ksystem, iplayers, ksystem)
1486         END DO
1487 !
1488         DO iplayers = 1, asyn(ksystem)%nplayers
1489 !
1490             IF (xtot .GT. eps_d) THEN
1491                 uplar(ksystem, iplayers) = x(ksystem, iplayers, ksystem) / xtot
1492             ELSE
1493                 uplar(ksystem, iplayers) = 0.0d+00
1494             END IF
1495 !
1496         END DO
1497 !
1498     END DO
1499 !
1500     DO ksystem = 1, gameopt%nsystems
1501         DO iplayers = 1, asyn(ksystem)%nplayers
1502 !
1503             j(ksystem, iplayers) = 0.0d+00
1504 !
1505             xtot = 0.0d+00
1506             DO jsystem = 1, gameopt%nsystems
1507 !
1508                 j(ksystem, iplayers) = j(ksystem, iplayers) + rev(ksystem, iplayers,
1509 jsystem)
1510                 xtot = xtot + x(ksystem, iplayers, jsystem)
1511 !
1512             END DO
1513
1514             j(ksystem, iplayers) = j(ksystem, iplayers) -
1515 (asyn(ksystem)%player(iplayers)%fc + &
1516                 asyn(ksystem)%player(iplayers)%a * xtot +
1517 &
1518                 asyn(ksystem)%player(iplayers)%b * xtot * xtot ) -
1519 &
1520                 uplift(ksystem) * uplar(ksystem, iplayers)
1521 !
1522             avej(ksystem, iplayers) = avej(ksystem, iplayers) +
1523 &
1524                 (j(ksystem, iplayers) - avej(ksystem, iplayers)) / REAL(iiterations
1525 + 1)
1526 !
1527             dj(ksystem, iplayers) = j(ksystem, iplayers) - jo(ksystem, iplayers)
1528 !
1529             jo(ksystem, iplayers) = j(ksystem, iplayers)
1530 !
1531         END DO
1532     ENDDO
1533 !
1534 ! Propability correction decision block
1535 !
1536     DO ksystem = 1, gameopt%nsystems
1537         DO iplayers = 1, asyn(ksystem)%nplayers
1538

```

```

1539      djt = dj(kssystem, iplayers)
1540      act = action(kssystem, iplayers, coef_flag(kssystem, iplayers))
1541 !
1542      IF (djt .GT. 0.0d+00 .AND. ABS(djt) .GT. eps_d) THEN
1543 !
1544          IF (act .EQ. 1) THEN
1545 !
1546              IF ( pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr .OR.
1547 &
1548                  pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr ) THEN
1549 !
1550                  IF (pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr) THEN
1551 !
1552                      dpi = theta(kssystem, iplayers) *
1553 &
1554                          pc(kssystem, iplayers, coef_flag(kssystem, iplayers))
1555 !
1556                          pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1557 &
1558                          pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1559 !
1560                  ELSE
1561 !
1562                      dpi = theta(kssystem, iplayers) *
1563 &
1564                          pd(kssystem, iplayers,coef_flag(kssystem, iplayers))
1565 !
1566                          pd(kssystem, iplayers,coef_flag(kssystem, iplayers)) =
1567 &
1568                          pd(kssystem, iplayers,coef_flag(kssystem, iplayers)) - dpi
1569 !
1570                  END IF
1571 !
1572                      pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1573 &
1574                      pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi
1575 !
1576                  ELSE
1577 !
1578                      dpi = theta(kssystem, iplayers) * (1.0d+00 -
1579 &
1580                          pi(kssystem, iplayers, coef_flag(kssystem, iplayers))) / 2.0d+00
1581 !
1582                      dp1 = pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1583                      dp2 = pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1584 !
1585                      IF (dp1 .LT. zr .OR. dp2 .LT. zr) THEN
1586 !
1587                          dpi = MIN( pd(kssystem, iplayers, coef_flag(kssystem, iplayers)),
1588 &
1589                              pc(kssystem, iplayers,coef_flag(kssystem, iplayers)) )
1590 !
1591                      END IF
1592 !
1593                          pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1594 &
1595                          pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1596 !
1597                          pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1598 &
1599                          pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1600                          pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1601 &
1602                          pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) + 2.0d+00 * dpi
1603 !
1604                      END IF
1605 !
1606                  ELSE IF (ACT .EQ. 2) THEN
1607 !
1608                      IF (pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr .OR.
1609 &
1610                          pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr )
1611 THEN
1612 !
1613                          IF (pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr) THEN
1614 !
1615                              dpi = theta(kssystem, iplayers) *
1616 &
1617                                  pc(kssystem, iplayers,coef_flag(kssystem, iplayers))
1618 !
1619                                  pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) =

```



```

1620 &
1621         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1622 !
1623     ELSE
1624 !
1625         dpi = theta(kssystem, iplayers) *
1626 &
1627             pi(kssystem, iplayers, coef_flag(kssystem, iplayers))
1628 !
1629         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1630 &
1631         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1632 !
1633     END IF
1634 !
1635         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1636 &
1637         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) + dpi
1638 !
1639     ELSE
1640 !
1641         dpi = theta(kssystem, iplayers) * (1.0d+00 -
1642 &
1643             pd(kssystem, iplayers, coef_flag(kssystem, iplayers))) / 2.0d+00
1644 !
1645         dp1 = pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1646         dp2 = pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1647 !
1648         IF (dp1 .LT. zr .OR. DP2 .LT. ZR) THEN
1649 !
1650             dpi = MIN( pi(kssystem, iplayers, coef_flag(kssystem, iplayers)),
1651 &
1652                 pc(kssystem, iplayers,coef_flag(kssystem, iplayers)) )
1653 !
1654         END IF
1655 !
1656         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1657 &
1658         pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1659 !
1660         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1661 &
1662         pc(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1663 !
1664         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1665 &
1666         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) + 2.0 * dpi
1667 !
1668     END IF
1669 !
1670     ELSE
1671 !
1672         IF (pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr .OR.
1673 &
1674         pd(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr ) THEN
1675 !
1676         IF (pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) .LE. zr) THEN
1677 !
1678             dpi = theta(kssystem, iplayers) *
1679 &
1680                 pd(kssystem, iplayers, coef_flag(kssystem, iplayers))
1681 !
1682             pd(kssystem, iplayers,coef_flag(kssystem, iplayers)) =
1683 &
1684             pd(kssystem, iplayers,coef_flag(kssystem, iplayers)) - dpi
1685 !
1686         ELSE
1687 !
1688             dpi = theta(kssystem, iplayers) *
1689 &
1690                 pi(kssystem, iplayers, coef_flag(kssystem, iplayers))
1691 !
1692             pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) =
1693 &
1694             pi(kssystem, iplayers, coef_flag(kssystem, iplayers)) - dpi
1695 !
1696         END IF
1697 !
1698         pc(kssystem, iplayers,coef_flag(kssystem, iplayers)) =
1699 &
1700         pc(kssystem, iplayers,coef_flag(kssystem, iplayers)) + dpi

```

```

1701 !
1702 ELSE
1703 !
1704     dpi = theta(ksystem, iplayers) * (1.0d+00 -
1705 &
1706         pc(ksystem, iplayers, coef_flag(ksystem, iplayers))) / 2.0d+00
1707 !
1708     dp1 = pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1709     dp2 = pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1710 !
1711     IF (dp1 .LT. zr .OR. dp2 .LT. zr) THEN
1712 !
1713         dpi = MIN( pi(ksystem, iplayers, coef_flag(ksystem, iplayers)),
1714 &
1715             pd(ksystem, iplayers,coef_flag(ksystem, iplayers)) )
1716 !
1717     END IF
1718 !
1719     pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1720 &
1721     pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1722 !
1723     pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1724 &
1725     pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1726 !
1727     pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1728 &
1729     pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) + 2.0d+00 * dpi
1730 !
1731     END IF
1732 !
1733     END IF
1734 !
1735 ELSE
1736 !
1737     IF (ACT .EQ. 1) THEN
1738 !
1739         dpi = theta(ksystem, iplayers) *
1740 &
1741             pi(ksystem, iplayers,coef_flag(ksystem, iplayers))
1742 !
1743         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1744 &
1745         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1746 !
1747         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1748 &
1749         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1750 !
1751         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1752 &
1753         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1754 !
1755     ELSE IF (ACT .EQ. 2) THEN
1756 !
1757         dpi = theta(ksystem, iplayers) *
1758 &
1759             pd(ksystem, iplayers, coef_flag(ksystem, iplayers))
1760 !
1761         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1762 &
1763         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi
1764 !
1765         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1766 &
1767         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1768 !
1769         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1770 &
1771         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1772 !
1773     ELSE
1774 !
1775         dpi = theta(ksystem, iplayers) *
1776 &
1777             pc(ksystem, iplayers, coef_flag(ksystem, iplayers))
1778 !
1779         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1780 &
1781         pc(ksystem, iplayers, coef_flag(ksystem, iplayers)) - dpi

```

```

1782 !
1783         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1784 &
1785         pd(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1786         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) =
1787 &
1788         pi(ksystem, iplayers, coef_flag(ksystem, iplayers)) + dpi / 2.0d+00
1789 !
1790         END IF
1791 !
1792         END IF
1793 !
1794         ENDDO !players
1795 !
1796     ENDDO !ksystem
1797 !
1798     Detailed output of initial state
1799 !
1800     IF (gameopt%ngames .EQ. 1) THEN
1801     DO ksystem = 1, gameopt%nsystems
1802     !
1803         asyn(ksystem)%out_one_game(1, iiterations+1) = REAL(price(ksystem), ikind4)
1804         asyn(ksystem)%out_one_game(2, iiterations+1) = REAL(aveprice(ksystem), ikind4)
1805         asyn(ksystem)%out_one_game(3, iiterations+1) = REAL(spot(ksystem), ikind4)
1806         asyn(ksystem)%out_one_game(4, iiterations+1) = REAL(avespot(ksystem), ikind4)
1807     !
1808         DO iplayers = 1, asyn(ksystem)%nplayers
1809         !
1810             kcol = 4 + iplayers
1811             asyn(ksystem)%out_one_game(kcol, iiterations+1) = x(ksystem, iplayers, 1)
1812         !
1813         END DO
1814     !
1815         DO iplayers = 1, asyn(ksystem)%nplayers
1816         !
1817             kcol = 4 + asyn(ksystem)%nplayers + iplayers
1818             asyn(ksystem)%out_one_game(kcol, iiterations+1) = x(ksystem, iplayers, 2)
1819         !
1820         END DO
1821     END DO
1822 !
1823     ksystem = 1
1824     DO iplayers = 1, nplayers(ksystem)
1825     !
1826         jplayers = conne(ksystem, iplayers, 1)
1827         jsystem = conne(ksystem, iplayers, 2)
1828     !
1829         kcol = 4 + 2*asyn(jsystem)%nplayers + jplayers
1830         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(ksystem, iplayers) +
1831 &
1832         cur_bb(ksystem, iplayers) * xmins(ksystem, iplayers)
1833     !
1834     END DO
1835 !
1836     ksystem = 2
1837     DO iplayers = 1, nplayers(ksystem)
1838     !
1839         jplayers = conne(ksystem, iplayers, 1)
1840         jsystem = conne(ksystem, iplayers, 2)
1841     !
1842         kcol = 4 + 3*asyn(jsystem)%nplayers + jplayers
1843         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(ksystem, iplayers) +
1844 &
1845         cur_bb(ksystem, iplayers) * xmins(ksystem, iplayers)
1846     !
1847     END DO
1848 !
1849     ksystem = 1
1850     DO iplayers = 1, nplayers(ksystem)
1851     !
1852         jplayers = conne(ksystem, iplayers, 1)
1853         jsystem = conne(ksystem, iplayers, 2)
1854     !
1855         kcol = 4 + 4*asyn(jsystem)%nplayers + jplayers
1856         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(ksystem, iplayers) +
1857 &
1858         cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers)
1859     !
1860     END DO
1861 !
1862

```

```

1863 !
1864     ksystem = 2
1865     DO iplayers = 1, nplayers(ksystem)
1866 !
1867         jplayers = conne(ksystem, iplayers, 1)
1868         jsystem = conne(ksystem, iplayers, 2)
1869
1870         kcol = 4 + 5*asyn(jsystem)%nplayers + jplayers
1871         asyn(jsystem)%out_one_game(kcol, iiterations+1) = cur_ba(ksystem, iplayers) +
1872 &
1873         cur_bb(ksystem, iplayers) * xmaxs(ksystem, iplayers)
1874 !
1875     END DO
1876 !
1877     ksystem = 1
1878     DO iplayers = 1, nplayers(ksystem)
1879 !
1880         jplayers = conne(ksystem, iplayers, 1)
1881         jsystem = conne(ksystem, iplayers, 2)
1882
1883         kcol = 4 + 6*asyn(jsystem)%nplayers + jplayers
1884         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmin(ksystem, iplayers)
1885 !
1886     END DO
1887 !
1888     ksystem = 2
1889     DO iplayers = 1, nplayers(ksystem)
1890 !
1891         jplayers = conne(ksystem, iplayers, 1)
1892         jsystem = conne(ksystem, iplayers, 2)
1893
1894         kcol = 4 + 7*asyn(jsystem)%nplayers + jplayers
1895         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmin(ksystem, iplayers)
1896 !
1897     END DO
1898 !
1899     ksystem = 1
1900     DO iplayers = 1, nplayers(ksystem)
1901 !
1902         jplayers = conne(ksystem, iplayers, 1)
1903         jsystem = conne(ksystem, iplayers, 2)
1904
1905         kcol = 4 + 8*asyn(jsystem)%nplayers + jplayers
1906         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmaxs(ksystem, iplayers)
1907 !
1908     END DO
1909 !
1910     ksystem = 2
1911     DO iplayers = 1, nplayers(ksystem)
1912 !
1913         jplayers = conne(ksystem, iplayers, 1)
1914         jsystem = conne(ksystem, iplayers, 2)
1915
1916         kcol = 4 + 9*asyn(jsystem)%nplayers + jplayers
1917         asyn(jsystem)%out_one_game(kcol, iiterations+1) = xmaxs(ksystem, iplayers)
1918 !
1919     END DO
1920 !
1921     END IF
1922 !
1923     DEALLOCATE (cur_ba, cur_bb, xmin, xmax)
1924 !
1925     END DO ! iterations
1926 !
1927     DO ksystem = 1, gameopt%nsystems
1928 !
1929         IF (sumsp_sq(ksystem) .LT. BIG .AND. sum_sp(ksystem) .LT. BIG) THEN
1930             sp_stdevp = SQRT( ABS(REAL(asyn(ksystem)%niterations) * sumsp_sq(ksystem) -
1931 &
1932             sum_sp(ksystem)**2.0d+00) ) /
1933 REAL(asyn(ksystem)%niterations)
1934         ELSE
1935 !
1936             sp_stdevp = BIG
1937 !
1938         ENDIF
1939 !
1940         sumabs_pr = 0.0d+00
1941 !
1942         DO iiterations = 1, asyn(ksystem)%niterations
1943 !

```

```

1944         sumabs_pr = sumabs_pr + ABS(varpr(ksystem, iterations) - aveprice(ksystem))
1945 !
1946     END DO
1947 !
1948     sp_var = sumabs_pr / REAL(asyn(ksystem)%niterations)
1949 !
1950     asyn(ksystem)%out_many_games(1, igames) = REAL(aveprice(ksystem), ikind4)
1951     asyn(ksystem)%out_many_games(2, igames) = REAL(sp_stdevp, ikind4)
1952     asyn(ksystem)%out_many_games(3, igames) = REAL(sp_var, ikind4)
1953 !
1954     DO iplayers = 1, asyn(ksystem)%nplayers
1955         k = iplayers + 3
1956         asyn(ksystem)%out_many_games(k, igames) = REAL(avex(ksystem, iplayers, 1), ikind4)
1957     END DO
1958 !
1959     DO iplayers = 1, asyn(ksystem)%nplayers
1960         k = asyn(ksystem)%nplayers + 3 + iplayers
1961         asyn(ksystem)%out_many_games(k, igames) = REAL(avex(ksystem, iplayers, 2), ikind4)
1962     END DO
1963 !
1964 !
1965     DO iplayers = 1, asyn(ksystem)%nplayers
1966         k = 2 * asyn(ksystem)%nplayers + 3 + iplayers
1967         asyn(ksystem)%out_many_games(k, igames) = REAL(avej(ksystem, iplayers), ikind4)
1968     END DO
1969 !
1970     DO iplayers = 1, asyn(ksystem)%nplayers
1971         k = 3 * asyn(ksystem)%nplayers + 3
1972         k = k + iplayers + (iplayers - 1)
1973 !
1974         asyn(ksystem)%out_many_games(k, igames) = REAL(aveba(ksystem, iplayers), ikind4)
1975         asyn(ksystem)%out_many_games(k+1, igames) = REAL(avebb(ksystem, iplayers), ikind4)
1976 !
1977     END DO
1978 !
1979     k = 5 * asyn(ksystem)%nplayers + 4
1980 !
1981     asyn(ksystem)%out_many_games(k, igames) = sum_purchases(ksystem)
1982     asyn(ksystem)%out_many_games(k+1, igames) = REAL(sum_isocost(ksystem), ikind4)
1983     asyn(ksystem)%out_many_games(k+2, igames) = REAL(sum_extra(ksystem), ikind4)
1984     asyn(ksystem)%out_many_games(k+3, igames) = REAL(avespot(ksystem), ikind4)
1985 !
1986     END DO
1987 !
1988     END DO ! games
1989     DEALLOCATE (varpr)
1990 !
1991     kerr = ASYN_SUCCESS
1992 !
1993 ! End Listing
1994 !.....
1995 !~~~~~
1996     END SUBROUTINE calc_asyn_2sys_ofsplit
1997 !~~~~~
1998 !+ <A one line description of this internal subroutine>
1999 !~~~~~
2000     SUBROUTINE spot_price (nplayers, cur_ba, cur_bb, xmins, xmaxs, d, kerr, spot, xx)
2001 !.....
2002 !.....
2003 !
2004 ! Description:
2005 !   <Say what this subroutine does>
2006 !
2007 ! Code description:
2008 !       Language: Fortran 90.
2009 !.....
2010 ! Declarations:
2011 !
2012 ! Subroutine Arguments
2013 ! Scalar arguments with intent(in):
2014     INTEGER(4), INTENT(IN)                :: nplayers
2015     REAL(8), INTENT(IN)                   :: d
2016 ! Array arguments with intent(in):
2017     REAL(8), DIMENSION(nplayers), INTENT(IN) :: cur_ba, cur_bb, xmins, xmaxs
2018 ! Scalar arguments with intent(inout):
2019     INTEGER(4), INTENT(INOUT)              :: kerr
2020 ! Array arguments with intent(inout):
2021 ! Scalar arguments with intent(out):
2022     REAL(8), INTENT(OUT)                   :: spot
2023 ! Array arguments with intent(out):
2024     REAL(8), DIMENSION(nplayers), INTENT(OUT) :: xx

```

```

2025 ! Local parameters:
2026     REAL(8), PARAMETER                :: EPS_D = 1.0D-20, BIG = 1.0D+30, EPS_S =
2027 1.0E-8
2028 ! Local scalars:
2029     INTEGER(4)                        :: iconst, counter, iplayers, jplayers, up,
2030 low, &
2031                                     excess
2032     REAL(8)                            :: sumab, sumlb, zpos, zneg, difd, l, demx
2033 ! Local arrays:
2034     REAL(8), DIMENSION(nplayers, NCONST) :: lp
2035     REAL(8), DIMENSION(nplayers*NCONST)  :: lpoint, qpoint
2036     REAL(8), DIMENSION(nplayers*NCONST, nplayers) :: qx
2037     REAL(8), DIMENSION(nplayers)         :: expl
2038 !.....
2039 ! Listing
2040 !
2041     lp(:, :) = 0.0
2042 !
2043     DO iplayers = 1, nplayers
2044 !
2045         lp(iplayers, 1) = cur_ba(iplayers) + cur_bb(iplayers) * xmin(iplayers)
2046 !
2047         lp(iplayers, 2) = cur_ba(iplayers) + cur_bb(iplayers) * xmax(iplayers)
2048 !
2049     ENDDO
2050 !
2051     Initial Demand Assignment/Reassignment
2052 !
2053     sumab = 0.0d+00; sumlb = 0.0d+00
2054 !
2055     DO iplayers = 1, nplayers
2056 !
2057         sumab = sumab + (cur_ba(iplayers) / cur_bb(iplayers))
2058         sumlb = sumlb + (1.0d+00 / cur_bb(iplayers))
2059 !
2060     END DO
2061 !
2062     Calculation of the Spot Price
2063 !
2064     counter = 0
2065 !
2066     Definition of the 2xPLAYERS nodal points of the aggregate Supply Curve
2067     and calculation of the available quantity (Qx) for each player at every point
2068 !
2069     DO iconst = 1, NCONST
2070 !
2071         DO iplayers = 1, nplayers
2072 !
2073             counter = counter + 1
2074 !
2075             lpoint(counter) = lp(iplayers, iconst) !assignment of all values
2076 !
2077             qpoint(counter) = 0
2078 !
2079             DO jplayers = 1, nplayers
2080 !
2081                 IF (lpoint(counter) .LE. lp(jplayers, 1)) THEN
2082 !
2083                     qx(counter, jplayers) = 0.0d+00 ! If the point is below player's min
2084 !
2085                 ELSE IF (lpoint(counter) .GE. lp(jplayers, 2)) THEN
2086 !
2087                     qx(counter, jplayers) = xmax(jplayers) ! If the point is above player's max
2088 !
2089                 ELSE
2090 !
2091                     qx(counter, jplayers) = (lpoint(counter) - cur_ba(jplayers)) / & ! The
2092 point between player's min and max
2093                                     cur_bb(jplayers)
2094 !
2095                 END IF
2096 !
2097 ! Also calculation of the total available quantity (QPoint) at each nodal point
2098         qpoint(counter) = qpoint(counter) + qx(counter, jplayers)
2099 !
2100     ENDDO
2101 !
2102     ENDDO
2103 !
2104     ENDDO
2105 !

```

```

2106      zpos = REAL(nplayers) * d + 1.0d+00
2107      zneg = -zpos
2108 !
2109      up = 0
2110      low = 0
2111 !
2112      DO counter = 2 * nplayers, 1, -1
2113 !
2114          difd = qpoint(counter) - d
2115 !
2116          IF ( ABS(difd) .LE. EPS_D ) THEN
2117 !
2118              DO iplayers = 1, nplayers
2119                  expl(iplayers) = iplayers
2120                  xx(iplayers) = qx(counter, iplayers)
2121              END DO
2122 !
2123              l = lpoint(counter)
2124              GOTO 530
2125          ELSEIF (difd .GT. 0) THEN
2126 !
2127              IF (zpos .GT. difd) THEN
2128 !
2129                  zpos = difd
2130                  up = counter
2131 !
2132              ELSE IF (ABS(zpos - difd) .LE. EPS_D) THEN
2133 !
2134                  IF (up .EQ. 0) THEN
2135                      kerr = ERR_SOLVE_UP
2136                      RETURN
2137                  END IF
2138                  IF (lpoint(up) .GT. lpoint(counter)) up = counter
2139 !
2140              END IF
2141 !
2142          ELSE
2143 !
2144              IF (zneg .LT. difd) THEN
2145 !
2146                  zneg = difd
2147                  low = counter
2148 !
2149              ELSE IF (ABS(zneg - difd) .LE. EPS_D) THEN
2150 !
2151                  IF (low .EQ. 0) THEN
2152                      kerr = ERR_SOLVE_LOW
2153                      RETURN
2154                  END IF
2155 !
2156                  IF (lpoint(low) .LT. lpoint(counter)) low = counter
2157 !
2158              END IF
2159 !
2160          END IF
2161 !
2162      END DO
2163 !
2164      demx = d
2165      excess = 0
2166 !
2167      IF (low .EQ. 0) THEN
2168          kerr = ERR_SOLVE_LOW
2169          RETURN
2170      ENDIF
2171      IF (up .EQ. 0) THEN
2172          kerr = ERR_SOLVE_UP
2173          RETURN
2174      END IF
2175 !
2176      DO iplayers = 1, nplayers
2177 !
2178          expl(iplayers) = 0
2179 !
2180          IF ( ABS(qx(low, iplayers) - xmaxs(iplayers)) .LE. EPS_D) THEN
2181 !
2182              expl(iplayers) = iplayers
2183              xx(iplayers) = xmaxs(iplayers)
2184 !
2185              demx = demx - xx(iplayers)
2186 !

```

```

2187         sumab = sumab - (cur_ba(iplayers) / cur_bb(iplayers))
2188         sumlb = sumlb - (1.0d+00 / cur_bb(iplayers))
2189 !
2190         ELSE IF ( ABS(qx(up, iplayers)) .LE. EPS_D ) THEN
2191 !
2192             expl(iplayers) = iplayers
2193             xx(iplayers) = 0.0d+00
2194 !
2195             sumab = sumab - (cur_ba(iplayers) / cur_bb(iplayers))
2196             sumlb = sumlb - (1.0d+00 / cur_bb(iplayers))
2197 !
2198         ELSE IF ( ABS(qx(low, iplayers)) .LE. EPS_D ) THEN
2199 !
2200             IF (low .EQ. 0 ) THEN
2201                 kerr = ERR_SOLVE_LOW
2202                 RETURN
2203             END IF
2204 !
2205             IF ( qpoint(low) + xmins(iplayers) .GT. d ) THEN
2206
2207                 excess = iplayers
2208 !
2209                 EXIT
2210             END IF
2211 !
2212         END IF
2213 !
2214     END DO
2215 !
2216         Calculation of Price's first value [SPOT]
2217         Assign initial value to Price's Moving Average [AveSPOT]
2218     IF (excess .GT. 0) THEN
2219 !
2220         IF (low .EQ. 0 ) THEN
2221             kerr = ERR_SOLVE_LOW
2222             RETURN
2223         END IF
2224 !
2225         l = lpoint(low)
2226 !
2227         DO iplayers = 1, nplayers
2228 !
2229             expl(iplayers) = iplayers
2230             xx(iplayers) = qx(low, iplayers)
2231 !
2232         END DO
2233 !
2234         xx(excess) = xmins(excess)
2235 !
2236     ELSE
2237 !
2238         l = (demx + sumab) / sumlb
2239 !
2240     END IF
2241 !
2242     530 CONTINUE
2243 !
2244     spot = l
2245 !
2246         Assign to Players of the first quantities [X(iiterations)] bought by the ISO
2247         Assign the same values to quantities' moving averages [AveX(iiterations)]
2248         Calculation of players' first income [Jo(iiterations)]
2249         Assign the same values to incomes' moving averages [AveJ(iiterations)]
2250 !
2251         DO iplayers = 1, nplayers
2252 !
2253             IF (expl(iplayers) .NE. iplayers)
2254 &
2255                 xx(iplayers) = (spot - cur_ba(iplayers)) / cur_bb(iplayers)
2256 !
2257         END DO
2258
2259 !
2260 ! End Listing
2261 !.....
2262 !~~~~~
2263     END SUBROUTINE spot_price
2264 !~~~~~
2265 !+ <A one line description of this internal subroutine>
2266 !~~~~~

```



```

2267     SUBROUTINE capacity_constraint (gameopt, asyn, nplayers, cur_ba, cur_bb, xmin,      &
2268                                     xmax, d, kerr, xx, ksystem, conne)
2269
2270 !.....
2271 !
2272 ! Description:
2273 !   <Say what this subroutine does>
2274 !
2275 ! Code description:
2276 !       Language: Fortran 90.
2277 !.....
2278 ! Declarations:
2279 !
2280 ! Subroutine Arguments
2281 ! Scalar arguments with intent(in):
2282     INTEGER(4), INTENT(IN)                :: nplayers, ksystem
2283     REAL(8), INTENT(IN)                   :: d
2284     TYPE (gameopt_t), INTENT(IN)          :: gameopt
2285     TYPE (asyn_t), INTENT(IN)             :: asyn(gameopt%nplayers)
2286 ! Array arguments with intent(in):
2287     INTEGER(4), DIMENSION(nplayers, 2), INTENT(IN) :: conne
2288     REAL(8), DIMENSION(nplayers), INTENT(IN) :: cur_ba, cur_bb, xmin, xmax
2289 ! Scalar arguments with intent(inout):
2290     INTEGER(4), INTENT(INOUT)             :: kerr
2291 ! Array arguments with intent(inout):
2292     REAL(8), DIMENSION(nplayers), INTENT(INOUT) :: xx
2293 ! Scalar arguments with intent(out):
2294 ! Array arguments with intent(out):
2295 ! Local parameters:
2296     REAL(8), PARAMETER                    :: EPS_S = 1.0E-8
2297 ! Local scalars:
2298     INTEGER(4)                             :: iplayers, jplayers, jsystem, nplayersf,
2299 &
2300                                     opsystem, lplayers, playeroff, nplayersl,
2301 &
2302                                     kplayers
2303     REAL(8)                                 :: trans_power, interconne, spot, temp, xxftot,
2304 &
2305                                     maxoffer, offer, availpower
2306 ! Local arrays:
2307     REAL(8), ALLOCATABLE, DIMENSION (:)    :: cur_baf, cur_bbf, xminf, xmaxf, cur_bal,
2308 &
2309                                     cur_bbl, xminsl, xmaxsl
2310     REAL(8), ALLOCATABLE, DIMENSION (:)    :: xxf, xxl
2311     INTEGER(4), ALLOCATABLE, DIMENSION (:) :: connef, connel
2312 !.....
2313 ! Listing
2314 !
2315     IF (ksystem .EQ. 1) opsystem = 2
2316     IF (ksystem .EQ. 2) opsystem = 1
2317     interconne = gameopt%interconne*1000.0d+00
2318 !
2319 ! Check whether the inteconnection capacity has been exceeded
2320 !
2321     trans_power = 0.0d+00
2322 !
2323     DO iplayers = 1, nplayers
2324 !
2325         jsystem = conne(iplayers, 2)
2326 !
2327         IF (jsystem .EQ. opsystem) THEN ! foreign players
2328 !
2329             trans_power = trans_power + xx(iplayers)
2330
2331         END IF
2332 !
2333     END DO
2334 !
2335     IF (trans_power .LE. interconne) RETURN
2336 !
2337     ALLOCATE ( cur_baf(asyn(opsystem)%nplayers), cur_bbf(asyn(opsystem)%nplayers),      &
2338               xminf(asyn(opsystem)%nplayers), xmaxsf(asyn(opsystem)%nplayers),      &
2339               connef(asyn(opsystem)%nplayers) )
2340 !
2341     nplayersf = 0
2342     DO iplayers = 1, nplayers
2343 !
2344         jsystem = conne(iplayers, 2)
2345 !
2346         IF (jsystem .EQ. opsystem .AND. xx(iplayers) .GT. 0.0d+00) THEN ! foreign players
2347

```

```

2348         nplayersf = nplayersf + 1
2349         connef(nplayersf) = iplayers
2350         cur_baf(nplayersf) = cur_ba(iplayers)
2351         cur_bbf(nplayersf) = cur_bb(iplayers)
2352         xminsf(nplayersf) = xmins(iplayers)
2353         xmaxsf(nplayersf) = xmaxs(iplayers)
2354 !
2355         END IF
2356     END DO
2357 !
2358     ALLOCATE ( xxf(nplayersf) )
2359 !
2360     CALL spot_price(nplayersf, cur_baf(1:nplayersf), cur_bbf(1:nplayersf), xminsf(1:nplayersf),
2361 &
2362                 xmaxsf(1:nplayersf), interconne, kerr, spot, xxf)
2363 !
2364     IF (kerr .NE. ASYN_SUCCESS) RETURN
2365
2366     xxftot = 0.0d+00
2367     DO iplayers = 1, nplayersf
2368     xxftot = xxftot + xxf(iplayers)
2369     end do
2370 !
2371     IF ( xxftot - interconne .GT. EPS_S) THEN
2372 !
2373         maxoffer = 0.0d+00
2374         playeroff = 0
2375 !
2376         DO iplayers = 1, nplayersf
2377 !
2378             IF ( ABS( xminsf(iplayers)-xxf(iplayers)) .LE. EPS_S ) THEN
2379                 offer = cur_baf(iplayers) + cur_bbf(iplayers) * xminsf(iplayers)
2380 !
2381                 IF (offer .GT. maxoffer) THEN
2382                     maxoffer = offer
2383                     playeroff = iplayers
2384                 END IF
2385 !
2386             END IF
2387         END DO
2388 !
2389     END IF
2390 !
2391     DEALLOCATE (cur_baf, cur_bbf, xminsf, xmaxsf)
2392 !
2393
2394     ALLOCATE ( cur_bal(nplayers), cur_bbl(nplayers), xmins1(nplayers), xmaxsl(nplayers),
2395 &
2396             connel(nplayers) )
2397 !
2398     nplayers1 = 0
2399     DO iplayers = 1, nplayers
2400 !
2401         jsystem = conne(iplayers, 2)
2402 !
2403         IF (jsystem .EQ. ksystem) THEN ! home players
2404 !
2405             nplayers1 = nplayers1 + 1
2406 !
2407             connel(nplayers1) = iplayers
2408             cur_bal(nplayers1) = cur_ba(iplayers)
2409             cur_bbl(nplayers1) = cur_bb(iplayers)
2410             xmins1(nplayers1) = xmins(iplayers)
2411             xmaxsl(nplayers1) = xmaxs(iplayers)
2412 !
2413         END IF
2414     END DO
2415 !
2416 !
2417 ! add the foreign players
2418
2419     DO iplayers = 1, nplayersf
2420 !
2421         kplayers = connef(iplayers)
2422 !
2423         IF (iplayers .NE. playeroff) THEN ! do not add the playeroff
2424 !
2425             nplayers1 = nplayers1 + 1
2426 !
2427             connel(nplayers1) = kplayers
2428             cur_bal(nplayers1) = cur_ba(kplayers)

```

```

2429         cur_bbl(nplayers1) = cur_bb(kplayers)
2430         xminsl(nplayers1) = xmin(kplayers)
2431         xmaxsl(nplayers1) = xxf(iplayers)
2432         IF ( ABS(xmaxsl(nplayers1)-xminsl(nplayers1)) .LT. EPS_S ) THEN
2433             xmaxsl(nplayers1) = xmaxsl(nplayers1) + EPS_S ! for the spot price algorithm bug
2434 (xmin=xmax)
2435             END IF
2436 !
2437         END IF
2438 !
2439     END DO
2440 !
2441     DEALLOCATE (connef)
2442     DEALLOCATE (xxf)
2443 !
2444     ALLOCATE ( xxl(nplayers1) )
2445 !
2446 !
2447 ! Check for available power with respect to the demand
2448     availpower = 0.0d+00
2449 !
2450     DO iplayers = 1, nplayers1
2451 !
2452         availpower = availpower + xmaxsl(iplayers)
2453
2454     END DO
2455 !
2456     IF (availpower .LT. d) THEN
2457 !
2458         kerr = ERR_DEMAND_CON
2459
2460         RETURN
2461 !
2462     END IF
2463 !
2464     CALL spot_price(nplayers1, cur_bal(1:nplayers1), cur_bbl(1:nplayers1), xminsl(1:nplayers1),
2465 &
2466         xmaxsl(1:nplayers1), d, kerr, spot, xxl)
2467 !
2468     IF (kerr .NE. ASYN_SUCCESS) RETURN
2469 !
2470     xx(:) = 0.0
2471     DO iplayers = 1, nplayers1
2472 !
2473         kplayers = connel(iplayers)
2474         xx(kplayers) = xxl(iplayers)
2475     END DO
2476 !
2477     DEALLOCATE (cur_bal, cur_bbl, xminsl, xmaxsl, connel)
2478     DEALLOCATE (xxl)
2479 !
2480 ! End Listing
2481 !.....
2482 !~~~~~
2483     END SUBROUTINE capacity_constraint
2484 !~~~~~
2485     END MODULE asyn_two_sys_split_m
2486 !=====
2487 !=====

```

```

1  !=====
2  !+ <A one line description of this program>
3  !=====
4  PROGRAM main
5  !.....
6  !
7  ! Description:
8  !   <Say what this program does>
9  !
10 ! Method:
11 !   <Say how it does it: include references to external documentation>
12 !   <If this routine is divided in to sections, be brief here,
13 !       and put Method comments at the start of each section>
14 !
15 ! Input files:
16 !   <Describe these, and say in which routine they are read>
17 !
18 ! Output files:
19 !   <Describe these, and say in which routine they are written>
20 !
21 ! Current code owner : C.Skoulidas
22 !
23 ! History:
24 ! Version      Date      Comment
25 ! -----      ----      -
26 ! <18.1>      <2011>   Original code <C.Skoulidas>
27 !
28 ! Code description:
29 ! Language: Fortran 90.
30 !.....
31 ! Declarations:
32 !
33 ! Modules Used:
34 !   USE asyn_m
35 !
36 ! Imported type definitions
37 ! Imported Parameters:
38 ! Imported scalar variables with intent (in):
39 ! Imported scalar variables with intent (out):
40 ! Imported Array variables with intent(in):
41 ! Imported Array variables with intent(out):
42 ! Imported routines:
43 !
44 !   IMPLICIT NONE
45 !
46 ! include statements
47 ! Local parameters:
48 !   integer(4), parameter          :: UNIT_IN = 10
49 !   integer(4), parameter          :: UNIT_GAMES = 11
50 !   integer(4), parameter          :: UNIT_OUT = 13
51 ! Local scalars:
52 !   integer(4) :: iplayers, ierr, ksystem
53 !   integer(4) :: igrames, ilim, iterations, nlim, kerror
54 !   type (asyn_t) :: asyn(2)
55 !   type (gameopt_t) :: gameopt
56 !   REAL(8) :: acritical
57 ! Local arrays:
58 !.....
59 ! Listing
60 !
61 !   CALL open_files
62 !
63 !
64 !   READ (UNIT_IN, *, iostat=ierr) gameopt%ngames, gameopt%nsystems
65 !   if (ierr.ne.0) goto 100
66 !
67 !   READ (UNIT_IN, *, iostat=ierr) asyn(1)%nplayers, asyn(1)%niterations,      &
68 !                                   asyn(1)%inid, asyn(1)%dd, asyn(1)%pricecap,
69 &
70 !                                   asyn(1)%period_min, asyn(1)%period_max,      &
71 !                                   asyn(1)%estep_decr, asyn(1)%thstep_decr,      &
72 !                                   asyn(1)%p_flag, asyn(1)%pr_method
73 !
74 !   if (ierr.ne.0) goto 100
75 !
76 !   ALLOCATE(asyn(1)%player(asyn(1)%nplayers))
77 !

```

```

78     do iplayers = 1, asyn(1)%nplayers
79         read(UNIT_IN, *, iostat=ierr) asyn(1)%player(iplayers)%fc,
80                                     &
81                                     asyn(1)%player(iplayers)%a,
82                                     &
83                                     asyn(1)%player(iplayers)%b,
84                                     &
85                                     asyn(1)%player(iplayers)%xmin,
86                                     &
87                                     asyn(1)%player(iplayers)%xmax,
88                                     &
89                                     asyn(1)%player(iplayers)%theta_ini,
90                                     &
91                                     asyn(1)%player(iplayers)%e_init(1:2),
92                                     &
93                                     asyn(1)%player(iplayers)%inipi(1:2),
94                                     &
95                                     asyn(1)%player(iplayers)%inipd(1:2),
96                                     &
97                                     asyn(1)%player(iplayers)%inipc(1:2),
98                                     &
99                                     asyn(1)%player(iplayers)%period(1:2),
100                                    &
101                                    asyn(1)%player(iplayers)%nexit,
102                                    &
103                                    asyn(1)%player(iplayers)%sys2_a,
104                                    &
105                                    asyn(1)%player(iplayers)%alpha_ini,
106                                    &
107                                    asyn(1)%player(iplayers)%alpha_tolerance
108
109         if (ierr.ne.0) goto 100
110     end do
111
112     !
113     IF (gameopt%nsystems .EQ. 2) THEN
114     !
115         READ (UNIT_IN, *, iostat=ierr) asyn(2)%nplayers, asyn(2)%niterations,
116     &
117         asyn(2)%inid, asyn(2)%dd, asyn(2)%pricecap,
118     &
119         asyn(2)%period_min, asyn(2)%period_max,
120     &
121         asyn(2)%estep_decr, asyn(2)%thstep_decr,
122     &
123         asyn(2)%p_flag, asyn(2)%pr_method
124
125     !
126     if (ierr.ne.0) goto 100
127     !
128     ALLOCATE(asyn(2)%player(asyn(2)%nplayers))
129     !
130     do iplayers = 1, asyn(2)%nplayers
131         read(UNIT_IN, *, iostat=ierr) asyn(2)%player(iplayers)%fc,
132     &
133         asyn(2)%player(iplayers)%a,
134     &
135         asyn(2)%player(iplayers)%b,
136     &
137         asyn(2)%player(iplayers)%xmin,
138     &
139         asyn(2)%player(iplayers)%xmax,
140     &
141         asyn(2)%player(iplayers)%theta_ini,
142     &
143         asyn(2)%player(iplayers)%e_init(1:2),
144     &
145         asyn(2)%player(iplayers)%inipi(1:2),
146     &
147         asyn(2)%player(iplayers)%inipd(1:2),
148     &
149         asyn(2)%player(iplayers)%inipc(1:2),
150     &
151         asyn(2)%player(iplayers)%period(1:2),
152     &
153         asyn(2)%player(iplayers)%nexit,
154     &
155         asyn(2)%player(iplayers)%sys2_a,
156     &
157         asyn(2)%player(iplayers)%alpha_ini,
158     &
159         asyn(2)%player(iplayers)%alpha_tolerance
160
161         if (ierr.ne.0) goto 100
162     end do
163
164     !
165     END IF
166
167     !
168     ! set the iterations at the second system
169     asyn(2)%niterations = asyn(1)%niterations
170
171     !
172     close(UNIT_IN)
173
174     !
175     IF (gameopt%nsystems .EQ. 1) THEN
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255

```

```

156      ALLOCATE(asyn(1)%out_many_games (4*asyn(1)%nplayers+8, gameopt%ngames) )
157      !
158      IF ( gameopt%ngames .EQ. 1) THEN
159          ALLOCATE( asyn(1)%out_one_game (8*asyn(1)%nplayers+7, asyn(1)%niterations) )
160      END IF
161      !
162      ELSE IF (gameopt%nsystems .EQ. 2) THEN
163      !
164          DO ksystem = 1, gameopt%nsystems
165      !
166              ALLOCATE( asyn(ksystem)%out_many_games(7 + 5*asyn(ksystem)%nplayers,
167 gameopt%ngames) )
168      !
169              IF ( gameopt%ngames .EQ. 1) THEN
170      !
171                  ALLOCATE( asyn(ksystem)%out_one_game (2, asyn(1)%niterations) )
172      !
173              END IF
174      !
175          END DO
176      !
177      END IF
178      !
179      !
180      CALL calc_asyn (gameopt, asyn, kerror)
181      !
182      IF (kerror .EQ. ERR_DEMAND_SYS1 .OR. kerror .EQ. ERR_DEMAND_SYS2) THEN
183          print*, "One of the systems run out of power"
184          pause
185          stop
186      END IF
187      !
188      IF (kerror .EQ. ERR_SOLVE_UP) THEN
189          print*, "Solving error : up"
190          pause
191          stop
192      END IF
193      !
194      IF (kerror .EQ. ERR_SOLVE_LOW) THEN
195          print*, "Solving error : low"
196          pause
197          stop
198      END IF
199      !
200      IF (gameopt%nsystems .EQ. 1) THEN
201          IF (gameopt%ngames .EQ. 1) THEN
202      !
203          !
204              nlim = 8 * asyn(1)%nplayers + 7
205      !nlim = 2 * asyn(1)%nplayers + 2
206      !
207              DO iterations = 1, asyn(1)%niterations
208                  WRITE(UNIT_OUT, '(280(1X, E14.7))', iostat=ierr)                                &
209                      (asyn(1)%out_one_game(ilim, iterations), ilim = 1, nlim)
210                  if (ierr.ne.0) goto 110
211              END DO
212      !
213          END IF
214      !
215          nlim = 4 * asyn(1)%nplayers + 8
216      !
217          DO igames = 1, gameopt%ngames
218      !
219              WRITE(UNIT_GAMES, '(280(1X, E14.7))', iostat=ierr)                                &
220                  (asyn(1)%out_many_games(ilim, igames), ilim = 1, nlim)
221              if (ierr.ne.0) goto 110
222          END DO
223      !
224      ELSE
225      !
226      !
227          IF (gameopt%ngames .EQ. 1) THEN
228      !
229              WRITE(UNIT_OUT, *) "price sys1 - ave price sys1 - price sys2 - aveprice sys2"
230      !
231              nlim = 2
232              DO iterations = 1, asyn(1)%niterations
233                  WRITE(UNIT_OUT, '(I6, 280(1X, E14.7))', iostat=ierr) iterations,                &

```

```

234             (asyn(1)%out_one_game(ilim, iterations), ilim = 1, nlim),      &
235             (asyn(2)%out_one_game(ilim, iterations), ilim = 1, nlim)
236         if (ierr.ne.0) goto 110
237     END DO
238 !
239     END IF
240 !
241
242     DO ksystem = 1, gameopt%nsystems
243 !
244         WRITE(UNIT_GAMES, *) "-----> Current System = " , ksystem
245 !
246         nlim = 7 + 5*asyn(ksystem)%nplayers
247 !
248         DO igames = 1, gameopt%ngames
249 !
250             WRITE(UNIT_GAMES, '(280(1X, E14.7))', iostat=ierr)
251 &
252             (asyn(ksystem)%out_many_games(ilim, igames), ilim = 1,
253 nlim)
254         if (ierr.ne.0) goto 110
255     END DO
256 !
257     END DO
258 !
259     END IF
260 !
261     IF (gameopt%nsystems .EQ. 1) THEN
262 !
263         DEALLOCATE( asyn(1)%out_many_games )
264         IF ( gameopt%ngames .EQ. 1) THEN
265             DEALLOCATE( asyn(1)%out_one_game)
266         END IF
267 !
268     ELSE
269 !
270         DO ksystem = 1, gameopt%nsystems
271 !
272             DEALLOCATE( asyn(ksystem)%out_many_games )
273
274             IF ( gameopt%ngames .EQ. 1) THEN
275                 DEALLOCATE( asyn(ksystem)%out_one_game)
276             END IF
277 !
278         END DO
279 !
280     END IF
281 !
282     DEALLOCATE(asyn(1)%player)
283 !
284     IF (gameopt%nsystems .EQ. 2) THEN
285         DEALLOCATE(asyn(2)%player)
286     END IF
287 !
288     stop
289 !
290 100 write(*,*) "Input error"
291     stop
292 110 write(*,*) "Output error"
293     stop
294 !
295     997 FORMAT(280(1X, E14.7))
296 !
297     CONTAINS
298 !~~~~~
299 !+ <A one line description of this internal subroutine>
300 !~~~~~
301     SUBROUTINE open_files
302 !.....
303 !
304 ! Description:
305 !   <Say what this subroutine does>
306 !
307 ! Code description:
308 !       Language: Fortran 90.
309 !.....
310 ! Declarations:
311 !

```

```

312 ! Subroutine Arguments
313 ! Scalar arguments with intent(in):
314 ! Array arguments with intent(in):
315 ! Scalar arguments with intent(inout):
316 ! Array arguments with intent(inout):
317 ! Scalar arguments with intent(out):
318 ! Array arguments with intent(out):
319 ! Local parameters:
320 ! Local scalars:
321     integer(4) :: io_error
322     character(len=256) :: file_name
323 ! Local arrays:
324 !.....
325 ! Listing
326 !
327     write(*,('Please type the name of input file : '), advance = 'no')
328     read (*,*) file_name
329 !
330     OPEN (UNIT_IN, FILE=file_name, STATUS = "old", IOSTAT = io_error)
331     if (io_error .ne. 0) then
332         write(*,*) "Cannot open file ", trim (file_name)
333         stop
334     endif
335 !
336     write(*,('Please type the name of games : '), advance = 'no')
337     read (*,*) file_name
338 !
339     OPEN (UNIT_GAMES, FILE=file_name, STATUS = "unknown", IOSTAT = io_error)
340     if (io_error .ne. 0) then
341         write(*,*) "Cannot open file ", trim (file_name)
342         stop
343     endif
344 !
345     write(*,('Please type the output file name : '), advance = 'no')
346     read (*,*) file_name
347     OPEN (UNIT_OUT, FILE = file_name, STATUS = "UNKNOWN", IOSTAT = io_error)
348     if (io_error .ne. 0) then
349         write(*,*) "Cannot open file ", trim (file_name)
350         stop
351     endif
352 !
353 ! End Listing
354 !.....
355     END SUBROUTINE open_files
356 !~~~~~
357 !+ <A one line description of this internal subroutine>
358 !~~~~~
359     SUBROUTINE Error_in (errId, warnId)
360 !.....
361 !
362 ! Description:
363 !   <Say what this subroutine does>
364 !
365 ! Code description:
366 !       Language: Fortran 90.
367 !.....
368 ! Declarations:
369 !
370 ! Subroutine Arguments
371 ! Scalar arguments with intent(in):
372     INTEGER(4), INTENT(IN)                :: errId, warnId
373 ! Array arguments with intent(in):
374 ! Scalar arguments with intent(inout):
375 ! Array arguments with intent(inout):
376 ! Scalar arguments with intent(out):
377 ! Array arguments with intent(out):
378 ! Local parameters:
379     INTEGER(4), PARAMETER                :: merror = 5, mwarning = 3
380 ! Local scalars:
381     INTEGER(4)                            :: iplayers
382     CHARACTER(LEN=65)                     :: message
383 ! Local arrays:
384     CHARACTER(LEN=50), DIMENSION (merror) :: error_message
385     CHARACTER(LEN=50), DIMENSION (mwarning) :: warning_message
386     DATA ( error_message(iplayers), iplayers= 1, merror)
387 &
388     / 'Cannot Open Input File',           &
389     'Cannot Open Games File',           &

```



```

390      'Cannot Open Periods File',      &
391      'Cannot Open Output File',      &
392      'error 5'      /
393  DATA  ( warning_message(iplayers), iplayers= 1, mwarning)
394  &
395      / 'warning 1',      &
396      'warning 2',      &
397      'warning 3'      /
398  !.....
399  ! Listing
400  !
401      IF (errId.GT.0) THEN
402          WRITE (message, FMT = 10) error_message (errId)
403      ENDIF
404      IF (wanrId.GT.0) THEN
405          WRITE (message,FMT = 20) warning_message (wanrId)
406      ENDIF
407  !
408      WRITE(*,'(A65)') message
409  !
410      IF (errId.GT.0) THEN
411          WRITE(*,*) ' press [ENTER] to terminate...';READ(*,*) ;STOP
412      ENDIF
413  !
414  10  FORMAT (2x,'Error : ',A55)
415  20  FORMAT (2x,'Warning : ',A55)
416  !
417  ! End Listing
418  !.....
419      END SUBROUTINE Error_in
420  ! End Listing
421  !.....
422      END PROGRAM main
423  !=====
424  !=====

```