



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΤΟΜΕΑΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ &  
ΕΠΙΧΕΙΡΗΣΙΑΚΗΣ ΕΡΕΥΝΑΣ

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Πειραματική μελέτη των παραμέτρων των εξελικτικών  
αλγορίθμων SL-PSO και εDEag πάνω στην επίλυση  
μη-γραμμικών προβλημάτων**

**Βαρδάκης Θ. Νικόλαος**

**Φοιτητής 14<sup>ου</sup> εξαμήνου**

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

Αθήνα, Ιούνιος 2016



Αφιερώνεται  
στην οικογένειά μου



## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κύριο Τόλη Αθανάσιο που με εμπιστεύτηκε για τη διεκπεραίωση της παρούσας διπλωματικής εργασίας, καθώς και για την πολύτιμη βοήθειά του καθ' όλη τη διάρκεια της ενασχόλησής μου με αυτή. Επίσης, θα ήθελα να ευχαριστήσω το συμφοιτητή και φίλο Λιάκο Νικόλαο που με βοήθησε στην έναρξη της εργασίας αυτής, ιδιαίτερα στη χρήση του υπολογιστικού προγράμματος της MATLAB®.

Ακόμα θα ήθελα να ευχαριστήσω όλους τους καθηγητές, τους συμφοιτητές και τους φίλους μου για όλες τις αναμνήσεις που έχω από τα φοιτητικά χρόνια που πέρασα στη σχολή Μηχανολόγων Μηχανικών.

Τέλος, θα ήθελα να ευχαριστήσω πάρα πολύ την οικογένειά μου που μου συμπαραστάθηκε σε όλες μου τις προσπάθειες, υλικά αλλά κυρίως ηθικά, καθ' όλη τη διάρκεια των σπουδών μου.



# Περιεχόμενα

Έποψη Διπλωματικής Εργασίας .....	11
-----------------------------------	----

## **ΚΕΦΑΛΑΙΟ 1**

### **ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΕΞΕΛΙΚΤΙΚΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ**

1.1	Οι βάσεις των Εξελικτικών Αλγορίθμων .....	15
1.2	Ιστορική ανασκόπηση των Εξελικτικών Αλγορίθμων .....	16
1.3	Πρακτικές Εφαρμογές .....	17

## **ΚΕΦΑΛΑΙΟ 2**

### **ΟΙ ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ**

2.1	Περιγραφή ενός Εξελικτικού Αλγορίθμου .....	21
2.2	Βασικά στοιχεία των εξελικτικών αλγορίθμων .....	22
2.3	Τρόποι διαχείρισης περιορισμών στους εξελικτικούς αλγορίθμους .....	27
2.3.1	Βασικός ορισμός ενός προβλήματος .....	27
2.3.2	Μέθοδοι διαχείρισης περιορισμών .....	28
2.3.2.1	Χρήση συναρτήσεων ποινής .....	28
2.3.2.2	Ειδικές εκπροσωπήσεις και ειδικοί παράγοντες .....	33
2.3.2.3	Διαχωρισμός περιορισμών και αντικειμενικής συνάρτησης .....	36
2.3.2.4	Άλλες τεχνικές .....	39

## **ΚΕΦΑΛΑΙΟ 3**

### **ΜΑΘΗΜΑΤΙΚΑ ΜΟΝΤΕΛΑ SL-PSO ΚΑΙ εDEag**

3.1	Μέθοδος SL-PSO .....	43
3.1.1	Μαθηματικά στοιχεία για την SL-PSO .....	43
3.1.2	Μαθηματική μοντελοποίηση της SL-PSO .....	44
3.1.3	Περιορισμοί της αντικειμενικής .....	55
3.1.4	Νέα στοιχεία στον κώδικα SL-PSO που εξετάσαμε .....	55
3.1.5	Μοντελοποίηση του αλγορίθμου σε λογισμικό MATLAB® .....	56
3.2	Μέθοδος εDEag .....	61

3.2.1	Διαδικασία Ανάπτυξης της εDEag .....	61
3.2.2	Μαθηματικά στοιχεία για την εDEag.....	62
3.2.3	Μαθηματική μοντελοποίηση της εDEag.....	62
3.2.4	Αλγόριθμος της εDEag.....	67

## **ΚΕΦΑΛΑΙΟ 4**

### **ΤΑ ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΕΞΕΤΑΣΤΗΚΑΝ**

4.1	Προβλήματα που εξετάστηκαν με την SL-PSO.....	75
4.2	Οικονομική κάλυψη ηλεκτρικού φορτίου συστήματος .....	75
4.3	Πρόβλημα Διαχείρισης Παραγωγής Ενέργειας (ΠΔΠΕ) .....	76

## **ΚΕΦΑΛΑΙΟ 5**

### **ΠΕΙΡΑΜΑΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ**

5.1	Παράμετροι για την SL-PSO στην MATLAB® .....	81
5.2	Πειραματική διαδικασία για την SL-PSO.....	82
5.3	Αποτελέσματα προβλημάτων CEC-2010 .....	85
5.3.1	Πρόβλημα 1 .....	85
5.3.2	Πρόβλημα 2 .....	87
5.3.3	Πρόβλημα 3 .....	89
5.3.4	Πρόβλημα 4 .....	91
5.3.5	Πρόβλημα 5 .....	93
5.3.6	Πρόβλημα 6 .....	95
5.3.7	Πρόβλημα 7 .....	97
5.3.8	Πρόβλημα 8 .....	98
5.3.9	Πρόβλημα 9 .....	101
5.3.10	Συμπεράσματα από την επίλυση προβλημάτων με την SL-PSO .....	103
5.4	Παράμετροι που ελέγχθηκαν στον κώδικα εDEag, σε γλώσσα C .....	104
5.5	Εισαγωγή του «ΠΔΠΕ» στον κώδικα εDEag .....	106
5.6	Πειραματική Διαδικασία και Αποτελέσματα («ΠΔΠΕ») .....	107
5.6.1	Σύγκριση με τη μέθοδο SL-PSO σε MATLAB® («ΠΔΠΕ») .....	109



**ΚΕΦΑΛΑΙΟ 6**  
**ΣΥΜΠΕΡΑΣΜΑΤΑ**

6.1	Συμπεράσματα – Επίλογος Διπλωματικής Εργασίας.....	115
	<b>Βιβλιογραφία .....</b>	<b>117</b>
	Παράρτημα Α.....	125
	Παράρτημα Β.....	130
	Παράρτημα Γ .....	131



## Έποψη Διπλωματικής Εργασίας

Σε αυτή τη διπλωματική εργασία εξετάσαμε το πώς επηρεάζουν οι διάφορες τιμές των παραμέτρων τη λειτουργία ενός αλγορίθμου SL-PSO (Self Learning Particle Swarm Optimizer), που έχει μοντελοποιηθεί στην MATLAB®, για 9 ξεχωριστά μη-γραμμικά προβλήματα. Ακόμα εξετάσαμε τη συμπεριφορά του αλγορίθμου eDEag (ε Constrained Differential Evolution with an Archive and Gradient-Based Mutation), που έχει μοντελοποιηθεί σε γλώσσα C, σε ένα μεγάλο μη-γραμμικό ενεργειακό πρόβλημα και συγκρίναμε τα αποτελέσματα με αυτά του αλγορίθμου SL-PSO στο ίδιο. Στόχος είναι να αναλύσουμε τις βέλτιστες τιμές που βρέθηκαν για κάθε πρόβλημα, σε σχέση με άλλες μεθόδους της βιβλιογραφίας, και να εντοπίσουμε τυχόν μοτίβα στις τιμές των παραμέτρων.

Στο πλαίσιο αυτό, πραγματοποιήθηκαν υπολογιστικά πειράματα με τη χρήση ηλεκτρονικού υπολογιστή για να εντοπιστούν οι καλύτερες δυνατές τιμές των παραμέτρων για τις οποίες βρίσκουμε τη βέλτιστη τιμή του κάθε προβλήματος. Για τα πειράματα αυτά εξετάσαμε μία πληθώρα συνδυασμών όλων των δυνατών παραμέτρων με στόχο να εξασφαλισθεί, όσο είναι δυνατό, η εύρεση της καλύτερης δυνατής λύσης. Αντίστοιχα, σε περίπτωση μη εύρεσης δυνατών λύσεων να μπορούμε να αποκλείσουμε όσο το δυνατόν πιο πολλούς συνδυασμούς τιμών παραμέτρων.

Με την καθοδήγηση του επιβλέποντα καθηγητή, εξετάσαμε:

- Την επιρροή που έχουν 3 βασικές παράμετροι στη συμπεριφορά του αλγορίθμου SL-PSO για 9 διαφορετικά μη-γραμμικά προβλήματα.
- Την επιρροή που έχουν 9 βασικές και δευτερεύουσες παράμετροι στη συμπεριφορά του αλγορίθμου eDEag σε ένα μεγάλο ενεργειακό μη-γραμμικό πρόβλημα.
- Πραγματοποιήθηκαν πειράματα για την εύρεση των καλύτερων δυνατών συνδυασμών των τιμών των παραμέτρων του αλγορίθμου SL-PSO, έγινε στατιστική τους ανάλυση, και αναφέρθηκαν τυχόν μοτίβα μεταξύ τους και πιθανά όρια στις τιμές τους. Ακόμα έγινε σύγκριση με βέλτιστα από άλλους αλγορίθμους και αναλύθηκε η συμπεριφορά του αλγορίθμου SL-PSO για κάθε πρόβλημα.
- Έγιναν πειράματα για τον εντοπισμό των καλύτερων τιμών των παραμέτρων του αλγορίθμου eDEag. Επίσης, εντοπίστηκαν τυχόν όρια στις τιμές τους, και

έγινε διαχωρισμός παραμέτρων σε αυτούς που επηρεάζουν έντονα τη συμπεριφορά του αλγορίθμου, και σε αυτούς που επηρεάζουν λιγότερο. Ακόμα έγινε σύγκριση των αποτελεσμάτων και της συμπεριφοράς των καλύτερων λύσεων με αυτών του αλγορίθμου SL-PSO, για το ενεργειακό πρόβλημα.

Αναλυτικά, στο Κεφάλαιο 1 γίνεται εισαγωγή στους εξελικτικούς αλγόριθμους και αναφέρονται τα σημαντικότερα ιστορικά τους σημεία. Στο Κεφάλαιο 2 αναλύονται τα βασικά στοιχεία από τα οποία αποτελείται ένας εξελικτικός αλγόριθμος και αναφέρονται διάφορες τεχνικές χειρισμού περιορισμών. Στο Κεφάλαιο 3 παρουσιάζονται αναλυτικά οι αλγόριθμοι που χρησιμοποιήθηκαν, καθώς και ο τρόπος που είναι μοντελοποιημένοι στον ηλεκτρονικό υπολογιστή. Στο Κεφάλαιο 4 περιγράφονται τα προβλήματα προς βελτιστοποίηση. Στο Κεφάλαιο 5 περιγράφεται αναλυτικά η πειραματική διαδικασία και τα αποτελέσματά της. Στο Κεφάλαιο 6 αναφέρονται τα γενικά συμπεράσματα της παρούσας εργασίας.

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΕΞΕΛΙΚΤΙΚΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ



## **1.1 Οι βάσεις των Εξελικτικών Αλγορίθμων**

Οι Εξελικτικοί Αλγόριθμοι έχουν τις βάσεις τους στις αρχές της βιολογίας και της εξέλιξης. Στη φύση με το πέρασμα των γενεών επέρχεται η εξέλιξη στους διαφόρους οργανισμούς, αλλάζοντας τα χαρακτηριστικά και τις ιδιότητές τους. Κατά τη διαδικασία αυτή οι ισχυρότεροι οργανισμοί έχουν μεγαλύτερη πιθανότητα να επιβιώσουν και να αναπαραχθούν, μεταβιβάζοντας έτσι τα ισχυρά (βελτιωμένα) χαρακτηριστικά τους στις επόμενες γενεές. Αυτή τη διαδικασία προσπαθούν να προσημειώσουν και οι εξελικτικοί αλγόριθμοι.

Η βασική ιδέα πίσω από τους εξελικτικούς αλγόριθμους, είναι η δημιουργία περιβάλλοντος πίεσης σε ένα πλήθος ατόμων, έτσι ώστε να επικρατήσει η φυσική επιλογή ανάμεσα σε αυτά, να επέλθει η διαδικασία της εξέλιξής τους, και στο τέλος η επιβίωση των ισχυρότερων (βέλτιστων), συγκρίνοντάς τα με μία τιμή της φυσικής τους κατάστασης, με ένα τρόπο μέτρησής της που έχουμε ορίσει ως πρόβλημα. Έτσι, με το πέρασμα πολλών γενεών η τιμή της φυσικής κατάστασης αυτού του πληθυσμού βελτιώνεται συνεχώς.

Στην πράξη, στους εξελικτικούς αλγορίθμους, έχουμε ένα πρόβλημα το οποίο το ορίζουμε με ένα τρόπο μέτρησης της φυσικής του κατάστασης, δηλαδή με την αντικειμενική συνάρτηση, τους περιορισμούς και το πεδίο ορισμού. Στη συνέχεια σε αυτό το πρόβλημα ορίζουμε έναν πληθυσμό, δηλαδή ένα πλήθος πιθανών λύσεων (ατόμων), στις οποίες ασκείται η περιβαλλοντική πίεση και ανταλλάσσουν πληροφορίες μεταξύ τους, με στόχο σε αυτές να επέλθει η διαδικασία της εξέλιξης, καθώς και η φυσική επιλογή, δηλαδή η επιβίωση του ισχυρότερου. Έτσι, με μικρές βελτιώσεις και προσαρμογές στις ιδιότητες του προβλήματος, από γενιά σε γενιά, πλησιάζουμε μία βέλτιστη φυσική κατάσταση (τιμή αντικειμενικής) για αυτό το πρόβλημα. Συνεπώς, οι εξελικτικοί αλγόριθμοι είναι μία στοχαστική μέθοδος, η οποία απαιτεί της επίλυση πολλών πράξεων σε κάθε γενιά και για κάθε άτομο του πληθυσμού.

Τα δύο πολύ βασικά θετικά στοιχεία των εξελικτικών αλγορίθμων είναι: η δυνατότητα να προσαρμόζονται στις ανάγκες του κάθε προβλήματος που θέλουμε να επιλύσουμε, και η δυνατότητα της επίλυσης αρκετά δύσκολων και πολύπλοκων προβλημάτων. Αυτό τους καθιστά απαραίτητα εργαλεία σε πολλές σύγχρονες τεχνολογικές επιστήμες, όπως στη μηχανική, τα οικονομικά, τη ρομποτική κ.α. Όμως

λόγω του ότι αποτελούν στοχαστικές μεθόδους, απαιτείται επαναληπτική διαδικασία, συνεπώς είναι απαραίτητη η χρήση σύγχρονων υπολογιστικών προγραμμάτων, καθώς και εξοπλισμού, για την εφαρμογή τους.

## **1.2 Ιστορική ανασκόπηση των Εξελικτικών Αλγορίθμων**

Γενικά, όπως διατυπώθηκε και παραπάνω, για την εφαρμογή των εξελικτικών αλγορίθμων σε μαθηματικά προβλήματα είναι απαραίτητη η χρήση προηγμένων υπολογιστικών συστημάτων, κυρίως λόγω της επαναληπτικής διαδικασίας και της απαίτησης πληθυσμού πιθανών λύσεων, κάτι που πρακτικά σημαίνει ότι είναι μία εφαρμογή αρκετά σύγχρονη. Παρόλα αυτά οι θεωρητικές βάσεις της προέρχονται από τα μισά του 20<sup>ου</sup> αιώνα, κυρίως από τους Bremermann (Bremermann, 1962), Friedberg (Friedberg, 1958) και Box (Box, 1957). Στα επόμενα χρόνια αναπτύχθηκαν νέες μέθοδοι, με τις κύριες τρεις να ανήκουν στους Holland, Fogel και Rechenberg-Schwefel.

Το 1962 ο Holland ανέφερε την έννοια του Γενετικού Αλγόριθμου (Holland, 1962), μέθοδο στην οποία βασίζονται πολλοί σύγχρονοι αλγόριθμοι βελτιστοποίησης. Την ίδια χρονιά ο Fogel διατύπωσε την έννοια του Εξελικτικού Προγραμματισμού (Fogel, 1962), με στόχο να βελτιώσει τον τρόπο «σκέψης» των μηχανών για την πρόβλεψη μελλοντικών γεγονότων, με βάση τις παρελθούσες καταστάσεις του προβλήματος. Το 1973 ο Rechenberg και το 1975 ο Schwefel εισήγαγαν τη μέθοδο των Εξελικτικών Στρατηγικών (Rechenberg, 1973) (Schwefel, 1975) για την επίλυση δύσκολων πειραματικών προβλημάτων με πολλές παραμέτρους. Σε αυτές τις τρεις μεθόδους βασίζεται σήμερα το μεγαλύτερο μέρος των σύγχρονων Εξελικτικών Αλγορίθμων.

Τα επόμενα χρόνια, με τη ραγδαία εξέλιξη της τεχνολογίας, άρχισε να διαφαίνεται η πρακτική χρησιμότητα των Εξελικτικών Αλγορίθμων. Έτσι υπήρξε μεγάλη προσέλκυση προς αυτό τον τομέα, με αποτέλεσμα τη μεγάλη βελτίωση και ανάπτυξη παλαιών και νέων μεθόδων. Το 1986 ο Glover ανέπτυξε μία μέθοδο η οποία χρησιμοποιεί τη γειτονική αναζήτηση (Glover, 1986) για εύρεση καλύτερης λύσης, και σημαδεύει περιοχές λύσεων που έχουν ήδη εξεταστεί για μη περαιτέρω έλεγχο. Ο Koza το 1990 δημιούργησε έναν κώδικα που μπορούσε να μεταβάλλει τον εαυτό του για την εύρεση καλύτερων λύσεων με τη χρήση Γενετικού Προγραμματισμού (Koza, 1990). Το 1991 οι Kelly και Davis με τη χρήση γενετικών αλγορίθμων (Kelly &



Davis, 1991) κατάφεραν να βελτιώσουν τους υπάρχοντες αλγόριθμους. Εκείνη, καθώς και τη μετέπειτα εποχή δημιουργήθηκαν πολλές άλλες μέθοδοι με τις πιο γνωστές να είναι του Moscato με την έννοια του μιμητικού αλγόριθμου (Moscato, 1989), του Reynolds με την εισαγωγή του διαστήματος πίστης (Reynolds et al., 1994), του Forrest με την προσπάθεια μίμησης του ανθρώπινου ανοσοποιητικού συστήματος (Forrest et al., 1994) για επίλυση προβλημάτων, και το 2002 με την ανάπτυξη των αλγορίθμων διασποράς από τους Larrañaga και Lozano (Larrañaga & Lozano, 2002).

Πολύ σημαντικό σημείο ιστορικά αποτελεί η ανάπτυξη της μεθόδου Particle Swarm Optimization, από τους Kennedy και Eberhart (Kennedy & Eberhart, 1995). Η μέθοδος αυτή χρησιμοποιεί ένα σμήνος σωματιδίων για να ερευνησει μεγάλες περιοχές λύσεων με στόχο μία πιθανόν καλή λύση, χωρίς όμως να γνωρίζει εάν είναι και η βέλτιστη. Μπορεί να ελέγξει προβλήματα με μεγάλο θόρυβο καθώς και προβλήματα χρονικά μεταβαλλόμενα.

### **1.3 Πρακτικές Εφαρμογές**

Ήδη από τις ιστορικές αναφορές διαφαίνεται ότι οι Εξελικτικοί Αλγόριθμοι μπορούν να έχουν πολύ μεγάλη πρακτική εφαρμογή, καθώς αυτός είναι και ο κύριος σκοπός ανάπτυξής τους, καθώς και ο λόγος της μεγάλης άνθησης που γνωρίζει ο τομέας τα τελευταία χρόνια. Η πρακτική εφαρμογή τους είναι κυρίως στη μηχανική και τα οικονομικά, αλλά υπάρχουν και σε πολλούς άλλους τομείς. Κάποιες από τις πιο διαδεδομένες είναι (Back et al., 1997) :

- Σε εφαρμογές μηχανικού σχεδιασμού με συνεχείς ή/και διακριτές παραμέτρους, όπως συστήματα αυτομάτου ελέγχου, ηλεκτρομαγνητικά συστήματα, υπολογισμός ψηφιακών φίλτρων, μοντελοποίηση μηχανικών εξαρτημάτων κ.α.
- Σε δυαδικά προβλήματα, όπως σε βέλτιστη ανάθεση εργασιών στο προσωπικό ενός τμήματος, ανάθεση καθηγητών σε σχολικά τμήματα κ.α.
- Συνδυαστικά προβλήματα βελτιστοποίησης με εκπροσώπηση μετάλλαξης των λύσεων στον τομέα του χρονικού προγραμματισμού, όπως λ.χ. ο προγραμματισμός παραγωγής, ο προγραμματισμός ενός έργου, ο προγραμματισμός βέλτιστων δρομολογίων φορτηγών κ.α.



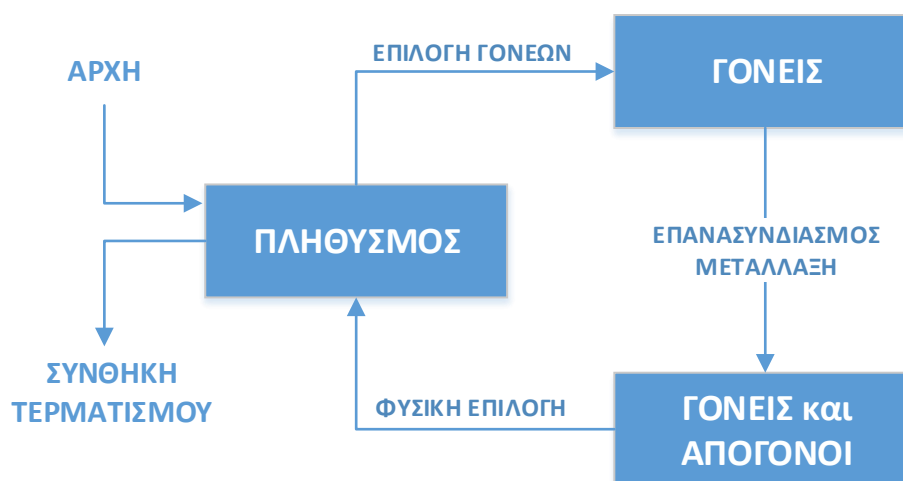
## **ΚΕΦΑΛΑΙΟ 2**

### **ΟΙ ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ**



## 2.1 Περιγραφή ενός Εξελικτικού Αλγορίθμου

Η βασική ιδέα πίσω από όλες τις τεχνικές των εξελικτικών αλγορίθμων είναι η ίδια: η δημιουργία ενός πειστικού περιβάλλοντος σε ένα πλήθος ατόμων, με στόχο τον εξαναγκασμό της φυσικής επιλογής και τη βελτίωση της «φυσικής» κατάστασης του πληθυσμού (Eiben & Smith, 2003). Έτσι εάν θέλουμε να μεγιστοποιήσουμε μία ποιοτική λειτουργία, δημιουργούμε κάποιες τυχαίες υποψήφιες λύσεις και τις ελέγχουμε πάνω σε αυτή την ποιοτική λειτουργία ως μία θεωρητική τιμή μέτρησης της φυσικής κατάστασης, με στόχο την υψηλότερη δυνατή. Με βάση αυτή την τιμή της φυσικής κατάστασης επιλέγονται οι καλύτεροι υποψήφιοι, οι οποίοι θα αναπαράγουν την επόμενη γενιά του πληθυσμού. Αυτό γίνεται είτε με ανασυνδυασμό, όπου δύο υποψήφιοι (γονείς) συνεργάζονται για να αναπαράγουν έναν ή περισσότερους νέους υποψήφιους (απόγονοι), είτε με μετάλλαξη, όπου ένας υποψήφιος (γονέας) αναπαράγει ένα νέο υποψήφιο (απόγονος). Με αυτές τις διαδικασίες δημιουργούνται οι νέοι υποψήφιοι οι οποίοι ανταγωνίζονται με τους παλιούς για μία θέση στη νέα γενιά του πληθυσμού. Αυτές οι διαδικασίες επαναλαμβάνονται έως ότου βρεθεί μία κατάλληλη λύση, με βάση την τιμή της φυσικής κατάστασης, από τους υποψήφιους ή έως ότου υπερβούμε ένα όριο τερματισμού της διαδικασίας.



Σχήμα 2.1 Γενικό σχεδιάγραμμα εξελικτικών αλγορίθμων

Με την εφαρμογή των δύο βασικών τεχνικών, δηλαδή του ανασυνδυασμού (και μετάλλαξης), και των δύο επιλογών, επιτυγχάνεται η βελτίωση της φυσικής κατάστασης του πληθυσμού από επανάληψη σε επανάληψη. Αυτό συμβαίνει γιατί με τον ανασχεδιασμό και τη μετάλλαξη δημιουργείται η απαραίτητη ποικιλομορφία στον

πληθυσμό και ωθείται η εξερεύνηση σε νέες καταστάσεις, και με τη φυσική επιλογή (και επιλογή των γονέων) δίνεται ώθηση στις ποιοτικές φυσικές καταστάσεις. Έτσι μπορούμε, σε βάθος επαναλήψεων, να προσεγγίσουμε μία λύση με βέλτιστη φυσική κατάσταση.

## **2.2 Βασικά στοιχεία των εξελικτικών αλγορίθμων**

Υπάρχουν κάποια βασικά στοιχεία που έχουν οι εξελικτικοί αλγόριθμοι (EA). Αυτά τα στοιχεία είναι:

- Εκπροσώπηση (Ορισμός των ατόμων)
- Συνάρτηση αξιολόγησης (Συνάρτηση φυσικής κατάστασης)
- Πληθυσμός
- Μηχανισμός επιλογής γονέων
- Παράγοντες απόκλισης (ή μεταβολής)
- Μηχανισμός επιλογής επιζώντων (φυσική επιλογή ή αντικατάσταση)
- Αρχικοποίηση
- Συνθήκη τερματισμού

Αυτά τα βασικά στοιχεία πρέπει να προσδιοριστούν προκειμένου να ορίσουμε ένα συγκεκριμένο EA. Παρακάτω περιγράφουμε τη λειτουργία του καθενός από αυτά.

### ➤ **Εκπροσώπηση (Ορισμός των ατόμων)**

Το πρώτο βασικό βήμα στους EA είναι η σύνδεση των στοιχείων του αρχικού προβλήματος στο χώρο επίλυσης προβλημάτων, όπου θα γίνει η εξελικτική διαδικασία. Οι πιθανές λύσεις που σχηματίζονται με βάση τα αρχικά στοιχεία του προβλήματος ονομάζονται «φαινότυποι», και οι κωδικοποιήσεις τους, τα άτομα εντός του EA ονομάζονται «γενότυποι». Αυτό το βήμα ονομάζεται εκπροσώπηση, καθώς ορίζει ένα μείγμα από τους φαινότυπους στους γενότυπους, έτσι ώστε αυτοί να τους εκπροσωπούν. Οι φαινότυποι μπορεί να διαφέρουν πολύ από τους γενότυπους, για αυτό η εξελικτική διαδικασία λαμβάνει χώρα στην περιοχή των γενότυπων. Στο τέλος ο καλύτερος γενότυπος αποκωδικοποιείται για να μας δώσει έναν καλό φαινότυπο,

έτσι ώστε να έχουμε μία καλή λύση του προβλήματος. Η διαδικασία κωδικοποίησης, αποκωδικοποίησης αποτελούν και αυτές κομμάτι της «εκπροσώπησης».

➤ **Συνάρτηση αξιολόγησης (Συνάρτηση φυσικής κατάστασης)**

Ο ρόλος της συνάρτησης αξιολόγησης είναι η εκπροσώπηση των απαιτήσεων, στις οποίες πρέπει να προσαρμοστούν τα άτομα. Ουσιαστικά αυτό που κάνει είναι να ορίζει την έννοια της «βελτίωσης». Από τη σκοπιά του προβλήματος, εκφράζει το «θέμα» που πρέπει να λυθεί μέσα στο περιβάλλον της εξελικτικής διαδικασίας. Αυτή η συνάρτηση συντίθεται από μία μέτρηση τιμής ποιότητας από το φαινότυπο χώρο και από την αντίστροφη εκπροσώπηση. Επειδή συνήθως στους ΕΑ τα αρχικά προβλήματα, που έχουμε να λύσουμε, είναι προβλήματα βελτιστοποίησης, η συνάρτηση αξιολόγησης ονομάζεται και αντικειμενική συνάρτηση και μπορεί να είναι ίδια και για το αρχικό πρόβλημα αλλά και για αυτό μέσα στο πεδίο που λαμβάνει χώρα η εξελικτική διαδικασία.

➤ **Πληθυσμός**

Ο ρόλος του πληθυσμού είναι να «κρατάει» τις πιθανές λύσεις, δηλαδή ο πληθυσμός είναι μία σειρά από γενοτύπους. Ο πληθυσμός σχηματίζει τη μονάδα της εξέλιξης. Τα άτομα είναι στατικά αντικείμενα, δεν αλλάζουν ούτε προσαρμόζονται, ο πληθυσμός είναι αυτός που αλλάζει και προσαρμόζεται. Ο πληθυσμός μπορεί να ορίζεται με απλούς τρόπους όπως από το πόσα άτομα περιλαμβάνει (μέγεθος πληθυσμού), αλλά μπορεί να περιλαμβάνει και πολύπλοκα στοιχεία όπως μετρητή απόστασης που διανύθηκε και τεχνικές αλληλεξάρτησης γειτονικών ατόμων. Οι τεχνικές των δύο επιλογών (επιλογή γονέων και φυσικής επιλογής) δουλεύουν στο επίπεδο πληθυσμού, δηλαδή λαμβάνουν υπόψη τους όλα τα άτομα που περιλαμβάνει ο εκάστοτε πληθυσμός και επιλέγουν πάντα με βάση την κατάστασή του, εκείνη τη σχετική στιγμή. Σχεδόν σε όλους τους εξελικτικούς αλγορίθμους ο πληθυσμός παραμένει σταθερός καθ' όλη την εξελικτική διαδικασία. Ένα σημαντικό χαρακτηριστικό του πληθυσμού είναι η «διαφορετικότητα», η οποία εκφράζει το πλήθος των διαφορετικών λύσεων που έχουμε σε αυτόν. Δεν υπάρχει ένας συγκεκριμένος τρόπος μέτρησης της διαφορετικότητας, συνήθως όμως χρησιμοποιούνται είτε το πλήθος των διαφορετικών τιμών της συνάρτησης αξιολόγησης, είτε το πλήθος των διαφορετικών φαινοτύπων ή γενοτύπων, αλλά μπορεί να μετρηθεί και με στατιστικά στοιχεία.

➤ **Μηχανισμός επιλογής γονέων**

Ο ρόλος της επιλογής γονέων είναι να ξεχωρίσει, ανάμεσα στα άτομα του πληθυσμού, τα καλύτερα, με βάση την ποιότητά τους, για να γίνουν γονείς την επόμενη γενιά. Ένα άτομο θεωρείται πατέρας εάν έχει επιλεγεί για να χρησιμοποιηθεί σε μία υφιστάμενη μεταβολή για την παραγωγή απογόνου. Η επιλογή γονέων μαζί με τη φυσική επιλογή είναι υπεύθυνες για τις πιέσεις που ωθούν σε βελτίωση της ποιότητας. Στον εξελικτικό προγραμματισμό, η επιλογή των γονέων είναι πιθανολογική, δηλαδή παρόλο που τα άτομα με την καλύτερη ποιότητα έχουν μεγαλύτερη πιθανότητα να επιλεγούν, δίνεται συχνά μία μικρή πιθανότητα και στα άτομα με χαμηλή ποιότητα να επιλεγούν. Εάν δεν συνέβαινε αυτό, η όλη αναζήτηση θα γινόταν «άπληστη» και θα παγιδευόταν εύκολα σε τοπικά βέλτιστα (ακρότατα).

➤ **Παράγοντες απόκλισης (ή μεταβολής)**

Η λειτουργία των παραγόντων απόκλισης είναι να δημιουργούν νέα άτομα (απογόνους) από τα παλιά (γονείς), που στο αρχικό πρόβλημα αντιπροσωπεύουν τις υποψήφιες λύσεις. Στον εξελικτικό προγραμματισμό υπάρχουν δύο βασικοί τύποι τέτοιων παραγόντων που διακρίνονται με βάση τον αριθμό των εισόδων στοιχείων (άτομα που δέχονται) που έχουν:

- **Παράγοντας μετάλλαξης.** Ο παράγοντας απόκλισης που δέχεται ως είσοδο μόνο ένα άτομο ονομάζεται «μετάλλαξη». Εφαρμόζεται σε ένα γενότυπο γονέα και παράγει έναν ελαφρώς τυποποιημένο απόγονό του. Ο παράγοντας μετάλλαξης είναι πάντα στοχαστικός, δηλαδή η έξοδος του (απόγονος) προέρχεται από την έκβαση μίας σειράς τυχαίων επιλογών. Ένας ευρετικός παράγοντας ενός συγκεκριμένου προβλήματος που εφαρμόζεται σε ένα άτομο μπορεί να χαρακτηριστεί ως μετάλλαξη, επειδή έχει μόνο ένα στοιχείο εισόδου. Παρόλα αυτά, στη γενική έννοια της μετάλλαξης υποτίθεται ότι πρέπει η έξοδος να είναι τυχαία και αβίαστη, για αυτό ο ρόλος της μετάλλαξης στον εξελικτικό προγραμματισμό διαφέρει ανάλογα με την προσέγγιση. Για παράδειγμα, στον Γενετικό Προγραμματισμό η μετάλλαξη πολύ συχνά δεν χρησιμοποιείται καθόλου και στους Γενετικούς Αλγόριθμους είναι ένας υποβοηθητικός παράγοντας, ενώ στον εξελικτικό προγραμματισμό είναι ο μοναδικός παράγοντας που χρησιμοποιείται, και έχει επιφορτισθεί με όλη την εργασία της αναζήτησης.



- **Παράγοντας ανασυνδυασμού.** Ο παράγοντας απόκλισης που δέχεται ως είσοδο δύο άτομα (γενοτύπους γονείς) και συνδυάζει τις πληροφορίες τους για την παραγωγή δύο ή περισσότερων ατόμων (γενότυποι απόγονοι) ονομάζεται ανασυνδυασμός ή διασταύρωση. Και ο παράγοντας ανασυνδυασμού, όπως και ο μετάλλαξης, είναι στοχαστικός, δηλαδή το ποια πληροφορία θα παρθεί από τον κάθε γονέα για τη δημιουργία του απόγονου είναι αποτέλεσμα τυχαίων επιλογών. Ο παράγοντας του ανασυνδυασμού διαφέρει, και αυτός, ανάλογα με την προσέγγιση, δηλαδή στον Γενετικό Προγραμματισμό είναι συνήθως ο μοναδικός παράγοντας και στους Γενετικούς Αλγορίθμους είναι ο βασικός παράγοντας της εργασίας της αναζήτησης, ενώ στον εξελικτικό προγραμματισμό δεν χρησιμοποιείται ποτέ. Η βασική αρχή του ανασυνδυασμού είναι το «ζευγάρισμα» δύο ατόμων με διαφορετικές αλλά επιθυμητές πληροφορίες με στόχο την παραγωγή απογόνων που συνθέτουν όλες αυτές τις δύο επιθυμητές πληροφορίες, κάτι το οποίο συμβαίνει και στη βιολογία με τους εκτροφείς ζώων. Στους εξελικτικούς αλγορίθμους ο ανασυνδυασμός γίνεται τυχαία. Έτσι είναι δεδομένο ότι θα παραχθούν και απόγονοι που θα είναι σχεδόν ίσοι ή χειρότεροι από τους γονείς, αλλά θα παραχθούν και κάποιοι που θα έχουν βελτιωμένα χαρακτηριστικά σε σχέση με αυτούς. Ακόμα, ο συνδυασμός πάνω από δύο ατόμων ως γονείς, παρόλο που δεν είναι συνηθισμένος και δεν υπάρχει αντίστοιχη διαδικασία στη φύση, είναι μαθηματικά εφικτός, και από έρευνες (Eiben et al., 1994) έχει φανεί ότι έχει θετικά οφέλη πάνω στην εξελικτική διαδικασία.

➤ **Μηχανισμός επιλογής επιζώντων (φυσική επιλογή ή αντικατάσταση)**

Ο ρόλος του μηχανισμού της φυσικής επιλογής είναι να ξεχωρίζει τα άτομα με βάση την ποιότητά τους. Είναι παρόμοιος μηχανισμός με την επιλογή των γονέων αλλά εφαρμόζεται σε διαφορετικό επίπεδο της εξελικτικής διαδικασίας, σε αυτό ακριβώς μετά τη δημιουργία των απογόνων. Επειδή το μέγεθος του πληθυσμού είναι (σχεδόν πάντα) σταθερό, πρέπει να γίνει επιλογή των ατόμων που θα δημιουργήσουν τη νέα γενιά του πληθυσμού. Αυτό γίνεται με δύο τρόπους: Ο πιο συνηθισμένος είναι να βασίζεται μόνο στη φυσική κατάσταση όλων των ατόμων (γονείς και απόγονοι) και να επιλέγονται αυτοί με την καλύτερη, αλλά υπάρχει και ο δεύτερος τρόπος όπου επιλέγονται αυτοί με την καλύτερη φυσική κατάσταση μόνο από τους απογόνους. Έτσι φαίνεται ότι, σε αντίθεση με το μηχανισμό επιλογής γονέων που είναι μία

στοχαστική διαδικασία, ο μηχανισμός φυσικής επιλογής είναι, συνήθως, ντετερμινιστικός. Πολλές φορές ονομάζεται και «αντικατάσταση» γιατί σε κάποιες διαδικασίες μπορεί να έχουμε πολλούς γονείς και ελάχιστους απογόνους για να φτιάξουμε τη νέα γενιά του πληθυσμού, και συνεπώς δεν επιλέγουμε πια τους επιζώντες, αφού σχεδόν οι πιο πολύ γονείς θα επιζήσουν, αλλά επιλέγουμε αυτούς που θα αντικατασταθούν.

### ➤ Αρχικοποίηση

Η διαδικασία της αρχικοποίησης είναι, στους πιο πολλούς ΕΑ, αρκετά απλή: Ο αρχικός πληθυσμός επιλέγεται από άτομα που δημιουργήθηκαν με τυχαιότητα. Επίσης μπορούν να χρησιμοποιηθούν ευρετικές τεχνικές, συγκεκριμένες για κάθε πρόβλημα, με στόχο την δημιουργία αρχικού πληθυσμού με καλύτερη φυσική κατάσταση, αλλά απαιτούν περισσότερους υπολογιστικούς πόρους, χωρίς πάντα να δίνουν τα ανάλογα οφέλη.

### ➤ Συνθήκη Τερματισμού

Υπάρχουν δύο περιπτώσεις που μπορούν να μας ορίσουν μία συνθήκη τερματισμού. Η μία είναι όταν γνωρίζουμε το επίπεδο (τιμή) της βέλτιστης φυσικής κατάστασης του προβλήματος, οπότε και μπορούμε να ορίσουμε τον τερματισμό του όταν «πλησιάσει» κοντά, με μία ακρίβεια  $\epsilon > 0$ , σε αυτό το επίπεδο της φυσικής κατάστασης. Επειδή όμως οι ΕΑ είναι στοχαστικές μέθοδοι, δεν μπορούνε, συνήθως, να εγγυηθούν ότι θα πλησιάσουν αυτό το βέλτιστο, συνεπώς αυτή η συνθήκη τερματισμού δεν είναι σίγουρο ότι θα ενεργοποιηθεί ποτέ σε έναν αλγόριθμο. Η άλλη συνθήκη τερματισμού μπορεί να είναι:

- Ένα μέγιστο χρονικό όριο λειτουργίας της Κεντρικής Μονάδας Επεξεργασίας (CPU).
- Ένα μέγιστο πλήθος αξιολογήσεων της φυσικής κατάσταση των υποψήφιων λύσεων του προβλήματος.
- Μία συνθήκη τερματισμού που ενεργοποιείται, εάν η φυσική κατάσταση παραμένει η ίδια για ένα συγκεκριμένο πλήθος επαναλήψεων ή αξιολογήσεών της.

- Μία συνθήκη τερματισμού που ενεργοποιείται, εάν η τιμή της «διαφορετικότητας» του πληθυσμού γίνει μικρότερη από ένα συγκεκριμένο μέγεθος.

Συνήθως, η συνθήκη τερματισμού ορίζεται με παραπάνω από έναν τρόπους. Δηλαδή μπορούμε να ορίσουμε ότι ο αλγόριθμος θα τερματιστεί εάν πλησιάσουμε πολύ κοντά στο βέλτιστο ή εάν ικανοποιηθεί ένα από τα παραπάνω κριτήρια, ό,τι έρθει πρώτο. Εάν δεν ξέρουμε το βέλτιστο, κάτι συνηθισμένο για προβλήματα που λύνουμε με ΕΑ, επιλέγουμε κάποιο από τα παραπάνω κριτήρια ή κάποιο άλλο παρόμοιο που μπορεί να μας εγγυηθεί τον τερματισμό του αλγορίθμου.

## **2.3 Τρόποι διαχείρισης περιορισμών στους εξελικτικούς αλγορίθμους**

### **2.3.1 Βασικός ορισμός ενός προβλήματος**

Αρχικά σε ένα εξελικτικό αλγόριθμο πρέπει ορίσουμε το πρόβλημα που θέλουμε να βελτιστοποιήσουμε. Αυτό γίνεται, ορίζοντας την αντικειμενική συνάρτηση, τους περιορισμούς και το πεδίο τιμών.

Αντικειμενική συνάρτηση:

$$\text{Εύρεση } \mathbf{X} \text{ που βελτιστοποιεί την } f(\mathbf{X}) \quad (2.1)$$

Και υπόκειται στους περιορισμούς:

$$\begin{aligned} g_i(\mathbf{X}) &\leq 0, & i &= 1, \dots, n \\ h_j(\mathbf{X}) &= 0, & j &= 1, \dots, p \end{aligned} \quad (2.2)$$

Όπου  $\mathbf{X} (= x_1, x_2, \dots, x_r)$  είναι το διάνυσμα λύσης και  $g_i(\mathbf{X}), h_j(\mathbf{X})$  οι περιορισμοί ισότητας και ανισότητας αντίστοιχα που ισχύουν για το πεδίο ορισμού της αντικειμενικής  $F \subseteq S$ , με  $S$  να εκφράζει όλο το πεδίο της αναζήτησης. Σε αυτό το κεφάλαιο, για λόγους απλότητας, θα χρησιμοποιήσουμε το  $\varphi_i(\mathbf{X}) \leq 0, i = 1, \dots, m$  και για τα δύο είδη περιορισμών.

### 2.3.2 Μέθοδοι διαχείρισης περιορισμών

Γενικά, στο θέμα των τρόπων διαχείρισης περιορισμών υπάρχουν πολλά άρθρα με διάφορες μεθόδους και παραλλαγές τους. Στο υποκεφάλαιο αυτό θα αναλύσουμε κάποιες από τις πιο συχνά εφαρμοζόμενες στους ΕΑ (Coello C. A., 1999). Αυτές είναι:

- Χρήση συναρτήσεων ποινής.
- Διατήρηση ενός πληθυσμού εφικτών λύσεων με χρήση ειδικής «εκπροσώπησης» και γενετικών παραγόντων.
- Διαχωρισμός αντικειμενικής συνάρτησης από περιορισμούς.
- Άλλες μέθοδοι.

#### 2.3.2.1 Χρήση συναρτήσεων ποινής

Η πιο διαδεδομένη μέθοδος στους ΕΑ είναι αυτή της χρήσης συναρτήσεων ποινής για τη διαχείριση των περιορισμών. Η βασική προσέγγιση αυτής της μεθόδου είναι ο ορισμός της τιμής της φυσικής κατάστασης ενός ατόμου  $i$  με την επέκταση της αντικειμενικής συνάρτησης σε αυτή τη μορφή (Michalewicz & Dasgupta, 1997):

$$fitness_i(\mathbf{X}) = f_i(\mathbf{X}) \pm Q_i \quad (2.3)$$

Όπου  $Q_i$  αντιπροσωπεύει είτε μία τιμή ποινής για μία μη-εφικτή λύση ενός ατόμου  $i$ , είτε το κόστος του να διορθωθεί αυτό το άτομο για να γίνει εφικτή λύση, με το  $Q_i$  να παίρνει τιμή 0 εάν δεν έχουμε παραβίαση περιορισμών. Στους Γενετικούς Αλγορίθμους συνήθως έχουμε:

$$Q_i = c \times \sum_{i=1}^m \Omega[\varphi_i(\mathbf{X})] \quad (2.4)$$

Με  $\Omega[\varphi_i(\mathbf{X})] = \varphi_i(\mathbf{X})^2$  για όλους τους παραβιασμένους περιορισμούς  $i$ , και με  $c$  να είναι μία σταθερά ποινής ορισμένη από το χρήστη. Αν και συνήθως οι στρατηγικές εξέλιξης χρησιμοποιούν ποινές «θανάτου» για τις μη-εφικτές λύσεις (αναλύουμε παρακάτω), υπάρχουν προτάσεις και για χρήση συναρτήσεων ποινής.

Ακόμα όπως αναφέραμε και στον υπολογισμό του  $Q_i$ , η τιμή της ποινής εισέρχεται μόνο εάν έχουμε παραβίαση περιορισμού, άρα:

$$\varphi_i(\mathbf{X}) = \begin{cases} \max(g_i(\mathbf{X}), 0), & \text{για } 1 \leq i < n \\ |h_i(\mathbf{X})|, & \text{για } q + 1 \leq i \leq m \end{cases} \quad (2.5)$$

Σε γενικές γραμμές, η ποινή πρέπει να πρέπει να παίρνει τιμές πολύ χαμηλές, κοντά στο όριο του να γίνει η βέλτιστη λύση μη-εφικτή. Αυτό λέγεται «κανόνας ελάχιστης ποινής» και έχει αποδειχθεί (Davis, 1987) ότι εάν η τιμή της ποινής είναι πολύ υψηλή ή οι περιορισμοί δεν είναι «οριακά», τότε το πρόβλημα δυσκολεύει αρκετά για τους Γενετικούς Αλγόριθμους.

#### ➤ Μέθοδος Runarsson & Yao

Η προσέγγιση αυτή (Runarsson & Yao, 2005) αποτελεί την πιο απλή μέθοδο εφαρμογής των συναρτήσεων ποινής. Η σχέση που χρησιμοποιείται είναι της μορφής:

$$fitness_i(\mathbf{X}) = f(\mathbf{X}) \pm \lambda * \sum_{j=1}^m \varphi_j(\mathbf{X}) \quad (2.6)$$

Με το  $\lambda$  να είναι μία σταθερά που ορίζεται από το χρήστη, και  $j(=1,2,\dots,m)$  για κάθε περιορισμό .

#### ➤ Στατικές ποινές

Στη προσέγγιση αυτή (Homaifar et al., 1994) προτάθηκε η χρήση πολλών επιπέδων παραβίασης, όπου η τιμή της σταθεράς παραβίασης αυξάνει όσο μεγαλώνει και το επίπεδο παραβίασης. Η σχέση που χρησιμοποιείται είναι (Michalewicz et al., 1996):

$$fitness_i(\mathbf{X}) = f_i(\mathbf{X}) + \sum_{j=1}^m R_{k,j} \varphi_j^2(\mathbf{X}) \quad (2.7)$$

Όπου  $R_{k,j}$  είναι η σταθερά ποινής για  $k(=1,2,\dots,o)$  επίπεδο παραβίασης και  $j(=1,2,\dots,m)$  περιορισμό. Το μεγάλο πρόβλημα σε αυτή την τεχνική είναι το μεγάλο πλήθος παραμέτρων  $[m(2 \cdot o + 1)]$  που απαιτούνται για την εφαρμογή της.

#### ➤ Δυναμικές Ποινές

Οι Joines και Houck (Joines & Houck, 1994) πρότειναν τη χρήση ποινών που αλλάζουνε συναρτήσει του χρόνου. Έτσι ανάλογα με τη γενιά(επανάληψη) έχουμε:

$$fitness_i(\mathbf{X}) = f_i(\mathbf{X}) + (C \times t)^\alpha \sum_{j=1}^m |\varphi_j(\mathbf{X})|^\beta \quad (2.8)$$

Όπου  $C$ ,  $\alpha$  και  $\beta$  είναι σταθερές ορισμένες από το χρήστη. Σύμφωνα με κάποιους ερευνητές οι δυναμικές ποινές δουλεύουν καλύτερα από τις στατικές, αλλά είναι πιο δύσκολο να βρεθούν οι κατάλληλες τιμές των σταθερών και το αποτέλεσμα του αλγόριθμου είναι πολύ πιο ευαίσθητο στις τιμές αυτών.

### ➤ Ποινές με ανόπτησης

Η μέθοδος αυτή (Michalewicz & Attia, 1994) βασίζεται στην ιδέα της προσομοίωσης της ανόπτησης, δηλαδή οι σταθερές της ποινής αλλάζουν μία φορά ανά πολλές γενεές(επαναλήψεις), κυρίως για όταν έχει παγιδευτεί σε τοπικό ακρότατο. Έτσι η ποινή μεγαλώνει συναρτήσει του χρόνου που τρέχει ο αλγόριθμος, με αποτέλεσμα οι μη-εφικτές λύσεις να έχουν μεγάλη ποινή στις τελευταίες γενεές.

Η μέθοδος αυτή απαιτεί το χωρισμό των περιορισμών σε 4 κατηγορίες: γραμμικές ισότητες, γραμμικές ανισότητες, μη-γραμμικές ισότητες και μη-γραμμικές ανισότητες. Επίσης δημιουργούμε ένα «A» που περιλαμβάνει όλες τις παραβιασμένες μη-γραμμικές ανισότητες και όλες τις μη-γραμμικές ισότητες. Όλα αυτά εφαρμόζονται με τη σχέση:

$$fitness_i(\mathbf{X}) = f_i(\mathbf{X}) + \frac{1}{2\tau} \sum_{j \in A} \varphi_j^2(\mathbf{X}) \quad (2.9)$$

Όπου  $\tau$  είναι το «πρόγραμμα ψύξης» (Kirkpatrick et al., 1983). Το ενδιαφέρον σε αυτή την εφαρμογή είναι ότι ο αρχικός πληθυσμός περιλαμβάνει ένα μόνο άτομο σε πολλά «αντίγραφα», καθώς έχειδειχθεί ότι δεν χρειάζονται παραπάνω (Michalewicz & Schoenauer, 1996). Σε κάθε επανάληψη η θερμοκρασία  $\tau$  μειώνεται και ως νέος πληθυσμός θεωρείται το καλύτερο άτομο της προηγούμενης επανάληψης. Η όλη διαδικασία σταματάει όταν η θερμοκρασία  $\tau$  φτάσει σε μία συγκεκριμένη τιμή  $\tau_f$  που έχει οριστεί από τον χρήστη.

Το μεγάλο πρόβλημα σε αυτή τη μέθοδο είναι ότι η λύση του προβλήματος έχει μεγάλη ευαισθησία στις τιμές των παραμέτρων, και θεωρείται αρκετά δύσκολο να επιλεγεί ένα κατάλληλο «πρόγραμμα ψύξης».

➤ **Προσαρμοστική Ποινή**

Οι Bean και Hadj-Alouane (Bean & Hadj-Alouane, 1992) δημιούργησαν μία μέθοδο που χρησιμοποιεί συνάρτηση ποινής η οποία λαμβάνει ανάδραση από διαδικασία αναζήτησης. Κάθε άτομο αξιολογείται από τη σχέση:

$$fitness_i(\mathbf{X}) = f_i(\mathbf{X}) + \lambda(t) \sum_{j=1}^m \varphi_j^2(\mathbf{X}) \quad (2.10)$$

Με το  $\lambda(t)$  να ενημερώνεται σε κάθε γενιά  $t$  ακολούθως:

$$\lambda(t+1) = \begin{cases} (1/\beta_1) \cdot \lambda(t), & \text{στην περίπτωση 1} \\ \beta_2 \cdot \lambda(t), & \text{στην περίπτωση 2} \\ \lambda(t), & \text{σε άλλη περίπτωση} \end{cases} \quad (2.11)$$

Όπου περίπτωση #1 είναι όταν το καλύτερο άτομο στην προηγούμενη  $k$  γενιά ήταν πάντα εφικτή λύση, και περίπτωση #2 εάν δεν ήταν ποτέ εφικτή λύση, με  $\beta_1, \beta_2 > 1$  και  $\beta_1 \neq \beta_2$ . Δηλαδή η παράμετρος ποινής  $\lambda(t+1)$  για τη γενιά  $t+1$  μειώνεται εάν όλα τα βέλτιστα άτομα στην προηγούμενη γενιά  $k$  ήταν εφικτές λύσεις, και αυξάνεται εάν δεν ήταν κανένα εφικτή λύση. Εάν είχαμε συνδυασμό εφικτών και μη-εφικτών η παράμετρος αυτή δεν αλλάζει τιμή.

Το δύσκολο σε αυτή τη μέθοδο είναι η επιλογή του «κενού» των γενναίων, δηλαδή του  $k$ , καθώς και οι επιλογή των  $\beta_1$  και  $\beta_2$ .

➤ **Συνάρτηση ποινής βασισμένη στο βαθμό που είναι εφικτή μία λύση**

Ο Deb (Deb, 2000) πρότεινε μία ενδιαφέρουσα προσέγγιση όπου ένα άτομο αξιολογείται ακολούθως:

$$fitness_i(\mathbf{X}) = \begin{cases} f_i(\mathbf{X}), & \text{εάν } \varphi_j(\mathbf{X}) \geq 0, \forall j = 1, 2, \dots, m \\ f_{worst} + \sum_{j=1}^m \varphi_j(\mathbf{X}), & \text{αλλιώς} \end{cases} \quad (2.12)$$

Όπου  $f_{worst}$  είναι η τιμή της αντικειμενικής συνάρτησης της χειρότερης εφικτής λύσης του πληθυσμού, και  $\varphi_j(\mathbf{X})$  αφορά μόνο τους περιορισμούς ανισότητας (οι περιορισμοί ισότητας μπορούν να μετατραπούν σε ανισότητας με ανοχή). Εάν δεν υπάρχει εφικτή λύση στον πληθυσμό τότε το  $f_{worst}$  παίρνει τιμή 0.

Χρησιμοποιώντας δυαδικό «τουρνουά» επιλογής, ο Deb χρησιμοποιεί τους παρακάτω κανόνες σύγκρισης για δύο άτομα:

- Μία εφικτή λύση είναι πάντα προτιμώμενη από μία μη-εφικτή.
- Μεταξύ δύο εφικτών λύσεων προτιμάται αυτή με την καλύτερη τιμή αντικειμενικής συνάρτησης.
- Μεταξύ δύο μη-εφικτών λύσεων προτιμάται αυτή με τη μικρότερη παραβίαση περιορισμών.

Δεν χρειάζεται συνάρτηση ποινής καθώς η διαδικασία επιλογής γίνεται με τις παραπάνω λογικές συγκρίσεις. Με αυτή τη μέθοδο η τιμή της συνάρτησης που εφαρμόζουμε στον αλγόριθμο δεν αλλάζει, ενώ με την εισαγωγή των τιμών παραβίασης των περιορισμών στις συγκρίσεις ωθούμε τις μη-εφικτές λύσεις στην περιοχή εφικτών λύσεων. Ακόμα, επειδή οι περιορισμοί δεν είναι ισοδύναμοι, ο Deb τους έκανε κανονικοποίηση.

Το κύριο πρόβλημα αυτής της μεθόδου είναι η δυσκολία της να κρατήσει ένα επίπεδο «διαφορετικότητας» στον πληθυσμό (Deb, 2000).

#### ➤ «Θανατική» ποινή

Η απόρριψη όλων των μη-εφικτών ατόμων είναι η πιο εύκολη και πιο αποδοτική μέθοδος χειρισμού περιορισμών, επειδή όταν μία λύση παραβιάζει έναν περιορισμό, της δίνεται τιμή φυσικής κατάστασης μηδέν και δεν γίνονται άλλοι υπολογισμοί πάνω σε αυτή, σχετικά με το βαθμό που δεν είναι εφικτή. Η συνηθισμένη προσέγγιση αυτής της μεθόδου είναι να επαναλαμβάνεται κυκλικά, με τη δημιουργία ενός νέου σημείου σε κάθε επανάληψη, έως ότου βρεθεί μία εφικτή λύση (Hoffmeister & Sprave, 1996). Αυτή η διαδικασία μπορεί να είναι ιδιαίτερα μακροσκελής σε προβλήματα με δυσκολία στην προσέγγιση της εφικτής περιοχής.

Μια ενδιαφέρουσα πρόταση αυτής της μεθόδου, έγινε από τον Kuri (Kuri & Quezada, 1998), με τη συνάρτηση φυσικής κατάστασης να ορίζεται:

$$fitness_i(\mathbf{X}) = \begin{cases} f_i(\mathbf{X}), & \text{εάν η λύση είναι εφικτή} \\ K + \sum_{i=1}^s \left(\frac{K}{m}\right), & \text{αλλιώς} \end{cases} \quad (2.13)$$



Όπου  $s$  είναι ο αριθμός των περιορισμών που ικανοποιούνται, και  $K$  μία μεγάλη σταθερά. Παρατηρείται ότι όταν μία λύση δεν είναι εφικτή δεν υπολογίζεται η φυσική της κατάσταση, και ότι όλες οι μη-εφικτές λύσεις παίρνουν την ίδια ποινή ανεξάρτητα του πόσο κοντά είναι στην εφικτή περιοχή.

Η μέθοδος της θανατικής ποινής είναι πολύ δημοφιλής στις στρατηγικές εξέλιξης, αλλά περιορίζεται σε προβλήματα με κυρτή εφικτή περιοχή αναζήτησης, η οποία πρέπει να καταλαμβάνει μεγάλο μέρος της συνολικής περιοχής αναζήτησης. Επίσης, δεν εκμεταλλεύεται πληροφορίες από τα μη-εφικτά σημεία που μπορεί να έχουν δημιουργηθεί από τον εξελικτικό αλγόριθμο κατά την αναζήτηση. Παρόλα αυτά ο Michalewicz (Michalewicz Z. , 1995) έδειξε ότι η χρήση της θανατικής ποινής είναι ανώτερη από τη χρήση ποινών που ορίζονται με όρους απόστασης από την εφικτή περιοχή.

### **2.3.2.2 Ειδικές εκπροσωπήσεις και ειδικοί παράγοντες**

Με στόχο την επίλυση κάποιων ιδιαίτερα δύσκολων προβλημάτων, που δεν μπορούσαν να επιλυθούν με την κλασική δυαδική εκπροσώπηση των γενετικών αλγορίθμων, μερικοί ερευνητές αποφάσισαν τη δημιουργία συστημάτων ειδικής εκπροσώπησης. Λόγω αυτής της αλλαγής είναι απαραίτητη και η ανάλογη αλλαγή στη σχεδίαση των γενετικών παραγόντων. Με αυτή την αλλαγή στοχεύεται η απλοποίηση του σχήματος του χώρου αναζήτησης, καθώς και η απλοποίηση των ειδικών παραγόντων που, κανονικά, χρησιμοποιούνται για τη διατήρηση του βαθμού που είναι εφικτή μία συγκεκριμένη λύση σε όλες τις χρονικές στιγμές. Η κύρια εφαρμογή αυτής της προσέγγισης είναι η επίλυση προβλημάτων που ακόμα και η εύρεση, έστω μίας, εφικτής λύσης είναι δύσκολη.

#### **➤ Οι εφαρμογές του Davis**

Στο βιβλίο του Davis «Handbook of Genetic Algorithms» (Davis, 1991) αναφέρονται πολλά παραδείγματα τέτοιων Γενετικών Αλγορίθμων που χρησιμοποιούν ειδικές εκπροσωπήσεις και ειδικούς παράγοντες, για την επίλυση πολύπλοκων ρεαλιστικών προβλημάτων. Π.Χ. Ο Davidor (Davidor, 1990) χρησιμοποίησε ένα μεταβλητού μήκους Γενετικό Αλγόριθμο για να υπολογίσει τροχιές κίνησης ρομπότ, και όρισε έναν ειδικό παράγοντα διασταύρωσης με ονομασία «analogous crossover» (Davidor, 1989), ο οποίος χρησιμοποιούσε «φαινότυπες» ομοιότητες για να ορίσει σημεία

διασταύρωσης στις ακολουθίες των γονέων. Επίσης χρησιμοποίησε πιθανότητες Lamarckian για τη διασταύρωση και τη μετάλλαξη.

Κάποιες εφαρμογές που αναφέρονται στο βιβλίο του Davis είναι: βελτιστοποίηση προγράμματος, σύνθεση αρχιτεκτονικής νευρωνικών δικτύων, και ανάλυση δομής DNA.

Το κύριο πρόβλημα των ειδικών εκπροσωπήσεων και παραγόντων είναι ότι, αν και πολύ χρήσιμες για προβλήματα που σχεδιάστηκαν, δεν μπορούν να γενικευτούν σε άλλα, ακόμα και παρόμοια, προβλήματα.

### ➤ **GENOCOP**

Άλλη μία μέθοδος αυτής της κατηγορίας χειρισμού περιορισμών αποτελεί η «Genetic algorithm for Numerical Optimization for COnstrained Problems», η οποία δημιουργήθηκε από τον Michalewicz (Michalewicz Z. , 1992). Αυτή η προσέγγιση χειρίζεται τους γραμμικούς περιορισμούς εξαλείφοντας τους περιορισμούς ισότητας και σχεδιάζοντας ειδικούς γενετικούς παράγοντες που εγγυώνται τη συγκράτηση όλων των πιθανών λύσεων στον «περιορισμένο» χώρο αναζήτησης. Η συγκεκριμένη μέθοδος είναι ιδιαίτερα πολύπλοκη.

Η μέθοδος αυτή απαιτεί ένα αρχικό εφικτό σημείο, ή έναν πληθυσμό, κάτι που σημαίνει ότι πρέπει ο χρήστης να έχει εισάγει μία τεχνική που να μπορεί να το κάνει αυτό σε ένα λογικό χρόνο. Επίσης, επειδή πρέπει να έχουμε μόνο γραμμικούς περιορισμούς, αναγκαστικά περιοριζόμαστε σε κυρτές περιοχές αναζήτησης.

### ➤ **Μέθοδος συνέπειας περιορισμών Γενετικών Αλγορίθμων**

Ο Kowalczyk (Kowalczyk, 1997) πρότεινε τη χρήση «περιορισμών με συνέπεια» (Kumar, 1992) με στόχο να μειώσει το χώρο αναζήτησης, με την αποτροπή της ενεργοποίησης παραμέτρων που δεν είναι συνεπείς με τους περιορισμούς του προβλήματος. Ο Kowalczyk ανέφερε ότι αυτή η μέθοδος μπορεί να χρησιμοποιηθεί και σε συνδυασμό με οποιαδήποτε άλλη μέθοδο χειρισμού περιορισμών.

Το κύριο πρόβλημα αυτής της μεθόδου είναι το επιπλέον υπολογιστικό κόστος που χρειάζεται για τη «διάδοση» των περιορισμών, διαδικασία που μπορεί να είναι πιο απαιτητική ακόμα και από την ίδια τη βελτιστοποίηση.

### ➤ **Εντοπισμός των ορίων της εφικτής περιοχής**

Η βασική ιδέα αυτής της τεχνικής είναι η αναζήτηση κοντά στα όρια της εφικτής περιοχής. Επειδή σε πολλά μη-γραμμικά προβλήματα βελτιστοποίησης, τουλάχιστον, κάποιοι περιορισμοί θα είναι ενεργοποιημένοι (οριακά στην τιμή πριν παραβιαστούν) στη θέση του ολικού βέλτιστου, είναι απόλυτα λογικό να ψάχνουμε στα όρια της περιοχής μεταξύ εφικτών και μη-εφικτών λύσεων.

Αρχικά αυτή η ιδέα προτάθηκε με το όνομα «strategic oscillation» (Glover, 1977) και χρησιμοποιήθηκε σε βελτιστοποίηση μη-γραμμικών προβλημάτων, και προβλημάτων πεπερασμένων υποψήφιων λύσεων. Η βασική προσέγγιση είναι η χρήση «προσαρμοστικής» ποινής ή παρόμοιας τεχνικής με στόχο να τέμνουμε τα όρια της εφικτής και μη-εφικτής περιοχής επαναληπτικά, αλλάζοντας κάθε φορά τον παράγοντα που επηρεάζει την κατεύθυνση της κίνησης, σε εκείνη τη θέση (Michalewicz & Schoenauer, 1996).

Τα δύο βασικά στοιχεία αυτής της προσέγγισης είναι: μία διαδικασία που να μπορεί να παράγει εφικτά σημεία, και η ύπαρξη γενετικών παραγόντων που να μπορούν να εξερευνούν την εφικτή περιοχή. Επιπλέον, οι παράγοντες αυτοί θα πρέπει να ικανοποιούν τις συνθήκες: η «διασταύρωση» θα πρέπει να παράγει όλα τα σημεία ανάμεσα σε δύο γονείς, και οι μικρές μεταλλάξεις θα πρέπει να δημιουργούν μικρές αλλαγές στην τιμή της φυσικής κατάστασης. Οι Michalewicz και Schoenauer ανέλυσαν ιδιαίτερα τις εφαρμογές αυτής της μεθόδου (Michalewicz & Schoenauer, 1996).

Αν και η μέθοδος παράγει αρκετά καλά αποτελέσματα και είναι αποδοτική, δεν μπορεί να γενικευτεί εύκολα για πολλά προβλήματα, και οι παράγοντες που έχουν σχεδιαστεί είναι ιδιαίτερα εξαρτημένοι από την παραμετροποίηση που κάνει ο χρήστης.

### ➤ **Αποκωδικοποιητές**

Σε αυτή την περίπτωση, σε ένα «χρωμόσωμα» (άτομο, υποψήφια λύση) δίνονται οδηγίες στο πώς να δημιουργήσει μία εφικτή λύση. Κάθε αποκωδικοποιητής επιβάλλει μία σχέση  $T$  μεταξύ μίας εφικτής λύσης και μίας αποκωδικοποιημένης λύσης (Michalewicz & Dasgupta, 1997). Για τη χρήση των αποκωδικοποιητών απαιτείται να ισχύουν αρκετές σημαντικές συνθήκες:

- Για κάθε εφικτή λύση  $s$  υπάρχει μία αποκωδικοποιημένη λύση  $d$ .
- Κάθε αποκωδικοποιημένη λύση  $d$  αντιστοιχεί σε μία εφικτή λύση  $s$ .
- Όλες οι εφικτές λύσεις πρέπει να εκπροσωπούνται από τον αντίστοιχο αριθμό αποκωδικοποιημένων  $d$ .
- Η «μεταμόρφωση» του  $T$  πρέπει να γίνεται γρήγορα (υπολογιστικά).
- Μικρές αλλαγές στην αποκωδικοποιημένη λύση πρέπει να έχουν ως αποτέλεσμα μικρές αλλαγές στην ίδια τη λύση (Michalewicz & Dasgupta, 1997).

Οι Koziel και Michalewicz (Koziel & Michalewicz, 1998) πρότειναν μία «ομοιόμορφη χαρτογράφηση» μεταξύ ενός  $n$  – διάστατου κύβου και του εφικτού χώρου αναζήτησης. Η βασική ιδέα αυτής της προσέγγισης είναι η μεταμόρφωση του αρχικού προβλήματος σε μία άλλη, ανάλογη, συνάρτηση που θα είναι πιο εύκολη για βελτιστοποιηθεί σε Γενετικούς Αλγορίθμους.

Η μέθοδος των Koziel και Michalewicz έχει δύο βασικά προβλήματα: απαιτεί μία νέα παράμετρο που η τιμή της βρίσκεται εμπειρικά (με δοκιμές), και απαιτεί πολλούς υπολογιστικούς πόρους.

### **2.3.2.3 Διαχωρισμός περιορισμών και αντικειμενικής συνάρτησης**

Σε αυτό το υποκεφάλαιο θα αναλύσουμε τις πιο αντιπροσωπευτικές μεθόδους χειρισμού των συναρτήσεων περιορισμών ξεχωριστά από την αντικειμενική συνάρτηση.

#### **➤ Μέθοδος της «συν-εξέλιξης»**

Ο Paredis (Paredis, 1994) πρότεινε μία τεχνική βασισμένη στο μοντέλο της «συν-εξέλιξης», όπου υπάρχουν δύο πληθυσμοί: ο πρώτος πληθυσμός, που περιλαμβάνει τους περιορισμούς που πρέπει να ικανοποιηθούν, και ο δεύτερος που περιλαμβάνει πιθανές λύσεις του προβλήματος. Με τη χρήση της αναλογίας «κυνηγού - θηράματος» η πίεση, στη διαδικασία της φυσικής επιλογής, που ασκείται στον ένα πληθυσμό εξαρτάται από τη φυσική κατάσταση των ατόμων του άλλου πληθυσμού. Έτσι, το άτομο που έχει υψηλή τιμή φυσικής κατάστασης στο δεύτερο πληθυσμό αντιπροσωπεύει μία λύση που ικανοποιεί πολλούς περιορισμούς, ενώ το άτομο που

έχει υψηλή τιμή στον πρώτο πληθυσμό αντιπροσωπεύει έναν περιορισμό που παραβιάζεται από πολλές λύσεις.

Οι λύσεις και οι περιορισμοί έχουν σημεία «συνάντησης», δηλαδή σημεία όπου άτομα που ανήκουν και στους δύο πληθυσμούς αξιολογούνται. Κάθε άτομο τηρεί ιστορικό των «συναντήσεων» του, και η φυσική του κατάσταση υπολογίζεται σύμφωνα με τον αριθμό των τελευταίων  $n$  συναντήσεων ( $n = 25$  από Paredis (Paredis, 1994)). Η βασική ιδέα αυτής της προσέγγισης είναι η βελτίωση της φυσικής κατάστασης των περιορισμών που είναι πιο δύσκολο να ικανοποιηθούν.

Αν και τα αποτελέσματα από δοκιμές (Paredis, 1994) είναι αρκετά ενθαρρυντικά, το κύριο πρόβλημα είναι ότι, επειδή η φυσική κατάσταση υπολογίζεται από τις τελευταίες  $n$  συναντήσεις, εάν οι πιο πολλοί περιορισμοί είναι σχεδόν το ίδιο δύσκολο να ικανοποιηθούν τότε ο αλγόριθμος μπορεί να βρεθεί σε ένα «τέλμα», δηλαδή η διαδικασία της αναζήτησης δεν θα εξελίσσεται περισσότερο.

#### ➤ Ανωτερότητα των εφικτών λύσεων

Οι Powell και Skolnick (Powell & Skolnick, 1993) ενσωμάτωσαν έναν ευρετικό κανόνα, ο οποίος προτάθηκε από τον Richardson (Richardson et al., 1989), για την επεξεργασία μη-εφικτών λύσεων. Αυτός είναι: Οι αξιολογήσεις των εφικτών λύσεων αντιπροσωπεύονται στο διάστημα μεταξύ  $(-\infty, 1)$ , και οι μη-εφικτές στο  $(1, \infty)$ . Τα άτομα αξιολογούνται ακολούθως:

$$fitness_i(\mathbf{X}) = \begin{cases} f_i(\mathbf{X}), & \text{εάν είναι εφικτή λύση} \\ 1 + r \sum_{j=1}^m \varphi(\mathbf{X}), & \text{αλλιώς} \end{cases} \quad (2.14)$$

Όπου  $f_i(\mathbf{X})$  μία «κλιμακωτή» τιμή μέσα στο διάστημα  $(-\infty, 1)$ ,  $\varphi(\mathbf{X})$  μία «κλιμακωτή» τιμή μέσα στο διάστημα  $(1, \infty)$ , και  $r$  να είναι μία σταθερά.

Οι Powell και Skolnick (Powell & Skolnick, 1993) χρησιμοποίησαν μία γραμμική κατάταξη επιλογής με τέτοιο τρόπο ώστε στις αρχικές γενεές η σύγκλιση να είναι αργή, και αργότερα η σύγκλιση εξαναγκάζεται, μεγαλώνοντας τον αριθμό των αντιγράφων των ατόμων με την υψηλότερη κατάταξη.

Το κύριο πρόβλημα αυτής της μεθόδου είναι ότι σε περιπτώσεις όπου οι περιορισμοί είναι πολύ δύσκολο να ικανοποιηθούν (μικρός χώρος αναζήτησης σε σχέση με τον ολικό χώρο) απαιτείται η είσοδος ενός εφικτού σημείου στον αρχικό πληθυσμό.

➤ **Μνήμη με «συμπεριφορά»**

Οι Schoenauer και Ξανθάκης (Schoenauer & Xanthakis, 1993) πρότειναν τη προέκταση μία τεχνικής που ονομάζεται «behavioral memory», η οποία αρχικά είχε προταθεί για τη βελτιστοποίηση προβλημάτων χωρίς περιορισμούς. Η βασική της ιδέα είναι ο χειρισμός των περιορισμών με συγκεκριμένη σειρά:

1. Αρχίζει με έναν τυχαίο πληθυσμό.
2. Ορίζεται  $j = 1$  (όπου  $j$  ο μετρητής των περιορισμών).
3. Γίνεται εξέλιξη στο συγκεκριμένο πληθυσμό με στόχο την ελαχιστοποίηση της παραβίασης του  $j_{\text{οστου}}$  περιορισμού, έως ότου το ποσοστό του πληθυσμού που είναι εφικτό, για αυτό τον περιορισμό, ξεπεράσει ένα ποσοστό  $\Phi$ . Εδώ έχουμε:

$$fitness(\mathbf{X}) = M - \varphi_1(\mathbf{X}) \quad (2.15)$$

4. Με  $M$  να είναι ένας ικανοποιητικά μεγάλος θετικός αριθμός, ο οποίος προσαρμόζεται δυναμικά σε κάθε γενιά.
5. Μεταβάλλεται το  $j = j + 1$ .
6. Ο τωρινός πληθυσμός είναι το σημείο εκκίνησης για την επόμενη φάση της εξέλιξης, όπου ελαχιστοποιείται η παραβίαση του  $j_{\text{οστου}}$  περιορισμού, με:

$$fitness(\mathbf{X}) = M - \varphi_1(\mathbf{X}) \quad (2.16)$$

Σε αυτή τη φάση, τα άτομα τα οποία δεν ικανοποιούν το λιγότερο έναν περιορισμό, από τους προηγούμενους που έχουμε εξετάσει, αποβάλλεται από τον πληθυσμό. Το κριτήριο συνέχειας, στην επόμενη φάση, είναι πάλι το ποσοστό του πληθυσμού που είναι εφικτό, για αυτό τον περιορισμό, να ξεπεράσει ένα ποσοστό  $\Phi$ .

7. Εάν  $j < m$ , επαναλαμβάνονται τα δύο τελευταία βήματα, αλλιώς βελτιστοποιείται η  $f$  και αποβάλλονται οι μη-εφικτές λύσεις.

Η βασική ιδέα είναι η ικανοποίηση του κάθε περιορισμού, με τη σειρά, από ένα ποσοστό  $\Phi$  του πληθυσμού, και στην συνέχεια ο έλεγχος του επόμενου, με προσπάθεια να συνεχίσουν να ικανοποιούνται οι προηγούμενοι. Στο τελευταίο βήμα χρησιμοποιείται η μέθοδος της «θανατικής ποινής» (Schoenauer & Xanthakis, 1993).

Το βασικό πρόβλημα του αλγορίθμου αυτού είναι ότι οι περιορισμοί πρέπει να ορισθούν με συγκεκριμένη σειρά, η οποία επηρεάζει τη λειτουργία του, και από άποψη υπολογιστικού χρόνου, αλλά και ακρίβειας λύσης. Ένα ακόμα πρόβλημα είναι το μεγάλο υπολογιστικό «κόστος» λόγω της έλλειψης του κανόνα της «ελάχιστης δυνατής ποινής».

#### **2.3.2.4 Άλλες τεχνικές**

Οι παραπάνω τεχνικές που αναφέρθηκαν είναι κάποιες από τις πιο βασικές που εφαρμόζονται συχνά στους εξελικτικούς, σύμφωνα με τον Coello (Coello C. A., 1999). Κάποιες ακόμα που υπάρχουν είναι:

- Η μέθοδος των αυτό-προσαρμοζόμενων ποινών του Michalewicz (Michalewicz et al., 1996).
- Οι αλγόριθμοι «επισκευής» (Liepins & Vose, 1990).
- Οι «Multi-objective Optimization Techniques» (Fonseca & Fleming, 1995).

Άλλη ομάδα τεχνικών χειρισμού περιορισμών είναι οι «υβριδικές μέθοδοι», οι οποίες συνδυάζονται με άλλες τεχνικές. Κάποιες γνωστές είναι:

- Οι πολλαπλασιαστές Lagrangian (Adeli & Cheng, 1994).
- Βελτιστοποίηση προβλημάτων με «τυχαία εξέλιξη» (Belur, 1997).
- «Ασαφής Λογική» (Fuzzy logic) (Le, 1995).

Σε αυτή την κατηγορία υπάρχουν πάρα πολλές τεχνικές, λόγω της ευκολίας δημιουργίας τους, καθώς αποτελούν συνδυασμό (και παραλλαγές) όλων των παραπάνω.





## **ΚΕΦΑΛΑΙΟ 3**

### **ΜΑΘΗΜΑΤΙΚΑ ΜΟΝΤΕΛΑ SL-PSO ΚΑΙ**

### **εDEag**



### 3.1 Μέθοδος SL-PSO

#### 3.1.1 Μαθηματικά στοιχεία για την SL-PSO

Με τη χρήση της μεθόδου SL-PSO ερευνήσαμε προς βελτιστοποίηση 9 διαφορετικές συναρτήσεις από το διαγωνισμό CEC-2010 (Mallipeddi & Suganthan, 2010). Οι συναρτήσεις που εισάγονται στον κώδικα προς βελτιστοποίηση έχουν τη μορφή :

- Αντικειμενική Συνάρτηση, δηλαδή η συνάρτηση που έχουμε ως στόχο να ελαχιστοποιήσουμε :

$$f(\vec{x}), \text{ όπου } \vec{x} = (x_1, x_2, x_3, \dots, x_n) \in \mathbb{R}^n \quad (3.1)$$

Σε περίπτωση που έχουμε να μεγιστοποιήσουμε τη συνάρτηση  $f(\vec{x})$  τότε τη μετατρέπουμε στη μορφή :

$$\Omega(\vec{x}) = [-f(\vec{x})] \text{ γνωρίζοντας ότι } \text{Max } f(\vec{x}) = -\text{Min } \Omega(\vec{x}) \quad (3.2)$$

Και εισάγουμε στον κώδικα, ως προς ελαχιστοποίηση, την  $\Omega(\vec{x})$ .

- Περιορισμοί ανισότητας, δηλαδή συναρτήσεις που το αποτέλεσμά τους, όταν εισάγουμε το παραπάνω διάνυσμα  $\vec{x}$ , πρέπει να είναι μικρότερο από μία τιμή «A» :

$$(g_i(\vec{x}) - A_i) \leq 0, \text{ για κάθε } i \in \{1, \dots, m\} \text{ και κάθε } A \in \mathbb{R} \quad (3.3)$$

Σε περίπτωση που έχουμε τη μορφή  $(g_i(\vec{x}) - A_i) \geq 0$  τη μετατρέπουμε σε  $(A_i - g_i(\vec{x})) \leq 0$ .

- Περιορισμοί ισότητας, δηλαδή συναρτήσεις που το αποτέλεσμά τους, όταν εισάγουμε το παραπάνω διάνυσμα  $\vec{x}$ , πρέπει να είναι ίσο μία τιμή «B» :

$$(h_j(\vec{x}) - B_j) = 0, \text{ για κάθε } j \in \{1, \dots, k\} \text{ και κάθε } B \in \mathbb{R} \quad (3.4)$$

Τις μετατρέπουμε στη μορφή :

$$|h_j(\vec{x}) - B_j| - \varepsilon \leq 0, \text{ για } \varepsilon \in \mathbb{R} \quad (3.5)$$

Όπου  $\varepsilon$  είναι μπορεί να είναι ένας πολύ μικρός αριθμός παραβίασης του περιορισμού.

- Πεδίο τιμών για το  $\vec{x}$ :

$$y_u \leq x_u \leq t_u, \text{ για κάθε } u \in \{1, \dots, n\} \quad (3.6)$$

Όπου  $(y_1, y_2, y_3, \dots, y_n) \in \mathbb{R}^n$  είναι οι ελάχιστες τιμές των  $(x_1, x_2, x_3, \dots, x_n)$ , και  $(t_1, t_2, t_3, \dots, t_n) \in \mathbb{R}^n$  οι μέγιστες αντίστοιχα.

### 3.1.2 Μαθηματική μοντελοποίηση της SL-PSO

Σε αυτό το υποκεφάλαιο θα περιγράψουμε τον τρόπο με τον οποίο λειτουργεί η μέθοδος SL-PSO (Changhe & Shengxiang, 2012).

Αρχικά ορίζουμε μια πιθανή λύση ως σωματίδιο, όπως και σε κάθε μέθοδο PSO. Κάθε σωματίδιο  $i$  αντιπροσωπεύεται από ένα διάνυσμα θέσης και ένα διάνυσμα ταχύτητας, τα οποία ενημερώνονται σε κάθε επανάληψη από τον τύπο:

$$\begin{aligned} v'_i{}^d &= \omega v_i{}^d + \eta_1 r_1 (x_{pbest_i}{}^d - x_i{}^d) + \eta_2 r_2 (x_{gbest}{}^d - x_i{}^d) \\ x'_i{}^d &= x_i{}^d + v'_i{}^d \end{aligned} \quad (3.7)$$

Όπου  $x'_i{}^d$  και  $x_i{}^d$  αντιπροσωπεύουν την τωρινή και την προηγούμενη θέση στη  $d_{\text{οστη}}$  διάσταση του σωματιδίου  $i$ , αντίστοιχα  $v'_i{}^d$  και  $v_i{}^d$  αντιπροσωπεύουν την τωρινή και την προηγούμενη ταχύτητα του σωματιδίου  $i$ . Τα  $\vec{x}_{pbest_i}$  και  $\vec{x}_{gbest}$  αντιπροσωπεύουν την καλύτερη θέση που έχει βρει το σωματίδιο  $i$  και την καλύτερη θέση που έχει βρεθεί από όλο το σμήνος των σωματιδίων αντίστοιχα. Το  $\omega \in (0,1)$  εκφράζει ένα συντελεστή αδράνειας, σχετικά με το πόσο η ταχύτητα διατηρείται, τα  $\eta_1$  και  $\eta_2$  είναι σταθερές επιτάχυνσης, και τα  $r_1$  και  $r_2$  είναι τυχαίοι αριθμοί οι οποίοι παράγονται εσωτερικά του κώδικα με τιμές μεταξύ 0.0 και 1.0.

Στους αλγόριθμους PSO υπάρχουν δύο κύρια μοντέλα: αυτά που βασίζονται στο  $g_{best}$  (*global best*) και το κάθε  $i$  σωματίδιο μαθαίνει από όλο το σμήνος, και το  $l_{best}$  (*local best*) που το κάθε σωματίδιο  $i$  μαθαίνει από κάποια άλλα συγκεκριμένα σωματίδια που έχουν ορισθεί ως γειτονικά του. Σύμφωνα με τους Kennedy and Eberhart (Kennedy & Eberhart, 2001) οι αλγόριθμοι που βασίζονται στο  $g_{best}$  συγκλίνουν πιο γρήγορα αλλά έχουν κίνδυνο να παγιδευτούν σε τοπικά ακρότατα, σε αντίθεση με τους  $l_{best}$  που είναι λιγότερο ευάλωτοι σε τέτοιες παγιδεύσεις, αλλά είναι αρκετά πιο αργοί. Το 2009 προτάθηκε η μέθοδος A-PSO (Zhan et al., 2009), η

οποία ανέφερε τη χρήση τεσσάρων εξελικτικών καταστάσεων: εκμετάλλευσης, εξερεύνησης, σύγκλισης και αναπήδησης, και η οποία εφαρμόστηκε με τη μέθοδο AL-PSO(Li & S. Yang, 2009) με επιτυχία..

Ακριβώς λόγω το δύο παραπάνω μοντέλων ( $g_{best}, l_{best}$ ) και της χρησιμότητας που έδειξε η εφαρμογή των τεσσάρων εξελικτικών καταστάσεων, δημιουργήθηκε η ανάγκη μίας παραλλαγής της PSO, η SL-PSO, που θα μπορεί να ισορροπεί τη χρήση όλων των παραπάνω με στόχο την καλύτερη δυνατή απόδοση. Εμπνευσμένοι από την ιδέα του διαχωρισμού των εργασιών, μπορούμε να δώσουμε σε κάθε σωματίδιο έναν από τους παρακάτω τέσσερις ρόλους :

- Σύγκλιση στην καλύτερη θέση που έχει βρεθεί από το σμήνος. Αυτό είναι πολύ χρήσιμο για περιοχές με μεγάλες αποστάσεις μεταξύ των διάφορων ακροτάτων.
- Εκμετάλλευση της δικής του καλύτερης θέσης, όταν το σωματίδιο βρίσκεται κοντά σε ακρότατο και μπορεί να βρει εκεί καλύτερη λύση.
- Εξερεύνηση νέων υποσχόμενων περιοχών. Όταν έχουμε πολλά ακρότατα είναι καλύτερα το σωματίδιο να μαθαίνει από γειτονικά σωματίδια, καθώς αυτό προάγει την εξερεύνηση σε νέες περιοχές.
- Αναπήδηση από τοπικό ακρότατο. Όταν το σωματίδιο έχει παγιδευτεί σε κάποιο ακρότατο αναπηδά από αυτό σε νέες περιοχές, πιθανόν, καλύτερων λύσεων.

Με βάση την παραπάνω ανάλυση, ορίσαμε τις τέσσερις στρατηγικές, με την αντίστοιχη χρησιμότητά τους, που εκμεταλλεύεται η μέθοδος SL-PSO. Οι πηγές από τις οποίες μαθαίνει πληροφορίες η SL-PSO είναι :

- $a_{best} (g_{best})$  , που είναι η καταγεγραμμένη θέση του καλύτερου σωματιδίου του σμήνους.
- $p_{best}$  , που είναι η καλύτερη ατομική θέση του κάθε σωματιδίου.
- $p_{bestrand}$  , που είναι η θέση ενός τυχαίου σωματιδίου, καλύτερη από τη δική του.
- $p_{rand}$  , που είναι μία τυχαία θέση κοντά στη δική του.

Η κάθε στρατηγική επιτρέπει σε κάθε σωματίδιο να ανταποκρίνεται ανεξάρτητα από τα άλλα, σε διαφορετικές καταστάσεις που μπορεί να βρεθεί. Έτσι για κάθε σωματίδιο  $k$ , οι εξισώσεις, για κάθε στρατηγική, που μπορούν να εφαρμοστούν είναι:

1. Παράγοντας (a): εκμάθηση από  $p_{best}$

$$v'_k{}^d = \omega v_k{}^d + \eta \cdot r_k{}^d \cdot (pbest_k{}^d - x_k{}^d) \quad (3.8)$$

2. Παράγοντας (b): εκμάθηση από  $p_{rand}$

$$x'_k{}^d = x_k{}^d + v_{avg}{}^d \cdot N(0,1) \quad (3.9)$$

3. Παράγοντας (c): εκμάθηση από  $p_{best_{rand}}$

$$v'_k{}^d = \omega v_k{}^d + \eta \cdot r_k{}^d \cdot (pbest_{rand}{}^d - x_k{}^d) \quad (3.10)$$

4. Παράγοντας (b): εκμάθηση από  $a_{best}$

$$v'_k{}^d = \omega v_k{}^d + \eta \cdot r_k{}^d \cdot (abest_k{}^d - x_k{}^d) \quad (3.11)$$

Το  $v_{avg}{}^d$  είναι η μέση ταχύτητα όλων των σωματιδίων στη  $d_{οστη}$  διάσταση, με τύπο  $\sum_{k=1}^N |v_k{}^d| / N$  όπου  $N$  το μέγεθος του σμήνους.  $N(0,1)$  είναι ένας τυχαίος αριθμός από την κανονική κατανομή με μέση τιμή 0 και τυπική απόκλιση 1. Όλοι αυτοί οι παράγοντες που αναλύσαμε χρησιμοποιούνται για την ανανέωση των τιμών θέσης και ταχύτητας του κάθε σωματιδίου από τον Αλγόριθμο 1 :

---

**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 (3.7);
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 (3.7);
9:   else
10:    Update the velocity and position of particle  $j$  using
   operator  $c$  and (3.7);
11:     $k := j$ ;
12:   end if
13: else
14:   Update the velocity and position of particle  $k$  using
   operator  $d$  and (3.7);
15: end if
16:  $fes++$ ; where  $fes$  is the current number of fitness
   evaluations.

```

---

**Σχήμα 3.1** Αλγόριθμος 1 της SL-PSO

Εδώ πρέπει να σημειώσουμε ότι το  $a_{best}$  δεν είναι σωματίδιο ακριβώς, αλλά είναι η καλύτερη «καταγεγραμμένη» θέση που έχει βρεθεί μέχρι στιγμής από το σωματίδιο  $g_{best}$ . Συνεπώς έχει ίση ή καλύτερη τιμή από το  $g_{best}$  ανά πάσα στιγμή κατά τη διαδικασία λειτουργίας της εξελικτικής διαδικασίας. Το  $a_{best}$  από το οποίο μαθαίνουν όλα τα σωματίδια, ακόμα και το  $g_{best}$  ανανεώνεται με βάση τον Αλγόριθμο 2 :

---

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

---

```

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

```

---

**Σχήμα 3.2** Αλγόριθμος 2 της SL-PSO

Η επιλογή του παράγοντα που είναι πιο κατάλληλος για ένα σωματίδιο εξαρτάται από την κοντινή τοπολογία του κάθε ενός. Επειδή δεν μπορούμε να ξέρουμε τη γενική τοπολογία, πρέπει κάθε σωματίδιο να μπορεί να αναγνωρίζει την κοντινή του τοπολογία. Στη συνέχεια θα αναλύσουμε με ποιο τρόπο επιλέγεται η χρήση του κάθε παράγοντα, καθώς και με ποιο τρόπο θα προσαρμόζεται αυτός στην αλλαγή των θέσεων των σωματιδίων κατά την εξελικτική διαδικασία.

Στην αρχή κάθε παράγοντα έχει αναλογία επιλογής  $\frac{1}{4}$  και ανανεώνεται με βάση τη σχετική του απόδοση. Για κάθε σωματίδιο επιλέγεται ένας παράγοντας ανάλογα με την αναλογία επιλογής που έχει εκείνη τη στιγμή. Ο παράγοντας που θα έχει τη μεγαλύτερη σχετική απόδοση, με βάση τη βελτίωση της αντικειμενικής, την τρέχουσα αναλογία επιτυχίας και τις προηγούμενες αναλογίες επιλογής, θα έχει μία αύξηση στην τρέχουσα «αναλογία» επιλογής του. Σταδιακά, ο καταλληλότερος παράγοντας θα επιλέγεται αυτόματα και θα ελέγχει τη διαδικασία εκμάθησης του κάθε σωματιδίου σε κάθε διαφορετικής εξελικτική διαδικασία και τοπολογία. Για κάθε παράγοντα  $i$  σε κάθε ανανέωση ενός σωματιδίου  $k$  η επιβράβευση δίνεται με βάση την παρακάτω λογική:

Για κάθε παράγοντα  $i$  του σωματιδίου  $k$  στην επανάληψη  $t$  υπολογίζεται ένας βαθμός προόδου:

$$p_i^k(t) = \begin{cases} |f(\vec{x}_k(t)) - f(\vec{x}_k(t-1))|, & \text{εάν ο παράγοντας } i \text{ επιλέχθηκε} \\ 0, & \text{αλλιώς} \end{cases} \quad (3.12)$$

Στη συνέχεια, με βάση τα στοιχεία που είπαμε παραπάνω για την καλύτερη σχετική απόδοση, υπολογίζεται ο βαθμός επιβράβευσης :

$$r_i^k(t) = \frac{p_i^k(t)}{\sum_{j=1}^R p_j^k(t)} + \frac{g_i^k}{G_i^k} (1 - a) + c_i^k s_i^k(t) \quad (3.13)$$

Όπου  $g_i^k$  είναι ένας μετρητής που καταγράφει τον αριθμό των επιτυχημένων ενημερώσεων, με βάση το εάν ο απόγονός του έχει καλύτερη τιμή από αυτόν, του σωματιδίου  $k$  με τη χρήση του παράγοντα  $i$  από την τελευταία ενημέρωση της τιμής της αναλογίας επιλογής.  $G_i^k$  είναι ο αριθμός των επαναλήψεων που ο παράγοντας  $i$  επιλέχθηκε από το σωματίδιο  $k$  από την τελευταία ενημέρωση της τιμής της αναλογίας επιλογής. Έτσι το κλάσμα  $g_i^k/G_i^k$  εκφράζει την αναλογία επιτυχίας του παράγοντα  $i$  για το σωματίδιο  $k$ . Το  $a$  είναι ένα τυχαίο «βάρος» με τιμή μεταξύ 0.0 και 1.0,  $R$  είναι ο αριθμός των παραγόντων που χρησιμοποιούνται και  $c_i^k$  είναι ένας συντελεστής ποινής του παράγοντα  $i$  για το σωματίδιο  $k$  με τιμή :

$$c_i^k = \begin{cases} 0.9, & \text{εάν } g_i^k = 0 \text{ και } s_i^k(t) = \max_{j=1}^R (s_j^k(t)) \\ 1, & \text{αλλιώς} \end{cases} \quad (3.14)$$

Το  $s_i^k$  εκφράζει την αναλογία επιλογής του παράγοντα  $i$  από το σωματίδιο  $k$  για την τρέχουσα επανάληψη. Σε περίπτωση που κανένας παράγοντας δεν βελτίωσε την τιμή του σωματιδίου  $k$  από την τελευταία ενημέρωση της τιμής της αναλογίας επιλογής, τότε  $\sum_{j=1}^R p_j^k(t) = 0$ , οπότε θα υπολογιστεί μόνο το  $c_i^k s_i^k(t)$  για το βαθμό επιβράβευσης.

Έτσι, με βάση όλα τα παραπάνω, η τιμή της αναλογίας επιλογής του παράγοντα  $i$  για το σωματίδιο  $k$  στην επόμενη επανάληψη  $t + 1$  ενημερώνεται ακολούθως :

$$s_i^k(t+1) = \frac{r_i^k(t)}{\sum_{j=1}^R r_j^k(t)} (1 - R * \gamma) + \gamma \quad (3.15)$$



Όπου  $\gamma$  είναι η ελάχιστη τιμή της αναλογίας επιλογής και για τις δοκιμές που πραγματοποιήσαμε πήρε την τιμή 0.01. Με βάση όλη την παραπάνω διαδικασία που αναφέραμε είναι προφανές ότι πάντα ένας παράγοντας θα έχει τη μεγαλύτερη αναλογία επιλογής, ο οποίος είναι και ο πιο επιτυχημένος για την τρέχουσα επανάληψη. Παρόλα αυτά, όταν ένα σωματίδιο συγκλίνει ή μετακινείται σε μία τοπική υποπεριοχή τότε αυτός ο παράγοντας μπορεί να μη φέρνει πια τα προηγούμενα οφέλη στο αντίστοιχο σωματίδιο. Αυτό αντιμετωπίζεται με τον συντελεστή ποινής που αναφέραμε ( $c_i^k$ ) ο οποίος θα μειώσει την αναλογία επιτυχίας αυτού του παράγοντα και θα αυξήσει των άλλων. Έτσι σταδιακά, με όλη την παραπάνω διαδικασία, ένας πιο κατάλληλος παράγοντας με καλύτερη απόδοση θα επιλεγεί και ο προηγούμενος παράγοντας θα χάσει τη θέση ισχύος που κατείχε. Αυτός είναι ο μηχανισμός προσαρμοστικότητας που εκμεταλλεύεται η μέθοδος SL-PSO. Επίσης πρέπει να αναφέρουμε ότι οι αναλογίες επιλογής για όλους τους παράγοντες ενημερώνονται την ίδια στιγμή και όχι σε κάθε επανάληψη, δηλαδή έχουμε ένα μετρητή  $m_k$  ο οποίος μετράει τις συνεχόμενες αποτυχημένες προσπάθειες εκμάθησης για κάθε σωματίδιο  $k$ . Εάν το σωματίδιο  $k$  βελτιωθεί από ένα παράγοντα πριν ο  $m_k$  φτάσει σε ένα μέγιστο αριθμό συνεχόμενων επαναλήψεων  $U_f^k$  τότε αυτός μηδενίζεται. Με αυτή τη μέθοδο διασφαλίζεται ότι δεν θα τιμωρηθεί ο καλύτερος παράγοντας επειδή ήταν αποτυχημένος για λίγες επαναλήψεις. Μετά την ενημέρωση της αναλογίας επιλογής όλων των παραγόντων, όλες οι πληροφορίες, εκτός από τις αναλογίες επιλογής, διαγράφονται.

Σε όλους τους σύγχρονους αλγόριθμους συμβαίνει αυτή η διαγραφή, καθώς οι νέες πληροφορίες αντικαθιστούν τις παλιές, με αποτέλεσμα πιθανόν χρήσιμες πληροφορίες να χαθούν. Πρακτικά αυτό σημαίνει ότι εάν σε ένα σωματίδιο μία διάσταση βελτιώνεται αλλά οι άλλες είναι κακές και επηρεάζουν συνολικά το σωματίδιο, τότε αυτή η υποσχόμενη πληροφορία μπορεί να χαθεί. Επίσης μπορεί να συμβαίνει και το αντίθετο, δηλαδή κάποιες διαστάσεις να επηρεάζουν πιο πολύ το σωματίδιο από κάποιες άλλες και έτσι εάν αυτές βελτιώνονται αλλά οι άλλες χειροτερεύουν, προσωρινά, τότε μπορεί να χαθούν υποσχόμενες προηγούμενες πληροφορίες. Για αυτό δημιουργήσαμε τη θέση  $a_{best}$  η οποία μαθαίνει από διαστάσεις διαφόρων σωματιδίων που έχουν δείξει σημάδια βελτίωσης μέσα στο χρόνο. Επειδή η προσαρμογή αυτής της ιδέας στην SL-PSO είναι κάτι αρκετά πολύπλοκο καθώς μπορεί να εμφανίσει προβλήματα από τη βελτίωση συγκεκριμένων

διαστάσεων του  $a_{best}$ , εισαγάγαμε μία πιθανότητα εκμάθησης  $P_l$  για κάθε διάσταση. Με αυτό τον τρόπο έχουμε δύο σημαντικά οφέλη: 1) ο αλγόριθμος εξοικονομεί υπολογιστική ισχύ και 2) μειώνεται η πιθανότητα το  $a_{best}$  να μάθει άχρηστες πληροφορίες, ακόμα και εάν έτσι μειώνεται και η πιθανότητα να μάθει κάποιες χρήσιμες. Η εκμάθηση των σωματιδίων από τη θέση  $a_{best}$  προσδίδει μία επιτάχυνση στη σύγκλιση του σμήνους, ειδικά στα σωματίδια που είναι κοντά σε αυτή τη θέση η έλξη είναι τόσο ισχυρή που δεν αφήνει κανένα άλλο παράγοντα να επηρεάσει. Για να εξισορροπήσουμε τη χρήση αυτού του παράγοντα, με στόχο την πλήρη χρήση του μηχανισμού προσαρμογής, πρέπει να ελέγξουμε τον αριθμό των σωματιδίων που μαθαίνουν από αυτή τη θέση. Υπάρχουν δύο λόγοι που το κάνουμε αυτό :

- Πρώτον, ο βέλτιστος αριθμός σωματιδίων που μαθαίνουν από τη θέση  $a_{best}$  εξαρτάται από τις ιδιότητες του προβλήματος που έχουμε να επιλύσουμε. Για παράδειγμα σε ένα ομοιόμορφο πρόβλημα της μορφής σφαίρας είναι καλύτερα να μαθαίνουν όλα τα σωματίδια από αυτή τη θέση, καθώς θέλουμε να έχουμε γρηγορότερη σύγκλιση στη βέλτιστη λύση. Αντίθετα, σε πιο πολύπλοκα, από άποψη τοπολογίας, προβλήματα είναι καλύτερα τα σωματίδια να πραγματοποιούν αναζήτηση στις τοπικές τους τοπολογίες, έτσι ώστε να έχουν καλύτερες πιθανότητες να βρουν τη βέλτιστη λύση.
- Δεύτερον, δεν είναι δίκαιο οι άλλοι τρεις παράγοντες να ανταγωνίζονται με τον παράγοντα της σύγκλισης καθώς συνεισφέρουν και αυτοί στη βελτίωσή του, είτε άμεσα με το βελτιωμένο σωματίδιο να γίνεται η νέα  $a_{best}$  θέση, είτε έμμεσα με τις πληροφορίες που εισάγονται στη θέση  $a_{best}$  από τα βελτιωμένα σωματίδια.

Με βάση την προηγούμενη ανάλυση είναι καλό να ελέγξουμε τον αριθμό των σωματιδίων που μαθαίνουν από την  $a_{best}$ . Επειδή είναι πολύ δύσκολο να αναγνωρίσουμε ποια σωματίδια είναι κατάλληλα για να μάθουν από αυτή τη θέση, επιλέγουμε κάποια τυχαία σε κάθε επανάληψη, τα οποία θα χρησιμοποιήσουν τον παράγοντα της σύγκλισης. Για να προσαρμόσουμε αυτή την ιδέα πρέπει σε κάθε σωματίδιο να δώσουμε μία πληροφορία η οποία θα μας δείχνει εάν ανάμεσα σε δύο διαδοχικές επαναλήψεις αυτό χρησιμοποίησε ή όχι τον παράγοντα σύγκλισης. Επίσης πρέπει να ενημερώσουμε το βαθμό προόδου, βαθμό επιβράβευσης, την αναλογία

επιτυχίας και την αναλογία επιλογής όλων των παραγόντων. Έτσι έχουμε δύο πιθανές καταστάσεις :

- Στην πρώτη επανάληψη να χρησιμοποιεί τον παράγοντα σύγκλιση και στη δεύτερη όχι. Τότε πρέπει να αφαιρέσουμε τη θέση  $a_{best}$  από τις πηγές εκμάθησης του σωματιδίου και να κάνουμε κανονικοποίηση στους άλλους τρεις παράγοντες με βάση τη τρέχουσα αναλογία επιλογής, αφήνοντας τις άλλες πληροφορίες ίδιες.
- Αντίθετα, στην πρώτη επανάληψη να μη χρησιμοποιεί τον παράγοντα σύγκλισης και στη δεύτερη να τον χρησιμοποιεί. Σε αυτή την περίπτωση όλες οι πληροφορίες σχετικά με τους παράγοντες επαναφέρονται στην αρχική κατάστασή τους, και οι αναλογίες επιτυχίας επαναφέρονται στο  $1/4$ .

Αυτή η διαδικασία περιγράφεται στον Αλγόριθμο 3 :

---

**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 3.3** Αλγόριθμος 3 της SL-PSO

Όπου το  $CF_k$  εκφράζει το εάν το σωματίδιο  $k$  χρησιμοποιεί ή όχι τον παράγοντα σύγκλισης στην τρέχουσα επανάληψη, και το  $PF_k$  εκφράζει το εάν το σωματίδιο  $k$  χρησιμοποίησε ή όχι τον παράγοντα σύγκλισης στην προηγούμενη επανάληψη.

Στην SL-PSO έχουμε ορίσει και μία  $V_{max}$  με την οποία καταφέρνουμε να περιορίσουμε τη μέγιστη ταχύτητα του κάθε σωματιδίου στο μισό της περιοχής αναζήτησής του, για κάθε γενικό πρόβλημα. Επίσης έχουμε ορίσει και μία διαδικασία

με την οποία ελέγχουμε τις εκτός ορίων αναζητήσεις: Για κάθε διάσταση, πριν ενημερώσουμε τη θέση της, αρχικά με βάση την προηγούμενη θέση  $x^d(t-1)$  υπολογίζουμε μια νέα προσωρινή θέση  $x_t$  με βάση τον Αλγόριθμο 1, και μετά ενημερώνουμε την τρέχουσα θέση  $x^d(t)$  ακολούθως :

$$x^d(t) = \begin{cases} R(X_{min}^d, x^d(t-1)), & \text{εάν } x_t < X_{min}^d \\ R(x^d(t-1), X_{max}^d), & \text{εάν } x_t > X_{max}^d \\ x_t, & \text{αλλιώς} \end{cases} \quad (3.16)$$

Όπου  $R(a, b)$  μας επιστρέφει έναν ομοιόμορφα κατανομημένο αριθμό μεταξύ του  $[a, b]$ , και  $[X_{min}^d, X_{max}^d]$  είναι τα όρια αναζήτησης της  $d_{οστης}$  διάστασης του δοσμένου προβλήματος.

Ακόμα, στην SL-PSO, έχουμε και τέσσερις βασικές παραμέτρους τις οποίες αναφέραμε πιο πάνω, χωρίς όμως να αναλύσουμε τον τρόπο υπολογισμού τους. Αυτές είναι οι :

- Η συχνότητα αναβάθμισης  $U_f^k$  για κάθε σωματίδιο  $k$ . Πρόκειται για το μέγιστο αριθμό συνεχόμενων επαναλήψεων πριν γίνει ενημέρωση στις αναλογίες των παραγόντων, και εκφράζεται από τον τύπο :

$$U_f^k = \max[10 * \exp(-(1,6 \cdot k/N)^4), 1] \quad (3.17)$$

Το  $N$  εκφράζει το μέγεθος του πληθυσμού. Έτσι το  $U_f^k$  παίρνει τιμές από 1 έως 10 για κάθε σωματίδιο, και με αυτό τον τύπο κάποια σωματίδια μπορούν να πιάσουν το βέλτιστο αριθμό του  $U_f$ , για διαφορετικά προβλήματα.

- Η πιθανότητα εκμάθησης  $P_l^k$  για κάθε σωματίδιο  $k$ . Ουσιαστικά χρησιμοποιήθηκε ίδιας μορφής τύπος με τη συχνότητα αναβάθμισης, με το εύρος των πιθανών τιμών να κυμαίνεται από 0.05 έως 1.00.

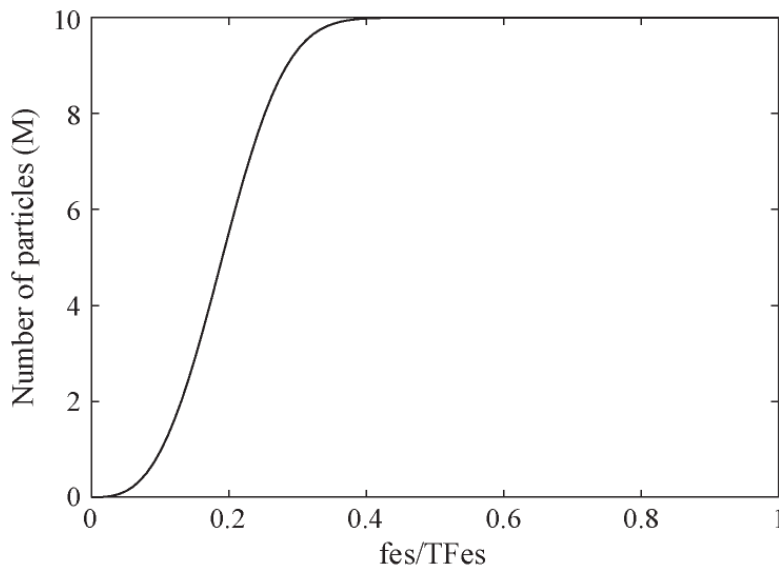
$$P_l^k = \max[1 - \exp(-(1,6 \cdot k/N)^4), 0.05] \quad (3.18)$$

- Με στόχο τη μείωση του ρίσκου να χρησιμοποιηθούν ακατάλληλες τιμές για τα  $U_f$  και  $P_l$  για ένα συγκεκριμένο σωματίδιο, δημιουργούμε μία μεταλλαγή ενός πίνακα αριθμών για τα σωματίδια σε κάθε επανάληψη και στη συνέχεια

ενημερώνουμε τις τιμές των  $U_f$  και  $P_l$  για κάθε ένα από αυτά. Έτσι για να ξέρουμε πόσα σωματίδια πρέπει να χρησιμοποιούν τον παράγοντα σύγκλισης χρησιμοποιούμε τον παρακάτω τύπο:

$$M(fes) = N \cdot [1 - \exp(-100(fes/T_{Fes})^3)] \quad (3.19)$$

Όπου  $fes$  είναι το πλήθος των αξιολογήσεων της αντικειμενικής συνάρτησης που έχει κάνει ο αλγόριθμος μέχρι στιγμής, και  $T_{Fes}$  το μέγιστο επιτρεπόμενο όριο του  $fes$  για την ολοκλήρωση όλου του αλγορίθμου. Η αναλογία του  $M$ , συναρτήσει του  $fes/T_{Fes}$  κυμαίνεται κάπως έτσι :



**Σχήμα 3.4** Αναλογία  $M$  συναρτήσει  $fes/T_{Fes}$

Με αυτή τη σχέση βλέπουμε ότι στην αρχή κανένα σωματίδιο δεν μαθαίνει από την  $a_{best}$  με στόχο να εστιάσουν στην κοντινή περιοχή του κάθε ενός, και σταδιακά αυτό αυξάνεται έως ότου  $fes/T_{Fes} = 40\%$  και συνεπώς το  $M$  θα αγγίξει τη μέγιστη τιμή 10. Ο λόγος που χρησιμοποιούμε αυτή τη μέθοδο για να ορίσουμε το μέγιστο αριθμό  $M$  των σωματιδίων που χρησιμοποιούν τον παράγοντα σύγκλισης, και όχι έναν τύπο συναρτήσει της αναλογίας επιλογής του, είναι ότι θα χρειαζόταν να δημιουργήσουμε και μία ακόμα παράμετρο που θα έπρεπε να εκφράζει τη μέγιστη αναλογία συμμετοχής του παράγοντα αυτού. Επίσης η βέλτιστη αναλογία συμμετοχής που θα έπρεπε να ορίσουμε για κάθε πρόβλημα είναι πιθανότατα άγνωστη και διαφορετική.

- Η παράμετρος της αδράνειας  $\omega$  που έχουμε αναφέρει στην αρχή του κεφαλαίου παίρνει τιμές από 0.9 έως 0.4, με γραμμική μείωσή της, και εκφράζεται ως:

$$\omega(fes) = 0.9 - 0.5 * fes/T\_Fes \quad (3.20)$$

Αυτές οι τέσσερις παράμετροι ανανεώνονται με τον παρακάτω Αλγόριθμο 5 :

---

**Algorithm 5** UpdatePar()

---

- 1: Create a permutation of index number;
  - 2: Update  $U_f$  for each particle by (3.17);
  - 3: Update  $P_l$  for each particle by (3.18);
  - 4: Calculate the number of particles using the convergence operator by (3.19);
  - 5: Update related information of the four operators for each particle by Algorithm 3;
  - 6: Calculate the inertia weight  $\omega$  by (3.20);
- 

**Σχήμα 3.5** Αλγόριθμος 5 της SL-PSO

Και όλη η διαδικασία που περιγράψαμε, μέχρι στιγμής, σε αυτό το κεφάλαιο συνοψίζεται στον Αλγόριθμο 4 της SL-PSO :

---

**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:     Update( $i, k, fes$ );
  - 7:      $G_i^k ++$ ;
  - 8:     **if**  $f(\vec{x}_k(t)) < f(\vec{x}_k(t-1))$  **then**
  - 9:        $g_i^k ++$ ;  $m_k := 0$ ;
  - 10:        $p_i^k += f(\vec{x}_k(t-1)) - f(\vec{x}_k(t))$ ;
  - 11:       Perform UpdateAbest( $k, fes$ ) for the *abest* position;
  - 12:     **else**
  - 13:        $m_k := m_k + 1$ ;
  - 14:     **end if**
  - 15:     **if**  $f(\vec{x}_k(t)) < f(\vec{x}_{pbest_k})$  **then**
  - 16:        $\vec{x}_{pbest_k} := \vec{x}_k$ ;
  - 17:       **if**  $f(\vec{x}_k) < f(\vec{x}_{abest})$  **then**
  - 18:          $\vec{x}_{abest} := \vec{x}_k$ ;
  - 19:       **end if**
  - 20:     **end if**
  - 21:     **if**  $m_k \geq U_f^k$  **then**
  - 22:       Update the selection ratios according to Eq.(3.15);
  - 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:   UpdatePar();
  - 29:    $t ++$ ;
  - 30: **end while**
- 

**Σχήμα 3.6** Αλγόριθμος 4 της SL-PSO

### 3.1.3 Περιορισμοί της αντικειμενικής

Για την εφαρμογή των περιορισμών στον κώδικα της μεθόδου SL-PSO, στην παρούσα εργασία, χρησιμοποιήσαμε την εισαγωγή ενός συντελεστή ποινής με τεχνητό βάρος  $\lambda$  σε μία νέα αντικειμενική της μορφής:

$$F(\vec{x}) = f(\vec{x}) + \lambda * \left[ \sum_i^m \max(g_i(\vec{x}), 0) + \sum_j^k \max(|h_j(\vec{x})| - \varepsilon, 0) \right], \lambda \in \mathbb{R} \quad (3.21)$$

Σε αυτή θεωρούμε ότι η  $f(\vec{x})$  έχει έρθει στη μορφή για εύρεση ελάχιστης τιμής, τα  $A_i$  και  $B_j$  έχουν προσαρμοστεί εσωτερικά των  $g_i(\vec{x})$  και  $h_j(\vec{x})$  αντίστοιχα, και ότι έχουν έρθει όλοι οι περιορισμοί στη μορφή « $\leq 0$ » ([από υποκεφάλαιο 3.1.1](#)). Ουσιαστικά ο αλγόριθμος της SL-PSO χρησιμοποιεί αυτή τη νέα αντικειμενική συνάρτηση, χωρίς να χρειάζονται κάποιες άλλες σημαντικές αλλαγές στη γενική του μορφή.

Ο συντελεστής ποινής  $\lambda$  αποτελεί μία ακόμα παράμετρο του αλγόριθμού μας, η οποία είναι διαφορετική για κάθε πρόβλημα, και σε αντίθεση με τις παραπάνω που αναφέρθηκαν δεν υπολογίζεται με κάποιο συγκεκριμένο τύπο, τουλάχιστον στην εργασία αυτή. Στα προβλήματα που εξετάσαμε το  $\lambda$  αποτελούσε κομμάτι των προς έλεγχο αυθαίρετων παραμέτρων που ερευνήσαμε με στόχο να βρεθεί η βέλτιστη τιμή του, καθώς και η βέλτιστη λύση για κάθε πρόβλημα.

### 3.1.4 Νέα στοιχεία στον κώδικα SL-PSO που εξετάσαμε

Στην παρούσα εργασία εξετάσαμε μία παραλλαγή της SL-PSO που παρουσιάσαμε παραπάνω, επαυξημένη με κάποιες νέες μεθόδους για βελτίωση της όλης διαδικασίας, πράγμα που είναι και το κομμάτι που είχαμε και ως έρευνα. Τα πιο σημαντικά από αυτά είναι :

- Η εισαγωγή ενός στοχαστικού νέφους γύρω από τη θέση του γενικού βέλτιστου ( $a_{best}$ ), που έχει βρεθεί, με στόχο την εύρεση πιθανόν βελτιωμένης λύσης από την ήδη υπάρχουσα. Για τη χρήση αυτής της διεργασίας του αλγορίθμου εισάγουμε τρεις νέες αυθαίρετες παραμέτρους που έχουν ως στόχο τη μεταλλαγή αυτού του νέφους ανάλογα με το κάθε πρόβλημα για την καλύτερη αξιοποίησή του. Η πρώτη είναι ο αριθμός των σωματιδίων που έχουμε στο νέφος, η δεύτερη είναι η αρχική παράμετρος τυπικής απόκλισης

των σωματιδίων στο νέφος αυτό, και η τρίτη είναι ο ρυθμός μείωσης αυτής της τυπικής απόκλισης. Με αυτό τον τρόπο μπορούμε να ορίσουμε τα όρια έρευνας του νέφους κοντά στην καλύτερη θέση και να διασφαλίσουμε τη σύγκλιση του με την πάροδο των επαναλήψεων. Ουσιαστικά πρόκειται για μία λεπτομερή ανάλυση της τοπολογίας κοντά στη βέλτιστη θέση, που μας διασφαλίζει, έως ένα βαθμό, ότι η λύση που βρήκαμε δεν είναι κάποιο τοπικό ακρότατο και ότι είναι η καλύτερη που θα μπορούσαμε να βρούμε σε εκείνη την τοπολογία

- Η εισαγωγή ενός μετρητή ο οποίος όταν φτάσει σε ένα συγκεκριμένο νούμερο επαναλήψεων κάνει έναν ανασχηματισμό του σμήνους των σωματιδίων κρατώντας τα καλύτερα σωματίδια των προηγούμενων επαναλήψεων, πετυχαίνοντας έτσι και να κρατάει το επίπεδο σύγκλισης που έχει μέχρι στιγμής, αλλά και να αποφεύγει την παγίδευση σε τοπικά ακρότατα.

### 3.1.5 Μοντελοποίηση του αλγορίθμου σε λογισμικό MATLAB®

Σε αυτό το κομμάτι της εργασίας θα παρουσιάσουμε συνοπτικά τον τρόπο με τον οποίο προσαρμόστηκαν όλα τα παραπάνω στον κώδικα της MATLAB® SL-PSO που εργαστήκαμε, καθώς και ποιες μεταβλητές ερευνήσαμε προς βελτιστοποίηση στα προβλήματα που είχαμε για έρευνα.

Αρχικά γνωρίζουμε τα στοιχεία των προβλημάτων που έχουμε :

- Την αντικειμενική συνάρτηση.
- Τους περιορισμούς στη μορφή « $\leq 0$ ».
- Τα όρια των διαστάσεων της μεταβλητής.

Στη συνέχεια, οι μεταβλητές που επεμβαίνουμε αυθαίρετα και μελετάμε σε αυτή την εργασία είναι:

- Ο αριθμός των σωματιδίων στο σμήνος.
- Ο αριθμός των διαστάσεων της μεταβλητής.
- Ο αριθμός των επαναλήψεων έως την ολοκλήρωση του αλγορίθμου SL-PSO.
- Ο συντελεστής  $\lambda$ , που εκφράζει το τεχνητό βάρος των περιορισμών.



- Ο αριθμός των σωματιδίων στο νέφος.
- Η τυπική απόκλιση των σωματιδίων του νέφους.
- Ο ρυθμός μείωσης της τυπικής απόκλισης του νέφους.

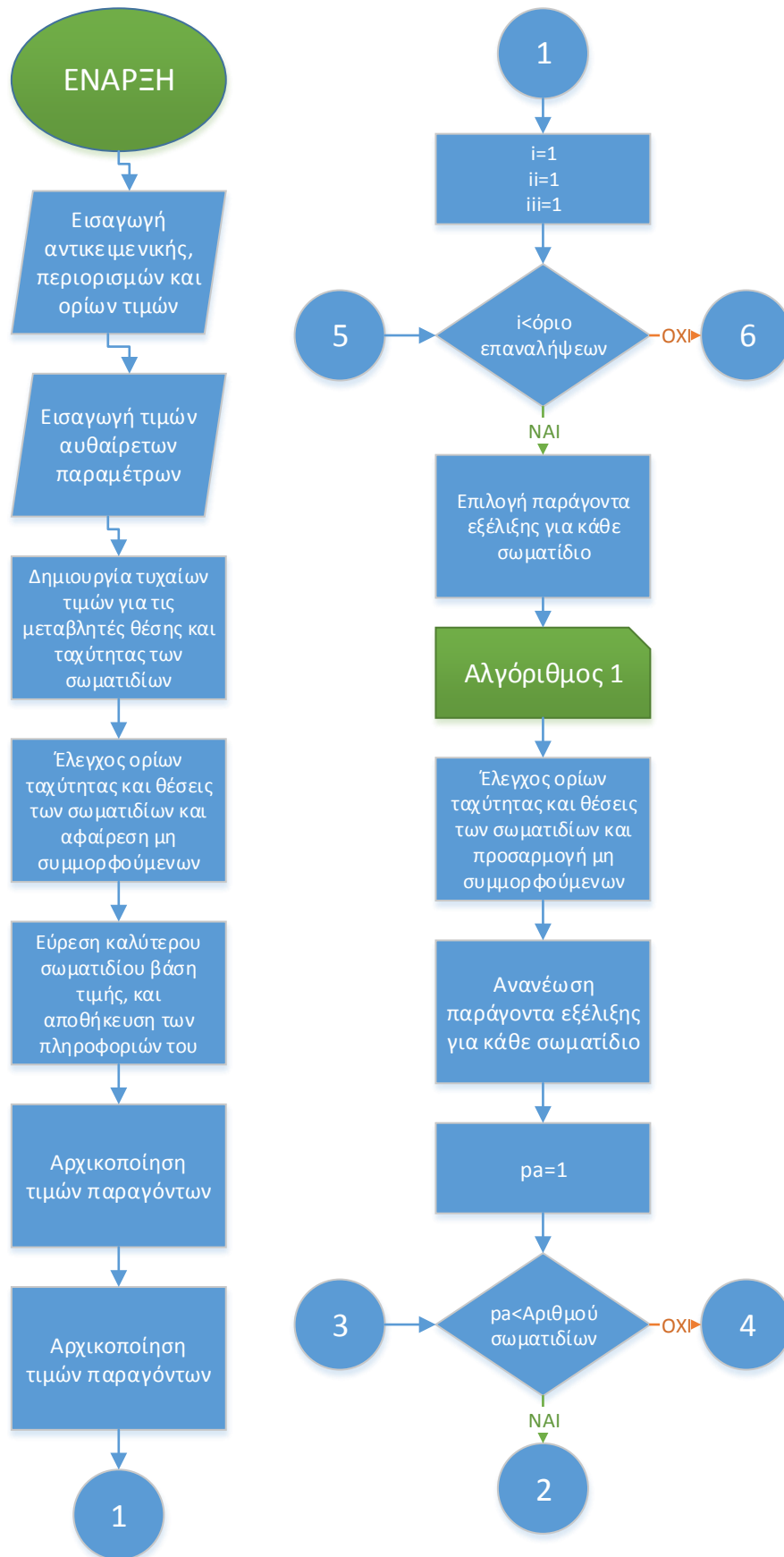
Αυτά είναι τα κύρια στοιχεία που επηρεάζουν τον κώδικα της MATLAB® SL-PSO από πρόβλημα σε πρόβλημα.

Αφού εισάγουμε όλα τα παραπάνω στον κώδικα, ξεκινάει η διαδικασία βελτιστοποίησης. Πρώτα, βρίσκουμε κάποιες αρχικές τιμές ξεκινώντας από τυχαίες τιμές για τη θέση και την ταχύτητα των σωματιδίων, και αφού ελέγξουμε το καθένα από αυτά με τους περιορισμούς και τα πεδία τιμών, ταξινομούμε αυτά που καλύπτουν τις προϋποθέσεις βάση της αντικειμενικής τιμής του κάθε ενός και διαλέγουμε το καλύτερο. Από αυτό κρατάμε όλες τις πληροφορίες του, δηλαδή τον αριθμό του, τη θέση του, την τιμή που βρήκε και τις τιμές των περιορισμών που έχει. Στη συνέχεια φορτώνονται οι αρχικές πληροφορίες όπως οι αναλογίες επιλογής των παραγόντων και ορίζονται 0 οι παράγοντες που πρέπει, όπως οι αναλογίες επιτυχίας, έτσι ώστε να ξεκινήσει η επαναληπτική διαδικασία. Έχοντας όλες τις παραπάνω τιμές ξεκινάμε τον Αλγόριθμο 1, μετά επανελέγχουμε τους περιορισμούς και τα όρια και προχωράμε στον Αλγόριθμο 2. Από αυτό βρίσκουμε τις τιμές των σωματιδίων και κρατάμε την καλύτερη του κάθε ενός και προχωράμε στον Αλγόριθμο 3. Μετά τον 3 τρέχουμε τον Αλγόριθμο 4, όπου εκεί ανανεώνουμε τις τιμές των παραμέτρων που αρχικοποιήσαμε. Εδώ ξεκινάει το κομμάτι των δύο αλλαγών που κάναμε από την κλασσική SL-PSO, δηλαδή ορίζουμε την περιοχή του νέφους συναρτήσει της βέλτιστης θέσης που έχει βρεθεί μέχρι στιγμής και της αρχικής τυπικής απόκλισης που ορίσαμε, και αρχίζουμε να μετράμε τον αριθμό των επαναλήψεων. Στις 100 επαναλήψεις κάνουμε ανακατασκευή του σμήνους κρατώντας τα καλύτερα σωματίδια και στις 1000 κάνουμε μείωση της τυπικής απόκλισης, με βάση τον αριθμό της μείωσης που ορίσαμε, για το νέφος. Τέλος, αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου ο μετρητής των επαναλήψεων φτάσει σε μια επιθυμητή τιμή που έχουμε ορίσει αυθαίρετα, η οποία όμως είναι αρκετά μεγάλη έτσι ώστε να συγκλίνει σε ικανοποιητικό επίπεδο κοντά στη βέλτιστη λύση, αλλά ταυτόχρονα να ολοκληρώνεται η όλη διαδικασία στο λιγότερο δυνατό χρόνο.

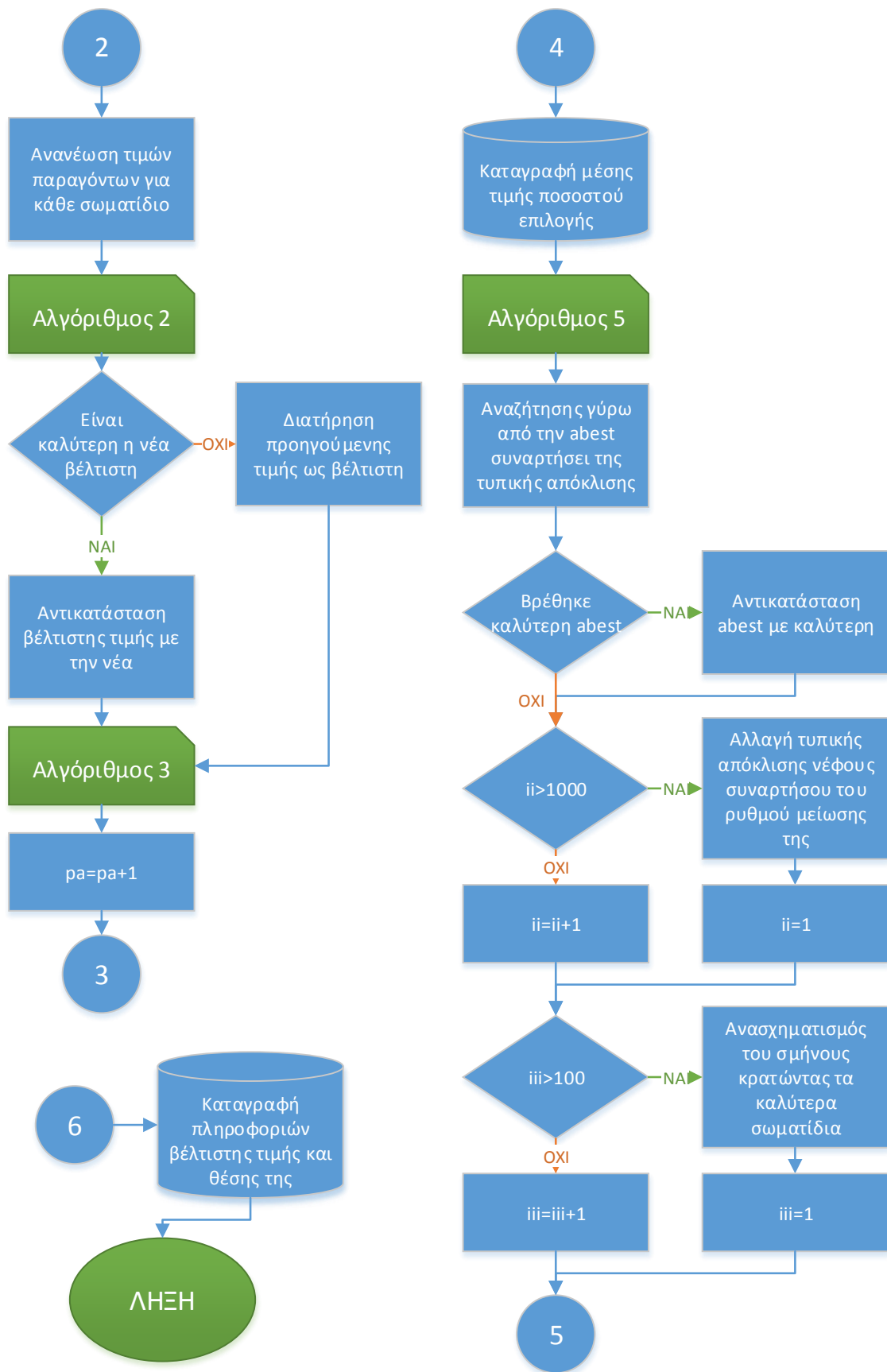
Από τη διαδικασία αυτή διαφαίνεται ότι το κομμάτι που έχουμε ως έρευνα είναι οι αυθαίρετες μεταβλητές, για τις οποίες θα πρέπει να δούμε το κατά πόσο επηρεάζουν

τις λύσεις που βρίσκουμε σε ένα πρόβλημα, αλλά και το πώς αλλάζουν από πρόβλημα σε πρόβλημα. Επίσης θα ελέγξουμε τα όρια της SL-PSO, με τις αλλαγές που έχουν πραγματοποιηθεί, σχετικά με τις μη-γραμμικές συναρτήσεις που μπορεί να επιλύσει, τα σημεία στα οποία ανταποκρίνεται ικανοποιητικά αλλά και ποια προβλήματα τυχόν θα εντοπίσει στις συναρτήσεις που θα την εφαρμόσουμε.

Αυτά τα βήματα φαίνονται στα διαγράμματα ροής (Σχήματα 3.7(a) και 3.7(b)), των επόμενων σελίδων:



**Σχήμα 3.7(a)** Διάγραμμα ροής (flowchart) του αλγορίθμου SL-PSO



Σχήμα 3.7(b) Διάγραμμα ροής (flowchart) του αλγορίθμου SL-PSO (συνέχεια)

## 3.2 Μέθοδος εDEag

### 3.2.1 Διαδικασία Ανάπτυξης της εDEag

Ένα πολύ σημαντικό κομμάτι των εξελικτικών αλγορίθμων είναι η εφαρμογή και συνεργασία των περιορισμών του προβλήματος πάνω στην αντικειμενική του συνάρτηση. Στις πιο πετυχημένες τεχνικές γίνεται ξεχωριστή διαχείριση αυτών των δύο, με στόχο τη βελτιστοποίηση της αντικειμενικής και του συνόλου των περιορισμών ταυτόχρονα. Με αυτή την αρχική λογική αναπτύχθηκε ο αλγόριθμος «ε constrained with an archive and gradient-based mutation, εDEag» (Takahama & Sakai, 2010).

Για να φτάσει ο αλγόριθμος στην τελική του μορφή πέρασε τρία βασικά στάδια :

- Αρχικά αναπτύχθηκε ο «εDE» (Takahama et al., 2006) που συνδυάζει δύο τεχνικές: μία μέθοδο που ονομάζεται «ε constrained method» για το χειρισμό των περιορισμών, και τη «διαφορική εξέλιξη» (differential evolution) για τη διαδικασία της βελτιστοποίησης.
- Στη συνέχεια αναπτύχθηκε ο «εDEg» (Takahama & Sakai, 2006) που εισήγαγε επιπλέον την έννοια της «μετάλλαξης με διαβάθμιση» (gradient-based mutation), ο οποίος μπορεί να εκμεταλλευτεί τη μέθοδο αυτή για να επιλύσει προβλήματα με δύσκολους περιορισμούς.
- Το τελευταίο κομμάτι που εισήχθη είναι της αρχειοθέτησης, με τον αλγόριθμο «εDEag» (Takahama & Sakai, 2010), ο οποίος είναι και αυτός που εξετάσαμε στην παρούσα εργασία. Με στόχο να βελτιωθεί η σταθερότητα, όλα τα σωματίδια θα έπρεπε να προσεγγίζουν τη βέλτιστη λύση με παρόμοια ταχύτητα, διατηρώντας ταυτόχρονα τη διαφορετικότητά τους. Για να πραγματοποιηθεί αυτό έγινε μία βασική αλλαγή: όταν το σωματίδιο «γονέας» δημιουργήσει έναν «απόγονο» χειρότερο από αυτόν, τότε ο «γονέας» μπορεί να δημιουργήσει έναν άλλο. Με αυτό τον τρόπο μετριάζουμε την ταχύτητα των σωματιδίων. Επίσης γνωρίζουμε ότι με το μεγαλύτερο μέγεθος του πληθυσμού βελτιώνεται η διαφορετικότητα αλλά επηρεάζεται η απόδοση του αλγορίθμου. Για να ισορροπηθούν αυτοί οι δύο παράγοντες δημιουργήθηκε ένα αρχείο (Storn & Price, 1997) που ανανεώνεται από τους ηττημένους της

εξελικτικής διαδικασίας. Επίσης, αυτοματοποιήθηκε η διαδικασία παραμετροποίησης του  $\varepsilon$  βάση της κατάστασης ενός ορισμένου αρχείου.

### 3.2.2 Μαθηματικά στοιχεία για την εDEag

Με τη χρήση της μεθόδου εDEag ερευνήσαμε προς βελτιστοποίηση ένα πρόβλημα, το οποίο αναπτύχθηκε από τον επιβλέποντα καθηγητή. Γενικά τα προβλήματα που εισάγονται στον κώδικα προς βελτιστοποίηση πρέπει να έχουν την ίδια μορφή με αυτά της μεθόδου SL-PSO ([Υποκεφάλαιο 3.1.1](#)).

### 3.2.3 Μαθηματική μοντελοποίηση της εDEag

Στην «ε constrained method» οι περιορισμοί μπορούν να δοθούν με δύο μορφές  $\varphi(x)$ . Είτε χρησιμοποιώντας τη μέγιστη τιμή από όλους τους περιορισμούς :

$$\varphi(x) = \max \{ \max_i \{0, g_i(x)\}, \max_j \{0, h_j(x)\} \} \quad (3.22)$$

Είτε με άθροιση όλων των περιορισμών :

$$\varphi(x) = \sum_i^m \|\max\{0, g_i(x)\}\|^p + \sum_j^k \|\max\{0, h_j(x)\}\|^p \quad (3.23)$$

Όπου  $p$  είναι ένας θετικός αριθμός.

Οι  $\varepsilon$  βαθμού συγκρίσεις ορίζονται ως μία «σχέση τάξης» ζεύγους τιμής αντικειμενικής συνάρτησης και τιμής παραβίασης περιορισμού ( $f(x), \varphi(x)$ ). Εάν η τιμή παραβίασης είναι μεγαλύτερη του 0 τότε η θέση είναι μη-εφικτή και συνεπώς έχει χαμηλή αξία. Οι  $\varepsilon$  βαθμού συγκρίσεις βασικά ορίζονται ως μια «λεξικογραφική» σειρά όπου το  $\varphi(x)$  προηγείται του  $f(x)$ , διότι είναι πιο σημαντικό το  $x$  να είναι εφικτή λύση από τη βελτιστοποίηση του  $f(x)$ . Αυτή η προτεραιότητα μπορεί να αλλάξει από την παράμετρο  $\varepsilon$ .

Έτσι, εάν έχουμε τιμή αντικειμενικής  $f_1, f_2$  και τιμή παραβίασης  $\varphi_1, \varphi_2$  δύο σημείων  $x_1, x_2$  αντίστοιχα, με  $\varepsilon \geq 0$ , οι  $\varepsilon$  βαθμού συγκρίσεις  $<_\varepsilon$  και  $\leq_\varepsilon$  μεταξύ των  $(f_1, \varphi_1)$  και  $(f_2, \varphi_2)$  ορίζονται ακολούθως:

$$(f_1, \varphi_1) <_\varepsilon (f_2, \varphi_2) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{εάν } \varphi_1, \varphi_2 \leq \varepsilon \\ f_1 < f_2, & \text{εάν } \varphi_1 = \varphi_2 \\ \varphi_1 < \varphi_2, & \text{αλλιώς} \end{cases} \quad (3.24)$$

$$(f_1, \varphi_1) \leq_\varepsilon (f_2, \varphi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{εάν } \varphi_1, \varphi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{εάν } \varphi_1 = \varphi_2 \\ \varphi_1 < \varphi_2, & \text{αλλιώς} \end{cases} \quad (3.25)$$

Σε περίπτωση που  $\varepsilon = \infty$ , οι  $\varepsilon$  βαθμού συγκρίσεις  $<_\infty$  και  $\leq_\infty$  είναι ισοδύναμες με τις συνηθισμένες συγκρίσεις  $<$  και  $\leq$ , μεταξύ τιμών της αντικειμενικής. Ακόμα, σε περίπτωση που  $\varepsilon = 0$ ,  $<_0$  και  $\leq_0$  είναι ισοδύναμες με τις λεξικογραφικές σειρές στις οποίες η παραβίαση των περιορισμών  $\varphi(x)$  προηγείται της τιμής της αντικειμενικής συνάρτησης  $f(x)$ .

Ουσιαστικά, η βασική ιδιότητα της «ε constrained method» είναι ότι καταφέρνει και μετατρέπει ένα πρόβλημα βελτιστοποίησης με περιορισμούς σε ένα χωρίς περιορισμούς, αντικαθιστώντας τη «σχέση τάξης» σε μία μέθοδο ευθείας αναζήτησης με τις  $\varepsilon$  βαθμού συγκρίσεις. Έτσι το αντίστοιχο νέο πρόβλημα βελτιστοποίησης ορίζεται ως:

$$P_{\leq_\varepsilon} \text{ ελαχιστοποίηση}_{\leq_\varepsilon} f(x) \quad (3.26)$$

Όπου η *ελαχιστοποίηση* <sub>$\leq_\varepsilon$</sub>  υποδηλώνει την ελαχιστοποίηση βάση της  $\varepsilon$  βαθμού σύγκρισης  $\leq_\varepsilon$ . Επίσης, ένα πρόβλημα  $P^\varepsilon$  ορίζεται έτσι ώστε οι περιορισμοί της  $P$ , που είναι  $\varphi(x) = 0$ , να είναι αδιάφοροι και να αντικαθίστανται με  $\varphi(x) \leq \varepsilon$ . Το πρόβλημα ορίζεται ως:

$$\begin{aligned} P^\varepsilon \text{ ελαχιστοποίηση}_{\leq_\varepsilon} f(x) \\ \text{που υπόκειται σε } \varphi(x) \leq \varepsilon \end{aligned} \quad (3.27)$$

Είναι προφανές ότι  $P^0$  ισούται με  $P$ . Επίσης έχει αποδειχτεί (Takahama & Sakai, 2005) ότι, κάτω από κατάλληλες προϋποθέσεις,  $P_{\leq_\varepsilon}$  είναι ισοδύναμο του  $P^\varepsilon$  και ότι η βέλτιστη λύση του  $P^0$  μπορεί να βρεθεί από τη σύγκλιση του  $\varepsilon$  στο 0, καθώς και με την αύξηση του συντελεστή ποινής στο άπειρο, στη μέθοδο με ποινή στην αντικειμενική συνάρτηση.

Για τη διαδικασία βελτιστοποίησης χρησιμοποιείται η μέθοδος της «διαφορικής εξέλιξης» (differential evolution, DE) που προτάθηκε από τους Storn και Price (Storn & Price, 1997), η οποία είναι μία μέθοδος στοχαστικής ευθείας αναζήτησης που χρησιμοποιεί είτε πληθυσμό είτε πολλαπλά σημεία αναζήτησης. Η «differential evolution» έχει εφαρμοστεί επιτυχημένα, γρήγορα και με σταθερότητα σε προβλήματα βελτιστοποίησης συμπεριλαμβανομένων μη-γραμμικών και άλλων πολύπλοκων συναρτήσεων. Στη μέθοδο αυτή υπάρχουν κάποιες παραλλαγές που ορίζονται με τη μορφή «DE/βάση(base)/αριθμός(num)/διασταύρωση(cross)» όπου το καθένα σημαίνει :

- Ως «βάση» ορίζεται η μέθοδος με την οποία θα επιλεγεί το σωματίδιο που θα σχηματίζει το βασικό διάνυσμα. Για παράδειγμα, εάν έχει την τιμή «τυχαίο» (rand) τότε θα επιλεγεί ένα τυχαίο από τον πληθυσμό, ενώ εάν έχει την τιμή «βέλτιστο» (best) θα επιλεγεί το καλύτερο. Έτσι εάν έχουμε «DE/τυχαίο/1», για κάθε σωματίδιο  $x^i$ , τρία σωματίδια  $x^{p1}$ ,  $x^{p2}$  και  $x^{p3}$  θα επιλεγούν από τον πληθυσμό χωρίς να υπερκαλύπτονται μεταξύ τους ή με το  $x^i$ , και θα μας δώσουν ένα νέο ή μεταλλαγμένο διάνυσμα  $x'$  με βάση το διάνυσμα  $x^{p1}$  και το διάνυσμα της διαφοράς των  $x^{p2} - x^{p3}$ , δηλαδή :

$$x' = x^{p1} + F(x^{p2} - x^{p3}) \quad (3.28)$$

Όπου  $F$  είναι ένας παράγοντας κλιμάκωσης.

- Ο δείκτης «αριθμός» (num) υποδεικνύει τον αριθμό των διανυσμάτων διαφοράς που θα χρησιμοποιηθούν για να επηρεάσουν το βασικό διάνυσμα.
- Ο δείκτης «διασταύρωση» υποδεικνύει το μηχανισμό διασταύρωσης που χρησιμοποιήθηκε για την παραγωγή ενός απογόνου. Για παράδειγμα, εάν αυτός έχει την τιμή «exp» (exponential) τότε η διασταύρωση γίνεται με έναν τύπο εκθετικής παρεμβολής με μειούμενο ρυθμό διασταύρωσης (CR). Ο ψευδοκώδικας μίας τέτοιας διασταύρωσης με  $x^{child}$  τον απόγονο,  $x^i$  το γονέα,  $x'$  το μεταλλαγμένο διάνυσμα και  $CR$  το ρυθμό διασταύρωσης είναι:



```

k=1; j=randint(1,n);
do {
    xjchild=x'j;
    k=k+1; j=(j+1)%n;
} while (k ≤ n && u(0,1) < CR);
while (k ≤ n) {
    xjchild=xji;
    k=k+1; j=(j+1)%n;
}

```

Fig. 1. Exponential crossover operation, where randint(1,n) generates an integer randomly from [1, n].

### Σχήμα 3.8 Ψευδοκώδικας διαδικασίας «διασταύρωσης»

Όσον αφορά τον έλεγχο και υπολογισμό του βαθμού  $\varepsilon$ , χρησιμοποιούμε τις παρακάτω εξισώσεις :

$$\varepsilon(0) = \varphi(x_\theta)$$

$$\varepsilon(t) = \begin{cases} \varepsilon(0) \left(1 - \frac{t}{T_c}\right)^{cp}, & 0 < t < T_c \\ 0, & t \geq T_c \end{cases} \quad (3.29)$$

Το αρχικό  $\varepsilon(0)$  είναι η ποινή των περιορισμών των καλύτερων  $\theta$  σωματιδίων ( $x_\theta$ ) από τις αρχικές θέσεις αναζήτησης. Επίσης, εάν  $\theta > 1$  τότε  $\varphi(x_\theta) = \theta \times \max_x[\varphi(x)]$ . Στη συνέχεια ο βαθμός του  $\varepsilon$  ανανεώνεται συνεχώς έως ότου ο αριθμός της τωρινής επανάληψης  $t$  γίνει ίσος με τη γενιά ελέγχου  $T_c$ . Από τη στιγμή που το  $t$  ξεπεράσει το  $T_c$ , ο βαθμός  $\varepsilon$  γίνεται ίσος με 0, έτσι ώστε να εστιάσουμε σε λύσεις με ελάχιστη δυνατή παραβίαση περιορισμών.

Σε αυτό τον κώδικα που χρησιμοποιήθηκε σε αυτή την εργασία, εφαρμόστηκε και μία απλή μέθοδος για τον υπολογισμό της παραμέτρου  $cp$  :

$$\varepsilon_\lambda = \varepsilon(T_\lambda) = \varepsilon(0)(1 - T_\lambda/T_c)^{cp} \quad (3.30)$$

$$\begin{aligned} cp &= (\log \varepsilon_\lambda - \log \varepsilon(0)) / \log(1 - T_\lambda/T_c) \\ &= (-5 - \log \varepsilon(0)) / \log 0.05 \end{aligned} \quad (3.31)$$

Με το βαθμό  $\varepsilon$  να έχει ορισθεί σε μία μικρή τιμή  $\varepsilon_\lambda = 10^{-5}$  για τη γενιά  $T_\lambda = 0.95T_c$ . Για να αποφύγουμε το  $cp$  να πάρει πολύ μικρές τιμές, ορίσαμε μία ελάχιστη τιμή  $cp_{min} = 3$ , έτσι ώστε εάν το  $cp$  γίνει μικρότερο από αυτή, να παραβλέπει τον παραπάνω τρόπο υπολογισμού και να ορίζεται με αυτή την ελάχιστη τιμή. Ακόμα,

μετά τη  $T_\lambda$  επανάληψη, για να μεγαλώσουμε το βαθμό  $\varepsilon$ , έτσι ώστε να μεγαλώσουμε την εφικτή περιοχή του και να ψάξουμε για καλύτερες λύσεις της αντικειμενικής, η τιμή της παραμέτρου  $cp$  μειώνεται και ταυτόχρονα της  $F$  αυξάνεται ακολούθως :

$$cp = 0.3cp + 0.7cp_{min} \quad (3.32)$$

$$F = 0.3F_0 + 0.7 \quad (3.33)$$

Όσον αφορά το κομμάτι «μετάλλαξης με διαβάθμιση» (gradient-based mutation) (Takahama & Sakai, 2006), είναι μία μέθοδος παρόμοια με τη «gradient-based repair method» (Chootinan & Chen, 2006). Δηλαδή, το διάνυσμα των συναρτήσεων των περιορισμών  $C(x)$ , το διάνυσμα των παραβιάσεων των περιορισμών  $\Delta C(x)$  και η αύξηση ενός σημείου  $x$  κατά  $\Delta x$ , για να ικανοποιούνται οι περιορισμοί ορίζονται ως :

$$C(x) = \left( g_1(x) \dots g_q(x) h_{q+1}(x) \dots h_m(x) \right)^T \quad (3.34)$$

$$\Delta C(x) = \left( \Delta g_1(x) \dots \Delta g_q(x) h_{q+1}(x) \dots h_m(x) \right)^T \quad (3.35)$$

$$\nabla C(x) \Delta x = -\Delta C(x) \quad (3.36)$$

$$\Delta x = -\nabla C(x)^{-1} \Delta C(x) \quad (3.37)$$

Όπου  $\Delta g_j(x) = \max\{0, g_j(x)\}$  και  $\nabla C(x)$  είναι το διάνυσμα της μετάλλαξης του  $C(x)$ .

$$\nabla C(x) = \begin{pmatrix} \frac{\theta g_1(x)}{\theta x_1} & \frac{\theta g_1(x)}{\theta x_2} & \dots & \frac{\theta g_1(x)}{\theta x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\theta g_q(x)}{\theta x_1} & \frac{\theta g_q(x)}{\theta x_2} & \dots & \frac{\theta g_q(x)}{\theta x_n} \\ \frac{\theta h_{q+1}(x)}{\theta x_1} & \frac{\theta h_{q+1}(x)}{\theta x_2} & \dots & \frac{\theta h_{q+1}(x)}{\theta x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\theta h_m(x)}{\theta x_1} & \frac{\theta h_m(x)}{\theta x_2} & \dots & \frac{\theta h_m(x)}{\theta x_n} \end{pmatrix} \quad (3.38)$$

Όταν κάποιες συναρτήσεις περιορισμών δεν είναι διαφορίσιμες, το διάνυσμα της μετάλλαξης  $\nabla C(x)$  μπορεί να υπολογιστεί αριθμητικά με υπολογισμό του  $C(x)$  επαναλαμβανόμενα, αλλάζοντας τις τιμές των μεταβλητών απόφασης από  $x_i$  σε  $x_i + \eta$ :

$$\nabla C(x) = \frac{1}{\eta} (C(x + \eta e_1) - C(x), \dots, C(x + \eta e_n) - C(x)) \quad (3.39)$$

Όπου  $e_i$  είναι ένα μοναδιαίο διάνυσμα, με το  $i_{\text{οστο}}$  στοιχείο να είναι 1, όλα τα άλλα 0, και το  $\eta$  μία πολύ μικρή ποσότητα. Παρόλο που το  $\nabla C(x)$  γενικά, δεν είναι αντιστρέψιμο, με τη μέθοδο Moore-Penrose (Campbell & Meyer, 1979) μπορούμε να ορίσουμε το ψευδο-αντίστροφο  $\nabla C(x)^+$ , για τους παραπάνω υπολογισμούς.

Έτσι, με τον υπολογισμό του  $\Delta x$ , μπορούμε να ορίσουμε ένα μεταλλαγμένο διάνυσμα:

$$x^{\text{new}} = x + \Delta x \quad (3.40)$$

Η διαδικασία της μετάλλαξης με διαβάθμιση πραγματοποιείται με πιθανότητα  $P_g$ , και με αυτή να επαναλαμβάνεται με σταθερό αριθμό φορών  $R_g$ , για όσο το σημείο δεν είναι εφικτό. Επίσης, όταν παραβιάζεται μόνο ένας περιορισμός, είναι πολύ δύσκολο να αποφασιστεί η κατάλληλη τιμή για το  $\Delta x$ , επειδή υπάρχουν πολλές τιμές που ικανοποιούν τη (3.36). Για αυτή την περίπτωση εισήχθη μία τεχνική παράλειψης, όπου η μετάλλαξη με διαβάθμιση δεν εφαρμόζεται, με πιθανότητα 0.5.

### 3.2.4 Αλγόριθμος της εDEag

Με βάση όλα τα παραπάνω, δημιουργήθηκαν τα παρακάτω βήματα που περιλαμβάνουν τη σειρά των διεργασιών της εξελικτικής διαδικασίας της μεθόδου εDEag :

- Βήμα 1** Για πλήθος  $M$  δημιουργείται ένα αρχείο  $A = \{x^i, i = 1, 2, \dots, M\}$  των σωματιδίων  $x^i$ , με τυχαίες θέσεις, για αυτά, εντός του πεδίου αναζήτησης  $S$ .
- Βήμα 2** Αρχικοποιείται ο βαθμός  $\varepsilon$ , με αρχική τιμή να υπολογίζεται από την εξίσωση ελέγχου  $\varepsilon(0)$ .
- Βήμα 3** Αρχικοποιείται ο πληθυσμός. Επιλέγονται τα καλύτερα  $N$  σωματίδια από το αρχείο  $A$  και έτσι δημιουργείται ο πληθυσμός  $P = \{x^i\}$ . Η κατάταξη με βάση την οποία επιλέγεται ο πληθυσμός δημιουργείται από τις βαθμού  $\varepsilon$  συγκρίσεις.

- Βήμα 4** Συνθήκη τερματισμού. Εάν ο αριθμός αξιολογήσεων της αντικειμενικής συνάρτησης έχει ξεπεράσει το μέγιστο επιτρεπόμενο όριο  $FE_{max}$  ο αλγόριθμος τερματίζεται.
- Βήμα 5** Διεργασίες της «differential evolution», με μέγιστο πλήθος αυτών, δύο. Κάθε σωματίδιο  $x^i$  επιλέγεται με τη σειρά, ένα-ένα, ως γονέας. Εάν όλα έχουν επιλεγεί προχωράμε στο Βήμα 7. Η διεργασία  $DE/rand/1/exp$  εφαρμόζεται και έτσι ένας νέος απόγονος δημιουργείται. Ένας σταθερός παράγοντας κλιμάκωσης  $F_0$ , καθώς και ένας σταθερός δείκτης διασταύρωσης  $CR_0$  χρησιμοποιούνται με πιθανότητα 0.95, ενώ ένας τυχαία δημιουργημένος παράγοντας κλιμάκωσης  $F$  και ο δείκτης διασταύρωσης  $CR$  χρησιμοποιούνται με πιθανότητα 0.05. Το τρίτο διάνυσμα  $x^{p3}$  επιλέγεται από το αρχείο  $A$  με πιθανότητα 0.95 και από τον πληθυσμό  $P$  με πιθανότητα 0.05. Εάν ο απόγονος είναι καλύτερος, βασισμένο στις βαθμού  $\varepsilon$  συγκρίσεις, από το γονέα, ο δεύτερος αντικαθίσταται από τον πρώτο και περνάμε στο Βήμα 6. Σε περίπτωση που το τελευταίο δεν ισχύει, τότε η διαδικασία αυτού του βήματος επαναλαμβάνεται για ακόμα μία φορά, για αυτό τον γονέα.
- Βήμα 6** Διαδικασία της «μετάλλαξης με διαβάθμιση». Εάν ο νέος απόγονος  $x^{child}$  που δημιουργήθηκε δεν είναι εφικτή λύση ή  $\varphi(x^{child}) > 0$ , τότε αυτός αλλάζει, με «μετάλλαξη με διαβάθμιση», με πιθανότητα  $P_g$  έως: είτε ο αριθμός των αλλαγών να φτάσει σε ένα μέγιστο όριο  $R_g$ , είτε ο απόγονος αυτός να αλλάξει σε μία εφικτή λύση. Στη συνέχεια επανερχόμαστε στο Βήμα 5.
- Βήμα 7** Έλεγχος του βαθμού  $\varepsilon$  και ενημέρωση του με βάση τη συνάρτηση  $\varepsilon(t)$ . Στη συνέχεια επανερχόμαστε στο Βήμα 4.

Αυτά τα βήματα φαίνονται στον παρακάτω ψευδοκώδικα (Σχήμα 3.9), και στο αντίστοιχο, επόμενο, διάγραμμα ροής (Σχήματα 3.10(a) και 3.10(b)):

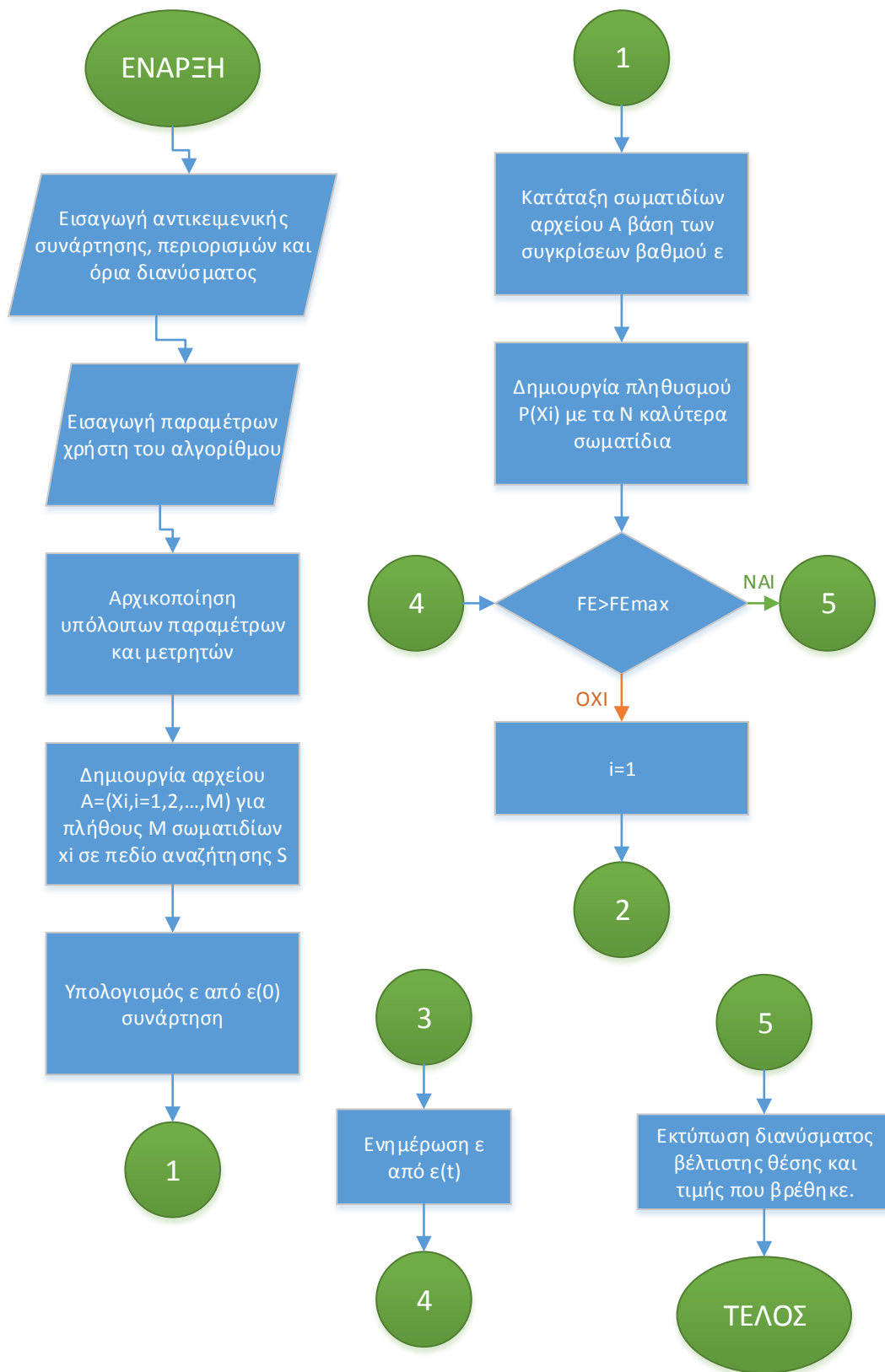
```

εDEag()
{
  F=F0; CR=CR0;
  // Initialize an archive
  A=M individuals generated randomly in S;
  FE=M;
  // Initialize the ε level
  ε=ε(0);
  // Initialize a population
  P=Top N individuals {xi} from A using <ε;
  A=A-P;
  for(t=1; FE ≤ FEmax; t++) {
    F=F0;
    if(t > 0.95TC && t < TC)
      Modify the ε level and F using Eq.(3.32),(3.33);
    if(u(0,1)<0.05)
      F=1+|randG(0,0.05)|, F is truncated to 1.1;
    for(i=1; i ≤ N; i++) {
      // DE operation
      for(k=1; k ≤ 2; k++) {
        xp1=Randomly selected from P;
        xp2=Randomly selected from P;
        if(u(0,1)<0.05)
          xp3=Randomly selected from P;
        else
          xp3=Randomly selected from P ∪ A;
        x' = xp1 + (xp2 - xp3);
        xchild=trial vector is generated
          from xi and x' by exponential crossover;
      // Gradient-based mutation
      if(t%n==0 && u(0,1) < Pg)
        for(h=1; h ≤ Rg && φ(xchild)>0; h++) {
          Apply gradient-based mutation to xchild;
          FE=FE+n+1;
        }
      else FE=FE+1;
      if((f(xchild), φ(xchild)) < ε (f(xi), φ(xi))) {
        xi=xchild;
        break;
      } else {
        xA=Randomly selected from A;
        A=A+{xchild}-{xA};
      }
    }
  }
  // Control the ε level
  ε=ε(t);
}

```

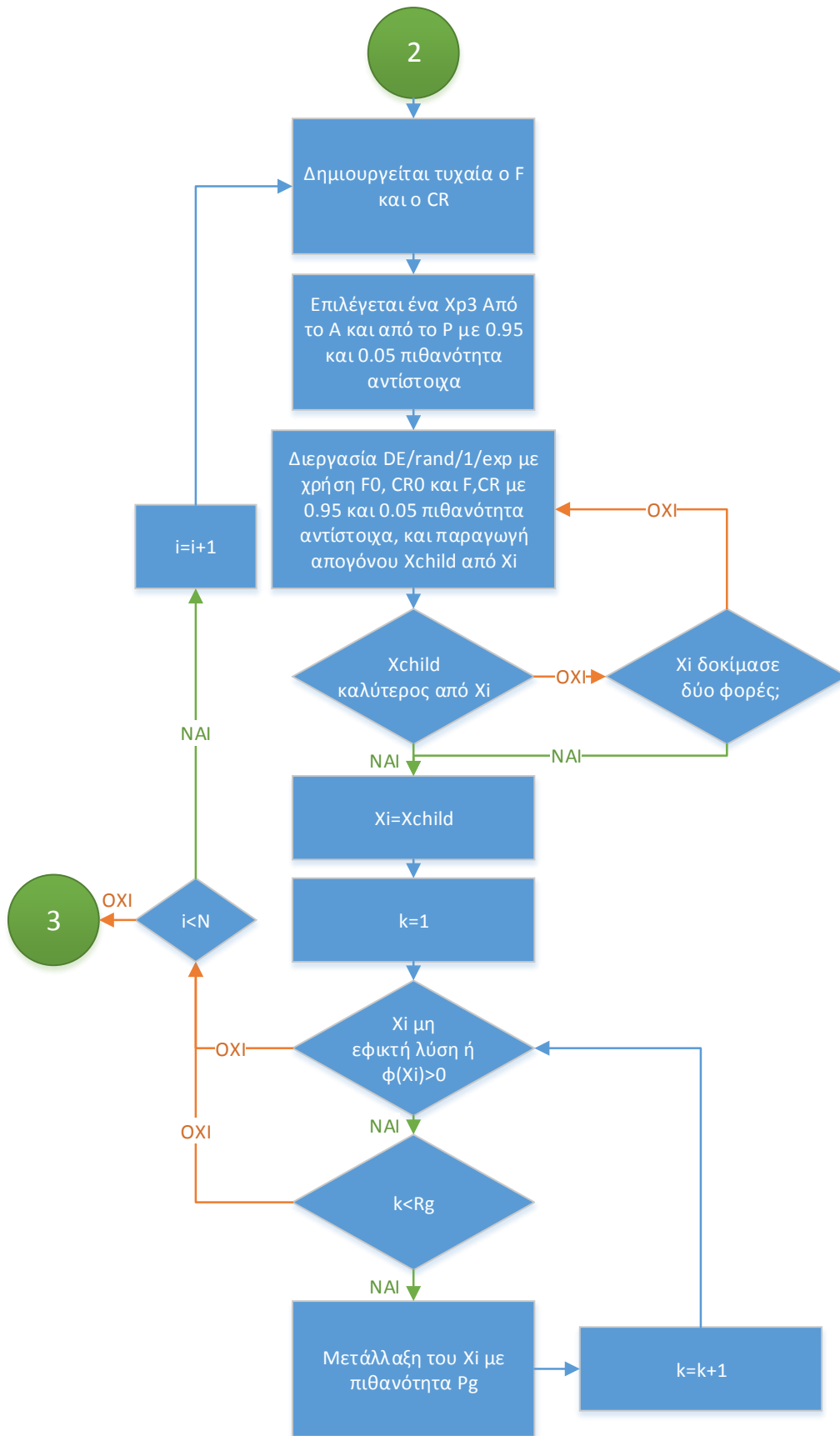
Fig. 2. The algorithm of the  $\varepsilon$ DEag, where  $\varepsilon(t)$  is the  $\varepsilon$  level control function,  $FE$  is the number of function evaluations,  $u(l, r)$  is a uniform random number generator in  $[l, r]$ ,  $rand_G(\mu, \sigma)$  is a random number generator obeying Gaussian distribution with the mean  $\mu$  and the standard deviation  $\sigma$ , and  $P_g$  and  $R_g$  are parameters for gradient-based mutation.

### Σχήμα 3.9 Ψευδοκώδικας αλγορίθμου εDEag



**Σχήμα 3.10(a)**

Διάγραμμα ροής (flowchart) του αλγορίθμου εDEag



Σχήμα 3.10(b)

Διάγραμμα ροής (flowchart) του αλγορίθμου eDEag (συνέχεια)





## **ΚΕΦΑΛΑΙΟ 4**

### **ΤΑ ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ** **ΕΞΕΤΑΣΤΗΚΑΝ**



#### **4.1 Προβλήματα που εξετάστηκαν με την SL-PSO**

Τα προβλήματα που εξετάσαμε, στα πλαίσια αυτής της διπλωματικής, στην SL-PSO είναι από το διαγωνισμό CEC 2010 (Mallipeddi & Suganthan, 2010), και οι συναρτήσεις τους βρίσκονται στο «[Παράρτημα Α](#)».

#### **4.2 Οικονομική κάλυψη ηλεκτρικού φορτίου συστήματος**

Ένα πολύ ενδιαφέρον πρόβλημα είναι η οικονομική κάλυψη της βραχυπρόθεσμης ζήτησης ενός δικτύου σε ηλεκτρικό φορτίο, ή πιο διαδεδομένα: «Economic Load Dispatch» (ELD). Η βασική ιδέα πίσω από αυτό είναι η βέλτιστη αξιοποίηση, από το διαχειριστή του συστήματος, των διαθέσιμων μονάδων παραγωγής ηλεκτρικής ενέργειας, για την ικανοποίηση των απαιτήσεων του δικτύου σε ηλεκτρικό φορτίο, με στόχο το ελάχιστο δυνατό κόστος (σε κατανάλωση καυσίμου), πάντα σεβόμενοι τους περιορισμούς της κάθε μονάδας.

Ο λόγος που το ELD αποτελεί από μόνο του μία ειδική κατηγορία προβλημάτων είναι ο ιδιαίτερος αντίκτυπος που έχει άμεσα και έμμεσα στην οικονομία μίας κοινωνίας. Αρχικά, επηρεάζει έντονα, και άμεσα, τη βιομηχανία ηλεκτρικής ενέργειας, καθώς τα κόστη είναι ιδιαίτερα μεγάλα και μία καλή διαχείριση μπορεί να επηρεάσει σημαντικά την κερδοφορία μίας εταιρίας που ανήκει σε αυτό τον κλάδο. Επηρεάζει, επίσης, έμμεσα την τιμή της ηλεκτρικής ενέργειας, με την οποία την προμηθεύονται οι καταναλωτές. Ακόμα, επηρεάζει το περιβάλλον, καθώς ελαχιστοποιείται η χρήση ορυκτών καυσίμων, χωρίς να μειώνεται η κάλυψη του φορτίου. Το τελευταίο επηρεάζει επίσης, γιατί πολύ διεθνείς οργανισμοί, όπως ο Ο.Η.Ε και η Ε.Ε., απαιτούν μείωση της εκπομπής αποβλήτων παραγωγής ενέργειας (διοξείδιο του άνθρακα και άλλα αέρια του «θερμοκηπίου») από τα κράτη και τις εταιρίες (Συνθήκη του Κιότο). Όλα αυτά καθιστούν την επίλυση αυτού του προβλήματος, με τον καλύτερο δυνατό τρόπο, πολύ σημαντική για τις εταιρίες του κλάδου, καθώς και για τα διάφορα κράτη και διεθνείς οργανισμούς, και συνεπώς αποτελεί πόλο έλξης για τους ερευνητές.

Το ιδιαίτερο που έχει αυτό το πρόβλημα είναι ότι η χαρακτηριστική καμπύλη κόστους κατανάλωσης καυσίμου είναι μη-γραμμική λόγω της ύπαρξης μη-γραμμικών, αλλά και γραμμικών, εξισώσεων περιορισμών (Singh & Kumar, 2013). Αυτό καθιστά το ELD ιδιαίτερα ενδιαφέρον, από μαθηματική άποψη, καθώς η επίλυσή του είναι δύσκολο να γίνει με συμβατικές μαθηματικές τεχνικές επίλυσης

προβλημάτων. Ακόμα, λόγω του πλήθους των εξισώσεων, της πολυπλοκότητάς τους (καθώς μπορεί να έχουμε πάρα πολλές, διαφορετικές, διαθέσιμες μονάδες παραγωγής ενέργειας), και της ταχύτητας που απαιτείται για την επίλυσή του (καθώς αφορά βραχυπρόθεσμη κάλυψη της ζήτησης), είναι απαραίτητη η χρήση προηγμένων υπολογιστικών συστημάτων. Όλα αυτά το καθιστούν σημαντικό υποψήφιο για έρευνα και επίλυση με τη χρήση εξελικτικών αλγορίθμων, κυρίως λόγω της ικανότητας των τελευταίων: 1) να προσαρμόζονται γρήγορα και εύκολα στις ιδιαιτερότητες κάθε προβλήματος και 2) να επιλύουν ιδιαίτερα πολύπλοκα μη-γραμμικά προβλήματα.

### **4.3 Πρόβλημα Διαχείρισης Παραγωγής Ενέργειας (ΠΔΠΕ)**

Το πρόβλημα αυτό αφορά την κάλυψη ζήτησης ενεργειακών αναγκών σε κάθε ώρα μίας ημέρας, εκμεταλλευόμενοι με τον καλύτερο τρόπο τα υφιστάμενα εργοστάσια παραγωγής και σεβόμενοι τους περιορισμούς του καθενός, με στόχο το ελάχιστο δυνατό κόστος σε χρηματικές μονάδες. Αποτελεί, ουσιαστικά, ένα πρόβλημα ELD. Σε αντίθεση με τα προηγούμενα προβλήματα ([υποκεφάλαιο 4.1](#)), το παρόν αφορά ένα ρεαλιστικό πρόβλημα, είναι αρκετά πιο δύσκολο στην υλοποίησή του στους κώδικες που ενασχοληθήκαμε, και απαιτεί περισσότερο υπολογιστικό χρόνο, λόγω του πλήθους των δεδομένων και περιορισμών που έχουμε.

Η αντικειμενική συνάρτηση, που αφορά το συνολικό κόστος προς ελαχιστοποίηση, είναι της μορφής:

$$f(x) = \sum_{i=1}^{24} \left[ \sum_{j=1}^{25} (a_j x_{j,i}^2 + b_j x_{j,i} + c_j) \right] \quad (4.1)$$

Όπου «i» αφορά την κάθε μία από τις 24 ώρες της ημέρας, και «j» αφορά το κάθε ένα από τα 25 εργοστάσια παραγωγής. Όπως βλέπουμε η λύση της  $f(x)$  είναι ένας πίνακας  $x(25,24)$ . Τα «a, b, c» βρίσκονται στον πίνακα «Uc» ([Πίνακας 4.2](#)) για κάθε εργοστάσιο.

Ο περιορισμός που έχουμε είναι της μορφής:

$$g_i(x) = \sum_{j=1}^{25} (x_{j,i}) - d_i \geq 0 \quad , \text{για } i \text{ απο } 1 \text{ έως } 24 \quad (4.2)$$

Το « $d$ » αφορά την ζήτηση για κάθε « $i$ ». Αυτός πρέπει να ισχύει για κάθε ώρα « $i$ » της ημέρας από 1 έως 24.

Όσον αφορά το πεδίο ορισμού, για κάθε  $x_{i,j}$  έχουμε μία ελάχιστη και μία μέγιστη τιμή. Αυτές οι τιμές είναι ανεξάρτητες του « $i$ » αλλά εξαρτημένες του « $j$ », δηλαδή το κάθε εργοστάσιο παραγωγής έχει ένα ελάχιστο και ένα μέγιστο όριο που μπορεί να παράγει, το οποίο για κάθε ώρα της ημέρας είναι σταθερό. Στο πίνακα « $mm$ » αναφέρονται αυτά τα όρια ανά μονάδα παραγωγής ενέργειας.

Για το συγκεκριμένο πρόβλημα, επειδή έχει δημιουργηθεί από τον επιβλέποντα καθηγητή, δεν έχουμε κάποια βιβλιογραφική αναφορά για την τιμή της βέλτιστης λύσης.

Στην επόμενη σελίδα βρίσκονται οι πίνακες: της ζήτησης για κάθε μία από τις 24 ώρες (Πίνακας 4.1), των « $a, b, c$ » για κάθε εργοστάσιο παραγωγής (Πίνακας 4.2) και τα όρια του κάθε εργοστασίου, σε ωριαία παραγωγή ενέργειας (Πίνακας 4.2).

Ζήτηση «d» ανά ώρα «i» της ημέρας											
1	2	3	4	5	6	7	8	9	10	11	12
700	750	468	432	580	590	610	670	760	790	920	1010
13	14	15	16	17	18	19	20	21	22	23	24
1103	955	904	840	805	748	676	650	649	621	635	790

**Πίνακας 4.1** Πίνακας ζήτησης συναρτήσει της ώρας της ημέρας

<i>Uc</i> Εργοστάσιο Παραγωγής «j»	Κόστη ανά μονάδα ενέργειας			<i>mm</i> Εργοστάσιο Παραγωγής «j»	Μονάδες ενέργειας	
	a	b	c		Ελάχιστο	Μέγιστο
1	1.8	55.9	55	1	10	50
2	0.8	39.2	40	2	12	48
3	1.7	290.4	5	3	15	55
4	21.9	120.9	43.3	4	13	67
5	21.9	120.9	43.3	5	23	110
6	20	33	45	6	22	43
7	1.3	53	3	7	26	76
8	24	32	3	8	18	54
9	12	300	46	9	29	45
10	14	138	43	10	30	52
11	22	147	55	11	16	43
12	1.5	35	42	12	12	51
13	2.5	45	5	13	15	50
14	3.1	120	43.3	14	25	48
15	1.5	128	43.3	15	26	55
16	19	116	17	16	14	67
17	1.5	120.9	29	17	16	127
18	19	120.9	35	18	17	54
19	1.8	33	55	19	22	45
20	2.3	53	40	20	13	52
21	27	32	7	21	16	43
22	21.7	300	43.3	22	12	51
23	2.4	260	43.3	23	27	98
24	2.8	400	51	24	20	86
25	1.5	35	42	25	26	47

**Πίνακας 4.2** Πίνακες για τα κόστη και το πεδίο ορισμού, για κάθε εργοστάσιο

## **ΚΕΦΑΛΑΙΟ 5**

# **ΠΕΙΡΑΜΑΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΙ** **ΑΠΟΤΕΛΕΣΜΑΤΑ**





## 5.1 Παράμετροι για την SL-PSO στην MATLAB®

Για τον αλγόριθμο της SL-PSO στη MATLAB®, που χρησιμοποιήσαμε, έχουμε πέντε γενικές παραμέτρους, με σταθερές προκαθορισμένες τιμές:

- loramax, με την οποία ορίζουμε τον αριθμό των φορών (κύκλων, «τρεξιμάτων») που θέλουμε να εκτελέσουμε ένα πείραμα σε ένα πρόβλημα, για στατιστική ανάλυση των αποτελεσμάτων. Η τιμή της ορίστηκε σταθερά στις 25, από τον επιβλέποντα καθηγητή.
- itermax, με την οποία δηλώνουμε τον αριθμό των επαναλήψεων που εκτελούνται σε ένα πρόβλημα και αποτελεί τη μόνη παράμετρο τερματισμού του αλγορίθμου. Η τιμή της ορίστηκε σταθερά στις 2000, από τον επιβλέποντα καθηγητή.
- sw. Με αυτή την παράμετρο ορίζουμε τις διαστάσεις του διανύσματος λύσης της αντικειμενικής συνάρτησης του προβλήματος. Σε όλα τα προβλήματα έγινε έλεγχος για 10 και 30 διαστάσεις, σύμφωνα με τις απαιτήσεις του διαγωνισμού CEC-2010.
- pa, με την οποία ορίζουμε τον αριθμό των σωματιδίων που έχουμε στο σμήνος του αλγορίθμου. Σε σύμφωνη γνώμη με τον επιβλέποντα καθηγητή, αυτή ορίστηκε στην τιμή 5.
- elfactor, που με αυτή δηλώνουμε πόσες φορές, παραπάνω, είναι μεγαλύτερο το πλήθος των σωματιδίων του νέφους από αυτών του σμήνους. Σε σύμφωνη γνώμη με τον επιβλέποντα καθηγητή, αυτή ορίστηκε στην τιμή 50.

Έχουμε ακόμα τρεις βασικές παραμέτρους, που εξετάσαμε ως προς τον τρόπο που επηρεάζουν τη σύγκλιση των προβλημάτων του διαγωνισμού CEC-2010. Αυτές είναι:

- initstd, με την οποία ορίζουμε την αρχική τυπική απόκλιση του νέφους γύρω από τη θέση  $a_{best}$ .
- istreductratefinal, με την οποία ορίζουμε τον ρυθμό μείωσης της τυπικής απόκλισης (initstd) του νέφους.
- λ, με την οποία ορίζουμε το τεχνητό βάρος των περιορισμών πάνω στην αντικειμενική συνάρτηση.

Για αυτές τις τρεις τελευταίες έγιναν διάφορα πειράματα για την εύρεση των καλύτερων δυνατών τιμών που βελτιστοποιούν την τιμή της αντικειμενικής συνάρτησης του κάθε προβλήματος.

Για κάθε πρόβλημα έχουμε δύο ακόμα μεταβλητές, την `meqconstr` και την `mpofveq`. Με την `meqconst` ορίζουμε το εάν υπάρχουν περιορισμοί ισότητας στο πρόβλημα, με τιμή «1» για «να», και «2» για «όχι», ενώ με την `mpofveq` ορίζουμε το πλήθος αυτών των περιορισμών. Για προβλήματα που έχουν έως έναν περιορισμό ισότητας, παρατηρήθηκε ότι μπορεί να βρεθεί λύση και με τις δύο περιπτώσεις για την μεταβλητή `meqconstr`, δηλαδή και για τιμή «1» αλλά και για τιμή «2». Στην μεταβλητή `mpofveq` θέταμε πάντα τον ορθό αριθμό του πλήθους των περιορισμών ισότητας, αφού για `meqconstr=2` δεν λαμβάνεται υπόψη, στον κώδικα, η τιμή της `mpofveq`. Έτσι ελέγχουμε, στη «φάση ένα» που περιγράφουμε παρακάτω, και για τους δύο συνδυασμούς, σε προβλήματα με έναν περιορισμό ισότητας.

## **5.2 Πειραματική διαδικασία για την SL-PSO**

Αρχικά να τονίσουμε ότι από τη θεωρητική εφαρμογή του αλγορίθμου, από τα πρώτα "ελεύθερα» πειράματα, αλλά και από όλα τα παρακάτω πειράματα, φάνηκε ότι υπάρχει μεγάλη αλληλεξάρτηση των παραμέτρων μεταξύ τους, αλλά και με την τιμή της αντικειμενικής συνάρτησης. Συνεπώς δεν μπορούμε να κάνουμε έλεγχο της καθεμίας ξεχωριστά.

### ➤ **Φάση 1<sup>η</sup>**

Έτσι, σε πρώτη φάση, δημιουργήσαμε μια αυτοματοποιημένη πειραματική διαδικασία υποθέτοντας κάποιες πιθανές τιμές για κάθε μία από τις τρεις βασικές μεταβλητές και στη συνέχεια εξετάζαμε όλους τους πιθανούς συνδυασμούς μεταξύ αυτών. Αυτή η διαδικασία εφαρμοζόταν για 4 κύκλους του κώδικα («τρεξίματα») σε κάθε συνδυασμό, κρατώντας σε αρχείο τη βέλτιστη τιμή της αντικειμενικής συνάρτησης και τη συνολική τιμή των παραβιάσεων των περιορισμών για κάθε κύκλο. Οι αρχικές τιμές που υποθέσαμε είναι:

$\lambda = 0.2, 1, 10, 100, 1000, 10000, 50000$

`initstd = 0.1, 1, 10, 100`

`istreductratefinal = 1, 2, 3, 4, 5, 6`

Σε περίπτωση που αυτή η αρχική υπόθεση δεν οδηγούσε σε εφικτές λύσεις και σε λύσεις κοντά, από άποψη κλίμακας μεγέθους, στη βέλτιστη από τη βιβλιογραφία, μεγαλώναμε αυτά τα εύρη των τιμών έως ότου βρούμε καλύτερες λύσεις, ή παρατηρήσουμε ότι η λύση απομακρυνόταν περισσότερο από τις αρχικές που βρήκαμε.

		λ						
initstd	0.2	1	10	100	1000	10000	50000	
0.1	1E+23	-9.99467E-06	-9.93447E-06	-9.43921E-06	14.01528	5.18E-05	0.0004	
	1E+23	-9.9953E-06	-9.9672E-06	-9.57011E-06	14.02186	3.29E-05	0.000324	
	1E+23	31.98966858	-9.94618E-06	-9.43259E-06	-2.8E-06	5.77E-05	0.000408	
	1E+23	-9.99378E-06	-9.94598E-06	-9.34467E-06	0.989362	3.27E-05	0.000468	
1	1E+23	-9.9284E-06	-9.36315E-06	-2.96352E-06	6.23E-05	0.001031	0.004002	
	1E+23	-9.95225E-06	-9.58562E-06	-5.12033E-06	3.3E-05	0.000577	0.003499	
	1E+23	-9.93237E-06	-9.25875E-06	16.18499529	6.12E-05	0.001052	0.006044	
	1E+23	-9.93987E-06	14.00934946	-2.87045E-06	5.41E-05	0.000964	0.006354	
10	-1.00554E-05	-9.22456E-06	-4.78704E-06	5.87769E-05	0.000813	0.015277	1E+23	
	-1.00195E-05	-9.50487E-06	-4.78704E-06	3.69635E-05	0.000559	0.007395	1E+23	
	-1.05899E-05	-9.29649E-06	-3.35123E-06	6.45819E-05	0.00119	0.01069	1E+23	
	-2.41354E-05	-9.40044E-06	-5.1468E-06	5.27927E-05	0.001057	0.010583	1E+23	

Σχήμα 5.1 Πίνακας δοκιμών της φάσης 1<sup>ης</sup>

Παράδειγμα από ένα πρόβλημα που εξετάσαμε, μέρος μίας σελίδας για μία από τις τιμές της *istreductratefinal*. Αντίστοιχα αρχεία δημιουργήσαμε και για τη συνολική τιμή των παραβιάσεων των περιορισμών για κάθε αντίστοιχο κύκλο.

## ➤ Φάση 2<sup>η</sup>

Στη συνέχεια σημειώναμε, στο αρχείο των πιθανών λύσεων, τις καλύτερες λύσεις (τιμή αντικειμενικής συνάρτησης) κοιτώντας αρχικά για κάθε *istreductratefinal* και μετά για το συνδυασμό των  $\lambda$  και *initstd*. Αυτό έγινε με αυτή τη σειρά γιατί φάνηκε, από τα πρώτα πειράματα, ότι τα όρια των τιμών της πρώτης ήταν πιο ευδιάκριτα από τις τιμές των άλλων δύο παραμέτρων. Επίσης σε περίπτωση που είχαμε πολλούς συνδυασμούς με πολύ κοντινές λύσεις, προσπαθούσαμε να αποφεύγουμε: 1) τους συνδυασμούς παραμέτρων που είχαν μερική αποτυχία εύρεσης εφικτής λύσης, δηλαδή από τους τέσσερις κύκλους ο ένας είχε αποτύχει να βρει εφικτή λύση (1E+23), και 2) αυτούς που είχαν μικρές αλλά μόνιμες παραβιάσεις των περιορισμών.

Σε προβλήματα που δεν είχαν συνδυασμούς με ξεκάθαρα καλή λύση ή με μία, ή περισσότερες, από τις λύσεις για έναν συνδυασμό να μην είναι εφικτή (1E+23) ή να είναι πολύ μακριά από τις άλλες, σημειώναμε τους καλύτερους, εν μέρει, συνδυασμούς. Δηλαδή, σημειώναμε ακόμα και αυτούς που μόνο μία λύση από τους 4 κύκλους ήταν εφικτή και είχε κοντινή τιμή στη βέλτιστη από την βιβλιογραφία.

INITStD	0.1	0.1	1	1	10	10	10
ISTDREDU	5	5	5	5	5	5	5
pa	5	5	5	5	5	5	5
elfactor	50	50	50	50	50	50	50
ITERMAX	2000	2000	2000	2000	2000	2000	2000
MEQCONS	1	1	1	1	1	1	1
MNofVeq	4	4	4	4	4	4	4
lambda	100	1000	1	100	0.2	0.5	1
object	1E+23	2.44E-06	22.97909	4.55E-06	-1E-05	-9.3E-06	-8.4E-06
2	1E+23	1.64E-06	22.663	1.52E-06	-1E-05	-9.6E-06	-9.1E-06
3	-8.2E-06	1E+23	-9.8E-06	7.24E-06	-1.7E-05	-9.2E-06	-8.9E-06
4	-8.8E-06	0.671396	20.06719	4.48E-06	-9.9E-06	-9.4E-06	0.158604
5	1E+23	1E+23	-9.8E-06	7.17E-06	-1E-05	-9.2E-06	16.16501
6	-8.5E-06	1E+23	-9.9E-06	6.67E-06	-1E-05	-9.2E-06	-8.6E-06
7	1E+23	1E+23	-9.8E-06	6.17E-06	-1.1E-05	-9.4E-06	-8.6E-06
8	14.00887	0.674217	-9.8E-06	8.27E-06	-1E-05	-9.4E-06	-8.4E-06
9	-8.3E-06	3.04E-06	-9.9E-06	1E+23	-1E-05	-9.5E-06	-8.4E-06
10	-8.4E-06	1E+23	-9.9E-06	6.13E-06	-1E-05	-9.4E-06	-8.3E-06
11	-7.9E-06	6.24E-06	-9.8E-06	5.55E-06	-1E-05	-9.4E-06	-8.3E-06
12	1E+23	8.18E-06	9.658297	5.42E-06	-1.5E-05	-9.5E-06	-8.6E-06
13	1E+23	8.18E-06	-9.8E-06	1E+23	-1E-05	-9.2E-06	-8.2E-06
14	1E+23	6.62E-06	-9.8E-06	2.55E-06	-2.5E-05	-9.4E-06	-8.8E-06
15	1E+23	4.28E-06	-9.8E-06	6.71E-06	-1.1E-05	-9.5E-06	-8.4E-06
16	1E+23	6.13E-06	-9.8E-06	0.98502	-9.9E-06	-9.5E-06	-8.7E-06
17	1E+23	3.07E-06	-9.8E-06	5.31E-06	-1E-05	-9.5E-06	-8.5E-06
18	1E+23	4.32E-06	-9.9E-06	0.665711	-9.9E-06	-9.5E-06	-8.7E-06
19	-8.2E-06	0.979741	-9.9E-06	1E+23	-3.1E-05	-9.3E-06	-9E-06
20	0.99146	1E+23	0.88358	6.73E-06	-1E-05	-9.5E-06	0.67756
21	-8.4E-06	5.95E-06	6.479968	8.12E-06	-1.3E-05	-9.4E-06	-8.3E-06
22	-8.8E-06	7.73E-06	7.268025	7.42E-06	-1E-05	-9.2E-06	0.738322
23	-8.4E-06	0.978903	-9.9E-06	6.23E-06	-1E-05	-9.3E-06	8.993207
24	1E+23	3.51E-06	-9.8E-06	16.18808	-1E-05	-9.3E-06	-8.2E-06
25	1E+23	5.29E-06	-9.9E-06	9.17E-06	-1.2E-05	-9.4E-06	-8.4E-06
median	1E+23	7.73E-06	-9.8E-06	6.73E-06	-1E-05	-9.4E-06	-8.4E-06
best	-8.8E-06	1.64E-06	-9.9E-06	1.52E-06	-3.1E-05	-9.6E-06	-9.1E-06
worst	1E+23	1E+23	22.97909	1E+23	-9.9E-06	-9.2E-06	16.16501
std	5.1E+22	4.36E+22	7.385875	3.32E+22	5.07E-06	1.14E-07	3.620714

Σχήμα 5.2 Πίνακας δοκιμών της 3<sup>ης</sup> φάσης

### ➤ Φάση 3<sup>η</sup>

Με βάση τους παραπάνω σημειωμένους συνδυασμούς, προχωρούσαμε στα αναλυτικά πειράματα, δηλαδή για 25 κύκλους («τρεξίματα») τη φορά. Ακόμα, για να δούμε τη συμπεριφορά ανάμεσα στις αρχικές τιμές που είχαμε ορίσει στις παραμέτρους  $\lambda$ ,  $initstd$  και  $istreductratefinal$  προχωρήσαμε και σε ένα δεύτερο έλεγχο κοντά στους παραπάνω σημειωμένους συνδυασμούς, κυρίως για την παράμετρο  $\lambda$  γιατί είχε πολύ μεγάλα εύρη στις τιμές της, για να εντοπίσουμε τυχόν μεγάλες βελτιώσεις στην τιμή της λύσης.

### ➤ Φάση 4<sup>η</sup>

Στη συνέχεια συγκρίναμε τις παραπάνω αναλυτικές λύσεις που βρήκαμε, σημειώναμε τις καλύτερες δυνατές για κάθε πρόβλημα και συγκρίναμε με τις λύσεις που βρέθηκαν από άλλους αλγόριθμους από τη βιβλιογραφία. Επίσης, καταγράφαμε τα διάφορα συμπεράσματα που εντοπίστηκαν κατά την πειραματική διαδικασία.

### 5.3 Αποτελέσματα προβλημάτων CEC-2010

Σε αυτό το υποκεφάλαιο θα αναφέρουμε τα αποτελέσματα που βρήκαμε από την πειραματική διαδικασία, και θα αναλύσουμε τον τρόπο που συμπεριφέρεται η SL-PSO σε καθένα από αυτά. Επίσης, θα συγκρίνουμε τις λύσεις που βρεθήκαν με τη βέλτιστη που έχει βρεθεί από άλλους κώδικες (Kusakci & Can, 2013) (Eita et al., 2014).

#### 5.3.1 Πρόβλημα 1

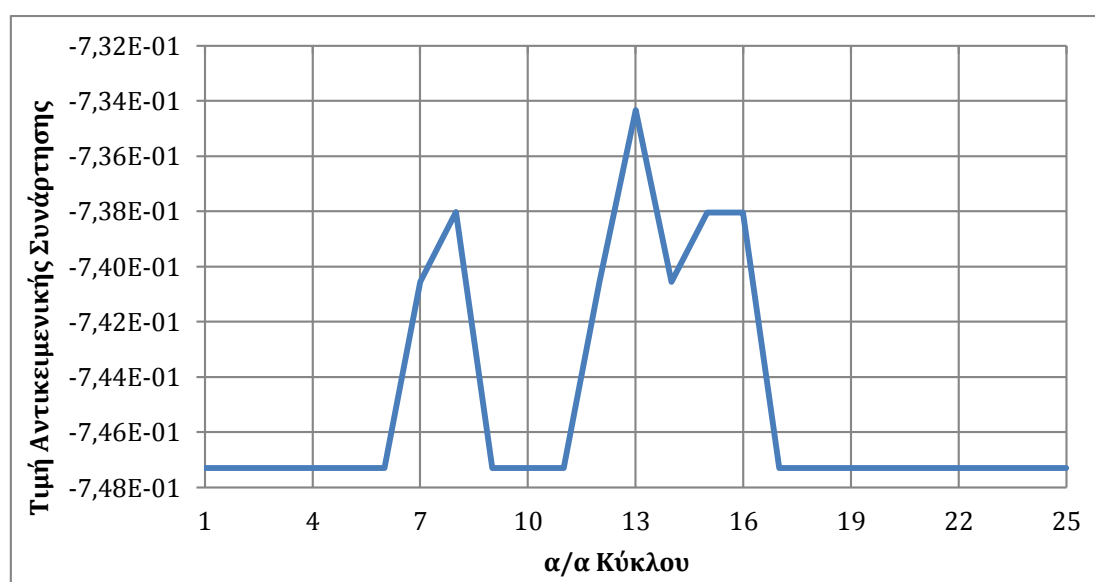
Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=0.1$ ,  $istreductratefinal=3$  και  $\lambda=10000$ , με τιμή:

$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-0.7473093	-0.7473099	-0.7343403	0.0041526
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

Πίνακας 5.1 Αποτελέσματα Προβλήματος 1 για 10 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): -0.7473104.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.3 Διάγραμμα Προβλήματος 1 για 10 διαστάσεις

Παρατηρούμε ότι οι λύσεις που βρήκαμε είναι πολύ κοντά στη βέλτιστη που έχει βρεθεί για το πρόβλημα αυτό. Επίσης δεν εντοπίζουμε καμία παραβίαση των περιορισμών, αλλά ούτε και κάποια μη-εφικτή λύση ή λύση που να φαίνεται έντονα ότι έχει παγιδευτεί σε τοπικό ακρότατο.

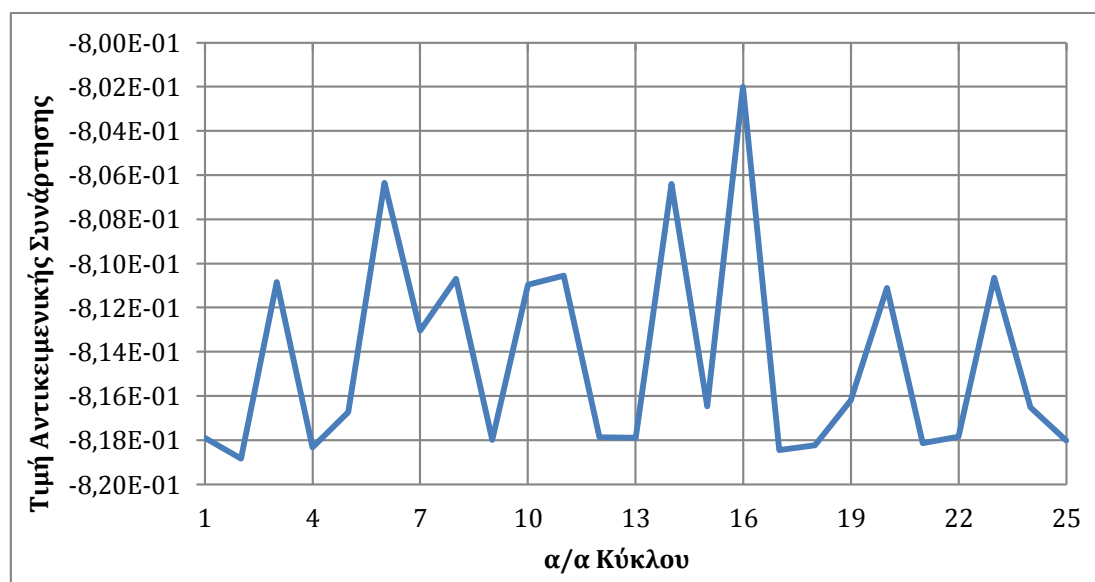
Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=100$ ,  $istreductratefinal=3$  και  $\lambda=1000$ , με τιμή:

$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-0.8165114	-0.8188281	-0.8020094	0.0047158
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

**Πίνακας 5.2** Αποτελέσματα Προβλήματος 1 για 30 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (CGCO): -0.8218384.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.4** Διάγραμμα Προβλήματος 1 για 30 διαστάσεις

Και για τις 30 διαστάσεις δεν εντοπίζουμε παραβιάσεις, και η καλύτερη λύση που βρήκαμε είναι αρκετά κοντά στη βέλτιστη που έχει βρεθεί.

Στις γενικές παρατηρήσεις, στο πρόβλημα αυτό βρέθηκαν πολύ συνδυασμοί λύσεων, με μηδενικές παραβιάσεις περιορισμών και λύσεις πολύ κοντά στις βέλτιστες που

έχουν βρεθεί. Ακόμα, κατά την πειραματική διαδικασία, αλλά και από τη «Φάση 1», φάνηκε έντονα η αλληλεξάρτηση των παραμέτρων  $initstd$ ,  $istreductratefinal$  και  $\lambda$ , καθώς με την αύξηση των δύο πρώτων έπρεπε να μειώσουμε την άλλη, για να εντοπίσουμε παρεμφερείς τιμές λύσεων. Αυτό συνέβη και για τις δύο περιπτώσεις των διαστάσεων της λύσης.

### 5.3.2 Πρόβλημα 2

Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=10$ ,  $istreductratefinal=3$  και  $\lambda=1$ , με τιμή:

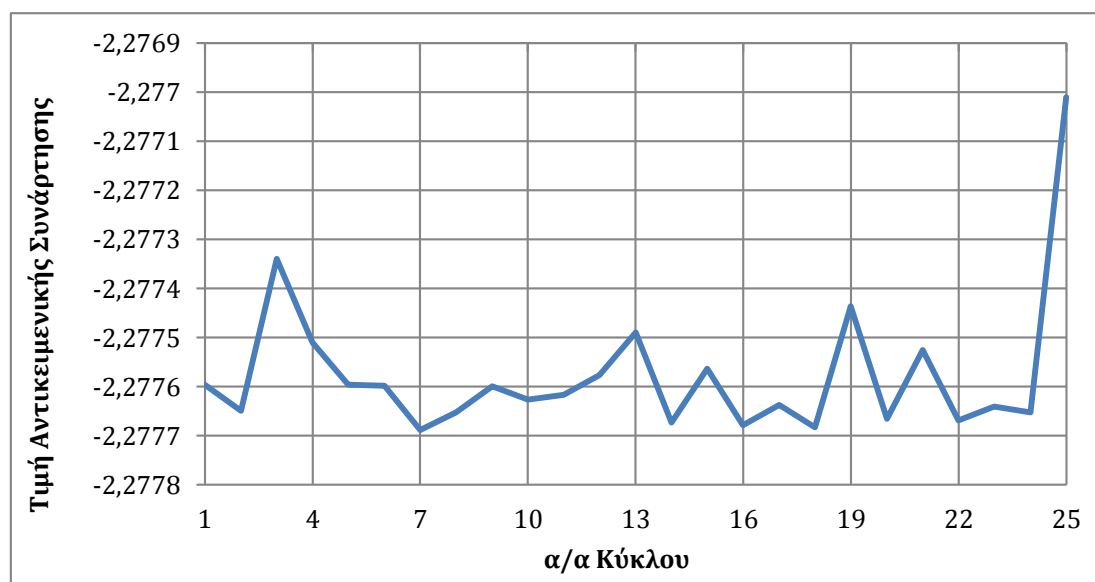
$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-2.277617	-2.277688	-2.277011	0.000145
Συνολική τιμή παραβιάσεων περιορισμών	0	0	1.87E-05	3.75E-06

Πίνακας 5.3 Αποτελέσματα Προβλήματος 2 για 10 διαστάσεις

Εδώ βρέθηκαν 2 (από 25) λύσεις που περιείχαν μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (CGCO): -2.277704.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.5 Διάγραμμα Προβλήματος 2 για 10 διαστάσεις

Παρατηρούμε ότι οι λύσεις που βρήκαμε είναι πολύ κοντά στη βέλτιστη που έχει βρεθεί για το πρόβλημα αυτό. Επίσης δεν εντοπίζουμε καμία σημαντική παραβίαση των περιορισμών, αλλά ούτε και κάποια μη-εφικτή λύση ή λύση που να φαίνεται έντονα ότι έχει παγιδευτεί σε τοπικό ακρότατο.

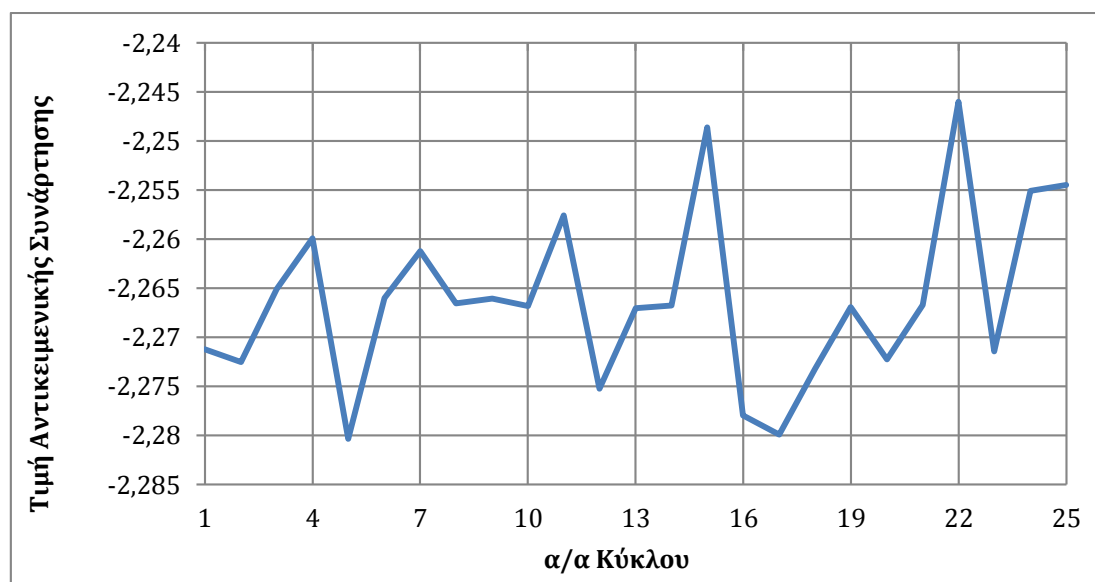
Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $\text{initstd}=10$ ,  $\text{istreductratefinal}=3$  και  $\lambda=1$ , με τιμή:

$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-2.266801	-2.280380	-2.246046	0.008913
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

**Πίνακας 5.4** Αποτελέσματα Προβλήματος 2 για 30 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (Co-CLPSO): -2.2809.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.6** Διάγραμμα Προβλήματος 2 για 30 διαστάσεις

Και για τις 30 διαστάσεις δεν εντοπίζουμε παραβιάσεις, και η καλύτερη λύση που βρήκαμε είναι αρκετά κοντά στη βέλτιστη που έχει βρεθεί.

Στις γενικές παρατηρήσεις, στο πρόβλημα αυτό βρέθηκαν πολύ συνδυασμοί παραμέτρων με λύσεις κοντά στις βέλτιστες. Παρόλα αυτά εντοπίστηκαν αρκετές



παραβιάσεις μικρού μεγέθους σε ένα μεγάλο μέρος του πλήθους των συνδυασμών. Αυτό συνέβη και για τις δύο περιπτώσεις των διαστάσεων της λύσης.

### 5.3.3 Πρόβλημα 3

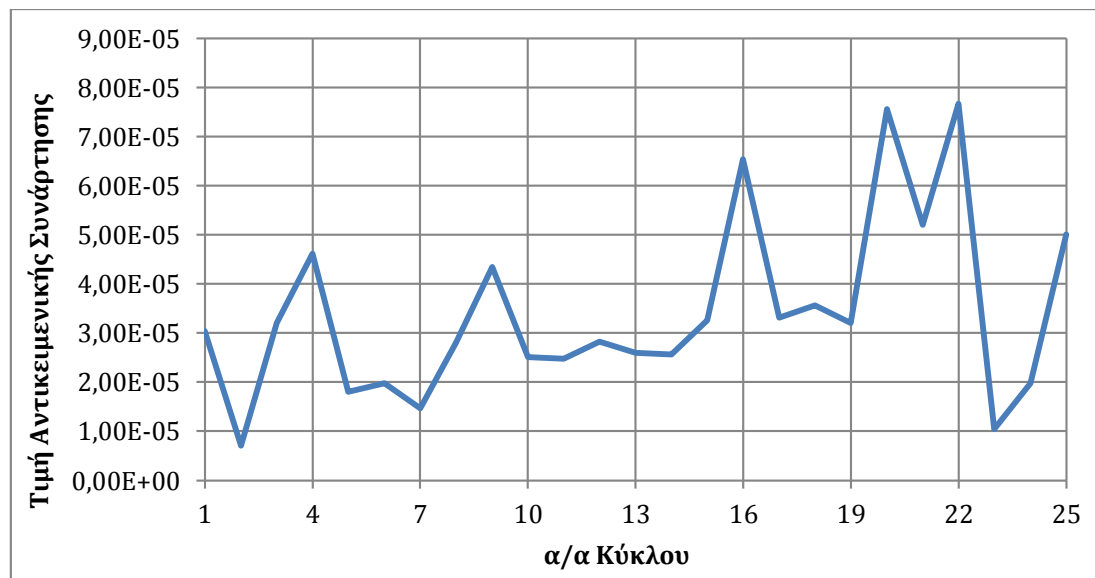
Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=100$ ,  $istreductratefinal=6$  και  $\lambda=10$ , με τιμή:

$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	3.05E-05	7.03E-06	7.67E-05	1.84E-05
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

Πίνακας 5.5 Αποτελέσματα Προβλήματος 3 για 10 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 0.000E+00.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.7 Διάγραμμα Προβλήματος 3 για 10 διαστάσεις

Οι συνδυασμοί των παραμέτρων που βρέθηκαν σε αυτό το πρόβλημα, που δίνουν λύσεις αρκετά καλές και τείνουν στη βέλτιστη, για τις 10 διαστάσεις, είναι πολύ λιγότεροι σε σχέση με τα δύο προηγούμενα προβλήματα. Επίσης είναι πολύ πιο ευδιάκριτο να εντοπίσουμε παγιδεύσεις σε τοπικά ακρότατα, καθώς αυτό συνέβη σε

μεγάλο πλήθος συνδυασμών. Παρόλα αυτά, εντοπίστηκαν συνδυασμοί που μπόρεσαν να «πηδήσουν» αυτό το ακρότατο. Ακόμα, φάνηκε μία τάση της παραμέτρου  $\lambda$  να δίνει καλές λύσεις για τιμές της κοντά στο 10, ανεξάρτητα των άλλων δύο παραμέτρων, και τάση της  $initstd$  να βελτιώνει τη λύση όσο μικραίνει η τιμή της, αυξάνοντας ταυτόχρονα τον κίνδυνο παγίδευσης σε τοπικό ακρότατο.

Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=0.1$ ,  $istreductratefinal=2$  και  $\lambda=2000$ , αλλά και για  $meqconst=2$ , με τιμή:

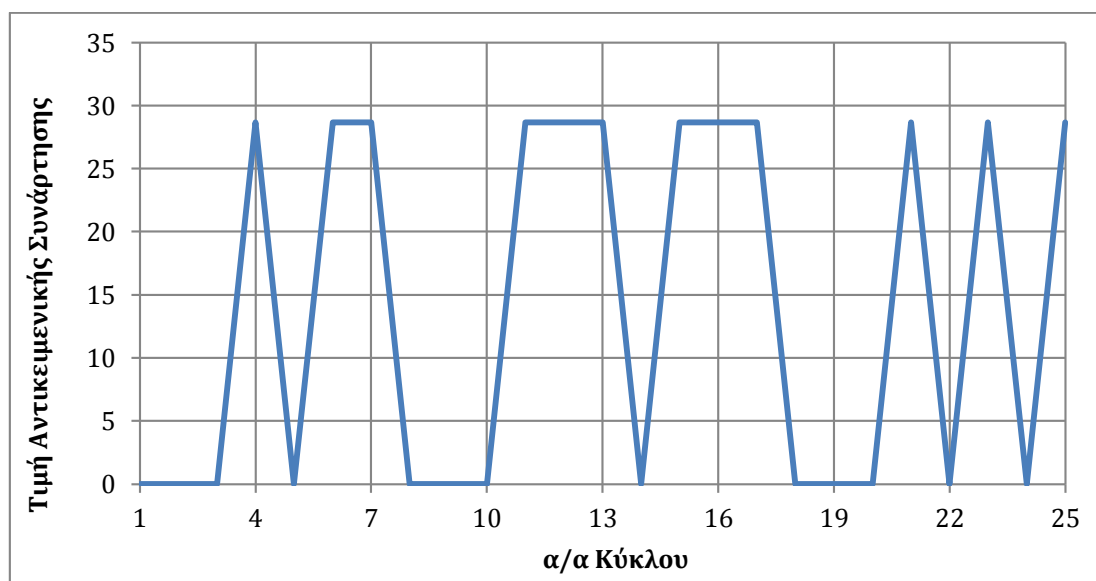
$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	0.006852	0.004005	28.68067	14.62111
Συνολική τιμή παραβιάσεων περιορισμών	0	0	2.38E-06	6.5E-07

**Πίνακας 5.6** Αποτελέσματα Προβλήματος 3 για 30 διαστάσεις

Εδώ βρέθηκαν 5 (από 25) λύσεις που περιείχαν μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (CGCO): 2.54051E-07.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.8** Διάγραμμα Προβλήματος 3 για 30 διαστάσεις

Εδώ πρέπει να τονίσουμε την έντονη τάση του αλγορίθμου να παγιδεύεται σε ένα συγκεκριμένο τοπικό ακρότατο. Επίσης, υπάρχουν λίγοι συνδυασμοί παραμέτρων που μας δίνουν λύση που να τείνει στη βέλτιστη και ταυτόχρονα να έχει τις λιγότερες δυνατές παγιδεύσεις στο ακρότατο αυτό.

### 5.3.4 Πρόβλημα 4

Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=10$ ,  $istreductratefinal=5$  και  $\lambda=0.5$ , με τιμή:

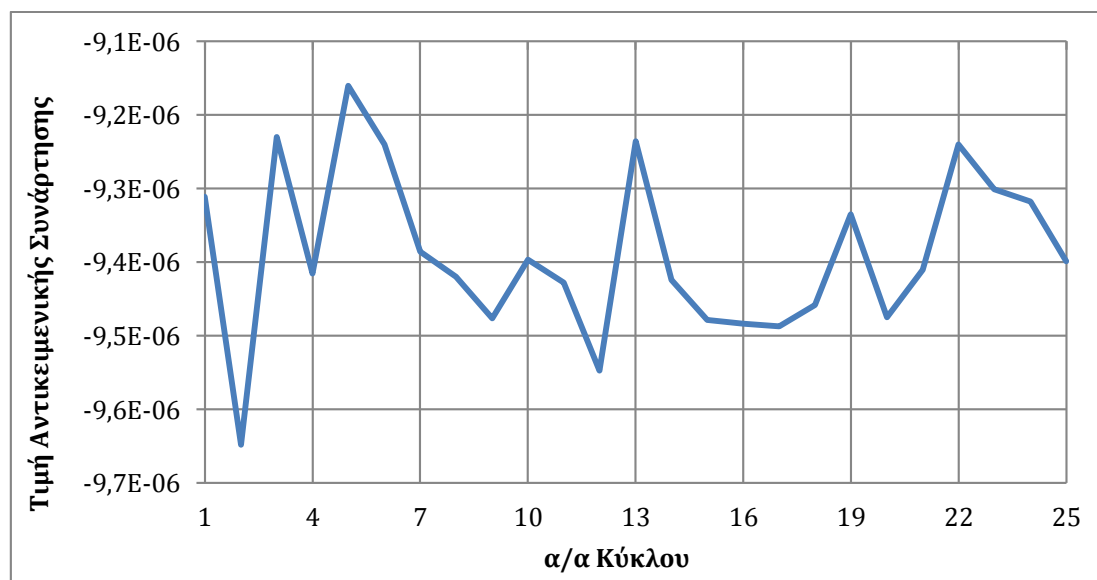
$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-9.4E-06	-9.6E-06	-9.2E-06	1.14E-07
Συνολική τιμή παραβιάσεων περιορισμών	7.69E-07	0	2.19E-06	7.01E-07

Πίνακας 5.7 Αποτελέσματα Προβλήματος 4 για 10 διαστάσεις

Εδώ βρέθηκαν 18 (από 25) λύσεις που περιείχαν μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (Co-CLPSO): -1.000E-05.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.9 Διάγραμμα Προβλήματος 3 για 30 διαστάσεις

Βρέθηκαν αρκετοί συνδυασμοί παραμέτρων που δίνουν λύσεις κοντά στη βέλτιστη, που έχει βρεθεί, με μερικούς να δίνουν καλύτερη από αυτή που αναφέραμε πιο πάνω. Παρόλα αυτά, οι πιο πολύ έχουν προβλήματα με μη-εφικτές λύσεις, πολλές παραβιάσεις περιορισμών και παγιδεύσεις σε τοπικά ακρότατα.

Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=2000$ ,  $istreductratefinal=6$  και  $\lambda=1$ , με τιμή:

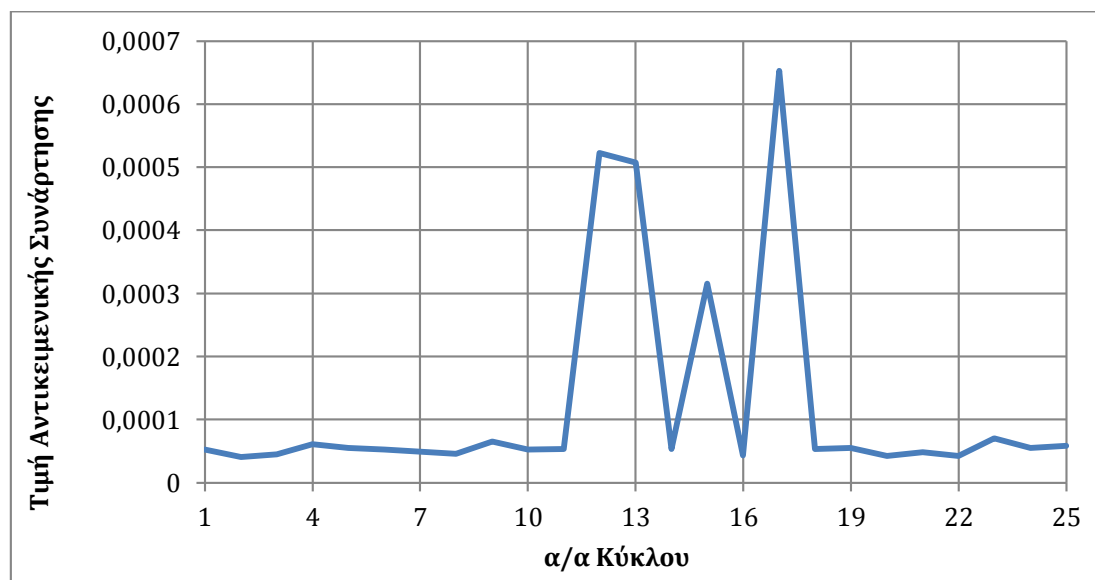
$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	5.37604E-05	4.11132E-05	0.000653106	0.000174547
Συνολική τιμή παραβιάσεων περιορισμών	0	0	3.46439E-06	6.92878E-07

**Πίνακας 5.8** Αποτελέσματα Προβλήματος 4 για 30 διαστάσεις

Εδώ βρέθηκε  $\underline{1}$  (από 25) λύση που περιείχε μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί από άλλους κώδικες (CGCO) :  $\underline{-1.77753E-07}$ .

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.10** Διάγραμμα Προβλήματος 4 για 30 διαστάσεις

Για 30 διαστάσεις, στο πρόβλημα 4, δεν βρέθηκε δυνατός συνδυασμός που να δίνει αρνητική λύση για αντικειμενική συνάρτηση. Επίσης, οι πιο πολύ συνδυασμοί των

παραμέτρων, που βρέθηκαν, μας έδιναν μεγάλο ποσοστό μη-εφικτών λύσεων και πολλές παραβιάσεις περιορισμών. Όσον αφορά τις παραμέτρους, η πλειονότητα των καλύτερων λύσεων βρέθηκε για  $\lambda$  γύρω από την τιμή 1, ενώ όσο μικραίναμε την τιμή του `initstd` οι λύσεις που βρίσκαμε ήταν καλύτερες αλλά ασταθείς, με μεγάλο ποσοστό μη-εφικτών. Ακόμα, όσο μικραίναμε το `istreductratefinal` τόσο «έκλειναν» οι περιοχές των εφικτών λύσεων γύρω από την τιμή  $\lambda=1$ .

### 5.3.5 Πρόβλημα 5

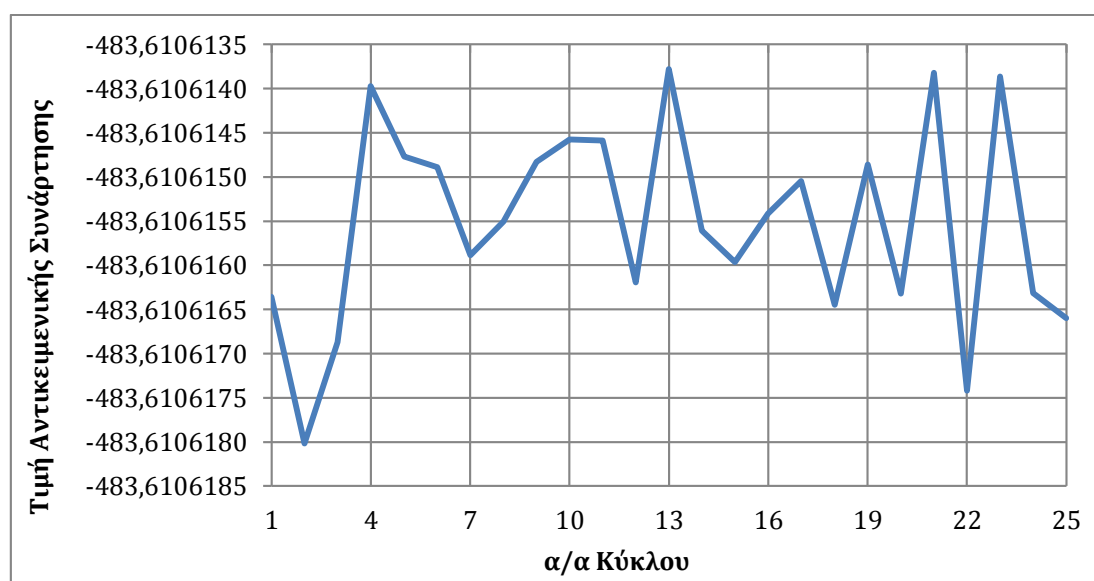
Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για `initstd=100`, `istreductratefinal=6` και  $\lambda=1$ , με τιμή:

$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-483.61062	-483.61062	-483.61061	1.14E-06
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

Πίνακας 5.9 Αποτελέσματα Προβλήματος 5 για 10 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί από άλλους κώδικες (εDEag) : -4.836106E+02.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.11 Διάγραμμα Προβλήματος 5 για 10 διαστάσεις

Βρέθηκαν πολύ συνδυασμοί παραμέτρων με λύσεις πολύ κοντά στη βέλτιστη, που έχει βρεθεί, και χωρίς παραβιάσεις περιορισμών. Οι καλύτερες λύσεις βρέθηκαν κοντά στην τιμή  $\lambda=1$  και λίγο πάνω από αυτήν.

Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $\text{initstd}=1000$ ,  $\text{istreductratefinal}=5$  και  $\lambda=1$ , με τιμή:

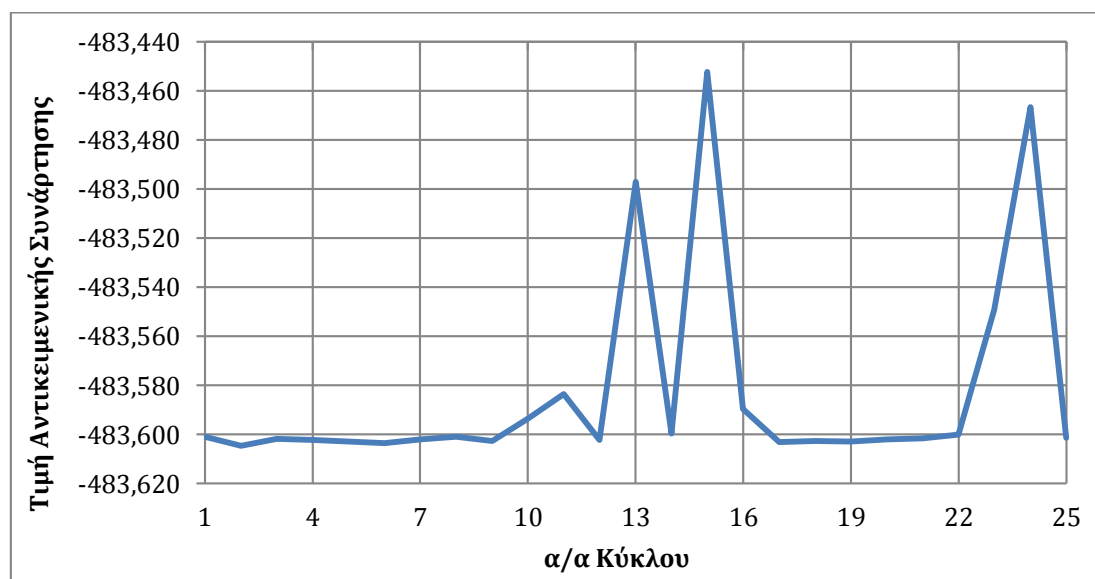
$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-483.60155	-483.60456	-483.45220	4.37E-02
Συνολική τιμή παραβιάσεων περιορισμών	0.000146	0	0.0013759	0.0003925

**Πίνακας 5.10** Αποτελέσματα Προβλήματος 5 για 30 διαστάσεις

Εδώ βρέθηκαν 19 (από 25) λύσεις που περιέχουν μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί από άλλους κώδικες (Co-CLPSO) : -4.83600E+02.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.12** Διάγραμμα Προβλήματος 5 για 30 διαστάσεις

Οι συνδυασμοί των παραμέτρων που μας έδωσαν καλές λύσεις ήταν αρκετοί, είχαμε όμως μεγάλο αριθμό πλήθους παραβιάσεων περιορισμών. Οι καλύτερες λύσεις

βρέθηκαν, όπως και για 10 διαστάσεις, για τιμές κοντά στην  $\lambda=1$  και λίγο πάνω από αυτήν.

### 5.3.6 Πρόβλημα 6

Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $\text{initstd}=100$ ,  $\text{istreductratefinal}=5$  και  $\lambda=1$ , με τιμή:

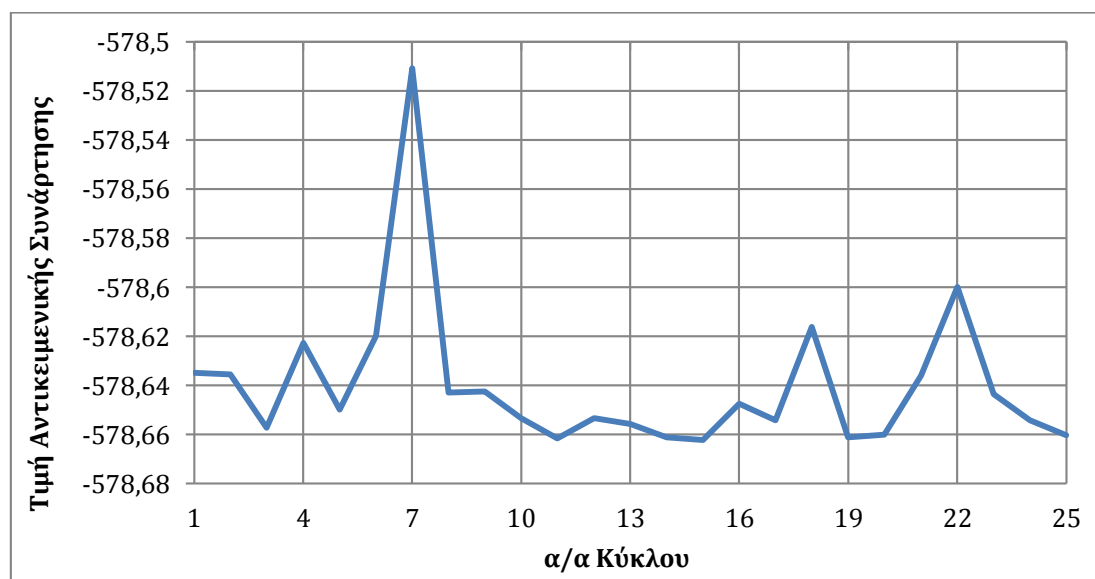
$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-578.649829	-578.662166	-578.510783	0.031476716
Συνολική τιμή παραβιάσεων περιορισμών	1.03962E-05	0	9.09398E-05	2.73169E-05

Πίνακας 5.11 Αποτελέσματα Προβλήματος 6 για 10 διαστάσεις

Εδώ βρέθηκαν 14(από 25) λύσεις που περιείχαν μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (Co-CLPSO): -5.7866E+02.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.13 Διάγραμμα Προβλήματος 6 για 10 διαστάσεις

Οι λύσεις που βρέθηκαν ήταν αρκετά κοντά στη βέλτιστη για πολλούς συνδυασμούς παραμέτρων, έχοντας όμως, σε μεγάλο μέρος αυτών, μικρές τιμές παραβίασης

περιορισμών. Επίσης παρατηρήθηκε ότι οι καλύτερες τιμές της αντικειμενικής συνάρτησης εντοπίστηκαν γύρω από την παράμετρο  $\lambda=1$  και λίγο πάνω από αυτήν.

Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $\text{initstd}=1000$ ,  $\text{istreductratefinal}=7$  και  $\lambda=1$ , με τιμή:

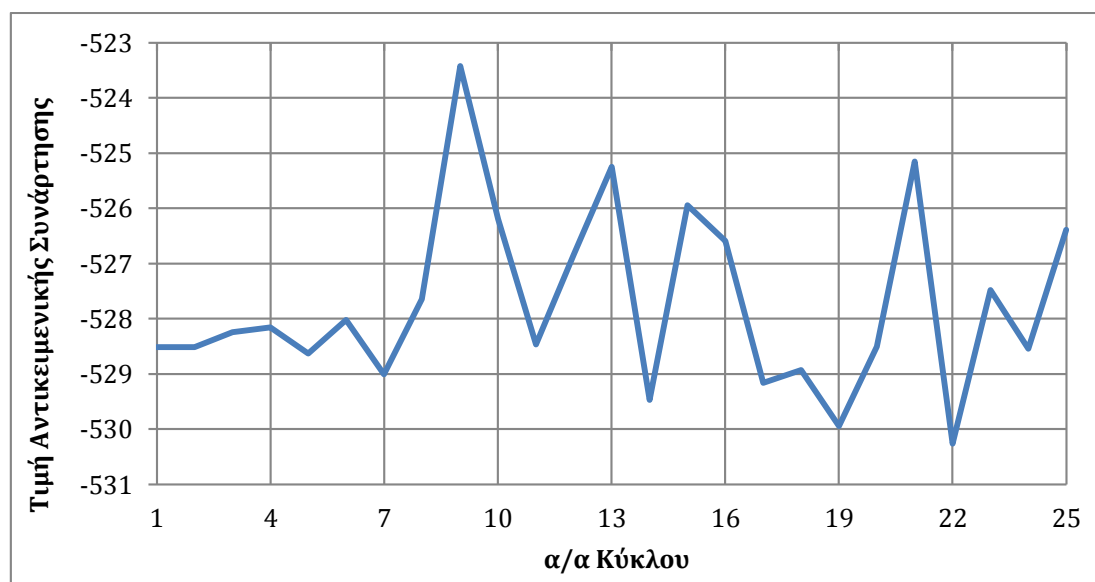
$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	-528.246229	-530.260983	-523.424802	1.640707
Συνολική τιμή παραβιάσεων περιορισμών	0.000146	0	1.5277E-06	4.6683E-07

**Πίνακας 5.12** Αποτελέσματα Προβλήματος 6 για 30 διαστάσεις

Εδώ βρέθηκαν 10 (από 25) λύσεις που περιέχουν μία μικρή συνολική τιμή παραβίασης των περιορισμών.

Βέλτιστη τιμή που έχει βρεθεί από άλλους κώδικες (εDEag) : -5.28575E+02.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.14** Διάγραμμα Προβλήματος 6 για 30 διαστάσεις

Και για τις 30 διαστάσεις, οι συνδυασμοί των παραμέτρων που μας έδιναν καλές λύσεις ήταν αρκετοί, αλλά υπήρχαν αρκετές μικρές παραβιάσεις των περιορισμών. Οι καλύτερες λύσεις βρέθηκαν για  $\lambda=1$ , ενώ παρατηρήθηκε ότι μικραίνοντας την τιμή του  $\text{istreductratefinal}$  η τιμή της αντικειμενικής συνάρτησης βελτιωνόταν,



αυξάνοντας όμως ταυτόχρονα το πλήθος και τις τιμές των παραβιάσεων των περιορισμών.

### 5.3.7 Πρόβλημα 7

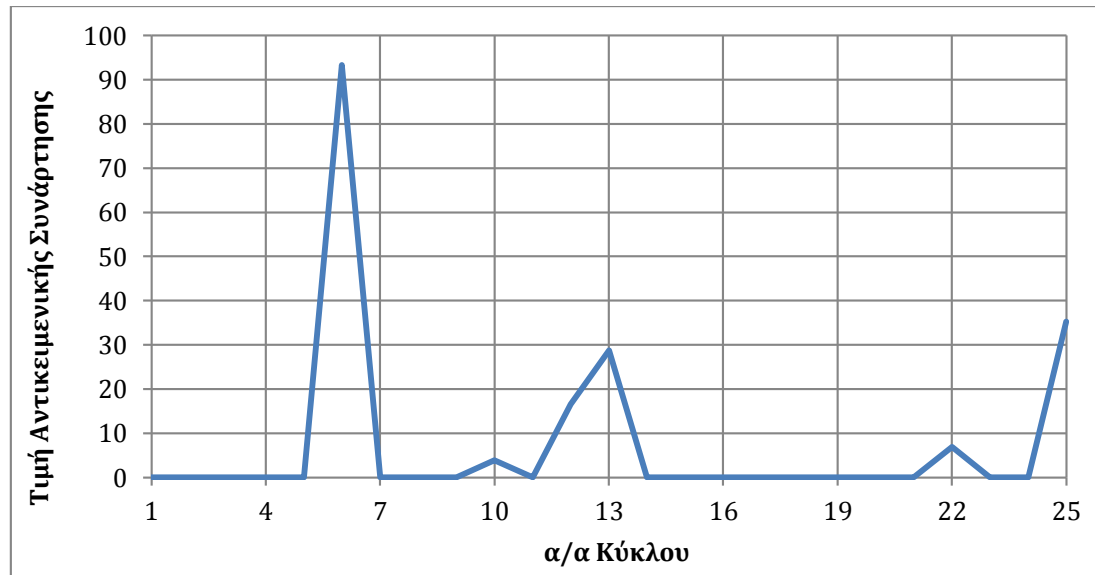
Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=10$ ,  $istreductratefinal=5$  και  $\lambda=10000$ , με τιμή:

$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	6.85095E-04	2.81698E-06	9.33161E+01	20.1564219
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

Πίνακας 5.13 Αποτελέσματα Προβλήματος 7 για 10 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 0.000E+00.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.15 Διάγραμμα Προβλήματος 7 για 10 διαστάσεις

Εδώ παρατηρούμε προβλήματα παγίδευσης σε διάφορα τοπικά ακρότατα. Τα προβλήματα αυτά εντοπίστηκαν στην πλειονότητα των συνδυασμών των παραμέτρων που μας έδιναν τις καλύτερες λύσεις, αλλά και γενικά σχεδόν σε όλους τους συνδυασμούς. Ακόμα παρατηρήθηκε ότι για μεγάλες τιμές  $initstd$  οι λύσεις

χειρότερου ανισθητά, αλλά είχαν λιγότερες παγιδεύσεις σε τοπικά ακρότατα, ενώ για μικρές τιμές  $initstd$  συνέβαινε το ακριβώς αντίθετο.

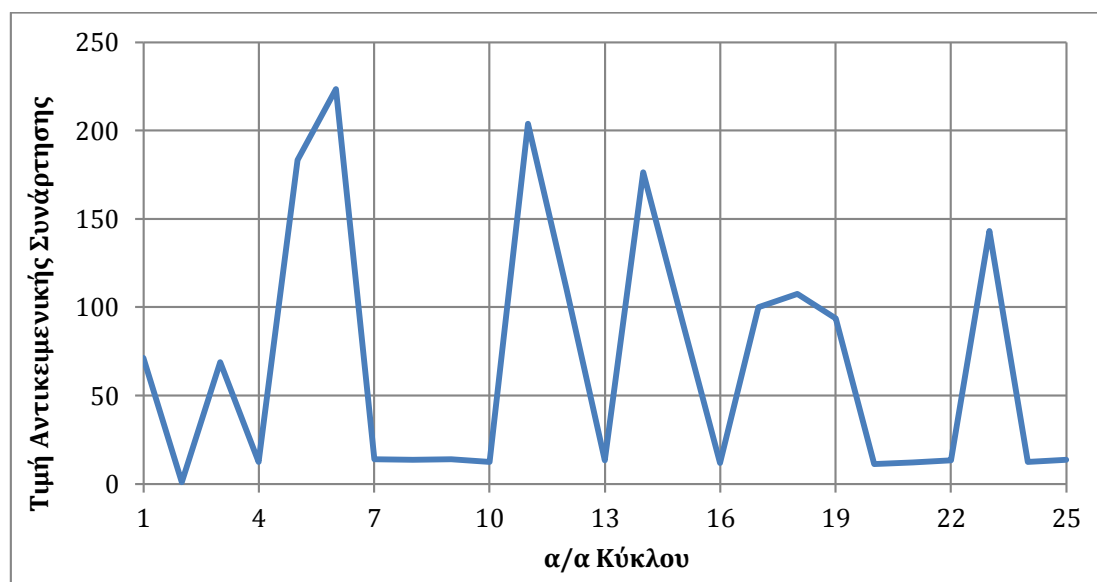
Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=100$ ,  $istreductratefinal=3$  και  $\lambda=100$ , με τιμή:

$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	14.07538	0.851543	223.5226	70.72434
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

**Πίνακας 5.14** Αποτελέσματα Προβλήματος 1 για 30 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 1.147112E-15.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



**Σχήμα 5.16** Διάγραμμα Προβλήματος 7 για 30 διαστάσεις

Εδώ τα προβλήματα παγίδευσης ήταν πολύ έντονα και όλες οι τιμές των λύσεων που βρέθηκαν απείχαν αρκετά από τη βέλτιστη.

### 5.3.8 Πρόβλημα 8

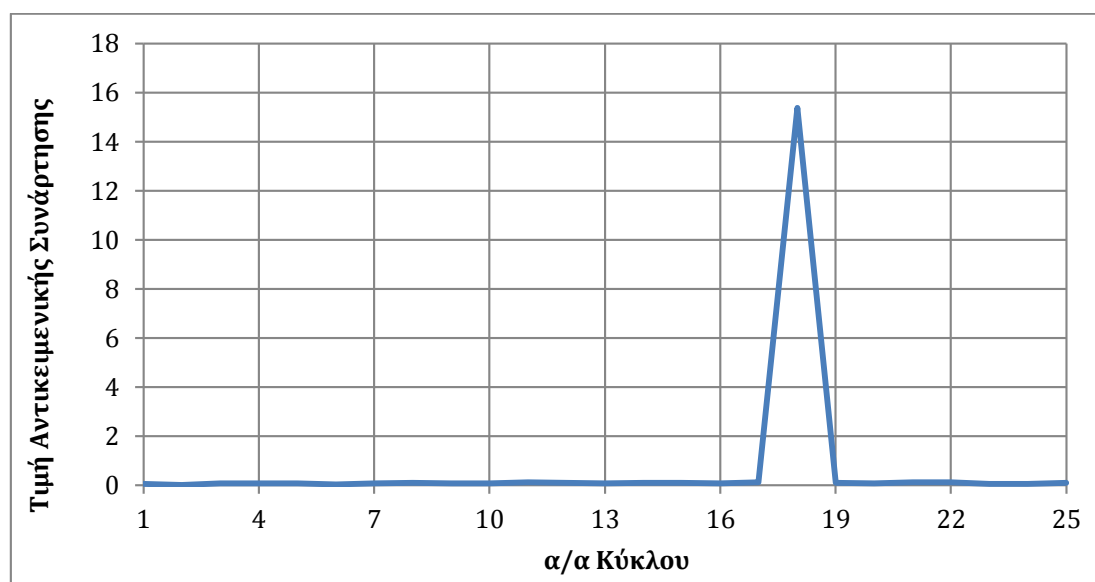
Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $initstd=1000$ ,  $istreductratefinal=2$  και  $\lambda=2$ , με τιμή:

$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	7.83004E-02	1.51441E-02	1.53731E+01	3.05939E+00
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

Πίνακας 5.15 Αποτελέσματα Προβλήματος 8 για 10 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 0.000E+00.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.17 Διάγραμμα Προβλήματος 8 για 10 διαστάσεις

Παρόλο που βρέθηκε ο παραπάνω συνδυασμός που είχε λίγο χειρότερη τιμή αντικειμενικής, από κάποιους άλλους που βρέθηκαν, αλλά είχε σχετικά σταθερή τιμή αντικειμενικής και πολύ λιγότερο αριθμό παγιδεύσεων σε ακρότατα., σε γενικές γραμμές, οι παγιδεύσεις σε τοπικά ακρότατα είναι πιο έντονες από το Πρόβλημα 7. Στην πλειονότητα των συνδυασμών, οι τιμές της αντικειμενικής που βρέθηκαν είναι πιο απομακρυσμένες από τη βέλτιστη που έχει βρεθεί. Αυτά τα δύο οφείλονται πιθανότατα στη συστροφή του πεδίου, με την εισαγωγή του πίνακα «M», καθώς είναι η μόνη αλλαγή που υπάρχει μεταξύ του Προβλήματος 7 και 8.

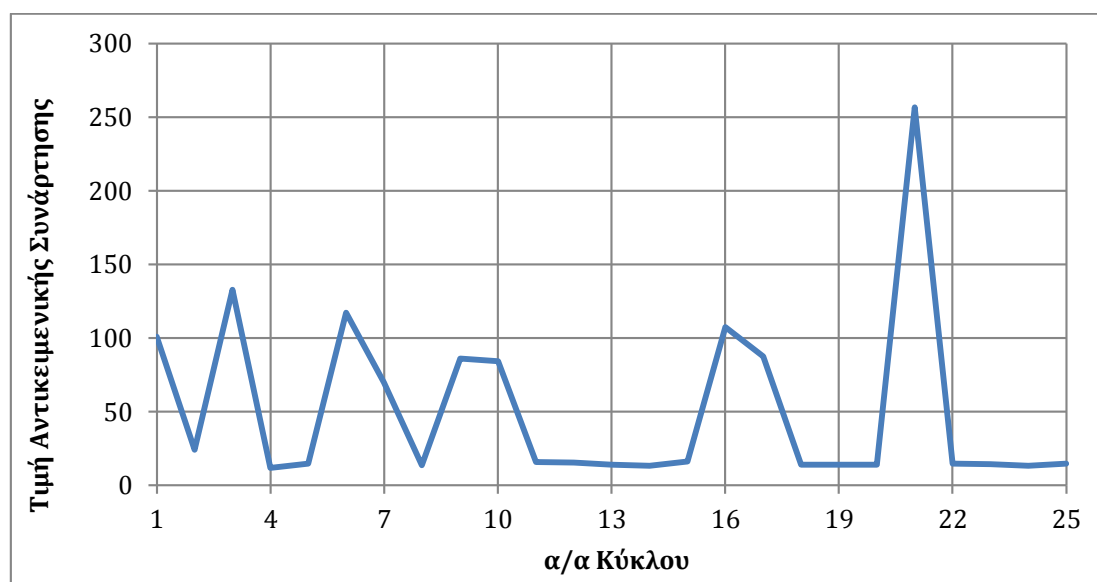
Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για initstd=200, istreductratefinal=2 και λ=10, με τιμή:

$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	15.33577	11.75561	256.72946	59.18842
Συνολική τιμή παραβιάσεων περιορισμών	0	0	0	0

Πίνακας 5.16 Αποτελέσματα Προβλήματος 8 για 30 διαστάσεις

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 2.518693E-14.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.18 Διάγραμμα Προβλήματος 8 για 30 διαστάσεις

Εδώ τα προβλήματα παγίδευσης σε τοπικά ακρότατα είναι πολύ έντονα, με ελάχιστους συνδυασμούς να έχουν πολύ λίγες λύσεις που να φαίνεται ότι δεν παγιδεύονται, και με τις καλύτερες τιμές της αντικειμενικής συνάρτησης, σε πολλές περιπτώσεις, να είναι πολύ μακριά από τη βέλτιστη. Σε σχέση με το Πρόβλημα 7, που είναι το ίδιο πρόβλημα χωρίς συστροφή του πεδίου των λύσεων, οι λύσεις που βρέθηκαν είναι λίγο ακόμα χειρότερες, κάτι που δείχνει τον έντονο επηρεασμό της εισαγωγής του πίνακα « $M$ ».

### 5.3.9 Πρόβλημα 9

Για 10 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $\text{initstd}=200$ ,  $\text{istreductratefinal}=5$  και  $\lambda=0.2$ , με τιμή:

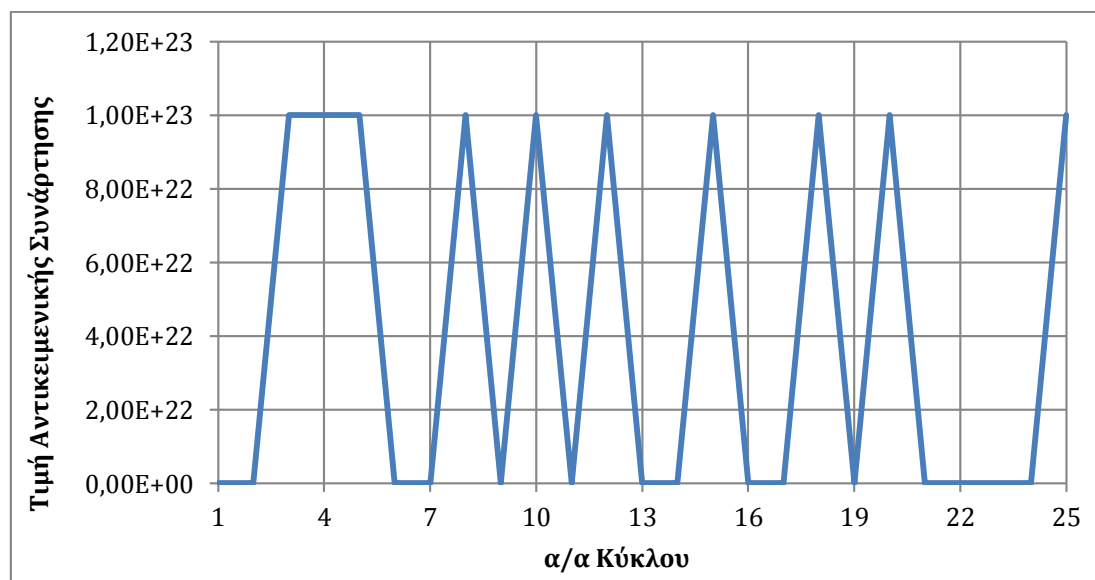
$su=10$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	4.204457E-06	2.645530E-06	5.590328E-06	7.118942E-07
Συνολική τιμή παραβιάσεων περιορισμών	0	0	2.072538E-06	5.351271E-07

Πίνακας 5.17 Αποτελέσματα Προβλήματος 9 για 10 διαστάσεις (χωρίς μη-εφικτές λύσεις)

Εδώ βρέθηκαν 10 (από 25) λύσεις που δεν ήταν εφικτές και 1 (από τις υπόλοιπες 15) λύση που περιείχε μία αμελητέα συνολική τιμή παραβίασης των περιορισμών. Οι μη-εφικτές λύσεις έχουν αφαιρεθεί από τον παραπάνω πίνακα.

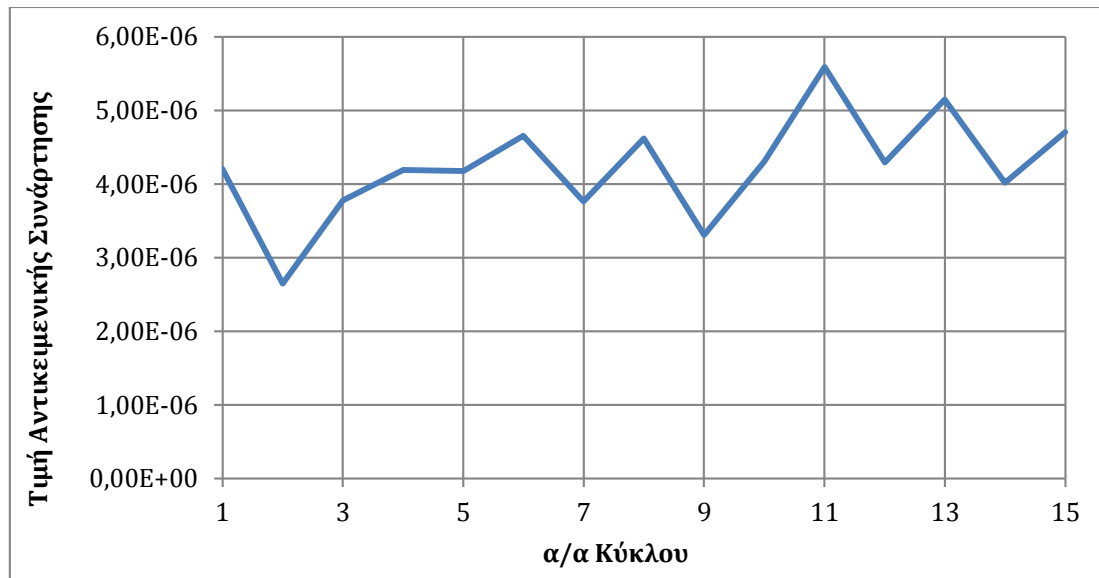
Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 0.000E+00.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι :



Σχήμα 5.19 Διάγραμμα Προβλήματος 9 για 10 διαστάσεις

Εάν αφαιρέσουμε τις μη-εφικτές λύσεις για να δούμε στατιστικά μόνο τις εφικτές :



**Σχήμα 5.20** Διάγραμμα Προβλήματος 9 για 10 διαστάσεις, χωρίς  $1.00E+23$  λύσεις

Σε αυτό το πρόβλημα, για τις 10 διαστάσεις, είχαμε μεγάλο πλήθος μη-εφικτών λύσεων σε όλο σχεδόν το εύρος των παραμέτρων. Επίσης οι καλύτερες τιμές βρέθηκαν για τιμή της  $\lambda$  κάτω από 1. Παρατηρήθηκε ότι για αυτές τις μικρές τιμές της  $\lambda$  είχαμε πολλές μη-εφικτές λύσεις αλλά οι υπόλοιπες ήταν κοντά στη βέλτιστη, που έχει βρεθεί, ενώ όσο αυξανόταν η  $\lambda$  τόσο χειροτέρευαν οι τιμές των λύσεων και άρχιζαν να εμφανίζονταν αρκετές παραβιάσεις στους περιορισμούς.

Για 30 διαστάσεις της λύσης της αντικειμενικής συνάρτησης, μια από τις καλύτερες τιμές βρέθηκε για  $\text{initstd}=1000$ ,  $\text{istreductratefinal}=6$  και  $\lambda=10$ , με τιμή:

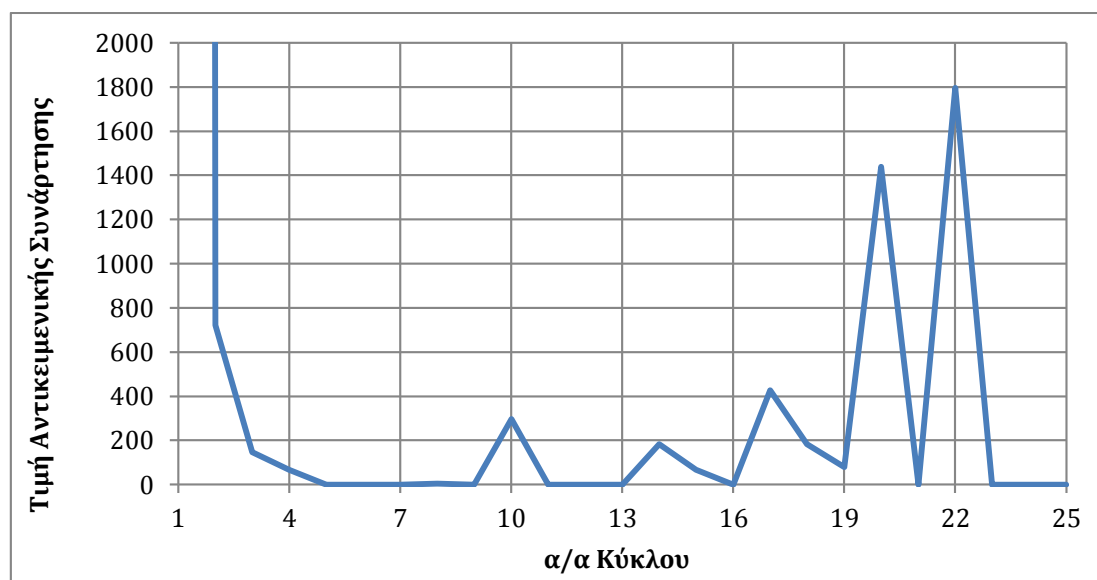
$su=30$	Διάμεση τιμή	Βέλτιστη τιμή	Χειρότερη τιμή	Τυπική Απόκλιση
Τιμή Αντικειμενικής Συνάρτησης	2.222217E+00	1.200358E-02	1.795479E+03	4.647065E+02
Συνολική τιμή παραβιάσεων περιορισμών	1.336018E-04	0	9.893639E-04	2.834809E-04

**Πίνακας 5.18** Αποτελέσματα Προβλήματος 9 για 30 διαστάσεις

Εδώ βρέθηκε 1 (από 25) λύση που δεν ήταν εφικτή και 20 (από τις υπόλοιπες 24) λύσεις που περιείχαν μία συνολική τιμή παραβίασης των περιορισμών. Οι μη-εφικτές λύσεις έχουν αφαιρεθεί από τον παραπάνω πίνακα.

Βέλτιστη τιμή που έχει βρεθεί, από άλλους κώδικες (εDEag): 2.770665E-16.

Η στατιστική ανάλυση της τιμής της αντικειμενικής, για 25 κύκλους είναι:



**Σχήμα 5.21** Διάγραμμα Προβλήματος 9 για 30 διαστάσεις

Παρατηρούμε αρκετές παραβιάσεις περιορισμών και παγιδεύσεις σε τοπικά ακρότατα. Επίσης οι λύσεις που βρήκαμε ήταν αρκετά μακριά από τη βέλτιστη, που έχει βρεθεί. Ακόμα παρατηρήσαμε, ότι για πολύ μικρές τιμές της  $\lambda$  έχουμε πολλές μη-εφικτές λύσεις χωρίς όμως να βρίσκουμε αντίστοιχα καλύτερες λύσεις, όπως για τις 10 διαστάσεις.

Γενικά παρατηρείται έντονη τάση του αλγορίθμου SL-PSO να παγιδεύεται σε τοπικά ακρότατα στα Προβλήματα 7, 8 και 9 (και εν μέρει στο 3), πράγμα το οποίο μπορεί να οφείλεται στην κοινή μορφή της αντικειμενικής συνάρτησης, που έχουν.

### 5.3.10 Συμπεράσματα από την επίλυση προβλημάτων με την SL-PSO

Από τη διαδικασία επίλυσης των παραπάνω προβλημάτων και από τα αποτελέσματά τους, εξαγάγαμε κάποια γενικά συμπεράσματα για τη συμπεριφορά του αλγορίθμου.

Τα πιο βασικά είναι:

- Εντοπίσαμε έντονες συσχετίσεις των παραμέτρων μεταξύ τους, οι οποίες επηρεάζουν την τιμή της αντικειμενικής, σημαντικά. Δηλαδή, δεν μπορεί να γίνει ανάλυση της κάθε παραμέτρου ξεχωριστά, καθώς αλλάζοντας την μία, οι τιμές των άλλων δύο παύουν να είναι οι καλύτερες δυνατές. Έτσι, εάν

αλλάξουμε την τιμή της μίας παραμέτρου, πρέπει να ξαναγίνουν πειράματα για την εύρεση των καλύτερων τιμών των άλλων δύο.

- Εντοπίστηκαν έντονα προβλήματα παγιδεύσεων σε τοπικά ακρότατα, σε τέσσερα προβλήματα, τα οποία είχαν κοινή μορφή αντικειμενικής συνάρτησης. (Προβλήματα 3, 7, 8 και 9)
- Η εισαγωγή συστροφής στο πεδίο των λύσεων, με τη χρήση του πίνακα  $M$ , δυσκόλεψε τη διαδικασία της εξέλιξης με την SL-PSO, όσον αφορά την τιμή της αντικειμενικής συνάρτησης, τις παραβιάσεις και τις παγιδεύσεις σε τοπικά ακρότατα. Ακόμα, κατά την πειραματική διαδικασία, παρατηρήθηκε σημαντική αύξηση του χρόνου εκτέλεσης του αλγορίθμου, σε αυτά τα προβλήματα. (Προβλήματα 6 και 8)
- Σε γενικές γραμμές, παρατηρήθηκε αύξηση των παραβιάσεων των περιορισμών, για τιμές του τεχνητού βάρους  $\lambda \leq 1$ . Αυτό συμβαίνει, πιθανόν, γιατί η τιμή της ποινής στην αντικειμενική συνάρτηση, για τις τιμές αυτές του  $\lambda$ , είναι πολύ μικρή, και συνεπώς δεν επηρεάζει πολύ την τιμή αυτής της λύσης, έτσι ώστε να την απορρίψει ο αλγόριθμος.
- Σε γενικές γραμμές, παρατηρήθηκε απαίτηση σε μεγαλύτερες τιμές της τυπικής απόκλισης του νέφους (*initstd*), ανάλογα με το βαθμό δυσκολίας εύρεσης ικανοποιητικών λύσεων με τιμές αντικειμενικής που να τείνουν στη βέλτιστη. Αυτή η αύξηση παρατηρήθηκε και μεταξύ των διαστάσεων του  $x$  (10 και 30), για το ίδιο πρόβλημα, όπου για 30 διαστάσεις υπήρχε απαίτηση για μεγαλύτερη τιμή της τυπικής απόκλισης του νέφους (*initstd*).

#### **5.4 Παράμετροι που ελέγχθηκαν στον κώδικα εDEag, σε γλώσσα C**

Γενικές Παράμετροι :

- exp, με την οποία ορίζουμε τον αριθμό των κύκλων που «τρέχει» ο κώδικας, κυρίως για λόγους στατιστικής ανάλυσης των αποτελεσμάτων του. Για το πρόβλημα αυτό, η παράμετρος exp πήρε την τιμή 25 για τα τελικά πειράματα, που επιθυμούσαμε στατιστική ανάλυση, και 1 για τα πειράματα ευαισθησίας.
- gen, με την οποία ορίζαμε τον αριθμό των μέγιστων γενεών-επαναλήψεων που επιθυμούμε, ως ένα ακόμα μέσο τερματισμού του αλγορίθμου.



- pop, με την οποία ορίζουμε τον αρχικό πληθυσμό των σωματιδίων που έπαιρναν τις τυχαίες αρχικές τιμές, και από τον οποίο επιλέγονταν τα καλύτερα σωματίδια για τη διαδικασία της εξέλιξης.
- eval, είναι ο μέγιστος αριθμός αξιολογήσεων της αντικειμενικής, τον οποίο όταν ο κώδικας ξεπεράσει, ξεκινάει η διαδικασία τερματισμού. Για την συγκεκριμένη παράμετρο επιλέχθηκε να γίνει έλεγχος για τρεις τιμές:  $5.00E+06$ ,  $1.00E+07$  και  $2.00E+07$ . Αυτές οι τιμές επιλέχθηκαν για λόγους σύγκρισης με τη μέθοδο SL-PSO, καθώς παρατηρήθηκε ότι για τιμές μικρότερες του  $5.00E+06$  οι λύσεις και στις δύο ήταν αρκετά μακριά από τη βέλτιστη, που είχε βρεθεί, και ότι για τιμές μεγαλύτερες του  $2.00E+07$  υπήρχαν πολύ μικρές βελτιώσεις με δυσανάλογη αύξηση του χρόνου ολοκλήρωσης και των δύο αλγορίθμων.

Παράμετροι της διαδικασίας βελτιστοποίησης :

- F, είναι ο παράγοντας κλιμάκωσης ( $F_0$ ), όπως αυτός ορίστηκε στη διαδικασία της «διαφορικής εξέλιξης» (differential evolution, DE).
- CR, είναι ο ρυθμός διασταύρωσης ( $CR_0$ ), όπως αυτός ορίστηκε στη διαδικασία της «διαφορικής εξέλιξης» (differential evolution, DE).
- epop, είναι ο ενεργός πληθυσμός σωματιδίων, τα οποία επιλέγονται από τον αρχικό πληθυσμό για τη διαδικασία της εξέλιξης, δηλαδή από αυτά επιλέγονται οι «γονείς».

Παράμετροι της «ε constrained method»

- e0r, με την οποία ορίζουμε τη μέγιστη αναλογία των σωματιδίων που επιλέγονται για το αρχικό  $\varepsilon(0)$ , δηλαδή με αυτό επηρεάζουμε την τιμή του  $\theta$ .
- er, με την οποία ορίζουμε τη μέγιστη αναλογία των ελεγχόμενων με το βαθμό  $\varepsilon$  γενεών, προς το συνολικό αριθμό γενεών. Έτσι επηρεάζουμε την τιμή του  $T_c$ .

Παράμετροι της «gradient-based mutation» :

- gr, όπου είναι η πιθανότητα πραγματοποίησης της διαδικασίας διαβαθμισμένης μετάλλαξης  $P_g$ .

- $gs$ , με την οποία ορίζουμε το μέγιστο αριθμό πραγματοποίησης διαβαθμισμένης μετάλλαξης για έναν «απόγονο», έως ότου αυτός μετατραπεί σε εφικτή λύση ( $R_g$ ).

## 5.5 Εισαγωγή του «ΠΑΠΕ» στον κώδικα εDEag

Αρχικά, πρέπει να επισημανθεί ότι το συγκεκριμένο τμήμα της εργασίας όπως και το επόμενο τμήμα των πειραματικών δοκιμών, στον κώδικα εDEag, πραγματοποιήθηκαν σε συνεργασία με το συμφοιτητή Μαρίνο Αντώνιο.

Ο κώδικας, που είχαμε να εξετάσουμε, είναι γραμμένος σε γλώσσα προγραμματισμού C. Επίσης η γεννήτρια τυχαίων αριθμών που έχει, καθώς και οι πίνακες που αποθηκεύει τα διανύσματα θέσης των σωματιδίων είναι εφαρμοσμένες με τέτοιο τρόπο ώστε να επεξεργάζονται τα υποψήφια διανύσματα λύση σε πίνακα με μόνο μία διάσταση, δηλαδή το διάνυσμα που έχουμε εμείς από την SL-PSO έπρεπε να γίνει από  $x[24,25]$  σε  $x[600,1]$ . Για να γίνει αυτό, ορίσαμε ότι: τα στοιχεία  $x(1,2, \dots, 24)$  αντιπροσωπεύουν τις 24 ώρες για το πρώτο εργοστάσιο, τα  $x(25,26, \dots, 48)$  για το δεύτερο κ.ο.κ. Άρα έχουμε το γενικό τύπο:

$$X_n = x(1 + 24(n - 1), 2 + 24(n - 1), \dots, 24 + 24(n - 1)) \quad (5.1)$$

όπου  $n$  ο αριθμός του εργοστασίου (1,2,...,25). Μπορούμε να ξέρουμε με αυτό τον τρόπο, το 24ωρο μέρος της λύσης που αφορά το αντίστοιχο εργοστάσιο. Έτσι μπορούμε να μετατρέψουμε εύκολα μία λύση από τη λογική της εDEag στη λογική της SL-PSO, για σύγκριση, και αντίστροφα.

Με αυτή τη λογική αλλάξαμε την αντικειμενική συνάρτηση. Αρχικά έπρεπε να αλλάξουμε τους πίνακες κόστους, με το ψευδοκώδικα:

```

k=0
για i από 1 έως 25
    για j από 1 έως 24
        A[k] = a[i]
        B[k] = b[i]
        C[k] = c[i]
        k = k+1
    τέλος
τέλος

```

**Σχήμα 5.22** Ψευδοκώδικας αλλαγής μορφής πίνακα

Και στη συνέχεια προσαρμόσαμε την αντικειμενική συνάρτηση:

$$f(x) = \sum_{k=1}^{600} [(A_k x_k^2 + B_k x_k + C_k)] \quad (5.2)$$

Μαζί με αυτό έπρεπε να αλλάξουμε αντίστοιχα και τη λογική των περιορισμών και του πεδίου ορισμού του διανύσματος θέσης. Έτσι:

```

k=0
για j από 1 έως 24
  t[j]=0
  για i από 1 έως 25
    k = j + i*24
    t[j] = t[j] + x[k]
  τέλος
τέλος

```

**Σχήμα 5.23** Ψευδοκώδικας αλλαγής μορφής περιορισμών

$$g(j) = t(j) - d(j) \text{ για } j \text{ από } 1 \text{ έως } 24 \quad (5.3)$$

Όπου  $d(j)$  εκφράζει τη ζήτηση για κάθε  $j$  ώρα της ημέρας.

Και για το πεδίο ορισμού:

```

για j από 1 έως 24
  k = j + 0*24
  Κάτω Όριο x[k]=...
  Πάνω Όριο x[k]=...
  k = j + 1*24
  Κάτω Όριο x[k]=...
  Πάνω Όριο x[k]=...
  ...
  k = j + 24*24
  Κάτω Όριο x[k]=...
  Πάνω Όριο x[k]=...
τέλος

```

**Σχήμα 5.24** Ψευδοκώδικας εισαγωγής πεδίου ορισμού

## **5.6 Πειραματική Διαδικασία και Αποτελέσματα («ΠΔΠΕ»)**

Για τη διαδικασία αυτή, παρατηρήσαμε ότι έχουμε μία πληθώρα παραμέτρων, με πολλές από αυτές να μην είναι ανεξάρτητες μεταξύ τους. Έτσι πραγματοποιήσαμε πειράματα σε κάποιες από τις γενικές παραμέτρους που έχουμε. Βρήκαμε έτσι τις βέλτιστες τιμές για την eror, για την F και για την CR. Στη συνέχεια, βρήκαμε μία ικανοποιητική τιμή για τον αρχικό πληθυσμό pop και για τον gen, οι οποίες όμως χρειάζεται να αυξηθούν ανάλογα με τον eval, λόγω των ορίων που έχει ο ίδιος ο κώδικας, αλλά και για να καταφέρουμε βελτίωση της λύσης. Για το τέλος αφήσαμε τις μεταβλητές που αλληλοεπηρεάζονται έντονα, δηλαδή τις er, eOr και gr, gs.

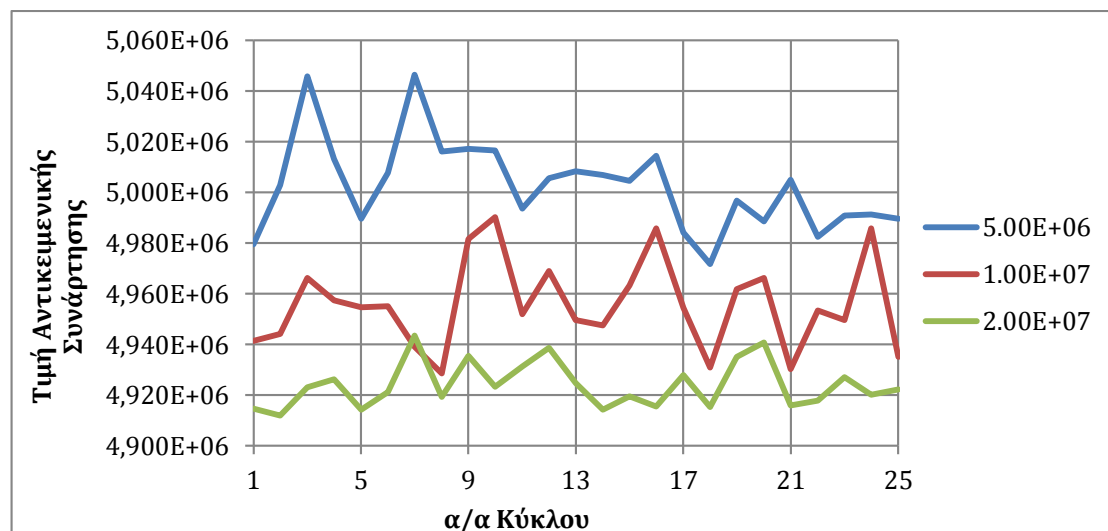
Τα καλύτερα αποτελέσματα που πήραμε, για 25 κύκλους, είναι τα ακόλουθα :

<b>FES</b>	<b>5.00E+06</b>	<b>1.00E+07</b>	<b>2.00E+07</b>	<b>2.40E+07</b>
<i>eval</i>	5.00E+06	1.00E+07	2.00E+07	1.00E+08
<i>pop</i>	5000	4000	10000	10000
<i>gen</i>	5000	10000	20000	20000
<i>er</i>	1	1	1	1
<i>eOr</i>	100	100	100	100
<i>gr</i>	0.2	0.1	0.1	0.1
<i>gs</i>	1	3	3	3
<i>F</i>	0.5	0.5	0.5	0.5
<i>CR</i>	1	1	10	10
<i>epop</i>	500	500	500	500
<i>median</i>	5.005E+06	4.954E+06	4.922E+06	4.920E+06
<i>best</i>	4.972E+06	4.929E+06	4.912E+06	4.909E+06
<i>worst</i>	5.046E+06	4.990E+06	4.943E+06	4.934E+06
<i>std</i>	1.813E+04	1.752E+04	9.039E+03	7.040E+03

**Πίνακας 5.19** Πίνακας αποτελεσμάτων του αλγόριθμου εDEag

Λόγω του τρόπου που λειτουργεί ο κώδικας, μετά τις 2.00E+07 αξιολογήσεις της αντικειμενικής (FES) αρχίζει και συγκλίνει σε τέτοιο βαθμό, που τερματίζεται μόνος του χωρίς να φτάσει ποτέ στην τιμή 1.00E+08 που του έχουμε ορίσει. Η βέλτιστη λύση βρέθηκε για FES κοντά στις 2.40E+07 με τιμή 4.909E+06.

Η αναλυτική σύγκριση μεταξύ τους:



**Σχήμα 5.25** Αποτελέσματα βέλτιστης λύσης συναρτήσεως του FES (εDEag)

Παρατηρούμε ότι όσο μεγαλώνουμε τον αριθμό των αξιολογήσεων της αντικειμενικής συνάρτησης, τόσο βελτιώνεται και ομαλοποιείται η λύση από επανάληψη σε επανάληψη, κάτι που ήταν αναμενόμενο καθώς ο αλγόριθμος μπορεί να συγκλίνει πιο κοντά στη βέλτιστη λύση, γιατί έχει τη δυνατότητα να «τρέξει» περισσότερες επαναλήψεις σε κάθε κύκλο.

Στη μέθοδο εDEag, λόγω του τρόπου που εργάζεται πάνω στην αντικειμενική, δεν έχουμε σχεδόν ποτέ παραβίαση των περιορισμών. Δηλαδή, από τις παραπάνω λύσεις, καμία δεν είχε παραβίαση.

### 5.6.1 Σύγκριση με τη μέθοδο SL-PSO σε MATLAB® («ΠΔΠΕ»)

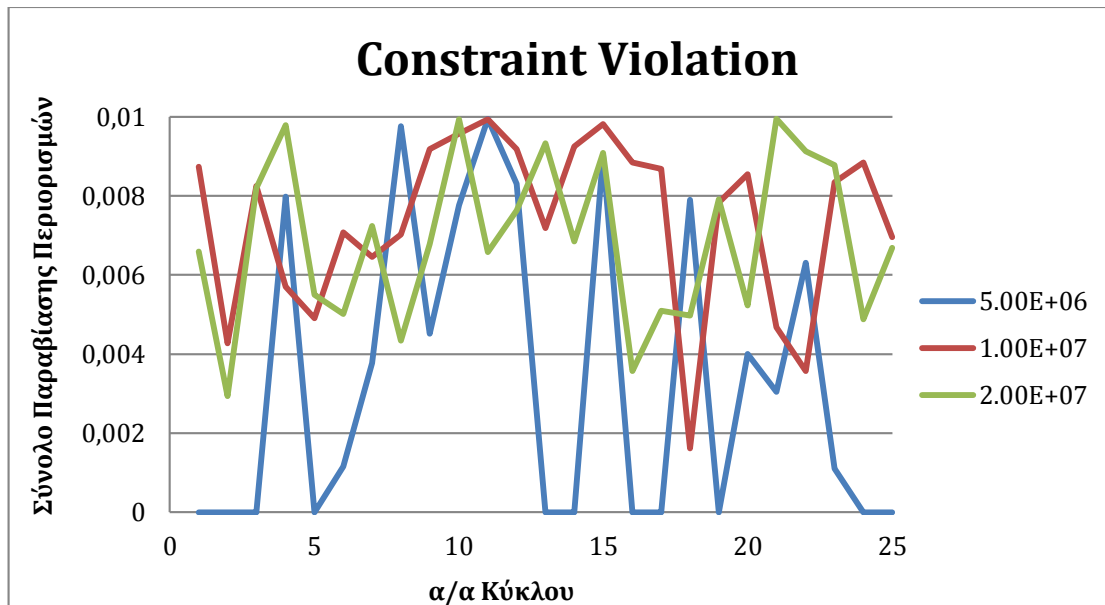
Για την SL-PSO στην MATLAB® υπήρχε ήδη μελέτη πάνω στις τιμές των παραμέτρων που είχαμε να ερευνήσουμε. Έτσι προχωρήσαμε κατευθείαν στο σημείο των αποτελεσμάτων.

Η σύγκριση έγινε για FES 5.00E+06, 1.00E+07 και 2.00E+07. Τα αποτελέσματα είναι: («Objective»=Τιμή αντικειμενικής συνάρτησης, «CNV»= Σύνολο παραβίασης περιορισμών)

	FES 5.00E+06		FES 1.00E+07		FES 2.00E+07	
	Objective	CNV	Objective	CNV	Objective	CNV
<i>median</i>	4.914E+06	1.159E-03	4.910E+06	8.246E-03	4.909E+06	6.780E-03
<i>best</i>	4.909E+06	0.000E+00	4.906E+06	1.611E-03	4.906E+06	2.943E-03
<i>worst</i>	4.918E+06	9.953E-03	4.913E+06	9.936E-03	4.914E+06	9.944E-03
<i>std</i>	2.608E+03	3.803E-03	1.781E+03	2.172E-03	1.961E+03	2.063E-03

**Πίνακας 5.20** Αποτελέσματα βέλτιστης λύσης συναρτήσεως του FES (SL-PSO)

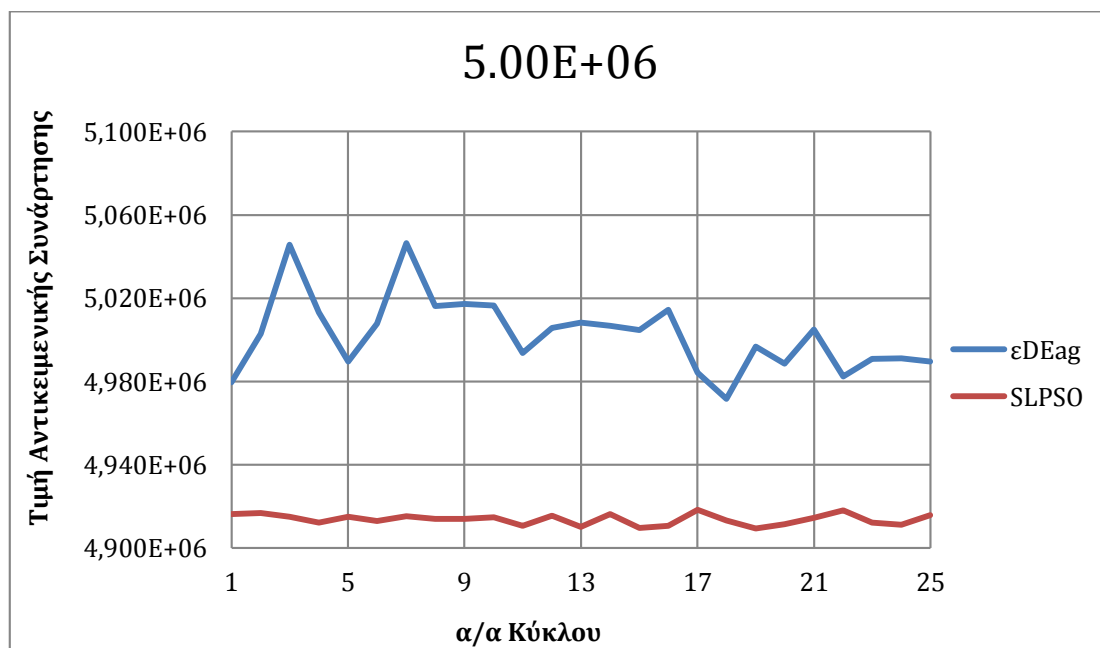
Αυτό που παρατηρούμε αρχικά είναι ότι σε αντίθεση με τη μέθοδο εDEag, έχουμε μικρές παραβιάσεις στους περιορισμούς. Αυτό συμβαίνει λόγω του τρόπου που λειτουργεί ο αλγόριθμος της SL-PSO (μέθοδος ποινής), η οποία αφήνει περιθώρια για πολύ μικρές παραβιάσεις. Αυτές τις παραβιάσεις τις αναφέρουμε:



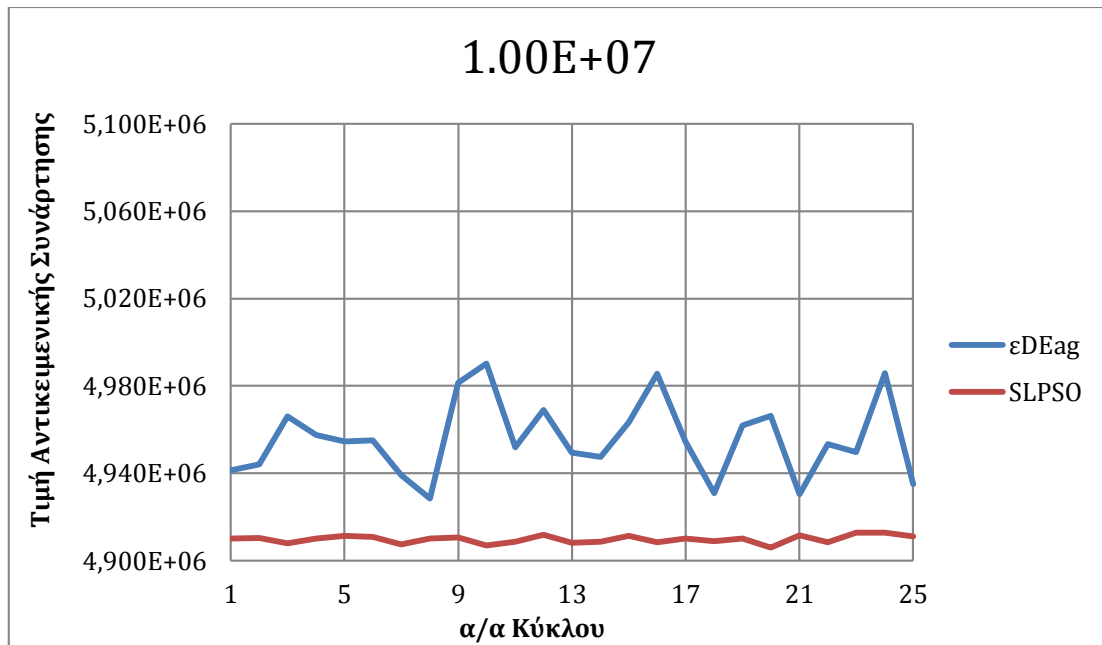
**Σχήμα 5.26** Παραβιάσεις βέλτιστης λύσης συναρτήσεως του FES (SL-PSO)

Για τις παραβιάσεις των περιορισμών, παρατηρούμε ότι όσο μεγαλώνει ο αριθμός των αξιολογήσεων της αντικειμενικής συνάρτησης που πραγματοποιούμε, τόσο μεγαλώνουν και αυτές, και αρχίζουν να αποκτούν συνοχή στην τιμή τους. Όλες φαίνεται να έχουν ένα ανώτατο όριο την τιμή 0.01, και να τείνουν προς τα εκεί.

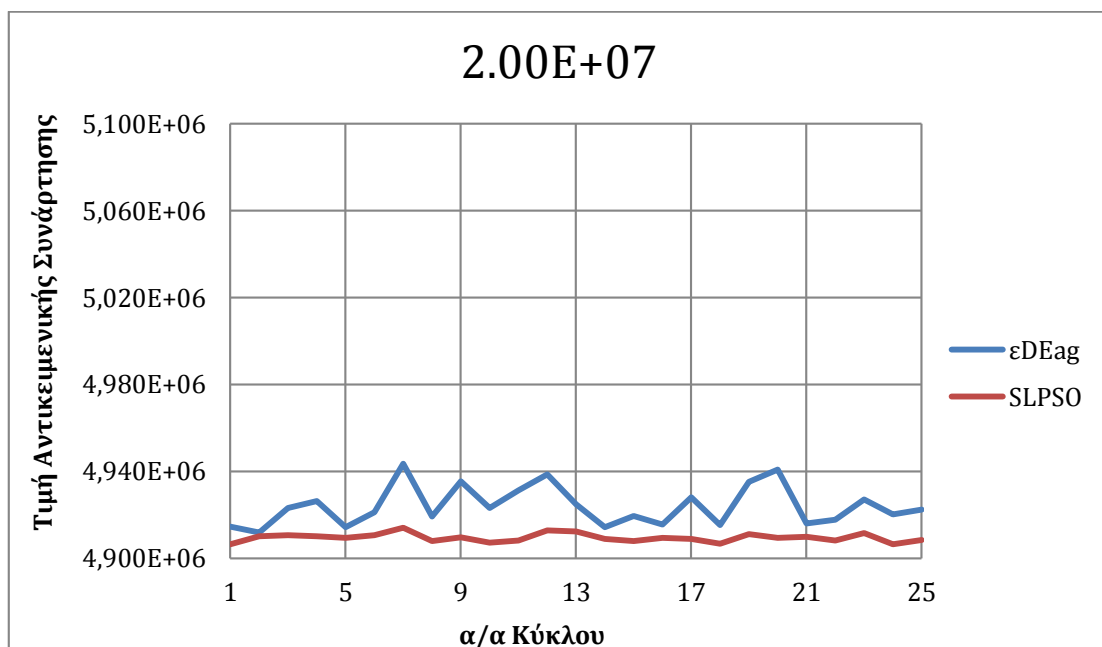
Στη συνέχεια προχωρούμε σε σύγκριση των δύο μεθόδων, συναρτήσεως του μέγιστου αριθμού των αξιολογήσεων της αντικειμενικής συνάρτησης.



**Σχήμα 5.27** Σύγκριση βέλτιστης λύσης για 5.00E+06 FES



**Σχήμα 5.28** Σύγκριση βέλτιστης λύσης για 1.00E+07 FES



**Σχήμα 5.29** Σύγκριση βέλτιστης λύσης για 2.00E+07 FES

Από αυτά παρατηρούμε ότι η SL-PSO έχει αρκετά μεγαλύτερη συνοχή όσον αφορά την τιμή της αντικειμενικής συνάρτησης από επανάληψη σε επανάληψη, με σταθερά καλύτερες λύσεις, για τις τιμές των παραμέτρων που εξετάσαμε σε αυτή την εργασία. Ακόμα παρατηρούμε ότι όσο αυξάνουμε τα FES τόσο η εDEag πλησιάζει τις τιμές, και τη συνοχή της SL-PSO.





## **ΚΕΦΑΛΑΙΟ 6**

### **ΣΥΜΠΕΡΑΣΜΑΤΑ**



## **6.1 Συμπεράσματα – Επίλογος Διπλωματικής Εργασίας**

Οι εξελικτικοί αλγόριθμοι αποτελούν σήμερα ένα πολύ σημαντικό εργαλείο για πολλές επιστήμες, καθώς έχουν τη δυνατότητα να βελτιστοποιούν μεγάλα και πολύπλοκα μη-γραμμικά προβλήματα σε αρκετά σύντομο χρονικό διάστημα. Η ποικιλία των μεθόδων συνεχώς μεγαλώνει, και οι μέθοδοι που χρησιμοποιούνται συνεχώς εξελίσσονται, δημιουργώντας έτσι το «έδαφος» για επέκταση των εφαρμογών τους. Αυτό δημιουργεί την ανάγκη για περισσότερη έρευνα πάνω στο συγκεκριμένο κλάδο, πράγμα που είναι αυτό στο οποίο προσπαθεί να συνεισφέρει και η παρούσα εργασία.

Με την εισαγωγή και μελέτη 9 διαφορετικών προβλημάτων στον αλγόριθμο SL-PSO, προσπαθήσαμε να ερευνήσουμε τη συμπεριφορά του σε δύσκολα μη-γραμμικά προβλήματα, να εντοπίσουμε πιθανά μοτίβα συμπεριφοράς, και να συγκρίνουμε τα αποτελέσματά του με άλλες μεθόδους, από τη βιβλιογραφία. Ταυτόχρονα, γνωρίζοντας τη συμπεριφορά του αλγορίθμου SL-PSO σε ένα πολύπλοκο ενεργειακό πρόβλημα, το εισαγάγαμε στον αλγόριθμο εDEag, ο οποίος ακολουθεί διαφορετικές προσεγγίσεις για τις διαδικασίες της εξέλιξης και του χειρισμού των περιορισμών, και συγκρίναμε τα αποτελέσματα και τις συμπεριφορές των δύο αυτών μεθόδων. Όλα αυτά πραγματοποιήθηκαν με πειράματα πάνω στις παραμέτρους των μεθόδων αυτών και με στόχο τον εντοπισμό των βέλτιστων τιμών τους.

Στον αλγόριθμο SL-PSO μελετήθηκε η συμπεριφορά τριών παραμέτρων: του τεχνητού βάρους των ποινών, της αρχικής τυπικής απόκλισης του νέφους και του ρυθμού μείωσής του. Οι βέλτιστες τιμές που βρέθηκαν, σε αρκετά προβλήματα είναι αρκετά κοντά σε αυτές από τη βιβλιογραφία. Σε πολλά προβλήματα εντοπίστηκαν έντονα στοιχεία αλληλεξάρτησης των τιμών της μία παραμέτρου από την άλλη, με διαφορετικό τρόπο σε κάθε πρόβλημα, οι οποίες μας έδιναν ποικιλία συνδυασμών με κοντινές τιμές αντικειμενικής συνάρτησης. Επίσης, παρατηρήθηκε δυσκολία του αλγορίθμου να αποφύγει τοπικά ακρότατα σε προβλήματα με συγκεκριμένης μορφής αντικειμενική συνάρτηση, κάτι το οποίο γινόταν ακόμα πιο έντονο με ταυτόχρονη αύξηση της πολυπλοκότητας των περιορισμών.

Στον αλγόριθμο εDEag μελετήθηκε μία πληθώρα παραμέτρων, με κύριες να είναι αυτές που επηρεάζουν τη μέθοδο χειρισμού των περιορισμών και τη διαδικασία της μετάλλαξης, αλλά και αυτές που επηρεάζουν τη διαδικασία της βελτιστοποίησης.

Παρατηρήθηκαν κάποιες αλληλεξαρτήσεις μεταξύ των παραμέτρων. Το μεγάλο πλήθος των παραμέτρων που έπρεπε να εξεταστούν δυσκόλεψε αρκετά την πειραματική διαδικασία, καθώς μεγαλώνει πολύ τη πιθανότητα να υπάρχει συνδυασμός τους που δεν εξετάστηκε και που πιθανόν να δίνει καλύτερη λύση από αυτή που βρέθηκε. Τα αποτελέσματα των πειραμάτων, και οι συγκρίσεις με τον SL-PSO, έδειξαν ότι ο αλγόριθμος μπορεί να δώσει αποτελέσματα κοντά σε αυτά SL-PSO, με το δεύτερο να δίνει λίγο πιο σταθερές και καλύτερες λύσεις.

Με την εισαγωγή και μελέτη του ρεαλιστικού πρακτικού προβλήματος ELD είδαμε ότι και οι δύο αλγόριθμοι είναι ικανοί για την επίλυση τέτοιου τύπου (μεγάλων ενεργειακών) προβλημάτων. Ταυτόχρονα, είχαν την ικανότητα να αποδώσουν σταθερές λύσεις σε όλο το πλήθος των επαναλήψεων, κάτι το οποίο εγγυάται ότι αυτοί οι αλγόριθμοι, παρότι χρησιμοποιούν στοχαστικές μεθόδους, μπορούν ακόμα και με μία επανάληψη να δώσουν μία πολύ καλή προσέγγιση της βέλτιστης τιμής. Αυτό τους καθιστά ιδανική λύση για διαχειριστές δικτύου ηλεκτρισμού. Δηλαδή, αυτοί οι αλγόριθμοι καθίστανται ικανοί για επίλυση και σε πραγματικές συνθήκες, πέρα από την επίλυση θεωρητικών μαθηματικών προβλημάτων. Παρόλα αυτά, φάνηκαν και κάποια μειονεκτήματα, με κύριο να είναι η απαίτηση ρύθμισης πολλών παραμέτρων, με την καθεμία να έχει σχετικά μεγάλο εύρος δυνατών τιμών, και με κάποιες να επηρεάζουν πολύ έντονα την τιμή που θα βρεθεί. Αυτό έγινε ιδιαίτερα αισθητό στον αλγόριθμο εDEag, λόγω των πολλών παραμέτρων που είχε, κάτι το οποίο σημαίνει ότι ο χρήστης (διαχειριστής του δικτύου) του αλγορίθμου θα πρέπει να αφιερώσει παραπάνω χρόνο στη διαδικασία της βελτιστοποίησής τους. Γενικά, οι εξελικτικοί αλγόριθμοι αποτελούν μία καλή επιλογή για βραχυπρόθεσμη κάλυψη ζήτησης ηλεκτρικής ενέργειας (όπως αναφέραμε και στο [«υποκεφάλαιο 4.2»](#)), και οι δύο αλγόριθμοι που μελετήσαμε δεν αποτελούν εξαιρέσεις.

Τέλος, τα παραπάνω συμπεράσματα μας δείχνουν τα «δυνατά» και «αδύναμα» σημεία του κάθε αλγορίθμου, καθώς και τη συμπεριφορά των διαφόρων τεχνικών και παραμέτρων που έχει ο καθένας. Ακόμα, αξίζει να επισημανθεί η ανάγκη έρευνας των αλγορίθμων αυτών σε ακόμα περισσότερες μορφές προβλημάτων, έτσι ώστε να γίνει πιο πλήρης ο τρόπος της συμπεριφοράς τους. Ταυτόχρονα θα μπορούσε να γίνει έρευνα για εισαγωγή νέων τεχνικών, σε αυτούς, που θα μπορούσαν να αντιμετωπίσουν τις διάφορες «αδυναμίες», όπως την παγίδευση σε τοπικά ακρότατα, ή την ανάγκη για εισαγωγή πολλών παραμέτρων.

## Βιβλιογραφία

Adeli, H., & Cheng, N.-T. (1994, January). Augmented lagrangian genetic algorithm for structural optimization. *Journal of Aerospace Engineering*, 7(1) , pp. 104-118.

Back et al., T. (1997). Evolutionary Computation: Comments on the History and Current State. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 1, no. 1 .

Bean, J. C., & Hadj-Alouane, A. B. (1992). A dual genetic algorithm for bounded integer programs. *Technical Report TR 92-53* .

Belur, S. V. (1997). CORE: Constrained Optimization by Random Evolution. In J. R. Koza, *Late Breaking Papers at the Genetic Programming 1997 Conference* (pp. 280-286). Stanford University, California: Stanford Bookstore.

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* .

Campbell, S., & Meyer, C. J. (1979). *Generalized Inverses of Linear Transformations*. Dover Publications.

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.

Chootinan, P., & Chen, A. (2006, Αύγουστος). Constraint handling in genetic algorithms using gradient-based repair method. *Computers & Operations Research*, vol. 33, no. 8, pp. , σσ. 2263–2281,.

Coello, C. A. (1999). A survey of constraint handling techniques used with evolutionary algorithms. *Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada* .

Davidor, Y. (1989). Analogous Crossover. In J. D. Schaffer, *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 98-103). San Mateo, California: Morgan Kaufmann Publisher.

- Davidor, Y. (1990). *Genetic Algorithms and Robotics : A Heuristic Strategy for Optimization*. Singapore: World Scientific Publishing Co.
- Davis, L. (1987). *Genetic Algorithms and Simulated Annealing*. Pitman, London.
- Davis, L. (1991). *Handbook of Genetic Algorithms*. New York, New York: Van Nostrand Reinhold.
- Deb, K. (2000). Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2) , 311-338.
- Eiben, A. E., Raue, P. E., & Ruttkay, Z. (1994). Genetic algorithms with multi-parent recombination. *Parallel Problem Solving from Nature—PPSN III* , pp. 78-87.
- 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.
- Eita, M. A., Shoukry, A. B., & Iba, H. C. (2014). Constrained Group Counseling Optimization. Στο *ALIFE 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*. Egypt-Japan University of Science and Technology, Alex, EGYPT; Alexandria University, Alex, EGYPT; The University of Tokyo, Tokyo, Japan.
- Fogel, L. (1962). Autonomous automata. *Ind. Res.*, vol. 4 , 14-19.
- Fonseca, C. M., & Fleming, P. J. (1995, Spring). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1) , 1-16.
- 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, R. M. (1958). A learning machine: Part I. *IBM J. vol. 2, no. 1* , 2-13.
- Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, vol. 13 , 533-549.
- Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences* 8 (1) .

- Hoffmeister, F., & Sprave, J. (1996). Problem-independent handling of constraints by use of metric penalty. Στο L. J. Fogel, P. J. Angeline, & T. Bäck, *Conference on Evolutionary Programming (EP'96) Proceedings of the Fifth Annual* (σσ. 289-294). San Diego, California, USA: The MIT Press.
- Holland, J. (1962). Outline for a logical theory of adaptive systems. *J. Assoc. Comput. Mach.*, vol. 3 , 297-314.
- Homaifar, A., Lai, S. H., & Qi, X. (1994). Constrained Optimization via Genetic Algorithms. *Simulation*, 62(4) , 242-254.
- 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., & 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.
- Kirkpatrick, S., Gelatt, J. C., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220 , 671-680.
- Kowalczyk, R. (1997, April). Constraint Consistent Genetic Algorithms. *Proceedings of the 1997 IEEE Conference on Evolutionary Computation* , pp. 343-348.
- Koza. (1990). Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. .
- Koziel, S., & Michalewicz, Z. (1998). A Decoder-based Evolutionary Algorithm for Constrained Parameter Optimization Problems. In T. Bäck, A. E. Eiben, M. Schoenauer, & H.-P. Schwefel, *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)* (pp. 231-240). Amsterdam: Springer-Verlag.
- Kumar, V. (1992, Spring). Algorithms for Constraint-Satisfaction Problems: A Survey. *AI Magazine* , σσ. 32-44.

- Kuri, A. M., & Quezada, C. V. (1998). A Universal Eclectic Genetic Algorithm for Constrained Optimization. In *Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98* (pp. 518-522). Aachen, Germany: Verlag Mainz.
- Kusakci, A. O., & Can, M. (2013). An adaptive penalty based covariance matrix adaptation–evolution strategy. *Computers & Operations Research* 40 , σσ. 2398–2417.
- Larrañaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Boston: Kluwer Academic Publishers.
- Le, T. V. (1995). A Fuzzy Evolutionary Approach to Constrained Optimization Problems. *Proceedings of the Second IEEE Conference on Evolutionary Computation* (pp. 274-278). Perth: IEEE.
- Li, C., & S. Yang. (2009). “An adaptive learning particle swarm optimizer for function optimization. *Conger. Evol. Comput.* , 381-388.
- Liepins, G. E., & Vose, M. D. (1990). Representational Issues in Genetic Optimization. *Journal of Experimental and Theoretical Computer Science* 2(2) , pp. 4-30.
- Mallipeddi, R., & Suganthan, P. N. (2010). Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, second edition.
- Michalewicz, Z. (1995). Genetic Algorithms, Numerical Optimization, and Constraints. Στο L. J. Eshelman, *Proceedings of the Sixth International Conference on Genetic Algorithms* (σσ. 151-158). San Mateo, California: University of Pittsburgh, Morgan Kaufmann Publishers.
- 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., & Dasgupta, D. (1997). *Evolutionary Algorithms in Engineering Applications*. Berlin: Springer-Verlag.



Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, vol. 4, issue 1 .

Michalewicz, Z., Dasgupta, D., Le Riche, R., & Schoenauer, M. (1996, September). Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal*, 30(4) , pp. 851-870.

Moscato, P. (1989). On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. *Caltech Concurrent Computation Program (report 826)* .

Paredis, J. (1994). Co-evolutionary constraint satisfaction. *Proceedings of the 3rd Conference on Parallel Problems Solving from Nature* (pp. 46-55). Springer Verlag.

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* (pp. 424-430). Morgan Kaufmann.

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.

Richardson, J. T., Palmer, M. R., Liepins, G., & Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 191-197). George Mason University: Morgan Kaufmann Publishers.

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.

Schoenauer, M., & Xanthakis, S. (1993). Constrained GA optimization. *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 573-580). Morgan Kaufmann.

Schwefel, H.-P. (1975). *Evolutionsstrategie und numerischer Optimierung* Dissertation. .

Singh, N., & Kumar, Y. (2013). Constrained economic load dispatch using evolutionary technique. *Asian Journal of Technology & Management Research [ISSN: 2249-0892]* .

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 11 , pp. 341–359.

Takahama, T., & Sakai, S. (2005, Οκτώβριος). Constrained optimization by applying the  $\alpha$  constrained method to the nonlinear simplex method with mutations. *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 5 , σσ. 437–451.

Takahama, T., & Sakai, S. (2010). Constrained Optimization by the  $\varepsilon$  Constrained Differential Evolution with an Archive and Gradient-Based Mutation. *WCCI 2010 IEEE World Congress on Computational Intelligence*. Barcelona, Spain.

Takahama, T., & Sakai, S. (2008). Constrained optimization by the  $\varepsilon$  constrained differential evolution with dynamic  $\varepsilon$ -level control. *Advances in Differential Evolution*, U. Chakraborty, Ed. Springer-Verlag , pp. 139–154.

Takahama, T., & Sakai, S. (2006, Ιούλιος). Constrained optimization by the  $\varepsilon$  constrained differential evolution with gradient-based mutation and feasible elites. *Proc. of the 2006 IEEE Congress on Evolutionary Computation* , σσ. 308–315.

Takahama, T., & Sakai, S. (2009, Μάιος). Fast and stable constrained optimization by the  $\varepsilon$  constrained differential evolution. *Pacific Journal of Optimization*, vol. 5, no. 2 , pp. 261–282,.

Takahama, T., Sakai, S., & Iwane, N. (2006, Οκτώβριος). Solving nonlinear constrained optimization problems by the  $\varepsilon$  constrained differential evolution. *Proc. of the 2006 IEEE Conference on Systems, Man, and Cybernetics* , pp. 2322–2327.

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* .

Zhan et al., Z. (2009). Adaptive particle swarm optimization. *IEEE Trans. Syst., Man, Cybern. B: Cybern*, vol. 39 , 1362-1381.

# **ΠΑΡΑΡΤΗΜΑΤΑ**



## Παράρτημα Α

Σε αυτό το παράρτημα περιγράφονται αναλυτικά τα προβλήματα που εξετάσαμε, στα πλαίσια αυτής της διπλωματικής, στην SL-PSO, τα οποία προέρχονται από το διαγωνισμό CEC 2010 (Mallipeddi & Suganthan, 2010).

Με  $f(x)$  συμβολίζεται η αντικειμενική συνάρτηση την οποία θέλουμε να ελαχιστοποιήσουμε σε όλες τις περιπτώσεις. Με  $g(x)$  συμβολίζονται οι περιορισμοί ανισότητας, ενώ με  $h(x)$  οι περιορισμοί ισότητας. Με  $D$  συμβολίζουμε τις διαστάσεις του πίνακα « $X$ », που σύμφωνα με το διαγωνισμό μας ενδιαφέρει για 10 και για 30. Σε όλες τις συναρτήσεις υπάρχει και ένας πίνακας « $o$ » που χρησιμοποιείται για να μεταβάλει το πεδίο των λύσεων, είναι μοναδικός για κάθε συνάρτηση και έχει μέγεθος  $[1,30]$  για να μπορεί να εφαρμοστεί για 10 και για 30 διαστάσεις, όπως ορίζει ο διαγωνισμός ([Παράρτημα Β](#)).

Επίσης, στα προβλήματα αναφέρουμε τη βέλτιστη τιμή της αντικειμενικής συνάρτησης που έχει βρεθεί για 10 διαστάσεις (Kusakci & Can, 2013).

### Πρόβλημα 1

Στο 1ο πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = - \left| \frac{\sum_{i=1}^D \cos^4(z_i) - 2 \prod_{i=1}^D \cos^2(z_i)}{\sqrt{\sum_{i=1}^D i z_i^2}} \right| \quad z = x - o \quad (\text{Π.1})$$

Δύο περιορισμούς ανισότητας:

$$g_1(x) = 0.75 - \prod_{i=1}^D z_i \leq 0 \quad (\text{Π.2})$$

$$g_2(x) = \sum_{i=1}^D z_i - 7.5D \leq 0 \quad (\text{Π.3})$$

Και το πεδίο ορισμού:  $x \in [0, 10]^D$

Βέλτιστη λύση:  $f_{best}^{10} = -7.4731e - 01$

## Πρόβλημα 2

Στο 2<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \max(z) \quad z = x - o, \quad y = z - 0.5 \quad (\text{Π.4})$$

Δύο περιορισμούς ανισότητας:

$$g_1(x) = 10 - \frac{1}{D} \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10] \leq 0 \quad (\text{Π.5})$$

$$g_2(x) = \frac{1}{D} \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10] - 15 \leq 0 \quad (\text{Π.6})$$

Έναν περιορισμό ισότητας:

$$h(x) = \frac{1}{D} \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10] - 20 \leq 0 \quad (\text{Π.7})$$

Και το πεδίο ορισμού:  $x \in [-5.12, 5.12]^D$

Βέλτιστη λύση:  $f_{best}^{10} = -2.278e + 00$

## Πρόβλημα 3

Στο 3<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right) \quad z = x - o \quad (\text{Π.8})$$

Έναν περιορισμό ισότητας:

$$h(x) = \sum_{i=1}^{D-1} (z_i - z_{i+1})^2 = 0 \quad (\text{Π.9})$$

Και το πεδίο ορισμού:  $x \in [-1000, 1000]^D$

Βέλτιστη λύση:  $f_{best}^{10} = 0.000e + 00$

## Πρόβλημα 4

Στο 4<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \max(z) \quad z = x - o \quad (\text{Π.10})$$

Τέσσερις περιορισμούς ισότητας:

$$h_1(x) = \frac{1}{D} \sum_{i=1}^D (z_i \cos(\sqrt{|z_i|})) = 0 \quad (\text{Π.11})$$

$$h_2(x) = \sum_{i=1}^{\frac{D}{2}-1} (z_i - z_{i+1})^2 = 0 \quad (\text{Π.12})$$

$$h_3(x) = \sum_{i=\frac{D}{2}+1}^{D-1} (z_i^2 - z_{i+1})^2 = 0 \quad (\text{Π.13})$$

$$h_4(x) = \sum_{i=1}^D z = 0 \quad (\text{Π.14})$$

Και το πεδίο ορισμού:  $x \in [-50, 50]^D$

Βέλτιστη λύση:  $f_{best}^{10} = -1.000e - 05$

## Πρόβλημα 5

Στο 5<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \max(z) \quad z = x - o \quad (\text{Π.15})$$

Δύο περιορισμούς ισότητας:

$$h_1(x) = \frac{1}{D} \sum_{i=1}^D (-z_i \sin(\sqrt{|z_i|})) = 0 \quad (\text{Π.16})$$

$$h_2(x) = \frac{1}{D} \sum_{i=1}^D (-z_i \cos(0.5\sqrt{|z_i|})) = 0 \quad (\text{Π.17})$$

Και το πεδίο ορισμού:  $x \in [-600, 600]^D$

Βέλτιστη λύση:  $f_{best}^{10} = -5.348e + 02$

## Πρόβλημα 6

Στο 6<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \max(z)$$

$$z = x - o$$

$$y = (x + 483,6106156535 - o)M - 483,6106156535 \quad (\text{Π.18})$$

\* Ο πίνακας « $M$ » είναι μοναδικός, και ορισμένος από το διαγωνισμό, για κάθε συνάρτηση που τον συναντάμε, και προσδίδει μία συστροφή στο πεδίο των λύσεων. Επίσης, σε αντίθεση με τον « $o$ », είναι διαφορετικός για 10 και 30 διαστάσεις ([Παράρτημα Γ](#)).

Δύο περιορισμούς ισότητας:

$$h_1(x) = \frac{1}{D} \sum_{i=1}^D (-y_i \sin(\sqrt{|y_i|})) = 0 \quad (\text{Π.19})$$

$$h_2(x) = \frac{1}{D} \sum_{i=1}^D (-y_i \cos(0.5\sqrt{|y_i|})) = 0 \quad (\text{Π.20})$$

Και το πεδίο ορισμού:  $x \in [-600, 600]^D$

Βέλτιστη λύση:  $f_{best}^{10} = -1.883e + 03$

## Πρόβλημα 7

Στο 7<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$$

$$z = x + 1 - o, \quad y = x - o \quad (\text{Π.21})$$

Έναν περιορισμό ανισότητας :

$$g(x) = 0.5 - \exp\left(-0.1 \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2}\right) - 3 \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(0.1y_i)\right) + \exp(1) \leq 0 \quad (\text{Π.22})$$



Και το πεδίο ορισμού:  $x \in [-140, 140]^D$

Βέλτιστη λύση:  $f_{best}^{10} = 0.000e + 00$

### Πρόβλημα 8

Στο 8<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση:

$$f(x) = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right)$$
$$z = x + 1 - o, \quad y = (x - o)M \quad (\text{Π.23})$$

Έναν περιορισμό ανισότητας:

$$g(x) = 0.5 - \exp\left(-0.1 \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2}\right) - 3 \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(0.1y_i)\right) + \exp(1) \leq 0 \quad (\text{Π.24})$$

Και το πεδίο ορισμού:  $x \in [-140, 140]^D$

Βέλτιστη λύση:  $f_{best}^{10} = 0.000e + 00$

### Πρόβλημα 9

Στο 9<sup>ο</sup> πρόβλημα έχουμε την αντικειμενική προς ελαχιστοποίηση :

$$f(x) = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right)$$
$$z = x + 1 - o, \quad y = x - o \quad (\text{Π.25})$$

Έναν περιορισμό ισότητας:

$$h(x) = \sum_{i=1}^D (y_i \sin(\sqrt{|y_i|})) = 0 \quad (\text{Π.26})$$

Και το πεδίο ορισμού:  $x \in [-500, 500]^D$

Βέλτιστη λύση:  $f_{best}^{10} = 0.000e + 00$

## Παράρτημα Β

Το παράρτημα αυτό περιλαμβάνει τους πίνακες «ο» για 30 διαστάσεις, του κάθε προβλήματος. Για 10 διαστάσεις χρησιμοποιούμε μόνο τα 10 πρώτα στοιχεία της κάθε στήλης, για το αντίστοιχο πρόβλημα.

ο	P01	P02	P03	P04	P05	P06	P07	P08	P09
1	0.030859	-0.06694	111.1763	0.820202	72.109	-1.59552	-1.46823	-1.46823	-41.0325
2	-0.07863	0.470966	92.0788	5.260154	9.007674	7.633467	47.51402	47.51402	-35.7028
3	0.048651	-0.49053	417.9819	-1.69461	51.86633	-5.7641	-30.4211	-30.4211	-48.6694
4	-0.06909	-0.3122	253.1619	-5.5893	41.3657	8.103197	-7.70794	-7.70794	94.51947
5	-0.08792	-0.12476	363.528	-0.14174	93.18769	-0.05952	-21.747	-21.747	31.687
6	0.088983	-0.24782	314.3341	9.454676	74.53342	-0.73719	-17.8812	-17.8812	99.69508
7	0.074143	-0.44808	187.3274	8.795745	63.74548	-9.19086	5.274442	5.274442	30.77828
8	-0.08653	0.326495	240.4363	9.687346	7.496986	4.220874	18.71404	18.71404	-31.0412
9	-0.02062	0.493436	422.6009	-3.24652	56.1673	-1.74544	-36.9597	-36.9597	-46.2101
10	0.055586	0.0617	327.6304	6.6474	17.71631	9.499045	-20.7295	-20.7295	27.2619
11	0.059286	-0.30251	62.04763	1.43449	28.00966	3.820686	25.4702	25.4702	-2.09362
12	-0.04067	-0.27405	25.43566	-0.50653	29.36358	2.569335	-25.44	-25.44	22.24627
13	-0.0874	-0.43297	360.5677	0.558594	26.96665	9.354368	1.054563	1.054563	-42.8874
14	-0.01843	0.062239	154.9227	7.919942	6.89219	-0.85211	-31.5566	-31.5566	89.88377
15	-0.00518	-0.18816	33.16129	1.383716	44.29071	4.714177	-19.3204	-19.3204	-6.73152
16	-0.03989	-0.10071	177.8092	-1.52015	84.35804	6.775421	17.16774	17.16774	97.86439
17	0.036509	-0.33353	262.582	-2.26674	81.16907	-9.0742	34.66537	34.66537	49.49994
18	0.026046	-0.49663	436.9801	6.48053	92.76919	-3.76065	-31.8037	-31.8037	23.2107
19	-0.06713	-0.28865	476.6401	-8.89321	3.826058	-0.77806	-12.9269	-12.9269	-81.3672
20	0.08278	0.435648	331.2168	-3.52874	7.231865	-7.48701	25.48969	25.48969	-20.1569
21	-0.04934	-0.34826	75.20595	6.063486	14.44607	4.435062	-45.23	-45.23	36.69216
22	0.018503	0.45655	484.3362	-4.51585	46.49943	-6.95271	36.31775	36.31775	44.37409
23	0.051611	-0.28684	258.4696	7.320478	22.15572	-8.75233	-18.3869	-18.3869	-15.9845
24	0.018613	0.145639	419.892	-8.99026	69.11724	-2.41133	34.86816	34.86816	-49.6839
25	0.093449	-0.03866	357.5147	9.446412	88.99629	-6.14989	-37.5307	-37.5307	98.37156
26	-0.07121	0.333292	166.3772	-6.41069	58.74824	1.049303	19.28885	19.28885	0.127593
27	-0.03654	-0.29369	47.59456	-9.13525	52.26537	-6.04909	0.684612	0.684612	61.70991
28	-0.03126	-0.34786	188.2061	2.077638	47.03012	0.950328	-12.6368	-12.6368	-84.019
29	0.099244	-0.0893	184.7965	8.051026	53.23322	1.443084	15.00545	15.00545	-35.3957
30	0.053872	0.142027	267.9201	-1.00269	5.778976	-0.16383	-40.4687	-40.4687	-5.14398

## Παράρτημα Γ

Το παράρτημα αυτό περιλαμβάνει τους πίνακες «*M*» για 10 και 30 διαστάσεις, των προβλημάτων C06 και C08.

Πρόβλημα 6 για 10 διαστάσεις:

<b>M</b>	1	2	3	4	5	6	7	8	9	10
1	-0.42791	0.333003	0.158499	0.278766	0.222299	-0.29475	0.370058	-0.50145	-0.28015	-0.00065
2	0.224165	-0.26171	0.461868	0.088426	0.181817	0.373897	0.129585	-0.3816	0.299801	0.484739
3	0.535378	-0.34692	-0.24519	0.616143	0.041842	-0.15958	0.105948	-0.0483	-0.32808	-0.06971
4	-0.41697	-0.38848	-0.09422	-0.0044	-0.23754	-0.08798	0.123623	0.287284	-0.33088	0.628376
5	-0.17311	-0.41173	-0.49943	-0.21982	-0.13415	0.107615	0.438599	-0.36754	0.291863	-0.24605
6	-0.23883	-0.37713	-0.03657	-0.11532	0.298746	0.11989	-0.65202	-0.3588	-0.3267	-0.14792
7	-0.43096	-0.22592	0.119112	0.573035	0.085368	-0.09038	-0.15829	0.252439	0.529634	-0.18954
8	0.038944	0.08206	-0.40797	-0.11615	0.807124	-0.10471	0.075052	0.217165	0.152079	0.271072
9	0.158807	-0.01482	-0.0815	-0.115	-0.20314	-0.72638	-0.31013	-0.31328	0.332761	0.284154
10	-0.08175	0.423797	-0.5045	0.343679	-0.22825	0.409152	-0.27458	-0.21262	0.090751	0.304532

Πρόβλημα 8 για 10 διαστάσεις:

<b>M</b>	1	2	3	4	5	6	7	8	9	10
1	-18.7858	33.61695	26.88211	-10.4331	0.944893	-3.3539	3.535213	7.394277	7.790909	2.091292
2	-0.38081	10.42097	9.347292	-20.9265	11.4259	1.105637	36.87928	-1.91034	7.561168	-9.74307
3	-12.3417	6.362199	8.24843	8.089256	0.069235	2.578624	-0.49734	-2.06272	1.430205	15.522
4	-17.0065	-12.6793	51.65812	-3.97661	3.934938	-30.7772	6.146597	-11.405	12.69421	-9.39514
5	-5.48475	-13.6435	-20.8126	12.48063	0.844978	24.83039	33.83851	-17.0036	-5.29396	26.0657
6	11.42288	10.22146	-5.99948	-8.93589	3.340792	3.924549	-6.76057	14.0163	2.353397	-15.9574
7	14.10698	-0.6898	25.92836	-30.1383	12.95307	-17.1258	19.1229	3.85021	14.44987	-37.7686
8	1.817162	-4.5229	2.596065	-3.07797	3.666238	-3.14227	-1.9391	-1.13285	-1.4594	-4.38507
9	17.05964	-40.8873	-9.04137	9.207817	2.483559	-3.13524	-0.51597	-10.4482	-3.97908	-5.01015
10	-2.10041	4.285743	18.1388	-5.56916	0.020415	-5.53157	1.750746	2.018382	8.967371	-3.59365

Πρόβλημα 6 για 30 διαστάσεις (οι πρώτες 15 στήλες):

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.023071	-0.24354	0.187927	0.104249	0.260717	0.221583	-0.2528	0.100937	-0.21137	0.202098	-0.10988	0.145097	0.085949	-0.14642	0.190867
2	0.219814	0.281621	-0.13981	-0.11042	0.111376	0.189503	-0.19726	-0.12057	0.150327	0.18144	-0.16038	0.119334	0.083422	0.205356	0.070742
3	-0.00456	-0.18501	0.301301	-0.10922	0.056649	-0.31926	-0.12195	-0.11495	0.299113	0.245003	0.095159	-0.05566	-0.01503	0.179092	-0.13912
4	0.084637	0.010235	0.016351	-0.29132	0.139924	0.208456	0.078963	0.12645	0.16592	0.278689	-0.09832	0.005769	-0.09268	-0.08945	0.094755
5	0.074312	-0.22557	0.082166	-0.03321	-0.39515	0.237532	0.190561	0.246872	0.290084	-0.16738	-0.36414	0.077714	0.173587	-0.06768	-0.21456
6	0.059764	0.280449	0.168027	0.107899	0.078129	0.016726	-0.06886	0.256062	0.354673	0.094803	-0.07036	-0.17171	-0.21786	0.101784	-0.318
7	-0.36634	-0.16641	-0.32407	0.22142	-0.01957	0.232912	-0.02912	-0.18649	0.397519	0.175188	0.051853	-0.07922	-0.16643	-0.02422	-0.02379
8	-0.08934	-0.15064	0.333978	-0.02329	0.071484	0.063242	-0.0078	0.194455	0.086167	0.192374	0.350398	0.02974	-0.01875	-0.03019	-0.07198
9	0.074888	0.083676	-0.08318	-0.06664	0.053095	-0.31686	-0.2944	-0.2444	0.017207	0.124338	-0.38575	0.297329	-0.23015	-0.2532	-0.20808
10	0.145179	0.29631	-0.20792	-0.01498	-0.37631	-0.07973	0.039301	0.280141	-0.04984	0.338345	0.112751	-0.20015	0.194011	0.039647	0.253987
11	-0.24777	-0.01303	0.208366	-0.01308	-0.42085	-0.01559	0.024416	-0.15758	0.000938	0.114839	0.152749	0.400531	-0.09959	0.046119	-0.08406
12	0.029335	0.424795	0.024507	0.119627	0.185031	0.174363	-0.13618	-0.05358	0.158112	-0.2349	0.140992	0.167718	-0.14723	0.025589	0.162073
13	-0.11841	-0.18657	0.104726	0.425554	-0.14202	0.23065	-0.25901	-0.12347	-0.02136	0.004901	-0.04669	-0.24916	-0.08573	0.066569	0.051855
14	-0.08586	-0.04421	-0.02419	-0.29446	-0.13655	0.190632	0.009078	-0.39138	-0.12687	0.342115	0.09117	-0.04715	0.059107	-0.17574	0.153264
15	-0.17406	0.2912	0.26645	-0.02889	-0.00187	-0.02838	0.394089	-0.19473	0.084776	0.032277	-0.24859	-0.2737	-0.17048	-0.26876	0.063101
16	-0.12254	0.055062	0.19226	0.111779	-0.09711	-0.06064	-0.19096	0.162097	-0.08981	0.286608	-0.46382	0.065668	0.110707	0.195571	0.127761
17	0.149871	-0.05398	-0.27229	0.212023	0.043391	-0.21261	0.27361	-0.22221	0.303226	0.137454	0.052722	0.10388	0.249854	0.062047	-0.02348
18	-0.02591	-0.01998	0.01211	-0.21376	0.03151	0.105961	-0.09919	-0.14513	-0.1876	-0.22016	-0.16852	-0.19964	0.286317	0.038597	-0.38976
19	-0.12186	-0.01465	0.035376	-0.21278	0.006149	0.02196	0.25686	0.213074	0.045473	-0.04294	0.005139	0.418561	-0.04319	-0.14749	0.055288
20	-0.12612	0.31319	0.134166	-0.13054	-0.11659	0.085444	-0.26995	-0.07868	0.084414	0.000298	0.241097	-0.01385	0.283258	-0.35605	-0.20052
21	0.213961	0.168198	0.318236	0.328503	-0.092	0.128969	0.146856	-0.13345	0.000946	0.049031	-0.00869	0.273316	0.186258	0.121673	0.10412
22	0.373877	-0.08712	-0.02389	0.077805	-0.01787	0.263588	0.065576	-0.28238	0.048	0.042279	0.013088	0.111191	0.198511	-0.01438	-0.20822
23	-0.12841	0.016661	0.111244	0.239761	0.068309	-0.20356	0.158469	-0.16035	-0.03092	-0.19153	-0.12216	0.103567	0.043537	-0.12306	0.252735
24	0.153674	-0.01102	0.202464	-0.02705	0.176263	-0.04738	0.085514	-0.05132	-0.0845	0.01023	0.236456	0.083744	-0.0237	0.25643	-0.17306
25	0.146904	-0.04164	-0.06069	-0.01485	-0.1989	-0.18985	0.126416	-0.09389	-0.27476	0.192563	-0.00956	-0.06022	-0.30412	0.228203	-0.157
26	-0.06087	0.146556	-0.19448	0.207496	-0.30527	-0.08666	-0.24817	0.147831	-0.12134	-0.0703	0.135924	0.199771	-0.07368	-0.10655	-0.19611
27	0.268415	-0.24395	-0.16604	-0.09907	-0.02373	0.061484	-0.11747	0.130389	0.151702	-0.06694	0.03572	0.161868	-0.2782	-0.14833	0.141598
28	0.02919	0.137996	-0.05267	0.082574	0.019676	0.406627	0.245142	-0.01155	-0.30905	0.120019	-0.02063	0.051225	-0.38034	0.051627	-0.24777
29	0.228679	-0.02965	0.210661	-0.2689	-0.35402	0.05882	-0.18577	-0.22765	0.155124	-0.32561	0.005404	-0.12035	-0.24943	0.19352	0.317258
30	-0.45498	0.081029	-0.15455	-0.24531	0.070739	0.114407	-0.0008	-0.02959	0.034842	-0.09038	-0.08725	0.201659	0.107556	0.538786	0.027361

Πρόβλημα 6 για 30 διαστάσεις (συνέχεια):

M	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0.200214	-0.2166	0.020808	0.116862	0.125705	-0.12742	0.154492	-0.02488	-0.13065	0.041242	0.137578	-0.11369	0.080918	0.231135	-0.50069
2	-0.01031	0.165894	0.286427	0.24255	-0.25745	-0.12484	-0.28471	-0.26605	0.149408	0.234265	0.173236	-0.2689	-0.03922	-0.07697	0.007846
3	0.015458	-0.47389	0.248799	0.082953	0.113131	-0.14901	0.130435	-0.13561	0.113089	0.036204	0.087605	0.120849	0.199406	-0.03681	0.27389
4	-0.20754	-0.03033	-0.33812	0.166208	0.345469	0.232248	-0.04408	0.19399	0.125847	-0.08988	-0.14662	-0.377	0.236892	-0.02921	0.196331
5	0.177972	-0.15676	0.218777	-0.08246	-0.02193	-0.07702	-0.05374	0.30252	0.010605	0.213434	-0.04053	-0.13051	-0.04266	-0.11933	-0.04718
6	-0.36649	-0.06209	-0.24855	0.021171	-0.15632	-0.13497	0.066948	0.003209	-0.27796	-0.03886	0.0549	0.085161	-0.1441	-0.04648	-0.33696
7	0.189549	0.250048	-0.02019	0.18122	0.133994	-0.1511	0.332314	0.058249	0.17405	-0.06823	0.116051	0.031831	-0.1739	-0.03533	0.006553
8	0.301158	0.171693	0.082918	0.270691	-0.01645	0.239236	-0.44725	0.023841	-0.13108	-0.09076	-0.15837	0.136668	-0.30976	-0.10989	0.009151
9	0.144354	0.064305	0.134789	-0.13892	0.116963	0.053628	-0.01583	0.127418	-0.27897	-0.22956	-0.11737	-0.11041	-0.22862	0.006442	0.093722
10	0.169124	-0.14764	0.090495	0.038999	0.044136	-0.15507	0.044689	0.121063	-0.17316	-0.36218	0.168985	-0.0508	-0.12493	0.154932	0.107529
11	0.037524	0.082945	-0.25043	-0.00311	-0.30863	-0.24996	-0.10378	-0.06581	0.008064	-0.20928	-0.05986	-0.23907	0.326049	0.120855	-0.08151
12	0.167887	-0.16359	0.076857	-0.05659	0.131435	-0.24945	-0.20362	0.423979	0.171047	-0.05327	-0.18071	0.264203	0.195835	0.05489	-0.01577
13	-0.37591	-0.01575	0.242445	-0.20969	0.101361	0.125634	-0.35061	0.062574	-0.06537	-0.15299	0.115992	-0.15139	0.047882	0.195175	0.150178
14	-0.24173	-0.21003	-0.07998	-0.2224	-0.02958	-0.17368	-0.13447	0.152816	-0.02685	0.177059	-0.03835	0.205436	-0.25859	-0.3162	-0.14182
15	0.290307	0.012768	-0.06294	-0.09876	0.134131	0.056937	-0.21215	-0.20707	-0.00839	0.115821	0.331791	0.03187	0.084758	0.166163	-0.0466
16	0.022293	0.096276	-0.21429	-0.06873	-0.14938	0.176925	0.039653	0.057846	0.428075	0.0177	-0.0886	0.375102	-0.10358	0.107912	0.082089
17	-0.03341	-0.19685	0.007222	-0.08938	0.014971	0.310982	-0.19028	-0.06552	0.188463	-0.15983	-0.18024	-0.00996	-0.00521	0.157429	-0.43492
18	0.012555	-0.01594	-0.08738	0.235992	0.159457	-0.16582	-0.16841	-0.0283	0.230888	-0.5114	0.148307	0.088388	0.002856	-0.12234	-0.13124
19	-0.43742	0.114964	0.309588	0.053646	0.217877	-0.14869	-0.00474	-0.13187	0.121987	-0.11244	0.13312	0.211878	-0.20121	0.315903	-0.00383
20	-0.07957	0.055072	0.203995	0.015634	-0.14433	0.437697	0.297726	0.155347	0.091017	0.064175	0.131924	-0.00948	0.127437	0.102566	-0.07268
21	-0.03769	0.181218	0.047505	-0.06713	0.326025	0.07346	0.227094	-0.14174	-0.11938	-0.14725	0.088613	0.04612	0.054086	-0.49139	0.008515
22	0.015231	0.018334	-0.28275	0.168979	-0.02568	-0.02665	0.003722	0.028352	-0.27914	0.139205	0.058469	0.303326	-0.01031	0.429095	0.323048
23	-0.17253	-0.26696	-0.07774	0.521721	-0.26934	0.056442	0.010189	0.293649	-0.01603	-0.04242	0.191166	-0.14076	-0.21734	-0.11136	0.11342
24	0.09521	0.045761	-0.1244	-0.34642	0.023768	-0.04104	0.068357	0.291246	0.286275	0.036849	0.353118	-0.33181	-0.38335	0.143642	0.027584
25	-0.06904	0.229571	0.204871	0.305087	0.148995	0.088976	-0.02095	0.388158	0.002435	0.194961	0.13884	0.147179	0.254255	0.006343	-0.26362
26	0.015107	-0.27507	-0.26305	0.13074	0.389134	0.063145	-0.18406	-0.22286	0.149047	0.327616	0.131465	-0.04037	-0.14927	-0.0677	0.007104
27	0.061627	-0.06027	-0.1117	-0.14057	-0.23233	0.222724	-0.09993	-0.01101	0.064347	-0.22188	0.527073	0.248443	0.171747	-0.21383	-0.0035
28	0.078995	-0.38098	0.199753	0.04104	-0.20386	0.198406	0.184526	-0.16248	0.123244	-0.16948	-0.16801	-0.00735	-0.09288	-0.00833	0.087329
29	0.035244	-0.02956	-0.03391	0.141764	0.109239	0.153433	0.165976	-0.14024	0.045756	-0.05682	-0.15976	-0.0306	-0.26089	0.174126	-0.18406
30	0.068052	-0.1627	-0.0277	-0.06267	0.088926	0.287079	-0.02693	0.059357	-0.38945	0.022368	0.192399	0.050439	0.023364	0.017553	-0.01324

Πρόβλημα 8 για 30 διαστάσεις (οι πρώτες 15 στήλες):

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2.51119	8.257741	-4.57697	11.66305	1.839271	-3.93048	6.381869	5.554164	3.79226	-8.11958	-6.74987	-4.16745	-5.24723	4.380091	1.636265
2	-1.17625	0.194672	9.262322	-4.87762	-3.01383	4.138941	-4.3814	-1.0956	3.362554	-2.41518	-6.86161	-0.00594	2.135933	-0.30728	9.502955
3	-1.30858	-2.05037	-3.87496	-2.13602	0.425374	3.043654	2.852464	-4.60969	-2.67699	0.018654	4.184325	-2.94145	-9.07025	-2.89017	-6.51709
4	-0.6913	-1.99453	-8.60475	-2.97611	-2.54628	0.130893	-2.09526	3.356968	0.259979	5.401823	3.061296	1.403978	3.459626	0.875145	-4.03787
5	-0.44423	-3.57139	13.28605	-12.8396	-4.48968	7.869014	-1.96511	-8.63664	-0.09482	-10.6735	-12.5316	-1.85498	-4.84583	-3.30319	14.50411
6	-10.4643	-0.18582	6.552041	-5.91195	8.168938	-10.9822	-3.00136	2.912066	2.671938	-3.25656	-1.01098	-8.18917	-12.0016	0.565141	-6.81613
7	-5.78112	-4.69543	9.384139	-7.57521	8.514278	-5.72063	-3.82849	5.909989	0.148908	-0.63119	-7.57208	-3.77056	-8.17858	-5.09826	-12.0711
8	2.268818	7.300452	-4.33158	10.74429	0.150493	-0.5227	6.686183	3.78222	2.188704	-8.96628	7.937093	-1.31567	-3.39522	1.559847	2.788835
9	0.041055	4.198164	0.380892	-8.23942	4.779958	-3.47879	-4.34023	-1.26962	6.14681	-3.07186	0.971338	-5.89612	-0.75755	0.878201	-1.53252
10	-4.75881	-1.2491	5.230242	-19.4424	3.990166	-5.21679	-4.96012	2.113665	0.365247	3.670942	-1.91429	-6.09433	-2.68297	1.227422	-9.43365
11	5.895714	5.673958	-7.55119	8.676366	-8.74213	7.876941	6.186315	-1.38529	2.339351	-7.3034	-0.62035	-3.2148	-0.52856	-0.32765	6.600177
12	5.050755	-1.01508	-4.48895	3.071909	-6.30921	4.607172	0.518772	2.576208	-2.53895	3.87117	0.860998	9.333193	3.898242	2.045637	-4.63217
13	-2.29674	-7.50769	3.878804	1.927871	8.606052	-4.90225	-0.36942	3.771106	-4.49724	6.920644	6.555607	3.979481	0.55167	-1.25038	-11.5779
14	-3.93752	-3.13993	7.063209	1.809248	8.870248	-7.11628	-0.56157	1.607624	-0.44298	-5.39263	3.253313	-0.5391	-7.50093	0.980573	-5.63425
15	-2.68024	0.356004	-0.41446	-0.06776	5.320643	0.737875	1.213668	-3.14581	-2.5077	2.536104	3.558431	2.131159	3.764201	0.816204	-4.63296
16	6.578208	0.81841	-9.28886	22.14414	-5.22327	1.804141	6.924767	-2.77678	-2.49438	4.743636	2.690822	6.088855	0.981039	-0.12367	-0.14013
17	-4.30243	1.030219	-0.0354	-3.8646	11.18171	1.799129	-4.63238	-2.78099	-2.53132	10.16555	1.550623	5.402783	7.066734	3.559465	-7.83851
18	-5.53299	1.368158	-2.99628	1.180588	-1.17274	-3.03344	0.240827	1.274555	3.664071	1.450239	0.437749	-5.13912	-2.18424	-1.72267	5.003903
19	4.749689	-3.0131	-0.89647	-5.40441	3.2519	4.700501	-0.48584	2.12902	-5.74399	5.050727	5.290427	10.01707	4.109783	-2.17159	-9.14397
20	4.559035	0.191335	-9.7247	16.67644	-11.8152	-2.96362	2.644195	-3.50453	1.539474	3.247953	4.136178	-1.41504	0.180777	-2.51679	6.160829
21	-2.55131	-5.62476	5.82275	-18.8398	2.287665	3.202651	-0.14316	-1.93934	-7.29749	4.714811	-2.28592	1.571915	1.188689	0.665125	-3.62112
22	-3.86129	-4.67545	9.251216	-23.6078	-8.69557	-2.98908	-4.73944	2.835678	-0.09817	-12.5994	-10.3717	-6.88027	-13.3235	0.836643	5.174041
23	4.161137	1.482038	1.287113	-11.7806	-13.7044	2.252971	-4.62898	-8.44853	4.501242	3.218082	-9.09669	-6.35977	2.458767	2.74724	11.35283
24	2.105311	0.544817	1.846346	-0.50978	-8.1914	-0.22685	-3.08518	-5.69921	3.293594	-2.90944	-7.76496	-1.1669	3.020314	6.799167	10.77465
25	4.460614	3.195298	-8.19032	17.72093	1.079904	4.032047	6.846171	-2.99092	-1.01385	4.21055	0.764711	6.307647	6.791811	1.358858	-1.62461
26	0.112995	0.371937	-7.02557	9.769014	-0.64079	-11.6771	1.451573	1.171502	1.692648	-0.88121	4.268283	-5.63746	-4.89745	0.171736	-0.92113
27	3.67212	2.626628	-6.29527	-1.06986	-3.81133	1.594901	-0.82265	-2.9961	0.011411	3.804974	0.29096	0.154413	6.009124	1.048841	0.114205
28	-5.81855	-0.91693	-0.36661	-5.12995	6.905064	-6.54458	-7.1622	-1.21839	0.028387	13.30466	5.212031	-0.19841	2.043938	-0.29548	-9.73877
29	3.486879	2.633939	-3.83454	6.338662	0.084402	0.891537	7.386752	-3.14334	0.175696	-1.18984	2.063746	-3.87374	7.631037	0.870408	5.416923
30	2.461945	-0.33673	-0.5524	-8.45267	-5.63384	4.461702	-3.77916	2.769246	-1.0068	1.372993	-2.88376	3.935345	1.588808	1.666333	-3.34511

Πρόβλημα 8 για 30 διαστάσεις (συνέχεια):

M	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1.545744	6.729651	-10.8417	5.271334	-7.31514	5.586455	0.13436	-4.75795	-7.49406	-5.71423	-3.49497	16.86892	-6.0586	-11.3534	-1.09236
2	3.789221	-2.76964	2.672717	-3.37412	-2.04111	-3.34025	-0.8375	-5.25905	-1.52149	3.560245	0.587849	-8.24966	1.577881	2.563924	1.537959
3	-6.45822	5.214758	-5.67283	-1.29183	-0.42524	2.004257	7.010207	3.245853	0.975852	1.406363	-2.05246	2.872079	2.445851	1.108925	-0.26063
4	2.698141	-1.04648	-0.16212	3.972121	-0.3601	-3.59043	1.135745	6.085551	-0.3066	-2.34958	3.90394	-1.48182	-2.8784	7.258504	4.796645
5	2.590045	-6.13569	-4.2709	-6.86168	-2.74487	-5.10268	5.943125	-8.76697	2.000662	0.482206	7.450887	-10.4501	3.625933	4.179178	3.982477
6	7.034213	12.40601	-14.5377	1.401546	0.070761	5.173019	17.50033	3.097673	-4.02664	3.007819	-6.98679	6.394196	3.687826	-10.5009	-13.1519
7	1.248738	0.478544	-8.67883	-0.94621	-0.78973	-8.64978	2.48542	12.15349	-3.06224	-0.17037	-2.09352	-5.15404	4.508278	1.610648	-3.27488
8	-2.13899	8.323858	1.10335	2.080996	3.671908	10.73019	-0.78657	-12.0771	-3.20048	1.964681	-2.657	12.54941	-5.04805	-8.9256	-3.29108
9	5.844372	-0.83701	-3.90728	-2.03017	3.202677	1.470137	6.858775	-5.75169	2.261109	5.446233	-0.99458	-4.19807	-0.91628	-4.34177	0.660003
10	4.966438	-3.61691	-0.14555	-2.28039	2.265409	-5.31782	4.356959	9.751395	2.184418	4.551729	-3.20394	-8.16188	9.881965	-0.94268	-4.09076
11	-4.03048	-2.7482	1.13172	4.041615	-3.80728	7.174795	-7.10591	-16.1682	-8.49074	-1.20528	-3.21378	10.1946	-8.15377	-3.34614	6.852577
12	-3.25105	-10.3408	9.938363	2.681875	-0.30929	-3.5538	-11.9092	3.407796	-0.85223	1.470544	0.018651	-0.40094	1.006604	5.546568	8.936287
13	0.989494	4.643449	0.039903	2.717889	2.628829	-1.3911	2.040535	14.5719	1.6458	-0.60113	-0.8404	-1.4632	3.202377	-1.67118	-3.37598
14	2.236908	3.147823	-5.93776	0.887239	3.778663	4.579852	9.682244	0.005008	-2.06661	-2.8401	1.07267	6.10712	-2.01482	-7.16582	-9.27207
15	0.18412	1.217433	-4.03607	2.258557	-0.91008	3.754187	7.457651	1.456847	-1.60861	-0.75765	3.790322	0.88704	-6.02779	-0.32819	0.866003
16	-13.7979	-1.91403	1.356053	2.485841	-2.6693	2.28937	-12.5996	0.938473	-3.00075	-6.74846	-2.99899	7.509274	-4.71279	2.583974	4.259407
17	0.872795	-6.7294	1.851944	-3.17302	0.303477	-7.29433	0.431243	12.10855	2.785512	3.951423	5.216572	-14.0181	3.680618	10.31253	9.072426
18	1.555677	4.592628	0.03153	1.962039	1.311519	2.97144	4.358629	-3.02091	-2.02629	-2.1791	0.199561	-0.64227	-0.4843	0.867697	-3.42826
19	2.158097	-4.2883	8.440797	0.622934	0.366858	-5.37267	-6.42221	12.41022	3.749357	2.769913	3.508222	-5.74017	6.606652	5.630678	6.253155
20	-6.79458	0.075818	0.727367	1.668306	1.253458	7.270581	-5.02068	-10.7103	-3.48231	-3.15996	-5.26423	7.244642	-10.6411	-0.02802	0.21957
21	7.315055	2.592081	-3.60497	-1.26925	-4.60363	-6.18686	8.62584	15.64062	9.339712	0.700565	8.033185	-8.98457	8.756788	3.000295	1.552702
22	10.49264	6.802053	-6.11617	0.612168	-1.56168	-2.96102	14.55563	-2.40892	4.164763	-1.28854	0.524922	-0.41974	6.847307	-2.41781	-7.70289
23	1.690072	-4.94465	1.287478	-5.51447	-0.43632	-2.46089	-2.76687	-5.24426	3.839765	1.974487	-3.85047	-5.71918	3.112555	2.191999	-0.13026
24	1.525039	-7.55423	-0.58701	-1.30076	-4.08072	-2.18647	-2.40648	-6.95805	-3.23575	-4.30423	0.206371	-2.62503	-5.71531	1.37559	7.313963
25	-10.1503	-8.95509	2.411296	2.991836	-2.08411	-0.38788	-12.0379	0.784781	-3.51259	-6.35805	2.354271	1.762979	-7.50893	4.153957	8.430589
26	-2.49051	9.607047	-7.2068	4.349648	4.621339	10.83236	5.100627	-4.07613	-0.93278	0.502655	-8.68582	15.29417	-5.99425	-11.6945	-11.8013
27	1.04114	-1.76649	-1.23226	3.296071	-2.37361	-1.25016	-0.72087	-0.61568	1.403259	3.005529	-1.21563	0.042094	-2.23188	1.611361	6.469735
28	4.373893	5.133821	-6.8188	2.032166	1.299068	-5.28249	8.797045	12.84989	2.787157	6.225882	-4.42939	-7.40361	3.478947	1.632891	-1.10575
29	-4.79617	-5.44148	7.11322	-4.02101	1.093338	-0.72371	-11.7789	-8.94751	0.371698	-3.36835	4.657443	0.453956	-1.7407	-1.65221	1.029318
30	9.018879	4.966344	-2.37841	4.984115	-6.36945	-4.81046	4.695632	8.870217	1.851266	0.478532	2.284171	-1.69979	4.115854	2.079682	6.064423