



## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ:

**Πειραματική μελέτη παραγόντων στην επίλυση προβλημάτων  
με τη μέθοδο των εξελικτικών αλγορίθμων με τη χρήση του  
λογισμικού MATLAB®**

**Λιάκος Σ. Νικόλαος**

**Φοιτητής 12ου εξαμήνου**

**Επιβλέπων:** Τόλης Αθανάσιος, Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2015



Αφιερώνεται  
στους παππούδες μου.



## Ευχαριστίες

Αντί προλόγου, θεωρώ υποχρέωσή μου να ευχαριστήσω όλους όσους συνέβαλαν στην ολοκλήρωση της διπλωματικής εργασίας αυτής. Πιο συγκεκριμένα, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή κ. Αθανάσιο Τόλη για την πολύτιμη βοήθεια που μου προσέφερε αλλά και για την συνολική καθοδήγησή του κατά την διάρκεια των σπουδών μου και ειδικότερα κατά την εκπόνησης της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Τσαλαβούτη Βασίλειο, υποψήφιο διδάκτορα και φίλο, για την συνεργασία μας (προϊόν της οποίας αποτελεί το κείμενο αυτό) αλλά και τις χρήσιμες παρατηρήσεις και συμβουλές του κατά την εκπόνησης της διπλωματικής μου εργασίας.

Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές μου αλλά και τους συμφοιτητές μου, καθώς ο καθένας από αυτούς συνέβαλε με τον δικό του τρόπο ώστε από αυτά τα χρόνια που διατέλεσα φοιτητής του τμήματος των Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου να μου μείνουν οι καλύτερες αναμνήσεις.

Τέλος, δεν θα μπορούσα παρά να απευθύνω ένα μεγάλο ευχαριστώ στην οικογένειά μου για την ηθική και υλική υποστήριξη την οποία μου προσέφερε απλόχερα καθ' όλη την διάρκεια της φοίτησής μου.



## Περιεχόμενα

Έποψη Διπλωματικής Εργασίας .....	11
<b>ΚΕΦΑΛΑΙΟ 1 .....</b>	<b>13</b>
<b>ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΕΞΕΛΙΚΤΙΚΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ.....</b>	<b>13</b>
1.1. Εισαγωγή – Ορισμός των Εξελικτικών Αλγορίθμων .....	15
1.2. Πρώτη εμφάνιση και εξέλιξη Εξελικτικών Αλγορίθμων .....	16
1.3. Πρακτικές Εφαρμογές των Εξελικτικών Αλγορίθμων .....	19
<b>ΚΕΦΑΛΑΙΟ 2 .....</b>	<b>21</b>
<b>ΟΙ ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ .....</b>	<b>21</b>
2.1. Περιγραφή Κλασσικού Εξελικτικού Αλγορίθμου .....	23
2.2. Μαθηματική έκφραση προβλήματος βελτιστοποίησης.....	25
2.3. Το πρόβλημα των τοπικών ακροτάτων.....	26
2.4. Μέθοδοι αντιμετώπισης περιορισμών .....	28
2.4.1. Μέθοδοι διατήρησης της δυνατότητας επίτευξης εφικτής λύσης.....	28
2.4.2. Μέθοδοι βασισμένοι σε συντελεστές ποινής .....	29
2.4.3. Μέθοδοι βασισμένοι στην αναζήτηση για βέλτιστη λύση.....	32
2.4.4. Υβριδικές μέθοδοι.....	33
<b>ΚΕΦΑΛΑΙΟ 3 .....</b>	<b>35</b>
<b>ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΜΕΛΕΤΗΘΕΝΤΩΝ ΑΛΓΟΡΙΘΜΩΝ</b>	<b>35</b>
3.1. Σύνοψη κεφαλαίου.....	37
3.2. Η μέθοδος Global Competitive Ranking .....	37

3.2.1.	Μαθηματικό μοντέλο ελαχιστοποίησης.....	37
3.2.2.	Μαθηματικό μοντέλο Global Competitive Ranking.....	38
3.2.3.	Βελτιστοποίηση με Εξελικτικό Αλγόριθμο .....	40
3.2.4.	Μοντελοποίηση του αλγορίθμου σε λογισμικό MATLAB ® .....	42
3.2.5.	Σκοπός μελέτης της μεθόδου Global Competitive Ranking.....	46
3.3.	Η μέθοδος Self-Learning Particle Swarm Optimizer for Global Optimization Problems .....	46
3.3.1.	Μαθηματικό μοντέλο ελαχιστοποίησης.....	46
3.3.2.	Μαθηματικό μοντέλο Particle Swarm Optimization .....	47
3.3.3.	Μαθηματικό μοντέλο Self-Learning Particle Swarm Optimizer .....	48
3.3.4.	Υπολογισμός ποιής.....	56
3.3.5.	Πιθανοθεωρητική αντιμετώπιση.....	58
3.3.6.	Μοντελοποίηση του αλγορίθμου σε λογισμικό MATLAB ® .....	59
3.3.7.	Σκοπός μελέτης της μεθόδου Self-Learning Particle Swarm Optimizer	63
<b>ΚΕΦΑΛΑΙΟ 4 .....</b>		<b>65</b>
<b>ΤΑ ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΕΛΕΓΧΘΗΚΑΝ .....</b>		<b>65</b>
4.1.	Πρόβλημα 1 <sup>ο</sup> .....	67
4.2.	Πρόβλημα 2 <sup>ο</sup> .....	71
4.3.	Πρόβλημα 3 <sup>ο</sup> .....	73
4.4.	Πρόβλημα 4 <sup>ο</sup> .....	74
4.5.	Πρόβλημα 5 <sup>ο</sup> .....	74
<b>ΚΕΦΑΛΑΙΟ 5 .....</b>		<b>75</b>
<b>ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ .....</b>		<b>75</b>



5.1.	Πειραματική διαδικασία αλγορίθμου Global Competitive Ranking .....	77
5.2.	Παρουσίαση αποτελεσμάτων αλγορίθμου Global Competitive Ranking .....	79
5.2.1.	Δοκιμές τιμής πολλαπλασιαστή βάσης .....	79
5.2.2.	Δοκιμές τιμής αριθμού επαναλήψεων.....	80
5.2.3.	Δοκιμές τιμής αριθμού σωματιδίων μ.....	81
5.2.4.	Δοκιμές τιμής πιθανότητας $Pf$ .....	84
5.2.5.	Βέλτιστη επίλυση του προβλήματος.....	85
5.2.6.	Επαλήθευση τιμών παραγόντων .....	87
5.2.7.	Ανάλυση αποτελεσμάτων .....	92
5.3.	Πειραματική διαδικασία αλγορίθμου Self-Learning Particle Swarm Optimizer .	94
5.4.	Παρουσίαση αποτελεσμάτων του αλγορίθμου Self-Learning Particle Swarm Optimizer .....	96
5.4.1.	Αποτελέσματα 2 <sup>ο</sup> προβλήματος.....	96
5.4.2.	Αποτελέσματα 3 <sup>ο</sup> προβλήματος.....	99
5.4.3.	Αποτελέσματα 4 <sup>ο</sup> προβλήματος.....	101
5.4.4.	Αποτελέσματα 5 <sup>ο</sup> προβλήματος.....	104
5.4.5.	Ανάλυση αποτελεσμάτων .....	107
<b>ΚΕΦΑΛΑΙΟ 6 .....</b>		<b>109</b>
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>		<b>109</b>
6.1.	Συμπεράσματα - Επίλογος διπλωματικής εργασίας .....	111
	Βιβλιογραφία .....	115



## Έποψη Διπλωματικής Εργασίας

Οι εξελικτικοί αλγόριθμοι αποτελούν μια σύγχρονη οικογένεια μεθόδων βελτιστοποίησης που βρίσκουν εφαρμογή σε πολλούς διαφορετικούς τομείς όπως η επιχειρησιακή έρευνα, η σχεδίαση αεροσκαφών, η ρομποτική κ.α. Η μεγάλη ποικιλομορφία των μεθόδων αυτών καθώς και η δυνατότητά τους να χρησιμοποιηθούν για την επίλυση πολύ μεγάλων μη γραμμικών προβλημάτων τις καθιστούν ως ένα πολύ ενδιαφέρον αντικείμενο προς μελέτη.

Στα πλαίσια της παρούσας διπλωματικής εργασίας μελετήθηκε η επιρροή διαφόρων παραγόντων στην απόδοση 2 διαφορετικών εξελικτικών αλγορίθμων, οι οποίοι έχουν μοντελοποιηθεί στο λογισμικό MATLAB®. Πιο συγκεκριμένα, μελετάται η απόδοση και πραγματοποιείται σύγκριση 2 διαφορετικών μεθόδων αλγορίθμων σε 4 μη-γραμμικά προβλήματα καθώς και σε ένα πολύ μεγάλο μη γραμμικό πρόβλημα πρόβλεψης ενεργειακού μίγματος (με την μία από τις 2 μεθόδους). Στόχος είναι να διαπιστωθεί ο τρόπος με τον οποίο διάφοροι παράγοντες μπορούν να επηρεάσουν το αποτέλεσμα του προβλήματος χωρίς να επιβαρύνουν χρονικά το πρόβλημα, αλλά και κατά πόσο η απόδοση της αντικειμενικής συνάρτησης επηρεάζεται από τον χρονικό παράγοντα.

Για τη μελέτη των συγκεκριμένων ζητημάτων, πραγματοποιήθηκαν μια σειρά από υπολογιστικά πειράματα με την χρήση H/Y και συγκεκριμένα του λογισμικού MATLAB®. Τα 4+1 προβλήματα αυτά μοντελοποιήθηκαν και μελετήθηκε η απόδοσή τους για πολλές διαφορετικές τιμές των μελετηθέντων παραγόντων, προκειμένου να προκύψει η βέλτιστη δυνατή τιμή για κάθε πρόβλημα, η οποία θα έχει ταυτόχρονα και το ελάχιστο δυνατό σφάλμα.

Συνοπτικά, σύμφωνα με την καθοδήγηση του επιβλέποντα καθηγητή, κατά την εκπόνηση της διπλωματικής εργασίας,

- Μελετάται η επιρροή που έχουν 7 διαφορετικοί παράγοντες στην βελτιστοποίηση προβλημάτων με εξελικτικούς αλγορίθμους:
  - αριθμός επαναλήψεων, πληθυσμός σωματιδίων, πολλαπλασιαστής βάσης και πιθανότητα η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση για τον αλγόριθμο Global Competitive Ranking (GCR).

- Αρχικό  $\lambda$ , αρχική τυπική απόκλιση και ρυθμός μείωσης της τυπικής απόκλισης για τον αλγόριθμο Self-Learning Particle Swarm Optimizer (SLPSO).
- Πραγματοποιείται εύρεση των βέλτιστων παραγόντων, οι οποίοι επηρεάζουν την επίλυση του αλγορίθμου GCR κατά την μελέτη του μεγάλου μη-γραμμικού προβλήματος. Στην συνέχεια, οι βέλτιστοι παράγοντες, οι οποίοι προέκυψαν, εφαρμόζονται στα 4 μικρότερα προβλήματα, τα οποία χρησιμοποιούνται συχνά στην βιβλιογραφία, προκειμένου να επαληθευτούν οι τιμές αυτές. Αντίστοιχα, στην μέθοδο SLPSO, πραγματοποιούνται δοκιμές για 3 παράγοντες στα 4 μικρά μη-γραμμικά προβλήματα, προκειμένου να μπορέσουμε να εξάγουμε χρήσιμα συμπεράσματα για τον αλγόριθμο αλλά και τις τιμές που συχνά παίρνουν οι παράγοντες.
- Παρατίθενται αναλυτικά τα αποτελέσματα, τα οποία προέκυψαν από τις δοκιμές, καθώς και αξιολόγησή τους. Συγκεκριμένα, από τις δοκιμές στον κώδικα GCR βλέπουμε την επιρροή των διαφόρων παραγόντων στο αποτέλεσμα. Από τον κώδικα SLPSO προκύπτουν πρωτίστως τα εύρη τιμών για τα οποία ο κώδικας έχει τις περισσότερες πιθανότητες να βρει βέλτιστη λύση.

Ο σκοπός της διπλωματικής εργασίας αυτής είναι να καλύψει το επιστημονικό κενό, το οποίο υπάρχει στον τομέα των τιμών των παραγόντων που χρησιμοποιούνται σε εξελικτικούς αλγορίθμους. Παρόλο που υπάρχει μεγάλο πλήθος βιβλιογραφίας που αναφέρεται στις μεθόδους αυτές, σπάνια γίνεται αναφορά στην μοντελοποίησή τους σε H/Y και τις τιμές των διαφόρων παραγόντων.

Στο 1<sup>ο</sup> κεφάλαιο γίνεται μια εισαγωγή στους εξελικτικούς αλγορίθμους. Το 2<sup>ο</sup> κεφάλαιο αναλώνεται στην περιγραφή των δημοφιλέστερων μεθόδων εξελικτικών αλγορίθμων καθώς και στις διάφορες μεθόδους αντιμετώπισης προβλημάτων, τα οποία υπόκεινται σε περιορισμούς. Στο 3<sup>ο</sup> κεφάλαιο παρουσιάζονται αναλυτικά οι αλγόριθμοι, οι οποίοι μελετήθηκαν ενώ στο 4<sup>ο</sup> κεφάλαιο παρατίθενται τα προβλήματα, τα οποία μελετήθηκαν. Στο 5<sup>ο</sup> κεφάλαιο παρουσιάζονται η πειραματική διαδικασία, τα αποτελέσματα αλλά και παρατίθεται η ανάλυσή των αποτελεσμάτων αυτών. Τέλος, στο 6<sup>ο</sup> κεφάλαιο περιγράφονται τα γενικά συμπεράσματα, τα οποία προέκυψαν.

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΕΞΕΛΙΚΤΙΚΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ



## **1.1. Εισαγωγή – Ορισμός των Εξελικτικών Αλγορίθμων**

Στην βιολογία, ως εξέλιξη, ορίζεται η αλλαγή στις ιδιότητες ενός πληθυσμού οργανισμών στο πέρασμα του χρόνου, μεταξύ διαφορετικών γενεών. Αν και οι αλλαγές αυτές είναι πολύ μικρές από γενιά σε γενιά, αθροιστικά μπορούν να οδηγήσουν σε σημαντικές διαφοροποιήσεις στις ιδιότητες του οργανισμού αυτού. Ένας βασικός μηχανισμός της εξέλιξης είναι η φυσική επιλογή, σύμφωνα με την οποία, οι οργανισμοί που είναι καλύτερα προσαρμοσμένοι στο περιβάλλον αφήνουν περισσότερους απογόνους από εκείνους που είναι λιγότερο προσαρμοσμένοι.

Ένας εξελικτικός αλγόριθμος, λοιπόν, όπως φανερώνει και η ονομασία του, αποτελεί μια αλληλουχία υπολογισμών, η οποία βασίζεται στην θεωρία της εξέλιξης και της φυσικής επιλογής. Πιο συγκεκριμένα, η βασική ιδέα πάνω στην οποία βασίζεται αυτή η οικογένεια αλγορίθμων είναι η ακόλουθη: δεδομένου ενός πλήθους σωματιδίων, η πίεση του περιβάλλοντος, η οποία ορίζεται στην περίπτωσή μας από την αντικειμενική συνάρτηση, τους περιορισμούς και τα πεδία ορισμού των μεταβλητών, προκαλεί φυσική επιλογή ανάμεσα στα σωματίδια (επικράτηση του ισχυρού), προκαλώντας βελτίωση στην τιμή της αντικειμενικής συνάρτησης (ελαχιστοποίησή της για πρόβλημα ελαχιστοποίησης και μεγιστοποίησή της για πρόβλημα μεγιστοποίησης αντίστοιχα). Γενικότερα, οι εξελικτικοί αλγόριθμοι είναι αλγόριθμοι ανίχνευσης-αναζήτησης καθότι συνδυάζουν την επιβίωση του ισχυρότερου σωματιδίου με μια οργανωμένη ανταλλαγή πληροφοριών, η οποία οδηγεί τα σωματίδια στην περιοχή όπου βρίσκεται η βέλτιστη δυνατή τιμή της αντικειμενικής συνάρτησης.

Κατέχοντας μετα-ευρεστικά και στοχαστικά χαρακτηριστικά, οι εξελικτικοί αλγόριθμοι, δείχνουν να είναι πολύ αποτελεσματικά εργαλεία επίλυσης ακόμα και για δύσκολα προβλήματα βελτιστοποίησης και για το λόγο αυτό η χρήση τους βρίσκει μεγάλο φάσμα εφαρμογών όπως η επιχειρησιακή έρευνα, η ρομποτική, η εκμάθηση μηχανών, η αναγνώριση μοτίβων, η πρόβλεψη της αγοράς κ.α. Τα κύρια πλεονεκτήματά τους είναι η προσαρμοστικότητά τους σε σχεδόν κάθε νέο πρόβλημα καθώς και η δυνατότητά τους να αποφύγουν τον εγκλωβισμό σε τοπικά ακρότατα, υπό ορισμένες προϋποθέσεις. Από την άλλη, ως στοχαστικές μέθοδοι, απαιτούν συχνά μεγάλο αριθμό επαναλήψεων, γεγονός που

γίνεται εμφανές κυρίως σε πολύ μεγάλα προβλήματα. Αποδεδειγμένα, πάντως, οι εξελικτικοί αλγόριθμοι (EA) αποτελούν μια πολύ ισχυρή επαναληπτική υπολογιστική τεχνική ενώ η μοντελοποίησή τους γίνεται, κατά κύριο λόγο, μέσω κώδικα σε σύγχρονες γλώσσες προγραμματισμού ή σύγχρονα λογισμικά που προσομοιώνουν προγραμματιστικό περιβάλλον (όπως η MATLAB®).

## **1.2. Πρώτη εμφάνιση και εξέλιξη Εξελικτικών Αλγορίθμων**

Η ιδέα της προσομοίωσης της διαδικασίας της εξέλιξης και της φυσικής επιλογής για την βελτιστοποίηση υπολογιστικών προβλημάτων φαίνεται (Back et al., 1997) ότι προσεγγίστηκε για πρώτη φορά στα τέλη της δεκαετίας του 1950 μέσα από την δουλειά των Bremermann (Bremermann, 1962), Friedberg (Friedberg, 1958), (Friedberg et al., 1959) και Box (Box, 1957). Η έλλειψη όμως της απαραίτητης υπολογιστικής ισχύος, σε συνδυασμό με κάποια μειονεκτήματα των μεθόδων αυτών, τις κατέστησαν μη εφαρμόσιμες για την περίοδο εκείνη.

**Πρώτη γενιά:** Η πλειοψηφία των διαφορετικών μεθόδων EA που χρησιμοποιούνται σήμερα, βασίζονται σε 3 συσχετιζόμενες αλλά διακριτές προσεγγίσεις, οι οποίες αναπτύχθηκαν κατά τις επόμενες δεκαετίες:

- Το 1962 ο Holland (Holland, 1962) εισήγαγε πρώτος την έννοια των Γενετικών Αλγορίθμων (Genetic Algorithm - GA). Παρά το μεγάλο εύρος εφαρμογής τους, χρησιμοποιούνται κυρίως στον τομέα της βελτιστοποίησης.
- Την ίδια χρονιά, ο Fogel (Fogel, 1962) εισήγαγε για πρώτη φορά την έννοια του Εξελικτικού Προγραμματισμού (Evolutionary Programming - EP), ως μια προσπάθεια να δημιουργήσει τεχνητή νοημοσύνη, επιχειρώντας να εξελίξει μηχανές ώστε να μπορούν να προβλέψουν γεγονότα με βάση προηγούμενες παρατηρήσεις.
- Πέρασαν περισσότερο από 10 χρόνια, έως ότου αναπτυχθεί (Rechenberg, 1973), (Schwefel, 1975) η μέθοδος των Εξελικτικών Στρατηγικών (Evolutionary Strategies - ES). Οι στρατηγικές αυτές είχαν σχεδιαστεί με αρχικό σκοπό την επίλυση δύσκολων πειραματικών παραμέτρων προβλημάτων βελτιστοποίησης.



Παρά την αρχική δυσκολία εφαρμογής έως τότε, κατά τη διάρκεια της δεκαετίας του 1980, οι μεγάλες βελτιώσεις στον τομέα της πληροφορικής κατέστησαν για πρώτη φορά δυνατή την επίλυση πολύπλοκων πραγματικών προβλημάτων, τραβώντας την προσοχή.

**Δεύτερη γενιά:** Η αποτελεσματικότητα που οι ΕΑ έδειξαν στην αντιμετώπιση των προβλημάτων αυτών, κέρδισαν το ενδιαφέρον πολύ περισσότερων ερευνητών, οι οποίοι αποφάσισαν να στραφούν προς τον τομέα αυτό. Μέσα από αυτή την τάση, προέκυψαν αρκετές αξιόλογες μέθοδοι, με βασικότερες τις ακόλουθες:

- Ο Glover ανέπτυξε το Tabu search (Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, 1986), μια μετα-ευρεστική μέθοδο, η οποία αξιοποιεί την γειτονική αναζήτηση προκειμένου να επιτύχει βελτιστοποίηση. Με βάση τη μέθοδο αυτή, εάν μια πιθανή λύση έχει μελετηθεί σε ένα προηγούμενο χρονικό διάστημα ή εάν έχει παραβιάσει κάποιον περιορισμό, σημειώνεται ως “tabu” και έτσι ο αλγόριθμος δεν την εξετάζει εκ νέου.
- Ο Koza (Koza, 1990) θεωρείται πρωτοπόρος, καθώς βάσισε το έργο του στην χρήση του γενετικού προγραμματισμού (Genetic Programming - GP) για τη βελτιστοποίηση πολύπλοκων προβλημάτων επιτρέποντας στον κώδικα να μεταβληθεί, προκειμένου να βρει πολύ πιο πολύπλοκες λύσεις.
- Το 1991 αναπτύχθηκε μια μέθοδος (Kelly & Davis, 1991), σύμφωνα με την οποία, η χρήση των γενετικών αλγόριθμων υποκαθιστά την χρήση τεχνητής νοημοσύνης, προκειμένου αυτοί να βελτιώσουν την απόδοση του κλασσικού αλγορίθμου.
- Το 1993 προτάθηκε (Michalewicz Z. , A hierarchy of evolution programs: An experimental study, 1993) η χρήση ορθών αλλά πιθανά πολύπλοκων δομών δεδομένων. Η χρήση των δομών αυτών, σε συνδυασμό με ένα διευρυμένο σύνολο γενετικών τελεστών, έρχεται σε αντίθεση με την κλασσική προσέγγιση των δυαδικά κωδικοποιημένων δεδομένων.

**Τρίτη γενιά:** Στην εποχή μας, οι πολλές διαφορετικές προσεγγίσεις των ΕΑ αλλά και οι εφαρμογές τους που αναπτύχθηκαν στις προηγούμενες δεκαετίες, είχαν ως συνέπεια την πολύ μεγάλη αύξηση στην δημοτικότητα των ΕΑ, καθώς και τη δημιουργία πολλών νέων

διαφορετικών μεθόδων, αφού όλο και περισσότεροι ερευνητές ασχολούνται με αυτούς. Οι σημαντικότερες εκδηλώσεις της προόδου των μεθόδων αυτών είναι οι ακόλουθες:

- Ο Moscato ανέφερε (Moscato, 1989) για πρώτη φορά την έννοια του μιμητικού αλγορίθμου (Memetic Algorithm – MA), ο οποίος είναι ένας αλγόριθμος κοντινός με έναν πληθυσμο-κεντρικό υβριδικό GA συνδυασμένο με μια μέθοδο εκμάθησης ατόμου ικανού να πραγματοποιήσει τοπικές διορθώσεις.
- Το 1991, ο Dorigo πρότεινε τον αλγόριθμο βελτιστοποίησης με αποικίες μυρμηγκιών (Ant Colony Optimization Algorithms – ACO) (Dorigo et al., 1991). Αρχικά, ο αλγόριθμος επιχείρησε να βρει ένα βέλτιστο μονοπάτι, βασισμένο στη συμπεριφορά των μυρμηγκιών, τα οποία αναζητούν ένα μονοπάτι ανάμεσα στην αποικία και την τροφή τους.
- Ο Reynolds εισήγαγε για πρώτη φορά ένα στοιχείο το οποίο ονομάζεται διάστημα πίστης (Belief Space) (Reynolds et al., 1994) και το οποίο συμπληρώνει τον πληθυσμό. Το σύνολο των αλγορίθμων που βασίζονται σε αυτό το διάστημα πίστης ονομάζονται αλγόριθμοι κουλτούρας (Cultural Algorithms – CA).
- Το 1994 αναπτύχθηκε στο Πανεπιστήμιο της Νότιας Καλιφόρνια (Adleman, 1994) η μέθοδος υπολογισμού DNA (DNA computing). Βασισμένο στην λογική ότι το DNA κωδικοποιεί τεράστιο αριθμό δεδομένων χρησιμοποιώντας μόνο 4 παράγοντες-κλειδί, κατασκεύασε έναν υπολογιστή, ο οποίος είχε συνδεθεί με ένα σωλήνα γεμάτο με υγρή ποσότητα DNA, και κατάφερε να επιλύσει απλά προβλήματα.
- Ο Kennedy και ο Eberhart είναι οι πρώτοι (Kennedy & Eberhart, Particle Swarm Optimization, 1995), (Kennedy, 1997), οι οποίοι ανέπτυξαν τη μέθοδο της βελτιστοποίησης με τη χρήση σμήνους σωματιδίων (Particle Swarm Optimization – PSO). Ο αρχικός σκοπός της μεθόδου ήταν η ακριβής απεικόνιση της κίνησης οργανισμών όπως τα σμήνη πτηνών ή τα κοπάδια των ψαριών. Ο αλγόριθμος αυτός απλοποιήθηκε και παρατηρήθηκε ότι βελτιστοποιούσε το πρόβλημα. Θεωρείται μετα-ευρεστικός, καθότι δεν κάνει υποθέσεις για το μελετούμενο πρόβλημα και μπορεί να ερευνήσει πολύ μεγάλους χώρους για την εύρεση βέλτιστης λύσης, χωρίς, όμως, να εγγυάται ότι η βέλτιστη λύση θα βρεθεί. Παρόλα αυτά, η φύση του

τον καθιστά ιδανικό για μερικώς ανώμαλα προβλήματα, θορυβώδη προβλήματα ή προβλήματα, τα οποία αλλάζουν στην πάροδο του χρόνου.

- Η Forrest (Forrest et al., 1994) ήταν η πρώτη που ανέπτυξε την μέθοδο των Artificial Immune Systems (AIS). Η μέθοδος αυτή προσπαθεί να μιμηθεί τον μηχανισμό με βάση τον οποίο δουλεύει το ανθρώπινο ανοσοποιητικό σύστημα, προκειμένου να επιλύσει μαθηματικά προβλήματα.
- Το 2002, οι Larrañaga και Lozano, ανέπτυξαν τους αλγορίθμους εκτίμησης διασποράς Estimation of distribution algorithms (EDAs) (Larrañaga & Lozano, 2002). Οι στοχαστικές αυτές μέθοδοι οδηγούν την αναζήτηση για βέλτιστη τιμή χτίζοντας και λαμβάνοντας δείγμα συγκεκριμένων πιθανολογικών μοντέλων (όπως η ομοιόμορφη κατανομή) για ευοίωνες υποψήφιες λύσεις.

### **1.3. Πρακτικές Εφαρμογές των Εξελκτικών Αλγορίθμων**

Από την φύση τους, οι EA θεωρούνται κατάλληλοι για την επίλυση μεγάλου εύρους πρακτικών προβλημάτων. Για τον λόγο αυτό, χρησιμοποιούνται σε πολλές πρακτικές εφαρμογές που αφορούν διάφορα πεδία της επιστήμης, όπως η μηχανική, οι φυσικές επιστήμες, τα οικονομικά κ.α. Μερικές από τις κυριότερες κατηγορίες πεδίων εφαρμογής, στις οποίες η χρήση των EA αποδεικνύεται εξαιρετικά χρήσιμη, είναι οι ακόλουθες (Back et al., 1997):

- Οι εφαρμογές μηχανικού σχεδιασμού, οι οποίες χρησιμοποιούν συνεχείς παραμέτρους, όπως λ.χ. η μοντελοποίηση αεροσκαφών, τα ηλεκτρομαγνητικά συστήματα καθώς και τα συστήματα αυτόματου ελέγχου κ.α..
- Οι εφαρμογές μηχανικού σχεδιασμού, οι οποίες χρησιμοποιούν διακριτές παραμέτρους, όπως λ.χ. η βελτιστοποίηση ψηφιακών φίλτρων, ο σχεδιασμός γραμμικών επιταχυντών, η βελτιστοποίηση της διαρρύθμισης καυσίμων πυρηνικών αντιδραστήρων κ.α..
- Οι εφαρμογές μηχανικού σχεδιασμού, οι οποίες χρησιμοποιούν παραμέτρους μικτών ακεραίων (προβλημάτων, δηλαδή, όπου μερικές παράμετροι είναι συνεχείς και μερικές είναι διακριτές), όπως λ.χ. ο σχεδιασμός επιβιώσιμων δικτύων κ.α..

- Συνδυαστικά προβλήματα βελτιστοποίησης με δυαδική εκπροσώπηση των λύσεων, κυρίως με την χρήση κανονικών GA, όπως λ.χ. η βέλτιστη διάκριση εργασιών πληρωμάτων αεροπορικών εταιρειών, το πρόβλημα του σάκου κ.α..
- Συνδυαστικά προβλήματα βελτιστοποίησης με εκπροσώπηση μετάλλαξης των λύσεων στον τομέα του χρονικού προγραμματισμού, όπως λ.χ. ο προγραμματισμός παραγωγής, ο προγραμματισμός βέλτιστων δρομολογίων φορτηγών κ.α..

Χάρη στην μεγάλη αποτελεσματικότητα και την αξιοπιστία με την οποία οι ΕΑ έχουν αποδειχθεί ότι αντιμετωπίζουν διάφορα προβλήματα, τα τελευταία χρόνια η δημοτικότητά τους έχει αυξηθεί κατακόρυφα. Μέσα από την έρευνα, προκύπτουν συνεχώς νέες παραλλαγές των υπαρχουσών μεθόδων αλλά και νέοι, διαφορετικοί τομείς εφαρμογής τους τόσο σε πρακτικά όσο και σε θεωρητικά προβλήματα, κάτι το οποίο αναμένεται να συνεχιστεί και στις επόμενες δεκαετίες.

## **ΚΕΦΑΛΑΙΟ 2**

### **ΟΙ ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ**



## **2.1. Περιγραφή Κλασσικού Εξελικτικού Αλγορίθμου**

Από την μέχρι στιγμής αναφορά μας στους εξελικτικούς αλγορίθμους (EA), έχει γίνει ξεκάθαρη η άμεση σχέση τους με την βιολογική διαδικασία της εξέλιξης. Αυτό, όμως, το οποίο δεν έχει αναλυθεί ακόμη, είναι ο τρόπος με τον οποίο η διαδικασία αυτή μπορεί να μοντελοποιηθεί, έτσι ώστε να προκύψουν τα επιθυμητά αποτελέσματα από την εφαρμογή της σε προβλήματα βελτιστοποίησης.

Αρχικά, δημιουργείται ένας πληθυσμός από υποψήφιες λύσεις, από ένα σύνολο τιμών, δηλαδή, το οποίο απαρτίζεται από μια τυχαία τιμή για κάθε μία από τις διαστάσεις του προβλήματός μας. Αυτές οι υποψήφιες λύσεις δημιουργούνται στη βάση της τυχαιότητας, λαμβάνοντας, όμως, πάντα υπόψιν το πεδίο ορισμού κάθε διάστασης, το οποίο και ορίζεται από τα δεδομένα του προβλήματος.

Στην συνέχεια, το σύνολο των υποψήφιων λύσεων αξιολογείται, επιβάλλοντας πάνω τους την αντικειμενική συνάρτηση, η οποία ορίζεται από το πρόβλημα. Ο λόγος για τον οποίο εφαρμόζουμε την αντικειμενική συνάρτηση πάνω στις υποψήφιες λύσεις, είναι προκειμένου να αποφασίσουμε εάν αυτές είναι επιτυχημένες ή αποτυχημένες. Οι υποψήφιες λύσεις ανακατατάσσονται με βάση το πόσο επιτυχημένες ήταν, όπου ως πιο επιτυχημένη θεωρείται αυτή με την βέλτιστη τιμή αντικειμενικής συνάρτησης (μέγιστη για πρόβλημα μεγιστοποίησης και ελάχιστη για πρόβλημα ελαχιστοποίησης).

Ο μελετητής επιλέγει αυθαίρετα το πλήθος των υποψήφιων λύσεων, τις οποίες θα κρατήσει ως «γονείς». Οι γονείς επιλέγονται με βάση την αντικειμενική τους συνάρτηση, είναι, δηλαδή, οι πρώτες από την ανακαταταγμένη λίστα που επεξηγήσαμε προηγουμένως. Αν για παράδειγμα, έχουμε 15 υποψήφιες λύσεις και θέλουμε να κρατήσουμε 7 «γονείς», θα κρατήσουμε τις 7 υποψήφιες λύσεις με την καλύτερη τιμή αντικειμενικής συνάρτησης. Η επιλογή αυτή αναφέρεται στη βιβλιογραφία ως ελιτισμός (elitism), καθώς κρατάμε την «ελίτ» των λύσεων. Ο αριθμός των «γονέων» παραμένει σταθερός καθ' όλη τη διάρκεια του προβλήματος.

Οι επιλεγμένοι γονείς δημιουργούν ένα νέο σύνολο απογόνων, οι οποίοι δίνουν νέες τιμές στο πρόβλημα. Όπως και πριν, επιλέγονται ως «γονείς» το πλήθος των υποψήφιων λύσεων

μέσα από τις οποίες προκύπτουν βέλτιστες αντικειμενικές συναρτήσεις. Από τους νέους γονείς, δημιουργούνται νέοι απόγονοι κ.ο.κ..

Ο αλγόριθμος συνεχίζει την επαναληπτική διαδικασία που επεξηγήσαμε έως ότου ικανοποιηθεί μια συνθήκη τερματισμού του. Αυτή η συνθήκη βρίσκεται στη διακριτική ευχέρεια του μελετητή και μπορεί να είναι η διαφορά ανάμεσα στις βέλτιστες λύσεις 2 ή περισσότερων επαναλήψεων, ο αριθμός των επαναλήψεων ή κάποιος χρονικός παράγοντας.

Το γενικό σχεδιάγραμμα της μεθόδου των εξελικτικών αλγορίθμων, όπως περιγράφηκε προηγουμένως φαίνεται στον ψευδο-κώδικα που ακολουθεί στο Σχήμα 2.1 καθώς και στο διάγραμμα ροής στο Σχήμα 2.2 της επόμενης σελίδας.

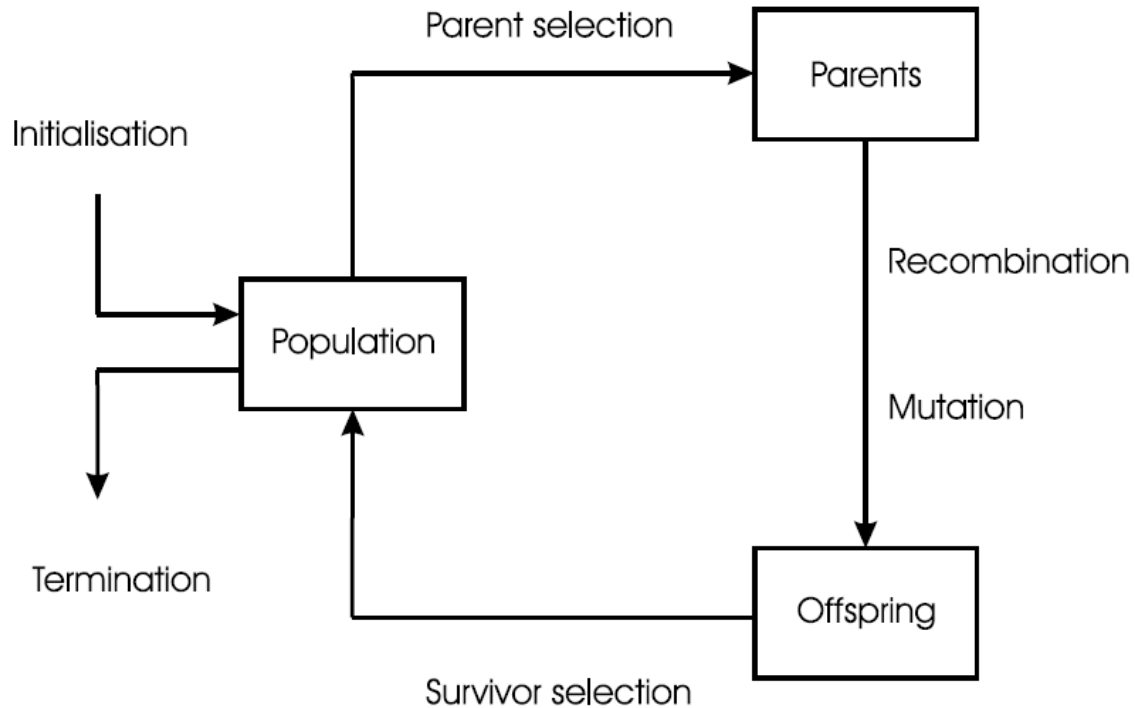
```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

**Σχήμα 2.1:** Γενικό σχεδιάγραμμα Εξελικτικών Αλγορίθμων σε ψευδο-κώδικα

Όπως είναι προφανές, οι εξελικτικοί αλγόριθμοι κατατάσσονται στην κατηγορία των ευρεστικών αλγορίθμων, αφού ψάχνουν μέσα από τυχαίες τιμές την βέλτιστη δυνατή λύση. Η αντικειμενική συνάρτηση, η οποία αναφέρεται συχνά στην βιβλιογραφία και ως fitness function, χρησιμοποιείται προκειμένου να προκύψει μια ποιοτική εκτίμηση της πραγματικής βέλτιστης λύσης ενός πρακτικού προβλήματος. Παράλληλα, η διαδικασία αναζήτησης οδηγείται στο επιθυμητό αποτέλεσμα μέσα από την μεταβλητότητα του προβλήματος αλλά και από τους τελεστές του. Από την φύση τους, οι ΕΑ βασίζονται στο μέγεθος του πληθυσμού ούτως ώστε να παράξουν ταυτόχρονα ένα σύνολο υποψήφιων



λύσεων, σε αντίθεση με αρκετές άλλες μεθόδους όπου λαμβάνεται μια τιμή ανά επανάληψη. Ακόμη, οι ΕΑ αποτελούν μια στοχαστική διαδικασία, αφού χρησιμοποιείται το στοιχείο της τυχαιότητας.



**Σχήμα 2.2:** Γενικό σχεδιάγραμμα Εξελικτικών Αλγορίθμων σε flow-chart

## 2.2. Μαθηματική έκφραση προβλήματος βελτιστοποίησης

Παρότι μέχρι στιγμής έχουμε αναφερθεί πολλές φορές σε προβλήματα βελτιστοποίησης ως το βασικό πεδίο εφαρμογής των ΕΑ, είναι απαραίτητο να τα ορίσουμε και να τα εκφράσουμε μαθηματικά, προκειμένου να μην υπάρχει καμία σύγχυση.

Γενικά, η επίλυση ενός προβλήματος βελτιστοποίησης απαιτεί την εύρεση ενός συνόλου τιμών, οι οποίες αποτελούν το διάνυσμα  $\vec{x} \in \Pi$ , όπου  $\Pi$  το πεδίο τιμών για κάθε παράμετρο του διανύσματος, τέτοιο ώστε (το  $\vec{x}$ ) να βελτιστοποιεί το «κριτήριο ποιότητας», το οποίο ονομάζεται και αντικειμενική συνάρτηση ή fitness function, της  $f: \Pi \rightarrow \mathbb{R}$ , δηλαδή να μεγιστοποιηθεί (ή να ελαχιστοποιηθεί, αντίστοιχα).

Θέλουμε δηλαδή:

$$f(\vec{x}) \rightarrow \max \quad (2.1)$$

Η λύση, η οποία μας δίνει το ολικό μέγιστο της εξίσωσης (2.1), απαιτεί την εύρεση ενός διανύσματος  $\vec{x}^*$  τέτοιο ώστε να ισχύει ότι  $\forall \vec{x} \in \Pi : f(\vec{x}) \leq f(\vec{x}^*) = f^*$  όπου  $f^*$  η βέλτιστη λύση της  $f$ .

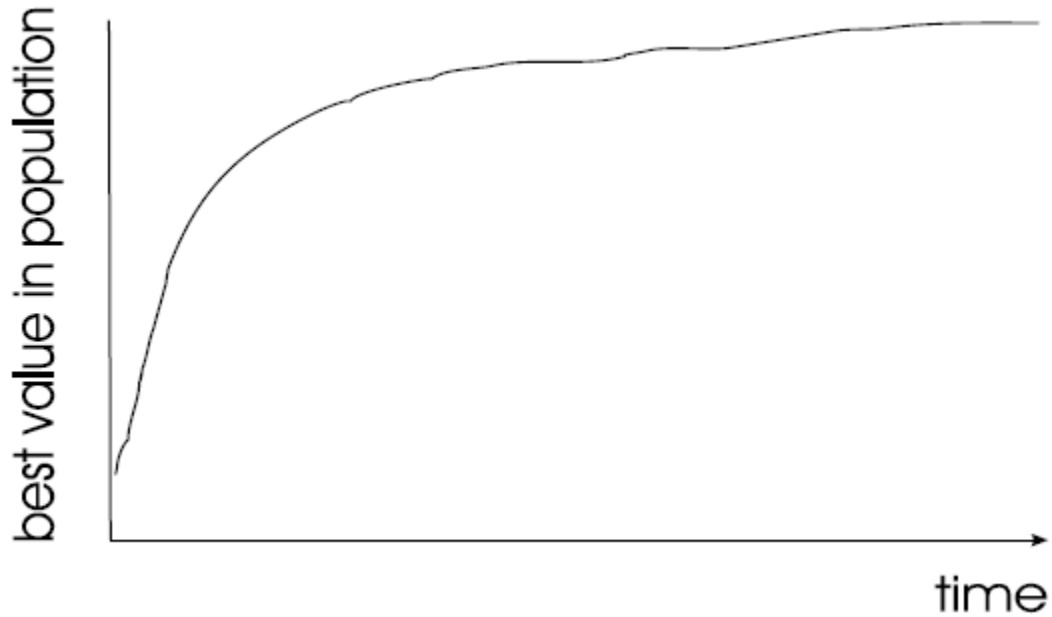
### **2.3. Το πρόβλημα των τοπικών ακροτάτων**

Από την φύση τους, οι ΕΑ ψάχνουν συνεχώς να βρουν την βέλτιστη δυνατή τιμή της αντικειμενικής συνάρτησης μέσα από τις υποψήφιες λύσεις που έχουν διαθέσιμες σε κάθε επανάληψη. Ένα από τα συχνότερα προβλήματα που αντιμετωπίζουν οι ΕΑ είναι η πιθανότητα εγκλωβισμού του αλγορίθμου σε τοπικά ακρότατα (είτε τοπικά μέγιστα για πρόβλημα μεγιστοποίησης, είτε τοπικά ελάχιστα για πρόβλημα ελαχιστοποίησης).

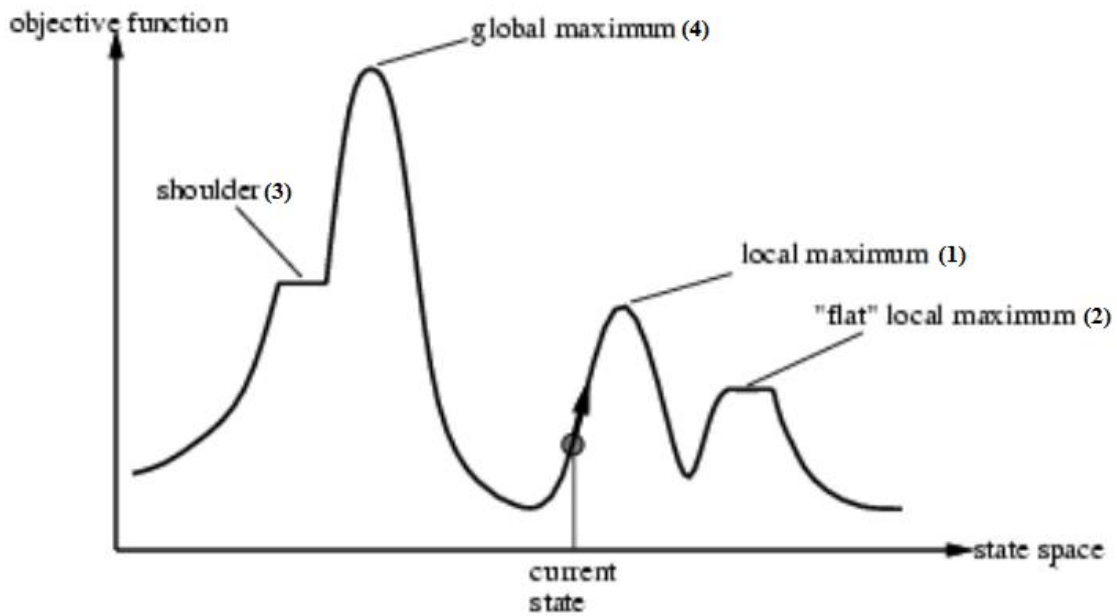
Ένα χαρακτηριστικό των μεθόδων, οι οποίοι βασίζονται σε ΕΑ, είναι η τάση τους να «τραβούν» τις υποψήφιες λύσεις προς το μέρος των βέλτιστων υποψήφιων λύσεων που έχουν βρεθεί μέχρι στιγμής. Στην ουσία, η εκμετάλλευση του χαρακτηριστικού αυτού, είναι ένας από τους κύριους λόγους, στους οποίους οφείλεται η αποτελεσματικότητα των ΕΑ. Κάποιες φορές όμως, αυτός είναι και ο λόγος που μπορεί να οδηγηθούμε σε λανθασμένη λύση, εξαιτίας της ύπαρξης τοπικών ακροτάτων στο πολυδιάστατο πρόβλημά μας. Όπως φαίνεται από το Σχήμα 2.3, το εύρος τιμών της βέλτιστης λύσης ομαλοποιείται σε σχετικά σύντομο χρονικό διάστημα και στην συνέχεια υπάρχουν μόνο μικρές βελτιώσεις στη τιμή της βέλτιστης λύσης της αντικειμενικής συνάρτησης.

Όπως φαίνεται από το Σχήμα 2.4, το οποίο είναι πρόβλημα μεγιστοποίησης, υπάρχουν διάφοροι λόγοι (Eiben & Smith, 2003) για τους οποίους οι υποψήφιες λύσεις μας μπορεί να εγκλωβιστούν και να μην οδηγηθούν ποτέ στην βέλτιστη λύση του προβλήματος (σημείο 4 σχήματος 2.4). Αρχικά, η ύπαρξη απλού τοπικού ακρότατου (σημείο 1 σχήματος 2.4) μπορεί να οδηγήσει τις υποψήφιες λύσεις σε λανθασμένη περιοχή. Ακόμα, η ύπαρξη επίπεδης περιοχής τοπικού ακρότατου (σημείο 2 σχήματος 2.4) μπορεί να κρατήσει τα σωματίδια μακριά από την εξερεύνηση της σωστής περιοχής. Τέλος, η ύπαρξη όμων

(shoulders) (σημείο 3 σχήματος 2.4) ακόμα και σε κοντινή περιοχή με το ολικό ακρότατο, μπορεί να αποτρέψει την εύρεση της βέλτιστης τιμής της αντικειμενικής συνάρτησης.



Σχήμα 2.3: Αναπαράσταση μεταβολής βέλτιστης τιμής σε σχέση με τον χρόνο



Σχήμα 2.4: Παραδείγματα εγκλωβισμού του προβλήματος σε τοπικά ακρότατα

Μέσα από πρακτικές εφαρμογές, έχει αποδειχθεί ότι οι αποτελεσματικότεροι τρόποι αντιμετώπισης εγκλωβισμού σε τοπικά ακρότατα είναι η αύξηση των «γονέων» που κρατάμε ύστερα από κάθε επανάληψη, η αύξηση του εύρους της τυχαιότητας που προσδίδεται από προηγούμενη γενιά σε επόμενη γενιά σωματιδίων-υποψήφιων λύσεων καθώς και η εκτέλεση πολλών πειραμάτων, προκειμένου να επιβεβαιωθεί η σωστή βέλτιστη λύση. Είναι προφανές ότι οι παραπάνω προτάσεις επιβαρύνουν χρονικά την επίλυση του προβλήματος, αυξάνουν όμως κατακόρυφα την αξιοπιστία των λύσεων που προκύπτουν από την χρήση των ΕΑ.

## **2.4. Μέθοδοι αντιμετώπισης περιορισμών**

Μέχρι στιγμής έχουμε αναφερθεί σε προβλήματα βελτιστοποίησης, τα οποία δεν ασχολούνται με κανέναν περιορισμό, πέρα από τους περιορισμούς που τίθενται με βάση το πεδίο τιμών των στοιχείων του διανύσματος  $\vec{x}$ . Όμως, όπως είναι προφανές, κανένα φυσικό πρόβλημα δεν μπορεί να αντιμετωπιστεί σαν να μην έχει κανέναν περιορισμό, καθότι αυτό είναι πρακτικά αδύνατο. Έτσι, υπήρξε η ανάγκη να αναπτυχθούν διάφορες μέθοδοι, οι οποίες θα βοηθούσαν στην μοντελοποίηση και την επίλυση προβλημάτων, τα οποία υπόκεινται σε περιορισμούς ισότητας και ανισότητας, με την χρήση των ΕΑ.

Σύμφωνα με την βιβλιογραφία (Michalewicz & Schoenauer, 1996) υπάρχουν 4 κατηγορίες μεθόδων αντιμετώπισης περιορισμών για προβλήματα βελτιστοποίησης, οι οποίες και αναλύονται στην συνέχεια.

### **2.4.1. Μέθοδοι διατήρησης της δυνατότητας επίτευξης εφικτής λύσης**

Σε αυτή την κατηγορία υπάγονται 2 διαφορετικές μέθοδοι:

- Ο Michalewicz και ο Janikow ανέπτυξαν (Michalewicz & Janikow, Handling constraints in genetic algorithms, 1991) ένα σύστημα, το οποίο ονομάζεται Genocop (Genetic algorithm for Numerical Optimization of COnstrained Problems). Η μέθοδος απαιτεί την ύπαρξη μόνο γραμμικών περιορισμών και έναν αρχικό πληθυσμό εφικτών λύσεων προκειμένου να λειτουργήσει. Η βασική ιδέα της μεθόδου βασίζεται σε ειδικούς τελεστές, οι οποίοι μεταλλάσσουν εφικτές υποψήφιες λύσεις μιας γενιάς σε εφικτές υποψήφιες της επόμενης γενιάς. Η

μέθοδος αυτή μπορεί να γενικευτεί ώστε να χρησιμοποιηθεί και σε μη γραμμικά προβλήματα, υπό την προϋπόθεση ότι ο χώρος εφικτών λύσεων είναι κυρτός.

- Παρότι η μέθοδος της στρατηγικής ταλάντωσης είχε προταθεί από το 1977 (Glover, 1977), πρόσφατα χρησιμοποιήθηκε για την βελτιστοποίηση μη-γραμμικών προβλημάτων (Glover & Kochenberger, 1995). Η μέθοδος βασίζεται στην αναγνώριση μιας κρίσιμης περιοχής, η οποία λειτουργεί σαν όριο ανάμεσα στην εφικτή και την μη-εφικτή περιοχή λύσεων. Η στρατηγική αυτή αποσκοπεί να οδηγηθεί στην μη-εφικτή περιοχή εσκεμμένα, θέτοντας στην συνέχεια ποινές στην αντικειμενική συνάρτηση, προκειμένου να δώσει στο πρόβλημα τη σωστή κατεύθυνση λύσεων.

#### 2.4.2. Μέθοδοι βασισμένοι σε συντελεστές ποινής

Η πλειοψηφία των μεθόδων αντιμετώπισης βελτιστοποίησης προβλημάτων, τα οποία υπόκεινται σε περιορισμούς, βασίζονται στην χρήση των συντελεστών ποινής (penalty functions). Οι συντελεστές αυτοί «τιμωρούν» τις μη-εφικτές λύσεις, μετασχηματίζοντας το κλασικό πρόβλημα ελαχιστοποίησης ΕΑ στο νέο:

$$f'(\vec{x}) = f(\vec{x}) + \text{penalty}(\vec{x}) \quad (2.2)$$

όπου το  $\text{penalty}(\vec{x})$  είναι 0 εάν δεν υπάρχει καμία παράβαση περιορισμού και αρνητικό σε κάθε άλλη περίπτωση, μειώνοντας έτσι την τιμή της αντικειμενικής συνάρτησης όσο μεγαλύτερο είναι το σφάλμα. Αυτή η «διόρθωση» της λύσης οδηγεί τελικά το πρόβλημα στην εφικτή περιοχή του (Προφανώς σε πρόβλημα μεγιστοποίησης ο συντελεστής ποινής προστίθεται αντί να αφαιρείται).

Στις μεθόδους αυτές είναι πολύ συχνή η χρήση των συναρτήσεων  $f_j(\vec{x})$  για  $(1 \leq j \leq m)$ , όπου  $m$  το άθροισμα του πλήθους  $q$  των συναρτήσεων ανισότητας  $g_j(\vec{x})$  και του πλήθους  $(m-q)$  των συναρτήσεων ισότητας  $h_j(\vec{x})$ , για τον υπολογισμό του  $\text{penalty}(\vec{x})$ . Ο υπολογισμός της  $f_j(\vec{x})$  γίνεται ως εξής:

$$f_j(\vec{x}) = \begin{cases} \max\{0, g_j(\vec{x})\}, & \text{για } 1 \leq j < q \\ |h_j(\vec{x})|, & \text{για } q + 1 < j \leq m \end{cases} \quad (2.3)$$

Ο τρόπος με τον οποίο οι ακόλουθες μέθοδοι διαχειρίζονται τον συντελεστή ποινής είναι αυτός, ο οποίος τις διακρίνει:

- Η πιο απλή και πιο συχνά χρησιμοποιούμενη μέθοδος προτάθηκε σχετικά πρόσφατα (Runarsson & Yao, 2005) και μετασχηματίζει την εξίσωση (2.2) στην:

$$f'(\vec{x}) = f(\vec{x}) + \lambda * f_j(\vec{x}) \quad (2.4)$$

Όπου η τιμή  $\lambda$  παίρνει αυθαίρετες τιμές, τις οποίες επιλέγει ο μελετητής με βάση τα αποτελέσματα που προκύπτουν σε ένα πρόβλημα.

- Η μέθοδος του στατικού συντελεστή ποινής προτάθηκε από τους Homaifar, Lai και Qi (Homaifar et al., 1994). Θεωρεί συντελεστές βαρύτητας του συντελεστή ποινής ανάλογα με το επίπεδο παραβίασης των περιορισμών και χωρίζει τα επίπεδα παραβίασης σε  $\lambda$  επίπεδα. Έτσι οι συντελεστές ορίζονται ως  $R_{ij}$  ( $i = 1, 2, \dots, \lambda, j = 1, 2, \dots, m$ ) για κάθε επίπεδο παραβίασης και για κάθε περιορισμό. Όσο μεγαλύτερη η παραβίαση τόσο μεγαλύτεροι και οι συντελεστές βαρύτητας. Η εξίσωση (2.2) μετασχηματίζεται στην:

$$f'(\vec{x}) = f(\vec{x}) + \sum_{j=1}^m R_{ij} f_j^2(\vec{x}) \quad (2.5)$$

- Η μέθοδος δυναμικού συντελεστή ποινής προτάθηκε το 1994 (Joines & Houck, 1994), μετατρέποντας τον συντελεστή σε δυναμικό. Συγκεκριμένα, στην  $t$  επανάληψη και για σταθερές  $\alpha, \beta$  και  $C$  η εξίσωση (2.2) μετασχηματίζεται στην:

$$f'(\vec{x}) = f(\vec{x}) + (C * t)^\alpha \sum_{j=1}^m f_j^\beta(\vec{x}) \quad (2.6)$$

- Η μέθοδος ανόπτησης συντελεστή ποινής, γνωστή και ως Genocop II, βασίζεται στους δυναμικούς συντελεστές (Michalewicz & Attia, 1994), (Michalewicz Z. , 1996). Αφού διαχωρίσει τους περιορισμούς με βάση τη γραμμικότητα τους και αν είναι ισότητες ή ανισότητες (4 κατηγορίες, δηλαδή), επιλέγει τυχαία ένα αρχικό σημείο, το οποίο πρέπει να ικανοποιεί τις γραμμικές συνθήκες. Θέτει έναν παράγοντα θερμοκρασίας  $\tau = \tau_0$  αρχικά και μετατρέπει τη σχέση (2.2) ως εξής:

$$f'(\vec{x}) = f(\vec{x}) + \frac{1}{2\tau} \sum_{j=1}^m f_j^2(\vec{x}) \quad (2.7)$$

Εάν το  $t$  γίνει μικρότερο από μια τιμή  $t_f$ , η διαδικασία διακόπτεται, αλλιώς το  $t$  συνεχίζει να μικραίνει, ενώ η βέλτιστη λύση λειτουργεί ως αρχική της επόμενης επανάληψης μέχρι την διακοπή της διαδικασίας.

- Υπάρχουν ακόμα 2 μέθοδοι προσαρμοσίμων συντελεστών ποινής. Η πρώτη εξελίχθηκε το 1992 (Bean & Hadj-Alouane, 1992). Η σχέση (2.2) μετασχηματίζεται στην:

$$f'(x) = f(x) + \lambda(t) \sum_{j=1}^m f_j^2(x) \quad (2.8)$$

Όπου  $t$  είναι ο αριθμός της επανάληψης-γενιάς για πεδίο ορισμού  $\Pi$  και διάστημα αναζήτησης  $S$  τέτοιο ώστε το  $\Pi \subseteq S$  να καθορίζει την εφικτή περιοχή. Το  $\lambda(t)$  υπολογίζεται ως εξής:

$$\lambda(t+1) = \begin{cases} \frac{1}{\beta_1} * \lambda(t), & \text{εάν } \bar{b}^i \in \Pi \forall t - k + 1 \leq i \leq t \\ \beta_2 * \lambda(t), & \text{εάν } \bar{b}^i \in S - \Pi \forall t - k + 1 \leq i \leq t \\ \lambda(t), & \text{σε κάθε άλλη περίπτωση} \end{cases} \quad (2.9)$$

Όπου  $\bar{b}^i$  η βέλτιστη υποψήφια λύση της  $f'(x)$ ,  $\beta_1$  και  $\beta_2$  σταθερές και  $k$  μια σταθερά που ορίζεται από τον μελετητή και αφορά τις  $k$  τελευταίες επαναλήψεις όπου η λύση βρίσκεται στην εφικτή περιοχή.

Η δεύτερη μέθοδος είναι αυτή, η οποία αναπτύχθηκε ένα χρόνο αργότερα (Smith & Tate, 1993). Περιλαμβάνει τον υπολογισμό ενός συνόλου σχεδόν εφικτών τιμών  $q_j \forall j: 1 \leq j \leq m$ . Έτσι η σχέση (2.2) γίνεται:

$$f'(x) = f(x) + F_{feas(t)} - F_{all(t)} \sum_{j=1}^m (f_j(x) / q_j(t))^k \quad (2.10)$$

Όπου  $F_{all(t)}$  η τιμή της βέλτιστης λύσης που έχει βρεθεί ως εκείνη την στιγμή, εάν μηδενιστεί η ποινή, ενώ  $F_{feas(t)}$  η τιμή της βέλτιστης λύσης που έχει βρεθεί ως εκείνη την στιγμή, ενώ  $k$  σταθερά.

- Η μέθοδος θανατικής ποινής, η οποία προτάθηκε το 1991 (Back et al., 1991) και απορρίπτει όλες τις μη-εφικτές τιμές.

- Η μέθοδος διαχωρισμού γενετικού αλγορίθμου, η οποία προτάθηκε το 1995 (Leriche et al., 1995). Σύμφωνα με την μέθοδο αυτή, επιλέγουμε εσκεμμένα μια πολύ μικρή τιμή ποινής  $p_1$  καθώς και μια πολύ μεγάλη τιμή ποινής  $p_2$ . Η σχέση (2.2) μετατρέπεται στην:

$$f'(\vec{x}) = f(\vec{x}) + p_i(\vec{x}), \quad i = 1,2 \quad (2.11)$$

Στην συνέχεια δημιουργούνται 2 διαφορετικές λίστες, οι οποίες στοιβάζουν τις υποψήφιες λύσεις με βάση τις τιμές της αντικειμενικής συνάρτησης  $f'(\vec{x})$ . Η επόμενη γενιά δημιουργείται επιλέγοντας εναλλάξ την καλύτερη υποψήφια λύση και διαγράφοντάς την από την άλλη λίστα.

### 2.4.3. Μέθοδοι βασισμένοι στην αναζήτηση για βέλτιστη λύση

Υπάρχουν 3 μέθοδοι, οι οποίες δίνουν έμφαση ανάμεσα στις εφικτές και τις μη-εφικτές λύσεις, σε ένα διάστημα αναζήτησης  $S$ :

- Η μέθοδος μνήμης συμπεριφοράς (Schoenauer & Xanthakis, 1993): Ξεκινά από μια τυχαία λύση (όχι απαραίτητα εντός περιορισμών), προχωράει τις επαναλήψεις έως ότου ένα ποσοστό του πληθυσμού είναι εφικτό για έναν περιορισμό. Την στιγμή εκείνη αυξάνεται ένας μετρητής που έχουμε θέσει και το πρόβλημα ξεκινάει εκ νέου με αρχικές τιμές τις τιμές στις οποίες είχαμε σταματήσει. Κάθε περιορισμός πρέπει να είναι γραμμικός και οι περιορισμοί ελέγχονται στην σειρά.
- Η μέθοδος ανωτερότητας εφικτών σημείων (Powell & Skolnick, 1993): Βασίζεται στην κλασσική προσέγγιση ποινής με μία βασική διαφορά, καθώς προσθέτει στην  $f'(\vec{x})$  μια εξαρτημένη συνάρτηση, η οποία επηρεάζει τον υπολογισμό των μη-εφικτών λύσεων. Έτσι, η εξίσωση (2.2) γίνεται:

$$f'(\vec{x}) = f(\vec{x}) + r \sum_{j=1}^m f_j(\vec{x}) + \theta(t, \vec{x}) \quad (2.12)$$

Όπου  $r$  σταθερά και το  $\theta(t, \vec{x})$  υπολογίζεται:

$$f(\vec{x}) = \begin{cases} 0, & \text{εάν } \vec{x} \in \Pi \\ \max \left\{ 0, \max_{\vec{x} \in \Pi} \{f(\vec{x})\} - \min_{\vec{x} \in S - \Pi} \{f(\vec{x})\} + r \sum_{j=1}^m f_j(\vec{x}) \right\}, & \text{αλλιώς} \end{cases} \quad (2.13)$$

- Η μέθοδος επιδιόρθωσης μη-εφικτών υποψήφιων λύσεων αναπτύχθηκε το 1995 (Michalewicz & Nazhiyath, 1995) και αποκαλείται στην βιβλιογραφία ως Genocop



III. Η μέθοδος αυτή χρησιμοποιεί την κλασική μέθοδο Genocop επεκτείνοντάς την, αφού διατηρεί 2 διακριτούς πληθυσμούς, όπου η ανάπτυξη του ενός πληθυσμού επηρεάζει τα σωματίδια του δεύτερου πληθυσμού. Ο πρώτος πληθυσμός ( $P_r$ ) θεωρείται ως ένα σημείο αναφοράς και αποτελείται από σωματίδια που ικανοποιούν τους γραμμικούς περιορισμούς ενώ ο δεύτερος ( $P_s$ ) ικανοποιεί όλους τους περιορισμούς. Ο πληθυσμός  $P_r$  μεταβάλλεται κάθε  $k$  (παράμετρος) γενιές. Ο ψευδο-κώδικας της μεθόδου φαίνεται αναλυτικά στο σχήμα 2.5 που ακολουθεί:

```

procedure Genocop III
begin
   $t \leftarrow 0$ 
  initialize  $P_s(t)$ 
  initialize  $P_r(t)$ 
  evaluate  $P_s(t)$ 
  evaluate  $P_r(t)$ 
  while (not termination-condition) do
    begin
       $t \leftarrow t + 1$ 
      select  $P_s(t)$  from  $P_s(t - 1)$ 
      alter  $P_s(t)$ 
      evaluate  $P_s(t)$ 
      if  $t \bmod k = 0$  then
        begin
          alter  $P_r(t)$ 
          select  $P_r(t)$  from  $P_r(t - 1)$ 
          evaluate  $P_r(t)$ 
        end
      end
    end
  end

```

**Σχήμα 2.5:** Ψευδο-κώδικας αλγορίθμου Genocop III

#### 2.4.4. Υβριδικές μέθοδοι

Εξαιτίας της σχετικής ευκολίας δημιουργίας υβριδικών μεθόδων, υπάρχουν αρκετές διαφορετικές μέθοδοι που έχουν αναπτυχθεί κατά καιρούς.

- Ο Waagan συνδύασε την τεχνική εξελικτικού προγραμματισμού (Waagen et al., 1992) με την μέθοδο άμεσης εύρεσης (Hooke & Jeeves, 1961) για την επίλυση προβλημάτων χωρίς περιορισμούς.
- Ο Myung ακολούθησε μια αντίστοιχη προσέγγιση (Myung et al., 1995) αλλά για προβλήματα που υπόκεινται σε περιορισμούς.
- Ο Schaffer ανέπτυξε τη μέθοδο γενετικού αλγορίθμου υπολογισμένου με διάνυσμα (Schaffer, 1985).
- Άλλοι μελετητές ακολούθησαν μια παρόμοια προσέγγιση στην ανάπτυξη τεχνικών σχεδίασης χώρων που υπόκεινται σε περιορισμούς (Parmee & Purchase, 1994).
- Ο Paredis βασίστηκε σε ένα συν-εξελικτικό μοντέλο όπου ένας πληθυσμός υποψήφιων λύσεων συν-εξελίσσεται με έναν πληθυσμό περιορισμών (Paredis, 1994). Οι καλύτερες λύσεις ικανοποιούν περισσότερους περιορισμούς και παράλληλα οι καλύτεροι περιορισμοί παραβιάζονται από περισσότερες λύσεις.
- Πρόσφατα, έχουν ερευνηθεί υβριδικές μέθοδοι που συνδυάζουν τις τεχνικές εξελικτικού προγραμματισμού με ντετερμινιστικές διαδικασίες για αριθμητικά προβλήματα βελτιστοποίησης (Papadrakakis et al., 1999).

## ΚΕΦΑΛΑΙΟ 3

# ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΜΕΛΕΤΗΘΕΝΤΩΝ ΑΛΓΟΡΙΘΜΩΝ



### 3.1. Σύνοψη κεφαλαίου

Κατά την εκπόνηση της διπλωματικής εργασίας, μοντελοποιήθηκαν και μελετήθηκαν 2 διαφορετικοί εξελικτικοί αλγόριθμοι. Στο κεφάλαιο αυτό αναλύουμε το μαθηματικό μοντέλο πάνω στο οποίο βασίζεται κάθε ένας από τους εξελικτικούς αλγορίθμους μας, τι τους διαχωρίζει από τις κλασσικές μεθόδους βελτιστοποίησης με εξελικτικούς αλγορίθμους καθώς και ποιο είναι το επιστημονικό κενό, το οποίο καλύπτεται μέσα από τα πειράματα που εκτελέστηκαν στο πλαίσιο της εκπόνησης της διπλωματικής εργασίας.

### 3.2. Η μέθοδος Global Competitive Ranking

#### 3.2.1. Μαθηματικό μοντέλο ελαχιστοποίησης

Το μαθηματικό μοντέλο του κλασσικού μη γραμμικού προβλήματος ελαχιστοποίησης αποτυπώνεται ως:

$$\min f(\vec{x}), \quad \vec{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n \quad (3.1)$$

Όπου  $f(\vec{x})$  η αντικειμενική συνάρτηση και το  $\vec{x} \in S \cap F, S \subseteq \mathbb{R}^n$  καθορίζει το διάστημα αναζήτησης, το οποίο είναι n-διάστατο. Το πεδίο τιμών του  $\vec{x}$  ορίζεται για κάθε διάσταση ως περιορισμοί παραμέτρων:

$$\underline{x}_j \leq x_j \leq \overline{x}_j, \quad j \in \{1, \dots, n\} \quad (3.2)$$

Και η εφικτή περιοχή  $F$  από την σχέση:

$$F = \{\vec{x} \in \mathbb{R}^n \mid g_k(x) \leq 0 \forall k \in \{1, \dots, m\}\} \quad (3.3)$$

Όπου  $g_k(x)$  περιορισμοί ανισότητας. Οι περιορισμοί ισότητας  $h(x)$  υπολογίζονται προσεγγιστικά από περιορισμούς ανισότητας, χρησιμοποιώντας την σχέση:

$$|h(x)| - \varepsilon \leq 0 \quad (3.4)$$

Όπου  $\varepsilon$  ένας πολύ μικρός θετικός αριθμός που μας δείχνει τον βαθμό της παραβίασης του περιορισμού. Τα προβλήματα μεγιστοποίησης, τα οποία μοντελοποιήσαμε μέσω της εξίσωσης (2.1), θα αντιμετωπιστούν εδώ ως προβλήματα ελαχιστοποίησης της αρνητικής αντικειμενικής εξίσωσης:

$$\max\{f(\vec{x})\} = -\min\{-f(\vec{x})\} \quad (3.5)$$

### 3.2.2. Μαθηματικό μοντέλο Global Competitive Ranking

Η μέθοδος Global Competitive Ranking, την οποία θα περιγράψουμε, παρουσιάστηκε για πρώτη φορά το 2003 (Runarsson & Yao, 2003). Σύμφωνα με την βιβλιογραφία (Azad & Fernandes, 2011), η βασική της διαφορά με τις άλλες μεθόδους βρίσκεται στον τρόπο με βάση τον οποίο διαχειρίζεται τους περιορισμούς, όσον αφορά την βαρύτητα που έχει το άθροισμα των ποινών σε σχέση με την αντικειμενική συνάρτηση. Στην συνέχεια η μέθοδος περιγράφεται αναλυτικά.

Η μέθοδος Global Competitive Ranking αποσκοπεί στην εύρεση της κατάλληλης ισορροπίας ανάμεσα στις ποινές και την αντικειμενική συνάρτηση, έτσι ώστε να οδηγήσει το πρόβλημα στην εύρεση της βέλτιστης δυνατής λύσης. Για την περιγραφή του προβλήματος, θα θεωρήσουμε ένα πρόβλημα ελαχιστοποίησης όπως αυτό έχει αποτυπωθεί από τις εξισώσεις που αναφέραμε προηγουμένως ((3.1), (3.2), (3.3), (3.4)), το οποίο μπορεί να έχει περιορισμούς ανισότητας αλλά και ισότητας.

Για την επίλυση του προβλήματος θεωρούμε ένα πληθυσμό  $\lambda$  σωματιδίων. Κάθε σωματίδιο χαρακτηρίζεται από ένα διάνυσμα  $\vec{x}$  (σχέση (3.1)). Σε αντίθεση με άλλες μεθόδους, όπως η Stochastic Ranking (Runarsson & Yao, 2000) όπου κάθε σωματίδιο συγκρίνεται μόνο με τα γειτονικά του προκειμένου να καταταγεί, ένα βασικό χαρακτηριστικό της Global Competitive Ranking είναι η σύγκριση ενός σωματιδίου  $i$  με όλα τα σωματίδια του πληθυσμού προκειμένου να καταταγεί (γι' αυτό τον λόγο, άλλωστε, και παίρνει τον χαρακτηρισμό Global).

Προκειμένου να υλοποιηθεί η συγκεκριμένη μέθοδος, τα σωματίδια κατατάσσονται με βάση 2 χαρακτηριστικές τιμές τους. Για κάθε σύνολο τιμών τους, υπολογίζουμε την τιμή που λαμβάνει η αντικειμενική συνάρτηση  $f(\vec{x})$  αλλά και το άθροισμα των  $q$  περιορισμών ( $m$  περιορισμοί ανισότητας και  $q-m$  περιορισμοί ισότητας) όπως φαίνονται στην εξίσωση:

$$g(x_i) = \text{sum}(\max\{0, g_k(x_i)\}) + \max\{0, |h_l(x_i)| - \varepsilon\}$$
$$k \in \{1, \dots, m\}, \quad l \in \{m + 1, \dots, q\} \quad (3.6)$$

Το άθροισμα των περιορισμών ονομάζεται συχνά και ποινή ή συνάρτηση ποινής.

Στην συνέχεια, ταξινομούμε σε 2 λίστες τα σωματίδια, μία με βάση την τιμή της αντικειμενικής και μία με βάση την τιμή του αθροίσματος των περιορισμών (οι

ταξινομήσεις γίνονται κατά αύξουσα σειρά, αφού σε πρόβλημα ελαχιστοποίησης θεωρούμε βέλτιστη τιμή της αντικειμενικής την μικρότερη - αντίστοιχα σε πρόβλημα μεγιστοποίησης οι ταξινομήσεις θα γινόντουσαν με αύξουσα σειρά ενώ το άθροισμα των περιορισμών θα υπολογιζόταν ως το άθροισμα των  $\min\{0, g_k(\vec{x})\}$  και  $\min\{0, |h_l(\vec{x})| - \varepsilon\}$ .

Έχουμε, λοιπόν, δύο λίστες με σωματίδια, ταξινομημένα κατά αύξουσα σειρά. Δίνουμε στα σωματίδια αυτά μια τιμή ανάλογα με την θέση τους στην κάθε κατάταξη. Αυτό δηλαδή με την βέλτιστη αντικειμενική παίρνει την τιμή 1, το επόμενο την τιμή 2 κ.ο.κ. ενώ αντίστοιχα αυτό με την χαμηλότερη τιμή παραβίασης περιορισμών την τιμή 1, το επόμενη την τιμή 2. Ιδιαίτερη μνεία πρέπει να γίνει στα σωματίδια, τα οποία εμφανίζουν ίδιες τιμές αντικειμενικής ή ίδιες τιμές παραβίασης. Στα σωματίδια αυτά, θα αποδοθεί η τιμή του χαμηλότερου σωματιδίου που γίνεται, δηλαδή εάν δύο σωματίδια είναι ισόπαλα για την δεύτερη καλύτερη τιμή, θα πάρουν και τα δύο την τιμή 2 στην κατάταξη ενώ το/τα επόμενα σωματίδια θα πάρουν την τιμή 4 κ.ο.κ..

Οι δύο αυτές λίστες θα χρησιμοποιηθούν προκειμένου να προκύψει η λίστα, η οποία θα ταξινομεί πλήρως το σωματίδιο. Θεωρούμε, στην συνέχεια, ότι για την πλήρη ταξινόμηση θα χρησιμοποιηθεί είτε η αντικειμενική συνάρτηση είτε η συνάρτηση της ποινής που περιγράψαμε προηγουμένως. Η απόφαση για το ποια από τις δύο αυτές συναρτήσεις θα χρησιμοποιηθεί κάθε φορά δεν είναι στοχαστικό μέγεθος. Χαρακτηρίζεται από την πιθανότητα  $P_f$  η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση. Αντίστοιχα, επειδή τα δύο αυτά ενδεχόμενα είναι τα μοναδικά πιθανά και είναι αλληλο-αποκλειόμενα, είναι προφανές ότι η πιθανότητα η σύγκριση να γίνει με βάση μόνο την συνάρτηση ποινής είναι  $(1 - P_f)$ .

Έτσι, λοιπόν, έχουμε  $\lambda$  σωματίδια  $i$ , τα οποία έχουν λάβει μια τιμή κατάταξης  $\pi_f(i)$  και  $\pi_\varphi(i)$ , ανάλογα με το εάν έχουν καταταγεί με βάση την αντικειμενική συνάρτηση ή με βάση την συνάρτηση ποινής αντίστοιχα. Εάν υπολογίσουμε και την πιθανότητα  $P_f$  η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση που αναφέραμε παραπάνω, η πιθανότητα ένα σωματίδιο  $i$  να διατηρήσει την κατάταξή του  $\pi(i)$  όταν συγκριθεί με ένα άλλο σωματίδιο είναι:

$$P(\pi(i)) = P_f \frac{\lambda - \pi_f(i)}{\lambda - 1} + (1 - P_f) \frac{\lambda - \pi_\phi(i)}{\lambda - 1} \quad (3.7)$$

Ενώ η αντικειμενική συνάρτηση για πρόβλημα ελαχιστοποίησης μετατρέπεται:

$$\psi(x_i) = P_f \frac{\pi_f(i) - 1}{\lambda - 1} + (1 - P_f) \frac{\pi_\phi(i) - 1}{\lambda - 1} \quad (3.8)$$

Με βάση την τιμή  $\psi(x_i)$  που παίρνει κάθε ένα από τα σωματίδια, αυτά ταξινομούνται πάλι με βάση την αύξουσα τιμή και λαμβάνουν μια τιμή ανάλογα με την θέση τους στην κάθε κατάταξη όπως συμβαίνει και στην κατάταξη με βάση την  $f(x_i)$  και την  $g(x_i)$ . Η κατάταξη αυτή με βάση το  $\psi(x_i)$  είναι αυτή με βάση την οποία ο αλγόριθμος κρατά τους γονείς στην επόμενη γενιά, όπως περιγράφεται στην παράγραφο 2.1.

Από την ανάλυση που έχουμε κάνει, είναι προφανές ότι το  $P_f$  μπορεί να χρησιμοποιηθεί για να προσδώσει στο πρόβλημα μεροληψία ως προς την προτίμηση της αντικειμενικής συνάρτησης ή της συνάρτησης ποινής. Στην πράξη (Runarsson & Yao, 2003), η πιθανότητα αυτή έχει αποδειχθεί ότι πρέπει να παίρνει τιμές  $0 < P_f < 0.5$  προκειμένου να εγγυηθεί η εύρεση μιας εφικτής λύσης. Όσο πιο κοντά είναι η πιθανότητα στο 0.5 τόσο μεγαλύτερη είναι η έμφαση στην ελαχιστοποίηση της αντικειμενικής συνάρτησης.

Η Global Competitive Ranking μπορεί να αποδοθεί συνοπτικά στον ψευδο-κώδικα του σχήματος 3.1.

**Global competitive ranking ( $P_f, f, \phi$ ):**  
 Step 1: Determine the ranking,  $\pi_f, \pi_\phi$ :  
 $f(\mathbf{x}_{\pi_f^{-1}(1)}) \leq f(\mathbf{x}_{\pi_f^{-1}(2)}) \leq \dots \leq f(\mathbf{x}_{\pi_f^{-1}(\lambda)})$   
 $\phi(\mathbf{x}_{\pi_\phi^{-1}(1)}) \leq \phi(\mathbf{x}_{\pi_\phi^{-1}(2)}) \leq \dots \leq \phi(\mathbf{x}_{\pi_\phi^{-1}(\lambda)})$   
 Step 2. Compute competitive fitness:  
 $\psi(\mathbf{x}_i) = P_f \frac{\pi_f(i) - 1}{\lambda - 1} + (1 - P_f) \frac{\pi_\phi(i) - 1}{\lambda - 1}.$   
 Step 3. Determine final ranking,  $\pi$ :  
 $\psi(\mathbf{x}_{\pi^{-1}(1)}) \leq \psi(\mathbf{x}_{\pi^{-1}(2)}) \leq \dots \leq \psi(\mathbf{x}_{\pi^{-1}(\lambda)})$

**Σχήμα 3.1:** Αναπαράσταση Global Competitive Ranking σε ψευδο-κώδικα

### 3.2.3. Βελτιστοποίηση με Εξελικτικό Αλγόριθμο

Η χρήση της τεχνικής Global Competitive Ranking αν εφαρμοστεί σε προβλήματα Εξελικτικών Στρατηγικών (ES) έχει αποδειχθεί ότι μπορεί να πετύχει σημαντικά καλύτερα αποτελέσματα από άλλες, ακόμα και πιο πολύπλοκες, τεχνικές.



Ας υποθέσουμε εξελικτικό αλγόριθμο Εξελικτικών Στρατηγικών (ES), ο οποίος ακολουθεί τις μεθόδους που αναλύσαμε στις υπο-παραγράφους 3.2.1 και 3.2.2 με πληθυσμό σωματιδίων  $\lambda$  και πληθυσμό γονέων  $\mu$  σε κάθε γενιά. Κάθε σωματίδιο  $i$  θα χαρακτηρίζεται από ένα ζεύγος πραγματικών τιμών που αποτελούν ένα διάνυσμα,  $(x_i, \sigma_i) \forall i \in \{1, \dots, \lambda\}$  (ο υπολογισμός του  $\sigma_i$  θα επεξηγηθεί στην συνέχεια).

Αρχικά δίνουμε στον αρχικό πληθυσμό μας  $\lambda$  σωματιδίων τυχαίες τιμές για κάθε μία από τις  $n$  μεταβλητές που περιγράφουν το διάνυσμα-σωματίδιο. Οι τυχαίες τιμές πρέπει να βρίσκονται μέσα στο διάστημα αναζήτησης  $S$  και προκύπτουν από την ομοιόμορφη κατανομή μέσα στο διάστημα αυτό. Μετά από την αρχική –τυχαία– παραγωγή της πρώτης γενιάς σωματιδίων, ο αλγόριθμος απαιτεί την δημιουργία του μέσου βήματος, το οποίο για την πρώτη γενιά υπολογίζεται, σύμφωνα με την βιβλιογραφία (Schwefel, 1995):

$$\sigma_{i,j}^{(0)} = \frac{\delta x_j}{\sqrt{n}} = \frac{\bar{x}_j - x_j}{\sqrt{n}}, i \in \{1, \dots, \lambda\}, j \in \{1, \dots, n\} \quad (3.9)$$

Όπου  $\sigma_{i,j}$  είναι το  $j$ -οστό στοιχείο του διανύσματος  $\sigma_i$ .

Ύστερα από την ταξινόμηση των σωματιδίων, όπως αυτή περιγράφηκε στην υπο-παραγράφο 3.2.2, τα  $\mu$  από τα  $\lambda$  καλύτερα σωματίδια επιλέγονται για την επόμενη γενιά. Ο λόγος αποκοπής έχει υπολογιστεί από την βιβλιογραφία (Bäck, 1996) να είναι  $\frac{\mu}{\lambda} = \frac{1}{7}$  και επιλέξαμε να τον κρατήσουμε ίδιο στις δοκιμές που πραγματοποιήσαμε και να μην δοκιμάσουμε διαφορετικές τιμές.

Το επόμενο βήμα είναι η πραγματοποίηση ανα-συνδυασμού στις αυτοπροσαρμοστικές παραμέτρους. Συγκεκριμένα, δημιουργείται ένα νέο  $\sigma$ , το οποίο είναι ο μέσος όρος δύο «γονέων». Ο ένας γονέας επιλέγεται με βάση την σειρά κατάταξης του, ενώ ο άλλος επιλέγεται τυχαία ανάμεσα στο σύνολο των γονέων (Schwefel, 1995). Έχουμε, δηλαδή, για την γενιά  $g$ :

$$\hat{\sigma}_{h,j}^{(g)} = (\sigma_{i,j}^{(g)} + \sigma_{k,j}^{(g)})/2, k_j \in \{1, \dots, \mu\} \quad (3.10)$$

Όπου  $k_j$  είναι ένα νούμερο, το οποίο αντιστοιχεί σε ένα σωματίδιο  $i$ , από εκείνα, τα οποία έχουν επιλεγεί ως γονείς, και παράγεται τυχαία για κάθε  $j$ .

Για να υπολογίσουμε τις νέες τιμές των διανυσμάτων θέλουμε να υπολογίσουμε την μεταβολή την οποία θα υποστούν οι προηγούμενες γενιές-γονείς προκειμένου να παραχθούν οι γενιές-απόγονοι. Για τον σκοπό αυτό θα χρειαστεί να ορίσουμε ορισμένες μεταβλητές. Αρχικά θα ορίσουμε τους ρυθμούς εκμάθησης  $\tau = \frac{\varphi^*}{\sqrt{2\sqrt{n}}}$  και  $\tau' = \frac{\varphi^*}{\sqrt{2n}}$  όπου  $\varphi^*$  είναι ο αναμενόμενος ρυθμός σύγκλισης (Schwefel, 1995) και τον θέτουμε  $\varphi^* = 1$  (Bäck, 1996). Ακόμη, θα ορίσουμε τον όρο  $N(0,1)$  που αποτελεί μια κανονική κατανομή μιας τυχαίας μονοδιάστατης μεταβλητής με μέση τιμή 0 και συντελεστή απόκλισης 1 και είναι ίδιος για όλες τις μεταβλητές του διανύσματος. Αντίστοιχα, ορίζουμε τον όρο  $N_j(0,1)$  που αποτελεί μια κανονική κατανομή μιας τυχαίας μονοδιάστατης μεταβλητής με μέση τιμή 0 και συντελεστή απόκλισης 1 αλλά ορίζεται εκ νέου για κάθε μεταβλητή του διανύσματος. Το μέσο βήμα γίνεται πλέον:

$$\sigma_{h,j}^{(g+1)} = \hat{\sigma}_{h,j}^{(g)} * e^{\tau'N(0,1) + \tau N_j(0,1)},$$

$$i \in \{1, \dots, \mu\}, h \in \{1, \dots, \lambda\}, j \in \{1, \dots, n\} \quad (3.11)$$

Έχοντας εισάγει το στοιχείο της τυχειότητας στις παραμέτρους στρατηγικής, κάθε σωματίδιο  $(x_i, \sigma_i) \forall i \in \{1, \dots, \mu\}$  δημιουργεί  $\frac{\lambda}{\mu}$  (= 7 στην περίπτωσή μας) σωματίδια, εισάγοντας εκ νέου στοιχείο τυχειότητας  $N_j(0,1)$  όπως και νωρίτερα, έτσι ώστε να δημιουργηθούν  $\lambda$  απόγονοι:

$$x_{h,j}^{(g+1)} = x_{i,j}^{(g)} + \sigma_{h,j}^{(g+1)} N_j(0,1) \quad (3.12)$$

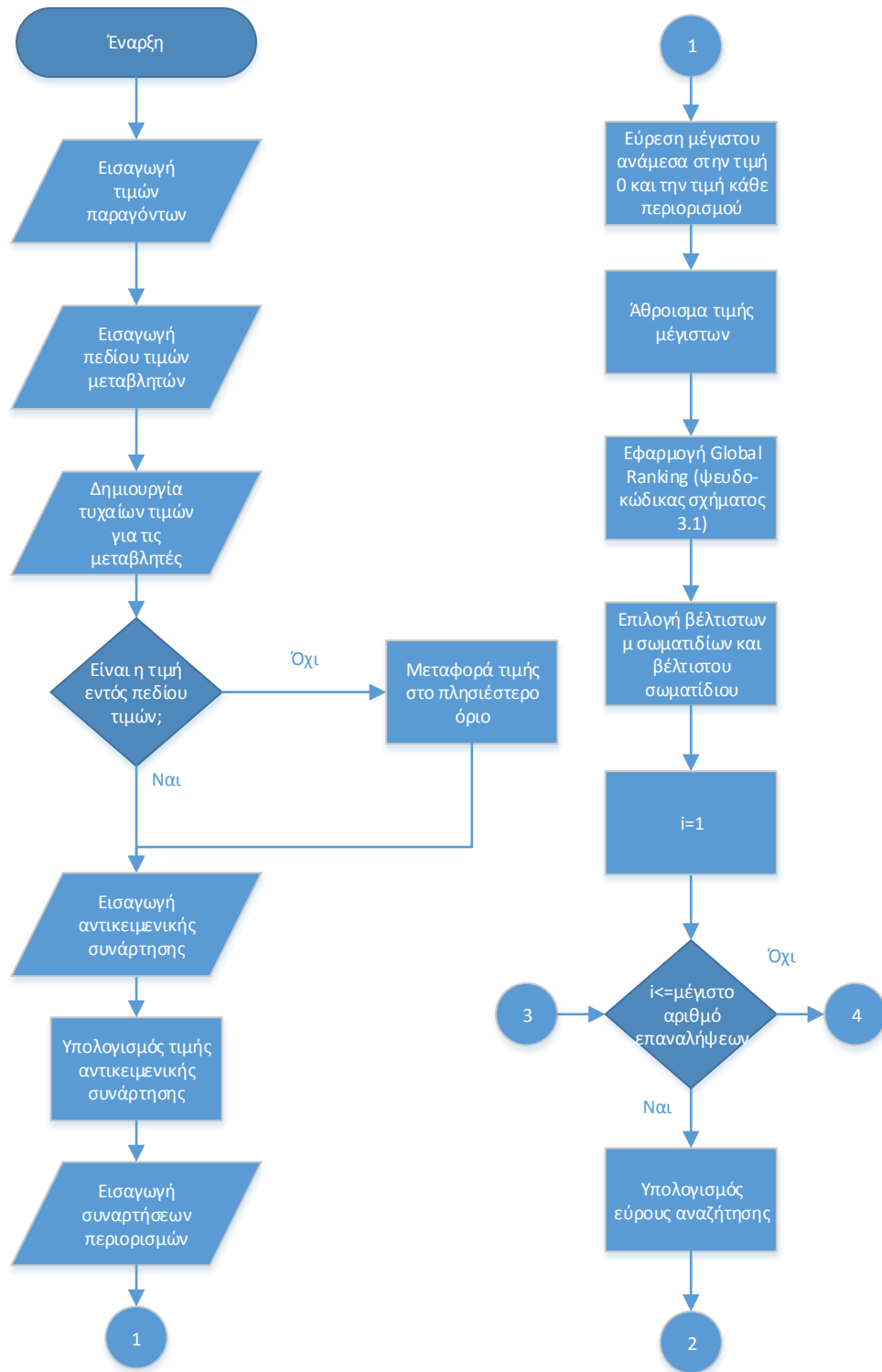
Αξίζει να σημειωθεί ότι εάν ένα σωματίδιο παραχθεί εκτός των ορίων που θέτει το πρόβλημα, τότε η τιμή της παραμέτρου αυτής θα μεταφερθεί πάνω στο όριο (είτε το θετικό, είτε το αρνητικό).

### 3.2.4. Μοντελοποίηση του αλγορίθμου σε λογισμικό MATLAB ®

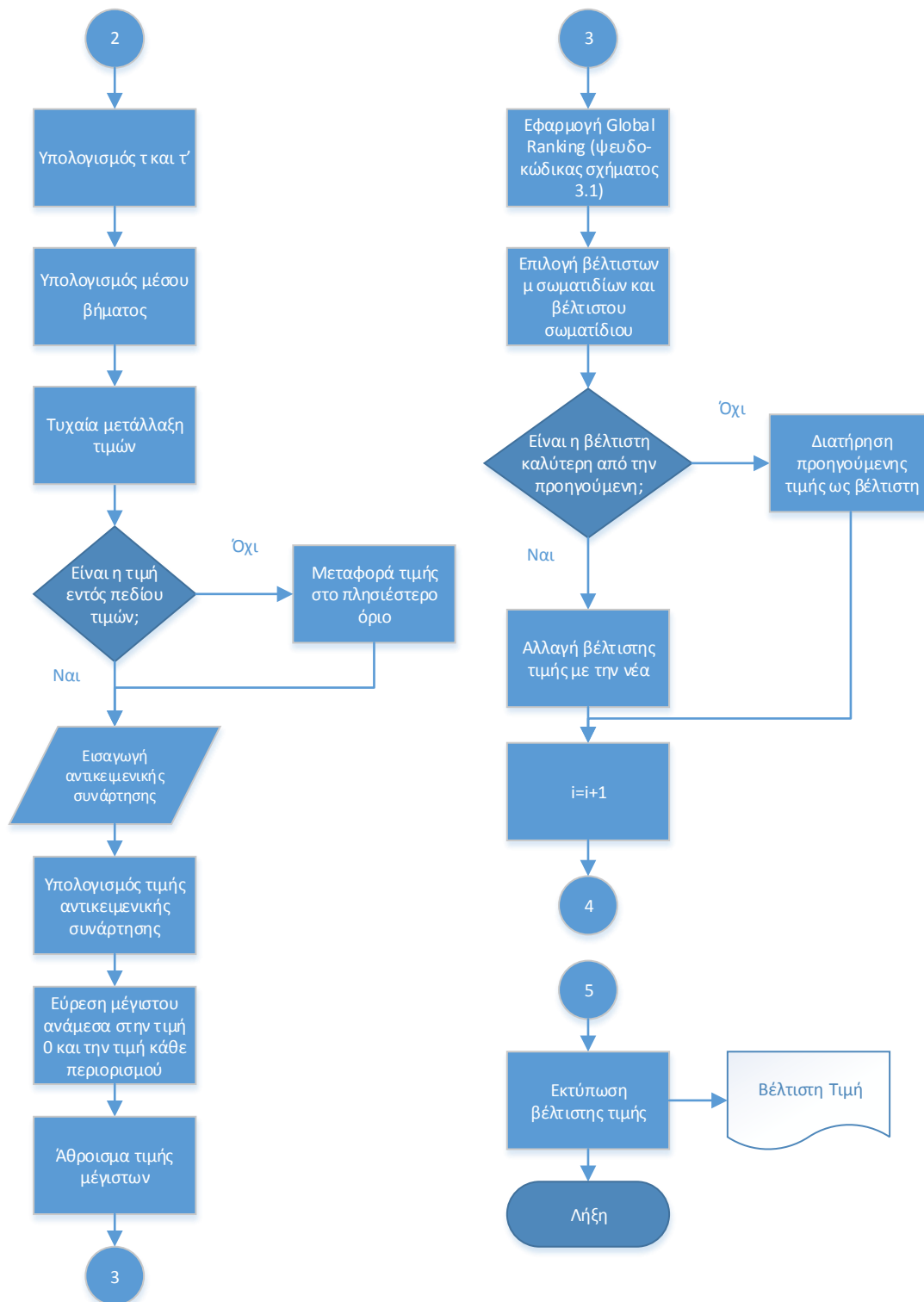
Θεωρούμε χρήσιμο να παραθέσουμε τα βήματα του αλγορίθμου που περιγράψαμε προηγουμένως, την οποία πραγματοποιήσαμε σε λογισμικό MATLAB ®, προκειμένου να γίνει ξεκάθαρος ο σκοπός και οι στόχοι για τους οποίους εκπονήθηκε η συγκεκριμένη διπλωματική, αλλά και να γίνουν κατανοητοί οι παράγοντες, τους οποίους μελετήσαμε. Η διαδικασία φαίνεται αναλυτικά και στο διάγραμμα ροής του σχήματος 3.2.

Αρχικά θεωρούμε ένα συγκεκριμένο αριθμό επαναλήψεων, ένα συγκεκριμένο αριθμό σωματιδίων  $\mu$ , έναν αριθμό σωματιδίων  $\lambda$ , ο οποίος υπολογίζεται με την χρήση του  $\mu$  και του λόγου  $\frac{\mu}{\lambda} = \frac{1}{7}$ . Αφού ορίσουμε τα άνω και κάτω όρια του πεδίου τιμών των μεταβλητών, δημιουργούμε τις αρχικές τιμές για το πρόβλημα και ελέγχουμε εάν αυτές βρίσκονται εκτός των ορίων. Η διαδικασία αυτή ονομάζεται αρχικοποίηση. Στην συνέχεια εφαρμόζουμε τις τιμές των μεταβλητών που υπολογίσαμε για τον υπολογισμό της τιμής της αντικειμενικής μας συνάρτησης καθώς και καθενός από τους περιορισμούς και αθροίζουμε προκειμένου να βρούμε το σύνολό τους. Προκειμένου να επιτύχουμε να κατατάξουμε τα σωματίδια, χρησιμοποιούμε την μέθοδο του Global Competitive Ranking όπως περιγράφηκε στην υπο-παράγραφο 3.2.2 και βρίσκουμε τα βέλτιστα  $\mu$  σωματίδια από τον συνολικό πληθυσμό  $\lambda$  σωματιδίων καθώς και το βέλτιστο σωματίδιο. Στην συνέχεια υπολογίζουμε το αρχικό  $\sigma_{i,j}^{(0)}$  όπως αυτό περιγράφεται από την εξίσωση (3.9) και φροντίζουμε να κρατήσουμε τις τιμές των  $\mu$  βέλτιστων σωματιδίων. Η αρχικοποίηση έχει ολοκληρωθεί και πλέον όλες οι διαδικασίες που ακολουθούν αποτελούν τμήμα της επαναληπτικής διαδικασίας που έχουμε περιγράψει.

Σε κάθε επανάληψη υπολογίζουμε τα  $\tau$  και  $\tau'$  και με τη χρήση ενός πληθυσμού τυχαίων σωματιδίων-γονέων βρίσκουμε την τιμή του νέου πληθυσμού, όπως έχουμε δείξει στις εξισώσεις (3.10), (3.11) και (3.12). Είναι πολύ σημαντικό να ελέγξουμε για τον νέο πληθυσμό ότι οι τιμές βρίσκονται εντός των άνω και κάτω ορίων για κάθε μεταβλητή του διανύσματος ακριβώς όπως συνέβη και στην αρχικοποίηση. Αν κάποια τιμή αποδειχθεί εκτός ορίων τότε μεταφέρεται στο πλησιέστερο όριο (άνω όριο ή κάτω όριο). Στην συνέχεια υπολογίζουμε την τιμή της αντικειμενικής συνάρτησης, την τιμή του κάθε περιορισμού αλλά και το άθροισμά τους όπως και νωρίτερα. Αφού εφαρμόσουμε την



**Σχήμα 3.2(α):** Πρώτο τμήμα διαγράμματος ροής μοντελοποιημένου αλγορίθμου GCR



**Σχήμα 3.2(β):** Δεύτερο τμήμα διαγράμματος ροής μοντελοποιημένου αλγορίθμου GCR

μέθοδο Global Competitive Ranking στα σωματίδια, κρατάμε τα μ καλύτερα ως γονείς για την επόμενη γενιά αλλά και το βέλτιστο σωματίδιο.

Τέλος πραγματοποιούμε έλεγχο προκειμένου, εάν η βέλτιστη τιμή που έχει προκύψει έως τώρα για την αντικειμενική συνάρτηση είναι χαμηλότερη από τη βέλτιστη που προέκυψε στην επανάληψη στην οποία βρισκόμαστε, η τιμή της αντικειμενικής συνάρτησης, της συνάρτησης ποινής, ο αριθμός της επανάληψης αλλά και τα στοιχεία που αποτελούν το σωματίδιο να αποθηκευτούν. Όπως είναι προφανές, η διαδικασία αυτή επαναλαμβάνεται για όσες επαναλήψεις έχουμε δώσει εμείς ως εντολή (το κριτήριο τερματισμού της, δηλαδή, είναι ο αριθμός των επαναλήψεων).

### 3.2.5. Σκοπός μελέτης της μεθόδου Global Competitive Ranking

Παρά τις εκτενείς βιβλιογραφικές αναφορές που υπάρχουν, τόσο για τους εξελικτικούς αλγορίθμους όσο και για την μέθοδο Global Competitive Ranking, παρατηρήθηκε ένα επιστημονικό κενό όσον αφορά τις τιμές, τις οποίες ο προγραμματιστής-μελετητής πρέπει να αποδώσει σε ορισμένους παράγοντες του προβλήματος, το οποίο μελετά. Μέσα από την εκπόνηση της διπλωματικής εργασίας αυτής, έγινε μια προσπάθεια να κατανοηθεί ο τρόπος με τον οποίο οι διάφοροι παράγοντες αυτοί μπορούν να επηρεάσουν την σύγκλιση στο επιθυμητό αποτέλεσμα, την τιμή, η οποία θα ευρεθεί, αλλά και τον υπολογιστικό χρόνο, τον οποίο απαιτεί το λογισμικό MATLAB®, για να εκτελέσει τις επαναλήψεις αυτές. Εκτενής αναφορά στα προβλήματα, τα οποία επιλύθηκαν, στους παράγοντες, τους οποίους μελετήσαμε, αλλά και στην γενικότερη μεθοδολογία που ακολουθήθηκε γίνεται στα επόμενα κεφάλαια.

## **3.3. Η μέθοδος Self-Learning Particle Swarm Optimizer for Global Optimization Problems**

### 3.3.1. Μαθηματικό μοντέλο ελαχιστοποίησης

Τα προβλήματα, τα οποία μελετήσαμε, υπόκεινται στις εξισώσεις (3.1), (3.2), (3.3) και (3.5) ακριβώς όπως αυτά έχουν περιγραφεί στην παράγραφο 3.2.1. Ο αναγνώστης μπορεί να ανατρέξει στην παράγραφο αυτή προκειμένου να αντλήσει οποιαδήποτε πληροφορία χρειάζεται για την μοντελοποίηση προβλημάτων μεγιστοποίησης ή ελαχιστοποίησης. Όπως και στην GCR, έτσι και στην χρήση της μεθόδου Self-Learning Particle Swarm

Optimizer (SLPSO) τα προβλήματα, τα οποία μελετήσαμε, είτε είναι ελαχιστοποίησης είτε μετατρέπονται από προβλήματα μεγιστοποίησης σε προβλήματα ελαχιστοποίησης μέσα από την εφαρμογή της εξίσωσης (3.5).

### 3.3.2. Μαθηματικό μοντέλο Particle Swarm Optimization

Παρότι η μέθοδος Particle Swarm Optimization (PSO) έχει αναφερθεί στην παράγραφο 1.2, είχε παραληφθεί η περιγραφή του μαθηματικού μοντέλου με βάση, την οποία αυτή λειτουργεί. Επειδή η μέθοδος SLPSO, την οποία μελετήσαμε, είναι βασισμένη στην μέθοδο PSO, αφού αποτελεί εξέλιξή της, κατά μια έννοια, θεωρείται σκόπιμη η ανάλυση του μοντέλου με βάση το οποίο η PSO λειτουργεί.

Θεωρούμε ένα σωματίδιο  $k$  από ένα πληθυσμό σωματιδίων, το οποίο αντιπροσωπεύεται από ένα διάνυσμα θέσης  $\vec{x}_k$  καθώς και ένα διάνυσμα ταχύτητας  $\vec{v}_k$ , όπου ταχύτητα θεωρούμε το διάνυσμα της μετακίνησης του σωματιδίου από την προηγούμενη θέση στην επόμενη. Τα δύο διανύσματα αυτά, με βάση την βιβλιογραφία (Kennedy & Eberhart, 1995) (Kennedy, 1997) υφίστανται την ακόλουθη μεταβολή από επανάληψη σε επανάληψη:

$$v'_k{}^d = \omega v_k{}^d + \eta_1 r_1 (x_{pbest_i}{}^d - x_k{}^d) + \eta_2 r_2 (x_{gbest_i}{}^d - x_k{}^d) \quad (3.13)$$

$$x'_k{}^d = x_k{}^d + v'_k{}^d \quad (3.14)$$

Όπου τα  $x'_k{}^d$  και  $x_k{}^d$  είναι η τωρινή και η προηγούμενη θέση της  $d$ -οστής διάστασης του σωματιδίου  $k$  αντίστοιχα, τα  $v'_k{}^d$  και  $v_k{}^d$  είναι η τωρινή και η προηγούμενη ταχύτητα της  $d$ -οστής διάστασης του σωματιδίου  $k$  αντίστοιχα, τα  $\vec{x}_{pbest}$  και  $\vec{x}_{gbest}$  η βέλτιστη θέση, την οποία έχει βρει έως τώρα το σωματίδιο  $k$  και το σύνολο του σμήνους σωματιδίων έως τώρα, το  $\omega \in (0,1)$  είναι ο τελεστής αδράνειας, ο οποίος καθορίζει κατά πόσο η προηγούμενη ταχύτητα διατηρείται, τα  $\eta_1$  και  $\eta_2$  είναι οι ρυθμοί εκμάθησης που αναφέρονται στις συνιστώσες επίγνωσης (cognition) και κοινωνικής συναναστροφής (social) αντίστοιχα και τα  $r_1$  και  $r_2$  είναι τυχαίοι αριθμοί, οι οποίοι παράγονται στο διάστημα  $[0,1]$  με την χρήση της ομοιόμορφης κατανομής.

Γενικότερα, υπάρχουν δύο βασικά μοντέλα του αλγορίθμου PSO, το μοντέλο global best (gbest) και το μοντέλο local best (lbest), τα οποία διαφέρουν στον τρόπο με τον οποίο ορίζεται η γειτονιά του κάθε σωματιδίου και άρα ποια σωματίδια μοιράζονται μεταξύ τους

πληροφορίες. Όπως είναι προφανές, στο μοντέλο global best η γειτονιά του σωματιδίου αποτελείται από το σύνολο των σωματιδίων ενώ στο μοντέλο local best η γειτονιά του σωματιδίου καθορίζεται από αρκετά συγκεκριμένα σωματίδια. Η συμπεριφορά του κάθε μοντέλου είναι διαφορετική στην επίλυση προβλημάτων. Το global best, σύμφωνα με την βιβλιογραφία (Kennedy & Eberhart, 2001), (Poli et al., 2007) επιτυγχάνει ταχύτερη σύγκλιση αλλά και μεγαλύτερη πιθανότητα να εγκλωβιστεί σε τοπικά ακρότατα. Αντίθετα, το local best είναι λιγότερο πιθανό να εγκλωβιστεί σε τοπικά ακρότατα αλλά η σύγκλιση του καθυστερεί (συγκριτικά).

### 3.3.3. Μαθηματικό μοντέλο Self-Learning Particle Swarm Optimizer

Η μέθοδος SLPSO αναπτύχθηκε πρόσφατα (Changhe & Shengxiang, 2012). Ένα βασικό χαρακτηριστικό της είναι η χρήση των 4 εξελικτικών καταστάσεων, οι οποίες αναφέρθηκαν για πρώτη φορά στην μέθοδο APSO (Zhan et al., 2009) και εφαρμόστηκαν με επιτυχία και στην μέθοδο ALPSO (Li & S. Yang, 2009). Οι 4 εξελικτικές καταστάσεις, στις οποίες αναφέρεται η βιβλιογραφία, είναι η «εκμετάλλευση», η «εξερεύνηση», η «σύγκλιση» και η «αναπήδηση». Εξαιτίας της αποδοτικότητας που έδειξε ο διαχωρισμός των καταστάσεων αυτών στις μεθόδους που αναπτύχθηκαν, αποφασίστηκε η χρήση τους στην ανάπτυξη του SLPSO.

Προκειμένου ένας αλγόριθμος PSO να επιτύχει την καλύτερη δυνατή απόδοση, είναι απαραίτητο να ευρεθεί η ισορροπία ανάμεσα στις μεθόδους local best και global best που αναπτύξαμε εκτενώς στην προηγούμενη υπο-παράγραφο (3.3.2). Αυτό, όμως, δεν είναι εύκολο, καθώς εάν ένα σωματίδιο εκπαιδευτεί ταυτόχρονα από την pbest θέση του αλλά και την gbest ή lbest θέση του, μπορεί να υποκύψει στα προβλήματα που διέπουν και τα 2 μοντέλα. Έτσι, μια πρόταση είναι η ξεχωριστή μελέτη των συνιστωσών επίγνωσης και κοινωνικής συναναστροφής για κάθε σωματίδιο. Η μελέτη αυτή επικεντρώνεται είτε στην «εκμετάλλευση» (pbest) είτε στην «σύγκλιση» (gbest) και δίνει, στην ουσία, διαφορετικούς ρόλους σε διαφορετικά σωματίδια κατά την διαδικασία αναζήτησης. Εξαιτίας της δυσκολίας επιλογής, σε κάθε περίπτωση, προτείνεται μια προσαρμοστική μέθοδος, η οποία υπολογίζει αυτόματα την στρατηγική αναζήτησης.



Αυτή η λογική κατανομής εργασίας, έδωσε την ιδέα να δοθεί σε κάθε σωματίδιο 1 από 4 πιθανούς ρόλους, που αποτελούν ένα βασικό χαρακτηριστικό της μεθόδου SLPSO και βασίζονται στις 4 εξελικτικές καταστάσεις που αναλύθηκαν προηγουμένως:

- (a) Εκμετάλλευση της βέλτιστης ατομικής θέσης του σωματιδίου. Επιτρέπει την εκμάθηση με βάση την δική του βέλτιστη θέση.
- (b) Αναπήδηση μακριά από τοπικά ακρότατα. Επιτρέπει -μέσα από μετάλλαξη- τον απεγκλωβισμό σωματιδίων που έχουν συγκλίνει σε μια συγκεκριμένη περιοχή.
- (c) Εξερεύνηση νέων περιοχών. Επιτρέπει την εκμάθηση μέσα από άλλα σωματίδια, ιδίως σε χώρους με μεγάλο πλήθος ισο-καταναμημένων τοπικών ακροτάτων.
- (d) Σύγκλιση στο παγκόσμιο βέλτιστο σωματίδιο. Επιτρέπει την αποτελεσματική αναζήτηση σε χώρους με μακρινές, μεταξύ τους, κορυφές.

Έτσι, με βάση τις 4 κατηγορίες αυτές, υπάρχουν 4 πηγές γνώσης για κάθε σωματίδιο:

- (a) Η βέλτιστη τιμή το σωματιδίου ως τώρα  $pbest$
- (b) Μια τυχαία θέση  $p_{rand}$  κοντά στο σωματίδιο
- (c) Η βέλτιστη τιμή ενός τυχαίου σωματιδίου  $pbest_{rand}$  με καλύτερη τιμή από την βέλτιστη του ίδιου του σωματιδίου
- (d) Η αποθηκευμένη τιμή του βέλτιστου  $abest$  σωματιδίου (εξήγηση παρακάτω)

Για κάθε σωματίδιο  $k$  οι σχέσεις, οι οποίες αντιστοιχούν σε κάθε έναν από τους παράγοντες (a),(b),(c) και (d), είναι οι ακόλουθες:

- Παράγοντας (a):

$$v'_k{}^d = \omega v_k{}^d + \eta r_k{}^d (pbest_k{}^d - x_k{}^d) \quad (3.15)$$

- Παράγοντας (b):

$$x_k{}^d = x_k{}^d + v_{avg}{}^d N(0,1) \quad (3.16)$$

- Παράγοντας (c):

$$v'_k{}^d = \omega v_k{}^d + \eta r_k{}^d (pbest_{rand}{}^d - x_k{}^d) \quad (3.17)$$

- Παράγοντας (d):

$$v'_k{}^d = \omega v_k{}^d + \eta r_k{}^d (abest_k{}^d - x_k{}^d) \quad (3.18)$$

Όπου  $v_{avg}^d$  η μέση ταχύτητα όλων των σωματιδίων στην d-οστή διάσταση και  $N(0,1)$  ένας τυχαίος αριθμός, ο οποίος παράγεται από την κανονική κατανομή με μέση τιμή 0 και τυπική απόκλιση 1.

Ιδιαίτερη προσοχή πρέπει να δοθεί στο σωματίδιο *abest* αφού είναι ίδιο με το *gbest* μόνο στην πρώτη επανάληψη, ενώ στην συνέχεια γίνεται καλύτερο από το σωματίδιο *gbest* με βάση τη διαδικασία που θα εξηγήσουμε στην συνέχεια.

Ο ψευδο-κώδικας της διαδικασίας που περιγράψαμε αποτυπώνεται στο σχήμα 3.3.

---

**Algorithm 1** Update(operator *i*, particle *k*, *fes*)

---

```

1: if i = a then
2:   Update the velocity and position of particle k using
   operator a and (3.14);
3: else if i = b then
4:   Update the position of particle k using operator b;
5: else if i = c then
6:   Choose a random particle j that is not particle k;
7:   if  $f(\vec{x}_{pbest_j}) < f(\vec{x}_{pbest_k})$  then
8:     Update the velocity and position of particle k using
   operator c and (3.14);
9:   else
10:    Update the velocity and position of particle j using
   operator c and (3.14);
11:    k := j;
12:   end if
13: else
14:   Update the velocity and position of particle k using
   operator d and (3.14);
15: end if
16: fes++; where fes is the current number of fitness
   evaluations.

```

---

### Σχήμα 3.3: Ψευδο-κώδικας εξισώσεων εκμάθησης SLPSO

Κάθε ένας από τους παράγοντες της SLPSO αποτελεί μια διαφορετική τοπολογία (ένα είδος γενικευμένης γεωμετρίας) που χαρακτηρίζει έναν πληθυσμό, ορίζοντας την γειτονιά του και την δομή της επικοινωνίας των σωματιδίων. Αλλάζοντας τον παράγοντα, αλλάζουμε και τη μέθοδο που κάποιο σωματίδιο επικοινωνεί με κάποιο άλλο.

Ο σκελετός της μεθόδου με τον οποίο οι παράγοντες προσαρμόζονται από την μία γενιά στην επόμενη βασίζεται στην υπόθεση ότι ο πιο επιτυχημένος παράγοντας που χρησιμοποιήθηκε στις προηγούμενες επαναλήψεις μπορεί να είναι επιτυχημένος και σε μερικές επόμενες επαναλήψεις.

Αρχικά, όλοι οι παράγοντες έχουν ίση πιθανότητα  $\frac{1}{4}$  για κάθε σωματίδιο και ανανεώνονται με βάση την σχετική απόδοσή τους. Για κάθε σωματίδιο επιλέγεται ένας παράγοντας με βάση την «αναλογία επιλογής». Ο παράγοντας, ο οποίος έχει υψηλότερη σχετική απόδοση, η οποία επηρεάζεται από ένα συνδυασμό της τιμής της αντικειμενικής, της τωρινής αναλογίας επιτυχίας και της προηγούμενης αναλογίας επιλογής, οδηγείται σε αύξηση της αναλογίας επιλογής του. Σταδιακά, ο καταλληλότερος παράγοντας θα επιλεγεί αυτόματα και θα ελέγξει την συμπεριφορά εκμάθησης του κάθε σωματιδίου σε διάφορα εξελικτικά στάδια.

Για κάθε παράγοντα  $i$  υπολογίζουμε τόσο την «τιμή προόδου» ( $p_i^k(t)$ ) όσο και την «τιμή επιβράβευσης» ( $r_i^k(t)$ ). Η τιμή προόδου υπολογίζεται από την σχέση:

$$p_i^k(t) = \begin{cases} |f(\bar{x}_k(t)) - f(\bar{x}_k(t-1))|, & \text{εάν ο } i \text{ επιλέχθηκε} \\ 0, & \text{αλλιώς} \end{cases} \quad (3.19)$$

Η τιμή επιβράβευσης αποτελείται από 3, κατά σειρά, μέρη: 1) την κανονικοποιημένη τιμή προόδου, 2) την αναλογία επιτυχίας και 3) την προηγούμενη αναλογία επιλογής, όπως αυτά φαίνονται και από την σχέση:

$$r_i^k(t) = \frac{p_i^k(t)}{\sum_{j=1}^R p_j^k(t)} + \frac{g_i^k}{\sigma_i^k} (1 - a) + c_i^k s_i^k(t) \quad (3.20)$$

Όπου  $g_i^k$  ο μετρητής επιτυχημένης εκμάθησης του σωματιδίου  $k$  (πόσες φορές δηλαδή ο απόγονος έχει καλύτερη τιμή αντικειμενικής από τον γονέα) όταν εφαρμόσουμε τον παράγοντα  $i$ , μετρώντας από την πιο πρόσφατη αναβάθμιση της αναλογίας επιλογής,  $G_i^k$  το συνολικό νούμερο επαναλήψεων κατά τις οποίες ο  $i$  επελέγη από το  $k$ , μετρώντας από την πιο πρόσφατη αναβάθμιση της αναλογίας επιλογής. Από τα προηγούμενα είναι εμφανές ότι η αναλογία ανάμεσα στα 2 είναι η αναλογία επιτυχίας του  $i$  στο  $k$ . Το  $a$  είναι ένα τυχαίο βάρος στο  $[0,1]$ ,  $R$  ο αριθμός των συντελεστών (εδώ 4),  $s_i^k(t)$  η αναλογία επιλογής του  $i$  για το  $k$  στην τωρινή επανάληψη και  $c_i^k$  είναι ένας συντελεστής ποινής του  $i$  στο  $k$ , ο οποίος υπολογίζεται από την σχέση:

$$c_i^k = \begin{cases} 0.9, & \text{εάν } g_i^k = 0 \text{ και } s_i^k(t) = \max_{j=1}^R (s_j^k(t)) \\ 1, & \text{αλλιώς} \end{cases} \quad (3.21)$$

Στην εξίσωση (3.20), στην περίπτωση που κανένας παράγοντας δεν έχει καταφέρει να βελτιώσει το σωματίδιο  $k$  από την πιο πρόσφατη αναβάθμιση της αναλογίας επιλογής, τότε το  $\sum_{j=1}^R p_j^k(t)$  θα είναι 0 και άρα μόνο ο παράγοντας  $c_i^k s_i^k(t)$  θα προσμετράται στον υπολογισμό του  $r_i^k(t)$ .

Βασισμένοι στα προηγούμενα, η αναλογία επιλογής του  $i$  για το σωματίδιο  $k$  στην επόμενη επανάληψη ( $t+1$ ) ανανεώνεται ως εξής:

$$s_i^k(t+1) = \frac{r_i^k(t)}{\sum_{j=1}^R r_j^k(t)} (1 - R * \gamma) + \gamma \quad (3.22)$$

Όπου  $\gamma$  είναι η ελάχιστη αναλογία επιλογής του κάθε παράγοντα και θα λαμβάνει την τιμή 0.01 σε όλες τις δοκιμές που πραγματοποιούμε.

Παρότι οι παραπάνω σχέσεις μας εξασφαλίζουν ότι υπάρχει ένας πιο επιθυμητός παράγοντας σε σχέση με τους άλλους, σε κάθε στιγμή, όταν αυτό το σωματίδιο συγκλίνει ή μετατοπίζεται σε μια νέα περιοχή με διαφορετικές ιδιότητες από την προηγούμενη, αυτός ο παράγοντας μπορεί πλέον να μην βελτιώνει το σωματίδιο. Με βάση όμως την σχέση (3.21), η αναλογία επιλογής αυτού του παράγοντα θα μειωθεί, ενώ θα αυξηθεί κάποιου άλλου, οδηγώντας στην επιλογή ενός άλλου καταλληλότερου. Έτσι λειτουργεί ο μηχανισμός προσαρμοστικότητας της SLPSO.

Προκειμένου να μειωθεί η πιθανότητα να τιμωρηθεί ο βέλτιστος παράγοντας επειδή μπορεί να έχει για ένα σύντομο διάστημα χαμηλή απόδοση, όλες οι αναλογίες επιλογής των παραγόντων ανανεώνονται ταυτόχρονα και όχι σε κάθε επανάληψη. Αντί να μετράμε τον αριθμό συνεχόμενων επιτυχημένων επαναλήψεων για την συχνότητα αναβάθμισης  $U_f$ , επιλέγουμε να μετρήσουμε τον αριθμό συνεχόμενων επαναλήψεων αποτυχημένης εκμάθησης ( $m_k$ ) για κάθε  $k$ . Εάν κάποιος παράγοντας ανανεωθεί πριν προσεγγίσουμε την μέγιστη τιμή των  $U_f$ , το  $m_k$  θα τεθεί 0.

Στους περισσότερους αλγόριθμους που βασίζονται σε πληθυσμούς, όταν ένα σωματίδιο ανανεωθεί ή αντικατασταθεί, όλες του οι διαστάσεις αντικαθίστανται. Αυτή η αναβάθμιση οδηγεί πολλές φορές στην απώλεια χρήσιμων πληροφοριών. Αφενός, δηλαδή, η συνολική αντικειμενική βελτιώνεται, αφετέρου, όμως, μια συγκεκριμένη διάσταση μπορεί να χειροτερέψει. Για τον λόγο αυτό, στην SLPSO χρησιμοποιείται η θέση *abest*. Προκειμένου να αποφύγουμε διάφορα προβλήματα που μπορεί να προκύψουν από την βελτίωση

συγκεκριμένων μόνο συντεταγμένων του *abest*, χρησιμοποιούμε μια πιθανότητα εκμάθησης ( $P_l$ ) για κάθε διάσταση, την πιθανότητα δηλαδή το *abest* να βελτιωθεί από το σωματίδιο που μελετάμε.

Με την χρήση της  $P_l$  εξοικονομούμε υπολογιστικό χρόνο και μειώνεται η πιθανότητα η *abest* να μάθει άχρηστες πληροφορίες, κάτι που μπορεί να επηρεάσει σημαντικά την απόδοση της SLPSO προς το χειρότερο. Το σχήμα 3.4 παρουσιάζει τον ψευδο-κώδικα του αλγορίθμου αναβάθμισης της *abest*.

---

**Algorithm 2** UpdateAbest(particle  $k, fes$ )

---

```

1: for each dimension  $d$  of abest do
2:   if  $rand() < P_l^k$  then
3:      $\bar{x}_{t\_abest} := \bar{x}_{abest}$ ;
4:      $\bar{x}_{t\_abest}[d] := \bar{x}_k[d]$ ;
5:     Evaluate  $\bar{x}_{t\_abest}$ ;
6:      $fes++$ ;
7:     if  $f(\bar{x}_{t\_abest}) < f(\bar{x}_{abest})$  then
8:        $\bar{x}_{abest}[d] := \bar{x}_{t\_abest}[d]$ ;
9:     end if
10:  end if
11: end for

```

---

**Σχήμα 3.4:** Ψευδο-κώδικας αναβάθμισης *abest*

Η *abest* χρησιμοποιείται προκειμένου ο πληθυσμός να συγκλίνει πιο γρήγορα. Επειδή όμως η έλξη της *abest* θέσης είναι πολύ δυνατή για να δώσει ευκαιρία στους άλλους παράγοντες να λειτουργήσουν, θέτουμε όριο στον αριθμό των σωματιδίων που μαθαίνουν από αυτήν. Ο αριθμός αυτός ορίζεται από τις ιδιότητες του προβλήματος ενώ πρέπει να προσέξουμε το πως θα τον ορίσουμε, καθώς οι άλλοι 3 παράγοντες συνεισφέρουν στον παράγοντα της *abest*. Επειδή αυτή η απόφαση είναι δύσκολη, επιλέγουμε τυχαία έναν αριθμό σωματιδίων σε κάθε επανάληψη. Στην συνέχεια, ορίζουμε κάποιες πληροφορίες, οι οποίες πρέπει να μεταβληθούν σε κάποιο σωματίδιο που άλλαξε ανάμεσα στην χρήση και την μη-χρήση του παράγοντα σύγκλισης (*abest*). Οι πληροφορίες περιλαμβάνουν τις τιμές προόδου, τις τιμές επιβράβευσης, τις αναλογίες επιτυχίας και επιλογής.

Στην περίπτωση που ένα σωματίδιο χρησιμοποιούσε τον παράγοντα σύγκλισης στην προηγούμενη αλλά όχι στην τωρινή επανάληψη, αφαιρούμε την πηγή εκμάθησης της *abest* και επανα-κανονικοποιούμε τις αναλογίες επιλογής των άλλων 3 παραγόντων σύμφωνα με τις τωρινές τιμές, ενώ κρατάμε ίδιες όλες τις άλλες πληροφορίες τους. Στην αντίθετη

περίπτωση όλες οι πληροφορίες που αναφέραμε επανέρχονται στις αρχικές τους καταστάσεις. Ο ψευδο-κώδικας της μεθόδου φαίνεται στο σχήμα 3.5.

---

**Algorithm 3** UpdateLearningOpt(particle  $k$ )

---

```

1: if  $CF_k! = true \ \&\& \ PF_k = true$  then
2:    $sum := \sum_{j=1}^3 s_j^k$ ;
3:   for  $j := 1$  to 3 do
4:      $s_j^k := s_j^k / sum$ ;
5:   end for
6:    $s_4^k := 0$ ;
7: end if
8: if  $CF_k = true \ \&\& \ PF_k! = true$  then
9:   for  $j := 1$  to 4 do
10:     $p_j^k := 0$ ;  $g_j^k := 0$ ;  $G_j^k := 0$ ;  $s_j^k := 1/4$ 
11:   end for
12: end if

```

where  $CF_k$  and  $PF_k$  are used to record whether particle  $k$  uses the convergence operator or not at the current and previous iteration, respectively

---

**Σχήμα 3.5:** Ψευδο-κώδικας αλλαγής από abest σε άλλους παράγοντες και αντίστροφα

Για την χρήση της SLPSO χρειάζεται να ορίσουμε μερικές ακόμα παραμέτρους:

- Η παράμετρος  $V_{max}$  χρησιμοποιείται για να περιοριστεί η μέγιστη ταχύτητα κάθε σωματιδίου και είναι το μισό διάστημα αναζήτησης. Για την αντιμετώπιση αναζήτησης εκτός ορίων, πριν ανανεώσουμε το σωματίδιο, έχουμε την προηγούμενη θέση  $x^d(t-1)$ , υπολογίζουμε ένα προσωρινό  $x_t$  με βάση τον ψευδο-κώδικα του σχήματος 3.2 και έτσι βρίσκουμε το  $x^d(t)$  ως εξής:

$$x^d(t) = \begin{cases} R(x_{min}^d, x^d(t-1)), & \text{εάν } x_t < x_{min}^d \\ R(x^d(t-1), x_{max}^d), & \text{εάν } x_t > x_{max}^d \\ x_t, & \text{αλλιώς} \end{cases} \quad (3.23)$$

Όπου  $R(a,b)$  ένας ομοιόμορφα κατανομημένος αριθμός εντός του ορίου  $[a,b]$  ενώ ως  $[x_{min}^d, x_{max}^d]$  ορίζουμε το διάστημα αναζήτησης στην  $d$ -οστή διάσταση του προβλήματος.

- Η συχνότητα αναβάθμισης  $U_f$  για κάθε σωματίδιο  $k$  παίρνει τιμές ανάμεσα στο 1 και στο 10 και υπολογίζεται:

$$U_f^k = \max(10 * e^{-1.6(\frac{k}{N})^4}, 1) \quad (3.24)$$

Όπου  $N$  το μέγεθος του πληθυσμού.

- Η πιθανότητα εκμάθησης  $P_l$  για κάθε σωματίδιο  $k$  παίρνει τιμές ανάμεσα στο 0.05 και το 1 και υπολογίζεται:

$$P_i^k = \max(1 - e^{-1.6(\frac{k}{N})^4}, 0.05) \quad (3.25)$$

- Ο αριθμός των σωματιδίων που πρέπει να χρησιμοποιούν την σύγκλιση (abest):

$$M(fes) = N(1 - e^{-100(\frac{fes}{T_{Fes}})^3}) \quad (3.26)$$

Όπου  $T_{Fes}$  το σύνολο των υπολογισμών της αντικειμενικής που επιτρέπονται σε ένα τρέξιμο.

- Ο συντελεστής βαρύτητας  $\omega$  μειώνεται από 0.9 σε 0.4 με βάση την σχέση:

$$\omega(fes) = 0.9 - 0.5 * \frac{fes}{T_{Fes}} \quad (3.27)$$

- Το  $\eta$  ορίστηκε στα πειράματα στην τιμή 1.496

Η εφαρμογή του SLPSO αλγορίθμου φαίνεται στα σχήματα 3.6 και 3.7:

---

#### Algorithm 4 The SLPSO Algorithm

---

```

1: Generate initial swarm and set up parameters for each particle;
2: Set  $fes := 0$ , iteration counter for initial swarm  $t := 0$ ;
3: while  $fes < T_{Fes}$  do
4:   for each particle  $k$  do
5:     Select one learning operator  $i$  using the roulette wheel selection rule;
6:      $U_{pdate}(i, k, fes)$ ;
7:      $G_i^{k++}$ ;
8:     if  $f(\bar{x}_k(t)) < f(\bar{x}_k(t-1))$  then
9:        $g_i^{k++}$ ;  $m_k := 0$ ;
10:       $p_i^k += f(\bar{x}_k(t-1)) - f(\bar{x}_k(t))$ ;
11:      Perform  $U_{pdateAbest}(k, fes)$  for the abest position;
12:     else
13:        $m_k := m_k + 1$ ;
14:     end if
15:     if  $f(\bar{x}_k(t)) < f(\bar{x}_{pbest_k})$  then
16:        $\bar{x}_{pbest_k} := \bar{x}_k$ ;
17:       if  $f(\bar{x}_k) < f(\bar{x}_{abest})$  then
18:          $\bar{x}_{abest} := \bar{x}_k$ ;
19:       end if
20:     end if
21:     if  $m_k \geq U_f^k$  then
22:       Update the selection ratios according to Eq. (3.22)
23:       for each operator  $j$  do
24:          $p_j^k := 0$ ;  $g_j^k := 0$ ;  $G_j^k := 0$ ;
25:       end for
26:     end if
27:   end for
28:    $U_{pdatePar}()$ ;
29:    $t++$ ;
30: end while

```

---

**Σχήμα 3.6:** Ψευδο-κώδικας αλγορίθμου SLPSO

---

**Algorithm 5** UpdatePar()

---

- 1: Create a permutation of index number;
  - 2: Update  $U_f$  for each particle by (3.24);
  - 3: Update  $P_l$  for each particle by (3.25);
  - 4: Calculate the number of particles using the convergence operator by (3.26);
  - 5: Update related information of the four operators for each particle by Algorithm 3;
  - 6: Calculate the inertia weight  $\omega$  by (3.27);
- 

**Σχήμα 3.7:** Ψευδο-κώδικας υπο-ρουτίνας αναβάθμισης σωματίδιου στην SLPSO

### 3.3.4. Υπολογισμός ποινής

Μέχρι στιγμής έχουμε αναφερθεί στον τρόπο με τον οποίο ο κώδικας συμπεριφέρεται για την εύρεση της βέλτιστης λύσης του ενός προβλήματος, χωρίς όμως να έχουμε κάνει καμία αναφορά στον τρόπο με τον οποίο συνυπολογίζονται στην λύση αυτή οι περιορισμοί, στους οποίους τυχόν υπόκειται το πρόβλημα.

Η μέθοδος, την οποία θα ακολουθήσουμε για να εισάγουμε την ποινή στο πρόβλημά μας, είναι η εισαγωγή του συντελεστή ποινής λάμδα ( $\lambda$ ). Η μέθοδος αυτή είναι από τις απλούστερες αλλά και πιο διαδεδομένες μεθόδους εισαγωγής του συντελεστή ποινής στην αντικειμενική συνάρτηση ελαχιστοποίησης. Συγκεκριμένα, η αντικειμενική συνάρτηση μετασχηματίζεται:

$$f'(\vec{x}) = f(\vec{x}) + \lambda * g(x_k) \quad (3.28)$$

Όπου  $g(x_k)$  το άθροισμα των περιορισμών του σωματιδίου  $k$  όπως ορίστηκε στην σχέση (3.6). Συνήθως, η τιμή του συντελεστή ποινής  $\lambda$  υπολογίζεται μέσα από δοκιμές προκειμένου να βρεθεί η βέλτιστη τιμή του ή το βέλτιστο εύρος τιμών, για το οποίο η συνάρτηση  $f'(\vec{x})$  παίρνει την βέλτιστη δυνατή τιμή. Προκειμένου να αποφευχθούν άσκοπες δοκιμές σε ένα τεράστιο πλήθος υποψήφιων τιμών για τον συντελεστή  $\lambda$ , υπό την καθοδήγηση του επιβλέποντα καθηγητή, δοκιμάστηκε μια διαφορετική μέθοδος προκειμένου να προσεγγιστεί με μεγαλύτερη ευκολία η βέλτιστη τιμή του  $\lambda$ , η οποία και περιγράφεται στην συνέχεια.



Έστω ότι θεωρούμε την αντικειμενική συνάρτηση  $f$  καθώς και την συνάρτηση ποινής  $g$  ανάμεσα σε 2 διαδοχικές επαναλήψεις  $t$  και  $(t+1)$ , όπου το  $\lambda$  είναι σταθερό και η συνάρτηση  $f'$  είναι ίση και στις 2. Μπορούμε τότε να θεωρήσουμε:  $f'_t = f'_{t+1}$

$$f'_t = f'_{t+1} \stackrel{(3.28)}{\implies} f_t + \lambda * g_t = f_{t+1} + \lambda * g_{t+1} \Leftrightarrow$$

$$\lambda = \frac{f_{t+1} - f_t}{g_t - g_{t+1}} \approx \frac{df}{-dg} \quad (3.29)$$

Μπορούμε να θεωρήσουμε την μεταβολή αυτή ως διαφορική και να την ολοκληρώσουμε. Έτσι προκύπτει ότι:

$$\lambda = \frac{df}{-dg} \Leftrightarrow \int df = -\lambda \int dg \Leftrightarrow \quad (3.30)$$

$$f_{max} - f_{min} = -\lambda(g_{max} - g_{min}) \quad (3.31)$$

Από την θεωρία η επιθυμητή ελάχιστη παραβίαση των περιορισμών είναι 0, οπότε υπό προϋποθέσεις μπορεί  $g_{min} = 0$ , οπότε η σχέση (3.31) γίνεται:

$$\lambda = \frac{f_{max} - f_{min}}{-g_{max}} \quad (3.32)$$

Ακόμη, θέλουμε η τιμή του  $\lambda$  να είναι θετική, υπολογίζουμε:

$$\lambda = \left| \frac{f_{max} - f_{min}}{g_{min} - g_{max}} \right| = \left| \frac{f_{max} - f_{min}}{g_{max} - g_{min}} \right| \quad (3.33)$$

Ενώ στην περίπτωση που  $g_{min}=0$  η (3.33) γίνεται:

$$\lambda = \left| \frac{f_{max} - f_{min}}{-g_{max}} \right| = \left| \frac{f_{max} - f_{min}}{g_{max}} \right| \quad (3.34)$$

Η μέθοδος που περιγράφηκε μας δίνει μια καλή πρώτη προσέγγιση για την τάξη μεγέθους του  $\lambda$  και μια καλή ενδεικτική τιμή για το  $\lambda$ . Παρότι πραγματοποιήσαμε αρκετές παραδοχές προκειμένου να υπολογίσουμε την τιμή αυτή, η τιμή αυτή παραμένει πειραματική και για τον λόγο αυτό την δεχόμαστε. Προκειμένου όμως να είμαστε σε θέση να πραγματοποιήσουμε δοκιμές και να μεταβάλλουμε την τιμή αυτή, υπό την καθοδήγηση του επιβλέποντα καθηγητή, αποφασίσαμε να θέσουμε έναν πολλαπλασιαστή, τον οποίο ονομάσαμε αρχικό λάμδα ( $\lambda_{init}$ ), και μέσω αυτού να επιχειρήσουμε να βρούμε την βέλτιστη

τιμή του συντελεστή ποινής ( $\lambda$ ). Συγκεκριμένα, με την χρήση του  $\lambda_{init}$  η σχέση (3.33) μετασχηματίζεται:

$$\lambda = \lambda_{init} \left| \frac{f_{max} - f_{min}}{g_{min} - g_{max}} \right| = \lambda_{init} \left| \frac{f_{max} - f_{min}}{g_{max} - g_{min}} \right| \quad (3.35)$$

Ενώ για  $g_{min} = 0$  η σχέση (3.35) γίνεται:

$$\lambda = \lambda_{init} \left| \frac{f_{max} - f_{min}}{-g_{max}} \right| = \lambda_{init} \left| \frac{f_{max} - f_{min}}{g_{max}} \right| \quad (3.36)$$

Οι τιμές για τα  $g_{min}$  και  $g_{max}$  υπολογίζονται αφότου πραγματοποιήσουμε μια αρχικοποίηση στο πρόβλημα θέτοντας την τιμή  $\lambda=0$  και πραγματοποιώντας ελαχιστοποίηση της αντικειμενικής συνάρτησης ( $f_{min}$ ) (για να βρούμε την ελάχιστη τιμή των περιορισμών ( $g_{min}$ )) και μεγιστοποίηση της αντικειμενικής συνάρτησης ( $f_{max}$ ) (για να βρούμε την μέγιστη τιμή των περιορισμών ( $g_{max}$ )).

Έτσι, αφότου επιλέξουμε αυθαίρετα το  $\lambda_{init}$ , ο αλγόριθμός μας υπολογίζει ένα  $\lambda$ , το οποίο θεωρητικά είναι ικανό να οδηγήσει το πρόβλημα στην βέλτιστη δυνατή λύση.

### 3.3.5. Πιθανοθεωρητική αντιμετώπιση

Προκειμένου να αποφευχθεί ο εγκλωβισμός σε τοπικά ακρότατα, αλλά και να διευκολυνθεί η εξερεύνηση του διαστήματος αναζήτησης, υπό την καθοδήγηση του επιβλέποντα καθηγητή, αποφασίσαμε να εισαγάγουμε μια πιθανοθεωρητική αντιμετώπιση, όσον αφορά την δημιουργία του νέου πληθυσμού σε κάθε επανάληψη.

Συγκεκριμένα, επιδιώξαμε να προσδώσουμε το στοιχείο της τυχαιότητας σε περιοχές κοντινές στο ολικό βέλτιστο, προκειμένου να βοηθήσουμε τον αλγόριθμο να ψάξει για τις καλύτερες δυνατές τιμές. Θεωρήσαμε, υπό την καθοδήγηση του επιβλέποντα καθηγητή, την δημιουργία ενός σύννεφου σωματιδίων, το οποίο λαμβάνει υπόψιν του έναν παράγοντα «αρχικής τυπικής απόκλισης» αλλά και έναν «ρυθμό μείωσης της τυπικής απόκλισης».

Ο παράγοντας «αρχικής τυπικής απόκλισης» χρησιμοποιείται για να προσδώσει το εύρος τιμών, στο οποίο ο αλγόριθμος θα αναζητήσει καλύτερη τιμή από την τωρινή ολική βέλτιστη, και ορίζεται αυθαίρετα από τον μελετητή (αποφασίζεται μέσα από δοκιμές) προκειμένου να ευρεθεί η βέλτιστη δυνατή τιμή.

Ο «ρυθμός μείωσης της τυπικής απόκλισης» χρησιμοποιείται προκειμένου να προσδώσει μείωση στην «αρχική τυπική απόκλιση» μετά από έναν καθορισμένο αριθμό επαναλήψεων. Μέσω της μείωσης του εύρους αναζήτησης επιτυγχάνεται η εύρεση καλύτερης δυνατής λύσης, αφού, όσο περνάει ο χρόνος, ο αλγόριθμος συγκλίνει σε όλο και πιο καλή λύση. Ο ρυθμός αυτός ορίζεται από τον μελετητή με βάση την εύρεση των βέλτιστων δυνατών αποτελεσμάτων.

### 3.3.6. Μοντελοποίηση του αλγορίθμου σε λογισμικό MATLAB ®

Στην συνέχεια παρουσιάζουμε τα βασικά σημεία του αλγορίθμου, τον οποίο μοντελοποιήσαμε στο λογισμικό MATLAB ®. Η διαδικασία φαίνεται αναλυτικά και στο διάγραμμα ροής του σχήματος 3.8.

Προκειμένου να αρχικοποιήσουμε το  $\lambda$  μας, υπό την καθοδήγηση του επιβλέποντα καθηγητή, πραγματοποιούμε 2 επαναληπτικές διαδικασίες, όπως αναφέραμε προηγουμένως. Με  $\lambda=0$  και ρυθμό μείωσης της τυπικής απόκλισης ίσο με 1, πραγματοποιούμε μια δοκιμή για πρόβλημα μεγιστοποίησης και μία για πρόβλημα ελαχιστοποίησης της αντικειμενικής. Στο διάγραμμα ροής, η ελαχιστοποίηση διαφοροποιείται από την μεγιστοποίηση μέσα από τον παράγοντα  $MM_i$ . Για πρόβλημα ελαχιστοποίησης θεωρούμε τιμή  $MM_i=1$  ενώ για πρόβλημα μεγιστοποίησης θεωρούμε τιμή  $MM_i=-1$ .

Αφότου εισάγουμε στο πρόβλημα τους περιορισμούς μας, το πεδίο τιμών και τις αντικειμενικές συναρτήσεις, βρίσκουμε αρχικές τιμές για το πρόβλημα, ξεκινώντας από τυχαίες τιμές για την αρχική θέση και την αρχική ταχύτητα των σωματιδίων. Αφού ελέγξουμε τα σωματίδια για τιμές εκτός ορίων, από τα σωματίδια, τα οποία πληρούν τους περιορισμούς, επιλέγουμε το βέλτιστο και κρατάμε τις τιμές του (αριθμός του σωματιδίου, συντεταγμένες, τιμή αντικειμενικής συνάρτησης και συνάρτησης ποινής).

Στην συνέχεια εισερχόμαστε στον αλγόριθμο SLPSO. Αφού αρχικοποιήσουμε τις τιμές που αυτός απαιτεί ( $U_f, G_i, P_i$ , κ.λ.π.) ξεκινάει η επαναληπτική διαδικασία εύρεσης της βέλτιστης τιμής. Χρησιμοποιώντας τις διάφορες τιμές που έχουν προκύψει από την αρχικοποίηση, εφαρμόζουμε τον Αλγόριθμο 1 που περιγράφεται στο σχήμα 3.3.

Επιδιορθώνουμε τις τιμές που βρίσκονται εκτός πεδίου τιμών και ανανεώνουμε την τιμή του  $G_i$  ενώ υπολογίζουμε την τιμή της σχέσης (3.28).

Ακολουθώντας, όπου χρειάζεται, εφαρμόζουμε τον Αλγόριθμο 2 (σχήμα 3.4) και αφού υπολογίσουμε τις νέες τιμές των σωματιδίων κρατάμε τη βέλτιστη τιμή του κάθε σωματιδίου έως τώρα. Αφού εφαρμόσουμε τον Αλγόριθμο 3 (σχήμα 3.5), ανανεώνουμε τις τιμές των παραμέτρων, όπως αυτές ορίζονται από τον Αλγόριθμο 4 (σχήμα 3.6). Σε αυτό το σημείο η εφαρμογή της SLPSO ολοκληρώνεται και μπαίνουμε στην εφαρμογή της μεθόδου με την πιθανοθεωρητική αντιμετώπιση που αναλύσαμε προηγουμένως.

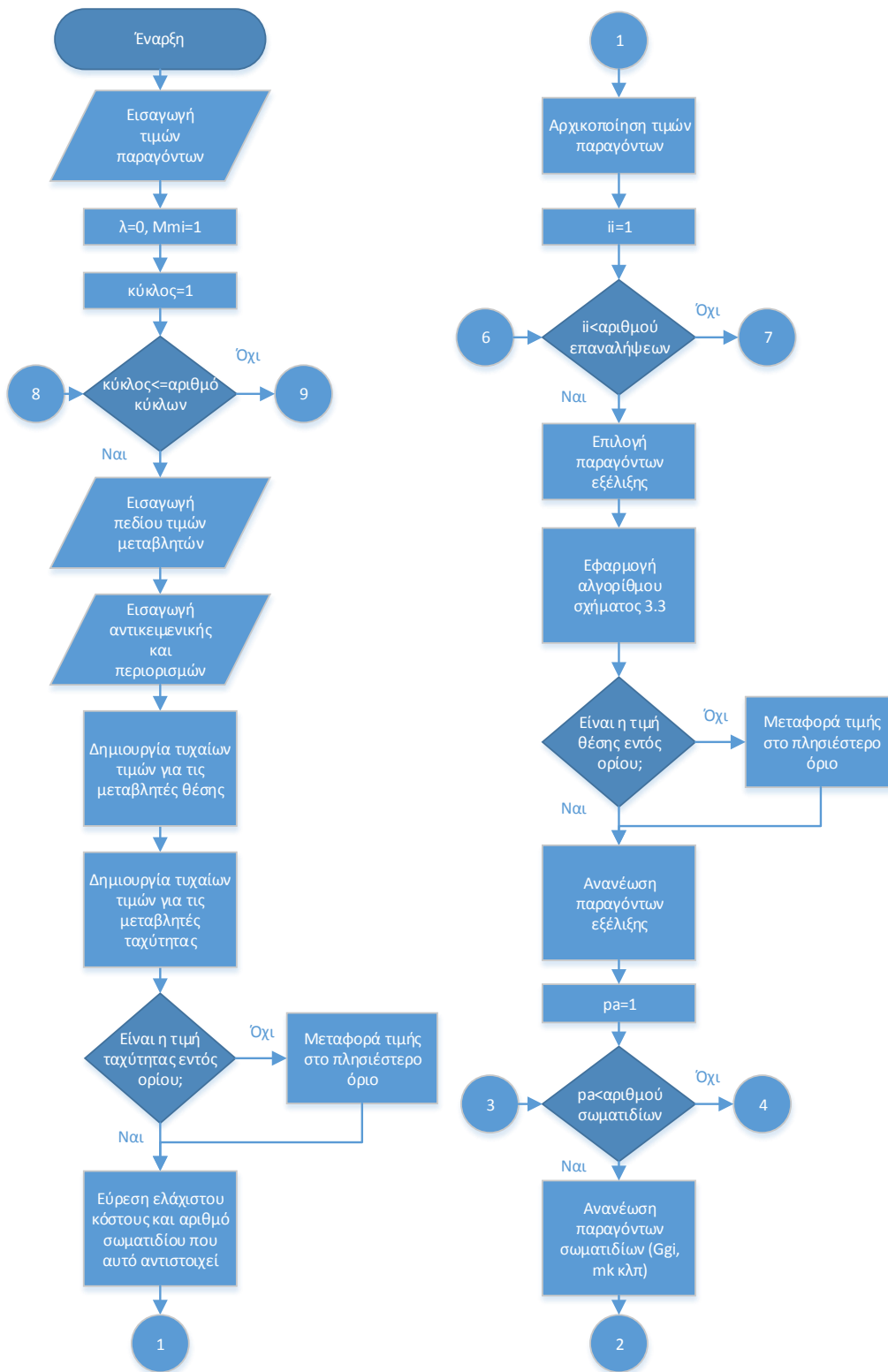
Τέλος, μετά τις πρώτες 1,000<sup>1</sup> επαναλήψεις ο ρυθμός μείωσης της τυπικής απόκλισης αρχίζει να μεταβάλλεται με βάση την τυχειότητα και μια αρχική τιμή την οποία ο μελετητής έχει δώσει αυθαίρετα, δίνοντας έτσι χαρακτήρα μετάλλαξης στον ρυθμό. Η επαναληπτική διαδικασία επαναλαμβάνεται για όσες επαναλήψεις θέσει ο μελετητής και δεν σταματά μέσω κάποιου άλλου κριτηρίου τερματισμού. Μέσα από την εφαρμογή της προηγούμενης διαδικασίας προκύπτουν οι αρχικές τιμές για τα  $f_{max}$ ,  $f_{min}$ ,  $g_{min}$  και  $g_{max}$ , τα οποία μας βοηθάνε στον υπολογισμό της τιμής του  $\lambda$  με τον τρόπο που αναλύθηκε στην υπο-παράγραφο 3.3.4.

Πλέον εφαρμόζουμε την ίδια μέθοδο που αναλύσαμε για την αρχικοποίηση, με την διαφορά ότι αντί για  $\lambda=0$  το  $\lambda$  έχει πάρει συγκεκριμένη τιμή, ο ρυθμός μείωσης της τυπικής απόκλισης ξεκινάει από την τιμή που δίνει ο μελετητής και όχι από την τιμή 1, όπως πριν. Με την εφαρμογή της μεθόδου αυτής προκύπτει η βέλτιστη δυνατή τιμή που βρήκε ο αλγόριθμος (βέλτιστη αντικειμενική αλλά και το σφάλμα-άθροισμα περιορισμών αυτής).

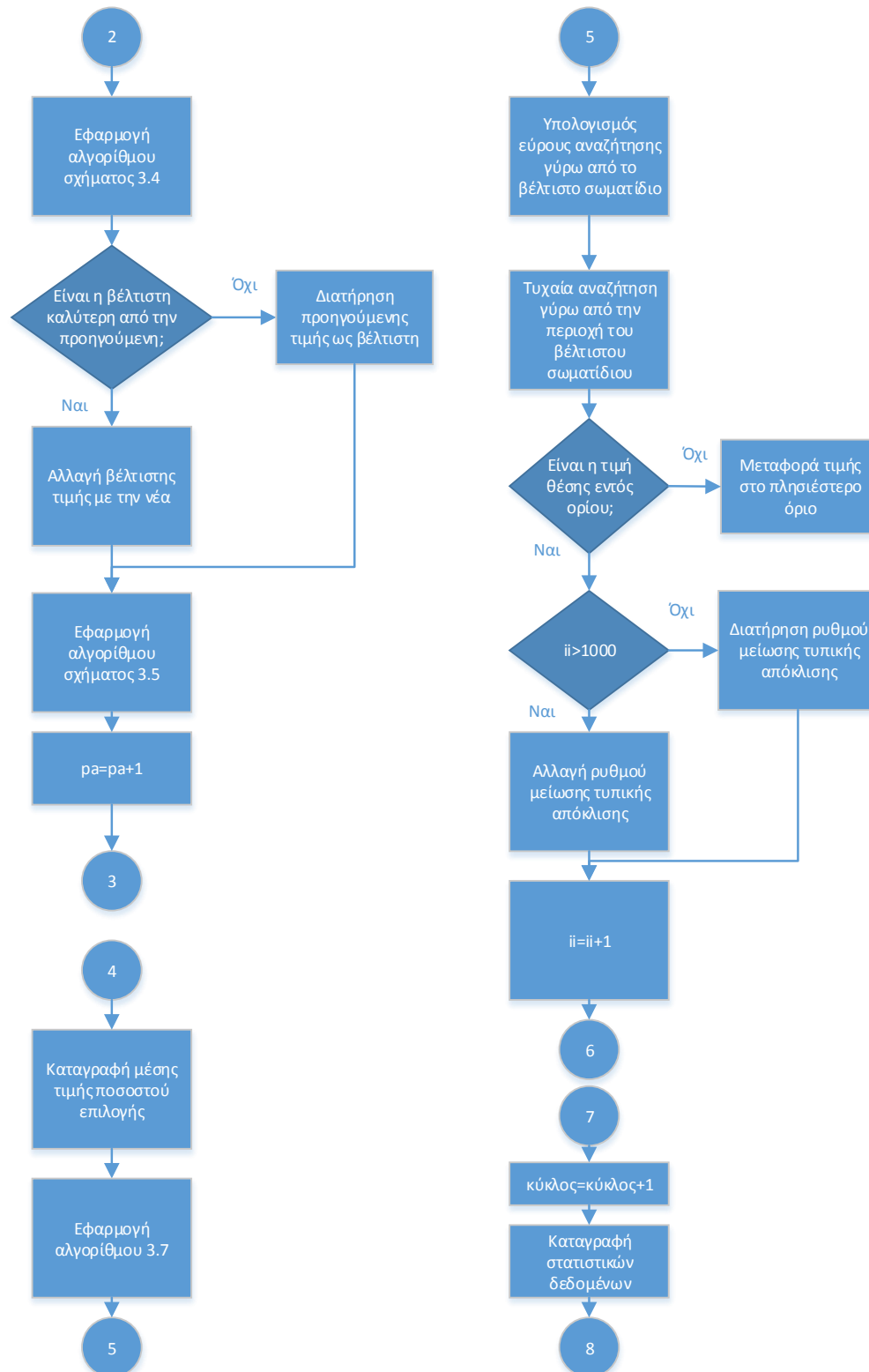
Σε περίπτωση όπου, με τα ίδια δεδομένα αρχικοποίησης, θέλουμε να εφαρμόσουμε τον ίδιο αλγόριθμο πολλές φορές, μπορούμε να το κάνουμε και να εξάγουμε καλές πληροφορίες για το πρόβλημα. Συγκεκριμένα, επιλέγουμε να το κάνουμε προκειμένου να προκύψει μια στατιστική ανάλυση των αποτελεσμάτων που έβγαλε ο κώδικας. Μέσα από την ανάλυση αυτή προκύπτει η μέση τιμή, η τυπική απόκλιση, η μέγιστη και η ελάχιστη

---

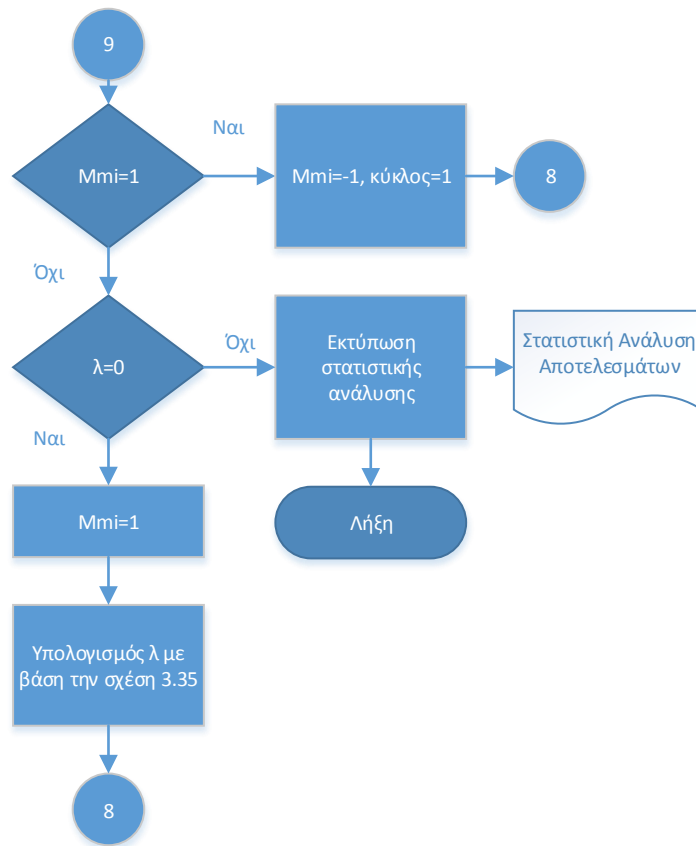
<sup>1</sup> Για όλα τα νούμερα τα οποία αναφέρονται στην διπλωματική εργασία, τόσο στο κείμενο όσο και σε πίνακες ή σχήματα, χρησιμοποιείται το αγγλοσαξονικό σύστημα αρίθμησης π.χ. δύο χιλιάδες: 2,000 ,δύο: 2.000



**Σχήμα 3.8(α):** Πρώτο τμήμα διαγράμματος ροής μοντελοποιημένου αλγορίθμου SLPSO



**Σχήμα 3.8(β):** Δεύτερο τμήμα διαγρα/τος ροής μοντελοποιημένου αλγορίθμου SLPSO



**Σχήμα 3.8(γ):** Τρίτο τμήμα διαγρά/τος ροής μοντελοποιημένου αλγορίθμου SLPSO

τιμή της τιμής της αντικειμενικής συνάρτησης, της τιμής της συνάρτησης ποινής και της τιμής της αντικειμενικής όταν σε αυτήν εφαρμόσουμε την συνάρτηση ποινής.

### 3.3.7. Σκοπός μελέτης της μεθόδου Self-Learning Particle Swarm Optimizer

Η μέθοδος της SLPSO είναι μια σχετικά καινούρια μέθοδος στον τομέα της βελτιστοποίησης με τη χρήση σμήνους σωματιδίων. Μέσα από την μελέτη της μεθόδου αυτής στο πλαίσιο της διπλωματικής εργασίας, έγινε δοκιμή προκειμένου να αποδειχθεί εάν η σύγκλιση των 3 μεθόδων που περιγράψαμε μπορεί να συνδυαστεί και να αποφέρει τα επιθυμητά αποτελέσματα στην επίλυση 4 κλασσικών προβλημάτων μη-γραμμικού προγραμματισμού με περιορισμούς. Για να το επιτύχουμε αυτό πραγματοποιήσαμε δοκιμές σε 3 βασικούς παράγοντες της μεθόδου, όπως αυτή περιγράφηκε παραπάνω: Στην αρχική τιμή του  $\lambda$ , η οποία μας διευκολύνει, περιορίζοντας σε πολύ μεγάλο βαθμό το πεδίο

αναζήτησης το οποίο θα έπρεπε να μελετήσουμε υπό άλλες συνθήκες, στην αρχική τιμή τυπικής απόκλισης, η οποία προσδίδει στον αλγόριθμο την δυνατότητα να εξερευνήσει περιοχές κοντινές στην βέλτιστη έως εκείνη την ώρα τιμή που έχει βρει αλλά και στον αρχικό ρυθμό μείωσης της τυπικής απόκλισης, ο οποίος βοηθά στην σύγκλιση της εξερεύνησης των τιμών κοντά στην ολική βέλτιστη, όσο η βέλτιστη λύση συγκλίνει και η ίδια προς την βέλτιστη τιμή της αντικειμενικής. Τα προβλήματα, τα οποία μελετήθηκαν, καθώς και τα αποτελέσματα, τα οποία προέκυψαν, μελετώνται στο επόμενο κεφάλαιο.



## **ΚΕΦΑΛΑΙΟ 4**

### **ΤΑ ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΕΛΕΓΧΘΗΚΑΝ**



Στο κεφάλαιο αυτό θα αναλύσουμε τα προβλήματα που χρησιμοποιήθηκαν κατά την εφαρμογή των 2 αλγορίθμων, στο πλαίσιο της εκπόνησης της διπλωματικής εργασίας. Συνοπτικά, αναφέρεται η αντικειμενική εξίσωση, οι περιορισμοί και οι βέλτιστες τιμές που μας δίνονται από την βιβλιογραφία, σε περίπτωση που αυτές είναι διαθέσιμες.

#### **4.1. Πρόβλημα 1<sup>ο</sup>**

Το πρώτο πρόβλημα, το οποίο μελετήσαμε, αφορά την μελέτη της επίδρασης της μακροπρόθεσμης επέκτασης της ενεργειακής αγοράς στην εξέλιξη της τιμής του ηλεκτρικού ρεύματος, όπως αυτή αναπτύχθηκε και μελετήθηκε πρόσφατα (Tolis, 2015). Το πρόβλημα αυτό μελετήθηκε μόνο με την εφαρμογή της μεθόδου Global Competitive Ranking, ακριβώς όπως αυτή επεξηγήθηκε στην παράγραφο 3.2 της παρούσας διπλωματικής εργασίας.

Ένα σημαντικό στοιχείο του προβλήματος είναι ο υπολογισμός της εξέλιξης της οριακής τιμής του συστήματος (System Marginal Price), της τιμής, δηλαδή, την οποία η εταιρεία διανομής ενέργειας χρεώνεται προκειμένου να αγοράσει μια επιπλέον κιλοβατώρα (kWh), η οποία μεταφράζεται, κατά κάποιον τρόπο, στην τιμή που φτάνει στον πολίτη. Μέσα από κάποιες υποθέσεις σχετικά με την διαθεσιμότητα και την τιμή των καυσίμων στην χρονική περίοδο που μελετάμε, ο αλγόριθμος επιδιώκει να βρει τη βέλτιστη δυνατή Καθαρή Παρούσα Αξία του προβλήματος. Η μελέτη πραγματοποιείται σε ορίζοντα 38 ετών, με το έτος 2013 να θεωρείται ως το έτος αναφοράς για τη σύγκριση οικονομικών τιμών.

Κατά την εκτέλεση του αλγορίθμου βελτιστοποίησης μελετώνται 2 διαφορετικά μοντέλα ενδογενών μεθόδων, ένας άμεσος και ένας έμμεσος. Η αντικειμενική συνάρτηση, η οποία αναπαριστά το μέσο NPV του συστήματος μοντελοποιείται με 2 παρεμφερείς εξισώσεις, μία άμεση και μία έμμεση. Στην πρώτη περίπτωση η συνάρτηση είναι η συνάρτηση 4.1 ενώ στην δεύτερη περίπτωση η συνάρτηση που ισχύει είναι η συνάρτηση 4.2. Στην μοντελοποίηση του προβλήματος λήφθηκαν υπόψιν και οι 2 μέθοδοι.

Στον πίνακα 4.1 που ακολουθεί φαίνονται οι διάφορες μεταβλητές που χρησιμοποιούνται στο πρόβλημα. Για μεγαλύτερη λεπτομέρεια ο αναγνώστης θα πρέπει να ανατρέξει στην σχετική βιβλιογραφία.

$$\begin{aligned}
& NPV_{explicit}(X_{i=1,v}, \dots, X_{i=l,v}, \theta_{i=1,z}, \dots, \theta_{i=l,z}) \\
&= \max \left\{ \sum_{z=1}^Y [\overline{SMP}_{z_1-0} * E1_z + \overline{SMP}_{z_2-1} E2_z] * D_z \right. \\
&+ \sum_{i=1}^{RE} \sum_{z=1}^Y pre_z * P_{i,z} * D_z + \sum_{z=1}^Y \overline{SMP}_z * (d_{z,f} \\
&- \sum_{i=1}^{RE} P_{i,z} - E1_z - E2_z) * D_z \\
&- \sum_{v=0}^Y \sum_{i=1}^I I_{i,v} * X_{i,v} - \sum_{i=1}^{I-RE} \sum_{z=1}^Y P_{i,z} * \frac{f_{co2i}}{n_i} * p_{co2z} * D_z \\
&+ \sum_{i=1}^{RE} \sum_{z=1}^Y P_{i,z} \frac{\sum_{i=1}^{I-RE} E_{co2i}}{\sum_{i=1}^{I-RE} P_{i,z}} p_{co2z} * D_z \\
&\left. - \sum_{i=1}^I \sum_{z=1}^Y Cf_{i,z} * P_{i,z} * D_z - \sum_{i=1}^I \sum_{z=1}^Y C_{v_{i,z}} * L_{i,z} * D_z \right\} \quad (4.1)
\end{aligned}$$

$$\begin{aligned}
& NPV_{implicit}(X_{i=1,v}, \dots, X_{i=l,v}, \theta_{i=1,z}, \dots, \theta_{i=l,z}) \\
&= \max \left\{ \sum_{z=1}^Y pe_z * \left( d_{z,f} - \sum_{i=1}^{RE} P_{i,z} \right) * D_z \right. \\
&+ \sum_{i=1}^{RE} \sum_{z=1}^Y pre_z * P_{i,z} * D_z \\
&- \sum_{i=1}^I \sum_{z=1}^Y Cf_{i,z} * P_{i,z} * D_z - \sum_{i=1}^I \sum_{z=1}^Y C_{v_{i,z}} * L_{i,z} * D_z \\
&- \sum_{v=0}^Y \sum_{i=1}^I I_{i,v} * X_{i,v} - \sum_{i=1}^{I-RE} \sum_{z=1}^Y P_{i,z} * \frac{f_{co2i}}{n_i} * p_{co2z} * D_z \\
&\left. + \sum_{i=1}^{RE} \sum_{z=1}^Y P_{i,z} \frac{\sum_{i=1}^{I-RE} E_{co2i}}{\sum_{i=1}^{I-RE} P_{i,z}} p_{co2z} * D_z \right\} \quad (4.2)
\end{aligned}$$

Symbol	Description and units
$a_{a,i}$	Availability factor for technology ( $i$ ) (in percent)
$a_{c,i}$	Capacity factor for technology ( $i$ ) (in percent)
$b_{(i)}$	Learning rate for technology ( $i$ )
$C_{i,v}$	Past capacity orders of technology ( $i$ ) decided in year ( $v$ ) before the beginning of time counter
$C_{f,i,z}$	Fuel cost for year ( $z$ ) and technology ( $i$ ) (€/megawatt hour electric)
$C_{v,i,z}$	Operational and maintenance costs for power plant type ( $i$ ) in year ( $z$ ) (euros per megawatt)
$dWt$	Wiener (Brownian) differential vector (Normally distributed)
$d_{z,f}$	Total electricity demand for year ( $z$ ) (simulated projection including losses: in megawatt hour electric)
$D$	Vector of diffusion for a generic stochastic process (-)
$D_z$	Discounting factor for year ( $z$ )
$E_{CO_2i}$	CO <sub>2</sub> emissions from technology ( $i$ ) (in CO <sub>2</sub> equivalent tons)
$En_i$	Natural resources potential for technology ( $i$ ) (in megawatt)
$f_{CO_2i}$	Emissions factor for unit ( $i$ ) (in CO <sub>2</sub> equivalent tons/MWh <sub>fuel</sub> )
$I$	Total number of power generating technologies/fuel types.
$I_{i,v}$	Capital cost for capacity orders of technology ( $i$ ) decided in year ( $v$ ) (€/megawatt electric)
$I_{Dz,i}$	Capital cost repayment (annuity) for capacity orders of technology ( $i$ ), paid in year $z$ (€/megawatt electric)
$L_{i,z}$	Installed capacity corresponding to technology ( $i$ ) in year $z$ (megawatt electric)
$m_r$	Reserve margin of peak electricity power (in percent)
$n_i$	Efficiency factor for unit ( $i$ ) (in percent)
NPV	Net Present Value (in euros)
$p_{CO_2z}$	CO <sub>2</sub> allowance price for year ( $z$ ) (€/tons CO <sub>2</sub> )
$p_{e_z}$	Average SMP for conventional technologies and hydro-plants in year $z$ (€/megawatt hour electric)
$p_{re_z}$	Fixed electricity price for renewable energies in year ( $z$ ) (€/megawatt hour electric)
$PP_i, Py_i$	Peak and low production levels annually observed for technology ( $i$ ) (megawatt electric)
$P_{c_z}$	Peak power demand in year $z$ (megawatt electric)
$P_{i,z}$	Total electricity energy production in year ( $z$ ) from power plants of technology ( $i$ ) (megawatt hour electric)
$P_{Oz}(t)$	Electricity consumption historically recorded in hour ( $t$ ) of year ( $z$ ) (megawatt electric)
$profits_{i,z}$	Profits for technology ( $i$ ) in year ( $z$ ) (in euros)
RE	Number of energy producing technologies based on renewable sources.
$r_{in,t}$	Inflation rate of time point $t$ (in percent)
$r_t$	Interest rate of time point $t$ (in percent)
$Sx_{i,z}$	Unknown price coefficient corresponding to technology ( $i$ ) and year ( $z$ ) (€/megawatt hour electric)
$Tl_i$	Lead-time (after investing entry) for constructing-commissioning of power-plant type ( $i$ ) (years)
$To_i$	Operational life-time for unit type ( $i$ ) (years)
$T_{zmax}$	Hours of a year (hours)
$v$	Investment entry time or order decision (specific year).
$V, \mu$	Volatility vector and mean drift for a Wiener Process (-)
$X_{i,v}$	Unknown Capacity orders of technology ( $i$ ) decided in year ( $v$ ) (megawatt electric).
$Y$	Maximum operational life-time of the system (years)
$z$	Generic time counter (years)
$\theta_{i,z}$	Unknown load intensity factor: actual operating over total available time for unit $i$ during year $z$ (in percent)
$\bar{Cf}$	Matrix of fuel prices for conventional technologies and hydro-plants ( $n$ -year block)
$\bar{P}$	Matrix of electricity production for conventional technologies and hydro-plants ( $n$ -year block)
$Pe$	Vector of system marginal prices for $n$ -year block (only the last element is saved and restored)
$Sx$	Vector of unknown price coefficient for conventional technologies and hydro-plants ( $n$ -year block)

**Πίνακας 4.1:** Μεταβλητές προβλήματος 1

Οι περιορισμοί, στους οποίους υπόκειται το πρόβλημα, ορίζονται μέσα από τις ανισότητες που ακολουθούν (για αναλυτική επεξήγηση του κάθε όρου ο αναγνώστης θα πρέπει να ανατρέξει στην βιβλιογραφία (Tolis, 2015)):

$$L_{i,z} \leq En_i, \quad \forall i, z \quad (4.3)$$

$$(1 + e) * d_{z,f} \geq \sum_{i=1}^I P_{i,z} \geq d_{z,f} \Leftrightarrow$$

$$(1 + e) * d_{z,f} \geq \sum_{i=1}^I \left( \sum_{v=1}^{z-TI_i} X_{i,v} + \sum_{v=-40}^0 C_{i,v} - \sum_{v=1}^z \overline{X_{i,v}} - \sum_{v=-40}^0 \overline{C_{i,v}} \right)$$

$$* \theta_{i,z} * a_{a,i} * a_{c,i} * 8760 \geq d_{z,f}, \quad \forall z, e \ll 1 \quad (4.4)$$

$$\sum_{i=1}^I a_{a,i} * L_{i,z} \geq P_{C_z} * (1 + m_r), [\forall z, a_{a,i} = 1,$$

$\forall i$  except wind hydro and solar for which

$$a_{a,i} \leq 1 \Leftrightarrow \sum_{i=1}^I a_{a,i} * \sum_{v=1}^{z-TI_i} X_{i,v} + \sum_{v=-40}^0 C_{i,v} - \sum_{v=1}^z \overline{X_{i,v}} - \sum_{v=-40}^0 \overline{C_{i,v}}$$

$$\geq P_{C_z} * (1 + m_r) \quad (4.5)$$

$$\sum_{i=1}^{RE} P_{i,z} \leq 1 * d_{z,f}, \quad \forall z \quad (4.6)$$

$$\sum_{i=1}^{RE} P_{i,z} > (0.201) * \sum_{i=1}^I P_{i,z}, \quad \forall z < 2022 \quad (4.7)$$

$$\sum_{i=1}^{RE} P_{i,z} > (0.3) * \sum_{i=1}^I P_{i,z}, \quad \forall z \geq 2022 \quad (4.8)$$

$$X_{i=1,v} \geq 350 (\alpha), X_{i=2,v} \geq 350, X_{i,v} > 0 \forall v \& i \neq 1,2$$

$$(c) t_{marginal,z} < t_{peak:projected duration,z} \forall z(d) \quad (4.9)$$

$$pe_z \leq \overline{SMP_z} * (1 + fm), \quad \forall z \quad (4.10)$$

$$profits_{i,z} = \left[ \theta_{i,min} * a_{a,i} * a_{c,i} * 8760 \right.$$

$$\left. * \left( pe_z - Cf_{i,z} - \frac{f_{co2i}}{n_i} * p_{co2z} \right) - C_{v,i,z} \right] - I_{D_z,i} \geq 0 \quad (4.11)$$

$$\begin{aligned}
profits_{i,z} = & -I_{D_z,i} + [\theta_{i,min} * a_{a,i} * a_{c,i} * 8760 \\
& * \left( t_{oil} * \overline{SMP}_{z_1-0} + (1 - \delta_{ij}) * (T_{max} - t_{oil}) \right. \\
& \left. * \overline{SMP}_{z_2-1} - Cf_{i,z} - \frac{f_{co_2i}}{n_i} * p_{co_2z} \right) - C_{v_{i,z}} \geq 0
\end{aligned} \tag{4.12}$$

Το μη-γραμμικό αυτό πρόβλημα που περιγράψαμε μόλις, μοντελοποιήθηκε και τα αποτελέσματά του ελέγχθηκαν με την μέθοδο Global Competitive Ranking. Η μεθοδολογία που ακολουθήθηκε καθώς και τα αποτελέσματα που προέκυψαν από τις δοκιμές παρατίθενται στο κεφάλαιο 5.

Με βάση την βιβλιογραφία, τα βέλτιστα αποτελέσματα, τα οποία προκύπτουν για το πρόβλημα αυτό, αποφέρουν την τιμή  $NPV_{best} = 177.6530$  (δισ €). Εξαιτίας του μεγάλου πλήθους του αριθμού των μεταβλητών, η βέλτιστη λύση δίδεται στον πίνακα 4.2.

## 4.2. Πρόβλημα 2<sup>ο</sup>

Τα υπόλοιπα 4 μη-γραμμικά προβλήματα, τα οποία μελετήσαμε, είναι μερικά από τα πιο συνηθισμένα χρησιμοποιούμενα προβλήματα που κυκλοφορούν στην βιβλιογραφία (Runarsson & Yao, 2003) (Liang et al., 2006). Η αντικειμενική συνάρτηση προς ελαχιστοποίηση του 2<sup>ου</sup> προβλήματός μας και 4<sup>ου</sup> κατά την βιβλιογραφία είναι:

$$\begin{aligned}
f(\vec{x}) = & 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 \\
& - 40792.141
\end{aligned} \tag{4.13}$$

Ενώ οι συναρτήσεις των περιορισμών είναι:

$$\begin{aligned}
g_1(\vec{x}) = & 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 \\
& - 0.0022053x_3x_5 - 92 \leq 0
\end{aligned} \tag{4.14}$$

$$\begin{aligned}
g_2(\vec{x}) = & -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 \\
& + 0.0022053x_3x_5 \leq 0
\end{aligned} \tag{4.15}$$

$$\begin{aligned}
g_3(\vec{x}) = & 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 \\
& + 0.0021813x_3^2 - 110 \leq 0
\end{aligned} \tag{4.16}$$

Καύσιμο\Έτος	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Natural Gas	0.0026	0.0000	0.1308	0.0654	0.0020	0.0002	0.0072	0.0102	0.0489	0.0485	0.0482	0.0481	0.0494	0.0512	0.0545	0.0569	0.0605	0.0632	0.0662
Lignite	0.0168	0.0143	0.0100	0.0067	0.0046	0.0029	0.0042	0.0722	0.1227	0.1196	0.1166	0.1142	0.1144	0.1153	0.1161	0.1175	0.1204	0.1218	0.1227
Oil	0.0025	0.0014	0.0331	0.0021	0.0008	0.0220	0.0000	0.0053	0.0158	0.0152	0.0158	0.0154	0.0154	0.0156	0.0158	0.0161	0.0166	0.0170	0.0173
Hydro medium	0.2086	0.2040	0.1292	0.1661	0.1598	0.1540	0.1498	0.1466	0.1064	0.1051	0.1376	0.1340	0.1300	0.1281	0.1231	0.1443	0.1161	0.1345	0.1310
Hydro pumped	0.2114	0.1527	0.1168	0.1102	0.1037	0.0980	0.0964	0.0938	0.1344	0.0880	0.1266	0.0872	0.0856	0.0822	0.0779	0.1343	0.1285	0.1222	0.1195
Biomass	0.2951	0.2545	0.2477	0.2400	0.2369	0.2856	0.2816	0.1993	0.2298	0.2365	0.2252	0.2176	0.2099	0.2035	0.1976	0.1928	0.1890	0.1855	0.1805
Solar PV	0.0029	0.0309	0.0290	0.0279	0.0267	0.0368	0.0353	0.0352	0.0465	0.0489	0.0317	0.0328	0.0300	0.0276	0.0262	0.0245	0.0228	0.0206	0.0195
Wind Turbine	0.0000	0.1265	0.1225	0.1217	0.1193	0.1408	0.1372	0.1370	0.1600	0.1637	0.1485	0.1427	0.1370	0.0987	0.0956	0.1243	0.1208	0.0885	0.1159
$\Theta_1$	0.7218	0.7295	0.6391	0.6462	0.6413	0.6000	0.6000	0.6000	0.6000	0.6391	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000
$\Theta_2$	1.0000	1.0000	1.0000	1.0000	1.0000	0.9906	0.9612	0.9728	0.9189	0.9926	0.9350	0.9159	0.8262	0.7516	0.7500	0.7500	0.7500	0.7500	0.7500
$\Theta_3$	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300
Καύσιμο\Έτος	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Natural Gas	0.0678	0.0696	0.1054	0.1069	0.1095	0.1121	0.1174	0.1235	0.1279	0.1503	0.0246	0.0181	0.0222	0.1368	0.1397	0.1416	0.1033	0.0391	0.0396
Lignite	0.1232	0.1208	0.1525	0.1498	0.1479	0.1477	0.1485	0.1508	0.1537	0.1769	0.1799	0.1818	0.1837	0.1481	0.1474	0.1462	0.1049	0.0361	0.0369
Oil	0.0172	0.0173	0.0181	0.0179	0.0181	0.0181	0.0186	0.0188	0.0195	0.0199	0.0207	0.0212	0.0216	0.0221	0.0224	0.0227	0.0231	0.0235	0.0001
Hydro medium	0.1285	0.1286	0.1255	0.1407	0.1417	0.1404	0.1385	0.1080	0.1036	0.0990	0.0638	0.0099	0.0102	0.0106	0.0110	0.0114	0.0118	0.0122	0.0126
Hydro pumped	0.1165	0.1141	0.1124	0.1301	0.1288	0.1269	0.1270	0.0909	0.0856	0.0798	0.0385	0.0022	0.0086	0.0037	0.0003	0.0177	0.0000	0.0016	0.0011
Biomass	0.1732	0.1644	0.2053	0.1994	0.1922	0.1880	0.1837	0.1840	0.1846	0.2140	0.2121	0.2105	0.2076	0.1501	0.1520	0.1400	0.0703	0.0000	0.0009
Solar PV	0.0188	0.0172	0.0155	0.0130	0.0205	0.0184	0.0162	0.0141	0.0131	0.0117	0.0089	0.0077	0.0206	0.0024	0.0117	0.0014	0.0019	0.0221	0.0029
Wind Turbine	0.1131	0.1104	0.1075	0.1021	0.1179	0.1129	0.1088	0.1051	0.1021	0.0909	0.1514	0.0090	0.0046	0.0072	0.0019	0.0784	0.0736	0.0684	0.0392
$\Theta_1$	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6276	0.6000	0.6084	0.6718	0.8582	1.0000
$\Theta_2$	0.7500	0.7500	0.9119	0.9471	0.8853	0.8091	0.8141	0.8011	0.7500	0.7500	0.7500	0.7500	0.7806	0.7722	0.9748	1.0000	1.0000	1.0000	1.0000
$\Theta_3$	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300

Πίνακας 4.2: Βέλτιστες τιμές των μεταβλητών του προβλήματος 1 με βάση την βιβλιογραφία



$$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \quad (4.17)$$

$$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \quad (4.18)$$

$$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0 \quad (4.19)$$

Όπου  $78 \leq x_1 \leq 102$ ,  $33 \leq x_2 \leq 45$ , και  $27 \leq x_i \leq 45$  ( $i = 3,4,5$ ). Η βέλτιστη τιμή σύμφωνα με την βιβλιογραφία βρίσκεται για  $\vec{x}^* = (78, 33, 29.9952560256815985, 45, 36.7758129057882073)$  και είναι  $f(\vec{x}^*) = -30665.53867178322$ .

### 4.3. Πρόβλημα 3<sup>ο</sup>

Η αντικειμενική προς ελαχιστοποίηση συνάρτηση του 3<sup>ου</sup> προβλήματός μας και 7<sup>ου</sup> κατά την βιβλιογραφία είναι:

$$f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \quad (4.20)$$

Και υπόκειται στους περιορισμούς:

$$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \quad (4.21)$$

$$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 - 2x_8 \leq 0 \quad (4.22)$$

$$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \quad (4.23)$$

$$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \quad (4.24)$$

$$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \quad (4.25)$$

$$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \quad (4.26)$$

$$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \quad (4.27)$$

$$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \quad (4.28)$$

Με  $-10 \leq x_i \leq 10$  ( $i = 1, \dots, 10$ ). Η βέλτιστη τιμή βρίσκεται στο:  $\vec{x}^* = (2.17199634142692, 2.3636830416034, 8.77392573913157, 5.09598443745173, 0.990654756560493, 1.43057392853463, 1.32164415364306, 9.82872576524495, 8.2800915887356, 8.3759266477347)$  και είναι  $f(\vec{x}^*) = -24.30620906818$ .

#### 4.4. Πρόβλημα 4<sup>ο</sup>

Η αντικειμενική συνάρτηση προς ελαχιστοποίηση του 4<sup>ου</sup> προβλήματός μας και 8<sup>ου</sup> κατά την βιβλιογραφία είναι:

$$f(\vec{x}) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \quad (4.29)$$

Και υπόκειται στους περιορισμούς:

$$g_1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0 \quad (4.30)$$

$$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0 \quad (4.31)$$

Όπου  $0 \leq x_1 \leq 10$  και  $0 \leq x_2 \leq 10$  ενώ βέλτιστη τιμή σύμφωνα με την βιβλιογραφία βρίσκεται στο:  $\vec{x}^* = (1.22797135260752599, 4.24537336612274885)$  και είναι ίση με  $f(\vec{x}^*) = -0.0958250414180359$ .

#### 4.5. Πρόβλημα 5<sup>ο</sup>

Η αντικειμενική συνάρτηση προς ελαχιστοποίηση του 4<sup>ου</sup> προβλήματός μας και 8<sup>ου</sup> κατά την βιβλιογραφία είναι:

$$f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \quad (4.32)$$

Και υπόκειται στους περιορισμούς:

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \quad (4.33)$$

$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \quad (4.34)$$

$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \quad (4.35)$$

$$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \quad (4.36)$$

Όπου  $-10 \leq x_i \leq 10$  ( $i = 1, \dots, 7$ ). Η βέλτιστη τιμή βρίσκεται στο  $\vec{x}^* = (2.33049935147405174, 1.95137236847114592, -0.477541399510615805, 4.36572624923625874, -0.624486959100388983, 1.03813099410962173, 1.5942266780671519)$  και είναι ίση με  $f(\vec{x}^*) = 680.630057374402$ .

## **ΚΕΦΑΛΑΙΟ 5**

### **ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ** **ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΤΩΝ** **ΑΠΟΤΕΛΕΣΜΑΤΩΝ**



## **5.1. Πειραματική διαδικασία αλγορίθμου Global Competitive Ranking**

Για τον έλεγχο του αλγορίθμου Global Competitive Ranking (GCR) μελετήσαμε την επιρροή 4 διαφορετικών παραγόντων στην απόδοση των προβλημάτων. Οι παράγοντες αυτοί, καθώς και όλα τα αριθμητικά πειράματα που διενεργήθηκαν στο πλαίσιο της διπλωματικής εργασίας, υπεισήλθαν σε διερεύνηση υπό την καθοδήγηση του επιβλέποντα καθηγητή. Οι παράγοντες αυτοί είναι οι ακόλουθοι:

- Ο αριθμός των επαναλήψεων. Συγκεκριμένα, δοκιμάσαμε κατά πόσο μπορεί να επηρεαστεί το αποτέλεσμα του αλγορίθμου της μεθόδου GCR αλλά και κατά πόσο επιβαρύνεται ο υπολογιστικός χρόνος με την αύξηση ή την μείωση του αριθμού των επαναλήψεων. Οι τιμές, οι οποίες δοκιμάστηκαν, είναι οι 10,000, οι 20,000, οι 25,000 και οι 30,000 επαναλήψεις, χωρίς ιδιαίτερη διερεύνηση στις ενδιάμεσες τιμές αυτών.
- Ο αριθμός των σωματιδίων  $\mu$ , όπως αυτός έχει οριστεί στην υπο-παράγραφο 3.2.4, διατηρώντας σταθερή την σχέση  $\frac{\mu}{\lambda} = \frac{1}{7}$ . Έτσι, μεταβάλλοντας τον αριθμό  $\mu$  μεταβάλλουμε και τον συνολικό πληθυσμό των σωματιδίων ( $\lambda$ ). Πιο συγκεκριμένα, μελετήσαμε κατά πόσο μπορεί να επηρεαστεί το αποτέλεσμα του αλγορίθμου της μεθόδου GCR αλλά και κατά πόσο επιβαρύνεται ο υπολογιστικός χρόνος με την αύξηση ή την μείωση του πληθυσμού των σωματιδίων. Οι τιμές, οι οποίες δοκιμάστηκαν, είναι τα 30, τα 40, τα 50, τα 60 και τα 70 σωματίδια  $\mu$ , τα οποία μεταφράζονται αντίστοιχα σε 210, 280, 350, 420 και 490 σωματίδια  $\lambda$  (συνολικός πληθυσμός).
- Την πιθανότητα  $P_f$  (Pffl) η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση. Σύμφωνα με την βιβλιογραφία (Runarsson & Yao, 2003) η πιθανότητα αυτή πρέπει να λάβει τιμές στο διάστημα  $0 < P_f < 0.5$  προκειμένου να εγγυηθεί η εύρεση μιας εφικτής λύσης. Επίσης, όσο αυτή η πιθανότητα είναι πιο κοντά στο 0.5 τόσο μεγαλύτερη η έμφαση στην ελαχιστοποίηση της αντικειμενικής συνάρτησης. Η τιμή αυτή δεν είχε καμία επιρροή στην αύξηση ή τη μείωση του υπολογιστικού χρόνου του αλγορίθμου. Επηρεάζει, όμως, σημαντικά το

αποτέλεσμα. Για τους λόγους αυτούς, μελετήσαμε τις τιμές 0, 0.30, καθώς και το εύρος τιμών 0.450-0.499, στο οποίο θα πρέπει να βρίσκεται η βέλτιστη επιλογή με βάση τα όσα αναφέρθηκαν προηγουμένως.

- Τέλος, λάβαμε την απόφαση να δημιουργήσουμε και να μελετήσουμε έναν ακόμα παράγοντα, τον οποίο ονομάσαμε πολλαπλασιαστή βάσης (BaseMult). Με βάση την θεωρία, η αρχική ρύθμιση της μέσης τιμής βήματος υπολογίζεται από την σχέση (3.9):

$$\sigma_{i,j}^{(0)} = \frac{\delta x_j}{\sqrt{n}} = \frac{\bar{x}_j - x_j}{\sqrt{n}}, i \in \{1, \dots, \lambda\}, j \in \{1, \dots, n\} \quad (3.9)$$

Παρότι η σχέση αυτή δίδεται χωρίς κάποιον πολλαπλασιαστή, θεωρήθηκε σκόπιμο να εξεταστεί κατά πόσο το μέγεθος αυτό του βήματος μπορεί να επηρεάσει την τιμή του αποτελέσματος του προβλήματος ή όχι. Έτσι η σχέση (3.9) γίνεται:

$$\sigma_{i,j}^{(0)} = B * \frac{\delta x_j}{\sqrt{n}} = B * \frac{\bar{x}_j - x_j}{\sqrt{n}}, i \in \{1, \dots, \lambda\}, j \in \{1, \dots, n\} \quad (5.1)$$

Όπου, όπως είναι προφανές, B είναι ο πολλαπλασιαστής της βάσης. Πριν από την εκτέλεση των δοκιμών, αποφασίσαμε να πραγματοποιήσουμε δοκιμές με ένα εύρος αριθμών από το 0.01-1 καθώς και με τους αριθμούς 2 και 5.

Μέσα από το μεγάλο πλήθος δοκιμών, το οποίο πραγματοποιήσαμε, καταλήξαμε στο συμπέρασμα ότι η συμβολή που μπορεί να έχει ένας παράγοντας στο πρόβλημα δεν επηρεάζεται από την τιμή των άλλων παραγόντων αλλά όλοι μαζί συμβάλουν ανεξάρτητα στην εύρεση της καλύτερης δυνατής λύσης του προβλήματος. Για τον λόγο αυτό, οι δοκιμές, τις οποίες πραγματοποιήσαμε, έγιναν διατηρώντας κάθε φορά σταθερούς τους 3 από τους 4 παράγοντες και αναζητώντας την τιμή του 4<sup>ου</sup> παράγοντα, η οποία οδηγούσε στην καλύτερη δυνατή τιμή του προβλήματος.

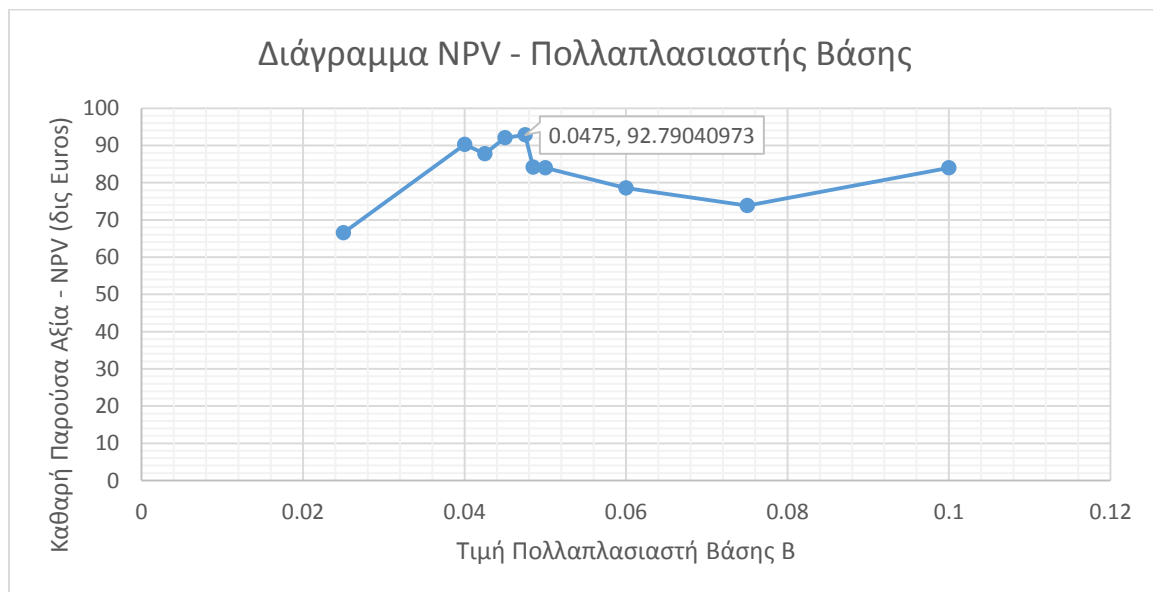
Όσον αφορά την μεθοδολογία, η οποία ακολουθήθηκε για τον αλγόριθμο GCR, οι δοκιμές, τις οποίες πραγματοποιήσαμε προκειμένου να βρούμε τις βέλτιστες τιμές των παραγόντων, τους οποίους περιγράψαμε στην παράγραφο αυτή, έγιναν στο 1<sup>ο</sup> πρόβλημα (παράγραφος 4.1) και στην συνέχεια τα υπόλοιπα προβλήματα (παράγραφοι 4.2, 4.3, 4.4, 4.5) χρησιμοποιήθηκαν προκειμένου να ελεγχθούν και να επαληθευτούν οι τιμές, στις οποίες καταλήξαμε.

## 5.2. Παρουσίαση αποτελεσμάτων αλγορίθμου Global Competitive Ranking

Στην παράγραφο αυτή θα παρουσιάσουμε με την μορφή διαγραμμάτων τα αποτελέσματα της αντικειμενικής συνάρτησης, τα οποία προέκυψαν για κάθε τιμή του παράγοντα, τον οποίο μελετάμε κάθε φορά.

### 5.2.1. Δοκιμές τιμής πολλαπλασιαστή βάσης

Επειδή, κατά τις αρχικές δοκιμές, τις οποίες πραγματοποιήσαμε με τον πολλαπλασιαστή βάσης ( $B=1$ ), υπήρχαν μεγάλα προβλήματα υπολογισμού, με τον αλγόριθμο να αδυνατεί να επιτύχει την εύρεση λύσης, οδηγηθήκαμε στην μελέτη του πολλαπλασιαστή βάσης, όπως περιγράψαμε προηγουμένως. Μετά από δοκιμές, προέκυψε ότι στο συγκεκριμένο πρόβλημα, προκειμένου να επιτευχθεί η εύρεση λύσης, ο παράγοντας αυτός πρέπει να βρίσκεται στο πεδίο τιμών 0.01-0.1 και για τον λόγο αυτό η λύση αναζητήθηκε σε αυτό το διάστημα. Οι τιμές, οι οποίες μελετήθηκαν, καθώς και τα αποτελέσματα της καθαρής παρούσας αξίας που προέκυψαν παρουσιάζονται στο σχήμα 5.1.



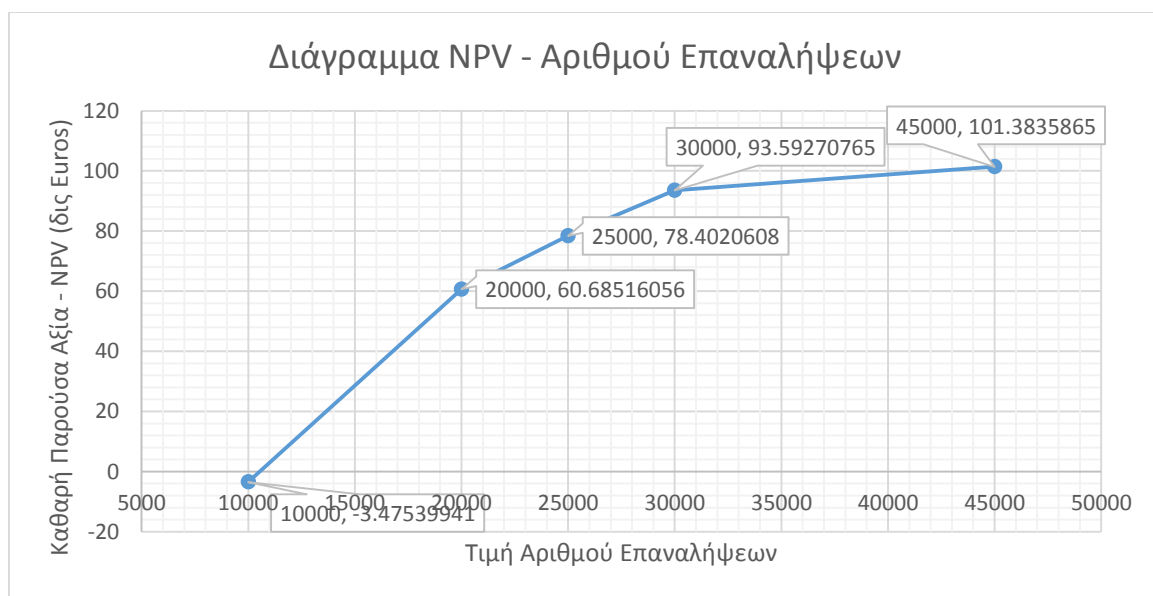
**Σχήμα 5.1:** Απεικόνιση επιρροής του Πολλ/στη Βάσης στην τιμή του NPV

Η τιμή, η οποία είναι εμφανής στο διάγραμμα ( $B=0.0475$ ), είναι η τιμή, η οποία προέκυψε ως η βέλτιστη τιμή για τον πολλαπλασιαστή βάσης μέσα από ένα σύνολο δοκιμών και

διατηρήθηκε σταθερή κατά την εκτέλεση όλων των υπόλοιπων μετρήσεων της διπλωματικής εργασίας.

### 5.2.2. Δοκιμές τιμής αριθμού επαναλήψεων

Η τιμή του αριθμού επαναλήψεων αποτέλεσε μία σημαντική απόφαση στην μελέτη της απόδοσης του αλγορίθμου GCR στο 1<sup>ο</sup> πρόβλημα. Εκτός από την επιρροή που έχει στην εύρεση της βέλτιστης λύσης της αντικειμενικής συνάρτησης, καθορίζει σε πολύ μεγάλο βαθμό (σε συνδυασμό με τον πληθυσμό των σωματιδίων) τον υπολογιστικό χρόνο, τον οποίο απαιτεί το λογισμικό MATLAB® για να εκτελέσει τον αλγόριθμο. Έτσι, εκτός από τα αποτελέσματα, λάβαμε υπόψιν μας και τον υπολογιστικό χρόνο, ο οποίος απαιτείται, προκειμένου να επιλέξουμε την βέλτιστη τιμή των επαναλήψεων. Τα αποτελέσματα των δοκιμών αποτυπώνονται στο σχήμα 5.2. Όλες οι δοκιμές, οι οποίες παρουσιάζονται γίνονται με αριθμό σωματιδίων  $\mu=30$ , εκτός από την δοκιμή των 45,000 επαναλήψεων, η οποία εκτελέστηκε με  $\mu=70$  σωματίδια, προκειμένου να φανεί εάν αυτή η ακραία τιμή θα αποφέρει σημαντική βελτίωση στην τιμή της καθαρής παρούσας αξίας.



**Σχήμα 5.2:** Απεικόνιση επιρροής του αριθμού επαναλήψεων στην τιμή του NPV

Όπως φαίνεται από το σχήμα 5.2, όσο μεγαλύτερη είναι η τιμή του αριθμού των επαναλήψεων τόσο μεγαλύτερο είναι και το αποτέλεσμα το οποίο μας δίνει ο αλγόριθμος ως βέλτιστο. Όμως, όπως φαίνεται από το σχήμα, όσο μεγαλώνει ο αριθμός των



επαναλήψεων, τόσο μικρότερος είναι ο ρυθμός βελτίωσης της τιμής του NPV. Δεδομένου ότι για την τιμή των 45,000 επαναλήψεων έχουμε χρησιμοποιήσει 70 σωματίδια αυξάνοντας πολύ σημαντικά τον χρόνο ο οποίος απαιτείται και παράλληλα επιτυγχάνοντας βελτίωση της τάξης του 8.32%, θεωρείται σκόπιμο να επιλέξουμε την τιμή των 30,000 επαναλήψεων προκειμένου να προχωρήσουμε στην περαιτέρω μελέτη των παραγόντων του προβλήματος. Η ποσοστιαία διαφορά της καθαρής παρούσας αξίας αλλά και του χρόνου που απαιτήθηκε προκειμένου αυτή να επιτευχθεί, φαίνεται στον πίνακα 5.1.

Επαναλήψεις	Καθαρή Παρούσα Αξία (δισ €)	Ποσοστό ΚΠΑ	Χρόνος (sec)	Ποσοστό Χρόνου
<b>45000</b>	101.3835865	108.32%	42732.37	326.62%
<b>30000</b>	93.59270765	100.00%	13083.35	100.00%
<b>25000</b>	78.4020608	83.77%	10831.59	82.79%
<b>20000</b>	60.68516056	64.84%	8761.487	66.97%
<b>10000</b>	-3.47539941	-3.71%	4399.047	33.62%

**Πίνακας 5.1:** Ποσοστιαίες μεταβολές ΚΠΑ και Χρόνου για διάφορους αριθμούς επαναλήψεων με τιμές αναφοράς τις τιμές για 30,000 επαναλήψεις

### 5.2.3. Δοκιμές τιμής αριθμού σωματιδίων μ

Μετά την εύρεση του αριθμού του πολλαπλασιαστή βάσης και του αριθμού των επαναλήψεων, εκτελέσαμε δοκιμές προκειμένου να βρούμε το βέλτιστο αριθμό σωματιδίων για το πρόβλημά μας αλλά και να αποτυπώσουμε το κατά πόσο μπορεί να επηρεάσει την τιμή της αντικειμενικής μας συνάρτησης αλλά και τον χρονικό παράγοντα. Υπενθυμίζουμε ότι σε όλες αυτές τις δοκιμές κρατάμε σταθερούς τους 3 άλλους παράγοντες που αναφέραμε και δοκιμάζουμε διαφορετικές τιμές μονάχα για τον αριθμό των σωματιδίων μ, ο οποίος επηρεάζει τον συνολικό πληθυσμό των σωματιδίων λ.

Έχοντας γνωστό τόσο τον πολλαπλασιαστή βάσης όσο και τον αριθμό των επαναλήψεων, αποφασίσαμε να εκτελέσουμε πολλαπλά πειράματα για κάθε τιμή, προκειμένου να μπορέσουμε να συμπεράνουμε εάν ο αλγόριθμος συγκλίνει στο ίδιο αποτέλεσμα για ίδια δεδομένα, κάθε φορά. Έτσι, πραγματοποιήσαμε και καταγράψαμε 10 ελέγχους για κάθε αριθμό σωματιδίων μ. Τα αποτελέσματα των μετρήσεων μαζί με την μέση τιμή, την τυπική

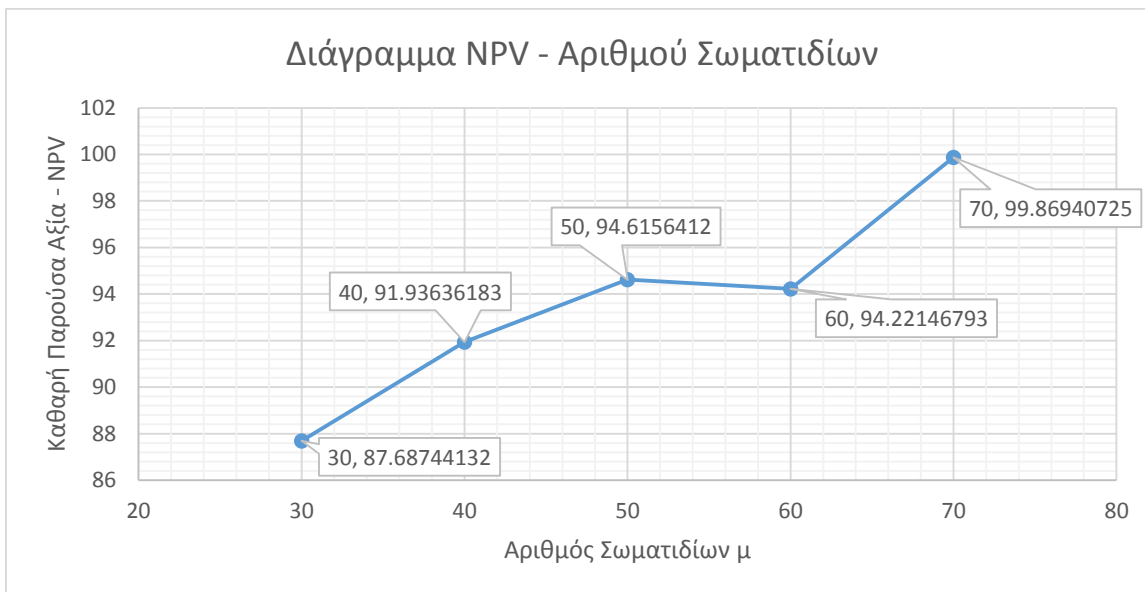
απόκλιση του δείγματος αλλά και το πιο συχνά εμφανιζόμενο αποτέλεσμα αποτυπώνονται στον πίνακα 5.2.

# Δοκιμής	Αριθμός σωματιδίων				
	30	40	50	60	70
<b>1</b>	92.7904	91.9364	94.6156	94.2215	99.8694
<b>2</b>	93.5927	91.9364	94.6156	94.2215	95.2519
<b>3</b>	89.2109	91.2390	94.6156	94.2215	99.8694
<b>4</b>	87.6874	91.9364	94.6156	94.2215	95.2519
<b>5</b>	87.6874	91.9364	94.6156	94.0242	99.8694
<b>6</b>	87.6874	91.9364	94.6156	94.2215	99.8694
<b>7</b>	85.2648	91.9364	94.6156	94.2215	99.8694
<b>8</b>	87.6874	92.3986	94.6156	94.2215	99.8694
<b>9</b>	86.1725	91.9364	94.6156	94.2215	95.2519
<b>10</b>	87.6874	91.9364	94.6156	96.5461	99.8694
<b>Μέση Τιμή</b>	88.5469	91.9128	94.6156	94.4342	98.4842
<b>Τυπική Απόκλιση</b>	2.666911	0.277781	0	0.744626	2.230473
<b>Συχνότερη Τιμή</b>	87.6874	91.9364	94.6156	94.2215	99.8694

**Πίνακας 5.2:** Αποτελέσματα δοκιμών αριθμού σωματιδίων

Από την μελέτη των αποτελεσμάτων παρατηρούμε ότι ο αριθμός συγκλίνει κατά μέσο όρο 8 στις 10 φορές σε μία συγκεκριμένη τιμή, ενώ υπάρχει περίπτωση η σύγκλιση να συμβεί και σε καλύτερη τιμή από την πιο συχνή.

Παρατηρώντας τα αποτελέσματα αυτά, εάν θεωρήσουμε ως τιμή σύγκλισης την συχνότερη τιμή, κατασκευάζουμε το διάγραμμα του σχήματος 5.3, το οποίο μας παρέχει μια αρκετά καλή απεικόνιση της επιρροής του αριθμού των σωματιδίων στην τιμή της καθαρής παρούσας αξίας (αντικειμενική μας συνάρτηση). Όπως είναι εμφανές, η τιμή της NPV βελτιώνεται όσο αυξάνονται τα σωματίδια, έως ότου αυτά φτάσουν την τιμή των 50. Από εκεί και πέρα παρατηρείται μια μικρή μείωση, ενώ στα 70 σωματίδια εμφανίστηκε 3 στις 10 φορές αρκετά χειρότερη τιμή από την συχνότερη. Μάλιστα, η τιμή αυτή (95.2519) εμφανίστηκε και τις 3 φορές που δεν επετεύχθη η συχνότερη τιμή και, επειδή είναι αρκετά κοντά στην τιμή των 50 σωματιδίων, μας οδηγεί να θεωρήσουμε αναξιόπιστο κάποιο αποτέλεσμα των 70 σωματιδίων. Ο μόνος τρόπος να ελεγχθούν τα αποτελέσματα από τα 70 σωματίδια και πάνω είναι να εκτελεστούν αρκετές διαφορετικές δοκιμές.



**Σχήμα 5.3:** Απεικόνιση επιρροής του αριθμού επαναλήψεων στην τιμή του NPV

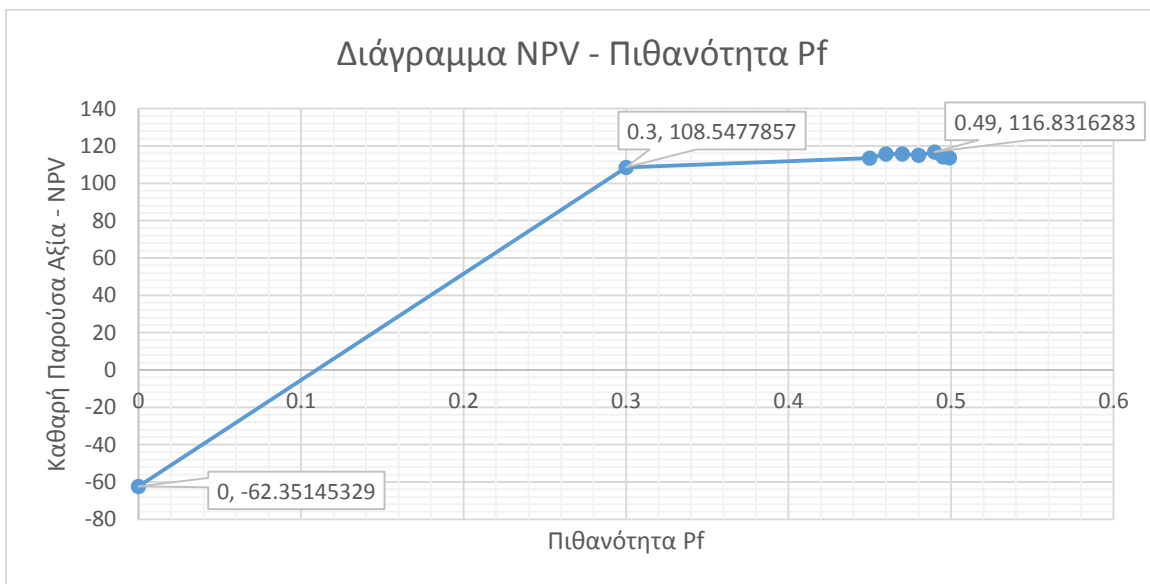
Εκτός από την βελτίωση που επιφέρει στα αποτελέσματα η αύξηση του αριθμού των σωματιδίων, είναι πολύ σημαντικό να αναφέρουμε και την επιβάρυνση που αυτή αποφέρει στον χρονικό παράγοντα του προβλήματος. Όσο αυτός αυξάνεται, τόσο αυξάνεται και ο υπολογιστικός χρόνος που απαιτείται για της εκτέλεση του αλγορίθμου. Θεωρώντας ως τιμές αναφοράς τις τιμές της μεσαίας τιμής των σωματιδίων που ελέγξαμε ( $\mu=50$ ), κατασκευάζουμε τον πίνακα 5.3, στον οποίο απεικονίζεται η ποσοστιαία διαφορά της αντικειμενικής συνάρτησης και του χρονικού παράγοντα σε σχέση με τις τιμές αναφοράς.

Αριθμός Σωματιδίων	Καθαρή Παρούσα Αξία (δισ €)	Ποσοστό ΚΠΑ	Χρόνος (sec)	Ποσοστό Χρόνου
30	87.68744132	92.68%	12792.68112	60.37%
40	91.93636183	97.17%	16727.83269	78.93%
50	94.6156412	100.00%	21191.94492	100.00%
60	94.22146793	99.58%	24444.41583	115.35%
70	99.86940725	105.55%	29682.99753	140.07%

**Πίνακας 5.3:** Ποσοστιαίες μεταβολές ΚΠΑ και Χρόνου για διάφορους αριθμούς σωματιδίων

#### 5.2.4. Δοκιμές τιμής πιθανότητας $P_f$

Ο τελευταίος παράγοντας, τον οποίο μελετήσαμε, είναι ο παράγοντας της πιθανότητας  $P_f$  η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση. Προκειμένου να καταλήξουμε σε αποτελέσματα με μεγαλύτερη ταχύτητα και επειδή, όπως εξηγήσαμε σε προηγούμενη παράγραφο, οι 4 αυτοί παράγοντες επηρεάζουν ανεξάρτητα μεταξύ τους την απόδοση της αντικειμενικής συνάρτησης, αποφασίσαμε να πραγματοποιήσουμε έλεγχο θεωρώντας 30,000 επαναλήψεις και 30 σωματίδια. Στην συνέχεια, αφού καταλήξαμε στην βέλτιστη τιμή της πιθανότητας  $P_f$  για τις τιμές αυτές, πραγματοποιήσαμε έλεγχο για 70 σωματίδια, προκειμένου να καταλήξουμε στην βέλτιστη δυνατή λύση. Τα αποτελέσματα των δοκιμών αποτυπώνονται στο διάγραμμα του σχήματος 5.4.



**Σχήμα 5.4:** Απεικόνιση επιρροής της πιθανότητας  $P_f$  στην τιμή του NPV

Στο διάγραμμα είναι ευδιάκριτες οι τιμές για το 0, για το 0.3 αλλά και για το 0.49 όπου βρίσκεται η βέλτιστη τιμή. Επειδή πραγματοποιήσαμε μεγάλο πλήθος δοκιμών στην περιοχή 0.45-0.499, τα αποτελέσματα παρατίθενται αναλυτικά στον πίνακα 5.4.

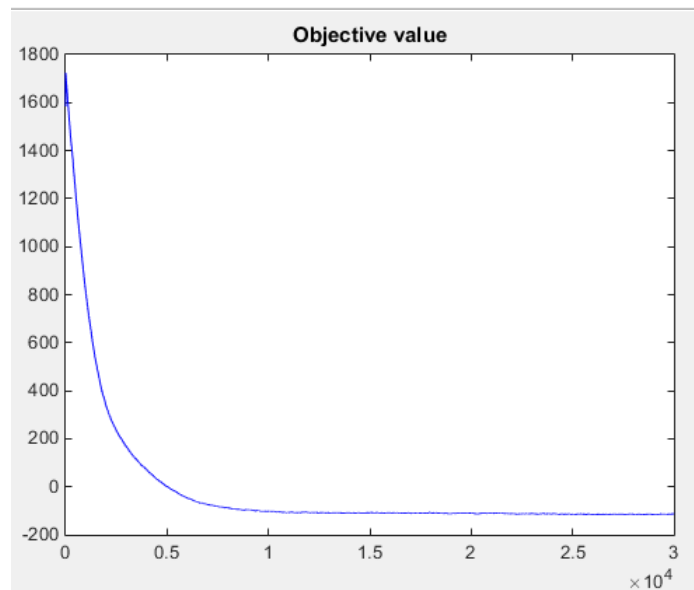
Pf	0	0.3	0.45	0.46	0.47	0.48	0.49	0.495	0.499
ΚΠΑ	-62.3515	108.5478	113.5665	115.7979	115.6368	115.0392	116.8316	114.2367	113.5873

**Πίνακας 5.4:** Συσχέτιση τιμών ΚΠΑ (NPV) και πιθανότητας  $P_f$

Τόσο από τον πίνακα 5.4 όσο και από το σχήμα 5.4 είναι προφανές ότι η βέλτιστη τιμή για την πιθανότητα η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση είναι ίση με  $P_f = 0.49$ . Εφόσον έχουμε καταλήξει στην τιμή αυτή, την χρησιμοποιούμε προκειμένου να προκύψει η βέλτιστη τιμή για την ΚΠΑ της αντικειμενικής συνάρτησης, η οποία προκύπτει ότι είναι για 70 σωματίδια 119.208 (δισ €).

### 5.2.5. Βέλτιστη επίλυση του προβλήματος

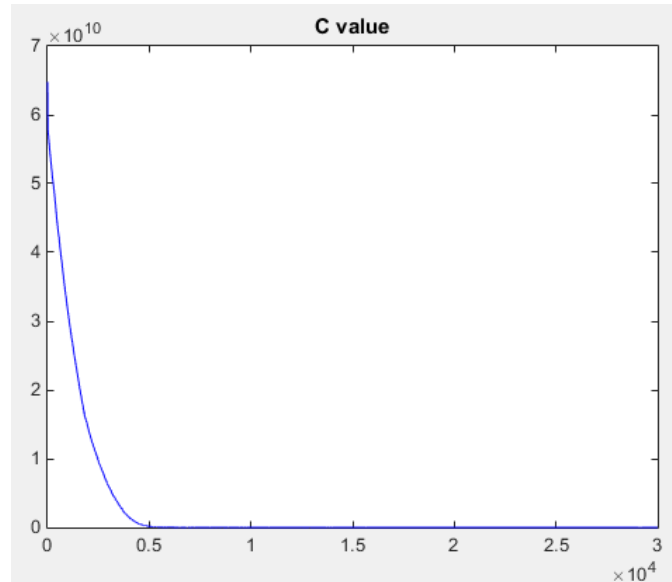
Από τα παραπάνω, γίνεται προφανές ότι η επίλυση του προβλήματος η οποία μπορεί να προκύψει από το εύρος των τιμών, τις οποίες μελετήσαμε, γίνεται βέλτιστη για τις τιμές:  $P_f = 0.49$ ,  $B = 0.0475$ ,  $ITERMAX = 30,000$ ,  $pa = 70$ . Μέσα από τις δοκιμές, προέκυψε η βέλτιστη τιμή που μπορεί να υπολογιστεί και με ισούται με  $maxNPV = 119.208$  (δισ €) εμφανίζοντας τιμή σφάλματος-παραβίασης περιορισμών  $minC = 0$ . Η λύση αυτή προκύπτει μέσα από τον αλγόριθμο GCR για τιμές μεταβλητών ίσες με αυτές που φαίνονται στον πίνακα 5.5 της επόμενης σελίδας. Η τιμή αυτή, απέχει αρκετά από την βέλτιστη τιμή η οποία προτείνεται από την βιβλιογραφία  $NPV_{best} = 177.6530$ . Η σύγκλιση της αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών, η οποία πραγματοποιείται από τον αλγόριθμο παρουσιάζονται στα σχήματα 5.5 και 5.6 αντίστοιχα.



**Σχήμα 5.5:** Σύγκλιση αντικειμενικής συνάρτησης 1<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις (Άξονας x: ΚΠΑ, Άξονας y: # επαναλήψεων)

Καύσιμο\Έτος	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Natural Gas	0.0026	0.0000	0.1308	0.0654	0.0020	0.0002	0.0072	0.0102	0.0489	0.0485	0.0482	0.0481	0.0494	0.0512	0.0545	0.0569	0.0605	0.0632	0.0662
Lignite	0.0168	0.0143	0.0100	0.0067	0.0046	0.0029	0.0042	0.0722	0.1227	0.1196	0.1166	0.1142	0.1144	0.1153	0.1161	0.1175	0.1204	0.1218	0.1227
Oil	0.0025	0.0014	0.0331	0.0021	0.0008	0.0220	0.0000	0.0053	0.0158	0.0152	0.0158	0.0154	0.0154	0.0156	0.0158	0.0161	0.0166	0.0170	0.0173
Hydro medium	0.2086	0.2040	0.1292	0.1661	0.1598	0.1540	0.1498	0.1466	0.1064	0.1051	0.1376	0.1340	0.1300	0.1281	0.1231	0.1443	0.1161	0.1345	0.1310
Hydro pumped	0.2114	0.1527	0.1168	0.1102	0.1037	0.0980	0.0964	0.0938	0.1344	0.0880	0.1266	0.0872	0.0856	0.0822	0.0779	0.1343	0.1285	0.1222	0.1195
Biomass	0.2951	0.2545	0.2477	0.2400	0.2369	0.2856	0.2816	0.1993	0.2298	0.2365	0.2252	0.2176	0.2099	0.2035	0.1976	0.1928	0.1890	0.1855	0.1805
Solar PV	0.0029	0.0309	0.0290	0.0279	0.0267	0.0368	0.0353	0.0352	0.0465	0.0489	0.0317	0.0328	0.0300	0.0276	0.0262	0.0245	0.0228	0.0206	0.0195
Wind Turbine	0.0000	0.1265	0.1225	0.1217	0.1193	0.1408	0.1372	0.1370	0.1600	0.1637	0.1485	0.1427	0.1370	0.0987	0.0956	0.1243	0.1208	0.0885	0.1159
$\Theta_1$	0.7218	0.7295	0.6391	0.6462	0.6413	0.6000	0.6000	0.6000	0.6000	0.6391	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000
$\Theta_2$	1.0000	1.0000	1.0000	1.0000	1.0000	0.9906	0.9612	0.9728	0.9189	0.9926	0.9350	0.9159	0.8262	0.7516	0.7500	0.7500	0.7500	0.7500	0.7500
$\Theta_3$	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300
Καύσιμο\Έτος	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Natural Gas	0.0678	0.0696	0.1054	0.1069	0.1095	0.1121	0.1174	0.1235	0.1279	0.1503	0.0246	0.0181	0.0222	0.1368	0.1397	0.1416	0.1033	0.0391	0.0396
Lignite	0.1232	0.1208	0.1525	0.1498	0.1479	0.1477	0.1485	0.1508	0.1537	0.1769	0.1799	0.1818	0.1837	0.1481	0.1474	0.1462	0.1049	0.0361	0.0369
Oil	0.0172	0.0173	0.0181	0.0179	0.0181	0.0181	0.0186	0.0188	0.0195	0.0199	0.0207	0.0212	0.0216	0.0221	0.0224	0.0227	0.0231	0.0235	0.0001
Hydro medium	0.1285	0.1286	0.1255	0.1407	0.1417	0.1404	0.1385	0.1080	0.1036	0.0990	0.0638	0.0099	0.0102	0.0106	0.0110	0.0114	0.0118	0.0122	0.0126
Hydro pumped	0.1165	0.1141	0.1124	0.1301	0.1288	0.1269	0.1270	0.0909	0.0856	0.0798	0.0385	0.0022	0.0086	0.0037	0.0003	0.0177	0.0000	0.0016	0.0011
Biomass	0.1732	0.1644	0.2053	0.1994	0.1922	0.1880	0.1837	0.1840	0.1846	0.2140	0.2121	0.2105	0.2076	0.1501	0.1520	0.1400	0.0703	0.0000	0.0009
Solar PV	0.0188	0.0172	0.0155	0.0130	0.0205	0.0184	0.0162	0.0141	0.0131	0.0117	0.0089	0.0077	0.0206	0.0024	0.0117	0.0014	0.0019	0.0221	0.0029
Wind Turbine	0.1131	0.1104	0.1075	0.1021	0.1179	0.1129	0.1088	0.1051	0.1021	0.0909	0.1514	0.0090	0.0046	0.0072	0.0019	0.0784	0.0736	0.0684	0.0392
$\Theta_1$	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6276	0.6000	0.6084	0.6718	0.8582	1.0000
$\Theta_2$	0.7500	0.7500	0.9119	0.9471	0.8853	0.8091	0.8141	0.8011	0.7500	0.7500	0.7500	0.7500	0.7806	0.7722	0.9748	1.0000	1.0000	1.0000	1.0000
$\Theta_3$	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300	0.0300

Πίνακας 5.5: Βέλτιστες τιμές των μεταβλητών του προβλήματος 1 με βάση την βιβλιογραφία



**Σχήμα 5.6:** Σύγκλιση τιμής σφάλματος  $1^{ου}$  προβλήματος για 30,000 επαναλήψεις

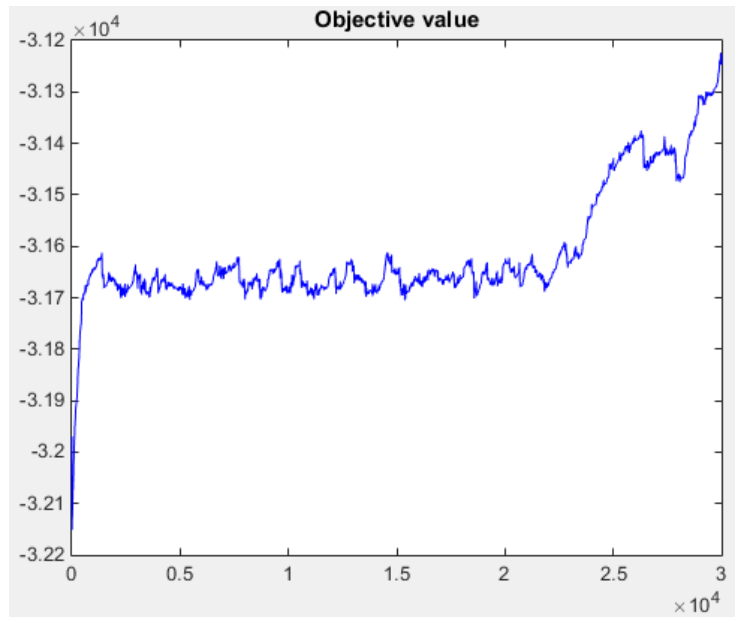
### 5.2.6. Επαλήθευση τιμών παραγόντων

Προκειμένου να επαληθεύσουμε ότι οι τιμές, οι οποίες έχουμε βρει ως τις καλύτερες, βρίσκουν εφαρμογή και σε άλλα προβλήματα, πραγματοποιήσαμε δοκιμές στα προβλήματα, τα οποία περιγράφουμε στις παραγράφους 4.2, 4.3, 4.4 και 4.5. Τα αποτελέσματα παρατίθενται αναλυτικά στην συνέχεια.

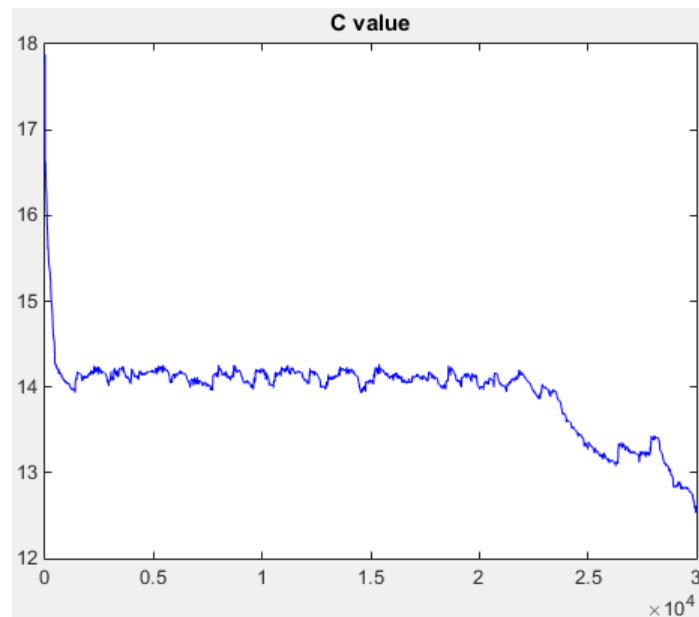
Για την επίλυση του  $2^{ου}$  προβλήματος με την εφαρμογή του αλγορίθμου GCR προέκυψε βέλτιστη τιμή για τιμές  $\vec{x}^* = (78, 33, 27, 44.9994, 34.8620)$ , η οποία είναι ίση με  $f(\vec{x}^*) = -31705$  ενώ το άθροισμα σφάλματος είναι  $C(\vec{x}^*) = 14.2642$ . Η τιμή αυτή είναι καλύτερη από την πρότυπη λύση που προτείνεται στην βιβλιογραφία. Οι συγκλίσεις της αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών, οι οποίες πραγματοποιούνται από τον αλγόριθμο, παρουσιάζονται στα σχήματα 5.7 και 5.8 αντίστοιχα.

Για την επίλυση του  $3^{ου}$  προβλήματος με την εφαρμογή του αλγορίθμου GCR προέκυψε βέλτιστη τιμή για τιμές  $\vec{x}^* = (2.14292151222768, 2.44714138525404, 8.78211365246749, 5.14235086748826, 1.00706508111444, 1.43959283775867, 1.28354191684823, 10, 8.25330877660024, 8.48641095058096)$ , η οποία είναι ίση με  $f(\vec{x}^*) = 24.6389$  ενώ το άθροισμα σφάλματος  $C(\vec{x}^*) = 0.0310$ . Η τιμή αυτή είναι

οριακά χειρότερη της λύσης της βιβλιογραφίας. Οι συγκλίσεις της αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών, οι οποίες πραγματοποιούνται από τον αλγόριθμο, παρουσιάζονται στα σχήματα 5.9 και 5.10 αντίστοιχα.

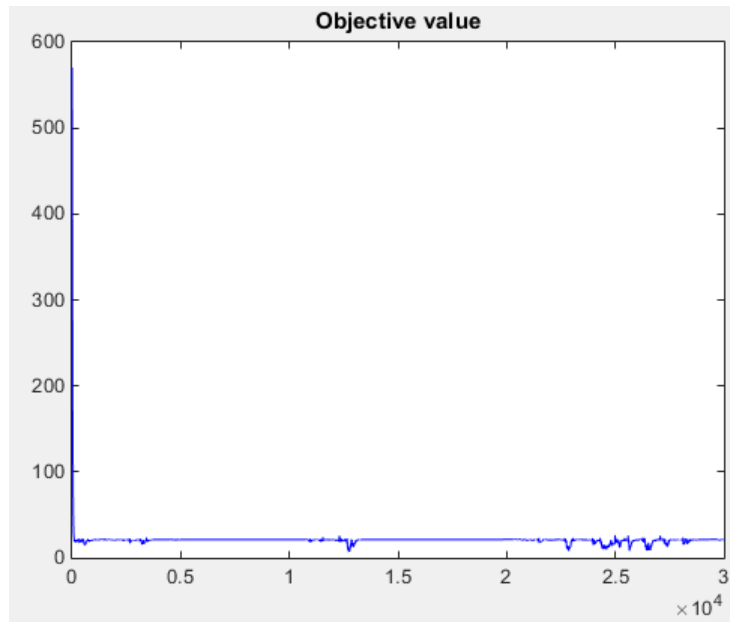


**Σχήμα 5.7:** Σύγκλιση αντικειμενικής συνάρτησης 2<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις

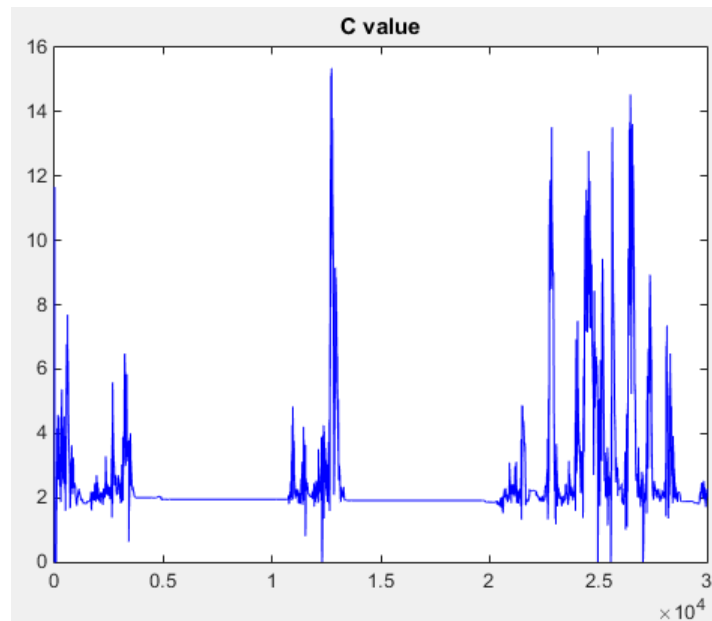


**Σχήμα 5.8:** Σύγκλιση τιμής σφάλματος 2<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις





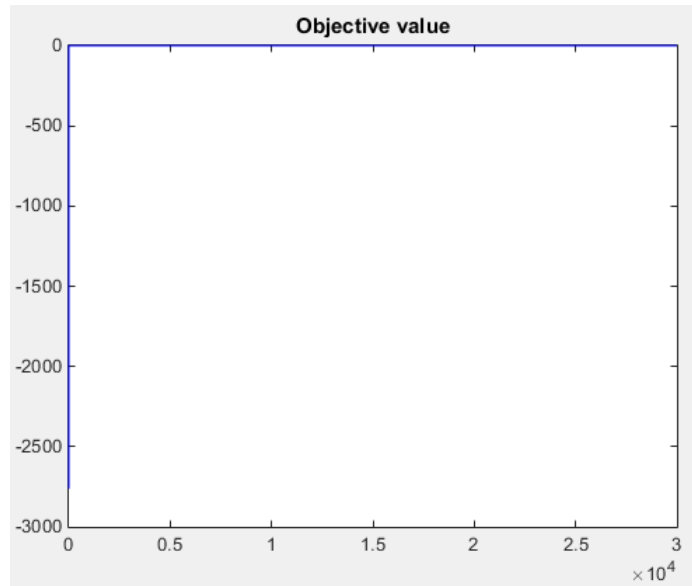
**Σχήμα 5.9:** Σύγκλιση αντικειμενικής συνάρτησης 3<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις



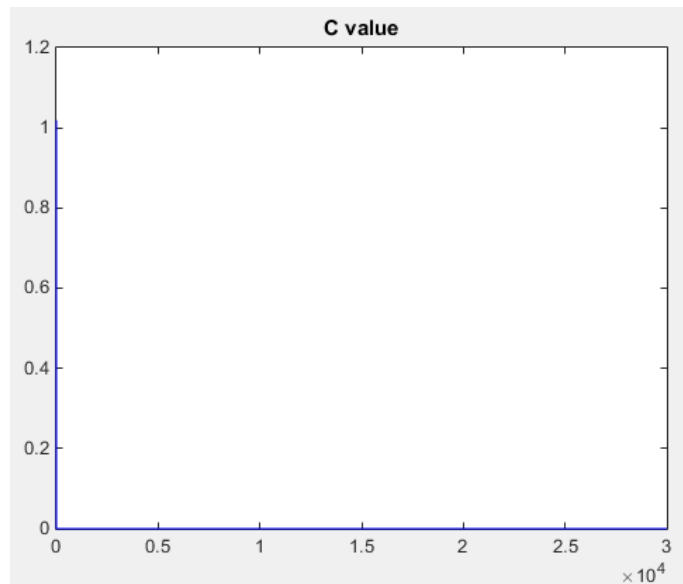
**Σχήμα 5.10:** Σύγκλιση τιμής σφάλματος 3<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις

Για την επίλυση του 4<sup>ου</sup> προβλήματος με την εφαρμογή του αλγορίθμου GCR προέκυψε βέλτιστη τιμή για διάνυσμα  $\vec{x}^* = (1.22264906597673, 4.24403434746040)$ , η οποία

είναι  $f(\bar{x}^*) = 0.1008$  ενώ το άθροισμα σφάλματος  $C(\bar{x}^*) = 0$ . Η τιμή αυτή είναι οριακά χειρότερη της λύσης της βιβλιογραφίας. Οι συγκλίσεις της αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών, οι οποίες πραγματοποιούνται από τον αλγόριθμο, παρουσιάζονται στα σχήματα 5.11 και 5.12 αντίστοιχα.

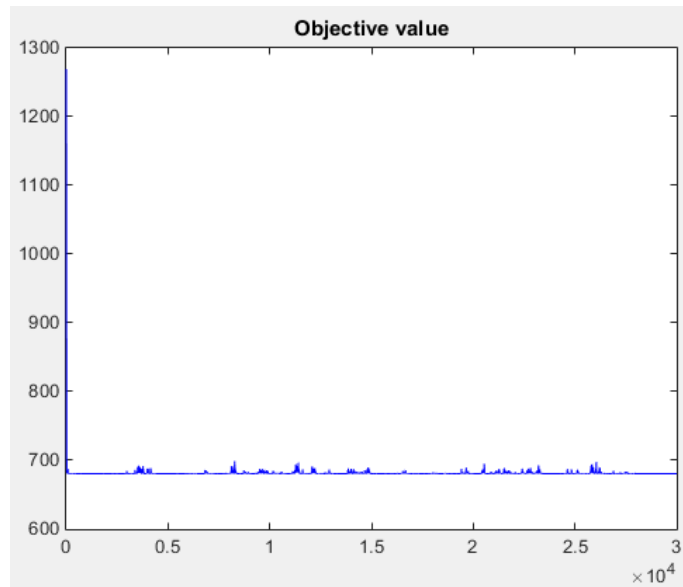


**Σχήμα 5.11:** Σύγκλιση αντικειμενικής συνάρτησης 4<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις

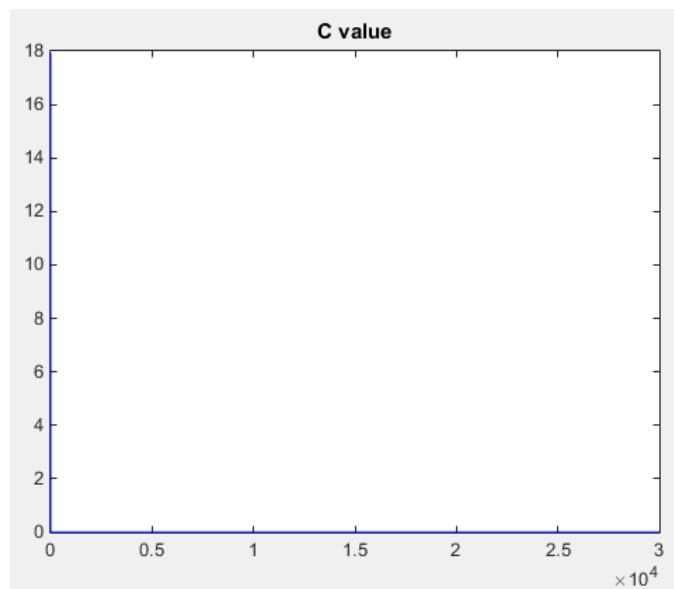


**Σχήμα 5.12:** Σύγκλιση τιμής σφάλματος 4<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις

Για την επίλυση του 5<sup>ου</sup> προβλήματος με την εφαρμογή του αλγορίθμου GCR προέκυψαν βέλτιστες τιμές για διάνυσμα  $\vec{x}^* = (2.32968033955636, 1.95301791998715, -0.48096462620205, 4.3617150947274, -0.62365267737835, 1.0401250268313, 1.59431615082263)$  με  $f(\vec{x}^*) = 680.6301$  και άθροισμα σφάλματος  $C(\vec{x}^*) = 0$ .



**Σχήμα 5.13:** Σύγκλιση αντικειμενικής συνάρτησης 5<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις



**Σχήμα 5.14:** Σύγκλιση τιμής σφάλματος 5<sup>ου</sup> προβλήματος για 30,000 επαναλήψεις

Η τιμή αυτή είναι ίδια με την λύση της βιβλιογραφίας. Οι συγκλίσεις της αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών, οι οποίες πραγματοποιούνται από τον αλγόριθμο, παρουσιάζονται στα σχήματα 5.13 και 5.14 αντίστοιχα.

### 5.2.7. Ανάλυση αποτελεσμάτων

Από την εκτέλεση των δοκιμών που πραγματοποιήσαμε μπορούμε να εξάγουμε διάφορα συμπεράσματα για τον τρόπο με τον οποίο οι 4 παράγοντες που μελετήσαμε μπορούν να επηρεάσουν την αναζήτηση βέλτιστης λύσης σε ένα μεγάλο μη-γραμμικού πρόβλημα με την μέθοδο των EA και συγκεκριμένα της GCR.

Αρχικά, όσον αφορά τον αριθμό των επαναλήψεων, είναι εμφανές ότι, όσο μεγαλύτερος είναι ο αριθμός αυτός, τόσο μεγαλύτερη είναι η πιθανότητα ο αλγόριθμός μας να συγκλίνει στην βέλτιστη δυνατή λύση του προβλήματος. Μετά, όμως, από ένα συγκεκριμένο αριθμό επαναλήψεων, ο οποίος ορίζεται διαφορετικά σε κάθε πρόβλημα, η αύξηση της τιμής της αντικειμενικής συνάρτησης είναι δυσανάλογη της αύξησης του αριθμού των επαναλήψεων και άρα του υπολογιστικού χρόνου. Ακόμη, μπορούμε να σημειώσουμε ότι σε ένα σωστά κωδικοποιημένο αλγόριθμο GCR, ο αριθμός των επαναλήψεων αυξάνει αναλογικά με τον υπολογιστικό χρόνο που απαιτείται ενώ σε άλλους αλγορίθμους ή σε άλλους παράγοντες, όπως ο αριθμός σωματιδίων στον συγκεκριμένο αλγόριθμο, μπορεί ο χρόνος αυτός να επιβαρύνεται δυσανάλογα.

Επίσης, όσον αφορά τον αριθμό των σωματιδίων, όπως και ο αριθμός των επαναλήψεων, όσο μεγαλύτερος είναι, τόσο καλύτερο είναι και το αποτέλεσμα, το οποίο μας δίνει ο αλγόριθμος. Αντίστοιχα, όσο μεγαλύτερος είναι ο αριθμός τους, τόσο μεγαλύτερος είναι και ο υπολογιστικός χρόνος. Όμως, σε αντίθεση με τον αριθμό των επαναλήψεων, η επιβάρυνση αυτή είναι δυσανάλογη και πολλές φορές απρόβλεπτη. Για παράδειγμα, στο πρόβλημα 1, η αύξηση των σωματιδίων από 30 σε 40 επιβαρύνει περίπου 30%, η αύξηση από 40 σε 50 επιβαρύνει 27%, η αύξηση από 50 σε 60 επιβαρύνει 15% και η αύξηση από 60 σε 70 επιβαρύνει 21%. Από την άλλη, ο ρυθμός αύξησης της τιμής της αντικειμενικής συνάρτησης μειώθηκε όσο αυξάνονταν τα σωματίδια, φτάνοντας μέχρι και αρνητική τιμή για τα 60 σωματίδια. Όμως, μετά τα 60 σωματίδια η λύση βελτιώθηκε αρκετά αλλά έκανε τον αλγόριθμο σχετικά ασταθή, οδηγώντας σε αστοχία εύρεσης της βέλτιστης δυνατής λύσης σε ποσοστό 30%.

Ακόμη, όσον αφορά τον παράγοντα της πιθανότητας  $P_f$  η σύγκριση να γίνει με βάση μόνο την αντικειμενική συνάρτηση, αυτός αποτελεί έναν πολύ σημαντικό παράγοντα, ο οποίος μας συμφέρει να οριστεί σωστά, καθώς δεν έχει κανένα κόστος υπολογιστικού χρόνου. Για τον λόγο αυτό, οι δοκιμές που γίνονται για την πιθανότητα αυτή πρέπει να είναι αρκετά ενδελεχείς, καθώς, όπως φάνηκε από την ανάλυση που κάναμε, μπορεί να προσφέρει αρκετά μεγάλη βελτίωση ( $maxNPV \approx -62$  για  $P_f = 0$ ,  $maxNPV \approx 108$  για  $P_f = 0.3$  και  $maxNPV \approx 116$  για  $P_f = 0.49$ ). Όπως συμβαίνει και με τους άλλους παράγοντες, όταν η τιμή της πιθανότητας αυτής ξεπεράσει ένα συγκεκριμένο σημείο, ο ρυθμός αύξησης της τιμής της αντικειμενικής σε σχέση με την πιθανότητα  $P_f$  μειώνεται.

Τέλος, ο παράγοντας της τιμής του πολλαπλασιαστική βάσης  $B$ , τον οποίο ορίσαμε εμείς, καθώς δεν υπήρχε κάποια παρόμοια βιβλιογραφική πηγή που να αναφέρεται στην μελέτη αυτού, αποδείχθηκε ότι μπορεί να συμβάλει σημαντικά στην εύρεση της βέλτιστης λύσης του προβλήματός μας. Ειδικότερα, για το 1<sup>ο</sup> πρόβλημα, υπήρχε δυσκολία εύρεσης λύσης για τιμές μεγαλύτερες του 0.1, παρότι στην βιβλιογραφία ο παράγοντας αυτός δεν υφίσταται και παίρνει την τιμή 1. Στην συνέχεια, φάνηκε ότι ο παράγοντας αυτός μπορεί να αποφέρει διαφορά στην τιμή 25.75% ακόμα και για πολύ μικρή διαφορά τιμής, της τάξης του 0.03. Ακόμη, είναι πολύ σημαντικό να επισημάνουμε το γεγονός ότι ο παράγοντας αυτός δεν επηρεάζει καθόλου τον χρονικό παράγοντα του αλγορίθμου, παρά την πολύ μεγάλη συμβολή την οποία, όπως αποδείξαμε, μπορεί να έχει.

Ο επιτυχής έλεγχος, τον οποίο πραγματοποιήσαμε στην συνέχεια, προκειμένου να επαληθεύσουμε τις τιμές των παραγόντων, στις οποίες καταλήξαμε, μέσα από τις δοκιμές που εκτελέσαμε στο 1<sup>ο</sup> πρόβλημα, μας βοηθά να καταλήξουμε στο συμπέρασμα ότι οι τιμές αυτές μπορούν να χρησιμοποιηθούν ως οδηγός και τιμές αναφοράς για τον έλεγχο επίλυσης παρόμοιων προβλημάτων ή επίλυσης του ίδιου προβλήματος με διαφορετικές μεθόδους βελτιστοποίησης. Βέβαια, το γεγονός ότι ο αλγόριθμός μας δεν κατάφερε να καταλήξει στην εύρεση ίδιας ή καλύτερης λύσης σε σχέση με αυτή, η οποία είχε βρεθεί με την βιβλιογραφία, μας οδηγεί στο γεγονός να αμφιβάλουμε για το κατά πόσο το Global Competitive Ranking μπορεί να χρησιμοποιηθεί αποτελεσματικά για την επίλυση πολύ μεγάλων προβλημάτων, με μεγάλο πλήθος μεταβλητών και περιορισμών.

### **5.3. Πειραματική διαδικασία αλγορίθμου Self-Learning Particle Swarm Optimizer**

Για τον έλεγχο του αλγορίθμου Self-Learning Particle Swarm Optimizer μελετήσαμε την επιρροή 3 διαφορετικών παραγόντων στην απόδοση προβλημάτων. Οι παράγοντες αυτοί, καθώς και όλα τα αριθμητικά πειράματα που διενεργήθηκαν στο πλαίσιο της διπλωματικής εργασίας, υπεισήλθαν σε διερεύνηση υπό την καθοδήγηση του επιβλέποντα καθηγητή. Οι παράγοντες αυτοί είναι οι ακόλουθοι:

- Ο παράγοντας αρχικού  $\lambda$  ( $\lambda_{init}$  εξ. 3.35). Συγκεκριμένα, ερευνήσαμε ποιες είναι οι τιμές για τις οποίες ο αλγόριθμος SLPSO οδηγεί το πρόβλημα σε σύγκλιση αλλά και κατά πόσο μπορεί να επηρεαστεί το αποτέλεσμα του αλγορίθμου με την μεταβολή του αρχικού  $\lambda$ . Κατά τις δοκιμές, τις οποίες εκτελέσαμε, δοκιμάσαμε το εύρος τιμών 0.01-5 καθότι παρατηρήσαμε ότι μικρότερες ή μεγαλύτερες τιμές δεν οδηγούσαν τον αλγόριθμο σε σύγκλιση. Ανεξάρτητα από την τιμή του, ο παράγοντας αυτός δεν επιβαρύνει χρονικά την εκτέλεση του αλγορίθμου.
- Ο παράγοντας αρχικής τυπικής απόκλισης (INITStD) με βάση την πιθανο-θεωρητική αντιμετώπιση (παράγραφος 3.3.5). Συγκεκριμένα, ερευνήσαμε ποιες είναι οι τιμές για τις οποίες ο αλγόριθμος SLPSO οδηγεί το πρόβλημα σε σύγκλιση αλλά και κατά πόσο μπορεί να επηρεαστεί το αποτέλεσμα του αλγορίθμου με την μεταβολή της αρχικής τυπικής απόκλισης. Κατά την εκτέλεση των δοκιμών, δοκιμάσαμε το εύρος τιμών 0.1-25,000 με τις βέλτιστες τιμές να βρίσκονται συνήθως -αλλά όχι πάντα- για μικρές τιμές της.. Ανεξάρτητα από την τιμή του, ο παράγοντας αυτός δεν επιβαρύνει χρονικά την εκτέλεση του αλγορίθμου.
- Ο παράγοντας ρυθμού μείωσης τυπικής απόκλισης (ISTDREDUCTRATEFINAL) με βάση την πιθανο-θεωρητική αντιμετώπιση (παράγραφος 3.3.5). Συγκεκριμένα, ερευνήσαμε ποιες είναι οι τιμές για τις οποίες ο αλγόριθμος SLPSO οδηγεί το πρόβλημα σε σύγκλιση αλλά και κατά πόσο μπορεί να επηρεαστεί το αποτέλεσμα του αλγορίθμου με την μεταβολή της αρχικής τυπικής απόκλισης. Κατά την εκτέλεση των δοκιμών δοκιμάσαμε τις ακέραιες τιμές 1, 2, 3 και 4 καθότι φάνηκε ότι μεγαλύτερες ή μικρότερες τιμές δεν έχουν το επιθυμητό αποτέλεσμα ενώ προκειμένου να παρατηρηθεί η απαραίτητη μεταβολή στο πρόβλημα οι τιμές

έπρεπε να έχουν μια συγκεκριμένη απόσταση. Ανεξάρτητα από την τιμή του, ο παράγοντας αυτός δεν επιβαρύνει χρονικά την εκτέλεση του αλγορίθμου.

Μέσα από το μεγάλο πλήθος δοκιμών, το οποίο πραγματοποιήσαμε, καταλήξαμε στο συμπέρασμα ότι η συμβολή, την οποία μπορεί να έχει ένας παράγοντας στο πρόβλημα, δεν επηρεάζεται από την τιμή των άλλων παραγόντων αλλά όλοι μαζί συμβάλλουν, ανεξάρτητα, στην εύρεση της καλύτερης δυνατής λύσης του προβλήματος. Για τον λόγο αυτό, οι δοκιμές, τις οποίες πραγματοποιήσαμε, έγιναν διατηρώντας κάθε φορά σταθερούς τους 2 από τους 3 παράγοντες και αναζητώντας την τιμή του 3<sup>ου</sup> παράγοντα, η οποία οδηγούσε στην καλύτερη δυνατή τιμή του προβλήματος. Επίσης, πολύ σημαντική για την επίλυση του προβλήματος είναι η διατήρηση της τιμής, την οποία λαμβάνει το  $\lambda$  μετά το πέρας των επαναλήψεων - ποιο είναι, δηλαδή, το βέλτιστο  $\lambda$ , το οποίο προκύπτει.

Όσον αφορά την μεθοδολογία για τον αλγόριθμο SLPSO, οι δοκιμές, τις οποίες πραγματοποιήσαμε, προκειμένου να βρούμε τις βέλτιστες τιμές των παραγόντων, τους οποίους περιγράψαμε στην παράγραφο αυτή, έγιναν στα προβλήματα 4.2, 4.3, 4.4 και 4.5, στα οποία ελέγχθηκε η αποτελεσματικότητα του αλγορίθμου αλλά και ο τρόπος με τον οποίο οι διάφορες τιμές οδηγούν στην επίλυση του κάθε προβλήματος.

Υιοθετήθηκαν 2 βήματα στην μεθοδολογία για την εύρεση της βέλτιστης δυνατής λύσης, διατηρώντας σταθερό τον αριθμό των επαναλήψεων στις 2,000 καθ' όλη την διάρκεια των δοκιμών. Αρχικά μελετήθηκαν τα αποτελέσματα, τα οποία προέκυψαν από έναν κύκλο επαναλήψεων της επαναληπτικής διαδικασίας. Στην συνέχεια, μελετήθηκαν τα αποτελέσματα, τα οποία προέκυψαν από την εκτέλεση 30 κύκλων της επαναληπτικής διαδικασίας. Σε κάθε έναν από τους 30 κύκλους χρησιμοποιούνται οι ίδιες αρχικές τιμές, η ίδια διαδικασία αρχικοποίησης, δηλαδή, εκτελείται μία φορά και ο αλγόριθμος εκμεταλλεύεται τις τιμές προκειμένου να καταλήξει σε 30 διαφορετικές τιμές. Στο τέλος, προκύπτει ένας συγκεντρωτικός πίνακας αποτελεσμάτων, στον οποίο αποτυπώνονται τα στατιστικά μεγέθη, τα οποία προέκυψαν από το πρόβλημα. Μέσα από την στατιστική ανάλυση αυτή προκύπτει η μέση τιμή, η τυπική απόκλιση, η μέγιστη και η ελάχιστη τιμή της τιμής της αντικειμενικής συνάρτησης, της τιμής της συνάρτησης ποινής και της τιμής της αντικειμενικής όταν σε αυτήν εφαρμόσουμε την συνάρτηση ποινής. Τέλος, επειδή τα

αποτελέσματα είναι διαφορετικά για κάθε πρόβλημα, η παράθεσή τους γίνεται με βάση το πρόβλημα μελέτης κάθε φορά και όχι με τις τιμές των παραγόντων, όπως στην GCR.

#### **5.4. Παρουσίαση αποτελεσμάτων του αλγορίθμου Self-Learning Particle Swarm Optimizer**

##### **5.4.1. Αποτελέσματα 2<sup>ου</sup> προβλήματος**

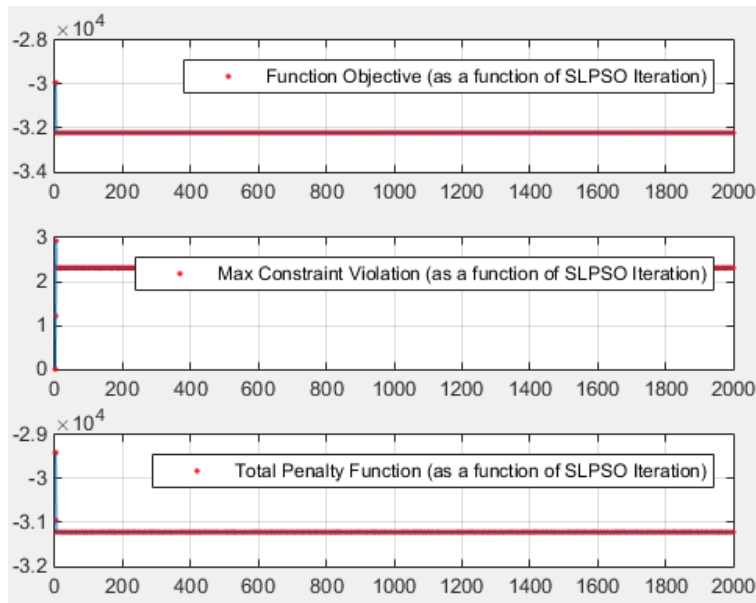
Για την επίλυση του 2<sup>ου</sup> προβλήματος με την εφαρμογή του αλγορίθμου SLPSO και 1 κύκλο επαναλήψεων προέκυψαν βέλτιστες τιμές για  $\vec{x}^* = (78, 33.0327264595735, 27, 44.9957767572682, 27)$  με  $f(\vec{x}^*) = -31812.12$  και άθροισμα σφάλματος  $C(\vec{x}^*) = 2.309618$ . Η λύση αυτή είναι καλύτερη από την πρότυπη λύση που προτείνεται στην βιβλιογραφία αλλά χειρότερη από την λύση, η οποία προέκυψε από την GCR, παρουσιάζει όμως μικρότερη παραβίαση περιορισμών. Η σύγκλιση της απλής αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών αλλά και της συνάρτησης ποινής, η οποία πραγματοποιείται από τον αλγόριθμο, παρουσιάζεται στο σχήμα 5.15.

Προκειμένου να καταλήξουμε στα αποτελέσματα αυτά έχουμε προβεί σε ένα πολύ μεγάλο πλήθος δοκιμών για τους 3 παράγοντες τους οποίους αναλύσαμε. Συγκεκριμένα, μελετήσαμε τιμές για το αρχικό λάμδα από 0.1 έως και 5, τιμές για την αρχική τυπική απόκλιση από 0.1 έως και 25,000 καθώς και τις τιμές 1, 2, 3 και 4 για τον ρυθμό μείωσης της τυπικής απόκλισης. Από τις δοκιμές αυτές καταλήξαμε ότι οι βέλτιστες τιμές για τους παράγοντες αυτούς είναι  $\lambda_{init} = 0.3$ ,  $INITStD = 0.6$  και  $ISTDREDUCTRATEFINAL = 1$  ή 2 ή 3 (χωρίς αλλοίωση του αποτελέσματος).

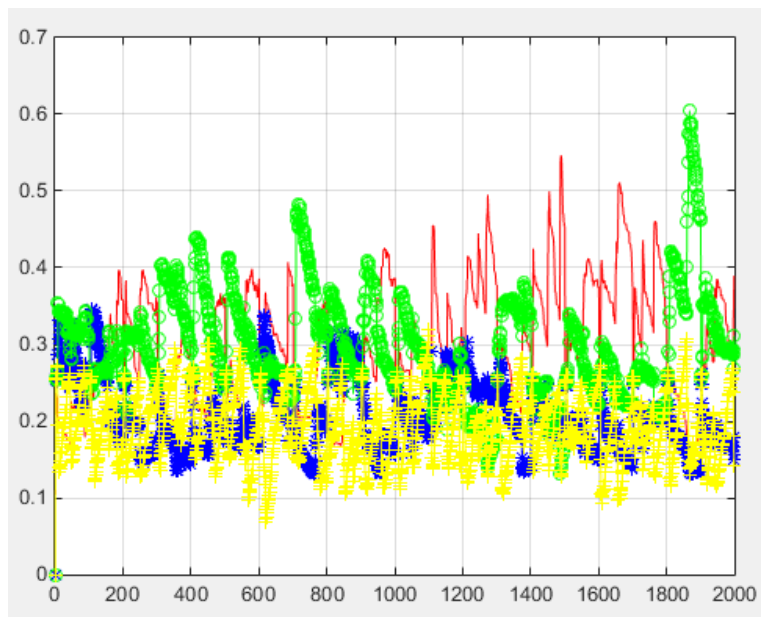
Το λάμδα από το οποίο προκύπτει σύγκλιση του προβλήματος και καλύτερη τιμή από αυτή που προτείνεται στην βιβλιογραφία βρίσκεται στο εύρος τιμών από 170 έως 460, με την βέλτιστη τιμή της αντικειμενικής να προκύπτει για τιμή  $\lambda=175.487080836566$ .

Ακόμη, μας ενδιαφέρει η εξέλιξη του ποσοστού επιλογής των 4 παραγόντων (a,b,c και d) της SLPSO κατά την διάρκεια της εκτέλεσης των 2000 επαναλήψεων. Αυτή φαίνεται στο





**Σχήμα 5.15:** Αποτελέσματα συναρτήσεων 2<sup>ου</sup> προβλήματος για 1 κύκλο επαναλήψεων σχήμα 5.16 (κόκκινο χρώμα και απλή γραμμή: παράγοντας a, μπλε χρώμα και \* σε κάθε σημείο: παράγοντας b, πράσινο χρώμα και κύκλος σε κάθε σημείο: παράγοντας c, κίτρινο χρώμα και + σε κάθε σημείο: παράγοντας d).

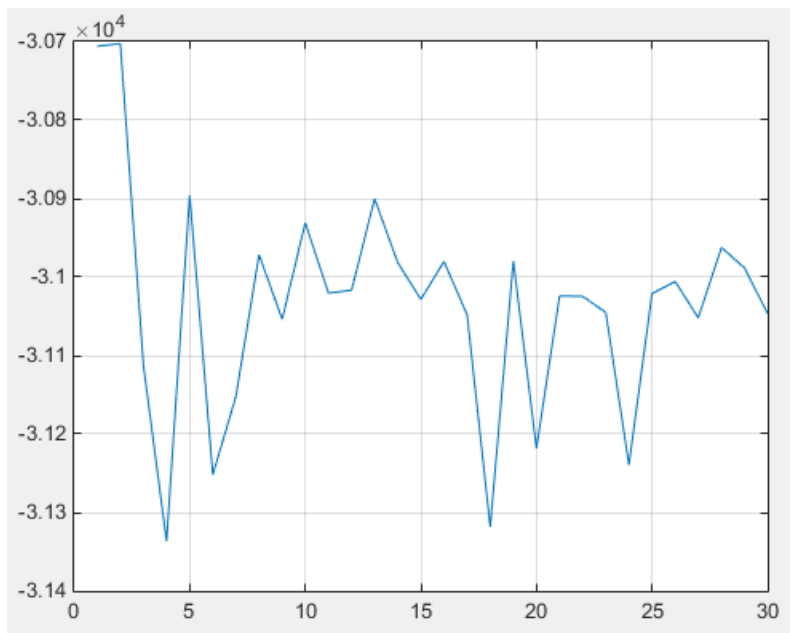


**Σχήμα 5.16:** Αποτελέσματα ρυθμού επιλογής παραγόντων SLPSO 2<sup>ου</sup> προβλήματος για 1 κύκλο επαναλήψεων

Στην συνέχεια εκτελούμε δοκιμές στο πρόβλημα για αριθμό κύκλων επαναλήψεων της επαναληπτικής διαδικασίας ίσο με 30, διατηρώντας σταθερές κάθε φορά τις τιμές που έχουν προκύψει μέσα από την αρχικοποίηση του προβλήματος, προκειμένου να μπορέσουμε να κατανοήσουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος. Από τις δοκιμές αυτές προκύπτουν τα ακόλουθα αποτελέσματα (πίνακας 5.6), ενώ η τιμή που λαμβάνει η συνάρτηση κατά την διάρκεια της εκτέλεσης των επαναλήψεων αποτυπώνεται στο σχήμα 5.17.

	Μέση τιμή	Τυπική Απόκλιση	Ελάχιστη τιμή	Μέγιστη τιμή
<b>Αντικειμενική Συνάρτηση</b>	-31034	144	-31337	-30703
<b>Άθροισμα Περιορισμών</b>	1	0	0	1
<b>Συνάρτηση Ποινής</b>	-30663	1	-30665	-30660

**Πίνακας 5.6:** Στατιστική ανάλυση αποτελεσμάτων 2<sup>ου</sup> προβλήματος για 30 κύκλους



**Σχήμα 5.17:** Αποτέλεσμα αντικειμενικής συνάρτησης 2<sup>ου</sup> προβλήματος για 30 κύκλους

### 5.4.2. Αποτελέσματα 3<sup>ου</sup> προβλήματος

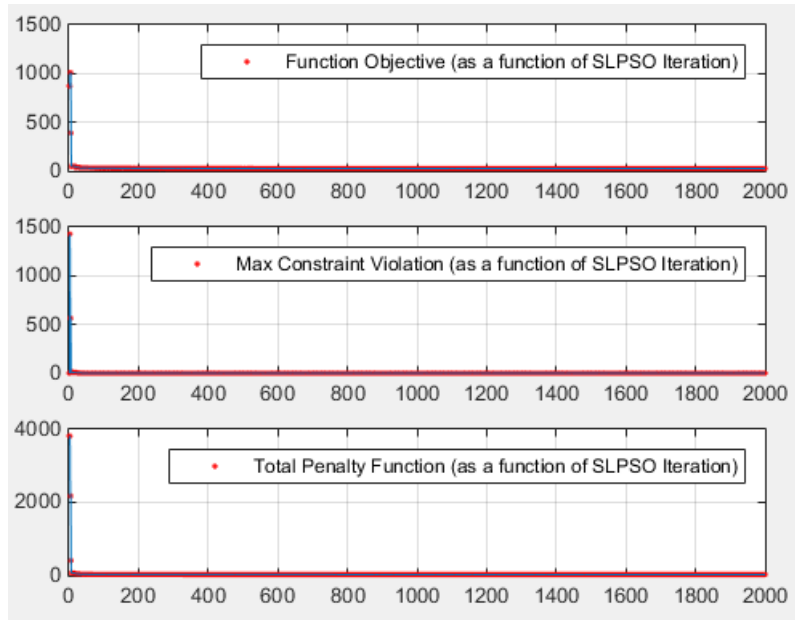
Για την επίλυση του 3<sup>ου</sup> προβλήματος με την εφαρμογή του αλγορίθμου SLPSO και 1 κύκλο επαναλήψεων προέκυψαν βέλτιστες τιμές για τιμές του  $\vec{x}^* = (2.17860629005448, 2.37761123833518, 8.72714303685090, 5.10624352476302, 0.976716207572770, 1.40677378222332, 1.31924517916190, 9.81778872289184, 8.31596971098212, 8.43511275355379)$  με  $f(\vec{x}^*) = 24.34885$  και άθροισμα σφάλματος  $C(\vec{x}^*) = 0.2227339$ . Η λύση αυτή είναι λίγο χειρότερη από την πρότυπη λύση που προτείνεται στην βιβλιογραφία αλλά χειρότερη από την λύση, η οποία προέκυψε από την GCR, ενώ παρουσιάζει μικρότερη παραβίαση περιορισμών. Η σύγκλιση της απλής αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών αλλά και της συνάρτησης ποινής μέσω του αλγορίθμου παρουσιάζεται στο σχήμα 5.18.

Προκειμένου να καταλήξουμε στα αποτελέσματα αυτά, έχουμε προβεί σε ένα πολύ μεγάλο πλήθος δοκιμών για τους 3 παράγοντες τους οποίους αναλύσαμε. Συγκεκριμένα, μελετήσαμε τιμές για το αρχικό λάμδα από 0.1 έως και 5, τιμές για την αρχική τυπική απόκλιση από 0.1 έως και 25,000 καθώς και τις τιμές 1, 2, 3 και 4 για τον ρυθμό μείωσης της τυπικής απόκλισης. Από τις δοκιμές αυτές καταλήξαμε ότι οι βέλτιστες τιμές για τους παράγοντες αυτούς είναι  $\lambda_{init} = 1.9$ ,  $INITStD = 11$  και  $ISTDREDUCTRATEFINAL = 3$ .

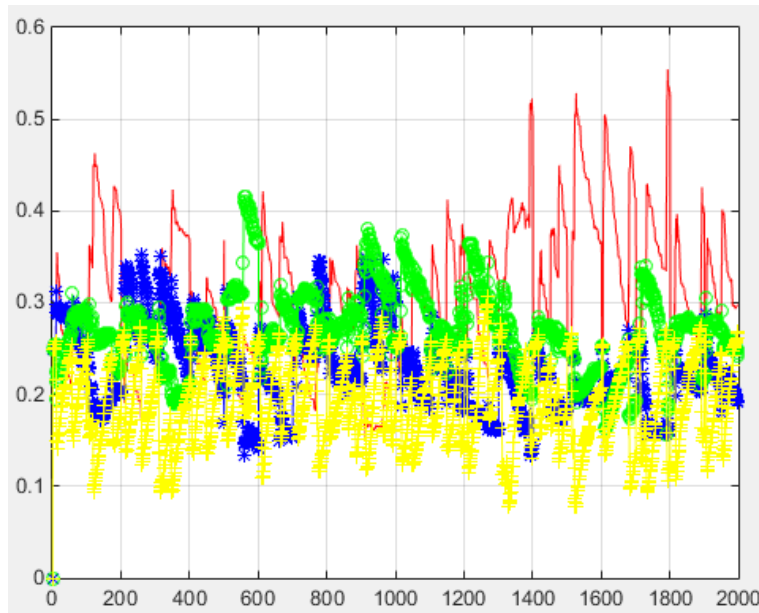
Το λάμδα από το οποίο προκύπτει σύγκλιση του προβλήματος και καλύτερη τιμή από αυτή που προτείνεται στην βιβλιογραφία βρίσκεται στο εύρος τιμών από 2.02 έως 2.07, με την βέλτιστη τιμή της αντικειμενικής να προκύπτει για τιμή  $\lambda = 2.06345812200672$ .

Ακόμη, μας ενδιαφέρει η εξέλιξη του ποσοστού επιλογής των 4 παραγόντων (a,b,c,d) της SLPSO κατά την διάρκεια της εκτέλεσης των 2000 επαναλήψεων. Αυτή φαίνεται στο σχήμα 5.19 (κόκκινο χρώμα και απλή γραμμή: παράγοντας a, μπλε χρώμα και \* σε κάθε σημείο: παράγοντας b, πράσινο χρώμα και κύκλος σε κάθε σημείο: παράγοντας c, κίτρινο χρώμα και + σε κάθε σημείο: παράγοντας d).

Στην συνέχεια εκτελούμε δοκιμές στο πρόβλημα για αριθμό κύκλων επαναλήψεων της επαναληπτικής διαδικασίας ίσο με 30, διατηρώντας σταθερές κάθε φορά τις τιμές που έχουν προκύψει μέσα από την αρχικοποίηση του προβλήματος, προκειμένου να μπορέσου-



**Σχήμα 5.18:** Αποτελέσματα συναρτήσεων 3<sup>ου</sup> προβλήματος για 1 κύκλο επαναλήψεων

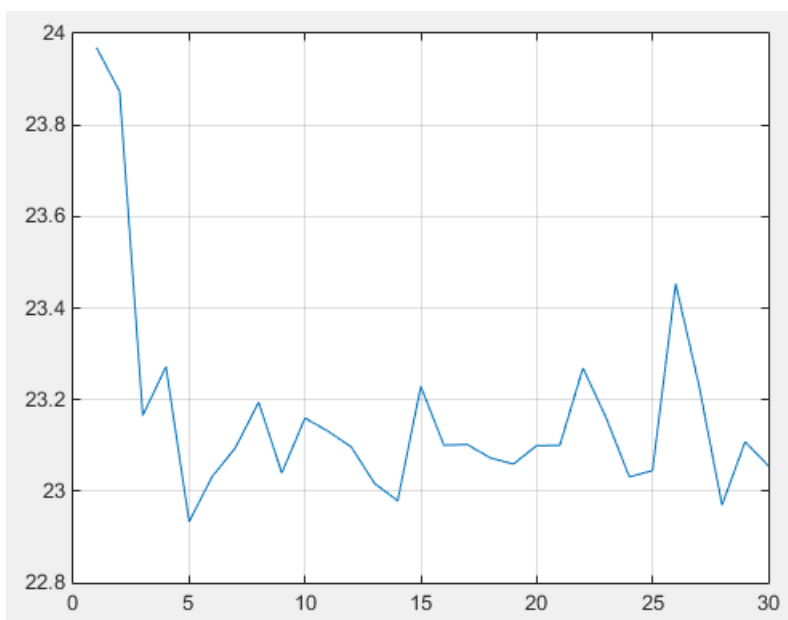


**Σχήμα 5.19:** Αποτελέσματα ρυθμού επιλογής παραγόντων SLPSO 3<sup>ου</sup> προβλήματος για 1 κύκλο επαναλήψεων

με να κατανοήσουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος. Από τις δοκιμές αυτές προκύπτουν τα ακόλουθα αποτελέσματα (πίνακας 5.7), ενώ η τιμή την οποία λαμβάνει η συνάρτηση κατά την διάρκεια της εκτέλεσης των επαναλήψεων αποτυπώνεται στο σχήμα 5.20.

	Μέση τιμή	Τυπική Απόκλιση	Ελάχιστη τιμή	Μέγιστη τιμή
Αντικειμενική Συνάρτηση	23.1681	0.2304	22.9336	23.9692
Άθροισμα Περιορισμών	0.8758	0.1771	0.2291	0.9996
Συνάρτηση Ποινής	24.3461	0.0087	24.3345	24.3649

Πίνακας 5.7: Στατιστική ανάλυση αποτελεσμάτων 3<sup>ου</sup> προβλήματος για 30 κύκλους



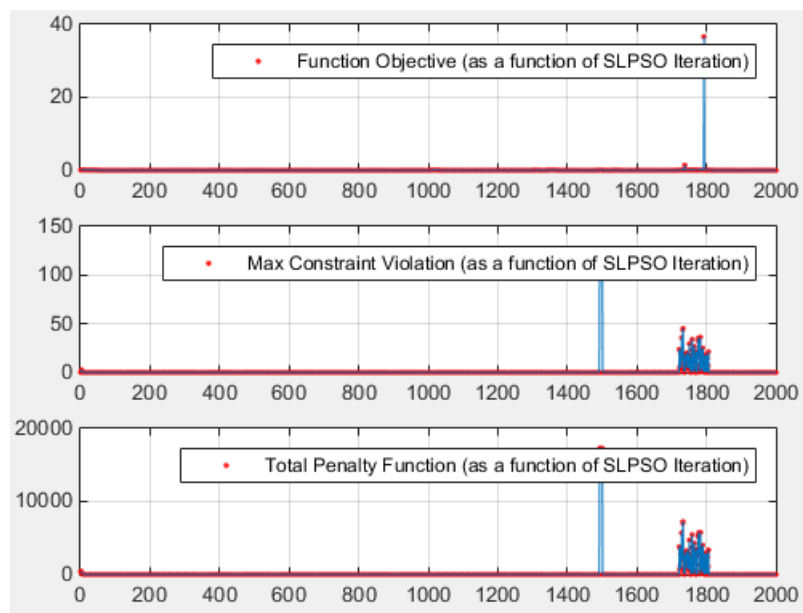
Σχήμα 5.20: Αποτέλεσμα αντικειμενικής συνάρτησης 3<sup>ου</sup> προβλήματος για 30 κύκλους

#### 5.4.3. Αποτελέσματα 4<sup>ου</sup> προβλήματος

Για την επίλυση του 4<sup>ου</sup> προβλήματος με την εφαρμογή του αλγορίθμου SLPSO και 1 κύκλο επαναλήψεων προέκυψαν βέλτιστες τιμές για τιμές  $\vec{x}^* = (1.22556608869998, 4.25605373877636)$  με  $f(\vec{x}^*) = -0.09582504$  και άθροισμα σφάλματος  $C(\vec{x}^*) = 0$ , η οποία είναι ίδια με την πρότυπη λύση που προτείνεται στην βιβλιογραφία και καλύτερη από την λύση, η οποία προέκυψε από την GCR. Η σύγκλιση της απλής αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών αλλά και της συνάρτησης ποινής, η οποία πραγματοποιείται από τον αλγόριθμο παρουσιάζεται στο σχήμα 5.21.

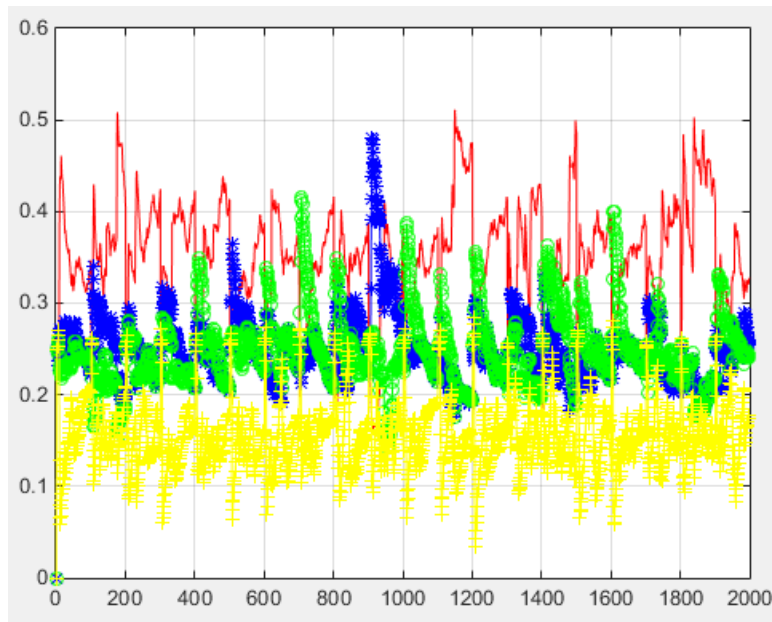
Προκειμένου να καταλήξουμε στα αποτελέσματα αυτά έχουμε προβεί σε ένα πολύ μεγάλο πλήθος δοκιμών για τους 3 παράγοντες, τους οποίους αναλύσαμε. Συγκεκριμένα, μελετήσαμε τιμές για το αρχικό λάμδα από 0.1 έως και 5, τιμές για την αρχική τυπική απόκλιση από 0.01 έως και 25,000 καθώς και τις τιμές 1, 2, 3 και 4 για τον ρυθμό μείωσης της τυπικής απόκλισης. Μέσα από τις δοκιμές αυτές καταλήξαμε ότι οι βέλτιστες τιμές για τους παράγοντες αυτούς είναι  $\lambda_{init} = 0.5$ ,  $INITStd = 0.01$  και  $ISTDREDUCTRATEFINAL = 3$ .

Το λάμδα από το οποίο προκύπτει σύγκλιση του προβλήματος και καλύτερη τιμή από αυτή που προτείνεται στην βιβλιογραφία βρίσκεται στο εύρος τιμών από 160.1-160.5, με την βέλτιστη τιμή της αντικειμενικής να προκύπτει για τιμή  $\lambda = 160.107450223430$ .



**Σχήμα 5.21:** Αποτελέσματα συναρτήσεων  $4^{ov}$  προβλήματος για 1 κύκλο επαναλήψεων

Ακόμη, μας ενδιαφέρει η εξέλιξη του ποσοστού επιλογής των 4 παραγόντων (a,b,c,d) της SLPSO κατά την διάρκεια της εκτέλεσης των 2000 επαναλήψεων. Αυτή φαίνεται στο σχήμα 5.19 (κόκκινο χρώμα και απλή γραμμή: παράγοντας a, μπλε χρώμα και \* σε κάθε σημείο: παράγοντας b, πράσινο χρώμα και κύκλος σε κάθε σημείο: παράγοντας c, κίτρινο χρώμα και + σε κάθε σημείο: παράγοντας d).

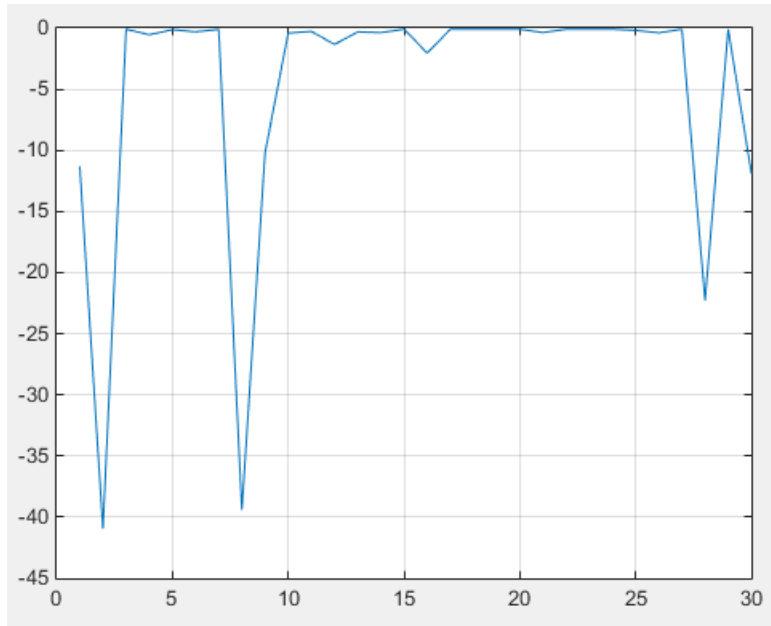


**Σχήμα 5.22:** Αποτελέσματα ρυθμού επιλογής παραγόντων SLPSO 4<sup>ου</sup> προβλήματος για 1 κύκλο επαναλήψεων

Στην συνέχεια εκτελούμε δοκιμές στο πρόβλημα για αριθμό κύκλων επαναλήψεων της επαναληπτικής διαδικασίας ίσο με 30, διατηρώντας σταθερές κάθε φορά τις τιμές που έχουν προκύψει μέσα από την αρχικοποίηση του προβλήματος, προκειμένου να μπορέσουμε να κατανοήσουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος. Από τις δοκιμές προκύπτουν τα αποτελέσματα του πίνακα 5.8. Η τιμή που λαμβάνει η συνάρτηση κατά την διάρκεια της εκτέλεσης των επαναλήψεων αποτυπώνεται στο σχήμα 5.20.

	Μέση τιμή	Τυπική Απόκλιση	Ελάχιστη τιμή	Μέγιστη τιμή
<b>Αντικειμενική Συνάρτηση</b>	-4.9770	10.8547	-40.9494	-0.0958
<b>Άθροισμα Περιορισμών</b>	0.4212	0.3770	0	0.9656
<b>Συνάρτηση Ποινής</b>	-0.0958	0	-0.0958	-0.0958

**Πίνακας 5.8:** Στατιστική ανάλυση αποτελεσμάτων 4<sup>ου</sup> προβλήματος για 30 κύκλους



**Σχήμα 5.23:** Αποτέλεσμα αντικειμενικής συνάρτησης 4<sup>ου</sup> προβλήματος για 30 κύκλους. Δυστυχώς στο πρόβλημα αυτό επειδή η τιμή είναι πολύ μικρή, είναι πολύ εύκολο ο αλγόριθμος να ξεφύγει σε περιοχή λύσεων πολύ μακριά από την σωστή. Στις περιπτώσεις, όμως, που βρίσκεται στην σωστή περιοχή, σχεδόν πάντα πετυχαίνει το αποτέλεσμα με μηδενικό σφάλμα.

#### 5.4.4. Αποτελέσματα 5<sup>ου</sup> προβλήματος

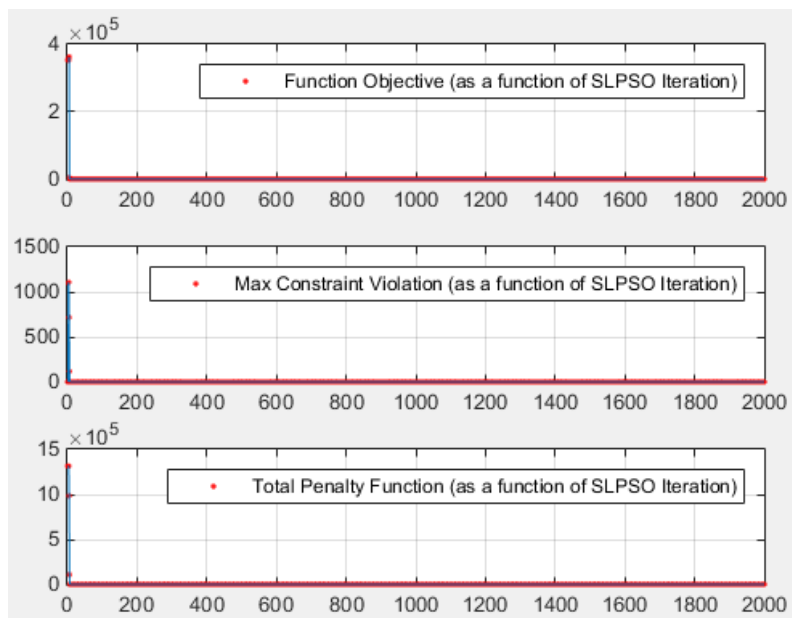
Για την επίλυση του 5<sup>ου</sup> προβλήματος με την εφαρμογή του αλγορίθμου SLPSO και 1 κύκλο επαναλήψεων προέκυψαν βέλτιστες τιμές για τιμές διανύσματος  $\vec{x}^* = (1.99475859946137, 1.93293467106819, -0.00000103509918, 4.44394012053229, -0.366183663728745, 1.13030203138010, 1.46315034121436)$  με τιμή της  $f(\vec{x}^*) = 683.6855$  και άθροισμα σφάλματος  $C(\vec{x}^*) = 0$ . Η τιμή αυτή είναι χειρότερη λύση από αυτή που προτείνεται στην βιβλιογραφία και από την λύση η οποία προέκυψε από την GCR. Η σύγκλιση της απλής αντικειμενικής συνάρτησης και του αθροίσματος των περιορισμών αλλά και της συνάρτησης ποινής, η οποία πραγματοποιείται από τον αλγόριθμο παρουσιάζεται στο σχήμα 5.24.

Προκειμένου να καταλήξουμε στα αποτελέσματα αυτά, έχουμε προβεί σε ένα πολύ μεγάλο πλήθος δοκιμών για τους 3 παράγοντες, τους οποίους αναλύσαμε. Συγκεκριμένα,



μελετήσαμε τιμές για το αρχικό λάμδα από 0.1 έως και 5, τιμές για την αρχική τυπική απόκλιση από 0.1 έως και 25,000 καθώς και τις τιμές 1, 2, 3 και 4 για τον ρυθμό μείωσης της τυπικής απόκλισης. Μέσα από τις δοκιμές αυτές καταλήξαμε ότι οι βέλτιστες τιμές για τους παράγοντες αυτούς είναι  $\lambda_{init} = 0.1$   $INITStD = 0.1$  και  $ISTDREDUCTRATEFINAL = 4$ .

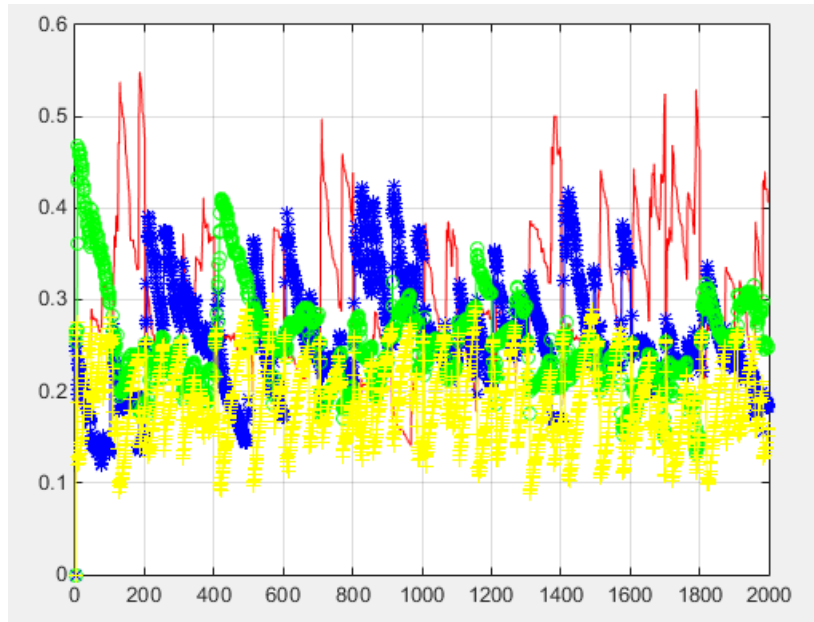
Το λάμδα από το οποίο προκύπτει σύγκλιση του προβλήματος και καλύτερη τιμή από αυτή που προτείνεται στην βιβλιογραφία βρίσκεται στο εύρος τιμών από 860-1500, με την βέλτιστη τιμή της αντικειμενικής να προκύπτει για τιμή  $\lambda = 869.904199983389$ .



**Σχήμα 5.24:** Αποτελέσματα συναρτήσεων  $5^{ov}$  προβλήματος για 1 κύκλο επαναλήψεων

Ακόμη, μας ενδιαφέρει η εξέλιξη του ποσοστού επιλογής των 4 παραγόντων (a,b,c,d) της SLPSO κατά την διάρκεια της εκτέλεσης των 2000 επαναλήψεων. Αυτή φαίνεται στο σχήμα 5.19 (κόκκινο χρώμα και απλή γραμμή: παράγοντας a, μπλε χρώμα και \* σε κάθε σημείο: παράγοντας b, πράσινο χρώμα και κύκλος σε κάθε σημείο: παράγοντας c, κίτρινο χρώμα και + σε κάθε σημείο: παράγοντας d).

Στην συνέχεια εκτελούμε δοκιμές στο πρόβλημα για αριθμό κύκλων επαναλήψεων της επαναληπτικής διαδικασίας ίσο με 30, διατηρώντας σταθερές κάθε φορά τις τιμές που έχουν προκύψει μέσα από την αρχικοποίηση του προβλήματος, προκειμένου να μπορέσου-

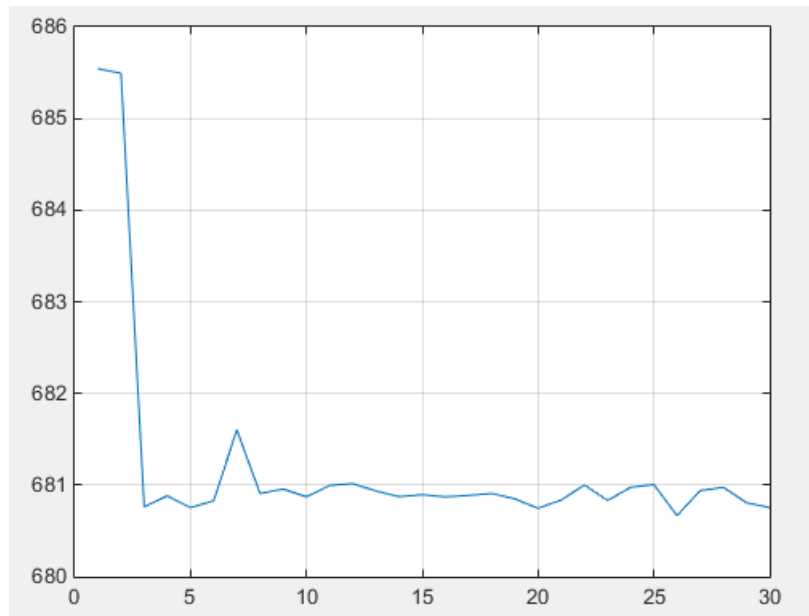


**Σχήμα 5.25:** Αποτελέσματα ρυθμού επιλογής παραγόντων SLPSO 5<sup>ου</sup> προβλήματος για 1 κύκλο επαναλήψεων

με να κατανοήσουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος. Από τις δοκιμές προκύπτουν τα αποτελέσματα του πίνακα 5.8. Η τιμή που λαμβάνει η συνάρτηση κατά την διάρκεια της εκτέλεσης των επαναλήψεων αποτυπώνεται στο σχήμα 5.26.

	Μέση τιμή	Τυπική Απόκλιση	Ελάχιστη τιμή	Μέγιστη τιμή
<b>Αντικειμενική Συνάρτηση</b>	681.2123	1.1811	680.6676	685.5421
<b>Άθροισμα Περιορισμών</b>	0	0	0	0
<b>Συνάρτηση Ποινής</b>	681.2123	1.1811	680.6676	685.5421

**Πίνακας 5.9:** Στατιστική ανάλυση αποτελεσμάτων 5<sup>ου</sup> προβλήματος για 30 κύκλους



**Σχήμα 5.26:** Αποτέλεσμα αντικειμενικής συνάρτησης 5<sup>ου</sup> προβλήματος για 30 κύκλους

#### 5.4.5. Ανάλυση αποτελεσμάτων

Από την εκτέλεση των δοκιμών που πραγματοποιήσαμε μπορούμε να εξάγουμε διάφορα αποτελέσματα για τον τρόπο με τον οποίο οι 3 παράγοντες που μελετήσαμε μπορούν να επηρεάσουν την αναζήτηση βέλτιστης λύσης σε ένα μη-γραμμικό πρόβλημα με την μέθοδο των ΕΑ και συγκεκριμένα της SLPSO.

Αρχικά, όσον αφορά τον παράγοντα του  $\lambda_{init}$ , φάνηκε ότι επηρεάζει σημαντικά την σύγκλιση του προβλήματος, καθότι, εάν η τιμή του δεν είναι η ενδεδειγμένη, οδηγεί την τιμή του  $\lambda$  σε λάθος περιοχή τιμών και άρα η τιμή του προβλήματος, η οποία προκύπτει δεν είναι η επιθυμητή. Οι τιμές, οι οποίες φαίνεται να παίρνει ο παράγοντας αυτός, βρίσκονται ανάμεσα στις τιμές 0.1-1 με μοναδική εξαίρεση το 3<sup>ο</sup> πρόβλημα, όπου χρειάστηκε τιμή 1.9. Σε κάθε περίπτωση, η εισαγωγή του  $\lambda_{init}$  στην επίλυση του προβλήματος φαίνεται αρκετά χρήσιμη, καθώς περιορίζει αρκετά το πλήθος των τιμών, το οποίο πρέπει να δοκιμάσουμε προκειμένου να πετύχουμε το καλύτερο δυνατό  $\lambda$ . Από τις δοκιμές φάνηκε ότι μια μικρή αλλαγή στην τιμή του  $\lambda$ , η οποία προκύπτει ουσιαστικά από το  $\lambda_{init}$ , μπορεί να έχει σημαντική επιρροή στο αποτέλεσμα του αλγορίθμου.

Στην συνέχεια, μελετήσαμε τον παράγοντα αρχικής τυπικής απόκλισης. Ο παράγοντας αυτός φαίνεται να επηρεάζει σημαντικά την σύγκλιση του προβλήματος στην βέλτιστη δυνατή τιμή. Παρότι η βέλτιστη τιμή του παράγοντα αυτού φαίνεται να βρίσκεται στην περιοχή του 0.01-1, δεν μπορούμε να αγνοήσουμε την πιθανότητα ο παράγοντας αυτός να πρέπει να πάρει διαφορετική τιμή, όπως στο 3<sup>ο</sup> πρόβλημα, όπου πήρε την τιμή 11. Πάντως, είναι σημαντικό να αναφέρουμε ότι, παρότι υπήρχε βέλτιστη τιμή, το αποτέλεσμα του αλγορίθμου δεν άλλαζε σημαντικά στις μη βέλτιστες περιοχές, παρά το μικρό ή το μεγάλο μέγεθος του παράγοντα κάθε φορά (π.χ. ίδια τιμή αντικειμενικής για μη βέλτιστα INITStD=5 και INITStD=50000).

Ακόμη, μελετήσαμε την τιμή του ρυθμού μείωσης της τυπικής απόκλισης. Ο αριθμός αυτός λαμβάνει την τιμή 1, 2, 3 ή 4 όπως αναλύσαμε προηγουμένως. Μέσα από τις δοκιμές, φάνηκε ότι υπάρχουν προβλήματα στα οποία η τιμή αυτή μπορεί να έχει μικρή ή και καμία διαφορά στο αποτέλεσμα (π.χ. στο 2<sup>ο</sup> πρόβλημα βγαίνει ίδιο αποτέλεσμα για τιμή 1, 2 και 3). Πάντως, φαίνεται ότι ο παράγοντας αυτός μπορεί να προσφέρει σημαντικά στην σύγκλιση του αλγορίθμου στην βέλτιστη λύση, καθώς, συνήθως, αποδίδει διαφορετικά αποτελέσματα για διαφορετικές τιμές του.

Τέλος, παρότι δεν επηρεάσαμε εμείς τον παράγοντα  $\lambda$ , εξαιτίας της μεγάλης σημασίας του, πρέπει να γίνει ιδιαίτερη αναφορά. Μέσα από τις δοκιμές φάνηκε ότι αυτός ήταν αρκετές φορές υπεύθυνος για την σύγκλιση ή μη του προβλήματος. Δεν παρατηρείται καμία αναλογία στην τιμή της λύσης της αντικειμενικής συνάρτησης και στην τιμή του  $\lambda$ . Όμως, η κύρια χρησιμότητά του βρίσκεται στο γεγονός ότι οδηγεί κατά έναν πολύ μεγάλο βαθμό το πρόβλημα στην βέλτιστη λύση με ελαχιστοποίηση της παραβίασης των περιορισμών.

## **ΚΕΦΑΛΑΙΟ 6**

### **ΣΥΜΠΕΡΑΣΜΑΤΑ**



## **6.1. Συμπεράσματα - Επίλογος διπλωματικής εργασίας**

Παρότι οι εξελικτικοί αλγόριθμοι προτάθηκαν για πρώτη φορά περισσότερα από 50 χρόνια πριν, σήμερα αποτελούν ένα από τα σημαντικότερα εργαλεία στον τομέα της βελτιστοποίησης. Η δυνατότητα βελτιστοποίησης μεγάλων γραμμικών και μη-γραμμικών προβλημάτων, τα οποία μπορεί να υπόκεινται σε μεγάλο πλήθος περιορισμών, αποτελεί το ισχυρότερο πλεονέκτημα σε σχέση με τις κλασσικές μεθόδους πρόβλεψης, ενώ η στοχαστικότητα που εισάγεται στην εύρεση της επίλυσης μέσω εξελικτικών αλγορίθμων είναι ένας από τους σημαντικότερους παράγοντες, οι οποίοι συμβάλλουν στην σύγκλιση των προβλημάτων στην βέλτιστη λύση.

Στην συγκεκριμένη διπλωματική εργασία, μέσα από την μοντελοποίηση μεθόδων και προβλημάτων στο λογισμικό MATLAB®, υπήρχαν 2 διαφορετικοί στόχοι. Αρχικά, ελέγχθηκε η απόδοση, την οποία μπορεί να έχει η επίλυση ενός πολύ μεγάλου πραγματικού προβλήματος, όπως η πρόβλεψη της ενεργειακής ζήτησης ενός κράτους για τα επόμενα 38 χρόνια, με την χρήση μιας εκ των πιο ευρέως χρησιμοποιούμενων μεθόδων. Το δεύτερο και κύριο τμήμα της διπλωματικής εργασίας αφορούσε την ίδια την φύση των δύο μεθόδων που μελετήθηκαν, της μεθόδου Global Competitive Ranking (Runarsson & Yao, 2003) και της μεθόδου Self-Learning Particle Swarm Optimizer (Changhe & Shengxiang, 2012), καθώς ελέγχθηκε κατά πόσο επηρεάζεται η απόδοσή τους μέσα από την τιμή την οποία μπορεί να δώσουμε σε διάφορους παράγοντες (διαφορετικοί σε κάθε μέθοδο). Για την διερεύνηση των παραγόντων, οι οποίοι εκτιμήθηκαν ότι μπορούν να επηρεάσουν σημαντικά τα αποτελέσματα του κάθε αλγορίθμου, πραγματοποιήθηκαν μια σειρά από δοκιμές.

Συγκεκριμένα, για τον αλγόριθμο GCR μελετήθηκε ο αριθμός των επαναλήψεων, ο αριθμός του σμήνους των σωματιδίων, η πιθανότητα η σύγκριση να γίνει μόνο με την αντικειμενική συνάρτηση αλλά και ο πολλαπλασιαστικός βάσης, η ανάγκη δημιουργίας του οποίου προέκυψε μέσα από την προσπάθεια σύγκλισης του μεγάλου προβλήματος στην βέλτιστη λύση, υπό την καθοδήγηση του επιβλέποντα καθηγητή. Μέσα από την μοντελοποίηση του προβλήματος προέκυψαν βέλτιστες τιμές για κάθε παράγοντα, οι οποίες και ελέγχθηκαν με την εφαρμογή τους σε 4 μικρά μη-γραμμικά προβλήματα, τα

οποία χρησιμοποιούνται πολύ συχνά στην βιβλιογραφία. Από τις δοκιμές αυτές προέκυψε ότι, παρότι ο αριθμός των επαναλήψεων και ο αριθμός των σωματιδίων βελτιώνουν το πρόβλημα όσο αυξάνονται, ο ρυθμός βελτίωσης της τιμής της αντικειμενικής συνάρτησης, μετά από μια τιμή των παραγόντων αυτών, αρχίζει να μειώνεται σημαντικά. Έτσι, σε συνδυασμό με το μεγάλο υπολογιστικό χρόνο που απαιτείται για μεγάλο αριθμό επαναλήψεων ή/και σωματιδίων, θεωρείται ασύμφορη η ανεξέλεγκτη αύξηση των παραγόντων μετά από ένα σημείο. Αντίθετα, η τιμή του πολλαπλασιαστή βάσης καθώς και της πιθανότητας η σύγκριση να γίνει μόνο με την αντικειμενική συνάρτηση μπορούν να βελτιώσουν το πρόβλημα χωρίς καμία χρονική επιβάρυνση και για τον λόγο αυτό θεωρείται πολύ μεγάλης σημασίας η εύρεση των βέλτιστων τιμών τους.

Από την άλλη, για τον αλγόριθμο SLPSO μελετήθηκε ο παράγοντας αρχικού  $\lambda$ , ο παράγοντας αρχικής τυπικής απόκλισης αλλά και ο παράγοντας ρυθμού μείωσης της τυπικής απόκλισης. Μέσα από την μοντελοποίηση της μεθόδου και την εφαρμογή του σε 4 μικρά μη-γραμμικά προβλήματα, τα οποία χρησιμοποιούνται πολύ συχνά στην βιβλιογραφία, προέκυψαν οι βέλτιστες τιμές για τους παράγοντες αυτούς (ξεχωριστοί για κάθε πρόβλημα) και πραγματοποιήθηκε έλεγχος των τιμών της αντικειμενικής συνάρτησης με την βιβλιογραφία αλλά και τον αλγόριθμο GCR. Από τις δοκιμές αυτές προέκυψε ότι η εισαγωγή του  $\lambda_{init}$  στην επίλυση του προβλήματος περιορίζει αρκετά το πλήθος των τιμών, το οποίο πρέπει να δοκιμάσουμε προκειμένου να πετύχουμε το καλύτερο δυνατό  $\lambda$  και άρα η ύπαρξή του είναι αρκετά χρήσιμη. Ακόμη, από τις δοκιμές προέκυψε ότι ο παράγοντας αρχικής τυπικής απόκλισης είναι σημαντικός για την εύρεση της βέλτιστης τιμής του προβλήματος, καθώς υπάρχει μόνο μια πολύ μικρή περιοχή όπου η λύση είναι βέλτιστη, ενώ σε όλες τις υπόλοιπες περιοχές η λύση, η οποία προκύπτει είναι περίπου ίδια. Τέλος, από τις δοκιμές προέκυψε ότι και η τιμή του ρυθμού μείωσης μπορεί να επηρεάσει την τιμή της αντικειμενικής συνάρτησης, αν και εξαρτάται από την τιμή, την οποία λαμβάνει η αρχική τυπική απόκλιση.

Εν κατακλείδι, είναι σημαντικό να σημειώσουμε ότι, όσον αφορά τον αλγόριθμο GCR, οι τιμές, οι οποίες προέκυψαν μέσα από τις δοκιμές στο μεγάλο μη-γραμμικό πρόβλημα, οδηγούν στην βέλτιστη λύση τα υπόλοιπα προβλήματα, τα οποία μελετήθηκαν, παρόλο που δεν γίνεται δυνατό να προσεγγιστεί η βέλτιστη λύση της βιβλιογραφίας για το 1<sup>ο</sup>



πρόβλημα. Αντίθετα, όσον αφορά τον αλγόριθμο SLPSO, κάθε πρόβλημα παίρνει δικές του, ξεχωριστές τιμές. Πάντως, πρέπει να αναφερθεί το γεγονός ότι, παρόλο που κάθε πρόβλημα παίρνει ξεχωριστές τιμές, υπάρχει μια συγκεκριμένη περιοχή αναζήτησης τιμών για κάθε παράγοντα του SLPSO, μέσα στην οποία θα βρεθεί, κατά μεγάλη πιθανότητα, η βέλτιστη τιμή του.

Μέσα από την εκπόνηση της διπλωματικής εργασίας είναι εμφανές ότι κάθε μέθοδος, η οποία βασίζεται στους εξελικτικούς αλγορίθμους, απαιτεί ξεχωριστή μελέτη των πιθανών παραγόντων που μπορεί να την οδηγήσουν στην εύρεση της βέλτιστης τιμής της αντικειμενικής συνάρτησης ενός μικρού ή μεγάλου προβλήματος μη-γραμμικού προγραμματισμού το οποίο υπόκειται περιορισμούς. Παράλληλα, δεν προκύπτει κάποιος κανόνας για το εάν οι παράγοντες, οι οποίοι επηρεάζουν κάποια μέθοδο και προέκυψαν μέσα από δοκιμές για ένα συγκεκριμένου πρόβλημα, μπορούν να εφαρμοστούν και σε άλλα προβλήματα της ίδιας μεθόδου με παρόμοια αποτελέσματα. Τέλος, αξίζει να σημειωθεί ότι, παρότι προέκυψε αδυναμία σύγκλισης της τιμής του μεγάλου προβλήματος, το οποίο μελετήθηκε, με την τιμή η οποία αναφέρεται ως η βέλτιστη στην βιβλιογραφία, μπορέσαμε να εξάγουμε πολύ σημαντικά συμπεράσματα όσον αφορά τους πιθανούς παράγοντες, οι οποίοι μπορεί να επηρεάσουν τα αποτελέσματα των διάφορων μεθόδων εξελικτικών αλγορίθμων.



## Βιβλιογραφία

- Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266 (5187), 1021–1024.
- Azad, A. K., & Fernandes, M. (2011). Modified Differential Evolution Based on Global Competitive Ranking for Engineering Design Optimization Problems. *Computational Science and Its Applications - ICCSA 2011: International Conference Proceedings, Part III*. Santander, Spain: Springer.
- Back et al., T. (1991). A survey of evolution strategies. *Proceedings of the 4th International Conference on Genetic Algorithms* (σσ. 2-9). Morgan Kaufmann.
- Back et al., T. (1997). Evolutionary Computation: Comments on the History and Current State. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 1, no. 1.
- Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. New York: Oxford.
- Bean, J. C., & Hadj-Alouane, A. B. (1992). A dual genetic algorithm for bounded integer programs. *Technical Report TR 92-53*.
- Box, G. E. (1957). Evolutionary operation: A method for increasing industrial productivity. *Appl. Statistics*, vol. VI, no. 2, 81-101.
- Bremermann, H. J. (1962). Optimization through evolution and recombination. *Self-Organizing Systems*.
- Changhe, L., & Shengxiang, Y. (2012). A Self-Learning Particle Swarm Optimizer for Global Optimization Problems. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, vol. 42, no. 3, 627-646.
- Dorigo et al., M. (1991). Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle. Paris, France: Elsevier Publishing.
- Eiben, A., & Smith, J. (2003). What is an Evolutionary Algorithm? Στο A. Eiben, & J. Smith, *Introduction to Evolutionary Computing* (σσ. 15-34). Springer, Natural Computing Series.

- Fogel, L. (1962). Autonomous automata. *Ind. Res.*, vol. 4, 14-19.
- Forrest et al., S. (1994). Self-nonsel self discrimination in a computer. *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*. Los Alamitos, CA., 202–212.
- Friedberg et al., R. (1959). A learning machine: Part 2. *IBM J.* vol. 3, no. 7, 282-287.
- Friedberg, R. M. (1958). A learning machine: Part I. *IBM J.* vol. 2, no. 1, 2-13.
- Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences* 8 (1).
- Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, vol. 13, 533-549.
- Glover, F., & Kochenberger, G. (1995). Critical event tabu search for multidimensional knapsack problems. *Proceedings of the International Conference on Metaheuristics for Optimization* (σσ. 113-133). Kluwer Publishing.
- Holland, J. (1962). Outline for a logical theory of adaptive systems. *J. Assoc. Comput. Mach.*, vol. 3, 297-314.
- Homaifar et al., A. (1994). Constrained optimization via genetic algorithms. *Simulation* 62 (4), 242-254.
- Hooke, R., & Jeeves, T. (1961). "Direct search" solution of numerical and statistical problems. *Journal of the Association for Computing Machinery (ACM)* 8 (2), 212-229.
- Joines, J., & Houck, C. (1994). On the use of non-stationary penalty functions to solve non-linear constrained optimization problems with gas. *Proceedings of the First IEEE International Conference on Evolutionary Computation* (σσ. 579-584). IEEE Press.
- Kelly, J., & Davis, L. (1991). Hybridizing the Genetic Algorithm and the K-Nearest Neighbors Classification Algorithm.
- Kennedy, J. (1997). The particle swarm: social adaptation of knowledge. *Proceedings of IEEE International Conference on Evolutionary Computation*, 303-308.

- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks IV.*, 1942–1948.
- Kennedy, J., & Eberhart, R. C. (2001). *Swarm Intelligence*. Morgan Kaufmann.
- Koza. (1990). Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. .
- Larrañaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Boston: Kluwer Academic Publishers.
- Leriché et al., R. G. (1995). A segregated genetic algorithm for constrained structural optimization. *Proceedings of the 6th International Conference on Genetic Algorithms*, (σσ. 558-565).
- Li, C., & S. Yang. (2009). “An adaptive learning particle swarm optimizer for function optimization. *Conger. Evol. Comput.*, 381-388.
- Liang et al., J. (2006). Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. *Nanyang Technological University, Singapore, Tech. Rep 01/2006*.
- Liang et al., J. (2006). Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. *Nanyang Technological University, Singapore, Tech. Rep.*
- Michalewicz, Z. (1993). A hierarchy of evolution programs: An experimental study. *Evolutionary Computation*.
- Michalewicz, Z. (1996). *Genetic Algorithms+Data Structures=Evolution Programs*. New-York: Springer Verlag.
- Michalewicz, Z., & Attia, N. (1994). Evolutionary optimization of constrained problems. *Proceedings of the 3rd Annual Conference on Evolutionary Programming* (σσ. 98-108). World Scientific.

- Michalewicz, Z., & Janikow, C. Z. (1991). Handling constraints in genetic algorithms. *Proceedings of the 4th International Conference on Genetic Algorithms* (σσ. 151-157). Morgan Kaufmann.
- Michalewicz, Z., & Nazhiyath, G. (1995). Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. *Proceedings of the Second IEEE International Conference on Evolutionary Computation* (σσ. 647-651). IEEE Press.
- Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation, vol. 4, issue 1*.
- Moscato, P. (1989). On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. *Caltech Concurrent Computation Program (report 826)*.
- Myung et al., H. (1995). Preliminary investigation into a two-stage method of evolutionary optimization on constrained problems. *Proceedings of the 4th Annual Conference on Evolutionary Programming* (σσ. 449-463). MIT Press.
- Papadrakakis et al. (1999). Innovative Computational Methods in Structural Optimization. *Innovative Computational Methods for Structural Mechanics*. Ανάκτηση 1999
- Paredis, J. (1994). Coevolutionary constraint satisfaction. *Proceedings of the 3rd Conference on Parallel Problems Solving from Nature* (σσ. 46-55). Springer Verlag.
- Parmee, I., & Purchase, G. (1994). The development of directed genetic search technique for heavily constrained design spaces. *Proceedings of the Conference on Adaptive Computing in Engineering Design and Control* (σσ. 97-102). University of Plymouth.
- Poli et al., R. (2007). Particle swarm optimization: An Overview. *Swarm Intelligence, vol. 1, no. 1*, 33-58.

- Powell, D., & Skolnick, M. M. (1993). Using genetic algorithms in engineering design optimization with non-linear constraints. *Proceedings of the 5th International Conference on Genetic Algorithms* (σσ. 424-430). Morgan Kaufmann.
- Rechenberg, I. (1965). Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment, Library translation No.. 1122*.
- Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Germany: Frommann-Holzboog.
- Reynolds et al., R. G. (1994). Agent-Based Modeling of Cultural Change.
- Runarsson, T. P., & Yao, X. (2000). Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 4, no. 3, 284-294.
- Runarsson, T. P., & Yao, X. (2005). Search Biases in Constrained Evolutionary Optimization. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS*, vol. 35, no. 2, 233-243.
- Runarsson, T., & Yao, X. (2003). Constrained evolutionary optimization – the penalty function approach. *Evolutionary Optimization: International Series in Operations Research and Management Science* vol. 48, 87-113.
- Schaffer, D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the 1st International Conference on Genetic Algorithms*. Laurence Erlbaum Associates.
- Schoenauer, M., & Xanthakis, S. (1993). Constrained GA optimization. *Proceedings of the 5th International Conference on Genetic Algorithms* (σσ. 573-580). Morgan Kaufmann.
- Schwefel, H.-P. (1968). Projekt MHD-Straustrahlrohr: Experimentelle Optimierung einer Zweiphasenduse, Teil I. *Technischer Bericht 11.034/68,35*.
- Schwefel, H.-P. (1975). *Evolutionsstrategie und numerischer Optimierung* Dissertation. .

- Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. New York.
- Smith, A., & Tate, D. (1993). Genetic optimization using a penalty function. *Proceedings of the 5th International Conference on Genetic Algorithms* (σσ. 499-503). Morgan Kaufmann.
- Tolis, A. (2015). The effect of long-term expansion on the evolution of electricity price: numerical analysis of a theoretically optimised electricity market. *European Journal of Operational Research*.
- Waagen et al., D. (1992). The stochastic direction set algorithm: A hybrid technique for finding function extrema. *Proceedings of the 1st Annual Conference on Evolutionary Programming* (σσ. 35-42). Evolutionary Programming Society.
- Zhan et al., Z. (2009). Adaptive particle swarm optimization. *IEEE Trans. Syst., Man, Cybern. B: Cybern*, vol. 39, 1362-1381.