

Ατομική Διπλωματική Εργασία

**ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΡΓΟΥ
ΚΑΙ ΑΝΑΘΕΣΗΣ ΕΡΓΑΣΙΩΝ ΣΤΗΝ ΠΑΡΑΓΩΓΗ
ΣΥΣΤΗΜΑΤΩΝ ΛΟΓΙΣΜΙΚΟΥ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ
ΤΕΧΝΙΚΗΣ ΣΜΗΝΟΥΣ ΣΩΜΑΤΙΔΙΩΝ**

ΓΕΡΑΣΙΜΟΥ ΣΙΜΟΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Βελτιστοποίηση χρονοπρογραμματισμού έργου και ανάθεσης εργασιών
στην παραγωγή συστημάτων λογισμικού με τη χρήση της τεχνικής
σμήνους σωματιδίων**

Γερασίμου Σίμος

Επιβλέπων Καθηγητής

Ανδρέου Ανδρέας

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων
απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου
Κύπρου

Μάιος 2010

Ευχαριστίες

Φτάνοντας στο τέλος μια πολύ μακρινής διαδρομής που διήρκησε για περισσότερο από δέκα μήνες θα ήταν παράλειψη μου να μην ευχαριστήσω τα άτομα που βοήθησαν με το δικό τους τρόπο στην πραγματοποίηση του έργου που ανέλαβα. Πρώτα απ' όλα θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Ανδρέα Ανδρέου για την υποστήριξη και καθοδήγηση που μου προσέφερε κατά τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας καθώς επίσης και για την επίτευξη μιας άψογης συνεργασίας.

Επιπλέον ένα μεγάλο ευχαριστώ πρέπει να δοθεί στο διδακτορικό φοιτητή του τμήματος Κωνσταντίνο Στυλιανού, για την πολύτιμη και ανιδιοτελή βοήθεια που μου προσέφερε και που ήταν πάντα δίπλα μου σε όλη τη διάρκεια της χρονιάς. Μέσα από εποικοδομητικές συζητήσεις μου παρείχε κατάλληλες συμβουλές για απορίες που είχα και θέματα που με απασχολούσαν.

Τέλος δεν μπορώ να μην ευχαριστήσω τους γονείς μου, Παναγιώτη και Τασούλα, οι οποίοι με την αμέριστη στήριξή τους σε κάθε μου βήμα και με κάθε δυνατό τρόπο και μέσο, στάθηκαν δίπλα μου καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Το ολοένα αυξανόμενο ποσοστό αποτυχίας που παρουσιάζουν τα έργα παραγωγής λογισμικού έχει ωθήσει τις εταιρείες να αναζητήσουν τους λόγους στους οποίους οφείλεται αυτή η συνεχής αύξηση. Η παρούσα διπλωματική εργασία προσεγγίζει το συγκεκριμένο πρόβλημα από τη σκοπιά του διαχειριστή έργων λογισμικού και προσπαθεί να επιλύσει τις κυριότερες δυσκολίες που παρουσιάζονται κατά τη διάρκεια παραγωγής λογισμικού εστιάζοντας την προσοχή στις δύο κύριες δραστηριότητες που απασχολούν το διαχειριστή έργου, το χρονοπρογραμματισμό του έργου και τη στελέχωση της ομάδας ανάπτυξης.

Στην προσπάθεια επίλυσης του συγκεκριμένου προβλήματος χρησιμοποιείται ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization). Ο εν λόγω αλγόριθμος ανήκει στην ευρύτερη κατηγορία των εξελικτικών αλγορίθμων και αποβλέπει στη στοχαστική βελτιστοποίηση που βασίζεται στη συνεργασία μεταξύ των ατόμων ενός πληθυσμού. Συνεπώς έχοντας ως δεδομένα εισόδου τα στοιχεία που αφορούν τις διεργασίες ενός έργου και τα στοιχεία των διαθέσιμων υπαλλήλων και τροφοδοτώντας με αυτά τον αλγόριθμο PSO επιτυγχάνεται ο χρονοπρογραμματισμός του έργου και η στελέχωση των ομάδων εργασίας.

Στόχος είναι το τελικό προϊόν να αποτελέσει ένα σύστημα υποστήριξης λήψεως αποφάσεων που να ελαχιστοποιεί το συνολικό χρόνο διάρκειας του έργου καθώς επίσης και το συνολικό του κόστος ενώ παράλληλα με την κατάλληλη στελέχωση των ομάδων εργασίας επιτυγχάνεται η αύξηση της ποιότητας του έργου. Απώτερο στόχος φυσικά είναι η μείωση του ποσοστού αποτυχίας των έργων λογισμικού.

Περιεχόμενα

Κεφάλαιο 1	1
Εισαγωγή	1
1.1 Παρουσίαση προβλήματος.....	1
1.2 Στόχος	2
1.3 Μέθοδος εργασίας.....	3
1.4 Δομή ατομικής διπλωματικής εργασίας.....	4
Κεφάλαιο 2	5
Ερευνητικό Υπόβαθρο και Σχετική Βιβλιογραφία	5
2.1 Χρονοπρογραμματισμός έργων λογισμικού	5
2.2 Μέθοδοι επίλυσης RCPSP	9
2.2.1 Ακριβείς μέθοδοι	9
2.2.2 Ευρετικές μέθοδοι.....	10
2.2.3 Μεταευρετικές μέθοδοι	11
2.2.4 Εξελικτικές μέθοδοι.....	13
Κεφάλαιο 3	16
Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων	16
3.1 Εισαγωγή στους εξελικτικούς αλγορίθμους	17
3.2 Η αρχική έκδοση	18
3.2.1 Ο αλγόριθμος PSO.....	19
3.2.2 Μαθηματική αναπαράσταση	20
3.2.3 Γνωστικά και κοινωνικά μοντέλα του αλγορίθμου	22
3.3 Επίδραση βάρους αδράνειας	23
3.4 Περιοριστικός αλγόριθμος PSO.....	24

3.5	Ενοποιημένος αλγόριθμος PSO	26
3.6	Διακριτός αλγόριθμος PSO	27
3.7	Τοπολογία σμήνους.....	29
3.8	Ειδικοί αλγόριθμοι PSO	33
3.8.1	Δυναμικά περιβάλλοντα	33
3.8.2	Θορυβώδη περιβάλλοντα.....	34
3.8.3	Περιβάλλοντα με πολλαπλούς στόχους.....	34
3.8.4	Υβριδικοί αλγόριθμοι PSO	35
3.9	Προβλήματα.....	35
3.10	Σύνοψη	37
Κεφάλαιο 4		39
Μοντελοποίηση προβλήματος και υλοποίηση πρωτοτύπου		39
4.1	Εισαγωγή στη μεθοδολογία υλοποίησης	39
4.2	Μοντελοποίηση.....	40
4.3	Αντικειμενική συνάρτηση αξιολόγησης	43
4.3.1	Αλληλεξαρτήσεις μεταξύ διεργασιών	44
4.3.2	Πλήρωση απαιτούμενων ικανοτήτων.....	46
4.3.3	Διαθεσιμότητα υπαλλήλων.....	49
4.4	Απεμπλοκή από στασιμότητα	50
4.5	Υλοποίηση πρωτοτύπου.....	51
Κεφάλαιο 5		57
Πειραματική αξιολόγηση.....		57
5.1	Εισαγωγή στη μεθοδολογία αξιολόγησης.....	57
5.2	Ελεγχόμενα πειράματα.....	58
5.2.1	Σενάριο ελεγχόμενων πειραμάτων	60

5.2.2 Αξιολόγηση κάθε επιμέρους συνάρτησης ξεχωριστά	62
5.2.3 Συνολική αξιολόγηση χωρίς τεχνική απεμπλοκής	63
5.2.4 Συνολική αξιολόγηση με τεχνική απεμπλοκής	65
5.2.5 Δυναμική συνάρτηση αξιολόγησης.....	66
5.3 Μη ελεγχόμενα πειράματα.....	69
5.3.1 Σενάριο μη ελεγχόμενων πειραμάτων.....	69
5.3.2 Αξιολόγηση αλγορίθμων	70
5.4 Σύνοψη	73
Κεφάλαιο 6	75
Συμπεράσματα και μελλοντική εργασία.....	75
6.1 Γενικά συμπεράσματα.....	75
6.2 Μελλοντική εργασία	77
6.3 Επίλογος.....	80
Βιβλιογραφία	81

Λίστα σχημάτων και πινάκων

Κεφάλαιο 1	1
Εισαγωγή	1
Κεφάλαιο 2	5
Ερευνητικό Υπόβαθρο και Σχετική Βιβλιογραφία	5
Σχήμα 2.1. – Παράδειγμα γράφου προτεραιοτήτων διεργασιών	8
Κεφάλαιο 3	16
Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων	16
Σχήμα 3.1 – Βασικός αλγόριθμος PSO.....	21
Σχήμα 3.2. – Μοντέλα για την απόδειξη του οφέλους της κοινωνικής επίδρασης από τον Kennedy	23
Σχήμα 3.3. – Διακριτός αλγόριθμος PSO	28
Σχήμα 3.4. – Οι τοπολογίες (α) τροχού, (β) δακτυλίου, (γ) αστέρα, (δ) von Neumann ...	32
Σχήμα 3.5. – Συγκριτικός πίνακας δημοσιεύσεων για τον αλγόριθμο PSO ανά έτος	38
Κεφάλαιο 4	39
Μοντελοποίηση προβλήματος και υλοποίηση πρωτοτύπου	39
Πίνακας 4.1 – Στοιχεία έργου και επιμέρους διεργασιών.....	45
Πίνακας 4.2 – Παράδειγμα διεργασιών για αξιολόγηση των μεταξύ τους αλληλεξαρτήσεων	46
Πίνακας 4.3. – Παράδειγμα διεργασιών και υπαλλήλων για αξιολόγηση των ικανοτήτων	48
Πίνακας 4.4. – Παράδειγμα υπαλλήλων για αξιολόγηση της διαθεσιμότητας.....	50
Σχήμα 4.1. – Η αρχική οθόνη της εφαρμογής.....	52
Σχήμα 4.2. – Η οθόνη καθορισμού του αρχείου δεδομένων.....	53

Σχήμα 4.3. – Το μήνυμα λάθους σε περίπτωση μη καθορισμού του αρχείου εισόδου	54
Σχήμα 4.4 – Η οθόνη επισκόπησης των δεδομένων εισόδου	55
Σχήμα 4.5. – Η οθόνη αποτελεσμάτων	55
Κεφάλαιο 5	57
Πειραματική αξιολόγηση.....	57
Πίνακας 5.1 – Στοιχεία υπαλλήλων.	60
Πίνακας 5.2 – Στοιχεία έργου και επιμέρους διεργασιών	61
Πίνακας 5.3 – Βέλτιστα αποτελέσματα εξεταζόμενου σεναρίου	61
Πίνακας 5.4 – Βέλτιστα αποτελέσματα συναρτήσεων αξιολόγησης.....	61
Σχήμα 5.1 – Ποσοστό εύρεσης βέλτιστης λύσης με αξιολόγηση μόνο μιας επιμέρους συνάρτησης	62
Σχήμα 5.2 – Αποτελέσματα αξιολόγησης επιμέρους συναρτήσεων για τον αλγόριθμο TVIW_GB.....	63
Σχήμα 5.3 – Αποτελέσματα εύρεσης βέλτιστης και εφικτής λύσης χωρίς την τεχνική απεμπλοκής	64
Σχήμα 5.4 – Αποτελέσματα εύρεσης βέλτιστης λύσης χωρίς τεχνική απεμπλοκής και με χρήση της τεχνικής απεμπλοκής	65
Σχήμα 5.5 – Αποτελέσματα παραγωγής εφικτού χρονοπρογράμματος χωρίς τεχνική απεμπλοκής και με χρήση της τεχνικής απεμπλοκής.....	66
Σχήμα 5.6 – Αποτελέσματα αξιολόγησης για τον αλγόριθμο TVIW_GB.....	67
Σχήμα 5.7 – Σύγκριση ποσοστού εύρεσης βέλτιστης και εφικτής λύσης με στατική και δυναμική συνάρτηση αξιολόγησης	68
Πίνακας 5.4 – Στοιχεία έργου και επιμέρους διεργασιών	70
Πίνακας 5.5 – Στοιχεία υπαλλήλων	70
Σχήμα 5.8 – Αποτελέσματα εύρεσης εφικτού χρονοπρογράμματος με στατική και δυναμική συνάρτηση.....	71

Σχήμα 5.9 – Αποτελέσματα αξιολόγησης συναρτήσεων με ποσοστό μεγαλύτερο του 0.8 με στατική συνάρτηση αξιολόγησης.....	72
Σχήμα 5.10 – Αποτελέσματα αξιολόγησης συναρτήσεων με ποσοστό μεγαλύτερο του 0.8 με δυναμική συνάρτηση αξιολόγησης	72
Σχήμα 5.11 – Σύγκριση σεναρίων για παραγωγή εφικτού χρονοπρογράμματος με τεχνική TVIW	74
Κεφάλαιο 6	75
Συμπεράσματα και μελλοντική εργασία.....	75
Βιβλιογραφία	81

Κεφάλαιο 1

Εισαγωγή

1.1	Παρουσίαση προβλήματος	1
1.2	Στόχος	2
1.3	Μέθοδος εργασίας	3
1.4	Δομή ατομικής διπλωματικής εργασίας	4

1.1 Παρουσίαση προβλήματος

Ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζουν οι εταιρείες ανάπτυξης συστημάτων λογισμικού είναι ο ψηλός ρυθμός αποτυχίας των έργων παραγωγής λογισμικού. Σύμφωνα με την αναφορά CHAOS της Standish Group παρατηρήθηκε ότι για το έτος 2009 το 44% των έργων λογισμικού παραδίνονται στους πελάτες εκπρόθεσμα, υπερβαίνουν τον προϋπολογισμό του έργου και δεν περιέχουν την απαιτούμενη ποιότητα. Επίσης, το 24% των έργων ακυρώνονται πριν την ολοκλήρωσή τους ή παραδίνονται αλλά δεν χρησιμοποιούνται ποτέ. Από την άλλη, μόνο το 32% των έργων λογισμικού παραδίνονται με επιτυχία. Αυτά τα ποσοστά δείχνουν ότι ο ρυθμός επιτυχίας των έργων λογισμικού μειώνεται, ενώ ο αριθμός των αποτυχημένων έργων είναι ο χαμηλότερος της τελευταίας δεκαετίας.

Οι λόγοι που πιθανόν να οδηγούν σε τέτοιους ψηλούς ρυθμούς αποτυχίας ποικίλουν, και πολλοί ερευνητές στην περιοχή της Τεχνολογίας Λογισμικού προσπαθούν να βρουν μια

λύση ή συνδυασμό λύσεων στο πρόβλημα αυτό. Άλλοτε ο χρονοπρογραμματισμός του έργου κρίνεται ανεπαρκής και αδυνατεί να καθοδηγήσει σωστά την διαδικασία ανάπτυξης λογισμικού ενώ άλλοτε η στελέχωση των ομάδων εργασίας δεν είναι κατάλληλη με αποτέλεσμα το τελικό σύστημα να χαρακτηρίζεται ως προϊόν χαμηλής ποιότητας. Επίσης η συνεχής αυξανόμενη πολυπλοκότητα καθώς επίσης και το μέγεθος των συστημάτων λογισμικού συμβάλλουν στο ποσοστό αποτυχίας των έργων, γεγονός που οδηγεί στο συμπέρασμα ότι οι υπάρχουσες μέθοδοι δεν είναι αποτελεσματικές, αλλά αντιθέτως προμηνύουν την αποτυχία του έργου.

1.2 Στόχος

Η διπλωματική εργασία αυτή πραγματεύεται το παραπάνω πρόβλημα και επιχειρεί να βρει ένα τρόπο επίλυσης των κυριότερων δυσκολιών που παρουσιάζονται κατά τη διάρκεια παραγωγής λογισμικού εστιάζοντας την προσοχή πάνω σε δύο σημαντικές δραστηριότητες που αναλαμβάνει ένας διαχειριστής έργων λογισμικού: στο χρονοπρογραμματισμό του έργου και στη στελέχωση της ομάδας ανάπτυξης. Κατά συνέπεια, ο στόχος της εργασίας είναι να παρουσιαστεί η προσπάθεια αυτοματοποίησης αυτών των δραστηριοτήτων έτσι ώστε ο διαχειριστής έργων παραγωγής λογισμικού να επωφεληθεί με τη μείωση του χρόνου και της υποκειμενικότητας που απαιτούνται για να εκπληρωθούν αυτές οι δύο δραστηριότητες. Συγκεκριμένα, η προσέγγιση η οποία προτείνεται επικεντρώνεται πάνω στην κατασκευή της βέλτιστης σειράς εκτέλεσης των επιμέρους εργασιών ενός έργου με σκοπό να διατηρηθούν οι αλληλεξαρτήσεις μεταξύ τους, καθώς επίσης και στη σύνθεση μιας λειτουργικής και αποτελεσματικής ομάδας εργασίας με στόχο την καλύτερη αξιοποίηση των διαθέσιμων πόρων και την όσο το δυνατό καταλληλότερη κατανομή τους στις εργασίες. Στην προσπάθεια επίλυσης του εν λόγω προβλήματος επιλέγηκε η αξιοποίηση του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization). Ο PSO ανήκει στην κατηγορία των εξελικτικών τεχνικών και είναι ένας αλγόριθμος στοχαστικής βελτιστοποίησης που βασίζεται στη συνεργασία μεταξύ των ατόμων ενός πληθυσμού.

Το τελικό προϊόν, συνεπώς, προορίζεται να αποτελέσει ένα σύστημα υποστήριξης λήψεως αποφάσεων, μιας και οι πληροφορίες που παρέχει είναι απαραίτητες για τους διαχειριστές έργων και επιδιώκει την ελαχιστοποίηση του χρόνου διάρκειας και του συνολικού κόστους του έργου, και, φυσικά, την αύξηση της ποιότητας του έργου με τη σωστή στελέχωση των ομάδων, με αποτέλεσμα μια σημαντική συνεισφορά στη μείωση του ρυθμού αποτυχίας των έργων παραγωγής λογισμικού. Η χρήση του συστήματος ως σύμβουλος στη διαδικασία παραγωγής ενός έργου λογισμικού δεν σκοπεύει την αντικατάσταση του διαχειριστή έργων, της ανθρώπινης κρίσης και σκέψης του, αλλά απλά να προσφέρει περαιτέρω γνώσεις και μια διαφορετική αντιμετώπιση του προβλήματος.

1.3 Μέθοδος εργασίας

Η αποπεράτωση της διπλωματικής εργασίας διενεργήθηκε σε τρεις φάσεις. Η πρώτη φάση διήρκησε από Σεπτέμβριο μέχρι Δεκέμβριο, 2009, όπου έγινε η μελέτη επιστημονικών άρθρων και βιβλίων στην προσπάθεια κατανόησης του προβλήματος και του γενικότερου υπόβαθρου του. Το ενδιαφέρον εστιάστηκε κυρίως στον τρόπο λειτουργίας διαφόρων συστημάτων διαχείρισης έργων, καθώς επίσης και στον τρόπο με τον οποίο διάφοροι ερευνητές έχουν προσπαθήσει να επιλύσουν το πρόβλημα και τις μεθοδολογίες τις οποίες χρησιμοποίησαν. Ιδιαίτερη έμφαση δόθηκε στα ευφυή συστήματα που αναπτύχθηκαν για το σκοπό αυτό, αφού κάτι ανάλογο ενσωματώνει και αξιοποιεί η προσέγγιση που παρουσιάζεται σε αυτή την ατομική διπλωματική εργασία.

Μετέπειτα έγινε εκλέπτυνση του προβλήματος και διάσπασή του στα επιμέρους συστατικά στοιχεία με στόχο τη σχεδίαση μιας προτεινόμενης προσέγγισης-λύσης. Επιπλέον, έγινε προσαρμογή των δεδομένων στη μεθοδολογία που επιλέχθηκε να χρησιμοποιηθεί ως τεχνική επίλυσης και συγκεκριμένα στη Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimisation, PSO). Η διαδικασία αυτή διήρκησε το πρώτο δίμηνο του 2010.

Στην τελευταία φάση ακολούθησε η υλοποίηση της προτεινόμενης προσέγγισης καθώς επίσης και η συγγραφή της διπλωματικής εργασίας. Κατά τους μήνες Μάρτιο έως Μάιο, 2010, οι γνώσεις που αποκτήθηκαν στις προηγούμενες φάσεις αξιοποιήθηκαν και μαζί με

τη σχεδίαση της λύσης αναπτύχθηκε η προσέγγιση σε μορφή συστήματος υποστήριξης λήψεως αποφάσεων.

1.4 Δομή ατομικής διπλωματικής εργασίας

Η δομή της ατομικής διπλωματικής εργασίας έχει ως ακολούθως: Το δεύτερο κεφάλαιο αναφέρεται στο ερευνητικό υπόβαθρο των περιοχών του χρονοπρογραμματισμού έργων λογισμικού και της στελέχωσης ομάδων ανάπτυξης, και τη σχετική έρευνα που έχει γίνει κατά καιρούς στην προσπάθεια επίλυσης των συγκεκριμένων προβλημάτων αυτών των περιοχών. Στο τρίτο κεφάλαιο επεξηγείται λεπτομερώς ο βασικός αλγόριθμος που υιοθετήθηκε στην προτεινόμενη προσέγγιση-λύση όπως έχει προταθεί από τους δημιουργούς του καθώς επίσης και ορισμένες παραλλαγές του που έχουν αναφερθεί από διάφορους ερευνητές και έχουν εφαρμοστεί με επιτυχία σε διάφορα άλλα προβλήματα. Στη συνέχεια, το τέταρτο κεφάλαιο αναλύει τη διαδικασία που χρησιμοποιήθηκε για τη μοντελοποίηση του προβλήματος και περιγράφει τη σχεδίαση και υλοποίηση της προτεινόμενης προσέγγισης για επίλυσή του. Έπειτα, στο πέμπτο κεφάλαιο παρουσιάζονται τα πειραματικά αποτελέσματα και αναλύονται λεπτομερώς. Τέλος, στο έκτο κεφάλαιο γίνεται μια περίληψη της ατομικής διπλωματικής εργασίας και καταγραφή των συμπερασμάτων για το υλοποιημένο σύστημα. Επίσης, παρουσιάζονται κάποιες εισηγήσεις για μελλοντική εργασία.

Κεφάλαιο 2

Ερευνητικό Υπόβαθρο και Σχετική Βιβλιογραφία

-
- 2.1 Χρονοπρογραμματισμός έργων λογισμικού
 - 2.2 Μέθοδοι επίλυσης RCPSP
 - 2.2.1 Ακριβείς μέθοδοι
 - 2.2.2 Ευρετικές μέθοδοι
 - 2.2.3 Μεταευρετικές μέθοδοι
 - 2.2.4 Εξελικτικές μέθοδοι
-

2.1 Χρονοπρογραμματισμός έργων λογισμικού

Ένα έργο λογισμικού αποτελείται από έναν αριθμό διεργασιών και ορόσημων που πρέπει να διεκπεραιωθούν με μια συγκεκριμένη σειρά ικανοποιώντας περιορισμούς προτεραιότητας και διαθέσιμων πόρων. Κάθε διεργασία έχει συγκεκριμένη διάρκεια σε ανθρωπομέρες και περιορίζεται από μια προκαθορισμένη σειρά προτεραιότητας και ένα αριθμό πόρων που απαιτούνται για την ολοκλήρωσή της. Ο διαχειριστής έργων λογισμικού έχει την ευθύνη να κατασκευάσει αυτό το χρονοδιάγραμμα εκτέλεσης του έργου, τοποθετώντας τις διεργασίες και τους πόρους σε μια σειρά στο χρόνο. Αυτό όμως δεν είναι εύκολη δραστηριότητα να αναλάβει κάποιος διαχειριστής έργου λόγω του μεγέθους και πολυπλοκότητας των συστημάτων λογισμικού.

Το πρόβλημα αυτό του χρονοπρογραμματισμού ενός έργου λογισμικού θεωρείται ως ένα πρόβλημα χρονοπρογραμματισμού έργων περιορισμένων πόρων (Resource Constrained Project Scheduling Problem, RCPSP). Αυτή η κατηγορία προβλημάτων καθορίστηκε από τον Davis το 1973 [1] ως «η μέθοδος χρονοπρογραμματισμού μια σειράς διεργασιών με περιορισμένο, σε κάθε χρονική στιγμή, αριθμό πόρων κατά τη διάρκεια εκτέλεσης του έργου με τέτοιο τρόπο ώστε να ελαχιστοποιηθεί η διάρκεια εκτέλεσής του». Μια σύγχρονη διατύπωση του παραπάνω προβλήματος έχει γίνει από τους Merkle, Middendorf και Schmeck το 2002 [2], καθορίζοντας το RCPSP ως ένα πρόβλημα βελτιστοποίησης με στόχο το χρονοπρογραμματισμό των διεργασιών ενός έργου έτσι ώστε ο χρόνος ολοκλήρωσής του να μειωθεί στο ελάχιστο δυνατό ικανοποιώντας από τη μια τους περιορισμούς προτεραιότητας που υπάρχουν μεταξύ των διεργασιών και από την άλλη διατηρώντας την αξιοποίηση των διαθέσιμων πόρων στις προγραμματισμένες διεργασίες σε κάθε χρονική στιγμή μέσα στα επιτρεπτά όρια. Τα περισσότερα προβλήματα χρονοπρογραμματισμού έργων ανήκουν στην κατηγορία των NP-complete προβλημάτων, ειδικά για προβλήματα χρονοπρογραμματισμού μεγάλης κλίμακας. Το 1983, ο Blazewicz κ.α. [3] απέδειξαν ότι το RCPSP είναι μια γενική μορφή του πολύ γνωστού προβλήματος χρονοπρογραμματισμού job-shop και επομένως ανήκει στην κατηγορία των NP-hard προβλημάτων. Αυτό το είδος προβλημάτων χαρακτηρίζεται από την εκθετική αύξηση των απαιτούμενων υπολογισμών για την εξεύρεση όλων των δυνατών λύσεων, καθώς αυξάνεται το μέγεθος του προβλήματος. Συνεπώς, παρουσιάζεται ένας αρκετά ψηλός βαθμός δυσκολίας εξεύρεσης λύσης για μεγάλα έργα παραγωγής λογισμικού, γι' αυτό άλλωστε έχει συγκεντρώσει το ενδιαφέρον πολλών επιστημόνων του χώρου ανά το παγκόσμιο.

Αναλυτικά, το πρόβλημα χρονοπρογραμματισμού ενός έργου περιορισμένων πόρων μπορεί να οριστεί ως ακολούθως:

- Ο στόχος του χρονοπρογραμματισμού είναι η εξεύρεση του ελάχιστου χρόνου ολοκλήρωσης του έργου.
- Το έργο αποτελείται από N αριθμό διεργασιών και κάθε διεργασία έχει συγκεκριμένη χρονική διάρκεια d_j ($j = 1, 2, \dots, N$). Επιπρόσθετα, οι διεργασίες δεν

μπορούν να διακοπούν ή να αντικατασταθούν από άλλες αν έχει ήδη ξεκινήσει η εκτέλεσή τους, παρά μονάχα με το τέλος τους.

- Οι διεργασίες υπακούουν σε περιορισμούς προτεραιότητας. Μια διεργασία j δεν μπορεί να ξεκινήσει την εκτέλεσή της πριν κάθε διεργασία $i, i \in P_j$ (όπου P_j το σύνολο των διεργασιών από τις οποίες εξαρτάται η διεργασία j), έχει ολοκληρωθεί. Επιπλέον, η διεργασία 0 είναι η αρχική διεργασία, ενώ η $(N + 1)$ είναι η τελική διεργασία και αναπαριστούν την αρχή και το τέλος του έργου αντίστοιχα. Πρέπει να σημειωθεί ότι αποτελούν εικονικές διεργασίες αφού έχουν μηδενική χρονική διάρκεια και δεν έχουν οποιεσδήποτε απαιτήσεις σε πόρους.
- Υπάρχουν διαφορετικά είδη πόρων διαθέσιμα για την εκπλήρωση του έργου και κάθε είδος έχει περιορισμένη διαθέσιμη ποσότητα. Έστω Q είναι το σύνολο q τύπων πόρων και R_k ($k = 1, 2, \dots, q$) είναι η διαθέσιμη ποσότητα πόρων τύπου k . Κάθε διεργασία j απαιτεί διάφορους πόρους $r_{j1}, r_{j2}, \dots, r_{jq}$ όπου r_{jk} αναπαριστά την απαιτούμενη ποσότητα πόρων τύπου k από τη διεργασία j . Επιπρόσθετα, όταν μια διεργασία ολοκληρώσει την εκτέλεσή της απελευθερώνει τους πόρους που χρησιμοποιεί και μπορούν να ανατεθούν σε άλλες διεργασίες.

Το μαθηματικό μοντέλο που χρησιμοποιείται για την αναπαράσταση του RCPSP είναι το ακόλουθο:

$$\min_j S_j \quad (1)$$

έτσι ώστε

$$S_j - S_i \geq d_i, i \in P_j \quad (2)$$

και

$$\sum_{j \in N} r_{jk} \leq R_k, \text{ για } k = 1, 2, \dots, q \quad (3)$$

όπου i : η i -οστή διεργασία που εξετάζεται

S_i : ο χρόνος έναρξης της i διεργασίας

d_i : η χρονική διάρκεια της i διεργασίας

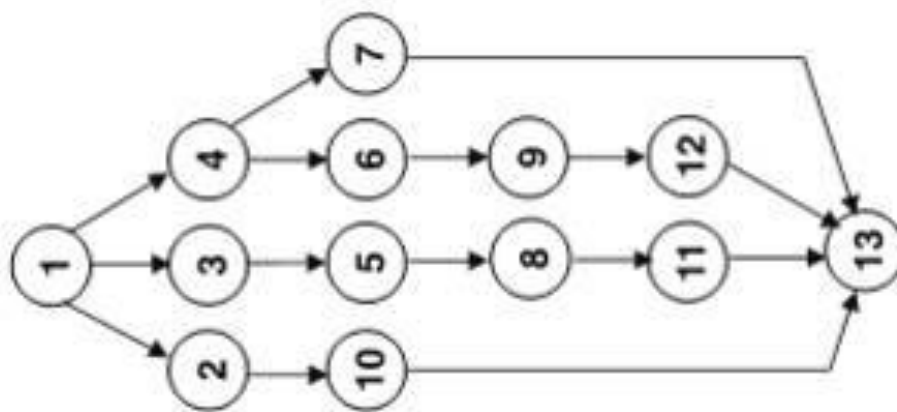
P_j : το σύνολο των διεργασιών από τις οποίες εξαρτάται η j διεργασία

r_{jk} : η απαιτούμενη ποσότητα πόρων τύπου k από τη j διεργασία

R_k : η διαθέσιμη ποσότητα πόρων τύπου k

Η Εξίσωση (1) είναι η αντικειμενική συνάρτηση, υποδηλώνοντας ότι η διάρκεια εκτέλεσης του έργου ελαχιστοποιείται, μειώνοντας στο ελάχιστο δυνατό την ημέρα έναρξης της τελικής διεργασίας, ενώ οι Εξισώσεις (2) και (3) είναι οι εξισώσεις που αναπαριστούν τους περιορισμούς. Συγκεκριμένα, η Εξίσωση (2) υποδεικνύει ότι η διεργασία j μπορεί να ξεκινήσει την εκτέλεσή της μόνο αφού όλες οι διεργασίες i , από τις οποίες εξαρτάται έχουν ολοκληρωθεί και η Εξίσωση (3) περιορίζει την κατανάλωση των πόρων σε κάθε χρονική στιγμή ώστε να μην υπερβαίνει τη μέγιστη διαθέσιμη ποσότητα.

Τα παραπάνω μπορούν να αναπαρασταθούν γραφικά με τη χρήση ενός γράφου προτεραιοτήτων διεργασιών (Task Precedence Graph, TPG), όπως φαίνεται στο Σχήμα 2.1. Ένας TPG αναπαριστάται ως $G = (V, E)$, όπου το σύνολο των κόμβων V υποδηλώνει τις διεργασίες ενώ το σύνολο των ακμών E παρουσιάζει τους περιορισμούς προτεραιότητας αρχής-τέλους με μηδενική χρονική καθυστέρηση. Αν (i, j) είναι μια ακμή του γράφου τότε η διεργασία i πρέπει να ολοκληρωθεί πριν να ξεκινήσει την εκτέλεση της η δραστηριότητα j .



Σχήμα 2.1. – Παράδειγμα γράφου προτεραιοτήτων διεργασιών

2.2 Μέθοδοι επίλυσης RCPSP

Η εμφάνιση και ο ορισμός του RCPSP, έχει κινήσει το ενδιαφέρον αρκετών ερευνητών και έχει γίνει ένα πολύ δημοφιλές θέμα έρευνας όσον αφορά τα NP-hard προβλήματα βελτιστοποίησης. Τα τελευταία είκοσι χρόνια έχουν πραγματοποιηθεί αρκετές έρευνες από την επιστημονική κοινότητα με αποτέλεσμα να έχει γίνει τρομακτική βελτίωση στους αλγορίθμους που αξιοποιούνται στην προσπάθεια επίλυσής του RCPSP. Οι εν λόγω αλγόριθμοι μπορούν να ταξινομηθούν σε ακριβείς, σε ευρετικές και σε μεταευρετικές μεθόδους καθώς και σε εξελικτικούς αλγορίθμους. Στα επόμενα υποκεφάλαια αναλύονται οι μέθοδοι παρουσιάζοντας την προσέγγιση και τον τρόπο με τον οποίο προσπαθούν να επιλύσουν το RCPSP διάφοροι ερευνητές αυτής της περιοχής.

2.2.1 Ακριβείς μέθοδοι

Πρόκειται για μεθόδους που υλοποιούν μαθηματικές προγραμματιστικές τεχνικές και τεχνικές απαρίθμησης όπου εφαρμόζονται κυρίως ο δυναμικός προγραμματισμός [4], ο 0-1 (δυαδικός) προγραμματισμός [5] και η ευρέως διαδεδομένη μέθοδος κλάδου φράγματος [6, 7, 8, 9, 10]. Οι Demeulemeester και Herroelen [6, 7] χρησιμοποίησαν τη μέθοδο κλάδου φράγματος μοντελοποιώντας τον κάθε κόμβο του δέντρου απαρίθμησης ώστε να μην αναπαριστά μια διεργασία που έχει προγραμματιστεί, όπως συνήθως, αλλά το σύνολο των διεργασιών που έχουν καθυστερήσει. Η βασική απαρίθμηση ενισχύεται με τη δυνατότητα κλαδέματος ενός μέρους του δέντρου απαρίθμησης αφού χρησιμοποιεί δύο κανόνες κυριαρχίας, πρώτον ότι ο ισχυρότερος κόμβος βρίσκεται πάντοτε στα αριστερά και δεύτερον χρησιμοποιώντας ένα σύνολο που περιέχει όλες τις μη προγραμματισμένες διεργασίες των οποίων οι εξαρτώμενες διεργασίες ανήκουν στο μέχρι τώρα μερικώς καθορισμένο πρόγραμμα ενώ η διαδικασία φραγμού επιτυγχάνεται με τη χρήση προκαθορισμένων κανόνων.

Αντίστοιχα ο Brucker κ.α. [9] παρουσίασαν ένα αλγόριθμο κλάδου φράγματος όπου στις διακλαδώσεις του δέντρου εφαρμόζεται ένα σύνολο συζεύξεων και διαζεύξεων μεταξύ ζευγών δραστηριοτήτων. Η λύση την οποία προτείνει ο Mingozzi κ.α. [8] χρησιμοποιεί μια

νέα μαθηματική διατύπωση που αξιοποιεί ορισμένα νέα κάτω φράγματα και νέα συστήματα απαρίθμησης. Οι προαναφερθείσες υλοποιήσεις μαζί με αυτή του Sprecher [10] αποτελούν τις πιο ανταγωνιστικές και αποτελεσματικές ακριβείς μεθόδους που υπάρχουν και χρησιμοποιούνται σήμερα. Αξιόλογες προσπάθειες έχουν παρατηρηθεί πρόσφατα όπως από τον Jalilvand κ.α. [11] όπου έγινε κατορθωτό να βρεθούν βέλτιστες λύσεις στο πρόβλημα χρονοπρογραμματισμού ενός έργου. Ωστόσο, ο χρόνος για την εκτέλεση των απαιτούμενων υπολογισμών αυξάνεται εκθετικά σε σχέση με τον αριθμό των διεργασιών, γεγονός που καθιστά τη συγκεκριμένη μέθοδο μη πρακτική και πολύ χρονοβόρα. Εξάλλου, όπως αναφέρουν οι Alcaraz και Maroto [12], η βέλτιστη λύση μπορεί να εξευρεθεί μόνο σε έργα μικρού μεγέθους, που δεν ξεπερνούν τις 60 διεργασίες και τα οποία δεν έχουν μεγάλο βαθμό περιορισμένων πόρων. Για το σκοπό αυτό έχουν αναπτυχθεί ευρετικές μέθοδοι ώστε να επιλύσουν το πρόβλημα χρονοπρογραμματισμού μεγάλων και απαιτητικών έργων.

2.2.2 Ευρετικές μέθοδοι

Οι ευρετικές τεχνικές πρόκειται για μεθόδους που μπορούν να φτάσουν στη βέλτιστη λύση ενός προβλήματος γρήγορα και εύκολα, αλλά παράλληλα δεν υπάρχει καμιά επίσημη απόδειξη για την ορθότητα των εν λόγω μεθόδων. Αν και δεν παρέχει ακριβείς οδηγίες για το πώς θα επιλυθεί το πρόβλημα, λειτουργεί σαν ένας γενικός οδηγός που κατευθύνει προς την εξεύρεση μιας λύσης που άλλοτε είναι αποδεκτή και άλλοτε όχι. Οι κυριότερες από αυτές είναι η κατατεταγμένη μέθοδος κλάδου φράγματος, η μέθοδος που βασίζεται στη διάζευξη ακμών και ο ευρετικός κανόνας προτεραιοτήτων [13, 14] .

Στην προσπάθεια που έχουν κάνει οι ερευνητές για επίλυση του RCPSP με χρήση των ευρετικών μεθόδων έχουν χρησιμοποιήσει κυρίως τον ευρετικό κανόνα προτεραιοτήτων αφού πρόκειται για έναν απλό και κατανοητό κανόνα όπου με τη χρήση ενός σχήματος δημιουργίας χρονοπρογράμματος (schedule generation scheme) σε κάθε στάδιο της μεθόδου, μπορεί εύκολα να εξευρεθεί μια εφικτή λύση. Το σχήμα σε κάθε φάση δημιουργεί το σύνολο απόφασης, δηλαδή το σύνολο των μη προγραμματισμένων διεργασιών που μπορούν να εκτελεστούν την επόμενη χρονική στιγμή, μέσα από το οποίο επιλέγεται η επόμενη διεργασία. Οι Alvarez-Valdes και Tamarit [15], οι Tormo και Lova

[16] και ο Laurence [17] έχουν χρησιμοποιήσει αυτήν την μέθοδο αξιοποιώντας την είτε σειριακά είτε παράλληλα, αλλά με καλά αποτελέσματα και στις δύο περιπτώσεις και δημιουργώντας πάντοτε ένα εφικτό χρονοπρόγραμμα.

Έρευνες έχουν γίνει αξιοποιώντας την κατατεταγμένη μέθοδο κλάδου φράγματος [15], η οποία διαφέρει από την κλασική μέθοδο κλάδου φράγματος στον τρόπο εξερεύνησης του δέντρου απαρίθμησης. Επίσης έχει χρησιμοποιηθεί η μέθοδος διάζευξης ακμών [15, 18] όπου βασική ιδέα είναι η επέκταση των σχέσεων προτεραιότητας (το σύνολο συζευκτικών ακμών) προσθέτοντας επιπλέον ακμές (διαζευκτικές ακμές) έτσι ώστε τα ελάχιστα απαγορευμένα σύνολα, δηλαδή τα σύνολα των δραστηριοτήτων που είναι εξαρτημένες και δεν μπορούν να προγραμματιστούν λόγω περιορισμών σε πόρους, να καταστραφούν και να παραμείνει το συντομότερο και πιο εφικτό χρονοπρόγραμμα.

2.2.3 Μεταευρετικές μέθοδοι

Πολλές μεταευρετικές μέθοδοι έχουν αναπτυχθεί στην προσπάθεια επίλυσης του δύσκολου συνδυαστικού προβλήματος του RCPSP και οι πιο διαδεδομένες τεχνικές περιλαμβάνουν την προσομοιωμένη ψύξη (simulated annealing) και την αναζήτηση Tabu (Tabu search).

Η προσομοιωμένη ψύξη έχει εισαχθεί στον ερευνητικό χώρο από τον Kirkpartick κ.α. [19] και είναι μια στοχαστική μέθοδος για βελτιστοποίηση συνδυαστικών προβλημάτων. Βασίζεται κυρίως στη φυσική διαδικασία ψύξης κατά την οποία ένα στερεό αντικείμενο που τήκεται αφήνεται να κρυώσει σε ένα χαμηλό ενεργειακό επίπεδο. Το συγκεκριμένο φαινόμενο μοντελοποιήθηκε στο RCPSP αφού ξεκινώντας από μια αρχική λύση δημιουργείται μια καινούρια λύση με μετάλλαξη της αρχικής, η επανομαζόμενη γειτονική, και αν η νέα λύση είναι καλύτερη από την αρχική, τότε γίνεται αποδεκτή και το σύστημα συνεχίζει από αυτό το σημείο έχοντας πλέον μια νέα λύση. Αντίθετα, αν η λύση είναι χειρότερη, τότε γίνεται αποδεκτή βάσει μιας πιθανότητας που εξαρτάται από το βαθμό επιδείνωσης και από μια παράμετρο που ονομάζεται θερμοκρασία. Καθώς ο αλγόριθμος προχωρά, η παράμετρος θερμοκρασίας μειώνεται έτσι ώστε να μειωθεί η πιθανότητα να γίνει αποδεκτή μια νέα χειρότερη λύση. Γενικά, η προσομοιωμένη ψύξη μπορεί να

θεωρηθεί ως μια επέκταση της στρατηγικής πρώτου ταιριάσματος (first fit strategy), αφού αποδέχεται αμέσως κάποια καλύτερη γειτονική λύση αλλά απορρίπτει τις χειρότερες. Οι Valls κ.α. [20] και οι Bouleimen και Lecocq [21] προσπάθησαν να επιλύσουν το RCPSP χρησιμοποιώντας τη συγκεκριμένη μέθοδο επιτυγχάνοντας καλά αποτελέσματα στα πειράματα που διεξήγαγαν.

Μια παρόμοια προσέγγιση που έχει μεγάλη απήχηση στην περιοχή της συνδυαστικής βελτιστοποίησης είναι και η αναζήτηση Tabu, η οποία έχει ως στόχο να αποτρέψει την αναζήτηση από το να κολλήσει σε τοπικά ελάχιστα. Το κλειδί σε αυτή την μέθοδο είναι η διατήρηση ενός ιστορικού των λύσεων που έχουν βρεθεί. Αναπτύχθηκε από τον Glover [22] και είναι ουσιαστικά μια μέθοδος απότομης καθόδου και ηπιότερης ανάβασης προς τη βέλτιστη λύση. Η προσέγγιση αρχικά αξιολογεί όλες τις λύσεις της γειτονιάς και αφού επιλέξει την καλύτερη συνεχίζει περαιτέρω επαναληπτικά με την αναζήτηση. Για να αποφευχθούν οι πιθανότητες δημιουργίας κύκλων, επιστροφής δηλαδή σε μια λύση που πρόσφατα έχει απορριφθεί, χρησιμοποιείται μια λίστα tabu που δουλεύει ως είδος μνήμης κατά τη διάρκεια της αναζήτησης. Συνήθως, η λίστα αξιοποιείται για να αποτρέψει οποιεσδήποτε κινήσεις μπορεί να γίνουν στη γειτονιά και να επαναφέρουν στην επιφάνεια μια χειρότερη λύση που είχε βρεθεί προηγουμένως. Γενικά, η αναζήτηση Tabu υλοποιεί τη στρατηγική καλύτερης λύσης αφού σαρώνει όλη την αρχική γειτονιά, βρίσκει την καλύτερη λύση και σταματά μόνο όταν δεν μπορεί να βρει κάποια άλλη λύση που να είναι καλύτερη από την ήδη υπάρχουσα. Οι Tomas και Salhi [23] παρουσίασαν μια προσέγγιση της μεθόδου αυτής για το RCPSP, με καλά αποτελέσματα, όπου καθόρισαν τρεις διαφορετικές στρατηγικές κινήσεων. Όταν μια γειτονιά παρουσιάζει μη εφικτά χρονοπρογράμματα, χρησιμοποιείται μια διαδικασία επισκευής έτσι ώστε να μετατραπούν τα χρονοπρογράμματα σε εφικτά. Επιπλέον, οι Nonobe και Ibaraki [24] προσπάθησαν να επιλύσουν το RCPSP χρησιμοποιώντας χρονικά μεταβαλλόμενους περιορισμούς στους πόρους, ενώ ο Klein [25] πρότεινε ένα μηχανισμό για γρήγορη μείωση του μεγέθους της γειτονιάς.

2.2.4 Εξελικτικές μέθοδοι

Οι εξελικτικές μέθοδοι αποτελούν το «νεότερο» είδος μεθόδων που χρησιμοποιούνται για επίλυση συνδυαστικών προβλημάτων βελτιστοποίησης και είναι εμπνευσμένες από τους βιολογικούς μηχανισμούς εξέλιξης. Συνδυάζοντας την επαναληπτική διαδικασία και τους εξελικτικούς μηχανισμούς, ο αρχικός πληθυσμός αλλάζει και μεταβάλλεται μέχρις ότου να οδηγηθεί στη βέλτιστη λύση και στο επιθυμητό αποτέλεσμα. Περιλαμβάνουν τη στοχαστική αναζήτηση διάχυσης (stochastic diffusion search), τον αλγόριθμο αναζήτησης βαρύτητας (gravitational search algorithm), τις ευφυείς σταγόνες νερού (intelligent water drops) και τους πιο διαδεδομένους γενετικούς αλγόριθμους (genetic algorithms) και αλγορίθμους βελτιστοποίησης αποικιών μυρμηγκιών (ant colony optimization) και βελτιστοποίησης σμήνους σωματιδίων (particle swarm optimization).

Οι γενετικοί αλγόριθμοι είναι μια προσέγγιση εμπνευσμένη από τη διαδικασία της βιολογικής εξέλιξης και προτάθηκε από τον Holland [26] το 1975. Σε αντίθεση με τις μεθόδους τοπικής αναζήτησης που έχουν αναφερθεί προηγουμένως, οι γενετικοί αλγόριθμοι διατηρούν ένα πληθυσμό πιθανών λύσεων αντί για μόνο μια. Έτσι αφού έχει δημιουργηθεί ο αρχικός πληθυσμός, δημιουργούνται νέες λύσεις είτε διασταυρώνοντας δύο υπάρχουσες είτε διαφοροποιώντας (μεταλλάσσοντας) κάποια άλλη. Μετά τη δημιουργία νέων λύσεων, οι καλύτερες λύσεις «επιβιώνουν» και σχηματίζουν την επόμενη γενεά ενώ οι υπόλοιπες «πεθαίνουν». Μετά από πολλές γενεές και με τη χρήση των βιολογικών λειτουργιών της διασταύρωσης και της μετάλλαξης αναμένεται να εξευρεθεί μια καλύτερη λύση για το πρόβλημα. Αποτελεί μια από τις πιο διαδεδομένες εξελικτικές μεθόδους, γι' αυτό και η πλειονότητα των ερευνητών προσπαθεί να επιλύσει το RCPSP αξιοποιώντας τις δυνατότητες τους [12, 27, 28, 29].

Αναλυτικά, οι Alcaraz και Maroto [12] ανέπτυξαν ένα γενετικό αλγόριθμο βασισμένο στη χρήση μιας λίστας διεργασιών και ένα σειριακό σχήμα δημιουργίας χρονοπρογράμματος. Ένα επιπλέον γονίδιο αποφασίζει αν θα γίνει χρονοπρογραμματισμός προς τα εμπρός ή προς τα πίσω ενώ η διασταύρωση επεκτείνεται με τρόπο ώστε η λίστα διεργασιών που

προκύπτει να δημιουργείται με ανάλογη κατεύθυνση. Επιπρόσθετα, ο Toklu [27] ανέπτυξε ένα γενετικό αλγόριθμο που αναπαριστά το χρονοπρόγραμμα, δηλαδή ένα διάγραμμα των χρόνων εκκίνησης των διεργασιών. Όμως επειδή οι γενετικοί αλγόριθμοι μπορούν να δημιουργήσουν ένα μη εφικτό χρονοπρόγραμμα, υπάρχει μια συνάρτηση τιμωρίας έτσι ώστε να αξιολογεί τις παραβάσεις των περιορισμών. Γενικά, έχουν καταγραφεί αξιοσημείωτα αποτελέσματα από τη χρήση των γενετικών αλγορίθμων όπως παρουσιάζονται από τους διάφορους ερευνητές της περιοχής.

Η βελτιστοποίηση αποικιών μυρμηγκιών προσομοιώνει τη συμπεριφορά των μυρμηγκιών καθώς κινούνται για αναζήτηση τροφής [30]. Τα μυρμήγκια αφήνουν καθ' όλη τη διαδρομή τους, από τη φύλια προς το σημείο προορισμού, μια ουσία που ονομάζεται φερομόνη, η οποία διεγείρει το ένστικτο της πείνας μέσω της οσμής. Η φερομόνη τοποθετείται στο μονοπάτι και χρησιμεύει στα υπόλοιπα μυρμήγκια για να αναγνωρίσουν το μονοπάτι και να το ακολουθήσουν. Επιπλέον, το ποσοστό της φερομόνης που βρίσκεται σε κάθε μονοπάτι αυξάνεται καθώς το διαπερνούν περισσότερα μυρμήγκια ενώ είναι αντιστρόφως ανάλογο του μήκους του. Συνεπώς, υπάρχει μεγαλύτερη ποσότητα φερομόνης στο συντομότερο μονοπάτι προς την τροφή. Γίνεται αντιληπτό ότι τα μυρμήγκια ακολουθούν το μονοπάτι που περιέχει το περισσότερο ποσοστό φερομόνης αφού αυτό λειτουργεί ως ένα σημάδι ότι το συγκεκριμένο μονοπάτι είναι το συντομότερο δυνατόν. Μοιάζει δηλαδή σαν μια κατάσταση εξαναγκασμού όπου όλος ο πληθυσμός προσπαθεί να βρει το συντομότερο μονοπάτι προς τον προορισμό. Έχει ευρεία απήχηση στον επιστημονικό κύκλο επιτυγχάνοντας καλά αποτελέσματα στην επίλυση του RCPSP [2, 31, 32].

Όμως, παρόλο που έχει κινήσει το ενδιαφέρον των επιστημόνων για εφαρμογή της σε συνδυαστικά προβλήματα βελτιστοποίησης, εντούτοις υποφέρει από πρόωρη σύγκλιση σε τοπικά ελάχιστα, δηλαδή σε λύσεις που δεν είναι οι καλύτερες δυνατές. Γι' αυτό το λόγο έχουν εξευρεθεί διάφοροι μηχανισμοί ούτως ώστε ο αλγόριθμος να μπορεί να ξεφύγει από τα τοπικά ελάχιστα/μέγιστα και να συνεχίσει την αναζήτησή του προς τη βέλτιστη λύση. Ο Merkle κ.α. [31] παρουσίασαν ένα καινούριο τρόπο αξιολόγησης της φερομόνης λαμβάνοντας υπόψη όχι μόνο την τοπική αξία αλλά και την συνολική αξία ενώ οι Stuzle

και Hoos [32] πρότειναν η συνάρτηση αξιολόγησης να λαμβάνει υπόψη το βαθμό σύγκλισης που υπάρχει στο σύστημα και την απόσταση από τη βέλτιστη λύση.

Παρόμοια μπορεί να θεωρηθεί η ιδέα που ώθησε τους Kennedy και Eberhart το 1995 [33] στην ανάπτυξη του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων. Ο συγκεκριμένος αλγόριθμος βασίζεται στον τρόπο με τον οποίο ένα σμήνος πουλιών κινείται στο χώρο αναζητώντας τροφή. Στο επόμενο κεφάλαιο ο αλγόριθμος της βελτιστοποίησης σμήνους σωματιδίων περιγράφεται με λεπτομέρεια και αναλύεται η εφαρμογή του στην επίλυση των RCPSP.

Κεφάλαιο 3

Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων

- 3.1 Εισαγωγή στους εξελικτικούς αλγορίθμους
 - 3.2 Η αρχική έκδοση
 - 3.2.1 Ο αλγόριθμος PSO
 - 3.2.2 Μαθηματική αναπαράσταση
 - 3.2.3 Γνωστικά και κοινωνικά μοντέλα του αλγορίθμου
 - 3.3 Επίδραση βάρους αδράνειας
 - 3.4 Περιοριστικός αλγόριθμος PSO
 - 3.5 Ενοποιημένος αλγόριθμος PSO
 - 3.6 Διακριτός αλγόριθμος PSO
 - 3.7 Τοπολογία σμήνους
 - 3.8 Ειδικοί αλγόριθμοι PSO
 - 3.8.1 Δυναμικά περιβάλλοντα
 - 3.8.2 Θορυβώδη περιβάλλοντα
 - 3.8.3 Περιβάλλοντα με πολλαπλούς στόχους
 - 3.8.4 Υβριδικοί αλγόριθμοι PSO
 - 3.9 Προβλήματα
 - 3.10 Σύνοψη
-

3.1 Εισαγωγή στους εξελικτικούς αλγορίθμους

Έχει αναφερθεί προηγουμένως ότι ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων ανήκει σε μια ευρύτερη κατηγορία αλγορίθμων, τους εξελικτικούς αλγορίθμους, που ανάμεσα σε άλλους συμπεριλαμβάνονται οι γενετικοί αλγόριθμοι, ο αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών και οι λιγότερο διαδεδομένοι αλγορίθμοι αναζήτησης αρμονίας, στοχαστικής αναζήτησης διάχυσης και του τεχνητού ανοσοποιητικού συστήματος. Αποτελούν ένα υποσύνολο των αλγορίθμων που κατατάσσονται στον τομέα της τεχνητής ζωής και έχουν εμπνευστεί από τη βιολογική εξέλιξη της ζωής, τη χλωρίδα και την πανίδα και τη φύση γενικότερα. Πρόκειται κυρίως για αλγορίθμους που ξεκινούν με ένα αρχικό πληθυσμό από υποψήφιες λύσεις και κατά τη διάρκεια των επαναλήψεων και μέσω των τροποποιητικών βημάτων κάθε αλγορίθμου, ο πληθυσμός μεταβάλλεται και οδηγείται στη βέλτιστη λύση.

Παρόλο που υπάρχουν στη γενικότερη κατηγορία των εξελικτικών αλγορίθμων, εντούτοις διαχωρίζονται σε δυο επιμέρους κατηγορίες: (1) αλγόριθμοι οι οποίοι παρουσιάζουν ανταγωνισμό μέσα στον πληθυσμό για το ποιο άτομο θα φτάσει πρώτο στη βέλτιστη λύση και (2) αλγόριθμοι οι οποίοι προωθούν τη συνεργασία μεταξύ των ατόμων του πληθυσμού ώστε να οδηγηθεί όλος ο πληθυσμός στη βέλτιστη λύση. Στην πρώτη κατηγορία κατατάσσονται οι γενετικοί αλγόριθμοι ενώ στη δεύτερη ο αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών και ο αντίστοιχος σμήνους σωματιδίων (Particle Swarm Optimization, PSO). Τα πολλά πλεονεκτήματα που έχουν, όπως η γενική εφαρμογή τους σε μεγάλος πλήθος προβλημάτων, η μη αναγκαία προηγούμενη γνώση του χώρου του προβλήματος, καθώς επίσης και η ευκολία ανάπτυξης των αλγορίθμων, τους έχει κάνει πολύ δημοφιλείς σε σημείο ώστε να καταλαμβάνουν μεγάλο μερίδιο στην επιστημονική έρευνα επίλυσης συνδυαστικών προβλημάτων βελτιστοποίησης. Όμως, όπως όλοι οι εξελικτικοί αλγόριθμοι, ακολουθούν πιστά το θεώρημα του «no free lunch» [34] όπου στις πλείστες των περιπτώσεων, η γενική εφαρμογή τους σε προβλήματα βελτιστοποίησης κοστίζει σε επίδοση καθώς δεν υπάρχει μια βέλτιστη γενική προσέγγιση για προβλήματα όλων των ειδών. Ακόμα ένα κοινό πρόβλημα που υπάρχει, είναι ότι δεν παρέχεται καμιά εγγύηση πως κάποιος αλγόριθμος θα καταλήξει στη βέλτιστη λύση και δεν θα κολλήσει σε

κάποιο τοπικό βέλτιστο δυσκολεύοντας έτσι την αξιοποίησή τους σε δύσκολα προβλήματα χωρίς την ενσωμάτωση έξυπνων μεθόδων για να ξεφεύγει από τα τοπικά βέλτιστα.

Η διπλωματική εργασία αυτή προσπαθεί να επιλύσει το πρόβλημα χρονοπρογραμματισμού ενός έργου λογισμικού περιορισμένων πόρων αξιοποιώντας τις δυνατότητες του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων. Στη συνέχεια επεξηγείται ο αλγόριθμος και παρουσιάζονται ορισμένες παραλλαγές που έχουν προταθεί κατά καιρούς και έχουν εφαρμοστεί σε δύσκολα προβλήματα βελτιστοποίησης.

3.2 Η αρχική έκδοση

Οι αρχικές ιδέες των Kennedy (ψυχολόγος) και Eberhart (ηλεκτρολόγος μηχανικός) στα σμήνη σωματιδίων επηρεάστηκαν από το μοντέλο των Heppner και Grenander [35] όπου προσπάθησαν να υλοποιήσουν ένα αλγόριθμο βασισμένο στην κοινωνική συμπεριφορά. Συγκεκριμένα, χρησιμοποίησαν ένα κοπάδι πουλιών και ένα κόκορα ως σημείο σύγκλισης για να μοντελοποιήσουν την προσπάθεια του κοπαδιού καθώς αναζητά φαγητό. Έχοντας αυτά υπόψη και σκεφτόμενοι τον τρόπο με τον οποίο κινείται ένα σμήνος πουλιών και μια αγέλη ψαριών οι Kennedy και Eberhart [33] ανέπτυξαν το 1995 μια ισχυρή μέθοδο βελτιστοποίησης, τον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων.

Στον αλγόριθμο PSO ένας αριθμός από οντότητες (τα σωματίδια) τοποθετούνται στο χώρο αναζήτησης του προβλήματος και κάθε ένα από αυτά αξιολογεί τη θέση του βάσει της αντικειμενικής συνάρτησης αξιολόγησης/καταλληλότητας. Στη συνέχεια, κάθε σωματίδιο αποφασίζει πώς θα κινηθεί λαμβάνοντας υπόψη τη δική του τρέχουσα θέση, τη δική του προηγούμενη καλύτερη θέση και τις θέσεις κάποιων άλλων σωματιδίων του σμήνους. Η επόμενη επανάληψη γίνεται όταν όλα τα σωματίδια του σμήνους έχουν μετακινηθεί. Ως αποτέλεσμα, το σμήνος μετακινείται όλο μαζί, σαν μια αγέλη πουλιών που αναζητεί φαγητό, και ολοένα πλησιάζει πιο κοντά στο στόχο.

3.2.1 Ο αλγόριθμος PSO

Ο αλγόριθμος PSO αποτελείται από ένα σμήνος σωματιδίων που βρίσκονται διασπαρμένα στο χώρο. Κάθε σωματίδιο χαρακτηρίζεται από τρία διανύσματα: την τρέχουσα θέση, την προηγούμενη καλύτερη θέση και την ταχύτητα του σωματιδίου. Πρέπει να σημειωθεί ότι η θέση κάθε σωματιδίου αναπαριστά μια λύση στο πρόβλημα το οποίο εξετάζεται. Ο αλγόριθμος αρχικοποιείται με ένα πληθυσμό από σωματίδια τυχαία διασκορπισμένα στο χώρο αναζήτησης και έπειτα ξεκινά η αναζήτηση προς την καλύτερη θέση. Σε κάθε επανάληψη αξιολογείται η θέση κάθε σωματιδίου βάσει του προβλήματος και υπολογίζεται η καλύτερη τιμή για κάθε σωματίδιο καθώς επίσης και η καλύτερη τιμή ολόκληρου του σμήνους. Στη συνέχεια, τα σωματίδια κινούνται προς μια καινούρια θέση βάσει της ταχύτητας που έχουν. Συνεπώς, η ταχύτητα στην μετακίνηση των σωματιδίων σε νέες θέσεις παίζει σημαντικό ρόλο στη δημιουργία νέων λύσεων.

Ο συγκεκριμένος αλγόριθμος διατηρεί δύο ειδών ιστορικό, την ολική καλύτερη θέση και την ατομική καλύτερη θέση. Η πρώτη αντιπροσωπεύει την καλύτερη λύση που έχει βρεθεί μέχρι τώρα από όλα τα σωματίδια του σμήνους ενώ η δεύτερη αναπαριστά την καλύτερη λύση που έχει βρεθεί από κάθε σωματίδιο του σμήνους ξεχωριστά. Χρησιμοποιώντας αυτά τα δύο και σε συνδυασμό με την προηγούμενη ταχύτητα του σωματιδίου υπολογίζεται η νέα του ταχύτητα, η οποία με τη σειρά της χρησιμοποιείται για να βρεθεί η νέα θέση που θα κινηθεί το σωματίδιο. Σημαντική παρατήρηση είναι ότι το σμήνος σωματιδίων δεν αποτελεί μονάχα μια συλλογή από σωματίδια αλλά κάτι πολύ πιο δυνατό. Επιπλέον, κανένα σωματίδιο από μόνο του δεν έχει τη δύναμη να λύσει κανένα πρόβλημα, αλλά αυτό επιτυγχάνεται μονάχα μέσω της συνεργασίας και της αλληλεπίδρασης που υπάρχει μεταξύ τους.

3.2.2 Μαθηματική αναπαράσταση

Έστω ένας χώρος αναζήτησης με N διαστάσεις¹ που περιέχει M σωματίδια. Η θέση του i -οστού σωματιδίου του πληθυσμού ($i = 1, 2, \dots, M$) αναπαριστάται από το διάνυσμα $X_i = (X_{i1}, X_{i2}, \dots, X_{iN})$ και το σημείο X_{ij} είναι το σημείο της j διάστασης του i -οστού σωματιδίου. Αντίστοιχα, η ταχύτητα του i -οστού σωματιδίου αναπαριστάται από διάνυσμα $V_i = (V_{i1}, V_{i2}, \dots, V_{iN})$ όπου V_{ij} είναι η ταχύτητα της j διάστασης του i -οστού σωματιδίου. Επιπλέον, η καλύτερη λύση που έχει βρεθεί από το i -οστό σωματίδιο υποδεικνύεται ως $P_i = (P_{i1}, P_{i2}, \dots, P_{iN})$ ενώ η καλύτερη λύση που έχει βρεθεί από όλα τα σωματίδια του σμήνους μέχρι τώρα υποδεικνύεται ως $G = (G_1, G_2, \dots, G_N)$. Οι παραπάνω παραμέτροι χρησιμοποιούνται για να ανανεώσουν την ταχύτητα και την θέση κάθε σημείου των σωματιδίων του σμήνους.

Παρακάτω παρατίθεται ο αρχικός αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων όπως έχει προταθεί από τους εισηγητές του [33]:

Αρχικοποιείται ο πληθυσμός των M σωματιδίων σε τυχαίες θέσεις και ταχύτητες μέσα σε ένα χώρο αναζήτησης N διαστάσεων.

Για κάθε σωματίδιο

1. Αξιολογείται η θέση του βάσει της αντικειμενικής συνάρτησης αξιολόγησης/καταλληλότητας.
2. Συγκρίνεται η τρέχουσα αξιολόγησή του με την προσωπική του καλύτερη αξιολόγηση.
3. Αν η τρέχουσα αξιολόγηση είναι καλύτερη από την καλύτερη προσωπική τότε ενημερώνεται η τιμή της καθώς επίσης και το διάνυσμα P_i .
4. Εντοπίζεται το σωματίδιο στο σμήνος που έχει την καλύτερη αξιολόγηση μέχρι στιγμής και καθορίζεται ως το ολικό καλύτερο σωματίδιο G .

¹ Ο αριθμός των διαστάσεων εξαρτάται από το είδος του προβλήματος.

5. Υπολογίζεται η νέα ταχύτητα και θέση του με βάση τις πιο κάτω εξισώσεις:

$$V_{ij}^{t+1} = V_{ij}^t + c_1 \times r_1 \times (P_{ij}^t - X_{ij}^t) + c_2 \times r_2 \times (G_j^t - X_{ij}^t) \quad (4)$$

και

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \quad (5)$$

όπου r_1, r_2 : τυχαίοι αριθμοί στο $U(0,1)$

c_1, c_2 : γνωστικός και κοινωνικός συντελεστής επιτάχυνσης αντίστοιχα

X_{ij}^t : η θέση του i -οστού σωματιδίου στη j -οστή διάσταση στο χρόνο t

V_{ij}^t : η ταχύτητα του i -οστού σωματιδίου στη j -οστή διάσταση στο χρόνο t

P_{ij}^t : η καλύτερη θέση που έχει βρεθεί το i -οστό σωματίδιο στη j -οστή διάσταση στο χρόνο t

G_j^t : η καλύτερη θέση που έχει βρεθεί το σμήνος στη j -οστή διάσταση στο χρόνο t

6. Αν τα κριτήρια τερματισμού ικανοποιούνται τότε τερματίζεται η εκτέλεση του αλγορίθμου αλλιώς συνεχίζεται η εκτέλεση από το βήμα 2.

Σχήμα 3.1 – Βασικός αλγόριθμος PSO

Ο βασικός αλγόριθμος PSO που περιγράφεται πιο πάνω έχει μόνο ένα μικρό αριθμό παραμέτρων που πρέπει να μεταβληθούν. Μία από αυτές είναι το μέγεθος του πληθυσμού όπου εμπειρικά τίθεται βάσει των διαστάσεων του προβλήματος και της δυσκολίας του. Οι τιμές μεταξύ είκοσι και πενήντα είναι αυτές που προτιμούνται πιο συχνά. Επιπλέον, οι συντελεστές r_1 και r_2 είναι τυχαίοι αριθμοί που ακολουθούν την κανονική κατανομή στο πεδίο $[0,1]$ και είναι υπεύθυνοι για να προσθέτουν ένα βαθμό τυχαιότητας στον τρόπο πτήσης του σμήνους ή αλλιώς μετακίνησης του σμήνους. Οι παράμετροι c_1 και c_2 ονομάζονται συντελεστές επιτάχυνσης και επιτρέπουν σε ένα σωματίδιο να ρυθμίσει την επιρροή που έχει η προσωπική καλύτερη θέση και η καλύτερη θέση του σμήνους

αντίστοιχα, στην εξίσωση μεταβολής της ταχύτητας. Ο συντελεστής c_1 καλείται γνωστικός συντελεστής και λαμβάνει υπόψη την προσωπική εμπειρία ενός σωματιδίου ενώ ο συντελεστής c_2 καλείται κοινωνικός συντελεστής και εκφράζει την επιρροή που έχει ένα σωματίδιο από το καλύτερο σωματίδιο του σμήνους. Μεγάλες τιμές για το c_1 προωθούν την εξερεύνηση του χώρου αναζήτησης ενώ μεγάλες τιμές για το c_2 ευνοούν την εκμετάλλευση των θέσεων που κατέχουν τα σωματίδια. Αρχικά, είχε υιοθετηθεί οι τιμές τους να ισούνται με δύο ούτως ώστε να υπάρχει ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης στο σμήνος. Όμως μέσα από πειραματικές διαδικασίες αποδείχτηκε ότι πολλές φορές αυτό είναι ζημιογόνο στη διαδικασία αναζήτησης επειδή τα σωματίδια μπορούσαν να ξεφύγουν από τον ενδεδειγμένο χώρο αναζήτησης. Συνεπώς για να περιοριστεί η πιθανότητα τα σωματίδια να ξεφύγουν ο Eberhart κ.α. [36] το 1996 εισηγήθηκαν την εφαρμογή κάποιου είδους περιορισμού στην ταχύτητα των σωματιδίων στο πεδίο $[-v_{max}, v_{max}]$ όπου v_{max} είναι μια τιμή μεταξύ του 0.1 και της μέγιστης θέσης που μπορεί να βρεθεί ένα σωματίδιο.

3.2.3 Γνωστικά και κοινωνικά μοντέλα του αλγορίθμου

Το 1997 ο Kennedy [52] έκανε μια από τις πρώτες και πιο χρήσιμες αναλύσεις για τον αλγόριθμο PSO. Σκοπός του ήταν να αποδείξει το όφελος της κοινωνικής επίδρασης του σμήνους στην προσπάθεια για να επιτύχει το σμήνος το στόχο του. Έτσι σύγκρινε το βασικό αλγόριθμο με τρία νέα μοντέλα με τρεις αντίστοιχες συναρτήσεις μεταβολής της ταχύτητας σωματιδίων (Σχήμα 3.2.): το γνωστικό μοντέλο όπου δεν υπάρχει καμιά επίδραση από άλλα σωματίδια του σμήνους, το κοινωνικό μοντέλο όπου χρησιμοποιείται το καλύτερο σωματίδιο του σμήνους και το ανιδιοτελές μοντέλο όπου το αντίστοιχο σωματίδιο αποκλείεται από το συναγωνισμό για το καλύτερο σωματίδιο του σμήνους. Στην πρώτη περίπτωση τα σωματίδια συμπεριφέρονταν σαν ανεξάρτητοι «ορειβάτες», στη δεύτερη το σμήνος συμπεριφερόταν σαν ένα ενιαίο «ορειβάτη» ενώ η τελευταία περίπτωση είχε χειρότερα αποτελέσματα από τις προηγούμενες δύο. Ο συγγραφέας συμπέρανε ότι είναι απαραίτητες και οι δύο επιδράσεις αφού αν παραληφθεί κάποια τότε είναι πολύ πιθανόν το σμήνος να συγκλίνει σε μια τοπικά βέλτιστη λύση.

Μοντέλο	Συνάρτηση μεταβολής ταχύτητας
Γνωστικό	$V_{ij}^{t+1} = V_{ij}^t + c_1 \times r_1 \times (P_{ij}^t - X_{ij}^t)$
Κοινωνικό	$V_{ij}^{t+1} = V_{ij}^t + c_2 \times r_2 \times (G_j^t - X_{ij}^t)$
Ανιδιοτελές	$V_{ij}^{t+1} = V_{ij}^t + c_2 \times r_2 \times (G_j^t - X_{ij}^t)^2$

Σχήμα 3.2. – Μοντέλα για την απόδειξη του οφέλους της κοινωνικής επίδρασης από τον Kennedy

3.3 Επίδραση βάρους αδράνειας

Οι Shi και Eberhart [37] πραγματοποίησαν ορισμένα πειράματα πάνω στο βασικό αλγόριθμο με σκοπό να παρατηρήσουν την αντίδραση του σε ειδικές περιπτώσεις. Αρχικά, αφαίρεσαν το πρώτο μέρος της εξίσωσης μεταβολής της ταχύτητας, τη μνήμη δηλαδή της προηγούμενης ταχύτητας (V_{ij}^t), και παρατήρησαν ότι το σμήνος συνέκλινε στην καλύτερη λύση που είχε βρεθεί στα αρχικά στάδια του αλγορίθμου προωθώντας περισσότερο την εκμετάλλευση. Αντίθετα, χρησιμοποιώντας τη μνήμη της προηγούμενης ταχύτητας, το σμήνος συμπεριφερόταν με αντίθετη συμπεριφορά και εξαπλωνόταν προωθώντας την αναζήτηση. Για το πρόβλημα αυτό έπρεπε να βρεθεί κάποιος τρόπος ώστε να μπορεί να διατηρηθεί ένα ισοζύγιο μεταξύ εκμετάλλευσης και εξερεύνησης, γεγονός που τους οδήγησε στην προώθηση ενός τροποποιημένου αλγορίθμου που ενσωματώνει το βάρος της αδράνειας (inertia weight):

$$V_{ij}^{t+1} = w \times V_{ij}^t + c_1 \times r_1 \times (P_{ij}^t - X_{ij}^t) + c_2 \times r_2 \times (G_j^t - X_{ij}^t) \quad (6)$$

όπου w είναι το βάρος αδράνειας στο πεδίο $[0.4, 1.2]$ και χρησιμοποιείται για να καθορίσει την επίδραση που έχει η προηγούμενη τιμή της ταχύτητας στη νέα ταχύτητα.

² Πρόκειται για το κοινωνικό μοντέλο με τη διαφορά ότι το κάθε σωματίδιο που εξετάζεται, αποκλείεται από το συναγωνισμό για το G_j^t και αυτό επιλέγεται από τα υπόλοιπα σωματίδια του σμήνους

Όπως αναφέρουν οι Shi και Eberhart στο άρθρο τους [38] «το βάρος αδράνειας χρησιμοποιείται για να καθορίσει την επίδραση που έχει η προηγούμενη τιμή της ταχύτητας στη νέα, συνεπώς για να επηρεαστούν οι ικανότητες των σωματιδίων για εκμετάλλευση και εξερεύνηση. Μεγαλύτερες τιμές για την αδράνεια διευκολύνουν την εξερεύνηση, δηλαδή την αναζήτηση σε νέους χώρους, ενώ μικρότερες τιμές διευκολύνουν την εκμετάλλευση του χώρου αναζήτησης που ήδη εξερευνείται. Συνεπώς η κατάλληλη επιλογή του βάρους της αδράνειας (w) πρέπει να γίνει ώστε να προωθείται τόσο η εξερεύνηση όσο και η εκμετάλλευση με αποτέλεσμα να απαιτούνται λιγότερες επαναλήψεις για να βρεθεί το βέλτιστο σημείο». Οι συγγραφείς πρότειναν επίσης ότι μια τιμή μεταξύ του 0.8 και του 1.2 έχει καλά αποτελέσματα. Μετέπειτα στο [39] απόδειξαν ότι μπορούν να παραχθούν καλύτερα αποτελέσματα αν η τιμή του τεθεί αρχικά 0.9 και μειώνεται γραμμικά προς το 0.4. Με αυτό τον τρόπο ευνοείται η εξερεύνηση στα πρώτα στάδια του αλγορίθμου και η εκμετάλλευση προς το βέλτιστο σημείο στις τελευταίες επαναλήψεις.

Δια μέσου των χρόνων πολλοί ερευνητές έχουν προσπαθήσει να βρουν την κατάλληλη τιμή για το βάρος αδράνειας και έχουν πραγματοποιήσει αρκετά πειράματα στην προσπάθεια τους αυτή. Οι Eberhart και Shi [40] αναφέρουν ότι καλά αποτελέσματα μπορούν να επιτευχθούν χρησιμοποιώντας σαν τιμή του βάρους αδράνειας μια τυχαία τιμή μεταξύ 0.5 και 1 όπου $w = U(0.5, 1)$. Επίσης, υπάρχουν έρευνες όπως τον Zheng κ.α. [41] όπου αυξάνοντας το βάρος της αδράνειας κατάφεραν να επιτύχουν καλά αποτελέσματα σε σειρά πειραμάτων που διεξήγαγαν. Ο Trelea [42] μέσω πειραματικών διαδικασιών κατάφερε να αποδείξει ότι το βάρος αδράνειας είναι χρήσιμο στον αλγόριθμο και επομένως πρέπει να χρησιμοποιείται.

3.4 Περιοριστικός αλγόριθμος PSO

Παρόλο που οι ερευνητές που ασχολήθηκαν με τον αλγόριθμο στα πρώτα του στάδια κατανοούσαν ότι ήταν αναγκαίος ο περιορισμός της δυναμικής των σωματιδίων (v_{max}), εντούτοις κανένας δεν μπορούσε να εξηγήσει το λόγο. Αλλά ήταν αντιληπτό από όλους πως όταν ο αλγόριθμος αφηνόταν να εκτελεστεί χωρίς να γίνεται κάποιου είδους περιορισμό

στην ταχύτητα των σωματιδίων, τότε η ταχύτητα τους αυξανόταν ραγδαία και υπερέβαινε τα επιτρεπτά όρια μέσα σε ορισμένες μόνο επαναλήψεις.

Το πρόβλημα αυτό οδήγησε τους Clerc και Kennedy [43] να προτείνουν μια εναλλακτική μέθοδο για να ελέγχεται η συμπεριφορά των σωματιδίων στο σμήνος. Έτσι αντί για το βάρος αδράνειας που ενεργούσε στη μνήμη της ταχύτητας ενός σωματιδίου πρότειναν την εφαρμογή ενός περιοριστικού παράγοντα x στη νέα ταχύτητα που είχε ως αποτέλεσμα τον καλύτερο έλεγχο του σμήνους αποτρέποντας, από τη μια, τη διαφυγή σε περιοχές μη επιθυμητές και, από την άλλη, εγγυόταν τη σύγκλισή του. Επιπλέον με αυτή τη μέθοδο η χρήση της παραμέτρου v_{max} , δεν ήταν πλέον αναγκαία και η συνάρτηση μεταβολής της ταχύτητας των σωματιδίων μετατράπηκε σε:

$$V_{ij}^{t+1} = x \times \left(V_{ij}^t + c_1 \times r_1 \times (P_{ij}^t - X_{ij}^t) + c_2 \times r_2 \times (G_j^t - X_{ij}^t) \right) \quad (7)$$

όπου

$$x = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (8)$$

με $\varphi = \varphi_1 + \varphi_2 > 4$, $\varphi_1 = c_1 \times r_1$, $\varphi_2 = c_2 \times r_2$ και $k \in [0, 1]$.

Όταν χρησιμοποιείται η μέθοδος αυτή τότε συνήθως η τιμή του συντελεστή φ ισούται με 4.1, το x ισούται με 0.7298 και οι συντελεστές c_1 , c_2 ισούνται με 2.05. Αυτό έχει ως αποτέλεσμα όλοι οι επιμέρους όροι της εξίσωσης της ταχύτητας να πολλαπλασιάζονται με 0.7298 και να περιορίζονται αναλόγως. Πρέπει να σημειωθεί ότι ο περιοριστικός PSO είναι αλγεβρικά ισοδύναμος με τον PSO με βάρος αδρανείας αν τεθεί $w = 0.7298$ και $c_1 = c_2 = 1.496$.

Έχει αποδειχτεί μέσα από έρευνες ότι αν $\varphi \geq 4$ και $k \in [0, 1]$ τότε το σμήνος συγκλίνει σε ένα σταθερό σημείο ενώ η τιμή που προτείνεται για το x είναι μεταξύ μηδέν και ένα, δηλαδή, $x \in [0, 1]$. Αξιοσημείωτη είναι επίσης η επίδραση του συντελεστή k στην νέα

μέθοδο αφού όταν η τιμή του πλησιάζει στο μηδέν τότε παρατηρείται εκμετάλλευση και γρήγορη σύγκλιση του σμήνους ενώ όταν η τιμή του είναι κοντά στο ένα παρατηρείται εξερεύνηση του χώρου έρευνας και αργή σύγκλιση του σμήνους.

3.5 Ενοποιημένος αλγόριθμος PSO

Η επίδοση του αλγορίθμου εξαρτάται σε μεγάλο βαθμό από την αξιοποίηση των ικανοτήτων εξερεύνησης και εκμετάλλευσης και συγκεκριμένα την ικανότητα για εξερεύνηση νέων περιοχών στο χώρο αναζήτησης και την ικανότητα να συγκλίνει γρήγορα σε περιοχές με υποσχόμενες λύσεις αντίστοιχα. Από τη μια η σφαιρική εκδοχή του αλγορίθμου προωθεί την εκμετάλλευση αφού όλα τα σωματίδια έλκονται από το καλύτερο σωματίδιο του σμήνους και επομένως συγκλίνουν γρήγορα στο ίδιο σημείο, ενώ από την άλλη η τοπική εκδοχή παρουσιάζει καλύτερα αποτελέσματα στην εξερεύνηση, αφού οι πληροφορίες σχετικά με το καλύτερο σωματίδιο κάθε γειτονιάς μεταφέρονται στο υπόλοιπο σμήνος διαμέσου των γειτονικών σωματιδίων. Γι' αυτό η επιρροή προς συγκεκριμένα σημεία του χώρου αναζήτησης είναι πιο αδύναμη και επομένως αποτρέπει τον εγκλωβισμό του σμήνους σε τοπικά βέλτιστα.

Λαμβάνοντας αυτά υπ' όψη οι Parsopoulos και Vrahatis [44] τροποποίησαν τον περιοριστικό PSO για να αξιοποιήσουν τόσο την ικανότητα εκμετάλλευσης της σφαιρικής εκδοχής όσο και την ικανότητα εξερεύνησης της τοπικής εκδοχής. Ενοποιώντας λοιπόν τις δύο αυτές εκδοχές πρότειναν η μεταβολή της ταχύτητας να εξαρτάται και από τις δύο, όπως υποδεικνύουν οι παρακάτω εξισώσεις:

$$S_{ij}^{t+1} = x \times \left(V_{ij}^t + c_1 \times r_1 \times (P_{ij}^t - X_{ij}^t) + c_2 \times r_2 \times (G_j^t - X_{ij}^t) \right) \quad (9)$$

και

$$L_{ij}^{t+1} = x \times \left(V_{ij}^t + c_1 \times r_1' \times (P_{ij}^t - X_{ij}^t) + c_2 \times r_2' \times (G_j^t - X_{ij}^t) \right) \quad (10)$$

όπου S_{ij}^{t+1} και L_{ij}^{t+1} είναι η ολική και η τοπική μεταβολή της ταχύτητας αντίστοιχα, G_j^t είναι η θέση του καλύτερου σωματιδίου του σμήνους και G_{ij}^t είναι η θέση του καλύτερου

σωματιδίου στη γειτονιά. Στη συνέχεια τα δύο αυτά αποτελέσματα συνδυάζονται για να μεταβάλουν ανάλογα την ταχύτητα του σωματιδίου:

$$V_{ij}^{t+1} = (1 - u) \times L_{ij}^{t+1} + u \times S_{ij}^{t+1} \quad (11)$$

όπου $u \in [0, 1]$ και είναι ένας παράγοντας ενοποίησης ο οποίος διατηρεί το ισοζύγιο μεταξύ σφαιρικών και τοπικών πτυχών της αναζήτησης. Αν $u = 1$ τότε ο αλγόριθμος μετατρέπεται αμέσως στο μοντέλο όπου όλο το σμήνος αντιμετωπίζεται σαν μια γειτονιά, ενώ αν $u = 0$ τότε χρησιμοποιείται η τοπική παραλλαγή του PSO (η τοπολογία που μπορεί να διαθέτει ένα σμήνος επεξηγείται στο υποκεφάλαιο 3.7). Πρέπει να σημειωθεί ότι εξίσωση μεταβολής της θέσης ενός σωματιδίου παραμένει ως είχε προηγουμένως.

Πειραματικές διαδικασίες που διεξήχθησαν από τους Parsopoulos και Vrahatis [45, 46] παρουσίασαν καλά αποτελέσματα και άφησαν υποσχέσεις για περαιτέρω αξιοποίηση της μεθόδου σε πιο δύσκολα προβλήματα.

3.6 Διακριτός αλγόριθμος PSO

Παρόλο που στα αρχικά στάδια της ανάπτυξης του αλγόριθμου PSO η έρευνα είχε επικεντρωθεί στη βελτιστοποίηση προβλημάτων στο συνεχή χώρο αναζήτησης, είχε γίνει προσπάθεια τροποποίησης του αλγορίθμου ώστε να μπορεί να εφαρμοστεί σε διακριτά προβλήματα. Οι Kennedy και Eberhart [47] ανέπτυξαν το 1997 την πρώτη διακριτή εκδοχή του αλγόριθμου PSO για δυαδικά προβλήματα. Επίλυσαν το πρόβλημα μετακίνησης του κάθε σωματιδίου στο χώρο αναζήτησης δίνοντας μια διαφορετική έννοια στον τρόπο που αντιμετωπίζεται η ταχύτητα του σωματιδίου. Ενώ στο συνεχή χώρο η ταχύτητα αντιπροσωπεύει τη μεταβολή της θέσης του σωματιδίου, στο διακριτό χώρο αναπαριστά το κατώφλι πιθανότητας κάθε bit να μετατραπεί σε ένα. Για παράδειγμα, σε μια διάσταση d , αν η v_{id} αναπαριστά την πιθανότητα η τιμή του x_{id} να είναι ένα και $v_{id} = 0.4$ αυτό σημαίνει ότι υπάρχει 40% πιθανότητα η τιμή του bit x_{id} να είναι ένα. Επιπλέον, η τιμή της

ταχύτητας περιορίζεται στο διάστημα $[0, 1]$ χρησιμοποιώντας την ακόλουθη σιγμοειδή συνάρτηση:

$$S(v_{id}^t) = \frac{1}{1 + e^{-v_{id}^t}} \quad (12)$$

όπου $S(v_{id}^t)$ αντιπροσωπεύει την πιθανότητα το bit x_{id}^t να πάρει την τιμή ένα. Στη συνέχεια παράγεται ένας τυχαίος αριθμός με βάση την κανονική κατανομή στο διάστημα $[0, 1]$ και συγκρίνεται με το αποτέλεσμα της σιγμοειδούς συνάρτησης $S(v_{id}^t)$. Αν ο τυχαίος αριθμός που έχει παραχθεί είναι μικρότερος από το $S(v_{id}^t)$ τότε το bit x_{id}^t γίνεται ένα, αλλιώς γίνεται μηδέν.

Για να αποφευχθεί η περίπτωση όπου το $S(v_{id}^t)$ πλησιάζει τις τιμές μηδέν ή ένα, χρησιμοποιείται μια σταθερά v_{max} για να περιορίσει το v_{id} . Συνήθως η τιμή v_{max} καθορίζεται ως 4 και η ταχύτητα κάθε σημείου του σωματιδίου περιορίζεται στο διάστημα $[-4, 4]$. Αξίζει να σημειωθεί ότι ο υπολογισμός της ταχύτητας δεν αλλάζει και παραμένει όπως προηγουμένως. Παρακάτω δίνεται ο ψευδοκώδικας μετατροπής της θέσης ενός σημείου στη διακριτή μορφή του PSO:

.....

.....

$$V_{ij}^{t+1} = w \times V_{ij}^t + c_1 \times r_1 \times (P_{ij}^t - X_{ij}^t) + c_2 \times r_2 \times (G_j^t - X_{ij}^t)$$

$$V_{ij}^{t+1} \in [-v_{max}, v_{max}]$$

Αν τυχαίος αριθμός $< S(v_{id}^t)$

$$x_{id}^t = 1$$

αλλιώς

$$x_{id}^t = 0$$

.....

.....

Σχήμα 3.3. – Διακριτός αλγόριθμος PSO

Οι Kennedy και Spears [48] σύγκριναν πειραματικά τον παραπάνω αλγόριθμο με τρεις παραλλαγές ενός γενετικού αλγόριθμου, (με μετάλλαξη αλλά χωρίς διασταύρωση, με διασταύρωση αλλά χωρίς μετάλλαξη, και με μετάλλαξη και διασταύρωση). Τα αποτελέσματα έδειξαν ότι ο διακριτός αλγόριθμος PSO ήταν ο μόνος που κατάφερε να βρει το καθολικό βέλτιστο σε κάθε πείραμα που πραγματοποιήθηκε. Επιπλέον, υπερείχε των υπολοίπων αλγορίθμων στο θέμα ταχύτητας σχεδόν σε όλα τα πειράματα.

Πολλοί ερευνητές έχουν προτείνει διάφορες παραλλαγές αυτού του αλγόριθμου ώστε να μπορεί να αξιοποιηθεί σε δυαδικούς χώρους. Οι Agrafiotis και Cedeno [49] χρησιμοποίησαν τις τοποθεσίες των σωματιδίων ως πιθανότητες για να επιλέξουν χαρακτηριστικά βασισμένα σε κάποιο πρότυπο. Κάθε χαρακτηριστικό συνδεόταν με ένα κομμάτι ενός τροχού ρουλέτας που βασιζόταν στις τιμές του δεκαδικού μέρους του σημείου, το οποίο μετά μετατρεπόταν σε μηδέν ή ένα, δείχνοντας αν το συγκεκριμένο χαρακτηριστικό επιλέχθηκε ή όχι. Επίσης οι Mohan και Al-Kazemi [50] εισηγήθηκαν αρκετούς άλλους τρόπους με τους οποίους μπορεί να χρησιμοποιηθεί ο PSO στο δυαδικό χώρο.

Μετέπειτα άλλοι ερευνητές προσπάθησαν να τροποποιήσουν το διακριτό PSO και να τον εφαρμόσουν στον ακέραιο προγραμματισμό. Συγκεκριμένα, η Laskari κ.α [51] παρουσίασαν μια υλοποίηση του αλγόριθμου PSO που λειτουργούσε σε δεδομένα συνεχούς χώρου αλλά απέκοπτε τα δεκαδικά ψηφία μετατρέποντας τον αριθμό σε ακέραιο. Στα πειράματα που διεξήχθησαν έγινε σύγκριση μιας απλής αποκοπής και μιας προοδευτικής αποκοπής με μια συμβατική τεχνική κλάδου φράγματος πετυχαίνοντας καλά αποτελέσματα για αρκετά προβλήματα ακέραιου προγραμματισμού.

3.7 Τοπολογία σμήνους

Οι αλγόριθμοι που έχουν επεξηγηθεί μέχρι τώρα επικεντρώνονται κυρίως στον τρόπο με τον οποίο υπολογίζεται η ταχύτητα των σωματιδίων του σμήνους. Μπορεί να διαφέρουν σε ορισμένα βασικά στοιχεία αλλά όλοι βασίζονται στην αρχική έκδοση του αλγορίθμου. Εξίσου σημαντική στη γενικότερη συμπεριφορά του σμήνους είναι και η μέθοδος με την

οποία υπολογίζεται το καλύτερο σωματίδιο (G) αφού επηρεάζει τον τρόπο που τα σωματίδια μπορούν να εξερευνήσουν καινούριες περιοχές (μακριά από το τρέχον καλύτερο σωματίδιο (G^t)).

Στα πρώιμα στάδια της εξέλιξης του αλγορίθμου οι εφευρέτες του, Kennedy και Eberhart, σκέφτηκαν να χρησιμοποιήσουν κάποιου είδους τοπολογία, τον τρόπο δηλαδή που γίνεται η διασύνδεση μεταξύ των σωματιδίων στο σμήνος διαφοροποιώντας εν μέρει το βασικό αλγόριθμο. Η αρχική ιδέα λάμβανε υπόψη τη διάταξη των σωματιδίων στο χώρο αναζήτησης, εννοώντας την απόσταση που έχουν μεταξύ τους στο χώρο. Όμως, η δυσκολία των υπολογισμών και κυρίως τα μη επιθυμητά αποτελέσματα όσον αφορά τη σύγκλιση των σωματιδίων οδήγησαν στην πρόωρη εγκατάλειψη αξιοποίησης της γειτονιάς βασισμένη στην Ευκλείδεια απόσταση.

Μετά την αποτυχημένη αυτή προσπάθεια αξιοποιήθηκε η γενική μορφή όπου κάθε σωματίδιο επηρεάζεται από το καλύτερο σωματίδιο του σμήνους. Η εν λόγω τοπολογία ισοδυναμεί με ένα δίκτυο όπου κάθε σωματίδιο είναι ενωμένο με όλα τα υπόλοιπα και ουσιαστικά θεωρείται όλος ο πληθυσμός σαν μια γειτονιά. Από πλευράς υλοποίησης, αρκεί μονάχα να αποθηκεύεται η θέση του καλύτερου σωματιδίου του σμήνους καθώς επίσης και η τιμή του για σύγκριση στις επόμενες επαναλήψεις. Οι πειραματικές διαδικασίες απέδειξαν ότι αυτή η τοπολογία αν και έχει την τάση να συγκλίνει πολύ γρήγορα σε καλές λύσεις, εντούτοις αποτυγχάνει να φτάσει στη βέλτιστη λύση του προβλήματος.

Μια άλλη τοπολογία που εξετάστηκε και τελικά βρέθηκε ότι υπερτερεί τις προαναφερόμενες είναι η τοπική εκδοχή, όπου κάθε σωματίδιο συνδέεται στα παρακείμενα σωματίδια μέσα σε ένα πληθυσμό. Στη συγκεκριμένη περίπτωση το σμήνος υποδιαιρείται σε αλληλεπικαλυπτόμενες γειτονιές και το σωματίδιο που εξετάζεται αναζητά μέσα στη δική του γειτονιά για να βρει το καλύτερο σωματίδιο. Για παράδειγμα, αν εξετάζεται το σωματίδιο X_i και το μέγεθος της γειτονιάς είναι μήκους k , τότε η γειτονιά του καθορίζεται ως το σύνολο $\{X_{i-k}, \dots, X_{i-1}, X_i, X_{i+1}, \dots, X_{i+k}\}$. Στην περίπτωση όπου το μέγεθος της γειτονιάς είναι δύο ($k = 2$), τότε η τοπολογία του σμήνους μοιάζει σαν ένας δακτύλιος

όπου κάθε σωματίδιο συνδέεται με το άμεσα προηγούμενο και το αμέσως επόμενο του. Η τοπική αυτή εκδοχή πλεονεκτεί έναντι της σφαιρικής εκδοχής γιατί επιτρέπει την παράλληλη αναζήτηση υποπληθυσμών του σμήνους και επομένως στη σύγκλιση σε διαφορετικές περιοχές του χώρου αναζήτησης. Η εξήγηση είναι απλή και έγκειται στο ότι η ροή των πληροφοριών μεταξύ των σωματιδίων του σμήνους είναι περιορισμένη με αποτέλεσμα σωματίδια μιας γειτονιάς να εξερευνούν μια περιοχή ενώ σωματίδια άλλης γειτονιάς να εξερευνούν κάποια άλλη περιοχή. Επιπλέον, μπορεί να συγκλίνει σχετικά με πιο αργό ρυθμό αλλά προάγει την εξερεύνηση και τελικά καταλήγει σε καλύτερες λύσεις αφού δεν επηρεάζεται σε μεγάλο βαθμό από τα τοπικά βέλτιστα στα οποία μπορούν να βρεθούν τα σωματίδια.

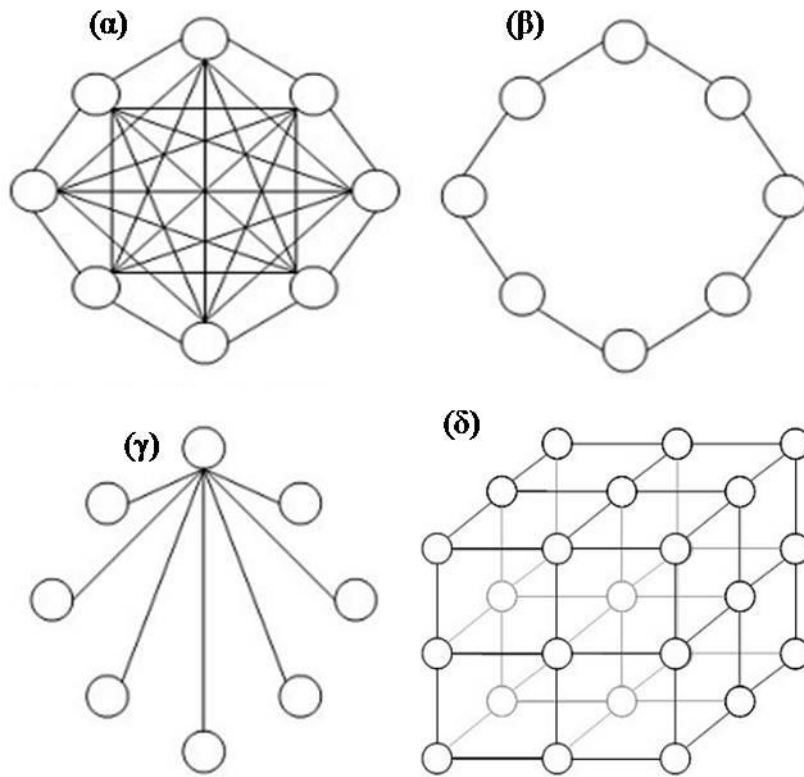
Ο Kennedy [53] το 1999 προσπάθησε να παρατηρήσει την επίδραση που θα είχε στο σμήνος η μεταβολή της τοπολογίας του και επικεντρώθηκε κυρίως σε τέσσερις διασυνδέσεις όμοιες με αυτές των κοινωνικών δικτύων:

- του τροχού, όπου κάθε μέλος του πληθυσμού είναι ενωμένο με όλα τα υπόλοιπα
- του δακτυλίου (τοπική εκδοχή), όπου κάθε μέλος είναι ενωμένο με τους πλησιέστερους του γείτονες
- του αστερά (σφαιρική εκδοχή), όπου όλα τα μέλη του πληθυσμού είναι ενωμένα με το καλύτερο μέλος του πληθυσμού
- των τυχαίων άκρων, όπου κάθε μέλος ενώνεται τυχαία με δύο άλλα μέλη.

Μέσα από πειράματα που διεξήγαγε σε πληθυσμό είκοσι σωματιδίων κατέληξε στο συμπέρασμα ότι η τοπολογία πράγματι έχει επίδραση στην επίδοση του σμήνους, αλλά η βέλτιστη επίδοση εξαρτάται κυρίως από τη μορφή και τη δράση της αντικειμενικής συνάρτησης αξιολόγησης. Επιπλέον, τα πειράματα απέδειξαν ότι στις τοπολογίες όπου η επικοινωνία μεταξύ των σωματιδίων ήταν λιγότερη, το σμήνος μπορούσε να αποφύγει την πρόωρη σύγκλιση σε τοπικά βέλτιστα και να έχει καλύτερα αποτελέσματα.

Έπειτα οι Kennedy και Mendes [54] το 2000 δοκίμασαν χίλιους τριακόσιους σαράντα τρεις τυχαίους γράφους σε διάφορα προβλήματα ώστε να κατανοήσουν την επίδραση που έχει στο σμήνος και να βρουν τη καλύτερη δυνατή τοπολογία. Ανάμεσα τους ήταν η τοπική

εκδοχή, η σφαιρική εκδοχή, η τοπολογία πυραμίδας και η τοπολογία von Neumann. Τα αποτελέσματα απόδειξαν την υπεροχή της τοπολογίας von Neumann η οποία καθορίζει τη γειτονιά σαν ένα πλέγμα με βαθμό $k = 4$.



Σχήμα 3.4. – Οι τοπολογίες (α) τροχού, (β) δακτυλίου, (γ) αστέρα, (δ) von Neumann

Οι προαναφερθείσες τοπολογίες ανήκουν στην κατηγορία των στατικών τοπολογιών αφού η δομή τους παραμένει σταθερή καθ' όλη τη διάρκεια εκτέλεσης του αλγορίθμου. Διάφοροι ερευνητές, ωστόσο, έχουν προτείνει κατά καιρούς τη χρησιμοποίηση δυναμικών τοπολογιών των οποίων η μορφή μεταβάλλεται κατά την προσομοίωση [58]. Ο Sugathan [55] εισηγήθηκε την υλοποίηση μιας τοπολογίας όπου η αναζήτηση ξεκινά με την τοπική εκδοχή του αλγορίθμου και μέχρι το τέλος των επαναλήψεων μετατρέπεται στη σφαιρική εκδοχή. Παρόλο που ο συγγραφέας αναφέρει ότι υπήρξε κάποια βελτίωση με την χρήση της γειτονιάς δυναμικού μεγέθους, εντούτοις τα δημοσιευμένα αποτελέσματα δεν ήταν ξεκάθαρα για την ποιότητά τους. Αντίστοιχα, οι Janson και Middendorf [56] κατέταξαν τα

σωματίδια βάσει μιας δυναμικής ιεραρχίας κατά την οποία κάθε σωματίδιο επηρεάζεται από τη δική του προηγούμενη επιτυχία και από αυτή του σωματιδίου που βρίσκεται πιο πάνω από αυτό. Οι Lian και Sugathan [57] με τη σειρά τους δημιούργησαν υποπληθυσμούς και σε τυχαίο χρόνο έκαναν αναδιάρθρωση των συνδέσεων, πετυχαίνοντας καλά αποτελέσματα στα πειράματα που εκτέλεσαν. Για περισσότερες λεπτομέρειες καθώς και για μια αναλυτική περιγραφή των τοπολογιών η διδακτορική διατριβή του Mendes [58] αποτελεί μια πολύ καλή πηγή πληροφοριών.

3.8 Ειδικοί αλγόριθμοι PSO

Όπως έχει αναφερθεί από τους εφευρέτες του, ο αλγόριθμος PSO είναι ένα ισχυρό εργαλείο το οποίο με την κατάλληλη τροποποίηση μπορεί να εφαρμοστεί με επιτυχία σε πλήθος προβλημάτων. Πέρα από τις βασικές παραλλαγές του αλγορίθμου που έχουν παρουσιαστεί προηγουμένως, υπάρχει μια πληθώρα άλλων που μπορούν να αξιοποιηθούν σε δυναμικά περιβάλλοντα, θορυβώδες περιβάλλοντα, περιβάλλοντα με πολλαπλούς στόχους (multi-objective) και γενικά δυσεπίλυτα προβλήματα.

3.8.1 Δυναμικά περιβάλλοντα

Τα δυναμικά προβλήματα αποτελούν πρόκληση για τον αλγόριθμο PSO μιας και πρόκειται για προβλήματα όπου η αντικειμενική συνάρτηση αξιολόγησης μεταβάλλεται ανάλογα με το χρόνο. Παρουσιάζουν αρκετές δυσκολίες, ιδιαίτερα όταν η βέλτιστη θέση (στόχος) βρίσκεται έξω από το χώρο αναζήτησης του σμήνους. Μερικοί ερευνητές έχουν προτείνει αρκετές επιλογές προς την αντιμετώπιση τους όπως την επαναρχικοποίηση του σμήνους ή τη χρήση του υφιστάμενου αλλά με μηδενισμό των προηγούμενων καλύτερων τιμών. Οι Carlisle και Dozier [59] περιστασιακά επαναξιολογούσαν την προηγούμενη καλύτερη τιμή (P_j) κάποιου συγκεκριμένου σωματιδίου θέτοντας την ίση με την τρέχουσα και αναγκάζοντας το να ξεχάσει την προηγούμενη εμπειρία το ώστε να μειωθεί η επιρροή από προηγούμενες καλές θέσεις. Αντίστοιχα, οι Parrot και Li [60] χρησιμοποίησαν την

προσέγγιση της ενδογένεσης (speciation), εμπνευσμένη από τους γενετικούς αλγόριθμους, για να εντοπίσουν τις δυναμικές κορυφές.

3.8.2 Θορυβώδη περιβάλλοντα

Εξίσου δύσκολες αλλά συνάμα ενδιαφέρουσες είναι οι θορυβώδεις αντικειμενικές συναρτήσεις αξιολόγησης που συναντώνται συχνά σε προβλήματα πραγματικού χρόνου. Με την έννοια θορυβώδης αντικειμενική συνάρτηση εννοείται ότι η αξιολόγηση είναι ασταθής αφού αν ο αλγόριθμος αξιολογήσει το ίδιο διάνυσμα πολλές φορές, η αποτίμηση μπορεί να διαφέρει. Οι Parsopoulos και Vrahatis [61] μελέτησαν την συμπεριφορά του αλγορίθμου με την πρόσθεση τυχαίου θορύβου που ακολουθεί Γκαουσιανή κατανομή και με την πραγματοποίηση τυχαίων περιστροφών του χώρου έρευνας. Τα αποτελέσματα των πειραμάτων απέδειξαν ότι ο αλγόριθμος PSO παραμένει αποτελεσματικός στην παρουσία θορύβου και σε ορισμένες περιπτώσεις ο θόρυβος βοήθησε τον αλγόριθμο στην αποφυγή των τοπικών βέλτιστων.

3.8.3 Περιβάλλοντα με πολλαπλούς στόχους

Εκτός από τα παραπάνω προβλήματα υπάρχουν και περιπτώσεις όπου ένα πρόβλημα έχει περισσότερες από μια συναρτήσεις αξιολόγησης που πρέπει να βελτιστοποιηθούν ταυτόχρονα, τα επονομαζόμενα multi-objective προβλήματα βελτιστοποίησης. Συχνά, οι λύσεις που προτείνονται φέρνουν σε σύγκρουση τις επιμέρους συναρτήσεις με αποτέλεσμα η καλύτερη λύση να είναι ένα αποτέλεσμα συμβιβασμού. Το σύνολο των περιοχών λύσεων των συναρτήσεων συνήθως περιλαμβάνει πολλαπλά σημεία, και έτσι υπάρχει ένα σύνολο λύσεων (το σύνολο Pareto) που μπορεί να ικανοποιεί εξίσου τις συγκρουόμενες συναρτήσεις. Ο Coello Coello [62] ανέπτυξε μια παραλλαγή του αλγορίθμου τον οποίο ονόμασε MOPSO (Multi-Objective PSO) στην προσπάθεια του να επιλύσει το εν λόγω πρόβλημα. Ο αλγόριθμος MOPSO επιβάλλει ένα είδος πλέγματος στο χώρο αναζήτησης και διατηρεί μια αποθήκη όπου φυλάει τα διανύσματα που πληρούν τα κριτήρια βελτιστοποίησης. Όπως αναφέρεται στο σχετικό άρθρο, ο MOPSO παρουσιάζει καλά αποτελέσματα και είναι ανταγωνιστικός συγκριτικά με άλλους αλγορίθμους που

αντιμετωπίζουν το ίδιο πρόβλημα. Επίσης, οι Parsopoulos και Vrahatis [63] χρησιμοποίησαν τεχνικές εμπνευσμένες από τους γενετικούς αλγορίθμους ώστε να κατορθώσουν να αξιολογούν πολλές συναρτήσεις ταυτόχρονα. Ο αλγόριθμος VEPSO (Vector Evaluated PSO), όπως τον ονόμασαν, διαιρεί τον αρχικό πληθυσμό σε υπο-σμήνη (όσα και ο αριθμός των συναρτήσεων αξιολόγησης) και σε κάθε υπο-σμήνος ανατίθεται μια συνάρτηση αξιολόγησης. Κάθε υπο-σμήνος αξιολογείται βάσει της δικής του συνάρτησης αλλά η κοινωνική επίδραση επέρχεται από το καλύτερο σωματίδιο κάποιου άλλου υπο-σμήνους. Τέλος, οι Hu και Eberhart [64] μέσα στην έρευνά τους τονίζουν ότι δεν υπάρχει κάποια «μαγική συνταγή» που να βελτιστοποιεί την απόδοση του αλγορίθμου στα multi-objective προβλήματα.

3.8.4 Υβριδικοί αλγόριθμοι PSO

Πολλοί ερευνητές έχουν προσπαθήσει να τροποποιήσουν τη μορφή του αλγορίθμου ενσωματώνοντας τεχνικές από άλλες μεθόδους του εξελικτικού υπολογισμού δίνοντας έτσι μια καινούρια δυναμική στον αλγόριθμο και παράλληλα την ευχέρεια να ξεπερνά τα οποιαδήποτε προβλήματα παρουσιάζονται κατά την εκτέλεσή του, όπως για παράδειγμα την παγίδευση σε τοπικό βέλτιστο. Το σύνολο των αλγορίθμων που συνδυάζουν περισσότερες από μια μεθόδους ονομάζονται υβριδικοί αλγόριθμοι. Πρόκειται για αλγορίθμους που τα μειονεκτήματα που παρουσιάζει η μια μέθοδος αντισταθμίζονται ή στην καλύτερη περίπτωση εκμηδενίζονται από την άλλη μέθοδο. Ερευνητές όπως οι Miranda και Fonseca [65] και Loønbjerg [66] συνδύασαν τον αλγόριθμο PSO με τους γενετικούς αλγορίθμους, ενώ οι Poli και Stephens [67, 68] πρότειναν τον υβριδικό PSO με τη χρήση ενός hill climber.

3.9 Προβλήματα

Ένα από τα κύρια πλεονεκτήματα του αλγορίθμου PSO σε σχέση με τους υπόλοιπους εξελικτικούς αλγορίθμους είναι η γρήγορη σύγκλιση του σμήνους στο βέλτιστο σημείο, φτάνοντας παράλληλα στη βέλτιστη λύση. Υπάρχουν φορές όμως που η συμπεριφορά του αυτή αποδεικνύεται προβληματική, αφού το σμήνος δεν οδηγείται πάντοτε στο βέλτιστο

σημείο αλλά σε κάποιο τοπικό βέλτιστο με αποτέλεσμα να παραμένει στάσιμο και να μην συνεχίζει την εξερεύνηση. Έχουν αναπτυχθεί αρκετές τεχνικές ώστε να διορθωθεί το συγκεκριμένο πρόβλημα και να κατορθώνει το σμήνος να ξεφεύγει από τα τοπικά βέλτιστα όπως, για παράδειγμα, η εισαγωγή νέων σωματιδίων στο σμήνος, η ανταλλαγή των διανυσμάτων των ταχυτήτων μεταξύ σωματιδίων του σμήνους, η δυναμική μεταβολή του βάρους της αδράνειας και η δυναμική μεταβολή των συντελεστών επιτάχυνσης,

Οι van den Bergh και Engelbrecht [69] πρότειναν τον αλγόριθμο GCPSO (Guaranteed Convergence PSO), ο οποίος χρησιμοποιεί μια διαφορετική εξίσωση για την ανανέωση της ταχύτητας του καλύτερου σωματιδίου αναγκάζοντας το σωματίδιο να ταλαντωθεί τυχαία γύρω από τη θέση του σε μια ακτίνα αναζητώντας μια καλύτερη θέση. Τα πειράματα είχαν καλά αποτελέσματα για τον GCPSO και το γεγονός ότι μπορεί να λειτουργήσει αποτελεσματικά με σμήνη μικρού μεγέθους, το κατατάσσει στα ψηλά επίπεδα των τεχνικών απεμπλοκής από τα τοπικά βέλτιστα. Το μοναδικό του μειονέκτημα παρουσιάζεται στο ότι η ακτίνα εξαρτάται από ένα παράγοντα κλιμάκωσης, στοιχείο που απαιτεί κάποια γνώση εκ των προτέρων για το είδος του προβλήματος.

Ένα ακόμα στοιχείο που δείχνει ότι ένα σμήνος παραμένει στάσιμο και κολλημένο σε ορισμένα σημεία είναι τα χαμηλά επίπεδα ποικιλίας που παρουσιάζονται, λόγω του ότι τα σωματίδια του σμήνους κατέχουν τις ίδιες θέσεις μέσα στο χώρο αναζήτησης. Οι Riget και Versterstrøm [70] προσπάθησαν να επιλύσουν το πρόβλημα υλοποιώντας ένα μέτρο ελέγχου της ποικιλομορφίας που άλλοτε προσέλκυε τα σωματίδια προς το καλύτερο του σμήνους και άλλοτε τα απωθούσε. Αρχικά, όλα τα σωματίδια προσέλκυαν το ένα το άλλο μέχρι που έφτανε στο ελάχιστο όριο η ποικιλομορφία του σμήνους και τότε τα σωματίδια απωθούσαν το ένα το άλλο μέχρι που η ποικιλομορφία να φτάσει στο μέγιστό της. Έγιναν πειραματικές διαδικασίες ανάμεσα στη συγκεκριμένη προσέγγιση και τις βασικές μορφές του αλγόριθμου PSO και των γενετικών αλγορίθμων και τα αποτελέσματα απέδειξαν την υπεροχή της σε όλες τις μορφές πειραμάτων που διεξήχθησαν.

Ο Ratnaweera κ.α. [71] πρότειναν δύο επιπλέον προσεγγίσεις του αλγόριθμου PSO στην προσπάθεια επίλυσης της στασιμότητας με τη χρήση μεταβαλλόμενων ανάλογα με το

χρόνο συντελεστών επιτάχυνσης. Η συγκεκριμένη προσέγγιση αποσκοπούσε στη μείωση της κοινωνικής επίδρασης στα αρχικά στάδια της εκτέλεσης του αλγορίθμου ώστε να αποτραπεί η πρόωρη σύγκλιση και να γίνει εξερεύνηση του χώρου. Τα πειράματα που πραγματοποίησε έδειξαν ότι μπορούν να επιτευχθούν καλά αποτελέσματα αν τα c_1 και c_2 αρχικοποιηθούν με 2.5 και 0.5 αντίστοιχα και κατά τη διάρκεια εκτέλεσης του αλγορίθμου μεταβάλλονται οι τιμές του έτσι ώστε στο τέλος το c_1 να είναι 0.5 και το c_2 να ισούται με 2.5. Το σκεπτικό της όλης προσέγγισης έγκειται στο ότι αρχικά πρέπει να γίνεται εξερεύνηση του χώρου αναζήτησης από το σμήνος και στην τελική φάση να προωθείται η εκμετάλλευση ώστε να μπορούν τα σωματίδια να φτάσουν στη βέλτιστη λύση.

Μια πιο αναλυτική παρουσίαση των προσεγγίσεων που παρουσιάστηκαν κατά καιρούς από τους ερευνητές για επίλυση των προβλημάτων υπάρχει στο άρθρο του Banks κ.α. [72].

3.10 Σύνοψη

Σε αυτό το κεφάλαιο έγινε προσπάθεια να παρουσιαστεί ο βασικός αλγόριθμος PSO, οι κυριότερες παραλλαγές του καθώς επίσης και ορισμένες παραμέτροι που επηρεάζουν την επίδοση και αποτελεσματικότητα του αλγορίθμου. Είναι γεγονός ότι οι δυνατότητες του είναι τεράστιες, γι' αυτό άλλωστε χρησιμοποιείται σε πολλούς τομείς όπως στην ανάλυση σημάτων, στη βελτιστοποίηση συνδυαστικών προβλημάτων, στη ρομποτική, σε εφαρμογές γραφικών, στη μεταλλουργία, στα χρηματοοικονομικά, στο βιομηχανικό μοντελισμό, και στο χρονοπρογραμματισμό.

Με βάση την έρευνα που παρουσίασαν ο Poli κ.α. [68], από το 1995 που πρωτοεμφανίστηκε ο αλγόριθμος έως το 2007 έχουν γραφτεί πάνω από 1100 άρθρα βασισμένα στον αλγόριθμο PSO. Κάποια αφορούν τροποποιήσεις και παραλλαγές του αλγορίθμου για να βελτιωθεί η επίδοση και η αποτελεσματικότητα του και κάποια αναφέρονται σε εφαρμογές του σε διάφορους τομείς και βιομηχανίες. Το αξιοσημείωτο σε αυτή την έρευνα όμως, είναι ότι τα άρθρα αυξάνονται συνεχώς χρόνο με το χρόνο. Αρκεί

κάποιος να κοιτάζει το πιο κάτω σχήμα για να καταλάβει την τεράστια έρευνα που γίνεται στον PSO και να αναλογιστεί τις δυνατότητες του.

Year	Google Scholar Hits	IEEE Xplore Publications
1995	■ (28)	(2)
1996	(14)	(0)
1997	■ (24)	(3)
1998	■ (49)	(4)
1999	■ (62)	(7)
2000	■ (78)	(8)
2001	■ (118)	■ (11)
2002	■ (256)	■ (46)
2003	■ (373)	■ (82)
2004	■ (850)	■ (181)
2005	■ (1210)	■ (279)
2006	■ (1230)	■ (462)

Σχήμα 3.5. – Συγκριτικός πίνακας δημοσιεύσεων για τον αλγόριθμο PSO ανά έτος

Ο συνδυασμός ενός τόσο δυνατού εργαλείου όπως ο αλγόριθμος PSO με το δυσεπίλυτο πρόβλημα του χρονοπρογραμματισμού ενός έργου περιορισμένων πόρων αποτέλεσε μια ιδιαίτερη ελκυστική ευκαιρία για εφαρμογή του στην περιοχή της Τεχνολογίας Λογισμικού και συγκεκριμένα στην εξεύρεση του βέλτιστου χρονοπρογράμματος και στελέχωσης ομάδας ενός έργου ανάπτυξης λογισμικού. Η μεθοδολογία που ακολουθήθηκε καθώς επίσης και ο τρόπος μοντελοποίησης του προβλήματος περιγράφονται στο κεφάλαιο που ακολουθεί.

Κεφάλαιο 4

Μοντελοποίηση προβλήματος και υλοποίηση πρωτοτύπου

- 4.1 Εισαγωγή στη μεθοδολογία υλοποίησης
 - 4.2 Μοντελοποίηση
 - 4.3 Αντικειμενική συνάρτηση αξιολόγησης
 - 4.3.1 Αλληλεξαρτήσεις μεταξύ διεργασιών
 - 4.3.2 Πλήρωση απαιτούμενων ικανοτήτων
 - 4.3.3 Διαθεσιμότητα υπαλλήλων
 - 4.4 Απεμπλοκή από στασιμότητα
 - 4.5 Υλοποίηση πρωτοτύπου
-

4.1 Εισαγωγή στη μεθοδολογία υλοποίησης

Είναι νομίζω, αντιληπτό στον καθένα ότι η θέση του διαχειριστή έργου είναι ζωτικής σημασίας στη διαδικασία παραγωγής λογισμικού. Οι αποφάσεις που λαμβάνει τόσο πριν όσο και κατά τη διάρκεια παραγωγής λογισμικού έχουν μεγάλη επίδραση στο τελικό αποτέλεσμα και συνήθως κρίνουν αν το έργο θα επιτύχει να εκπληρώσει τους στόχους του ή όχι. Υπάρχουν φορές που οι αποφάσεις του για τη σειρά εκτέλεσης των επιμέρους εργασιών του έργου αντί να διευκολύνουν την αλληλοδιαδοχή των εργασιών και να αξιοποιούν στο μέγιστο τους διαθέσιμους πόρους, αποτελούν τροχοπέδη στην όλη διαδικασία παραγωγής. Άλλοτε πάλι, οι αποφάσεις του όσον αφορά τη σύσταση των υποομάδων για την ανάθεσή τους στις επιμέρους εργασίες δεν είναι η κατάλληλη με

συνέπεια να δημιουργούνται μη λειτουργικές και αναποτελεσματικές ομάδες εργασίας. Αποτέλεσμα όλων των παραπάνω είναι το τελικό προϊόν πολλές φορές να παραδίδεται εκπρόθεσμα, με πολλά λάθη, παραβλέψεις και παρατυπίες και συνεπώς να οδηγείται με μαθηματική ακρίβεια στην αποτυχία. Η εξεύρεση λοιπόν μιας μεθόδου που θα βοηθά το διαχειριστή έργου στη λήψη καίριων αποφάσεων δεν αποτελεί μια προαιρετική και μελλοντική εργασία, αλλά μια άμεση ανάγκη που απαιτεί την επίλυσή της το συντομότερο.

Το συγκεκριμένο πρόβλημα, το οποίο έχει αναλυθεί εκτενώς στα προηγούμενα κεφάλαια, προσπαθεί να αντιμετωπίσει η παρούσα διπλωματική εργασία. Η προσοχή εστιάζεται στην εξεύρεση της βέλτιστης σειράς εκτέλεσης των επιμέρους εργασιών ενός έργου λογισμικού διατηρώντας τις αλληλεξαρτήσεις που υπάρχουν μεταξύ των εργασιών του έργου και στη δημιουργία λειτουργικών και αποτελεσματικών ομάδων εργασίας αξιοποιώντας στο μέγιστο δυνατό βαθμό τους διαθέσιμους πόρους. Γι' αυτό έχει επιλεγεί ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων αφού αποδεδειγμένα πρόκειται για ένα αξιόπιστο εργαλείο, που με την κατάλληλη προσαρμογή στα δεδομένα του εκάστοτε προβλήματος, μπορεί να οδηγήσει σε καλές και αποτελεσματικές λύσεις.

4.2 Μοντελοποίηση

Αρχικά υιοθετήθηκαν οι παρακάτω υποθέσεις ώστε ο αλγόριθμός να τεθεί σε μια σταθερή βάση και να μπορεί να γίνει μοντελοποίηση του:

- i. Καμιά διεργασία δεν μπορεί να διακοπεί αφού ξεκινήσει η εκτέλεσή της αλλά αντίθετα πρέπει να ολοκληρωθεί.
- ii. Κάθε διεργασία με την ολοκλήρωση της απελευθερώνει τους πόρους (υπάλληλοι) που απασχολεί.
- iii. Κανένας υπάλληλος δεν μπορεί να διακόψει την απασχόλησή του σε μια συγκεκριμένη διεργασία, πριν ολοκληρωθεί, και να ανατεθεί σε κάποια άλλη.
- iv. Σε κάθε διεργασία πρέπει να έχει ανατεθεί τουλάχιστον ένας υπάλληλος.
- v. Ένας υπάλληλος δεν μπορεί να εργάζεται σε περισσότερες από μια διεργασία την ίδια χρονική στιγμή.

- vi. Ο καθορισμός των ικανοτήτων που διαθέτουν οι υπαλλήλοι είναι ευθύνη του διαχειριστή έργου και αυτές είναι γνωστές εκ των προτέρων.
- vii. Ο καθορισμός των εξαρτήσεων που υπάρχουν μεταξύ των διεργασιών του έργου είναι ευθύνη του διαχειριστή έργου και οι εξαρτήσεις είναι γνωστές εκ των προτέρων.
- viii. Ο καθορισμός της χρονικής διάρκειας της διεργασίας καθώς επίσης και των ικανοτήτων που απαιτούνται για την εκπλήρωσή της είναι ευθύνη του διαχειριστή έργου και τα στοιχεία αυτά είναι γνωστά εκ των προτέρων.

Η ενσωμάτωση των δεδομένων που συγκροτούν το γενικότερο πλαίσιο της διαδικασίας παραγωγής έργου στα συστατικά στοιχεία του αλγορίθμου ήταν ομολογουμένως ένα από τα πιο δύσκολα σημεία που παρουσιάστηκαν στη διαδικασία μοντελοποίησης. Λανθασμένη αναπαράσταση των δεδομένων θα είχε καταστροφικά αποτελέσματα και η όλη προσπάθεια θα πήγαινε στράφι. Επομένως έπρεπε να επιλεγεί η κατάλληλη αναπαράσταση ώστε οι δυνατότητες του αλγορίθμου να αξιοποιηθούν στο έπακρο.

Το πρόβλημα αποτελείται από δύο κύρια δεδομένα, τις διεργασίες και τους υπαλλήλους. Λαμβάνοντας υπόψη τη γενική μορφή του αλγορίθμου και τα προαναφερθέντα δεδομένα επιλέχθηκε η επέκταση του τρόπου αναπαράστασης της θέσεως του σωματιδίου στο χώρο αναζήτησης ώστε να συμπεριλαμβάνει και τα δύο. Παρακάτω επεξηγείται η αναπαράσταση που χρησιμοποιήθηκε για τη μοντελοποίηση των δεδομένων.

Έστω ένα έργο που αποτελείται από t διεργασίες και x_i το i -οστό σωματίδιο του σμήνους. Τότε η θέση του στο χώρο υποδεικνύεται από το διάνυσμα:

$$x_i = (\text{task}_1, \text{task}_2, \dots, \text{task}_t) \quad (13)$$

Αντίστοιχα, κάθε διεργασία (task) αναπαριστά το χρόνο έναρξης της και τους υπαλλήλους που έχουν ανατεθεί σε αυτή με την εξής μορφή:

$$\text{task}_i = \{\text{startT}, \{\text{emp}_1, \dots, \text{emp}_q\}\} \quad (14)$$

όπου q είναι ο αριθμός των υπαλλήλων που διατίθενται για την εκτέλεση του έργου.

Η γενική μορφή του i -οστού σωματιδίου είναι:

$$x_i = (\{startT_1, \{emp_k, \dots, emp_l\}\}, \dots, \{startT_b, \{emp_m, \dots, emp_n\}\}) \quad (15)$$

Πρέπει να σημειωθεί ότι η κωδικοποίηση των υπαλλήλων γίνεται με τη χρήση της δυαδικής ακολουθίας. Έτσι οι δυνατές τιμές που μπορεί να έχει ο υπάλληλος emp_i είναι ένα (1) και μηδέν (0), που υποδεικνύουν ότι ο συγκεκριμένος υπάλληλος έχει (1) ή δεν έχει (0) ανατεθεί στη διεργασία. Αντίστοιχα, ο χρόνος έναρξης της κάθε διεργασίας είναι ένας ακέραιος αριθμός στο πεδίο $[1, maxPosition]$ όπου

$$maxPosition = \text{συνολική διάρκεια έργου} - \text{διεργασία με μικρότερη διάρκεια} \quad (16)$$

Το σκεπτικό για την πιο πάνω προσαρμογή έγκειται στο ότι η ελάχιστη ημέρα έναρξης που μπορεί να ξεκινήσει να εκτελείται μια διεργασία είναι η ημέρα που ξεκινά η εκτέλεση του έργου, δηλαδή ένα. Ανάλογα η μέγιστη ημέρα έναρξης μιας διεργασίας είναι η συνολική διάρκεια εκτέλεσης του έργου αν όλες οι διεργασίες εκτελεστούν σειριακά μείον τον απαιτούμενο χρόνο της διεργασίας με τη μικρότερη διάρκεια.

Το πιο κάτω παράδειγμα είναι διαφωτιστικό για τον τρόπο παρουσίασης των δεδομένων. Έστω ότι το έργο προς εκτέλεση αποτελείται από τρεις διεργασίες και για την υλοποίησή του η εταιρεία διαθέτει τέσσερις υπαλλήλους. Επιπλέον, έστω ότι η θέση του i -οστού σωματιδίου του σμήνους έχει την εξής μορφή:

$$x_i = (\{3, \{0100\}\}, \{6, \{1110\}\}, \{5, \{1010\}\})$$

Αυτό σημαίνει ότι η πρώτη διεργασία θα ξεκινήσει να εκτελείται την τρίτη ημέρα και σ' αυτή έχει ανατεθεί ο δεύτερος υπάλληλος. Οι υπόλοιπες διεργασίες θα ξεκινήσουν να

εκτελούνται την έκτη και πέμπτη μέρα από τους υπαλλήλους ένα, δύο, τρία και ένα, τρία αντίστοιχα.

Κομβικό σημείο στη διαδικασία υλοποίησης ήταν η επιλογή της παραλλαγής του αλγορίθμου που θα χρησιμοποιείτο. Σημαντικό πλεονέκτημα στην διαδικασία επιλογής ήταν ότι παρόλο που οι πλείστες παραλλαγές του PSO εφαρμόζονται σε προβλήματα συνεχούς χώρου, εντούτοις οι ερευνητές αναφέρουν ότι δεν είναι απαραίτητη η χρήση οποιουδήποτε εξειδικευμένου αλγορίθμου για προβλήματα ακέραιου προγραμματισμού. Μπορούν να χρησιμοποιηθούν οι υφιστάμενοι αλγορίθμοι αξιοποιώντας τεχνικές αποκοπής των δεκαδικών ψηφίων. Μετά από μελέτη αρκετών άρθρων που αφορούσαν εισηγήσεις από διάφορους ερευνητές για επίλυση του RCPSP χρησιμοποιώντας τον αλγόριθμο PSO, επιλέχθηκε η χρήση τόσο του PSO με επίδραση του βάρους αδρανείας όσο και του διακριτού PSO. Ο μεν πρώτος θα αξιοποιείτο στην αναπαράσταση και μεταβολή της ημέρας έναρξης κάθε διεργασίας και ο δεύτερος στις αντίστοιχες διαδικασίες των υπαλλήλων του έργου.

4.3 Αντικειμενική συνάρτηση αξιολόγησης

Ο Kennedy [53] αναφέρει ότι η συνάρτηση αξιολόγησης έχει πρωτεύοντα ρόλο στη σύγκλιση του σμήνους και στην εξεύρεση της βέλτιστης λύσης του προβλήματος, αφού αποτελεί κατά κάποιο τρόπο την πυξίδα στην περιπλάνηση του σμήνους στο χώρο αναζήτησης. Μια καλά σχεδιασμένη και αποτελεσματική συνάρτηση καθοδηγεί το σμήνος προς τη βέλτιστη λύση ενώ μια κακοσχεδιασμένη το αποπροσανατολίζει και το κατευθύνει σε λανθασμένες περιοχές. Συνεπώς, καταλαβαίνει κανείς την τεράστια σημασία που έχει η επιλογή της κατάλληλης συνάρτησης στην επιτυχία του αλγορίθμου και περαιτέρω στην επιτυχία επίλυσης του προβλήματος.

Έχοντας κατά νου τα δεδομένα του προβλήματος και την οπτική γωνιά από την οποία προσεγγίζει η διπλωματική αυτή το RCPSP, γίνεται αντιληπτό ότι η συνάρτηση αξιολόγησης πρέπει να λαμβάνει υπόψη τρεις παράγοντες: τις αλληλεξαρτήσεις των διεργασιών μεταξύ τους, τις απαιτούμενες ικανότητες που πληρούνται από τους

υπαλλήλους που έχουν ανατεθεί σε μια διεργασία και τη διαθεσιμότητα των υπαλλήλων. Η γενική μορφή της αντικειμενικής συνάρτησης αξιολόγησης έχει ως εξής:

$$R = \frac{f(D_p) + f(S_p) + f(A_p)}{3} \quad (17)$$

όπου R: το γενικό αποτέλεσμα της αξιολόγησης ενός σωματιδίου. Ουσιαστικά είναι ο σταθμισμένος μέσος όρων των τριών επιμέρους συναρτήσεων.

p: το p-οστό σωματίδιο του σμήνους που εξετάζεται

$f(D_p)$: το αποτέλεσμα της αξιολόγησης του σωματιδίου όσον αφορά τις αλληλεξαρτήσεις των διεργασιών μεταξύ τους

$f(S_p)$: το αποτέλεσμα της αξιολόγησης του σωματιδίου για τις ικανότητες των υπαλλήλων που έχουν ανατεθεί στις διεργασίες

$f(A_p)$: το αποτέλεσμα της αξιολόγησης του σωματιδίου που αφορά τη διαθεσιμότητα των υπαλλήλων σε κάθε χρονική στιγμή

4.3.1 Αλληλεξαρτήσεις μεταξύ διεργασιών

Πρόκειται για την πλέον σημαντική συνάρτηση αξιολόγησης αφού καθορίζει τη σειρά με την οποία θα εκτελεστούν οι διεργασίες ενός έργου. Για να θεωρείται εφικτός ο χρονοπρογραμματισμός ενός έργου πρέπει οι αλληλεξαρτήσεις μεταξύ των διεργασιών να μην παραβιάζονται, εννοώντας ότι καμιά διεργασία δεν μπορεί να ξεκινήσει την εκτέλεσή της πριν να ολοκληρωθούν οι διεργασίες από τις οποίες εξαρτάται. Για παράδειγμα, έστω ένα έργο που αποτελείται από τις διεργασίες x, y, και z με διάρκεια εκτέλεσης, αλληλεξαρτήσεις και μέρα έναρξης όπως παρουσιάζεται στον πιο κάτω πίνακα:

Πίνακας 4.1 – Στοιχεία έργου και επιμέρους διεργασιών

Διεργασία	Διάρκεια εκτέλεσης	Εξαρτήσεις	Μέρα έναρξης
x	3	-	3
y	6	-	1
z	4	x, y	???

Σύμφωνα λοιπόν, με το πιο πάνω παράδειγμα η διεργασία x ολοκληρώνεται την πέμπτη μέρα ενώ η y την έκτη. Οποιαδήποτε μέρα έναρξης της z πριν την ολοκλήρωση των x και y, δηλαδή έβδομη μέρα, καθιστά το χρονοπρόγραμμα ανέφικτο.

Όμως εκτός από τη διατήρηση των περιορισμών μεταξύ των διεργασιών, εξίσου σημαντικό για την ελαχιστοποίηση του χρόνου εκτέλεσης του έργου είναι και η μείωση του “κενού” χρόνου (slack time) που παρουσιάζεται μεταξύ των διεργασιών. “Κενός” χρόνος είναι η περίπτωση κατά την οποία μια διεργασία που μπορεί να ξεκινήσει την εκτέλεση της άμεσα, δρομολογείται για αργότερα επηρεάζοντας τις υπόλοιπες διεργασίες με αποτέλεσμα να παρατηρείται καθυστέρηση στο συνολικό χρόνο ολοκλήρωσης του έργου. Η καθυστέρηση αυτή πρέπει να μειωθεί στο ελάχιστο δυνατό ή στην καλύτερη περίπτωση να εξαλειφθεί εντελώς αφού δεν προσφέρει τίποτα εποικοδομητικό στη διαδικασία εκτέλεσης του έργου.

Με βάση τις προαναφερθείσες παραμέτρους η εξίσωση αξιολόγησης μιας διεργασίας έχει αποφασισθεί ως:

$$f(task_i) = \begin{cases} 0 & \text{αν } startDate_i < \max(finishDate_j), \quad j \in P_{task_i} \\ \frac{1}{1 + delay} & \text{διαφορετικά} \end{cases} \quad (18)$$

όπου $task_i$: η i-οστή διεργασία του σωματιδίου που εξετάζεται

$startDate_i$: η μέρα έναρξης εκτέλεσης διεργασίας

$\max(finishDate_j)$: η μέγιστη μέρα ολοκλήρωσης όλων των διεργασιών από τις οποίες εξαρτάται η διεργασία που εξετάζεται

P_{task_i} : το σύνολο των διεργασιών από τις οποίες εξαρτάται η διεργασία i

delay: η καθυστέρηση που παρατηρείται στην έναρξη εκτέλεσης της διεργασίας

Στη συνέχεια προσθέτονται οι αξιολογήσεις όλων των σημείων (διεργασιών) του σωματιδίου που εξετάζεται και υπολογίζεται ο σταθμισμένος μέσος όρος, ο οποίος αποτελεί την τελική αξιολόγηση του σωματιδίου όσον αφορά τις αλληλεξαρτήσεις μεταξύ των διεργασιών.

Το πιο κάτω παράδειγμα παρουσιάζει τους υπολογισμούς που απαιτούνται για την εύρεση της αξιολόγησης του p-οστού σωματιδίου του σμήνους.

Πίνακας 4.2 – Παράδειγμα διεργασιών για αξιολόγηση των μεταξύ τους αλληλεξαρτήσεων

Διεργασία	Διάρκεια εκτέλεσης	Εξαρτήσεις	Μέρα έναρξης
X	2	-	3
Y	5	-	1
Z	4	x, y	8

$$f(D_p) = \begin{cases} f(task_x) = \frac{1}{1 + (3 - 1)} = \frac{1}{1 + 2} = \frac{1}{3} \\ f(task_y) = \frac{1}{1 + (1 - 1)} = \frac{1}{1 + 0} = \frac{1}{1} \\ f(task_z) = \frac{1}{1 + (8 - 6)} = \frac{1}{1 + 2} = \frac{1}{3} \end{cases} \rightarrow f(D_p) = Avg(f(D_{task_i})) = \frac{5}{9}$$

4.3.2 Πλήρωση απαιτούμενων ικανοτήτων

Βέβαια, πολλές φορές τα έργα λογισμικού μπορεί να παραδίδονται εντός των επιτρεπτών χρονικών ορίων, αλλά υποφέρουν από έλλειψη ποιότητας και παρατυπίες με αποτέλεσμα να κρίνονται ως ελαττωματικά. Η πιθανή αιτία που συμβαίνει κάτι τέτοιο είναι ότι οι ομάδες εργασίας που έχουν συγκροτηθεί κατά τη διάρκεια εκτέλεσης του έργου, και έχουν ανατεθεί στις επιμέρους διεργασίες, δεν πληρούν τις απαραίτητες ικανότητες που

απαιτούνται για την επιτυχή ολοκλήρωση της εκάστοτε διεργασίας. Απότοκο αυτού είναι η προβληματική εκτέλεση της διεργασίας με επακόλουθο την ανασφαλή πορεία του έργου.

Συνεπώς, η διπλωματική αυτή εργασία δεν επικεντρώνει τα ερευνητικά φώτα μόνο στην εξεύρεση της βέλτιστης σειράς εκτέλεσης των διεργασιών, αλλά παράλληλα στοχεύει στη δημιουργία λειτουργικών ομάδων εργασίας αξιοποιώντας τους διαθέσιμους πόρους. Με αυτό τον τρόπο αυξάνονται τόσο οι πιθανότητες επιτυχημένης ολοκλήρωσης κάθε διεργασίας ξεχωριστά, όσο και της συνολικής ποιότητας του έργου. Επομένως, στην αξιολόγηση των σωματιδίων του σμήνους και στην προσπάθεια εξεύρεσης της βέλτιστης λύσης του προβλήματος πρέπει να ληφθούν σοβαρά υπόψη οι ικανότητες που απαιτούνται για τη διεκπεραίωση της διεργασίας καθώς επίσης και οι αντίστοιχες ικανότητες που διαθέτουν οι υπαλλήλοι που έχουν ανατεθεί στη συγκεκριμένη διεργασία.

Εξίσου σημαντικός παράγοντας που καθορίζει την αποτελεσματικότητα μιας ομάδας εργασίας είναι η συνοχή μεταξύ των υπαλλήλων, στοιχείο το οποίο πρέπει να διατηρείται σε υψηλό επίπεδο. Συγκεκριμένα, μια ομάδα εργασίας δεν πρέπει να είναι μεγάλη σε μέγεθος ούτε να περιέχει ανενεργούς υπαλλήλους, όπως επίσης και υπαλλήλους που δεν προσφέρουν καμιά επιπλέον γνώση στην ομάδα. Αν προσμετρηθούν τα παραπάνω στη γενικότερη δομή της ομάδας τότε επιτυγχάνεται η δημιουργία λειτουργικών και αποτελεσματικών ομάδων που μπορούν να φέρουν σε πέρας την εκάστοτε διεργασία.

Βάσει των προαναφερθέντων παραμέτρων η εξίσωση αξιολόγησης των ικανοτήτων μιας διεργασίας έχει καθοριστεί ως:

$$f(task_i) = Avg \left(\frac{1}{teamSize} + \frac{1}{1 + \#idleEmps + \#extraEmps} \right) \times \frac{1}{1 + \#unsSkills} \quad (19)$$

όπου $task_i$: η i-οστή διεργασία του σωματιδίου που εξετάζεται

$teamSize$: το πλήθος των υπαλλήλων που έχουν ανατεθεί στη διεργασία

$\#idleEmps$: το πλήθος των υπαλλήλων που έχουν ανατεθεί στην ομάδα και δεν διαθέτουν κανένα από τις απαιτούμενες ικανότητες

#extraEmps: το πλήθος των υπαλλήλων που δεν προσφέρουν καμιά απαραίτητη ικανότητα στην ομάδα για την εκτέλεση της διεργασίας

#unsSkills: το πλήθος των απαιτούμενων ικανοτήτων που δεν πληρούνται από κανένα υπάλληλο που έχει ανατεθεί στην ομάδα

Έπειτα προσθέτονται οι αξιολογήσεις όλων των σημείων (διεργασιών) του σωματιδίου που εξετάζεται και ακολούθως υπολογίζεται ο σταθμισμένος μέσος όρος, που αποτελεί την τελική αξιολόγηση του σωματιδίου όσον αφορά την πλήρωση των απαιτούμενων ικανοτήτων των διεργασιών.

Στο παρακάτω παράδειγμα παρουσιάζονται οι απαραίτητοι υπολογισμοί για την αποτίμηση της αξιολόγησης του p-οστού σωματιδίου του σμήνους.

Σημ.: οι ικανότητες κωδικοποιούνται ως ακέραιοι αριθμοί για χάριν του παραδείγματος.

Πίνακας 4.3. – Παράδειγμα διεργασιών και υπαλλήλων για αξιολόγηση των ικανοτήτων

Διεργασία	Απαιτούμενες ικανότητες	Υπαλλήλοι – ικανότητες
X	1, 2, 3	υπ _α :1,2 υπ _β :3
Y	3, 5, 6	υπ _α :1,2 υπ _β :3 υπ _γ :5,6
Z	4	υπ _δ :4 υπ _ε :4,5

$$f(S_p) = \begin{cases} f(task_x) = Avg\left(\frac{1}{2} + \frac{1}{1+0+0}\right) \times \frac{1}{1+0} = \frac{3}{4} \times \frac{1}{1} = \frac{3}{4} \\ f(task_y) = Avg\left(\frac{1}{3} + \frac{1}{1+1+0}\right) \times \frac{1}{1+0} = \frac{5}{12} \times \frac{1}{1} = \frac{5}{12} \\ f(task_z) = Avg\left(\frac{1}{2} + \frac{1}{1+0+1}\right) \times \frac{1}{1+0} = \frac{1}{2} \times \frac{1}{1} = \frac{1}{2} \end{cases}$$

$$\rightarrow f(S_i) = Avg\left(f(S_{task_i})\right) = \frac{5}{9}$$

4.3.3 Διαθεσιμότητα υπαλλήλων

Ανάμεσα στις υποθέσεις που διατυπώθηκαν στην αρχή του κεφαλαίου, συμπεριλαμβανόταν ότι κανένας υπάλληλος δεν μπορεί να εργάζεται σε περισσότερες από μια διεργασίες την ίδια χρονική στιγμή. Κάτι τέτοιο εξυπακούεται αφού είναι πολύπλοκο και συχνά ζημιογόνο χρονικά να απασχολείται κάποιος υπάλληλος σε δύο ή περισσότερες εργασίες ταυτόχρονα. Συνεπώς, είναι κατανοητό ότι για να θεωρείται το χρονοπρόγραμμα εφικτό και υλοποιήσιμο πρέπει να εκπληρώνει τη συγκεκριμένη απαίτηση.

Βασικό στοιχείο λοιπόν αποτελεί η διαθεσιμότητα των υπαλλήλων για την ανάθεσή τους σε ομάδες και την εκτέλεση μιας διεργασίας. Λαμβάνοντας υπόψη τα προαναφερθέντα, η διαθεσιμότητα ενός υπαλλήλου καθ' όλη τη διάρκεια του έργου συμμετέχει σε συνάρτηση καταλληλότητας ως εξής:

$$f(employee_e) = 1 - \frac{\#conflictDays}{\#workingDays} \quad (20)$$

όπου $employee_e$: ο e-οστός υπάλληλος που εξετάζεται

$\#conflictDays$: ο αριθμός των ημερών που ο συγκεκριμένος υπάλληλος έχει ανατεθεί να εργαστεί σε περισσότερες από μια διεργασίες

$\#workingDays$: ο αριθμός των εργάσιμων ημερών του συγκεκριμένου υπαλλήλου

Στη συνέχεια προσθέτονται οι αξιολογήσεις όλων των υπαλλήλων και υπολογίζεται ο σταθμισμένος μέσος όρος, ο οποίος αποτελεί την τελική αξιολόγηση του σωματιδίου όσον αφορά τη διαθεσιμότητα των υπαλλήλων για το συγκεκριμένο σωματίδιο.

Παρακάτω παρουσιάζεται ένα απλό παράδειγμα με τους απαραίτητους υπολογισμούς για την αξιολόγηση της διαθεσιμότητας των υπαλλήλων του p-οστού σωματιδίου του σμήνους.

Σημ.: $Hm.d = m \rightarrow$ ο υπάλληλος έχει ανατεθεί σε m διεργασίες την μέρα d.

Το μήκος του πίνακα είναι η συνολική διάρκεια εκτέλεσης του έργου, $\max(d)$.

Πίνακας 4.4. – Παράδειγμα υπαλλήλων για αξιολόγηση της διαθεσιμότητας

Υπάλληλος	Ημ.1	Ημ.2	Ημ.3	Ημ.4	Ημ.5	Ημ.6	Ημ.7	Ημ.8
X	0	0	1	1	1	0	0	0
Y	1	1	2	2	2	2	0	0
Z	2	2	3	3	3	3	4	4

$$f(A_p) = \begin{cases} f(employee_x) = 1 - \frac{0}{3} = \frac{1}{1} \\ f(employee_y) = 1 - \frac{4}{6} = \frac{1}{3} \\ f(employee_z) = 1 - \frac{8}{8} = 0 \end{cases} \rightarrow f(A_p) = Avg(f(employee_e)) = \frac{4}{9}$$

4.4 Απεμπλοκή από στασιμότητα

Όπως έχει αναφερθεί στο Κεφάλαιο 3.9, ένα από τα κυριότερα προβλήματα που παρουσιάζει ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων και οι εξελικτικοί αλγορίθμοι γενικότερα, είναι η στασιμότητα του σμήνους και η σύγκλιση του σε τοπικά βέλτιστες λύσεις. Το συγκεκριμένο μειονέκτημα έχει καταστήσει πλέον αναγκαία την εισαγωγή τεχνικών απεμπλοκής στη γενικότερη εκδοχή του αλγορίθμου, ώστε το σμήνος να μπορεί να ξεφεύγει από τη στασιμότητα. Έχουν εισηγηθεί κατά καιρούς διάφορες μεθόδους μέσω των οποίων επιχειρείται η απεμπλοκή του σμήνους από τα τοπικά ελάχιστα με σκοπό την συνέχιση της εξερεύνησης του. Αρκετές μάλιστα, το έχουν επιτύχει σημειώνοντας αξιοσημείωτα αποτελέσματα.

Η τεχνική που υλοποιείται σε αυτή τη διπλωματική εργασία ονομάζεται <<μερική επαναφορά>> και όπως προδίδει η ονομασία της γίνεται επαναφορά της ταχύτητας ορισμένων σωματιδίων. Συγκεκριμένα, αν κατά τη διάρκεια της επαναληπτικής διαδικασίας του αλγορίθμου παρουσιαστεί στασιμότητα στο σμήνος για προκαθορισμένο αριθμό επαναλήψεων, τότε μέσω αυτής της τεχνικής επιλέγεται τυχαία ένας αριθμός σωματιδίων από το σμήνος και αρχικοποιείται ξανά η ταχύτητα τους. Επιπλέον, όταν

χρησιμοποιείται η σφαιρική εκδοχή επαναρχικοποιείται η ταχύτητα του καλύτερου σωματιδίου του σμήνους, ενώ όταν χρησιμοποιείται η τοπική εκδοχή του αλγορίθμου γίνεται επαναρχικοποίηση της ταχύτητας όλων των σωματιδίων που ανήκουν στη γειτονιά του καλύτερου σφαιρικά σωματιδίου. Έτσι αν ένα σωματίδιο είναι στάσιμο σε κάποιο σημείο και δεν διαθέτει την απαραίτητη ταχύτητα για να ξεφύγει από το σημείο αυτό, είναι δυνατό με την επαναρχικοποίηση της ταχύτητας του να κατορθώσει να ξεφύγει και να συνεχίσει την εξερεύνηση.

Πρόκειται για μια σχετικά απλή τεχνική που όμως βασίζεται στην παρατήρηση ότι ο κυριότερος λόγος που παραμένει στάσιμο ένα σμήνος σε κάποιο σημείο, είναι η αδυναμία του καλύτερου του σωματιδίου να ξεφύγει από το τοπικό ελάχιστο. Αν στις επόμενες επαναλήψεις δεν βρεθεί καλύτερη λύση από κάποιο άλλο σωματίδιο του σμήνους τότε το συγκεκριμένο σωματίδιο συμπαρασέρνει όλα τα υπόλοιπα σωματίδια στο σημείο αυτό. Επακόλουθο είναι η σταδιακή και συνεχής μείωση της ταχύτητας των σωματιδίων και ο αναπόφευκτος μηδενισμός της που οδηγεί τελικά στη στασιμότητα του σμήνους.

4.5 Υλοποίηση πρωτοτύπου

Έχοντας καθορίσει τον τρόπο μοντελοποίησης των σωματιδίων του σμήνους ώστε να αναπαριστούν τα δεδομένα του προβλήματος καθώς επίσης και τη μεθοδολογία που θα ακολουθείτω για την αξιολόγηση του κάθε σωματιδίου, επόμενο στάδιο ήταν η υλοποίηση του πρωτοτύπου. Μετά από ενδελεχή έρευνα όσον αφορά τα διαθέσιμα εργαλεία ανάπτυξης λογισμικού αποφασίστηκε η χρήση του Microsoft Visual Studio 2008 χρησιμοποιώντας τη C# σαν γλώσσα προγραμματισμού, αφού είναι ένα από τα πιο ολοκληρωμένα εργαλεία ανάπτυξης λογισμικού με εξαιρετικό προγραμματιστικό περιβάλλον. Πρόκειται ουσιαστικά για μια εφαρμογή περιβάλλοντος windows όπου αρκεί να τρέξει το εκτελέσιμο αρχείο ώστε να ξεκινήσει η εκτέλεση της. Αν και η διεπιφάνεια χρήστη είναι σχετικά απλή, εντούτοις η ευρεία απήχηση που έχει το εργαλείο ανάπτυξης σε συνδυασμό με τις τεράστιες δυνατότητες του, υποβοηθούν οποιαδήποτε μελλοντική επέκταση της σε ένα πλήρες σύστημα υποστήριξης αποφάσεων. Στη συνέχεια

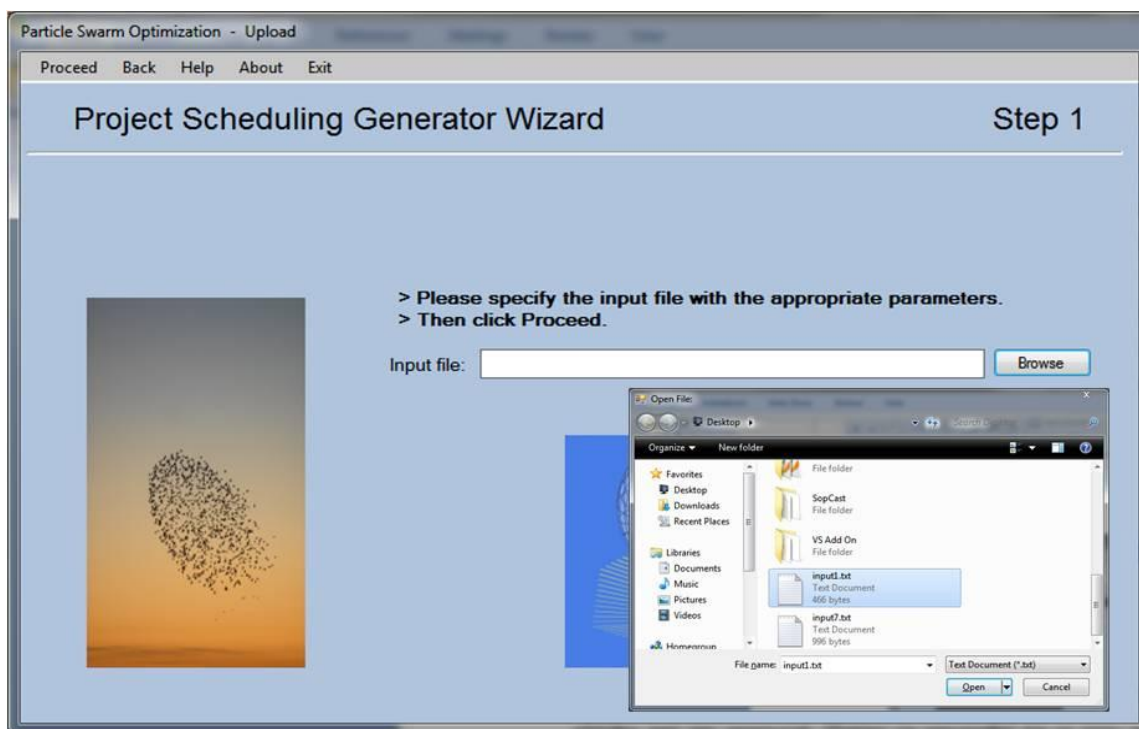
παρουσιάζεται η διαδικασία που πρέπει να ακολουθηθεί ώστε να εκτελεστεί ο αλγόριθμος και να εξαχθεί το τελικό αποτέλεσμα.

Αρχικά εμφανίζεται στο χρήστη η εισαγωγική οθόνη της διεπιφάνειας (Σχήμα 4.1.) παρουσιάζοντας ορισμένες πληροφορίες σχετικά με την εφαρμογή ενώ παράλληλα ενημερώνεται ότι πρέπει να επιλέξει την εντολή <<Start>> ώστε να συνεχίσει την εκτέλεση της. Επιπλέον, ο χρήστης χρησιμοποιώντας το διαθέσιμο μενού μπορεί επιλέγοντας την εντολή <<Help>> να πληροφορηθεί για τη διαδικασία που θα ακολουθηθεί ενώ με την εντολή <<About>> μπορεί να δει ορισμένες περαιτέρω λεπτομέρειες σχετικά με την εφαρμογή. Αντίστοιχα με την εντολή <<Exit>> μπορεί να εξέλθει από την εφαρμογή. Πρέπει να σημειωθεί ότι οι τρεις προαναφερθείσες εντολές υπάρχουν στο μενού σε όλες τις οθόνες της εφαρμογής ώστε ο χρήστης να μπορεί ανά πάσα στιγμή να προβεί στην επιθυμητή γι' αυτόν ενέργεια.



Σχήμα 4.1. – Η αρχική οθόνη της εφαρμογής

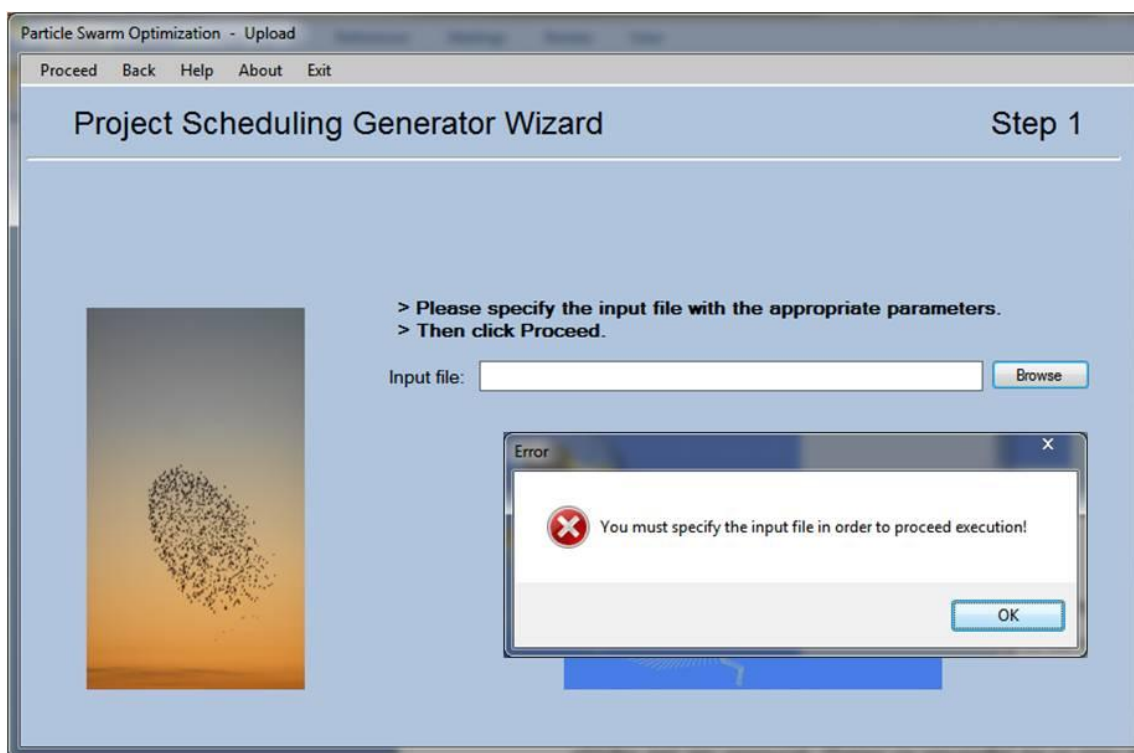
Έπειτα εμφανίζεται η επόμενη οθόνη του συστήματος όπου ο χρήστης καλείται να καθορίσει το αρχείο που περιέχει τα δεδομένα εισόδου πατώντας το κουμπί <<Browse>> (Σχήμα 4.2.) και μετά να επιλέξει την εντολή <<Proceed>>. Η εφαρμογή αναλύοντας τα δεδομένα του αρχείου δημιουργεί τις δομές δεδομένων που είναι απαραίτητες στα μετέπειτα στάδια. Αν κατ' εξαίρεση δεν καθοριστεί το αρχείο εισόδου και ο χρήστης επιλέξει την εντολή <<Proceed>> εμφανίζεται μήνυμα λάθους που τον καλεί να καθορίσει το αρχείο για να μπορεί να συνεχίσει η εκτέλεση της εφαρμογής (Σχήμα 4.3.).



Σχήμα 4.2. – Η οθόνη καθορισμού του αρχείου δεδομένων

Στη συνέχεια ο χρήστης μπορεί να επισκοπήσει τα δεδομένα που έχουν εισαχθεί στο σύστημα (Σχήμα 4.4.) ώστε να εγγυηθεί η ορθότητα τους για την ομαλή εκτέλεση του αλγορίθμου. Αν όμως έχει επιλεγεί λανθασμένο αρχείο τότε ο χρήστης μέσω της εντολής <<Back>> μπορεί να επιστρέψει στην προηγούμενη οθόνη και να καθορίσει κάποιο εναλλακτικό αρχείο. Έπειτα επιλέγοντας την εντολή <<Run>> ξεκινά η εκτέλεση του αλγορίθμου. Πρέπει να σημειωθεί ότι αμέσως μετά την επιλογή της προαναφερθείσας εντολής εμφανίζεται κατάλληλο μήνυμα που ειδοποιεί το χρήστη να περιμένει μέχρι να

τελειώσει η εκτέλεση του αλγορίθμου. Κατά τη διάρκεια της εκτέλεσης³ ο χρήστης δεν μπορεί να εκτελέσει καμιά άλλη διεργασία γιατί οποιαδήποτε άλλη λειτουργία επηρεάζει την εκτέλεση προκαλώντας ανεπιθύμητα αποτελέσματα.



Σχήμα 4.3. – Το μήνυμα λάθους σε περίπτωση μη καθορισμού του αρχείου εισόδου

Ακολούθως παρουσιάζονται τα εξαχθέντα, από την εκτέλεση του αλγορίθμου, αποτελέσματα στην αντίστοιχη οθόνη (Σχήμα 4.5.). Ο χρήστης μπορεί να πλοηγηθεί στη συγκεκριμένη οθόνη, να επισκοπήσει τα αποτελέσματα και στη συνέχεια επιλέγοντας την εντολή <<Finish>> να τερματίσει την εκτέλεση της εφαρμογής. Σε περίπτωση όμως που δεν είναι ικανοποιημένος από τα αποτελέσματα του αλγορίθμου, μπορεί επιλέγοντας την εντολή <<Run again>> να εκτελέσει ξανά την όλη διαδικασία προσδοκώντας σε καλύτερα αποτελέσματα.

³ Ο απαιτούμενος χρόνος εκτέλεσης αυξάνεται εκθετικά σε σχέση με το μέγεθος του προβλήματος αφού όπως έχει εξηγηθεί στο Κεφάλαιο 4.2 κάθε επιπλέον διεργασία ισοδυναμεί με μια επιπλέον διάσταση στο χώρο αναζήτησης.

Particle Swarm Optimization - Details

Run Back Help About Exit

Project Scheduling Generator Wizard

Steps 2 - 3

Project Details:

Tasks

Task Name	Duration	Dependencies	Required Skills
T1	10	T0	S1 S3
T2	15	T0	S3 S4
T3	20	T1	S5
T4	10	T1	S1 S3
T5	15	T2 T3	S2 S3 S4

* 1

Employees

Employee Name	Skills
E1	S1 S2
E2	S2 S3
E3	S4 S5
E4	S3 S5
E5	S1 S3

* 1

Program running

Please wait until the execution finishes

OK

Σχήμα 4.4 – Η οθόνη επισκόπησης των δεδομένων εισόδου

Particle Swarm Optimization - Results

Finish Run again Back Help About Exit

Project Scheduling Generator Wizard

Results

Project Results:

Task Name	Start Date	Employees Assigned
T1	1	2 3
T2	63	5
T3	11	2 5
T4	48	3
T5	59	2 4 5
T6	1	4
T7	50	1 4 5

* 1

Duration: 83 Evaluation: 71%

Σχήμα 4.5. – Η οθόνη αποτελεσμάτων

Στις προηγούμενες παραγράφους έχουν αναλυθεί εκτενώς τα βήματα που ακολουθούνται ώστε η εφαρμογή να εκτελέσει τις απαιτούμενες ενέργειες και υπολογισμούς για τη δημιουργία του χρονοπρογράμματος ενός έργου λογισμικού. Όπως γίνεται αντιληπτό πρόκειται για μια απλή διεπιφάνεια χρήστη που παρέχει απλώς τις απαραίτητες οθόνες για την εκτέλεση του αλγορίθμου. Πέρα όμως από τον κώδικα που έχει γραφτεί για την υλοποίηση της διεπιφάνειας το κυρίως μέρος της διπλωματικής εργασίας έγκειται στον κώδικα που έχει υλοποιηθεί για την εκτέλεση του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων.

Επόμενο στάδιο ήταν η πραγματοποίηση πειραμάτων ώστε να αξιολογηθεί ο αλγόριθμος όσον αφορά την αποτελεσματικότητα και την αξιοπιστία του. Για το σκοπό αυτό πραγματοποιήθηκαν δύο ειδών πειράματα, τα ελεγχόμενα και τα μη ελεγχόμενα. Στο πρώτο είδος πειραμάτων η βέλτιστη λύση ήταν γνωστή εκ των προτέρων, ενώ στο δεύτερο είδος η βέλτιστη λύση ήταν άγνωστη. Λεπτομερής ανάλυση των πειραμάτων και εξαγωγή συγκριτικών δεδομένων παρουσιάζεται στο κεφάλαιο που ακολουθεί.

Κεφάλαιο 5

Πειραματική αξιολόγηση

5.1 Εισαγωγή στη μεθοδολογία αξιολόγησης

5.2 Ελεγχόμενα πειράματα

5.2.1 Σενάριο ελεγχόμενων πειραμάτων

5.2.2 Αξιολόγηση κάθε επιμέρους συνάρτησης

5.2.3 Συνολική αξιολόγηση χωρίς τεχνική απεμπλοκής

5.2.4 Συνολική αξιολόγηση με τεχνική απεμπλοκής

5.2.5 Δυναμική συνάρτηση αξιολόγησης

5.3 Μη ελεγχόμενα πειράματα

5.3.1 Σενάριο μη ελεγχόμενων πειραμάτων

5.3.2 Αξιολόγηση αλγορίθμων

5.4 Σύνοψη

5.1 Εισαγωγή στη μεθοδολογία αξιολόγησης

Είναι γνωστό στην ερευνητική κοινότητα ότι η λεκτική διατύπωση ενός αλγορίθμου μπορεί να φαίνεται ορθή και να ικανοποιεί όλα τα απαιτούμενα κριτήρια αλλά αυτό από μόνο του δεν αποτελεί εχέγγυο για αποτελεσματική εφαρμογή του στο πρόβλημα που προσεγγίζει. Απαραίτητο συστατικό στοιχείο είναι η πρακτική απόδειξη της ορθότητας, της αποτελεσματικότητας και κυρίως της ικανότητας του να βρίσκει λύσεις στο πρόβλημα με το οποίο καταπιάνεται. Συνεπώς απαιτείται η χρήση ορισμένων παραδειγμάτων και

σημείων αναφοράς για να ελεγχθεί πειραματικά και κατ' επέκταση να συγκριθεί με ήδη υπάρχοντες αλγορίθμους που εφαρμόζονται στην περιοχή. Για το σκοπό αυτό έχει αποφασισθεί η πραγματοποίηση πειραματικών διαδικασιών σε σενάρια με πραγματικά δεδομένα ώστε να αξιολογηθεί η ποιότητα και η ορθότητα των εξαγόμενων αποτελεσμάτων.

Στο σημείο αυτό παρουσιάστηκε ιδιαίτερη δυσκολία στην εξεύρεση σεναρίων που θα χρησιμοποιούνταν στα πειράματα αφού δεν βρέθηκαν αποθήκες δεδομένων (repositories) στο διαδίκτυο που να παρέχουν τα απαραίτητα δεδομένα. Αντίθετα, οι πλείστες αποθήκες που είχαν βρεθεί και ασχολούνταν με το RCPSP, δεν αφορούσαν τη δημιουργία ομάδων εργασίας αλλά την ανάθεση συγκεκριμένων πόρων σε διεργασίες. Επακόλουθο ήταν η αξιοποίηση σεναρίων, που είχαν χρησιμοποιηθεί σε επιστημονικά άρθρα, τα οποία ωστόσο δεν ανέφεραν τη βέλτιστη λύση του σεναρίου. Επομένως, έπρεπε να εξευρεθεί το βέλτιστο αποτέλεσμα του σεναρίου προτού εφαρμοστεί πειραματικά και γι' αυτό το πλήθος των σεναρίων είναι περιορισμένο.

Η συνολική πειραματική διαδικασία έχει χωριστεί σε δύο επιμέρους κατηγορίες, στα ελεγχόμενα και στα μη ελεγχόμενα πειράματα. Η πρώτη κατηγορία περιλαμβάνει τα πειράματα όπου το βέλτιστο αποτέλεσμα είναι γνωστό εκ των προτέρων και χρησιμοποιείται κυρίως για να ελεγχθεί σε ποιο ποσοστό μπορεί να φτάσει ο εξεταζόμενος αλγόριθμος στη βέλτιστη λύση. Αντίθετα, στη δεύτερη πειραματική κατηγορία τα αποτελέσματα δεν είναι γνωστά και αφήνεται ο αλγόριθμος μέσω της εξελικτικής του διαδικασίας να δημιουργήσει το αποτέλεσμα. Στη συνέχεια το εξαγόμενο χρονοπρόγραμμα ελέγχεται για την ορθότητα του και κατά πόσο έχει επιτευχθεί η παραγωγή ενός εφικτού χρονοπρογράμματος.

5.2 Ελεγχόμενα πειράματα

Βασικός στόχος στα πειράματα αυτά, όπως έχει αναφερθεί παραπάνω, ήταν η αξιολόγηση του αλγορίθμου όσον αφορά την ικανότητα να φτάσει στη βέλτιστη λύση. Όμως, επειδή ποικίλλουν οι παραλλαγές του αλγορίθμου που μπορούν να χρησιμοποιηθούν, κρίθηκε

απαραίτητη η αξιολόγηση των βασικότερων του παραλλαγών με απώτερο σκοπό την εξεύρεση των πιο αποδοτικών που θα χρησιμοποιούνταν στο επόμενο είδος πειραμάτων. Συνεπώς, στα εν λόγω πειράματα χρησιμοποιούνται οι πιο κάτω παραλλαγές του βασικού αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων:

- i. PSO με γραμμική μείωση του βάρους αδρανείας (PSO with linear decreasing inertia).
- ii. PSO με τυχαίο βάρος αδρανείας στο πεδίο $U(0.5, 1)$ (PSO with random inertia)⁴.
- iii. PSO με γραμμική προσαρμογή της τιμής των συντελεστών επιτάχυνσης (PSO with linear adjusting acceleration coefficients)⁵.
- iv. Περιοριστικός PSO (Constriction PSO).

Όσον αφορά την τοπολογία του σμήνους και τον τρόπο διάδοσης των πληροφοριών μεταξύ των σωματιδίων, έχει αποφασισθεί η χρησιμοποίηση των τοπολογιών αστέρα (GB) και δακτυλίου (LB). Στο πρώτο είδος τοπολογίας ολόκληρο το σμήνος αποτελεί μια γειτονιά ενώ στο δεύτερο υπάρχουν γειτονιές μεγέθους δύο, εννοώντας ότι το σωματίδιο επηρεάζεται από τα σωματίδια που βρίσκονται δεξιά και αριστερά του. Ο λόγος που υλοποιούνται δύο είδη τοπολογιών είναι για γίνει κατορθωτή η αξιολόγηση του ίδιου αλγορίθμου για διαφορετικό τρόπο διασύνδεσης μεταξύ των σωματιδίων.

Πρέπει να σημειωθεί ότι κάθε παραλλαγή του αλγορίθμου αξιολογήθηκε 100 φορές με σμήνος που αριθμούσε 100 σωματίδια ενώ ο αλγόριθμος εκτελείται για 10,000 και 15,000 επαναλήψεις για τις τοπολογίες αστέρα και δακτυλίου αντίστοιχα. Μπορεί να θεωρείται αρκετά μεγάλο το πλήθος των σωματιδίων στο σμήνος, σε σχέση πάντοτε με το προτεινόμενο πλήθος των 50 σωματιδίων, αλλά αυτό αποσκοπεί στην όσο το δυνατό καλύτερη κάλυψη του χώρου αναζήτησης. Αντίστοιχα, ο λόγος για τον οποίο επιλέχθηκε η εκτέλεση της εκάστοτε παραλλαγής του αλγορίθμου με τόσες επαναλήψεις είναι για να καταστεί εφικτή η παρατήρηση της συμπεριφοράς του για μεγάλο χρονικό διάστημα και η μετέπειτα εξαγωγή χρήσιμων συμπερασμάτων.

⁴ Χρησιμοποιείται η μέθοδος που εισηγήθηκαν οι Shi και Eberhart [38].

⁵ Χρησιμοποιείται η μέθοδος που εισηγήθηκε ο Ratnaweera κ.α. [71].

Επιπλέον έγινε προσπάθεια να δημιουργηθεί μια εποπτική εικόνα όσον αφορά την αντίδραση του αλγορίθμου στην σταδιακή αύξηση της ποσότητας και της πολυπλοκότητας των δεδομένων που επεξεργάζεται. Αυτό πραγματοποιήθηκε χωρίζοντας την γενικότερη διαδικασία σε τρία στάδια. Στο πρώτο στάδιο αξιολογήθηκε η κάθε συνάρτηση αξιολόγησης ξεχωριστά, στο δεύτερο στάδιο έγινε αξιολόγηση όλων των επιμέρους συναρτήσεων χωρίς αξιοποίηση της τεχνικής απεμπλοκής ενώ στο τελευταίο στάδιο εισήχθη η εν λόγω τεχνική ώστε να παρατηρηθεί η διαφορά στην συμπεριφορά του αλγορίθμου.

5.2.1 Σενάριο ελεγχόμενων πειραμάτων

Στο συγκεκριμένο είδος πειραμάτων χρησιμοποιήθηκε ένα σχετικά απλό σενάριο επτά διεργασιών με πέντε υπαλλήλους. Τα ονομαστικά στοιχεία των υπαλλήλων καθώς επίσης και οι ικανότητες που διαθέτει ο κάθε υπάλληλος παρατίθενται στον Πίνακα 5.1. Αντίστοιχα, τα στοιχεία που αφορούν τις διεργασίες του έργου εννοώντας τις αλληλεξαρτήσεις μεταξύ των διεργασιών, τη διάρκεια εκτέλεσής τους και τις απαιτούμενες ικανότητες για την εκτέλεση της κάθε διεργασίας παρατίθενται στον Πίνακα 5.2. Επίσης τα βέλτιστα αποτελέσματα του εν λόγω σεναρίου παρουσιάζονται στον Πίνακα 5.3. και οι βέλτιστες τιμές για κάθε συνάρτηση αξιολόγηση παρουσιάζονται στον Πίνακα 5.4.

Πίνακας 5.1 – Στοιχεία υπαλλήλων.

Όνομα	Διαθέσιμες ικανότητες	
E1	S1	S2
E2	S2	S3
E3	S4	S5
E4	S3	S5
E5	S1	S3

Πίνακας 5.2 – Στοιχεία έργου και επιμέρους διεργασιών ⁶

Όνομα	Διάρκεια	Εξαρτήσεις		Απαιτούμενες ικανότητες		
T1	10	T0		S2	S4	S5
T2	10	T1		S1	S5	
T3	5	T1		S2		
T4	35	T3		S3	S5	
T5	15	T2		S1	S2	S3
T6	20	T0		S3		
T7	10	T4	T5	S2	S5	

Πίνακας 5.3 – Βέλτιστα αποτελέσματα εξεταζόμενου σεναρίου

Όνομα διεργασίας	Ημέρα έναρξης	Ανατεθειμένοι Υπαλλήλοι	
T1	1	E1	E3
T2	11	E1	E4
T3	11	E2	
T4	16	E4	
T5	21	E2	E5
T6	1	E2	
T7	51	E2	E4

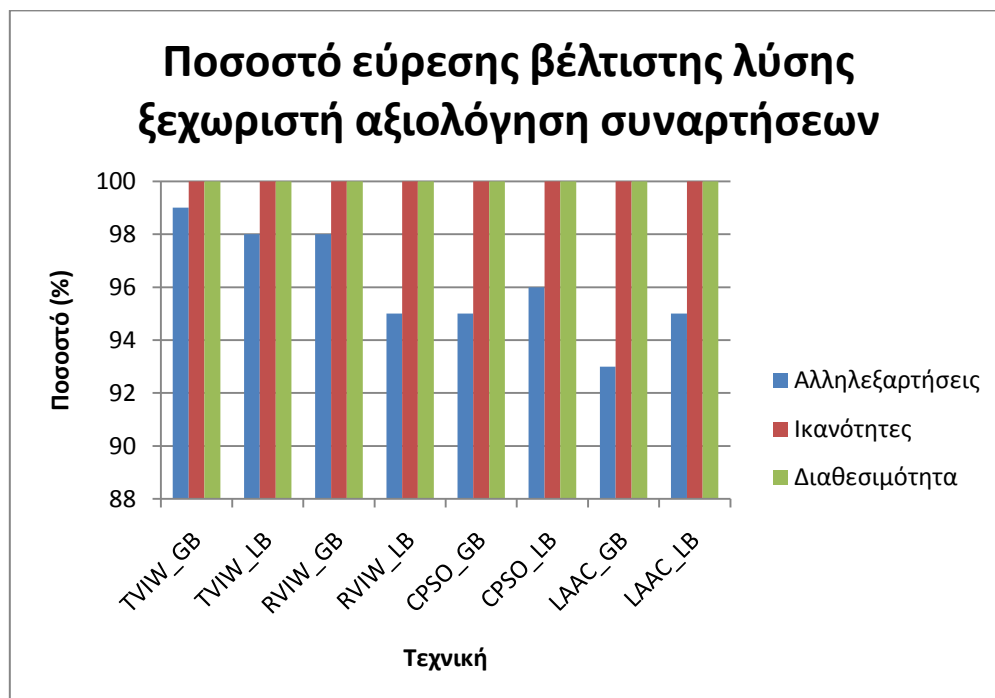
Πίνακας 5.4 – Βέλτιστα αποτελέσματα συναρτήσεων αξιολόγησης

Συνάρτηση	Βέλτιστο Αποτέλεσμα
Αλληλεξαρτήσεις διεργασιών – f(D)	1
Ικανοποίηση ικανοτήτων – f(S)	0.8929
Διαθεσιμότητα υπαλλήλων – f(A)	1
Συνολική συνάρτηση - R	0.9643

⁶ Οι διεργασίες T1 και T6 εξαρτώνται από την αρχική εικονική διεργασία (T0) του έργου.

5.2.2 Αξιολόγηση κάθε επιμέρους συνάρτησης ξεχωριστά

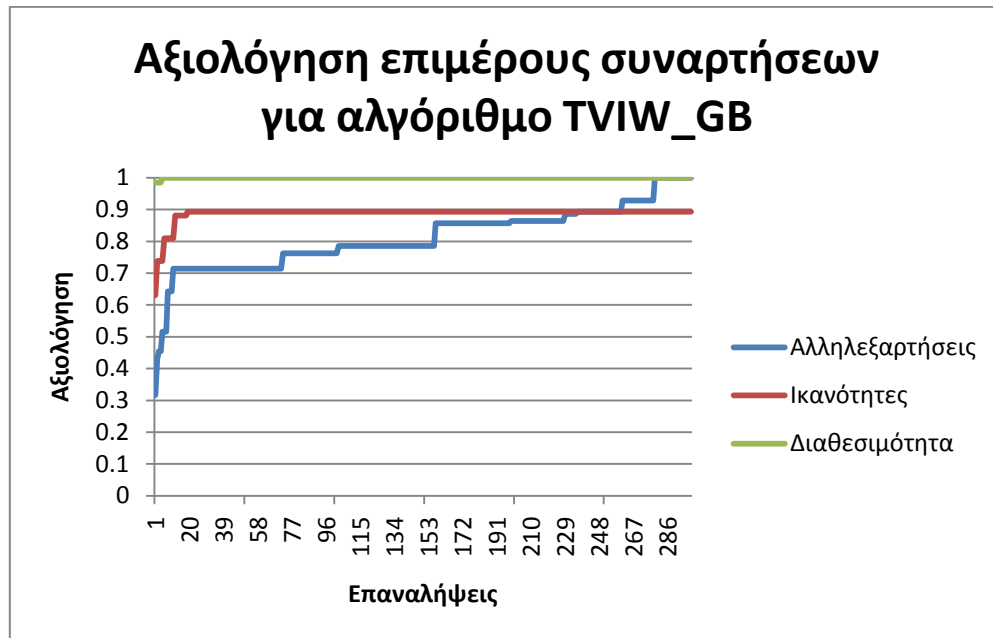
Στην περίπτωση αυτή οι εξεταζόμενοι αλγόριθμοι κατάφεραν να φτάσουν στη βέλτιστη λύση του προβλήματος πολύ γρήγορα, σε χρόνο λιγότερο των 500 επαναλήψεων. Το γεγονός ότι μόνο μια συνάρτηση αξιολογείται κάθε φορά δεν επιβάρυνε τους υπολογισμούς του προβλήματος με αποτέλεσμα να είναι πολύ εύκολο για το σμήνος να βρει τη βέλτιστη λύση. Επιπλέον στη γραφική παράσταση της συνάρτησης αξιολόγησης δεν υπήρχαν δύσκολα τοπικά ελάχιστα και ως εκ τούτου οι περιπτώσεις όπου το σμήνος παρέμενε στάσιμο ήταν ελάχιστες. Απόδειξη αποτελεί ότι το ποσοστό εύρεσης της βέλτιστης λύσης με αξιολόγηση μόνο μιας επιμέρους συνάρτησης (Σχήμα 5.1) είναι πάνω από 90% για όλους τους αλγόριθμους.



Σχήμα 5.1 – Ποσοστό εύρεσης βέλτιστης λύσης με αξιολόγηση μόνο μιας επιμέρους συνάρτησης

Το συγκεκριμένο πείραμα δεν κρίζει επιπλέον αναφοράς αφού τέτοιου είδους προβλήματα δεν παρουσιάζουν ιδιαίτερη σημασία μιας και είναι εύκολα στην επίλυσή τους. Άλλωστε δεν είναι ο σκοπός της διπλωματικής εργασίας και πραγματοποιήθηκε απλώς για να εξακριβωθεί η ορθή λειτουργία των αλγορίθμων. Όπως φαίνεται στο Σχήμα 5.2 για τον

αλγόριθμο PSO με γραμμική μείωση του βάρους αδρανείας με τοπολογία αστέρα (TVIW_GB), οι τιμές των συναρτήσεων αξιολόγησης αυξάνονται συνεχώς μέχρι που να φτάσουν στη μέγιστη τιμή τους.

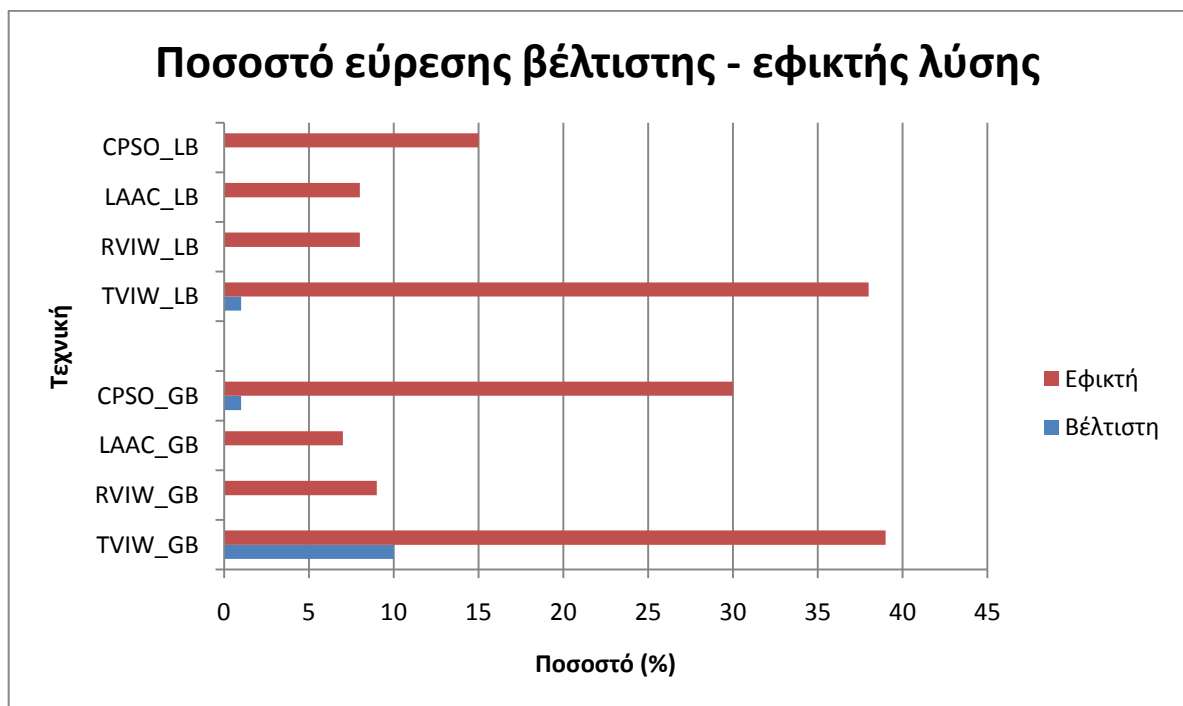


Σχήμα 5.2 – Αποτελέσματα αξιολόγησης επιμέρους συναρτήσεων για τον αλγόριθμο TVIW_GB

5.2.3 Συνολική αξιολόγηση χωρίς τεχνική απεμπλοκής

Επόμενο στάδιο ήταν η αξιολόγηση του σωματιδίου βάσει του σταθμισμένου μέσου όρου των επιμέρους συναρτήσεων (Εξίσωση 17) χωρίς όμως ενσωμάτωση της τεχνικής απεμπλοκής από τη στασιμότητα. Σκοπός της πειραματικής διαδικασίας αυτής ήταν η αξιολόγηση των αλγορίθμων όσον αφορά την ικανότητα να φτάνουν τόσο στη βέλτιστη λύση (Πίνακας 5.3) όσο και στη δημιουργία εφικτού χρονοπρογράμματος. Με την έννοια εφικτό χρονοπρόγραμμα εννοείται το χρονοπρόγραμμα στο οποίο δεν υπάρχει παραβίαση των περιορισμών προτεραιότητας μεταξύ των διεργασιών και επιπλέον οι ανατιθέμενοι υπάλληλοι σε κάθε διεργασία διαθέτουν τις απαραίτητες ικανότητες για την εκτέλεσή της. Βέβαια, δεν είναι το βέλτιστο αποτέλεσμα που μπορεί να προκύψει, αλλά αν τεθεί σε εφαρμογή μπορεί να οδηγήσει στην επιτυχημένη εκτέλεση του έργου.

Όπως φαίνεται στο Σχήμα 5.3, τα καλύτερα αποτελέσματα πετυχαίνει ο PSO με γραμμική μείωση του βάρους αδρανείας χρησιμοποιώντας την τοπολογία αστέρα (TVIW_GB) αφού φτάνει στη βέλτιστη λύση σε ποσοστό 10% ενώ παράγει εφικτό χρονοπρόγραμμα σε ποσοστό 39%. Αντίστοιχο ποσοστό εφικτού χρονοπρογράμματος επιτυγχάνει και ο PSO με γραμμική μείωση του βάρους αδρανείας χρησιμοποιώντας την τοπολογία δακτυλίου (TVIW_LB), αλλά μόλις μια φορά έχει καταφέρει να φτάσει στη βέλτιστη λύση. Επιπλέον, καλά αποτελέσματα έχει επιτύχει ο περιοριστικός PSO με τοπολογία αστέρα (CPSO_GB) αφού έχει δημιουργήσει 30% εφικτό χρονοπρόγραμμα.



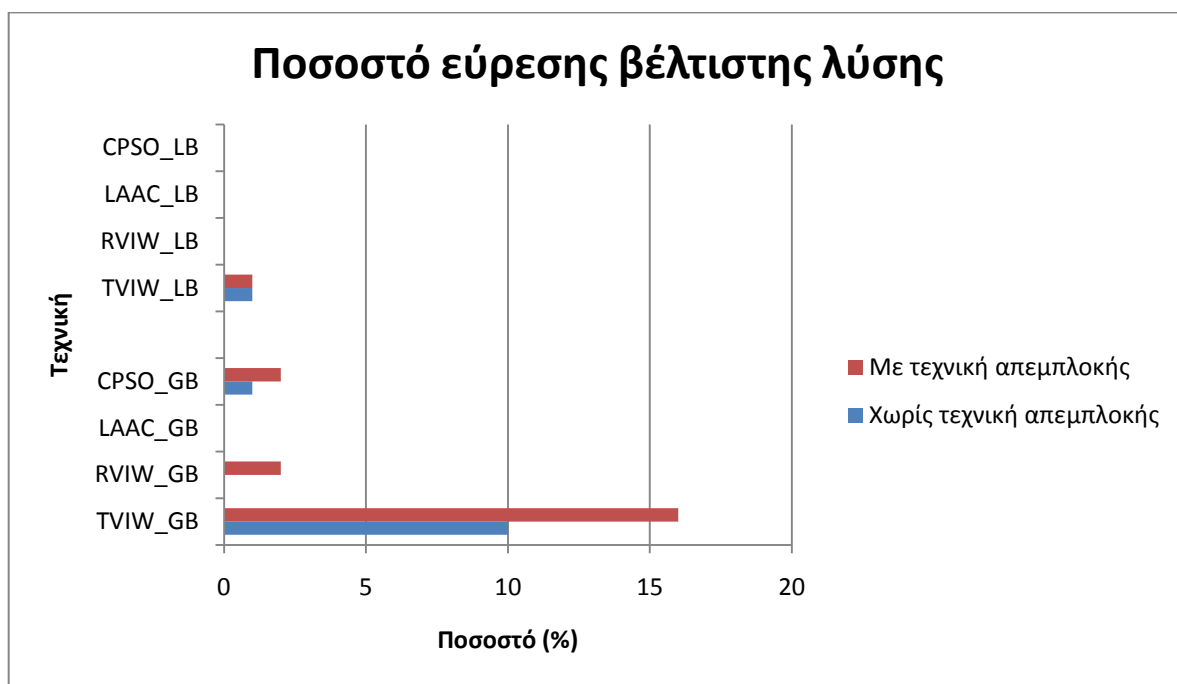
Σχήμα 5.3 – Αποτελέσματα εύρεσης βέλτιστης και εφικτής λύσης χωρίς την τεχνική απεμπλοκής

Όμως, το γεγονός ότι οι πέντε από τους οκτώ αλγορίθμους (περιοριστικός PSO με τοπολογία δακτυλίου – CPSO_LB, PSO με γραμμική αναπροσαρμογή των συντελεστών επιτάχυνσης και με τις δύο τοπολογίες – LAAC_LB, LAAC_GB, PSO με τυχαίο βάρος αδρανείας και με τις δύο τοπολογίες – RVIW_LB, RVIW_GB) δεν κατορθώνουν να φτάσουν ούτε μια φορά στη βέλτιστη λύση και ότι το ποσοστό παραγωγής εφικτού χρονοπρογράμματος είναι μικρότερο του 15% δημιουργεί ερωτήματα όσον αφορά την αποτελεσματικότητα των εν λόγω αλγορίθμων. Μια προσεκτική παρατήρηση αποδεικνύει

ότι το σμήνος παραμένει στάσιμο σε τοπικά ελάχιστα σημεία γεγονός που περιορίζει τις δυνατότητες για περαιτέρω εξερεύνηση του χώρου αναζήτησης προς μια καλύτερη θέση. Στην προσπάθεια επίλυσης του προβλήματος αυτού, στο επόμενο στάδιο των πειραμάτων αξιοποιείται η προτεινόμενη τεχνική απεμπλοκής από τη στασιμότητα.

5.2.4 Συνολική αξιολόγηση με τεχνική απεμπλοκής

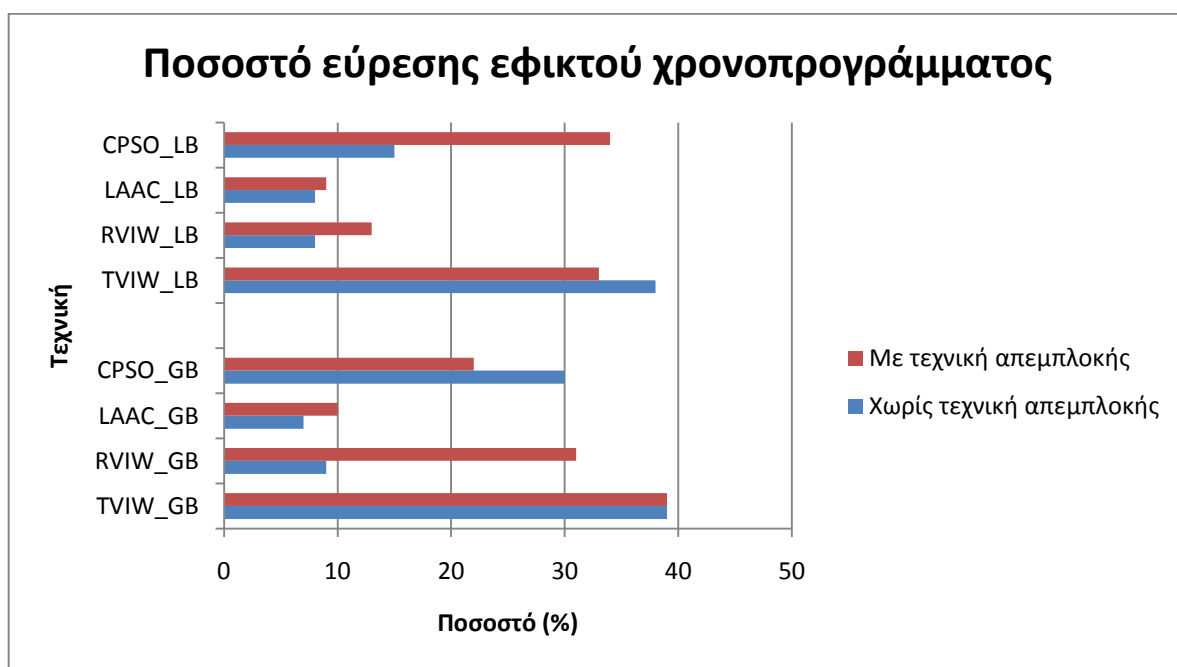
Σε αυτή τη φάση των πειραματικών διαδικασιών ενσωματώνεται στους εξεταζόμενους αλγορίθμους η τεχνική απεμπλοκής <<μερική επαναφορά>> (Κεφάλαιο 4.9), ώστε να παρατηρηθεί η συμπεριφορά του σμήνους και κατά πόσο κατορθώνει να επιτύχει καλύτερα αποτελέσματα από προηγούμενως.



Σχήμα 5.4 – Αποτελέσματα εύρεσης βέλτιστης λύσης χωρίς τεχνική απεμπλοκής και με χρήση της τεχνικής απεμπλοκής

Όπως φαίνεται από τα Σχήματα 5.4 και 5.5 στις πλείστες των περιπτώσεων τα αποτελέσματα εύρεσης βέλτιστης λύσης και παραγωγής εφικτού χρονοπρογράμματος αντίστοιχα, βελτιώνονται αισθητά αξιοποιώντας την τεχνική απεμπλοκής. Συγκεκριμένα

γίνεται κατορθωτό να φτάσει στη βέλτιστη λύση αλγόριθμος (RVIW_GB) που χωρίς τη χρήση της τεχνικής απεμπλοκής είχε μηδενικό ποσοστό εύρεσης της βέλτιστης λύσης. Επιπλέον τα ποσοστά παραγωγής εφικτού χρονοπρογράμματος αυξάνονται σχεδόν σε όλες τις παραλλαγές του αλγορίθμου και κυμαίνονται μεταξύ 20% έως 40% στους αποτελεσματικότερους αλγορίθμους. Το συγκεκριμένο εύρημα αποδεικνύει την απαραίτητη υλοποίηση τεχνικής απεμπλοκής ώστε το σμήνος να κατορθώνει να ξεφεύγει από τις τοπικά βέλτιστες θέσεις και να συνεχίζει την αναζήτηση του προς τη συνολικά βέλτιστη θέση.

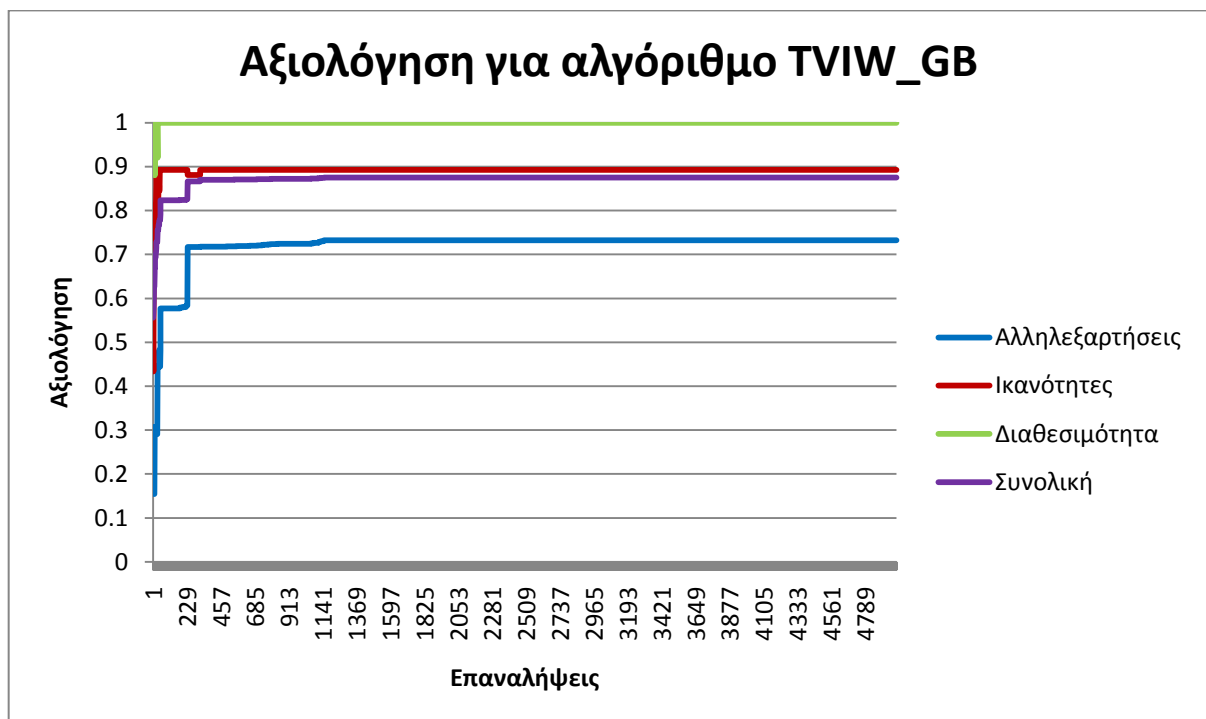


Σχήμα 5.5 – Αποτελέσματα παραγωγής εφικτού χρονοπρογράμματος χωρίς τεχνική απεμπλοκής και με χρήση της τεχνικής απεμπλοκής

5.2.5 Δυναμική συνάρτηση αξιολόγησης

Παρόλο που οι αλγόριθμοι κατόρθωσαν να φτάσουν στη βέλτιστη λύση ορισμένες φορές και μπορούσαν να δημιουργήσουν εφικτό χρονοπρόγραμμα σε σχετικά ικανοποιητικό ποσοστό, εντούτοις τα αποτελέσματα δεν ήταν αρεστά αφού αναμενόταν μεγαλύτερο ποσοστό επιτυχίας στην εύρεση εφικτού χρονοπρογράμματος. Γι' αυτό αναζητήθηκαν οι

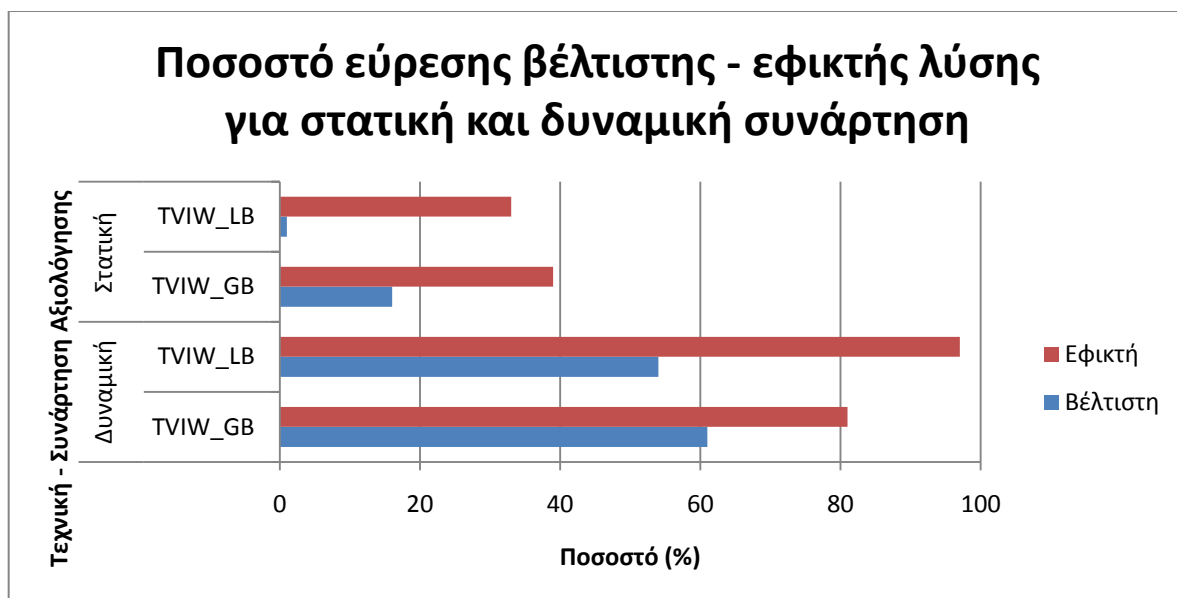
λόγοι που εμποδίζουν τον αλγόριθμο να παράγει εφικτό χρονοπρόγραμμα και όπως φαίνεται στο Σχήμα 5.6, η συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών είναι αυτή που <<περιορίζει>> την απόδοση του αλγορίθμου. Συγκεκριμένα, οι συναρτήσεις αξιολόγησης των ικανοτήτων και της διαθεσιμότητας των υπαλλήλων, φτάνουν στις μέγιστες τιμές τους αρκετά νωρίς στην εξελικτική διαδικασία και παραμένουν σταθερές καθ' όλη τη διάρκεια της διαδικασίας. Όμως η συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών αν και αυξάνεται στα αρχικά στάδια εκτέλεσης του αλγορίθμου, παραμένει σταθερή σε μια τιμή στη συνέχεια μη επιτρέποντας στον αλγόριθμο να συνεχίσει την εξελικτική του πορεία. Επακόλουθο είναι η αδυναμία του σμήνους να συνεχίσει την εξερεύνησή του στο χώρο αναζήτησης και επομένως η αποτυχία παραγωγής εφικτού χρονοπρογράμματος.



Σχήμα 5.6 – Αποτελέσματα αξιολόγησης για τον αλγόριθμο TVIW_GB

Στην προσπάθεια επίλυσης του πιο πάνω προβλήματος κρίθηκε συνετό το σμήνος για ορισμένο αριθμό επαναλήψεων να αξιολογείται μόνο βάσει της συνάρτησης αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών και μετά να ενσωματώνονται οι υπόλοιπες

συναρτήσεις αξιολόγησης. Το σκεπτικό πίσω από αυτή την ενέργεια έγκειται στο ότι η παρεμπόδιση στην παραγωγή εφικτού χρονοπρογράμματος οφείλεται κατά κύριο λόγο στις αλληλεξαρτήσεις μεταξύ των διεργασιών. Έτσι με αυτή την τεχνική θα μπορούν να δρομολογηθούν πρώτα οι διεργασίες διασφαλίζοντας την ικανοποίηση των περιορισμών μεταξύ τους και μετά να αξιολογηθούν τα σωματίδια βάσει όλων των περιορισμών. Πρέπει να σημειωθεί ότι για την πραγματοποίηση των πειραμάτων με τη δυναμική συνάρτηση αξιολόγησης επιλέχθηκε η χρήση των αλγορίθμων TVIW_GB και TVIW_LB οι οποίοι είχαν αποφέρει τα καλύτερα αποτελέσματα στις προηγούμενες πειραματικές διαδικασίες.



Σχήμα 5.7 – Σύγκριση ποσοστού εύρεσης βέλτιστης και εφικτής λύσης με στατική και δυναμική συνάρτηση αξιολόγησης

Τα αποτελέσματα από τα πειράματα που διενεργήθηκαν με αυτή την τεχνική παρουσιάζονται στο Σχήμα 5.7 (Δυναμική) και πράγματι έχει αυξηθεί σημαντικά τόσο το ποσοστό παραγωγής βέλτιστης λύσης όσο και το ποσοστό παραγωγής εφικτής λύσης. Αναλυτικά, η εύρεση της βέλτιστης λύσης από τους αλγορίθμους χρησιμοποιώντας τη δυναμική συνάρτηση αξιολόγησης κυμαίνεται μεταξύ 50% και 60% ενώ το αντίστοιχο ποσοστό παραγωγής εφικτού χρονοπρογράμματος υπερβαίνει το 80% και φθάνει έως το 97%. Τα ευρήματα αυτά αποδεικνύουν την υπεροχή της δυναμικής συνάρτησης σε σχέση

με τη στατική συνάρτηση αξιολόγησης, ενώ παράλληλα επιβεβαιώνεται η παρεμπόδιση που έχει στην εξελικτική διαδικασία του αλγορίθμου η συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών.

5.3 Μη ελεγχόμενα πειράματα

Επόμενο στάδιο στην πειραματική διαδικασία ήταν η πραγματοποίηση μη ελεγχόμενων πειραμάτων, δηλαδή η εκτέλεση σεναρίων στα οποία το βέλτιστο αποτέλεσμα δεν είναι γνωστό εκ των προτέρων. Σκοπός ήταν να ελεγχθεί η ικανότητα του αλγορίθμου για παραγωγή εφικτού χρονοπρογράμματος. Έτσι έγινε διαλογή ανάμεσα στους εξεταζόμενους αλγορίθμους και επιλέγηκαν αυτοί που είχαν τα καλύτερα αποτελέσματα στα ελεγχόμενα πειράματα, ο περιοριστικός PSO (CPSO) και ο PSO με γραμμική μείωση του βάρους αδρανείας (TVIW). Επιπλέον, οι δύο αυτοί αλγορίθμοι θα χρησιμοποιούνταν στη συγκεκριμένη φάση μαζί με τις δύο τοπολογίες που καθορίστηκαν εξ αρχής, τις τοπολογίες αστέρα (GB) και δακτυλίου (LB). Πρέπει να σημειωθεί ότι οι παραμέτροι εκτέλεσης κάθε αλγορίθμου παρέμειναν όπως και στα ελεγχόμενα πειράματα. Κάθε παραλλαγή του αλγορίθμου αξιολογήθηκε 100 φορές με σμήνος που αριθμούσε 100 σωματίδια ενώ ο αλγόριθμος εκτελείται για 10,000 και 15,000 επαναλήψεις για τις τοπολογίες αστέρα και δακτυλίου αντίστοιχα.

5.3.1 Σενάριο μη ελεγχόμενων πειραμάτων

Για την αξιολόγηση των αλγορίθμων χρησιμοποιήθηκε ένα πιο σύνθετο σενάριο με δέκα διεργασίες και έξι υπαλλήλους. Τα στοιχεία που αφορούν τις διεργασίες του έργου δηλαδή τις αλληλεξαρτήσεις μεταξύ των διεργασιών, τη διάρκεια εκτέλεσής τους και τις απαιτούμενες ικανότητες για την εκτέλεση της κάθε διεργασίας παρουσιάζονται στον Πίνακα 5.4. Αντίστοιχα, τα ονομαστικά στοιχεία των υπαλλήλων και οι ικανότητες που διαθέτει ο κάθε υπάλληλος παρατίθενται στον Πίνακα 5.5.

Πίνακας 5.4 – Στοιχεία έργου και επιμέρους διεργασιών ⁷

Όνομα	Διάρκεια	Εξαρτήσεις			Απαιτούμενες ικανότητες		
T1	10	T0			S1	S3	
T2	15	T0			S3	S4	
T3	20	T1			S5		
T4	10	T1			S1	S3	
T5	15	T2	T3		S2	S3	S4
T6	20	T2			S2	S3	
T7	10	T5	T6		S2	S4	
T8	15	T4	T7		S1	S5	
T9	20	T6			S3	S4	
T10	20	T7	T8	T9	S3	S5	

Πίνακας 5.5 – Στοιχεία υπαλλήλων

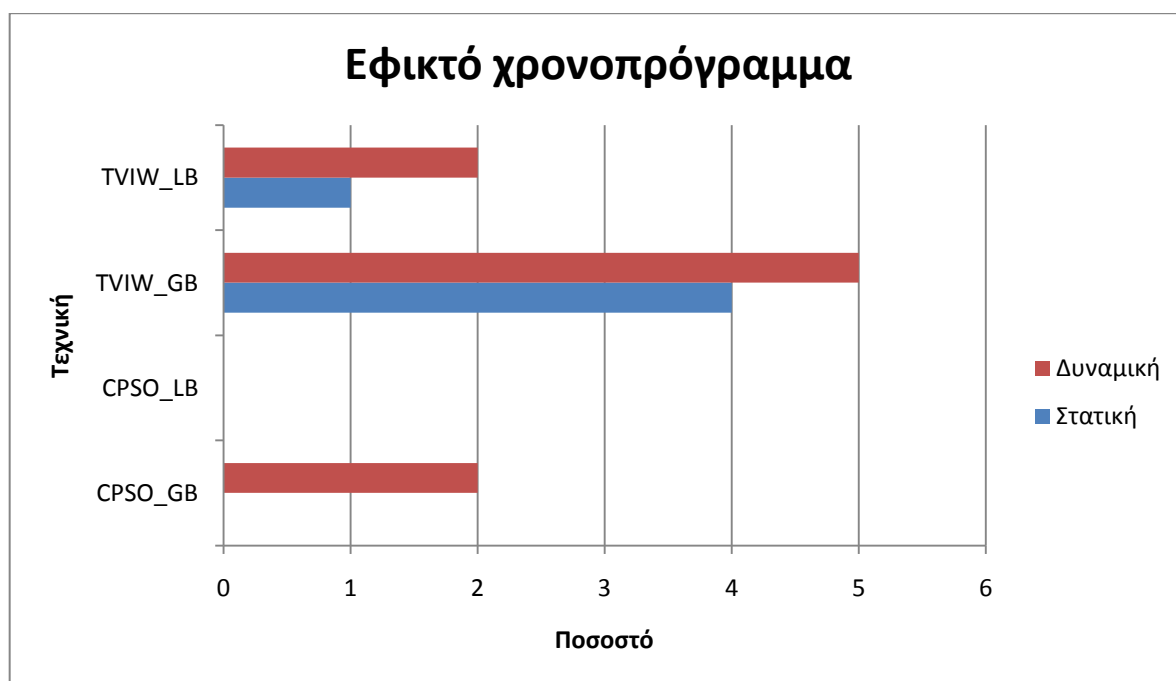
Όνομα	Διαθέσιμες ικανότητες	
E1	S1	S2
E2	S2	S3
E3	S4	S5
E4	S3	S5
E5	S1	S3
E6	S1	

5.3.2 Αξιολόγηση αλγορίθμων

Αρχικά οι αλγορίθμοι αξιολογήθηκαν βάσει της Εξίσωσης (17) από την αρχή εκτέλεσης της εξελικτικής διαδικασίας, δηλαδή με στατική συνάρτηση αξιολόγησης (Κεφάλαιο 5.2.4)

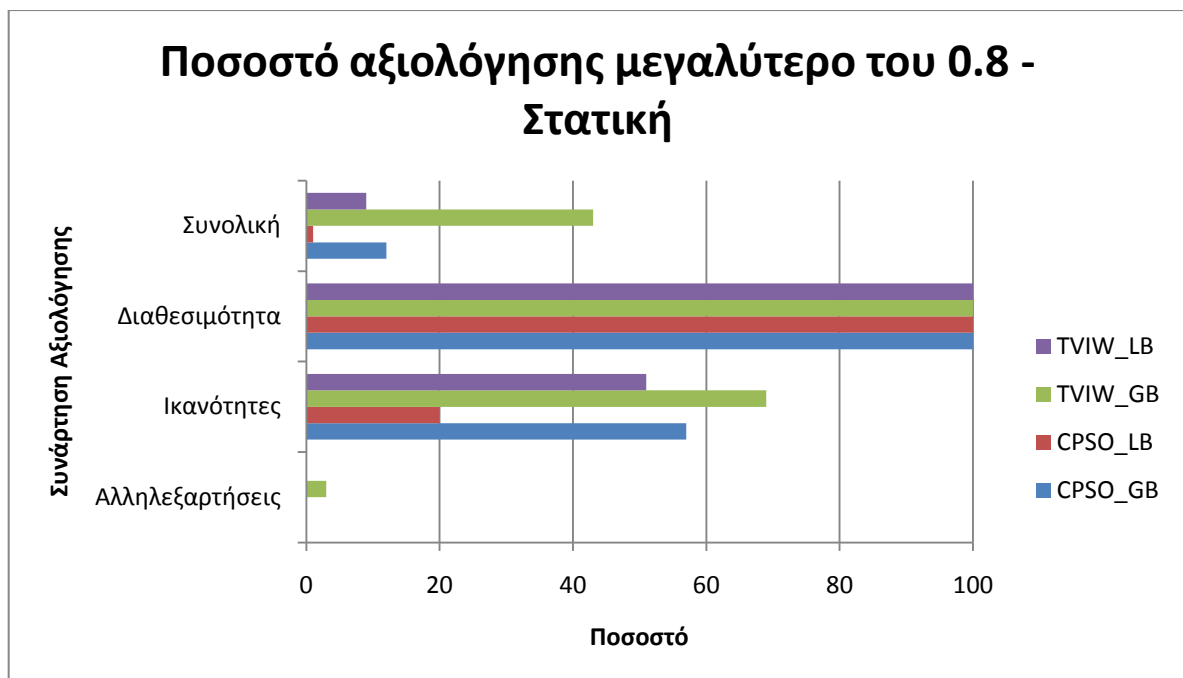
⁷ Οι διεργασίες T1 και T2 εξαρτώνται από την αρχική εικονική διεργασία (T0) του έργου.

ενώ στη συνέχεια χρησιμοποιήθηκε η δυναμική συνάρτηση αξιολόγησης όπως περιγράφεται στο Κεφάλαιο 5.2.5. Τα αποτελέσματα των πειραμάτων (Σχήμα 5.7) χρησιμοποιώντας τη στατική συνάρτηση αξιολόγησης δείχνουν ότι μόνο ο αλγόριθμος TVIW κατάφερε να δημιουργήσει εφικτό χρονοπρόγραμμα σε ποσοστό 4% και 1% αξιοποιώντας τις τοπολογίες αστέρα και δακτυλίου αντίστοιχα. Χρησιμοποιώντας όμως τη δυναμική συνάρτηση αξιολόγησης έγινε κατορθωτό να παράξει εφικτό χρονοπρόγραμμα και ο CPSO σε ποσοστό 2% ενώ επιτεύχθηκαν συνολικά καλύτερα αποτελέσματα αφού τα ποσοστά των υπόλοιπων αλγορίθμων αυξήθηκαν. Αποδεικνύεται ξανά η <<παρεμπόδιση>> της οποίας τυγχάνει το σμήνος από τη συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών.

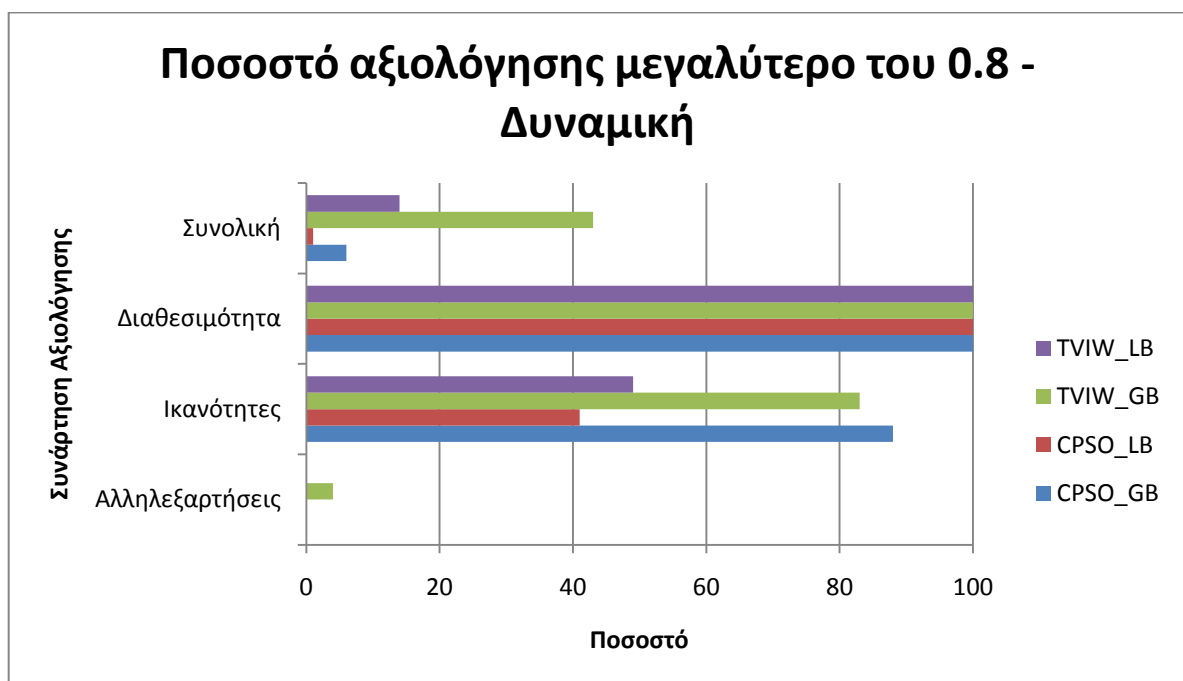


Σχήμα 5.8 – Αποτελέσματα εύρεσης εφικτού χρονοπρογράμματος με στατική και δυναμική συνάρτηση

Στην προσπάθεια να παρουσιαστεί με περισσότερα αποδεικτικά στοιχεία η δυσκολία που παρουσιάζεται στην εξελικτική πορεία του σμήνους έγινε καταγραφή της αποτίμησης κάθε επιμέρους συνάρτησης για όλους τους αλγορίθμους που εξετάστηκαν σε αυτό το σενάριο για τη στατική και τη δυναμική συνάρτηση αξιολόγησης. Συγκεκριμένα έγινε καταμέτρηση του ποσοστού των συναρτήσεων που είχαν ποσοστό αξιολόγησης μεγαλύτερο του 0.8.



Σχήμα 5.9 – Αποτελέσματα αξιολόγησης συναρτήσεων με ποσοστό μεγαλύτερο του 0.8 με στατική συνάρτηση αξιολόγησης



Σχήμα 5.10 – Αποτελέσματα αξιολόγησης συναρτήσεων με ποσοστό μεγαλύτερο του 0.8 με δυναμική συνάρτηση αξιολόγησης

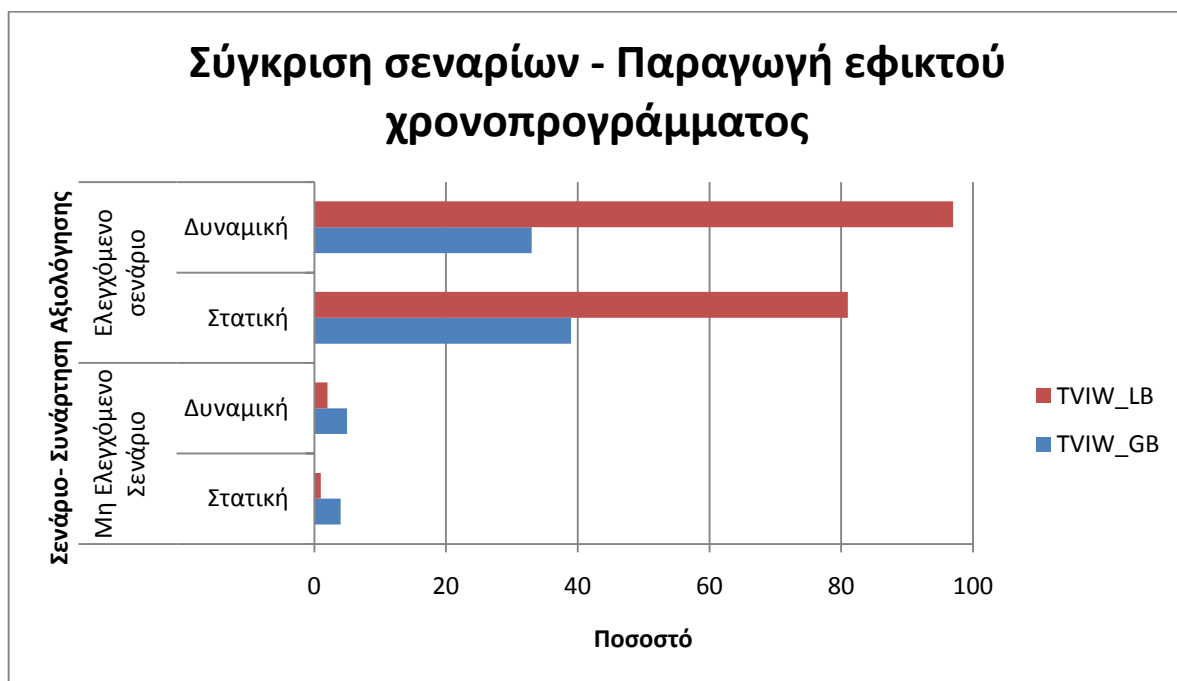
Όπως φαίνεται από τα Σχήματα 5.9 και 5.10 η συνάρτηση αξιολόγησης της διαθεσιμότητας των υπαλλήλων σε όλες τις αποτιμήσεις έχει ποσοστό μεγαλύτερο του 0.8 ενώ η αντίστοιχη συνάρτηση εκπλήρωσης των απαιτούμενων ικανοτήτων έχει αρκετά ικανοποιητικό ποσοστό για όλους τους αλγορίθμους. Αντίθετα, η συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών είναι ελάχιστες οι φορές που η τιμή της υπερβαίνει το 0.8 τόσο στη στατική όσο και στη δυναμική συνάρτηση αξιολόγησης. Γι' αυτό άλλωστε είναι μειωμένο το ποσοστό εύρεσης εφικτού χρονοπρογράμματος (Σχήμα 5.8). Είναι φανερό λοιπόν ότι η συνάρτηση αξιολόγησης των αλληλεξαρτήσεων δεν καθοδηγεί ορθά το σμήνος και <<παρεμποδίζει>> κατά κάποιο τρόπο την εξελικτική του λειτουργία.

5.4 Σύνοψη

Μέσα από τις πειραματικές διαδικασίες που περιγράφονται στο κεφάλαιο αυτό έγινε προσπάθεια για αξιολόγηση των κυριότερων παραλλαγών του αλγορίθμου PSO τόσο σε ελεγχόμενα όσο και σε μη ελεγχόμενα πειράματα. Τα αποτελέσματα απέδειξαν ότι ο αλγόριθμος είναι ικανός να βρει σε αρκετά ικανοποιητικό ποσοστό τη βέλτιστη λύση του προβλήματος καθώς επίσης και να παράξει εφικτό χρονοπρόγραμμα. Επιπλέον καθίσταται αναγκαία η ύπαρξη τεχνικής απεμπλοκής ώστε το σμήνος να μπορεί να ξεφεύγει από τη στασιμότητα και να συνεχίζει την εξελικτική του πορεία. Αντίστοιχα παρατηρήθηκε <<παρεμπόδιση>> στην εξελικτική λειτουργία των αλγορίθμων που οφειλόταν στην συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών.

Παράλληλα όμως αποδείχτηκε ότι όσο αυξάνεται το μέγεθος των έργων μειώνεται αντίστοιχα το ποσοστό εύρεσης εφικτού χρονοπρογράμματος. Αυτό φαίνεται εξάλλου στο Σχήμα 5.11 όπου στο σενάριο με επτά διεργασίες και πέντε υπαλλήλους παράχθηκε εφικτό χρονοπρόγραμμα σε ποσοστό 30% έως 97% για όλες τις τεχνικές και μεθόδους ενώ στο σενάριο με δέκα διεργασίες και έξι υπαλλήλους το καλύτερο αποτέλεσμα δεν ξεπερνά το ποσοστό του 5%. Αυτό δικαιολογείται αν ληφθεί υπόψη ότι μεγαλύτερο μέγεθος έργου σημαίνει και μεγαλύτερος αριθμός διαστάσεων που πρέπει να εξερευνηθούν και συνεπώς

μεγαλύτερος χώρος αναζήτησης που πρέπει να καλύψει το σμήνος για να βρει ένα εφικτό χρονοπρόγραμμα.



Σχήμα 5.11 – Σύγκριση σεναρίων για παραγωγή εφικτού χρονοπρογράμματος με τεχνική TVIW

Κεφάλαιο 6

Συμπεράσματα και μελλοντική εργασία

6.1 Γενικά συμπεράσματα

6.2 Μελλοντική εργασία

6.3 Επίλογος

6.1 Γενικά συμπεράσματα

Η παρούσα διπλωματική εργασία προσεγγίζει το πρόβλημα αποτυχίας των έργων παραγωγής λογισμικού και προσπαθεί να το επιλύσει αξιοποιώντας τον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων. Σκοπός είναι η επίλυση των κυριοτέρων δυσκολιών που παρουσιάζονται κατά τη διάρκεια παραγωγής λογισμικού και η προσοχή εστιάζεται στις δύο κύριες δραστηριότητες ενός διαχειριστή έργων εννοώντας το χρονοπρογραμματισμό του έργου και τη στελέχωση της ομάδας ανάπτυξης. Απαραίτητα στοιχεία είναι η ικανοποίηση των περιορισμών προτεραιότητας μεταξύ των διεργασιών και η διατήρηση της εκμετάλλευσης των διαθέσιμων πόρων σε κάθε χρονική στιγμή μέσα στα επιτρεπτά πλαίσια. Έτσι έχοντας ως δεδομένα εισόδου τις διεργασίες του έργου και τους διαθέσιμους υπαλλήλους και τροφοδοτώντας με αυτά τον αλγόριθμο PSO επιτυγχάνεται ο χρονοπρογραμματισμός του έργου και η στελέχωση των ομάδων εργασίας. Όλα αυτά πραγματοποιούνται με στόχο την ελαχιστοποίηση τόσο της χρονικής διάρκειας του έργου όσο και του συνολικού του κόστους ενώ παράλληλα η κατάλληλη στελέχωση των ομάδων εργασίας αυξάνει την ποιότητα του προϊόντος. Απώτερος στόχος βέβαια είναι η μείωση

του ποσοστού αποτυχίας των έργων παραγωγής λογισμικού. Για την εκπλήρωση των παραπάνω στόχων ήταν απαραίτητη η μελέτη αρκετών άρθρων που αφορούσαν τόσο το πρόβλημα του RCPSP όσο και τον αλγόριθμο PSO, τις δυνατότητες και τις κυριότερες παραλλαγές του ώστε να κατανοηθούν τα δύο αυτά στοιχεία σε βάθος. Στη συνέχεια μέσω της μοντελοποίησης των δεδομένων γίνεται ο συνδυασμός τους και μέσω της εξελικτικής πορείας του αλγορίθμου επιτυγχάνεται η εύρεση ενός εφικτού χρονοπρογράμματος.

Για να γίνει κατορθωτή η αξιολόγηση του αλγορίθμου πραγματοποιήθηκαν δύο ειδών πειραματικές διαδικασίες, τα ελεγχόμενα και τα μη ελεγχόμενα πειράματα. Στο πρώτο είδος πειραμάτων το βέλτιστο αποτέλεσμα ήταν γνωστό εκ των προτέρων και οι πειραματικές διαδικασίες διενεργήθηκαν για να εξευρεθούν οι παραλλαγές του αλγορίθμου με το μεγαλύτερο ποσοστό παραγωγής εφικτού χρονοπρογράμματος. Τα αποτελέσματα των πειραμάτων έδειξαν ότι οι πιο αποτελεσματικοί αλγορίθμοι ήταν ο PSO με γραμμική μείωση του βάρους αδρανείας (TVIW) και ο περιοριστικός PSO (CPSO). Στα μη ελεγχόμενα πειράματα το βέλτιστο αποτέλεσμα δεν ήταν γνωστό και ο αλγόριθμος ελεγχόταν όσον αφορά την ικανότητα παραγωγής εφικτού χρονοπρογράμματος. Πρέπει να σημειωθεί ότι κάθε αλγόριθμος αξιολογήθηκε για δύο ειδών τοπολογίες, την τοπολογία αστέρα (GB) και την τοπολογία δακτυλίου (LB).

Παρατηρώντας τα αποτελέσματα που προέκυψαν από την πειραματική αξιολόγηση των αλγορίθμων είναι φανερό ότι ο αλγόριθμος PSO μπορεί να βοηθήσει στην δημιουργία ενός εφικτού χρονοπρογράμματος και συνεπώς να αποτελέσει ένα αξιόπιστο βοηθό στο διαχειριστή έργων. Το γεγονός ότι συγκλίνει γρήγορα σε κάποιο τοπικά βέλτιστο σημείο αποδεικνύει ότι δεν χρειάζονται αρκετές επαναλήψεις και επομένως σπατάλη πολύτιμου χρόνου για να βρεθεί κάποια λύση. Επίσης είναι υποτακτική ανάγκη η εφαρμογή τεχνικής απεμπλοκής από τη στασιμότητα αφού υλοποιώντας μια τέτοια τεχνική αυξάνεται η πιθανότητα το σμήνος να ξεφύγει από κάποια τοπικά βέλτιστη λύση και να συνεχίσει την εξελικτική του πορεία προς κάποια καλύτερη λύση. Εξίσου σημαντικό ρόλο στην γενικότερη εξελικτική λειτουργία του αλγορίθμου έχει και η τοπολογία του σμήνους, ο τρόπος διασύνδεσης δηλαδή μεταξύ των σωματιδίων. Διαφοροποιώντας την τοπολογία αλλάζει ταυτόχρονα ο τρόπος που μεταφέρονται οι πληροφορίες από μια γειτονιά

σωματιδίων σε κάποια άλλη και κατ' επέκταση από ένα σωματίδιο σε άλλο. Γίνεται λοιπόν αντιληπτό ότι η απόδοση του αλγορίθμου εξαρτάται σε σημαντικό βαθμό από τις τεχνικές που εφαρμόζονται ώστε να βοηθήσουν την πορεία του αλγορίθμου στην προσπάθεια εύρεσης ενός εφικτού χρονοπρογράμματος.

Στη διπλωματική αυτή εργασία γίνεται προσπάθεια επίλυσης του RCPSP χρησιμοποιώντας τον αλγόριθμο PSO. Είναι η πρώτη ερευνητική προσπάθεια που καταβάλλεται αξιοποιώντας το συγκεκριμένο αλγόριθμο και ήταν επόμενο να καταγραφούν ορισμένα προβλήματα, τα οποία σε επόμενη έρευνα πρέπει να επιλυθούν. Καταρχάς η συνάρτηση αξιολόγησης των αλληλεξαρτήσεων μεταξύ των διεργασιών δεν καθοδηγεί ορθά το σμήνος αλλά εν μέρει το παρεμποδίζει. Προσθέτοντας σε αυτό το γεγονός ότι οι συναρτήσεις αξιολόγησης των ικανοτήτων και της διαθεσιμότητας των υπαλλήλων υπερκαλύπτουν αυτή των αλληλεξαρτήσεων μεταξύ των διεργασιών, το σμήνος αποπροσανατολίζεται με αποτέλεσμα σε αρκετές περιπτώσεις να αδυνατεί να δημιουργήσει ένα εφικτό χρονοπρόγραμμα. Επιπλέον παρατηρήθηκε δυσκολία στην εύρεση εφικτής λύσης σε σενάρια με έργα πολλών διεργασιών. Όσο αυξάνεται το πλήθος των διεργασιών που αποτελούν ένα έργο ανάλογα αυξάνονται οι διαστάσεις του χώρου αναζήτησης και ο απαιτούμενος χώρος που έχει να καλύψει το σμήνος ώστε να βρει εφικτή λύση. Επόμενο είναι να δυσκολεύει η εξελικτική λειτουργία του σμήνους και να παρατηρείται μείωση του ποσοστού παραγωγής εφικτού χρονοπρογράμματος.

6.2 Μελλοντική εργασία

Έχει αποδειχτεί ότι το συγκεκριμένο σύστημα μπορεί να χρησιμοποιηθεί ως μια αρχική λύση στο πρόβλημα του RCPSP αλλά αυτό δεν σημαίνει ότι δεν επιδέχεται περαιτέρω βελτιώσεων και αναβαθμίσεων στην προσπάθεια βελτίωσης των αποτελεσμάτων. Αντιθέτως μπορούν να γίνουν επιπλέον αλλαγές και μετατροπές στο σύστημα ώστε να λαμβάνει περισσότερα στοιχεία υπόψη.

Καλό θα είναι να καταστεί εφικτή η κωδικοποίηση της προσωπικότητας που διαθέτουν οι υπαλλήλοι και η εισαγωγή της στο σύστημα αφού η συνοχή μιας ομάδας εργασίας έχει

πρωτεύοντα ρόλο στην αποδοτικότητά της. Αυτό αντανακλά συνήθως την τελική ποιότητα του έργου αφού ομάδες εργασίας που έχουν παρόμοια χαρακτηριστικά έχουν κατορθώσει να παράξουν καλύτερης ποιότητας έργο σε σχέση με ομάδες εργασίας με αντίθετα χαρακτηριστικά. Αν για παράδειγμα βρεθούν στην ίδια ομάδα δύο υπαλλήλοι με αντίθετα χαρακτηριστικά προσωπικότητας τότε πιθανόν να μην μπορούν να συνεργαστούν και να φέρουν εις πέρας την εργασία που τους έχει ανατεθεί. Αποτέλεσμα αυτού είναι είτε η αποτυχία εκτέλεσης της εργασίας είτε η μειωμένη ποιότητά της. Για τον σκοπό αυτό μπορεί να κωδικοποιηθεί η προσωπικότητα της ομάδας ανάπτυξης βάσει του Myers-Briggs Type Indicator (MBTI) και να διευκολυνθεί η στελέχωση των ομάδων για κάθε διεργασία.

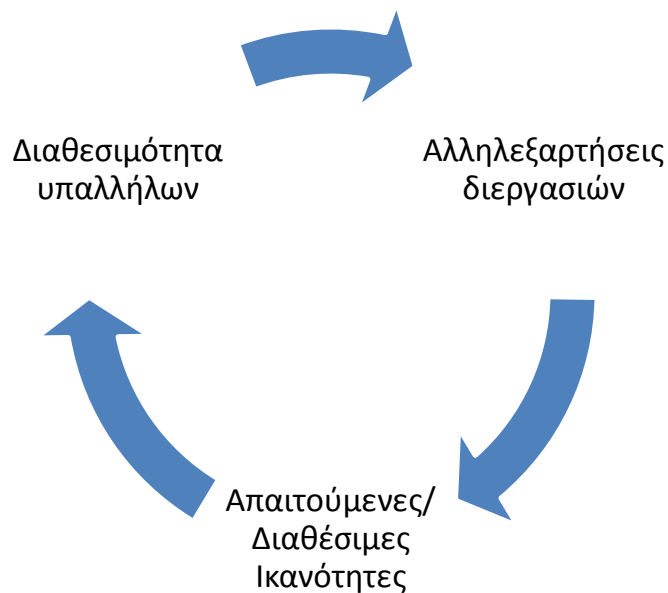
Στο υπάρχον σύστημα τόσο οι απαιτούμενες ικανότητες για την εκπλήρωση μιας διεργασίας όσο και οι διαθέσιμες ικανότητες από κάποιον υπάλληλο καθορίζονται από τη δυαδική ακολουθία, δηλαδή μια ικανότητα είτε υπάρχει (1) είτε δεν υπάρχει (0). Επιπλέον στοιχείο που μπορεί να χρησιμοποιηθεί για αναβάθμιση του υπάρχοντος συστήματος είναι η εισαγωγή της εμπειρίας για κάθε ικανότητα. Έτσι κάθε ικανότητα χαρακτηρίζεται από ένα ποσοστό εμπειρίας που απαιτείται για την εκπλήρωση της διεργασίας ή που διαθέτει ο εκάστοτε υπάλληλος. Επιτυγχάνεται επομένως με τη διαβάθμιση των ικανοτήτων ακόμα καλύτερη στελέχωση των ομάδων εργασίας με σκοπό πάντοτε τη βελτίωση της ποιότητας του έργου.

Βέβαια η απόδοση του αλγορίθμου μπορεί να βελτιωθεί αν επιλυθούν οι δυσκολίες που αναφέρονται στο προηγούμενο κεφάλαιο. Παράλληλα όμως μπορούν να γίνουν επιπλέον αλλαγές ώστε να γίνει αποδοτικότερος ο αλγόριθμος και να επιτευχθούν καλύτερα ποσοστά παραγωγής εφικτού χρονοπρογράμματος.

Συγκεκριμένα, τα σωματίδια του σμήνους κατά την αρχικοποίηση τους τοποθετούνται σε τυχαίες θέσεις και με τυχαίες ταχύτητες στο χώρο αναζήτησης. Αυτό όμως από μόνο του δεν εξασφαλίζει ότι θα καλύψουν όλο το χώρο αναζήτησης αφού είναι πιθανόν να κατανεμηθούν μόνο ένα μέρος του χώρου λόγω της τυχαιότητας που υπάρχει. Συνεπώς μπορεί να χρησιμοποιηθεί κάποια τεχνική που να κατανέμει καλύτερα τα σωματίδια στο χώρο αναζήτησης αποσκοπώντας στην όσο το δυνατό μεγαλύτερη κάλυψη του χώρου.

Αξιοποιώντας μια τέτοια τεχνική αυξάνονται παράλληλα οι πιθανότητες του σμήνους να μην κολλήσει σε κάποια τοπικά βέλτιστη λύση και να βρει την καλύτερη λύση.

Από την άλλη όμως μπορεί να υλοποιηθεί μια πιο εξεζητημένη παραλλαγή του αλγορίθμου η οποία ίσως επιφέρει καλύτερα αποτελέσματα. Σε αυτή τη διπλωματική εργασία υλοποιήθηκαν οι κυριότερες παραλλαγές του βασικού αλγορίθμου PSO όπως ο PSO με γραμμική μείωση του βάρους αδρανείας (TVIW) και ο αντίστοιχος με τυχαίο βάρος αδρανείας (RVIW), ο περιοριστικός PSO (CPSO) και ο PSO με γραμμική προσαρμογή των συντελεστών επιτάχυνση (LAAC). Μια εναλλακτική λύση θα ήταν η υλοποίηση του PSO με πολλαπλούς στόχους αφού ουσιαστικά οι τρεις επιμέρους παραμέτροι που προσπαθεί να βελτιστοποιήσει η παρούσα μελέτη συγκρούονται μεταξύ τους όπως φαίνεται στο Σχήμα 6.1. Με αυτή τη μέθοδο θα μπορούσε να βρεθεί η καμπύλη στη γραφική παράσταση αξιολόγησης των συναρτήσεων στην οποία επιτυγχάνονται οι βέλτιστες τιμές για το πρόβλημα.



Σχήμα 6.1 – Σύγκρουση επιμέρους συναρτήσεων αξιολόγησης

6.3 Επίλογος

Είναι πλέον αντιληπτό ότι ο ρόλος ενός διαχειριστή έργων είναι ζωτικής σημασίας ενώ η ευθύνη που αναλαμβάνει είναι τεράστια καθώς είναι ο άμεσα υπεύθυνος σε ενδεχόμενη αποτυχία ολοκλήρωσης ενός έργου παραγωγής λογισμικού. Ο φόρτος εργασίας και ο όγκος δεδομένων που καλείται να διαχειριστεί είναι μεγάλος με αποτέλεσμα πολλές φορές να μην μπορεί να πάρει τις κατάλληλες αποφάσεις προς όφελος του έργου. Γι' αυτό άλλωστε τα τελευταία χρόνια έχουν καταγραφεί πολλές προσπάθειες για δημιουργία συστημάτων υποστήριξης αποφάσεων με σκοπό να διευκολύνουν τις αποφάσεις που παίρνει ο διαχειριστής έργων. Το σύστημα που έχει υλοποιηθεί στην παρούσα διπλωματική εργασία αποτελεί το πρώτο βήμα στην υλοποίηση ενός τέτοιου συστήματος. Προσθέτοντας ορισμένα στοιχεία ούτως ώστε να λαμβάνει υπόψη περισσότερες παραμέτρους και βελτιώνοντας τον αλγόριθμο PSO μπορεί να καταστεί ένας πλήρες σύστημα υποστήριξης αποφάσεων.

Συνοψίζοντας θα ήθελα να αναφέρω ότι οι τελευταίοι μήνες αποτέλεσαν για μένα ένα λαμπρό παράδειγμα της διαδικασίας έρευνας και ανάπτυξης που ακολουθείται στα ανώτατα εκπαιδευτικά ιδρύματα. Μέσα από τη μελέτη των επιστημονικών άρθρων εκτός του ότι διευρύνθηκαν οι γνώσεις μου, έγινε κατορθωτή η ανάπλαση του τρόπου σκέψης και της μέχρι τώρα νοοτροπίας για αντιμετώπιση προβλημάτων και δυσκολιών. Πρέπει να ομολογήσω ότι αν και στα αρχικά στάδια το γενικότερο ερευνητικό πεδίο δεν ήταν στα άμεσα ενδιαφέροντά μου, εντούτοις η τριβή με το θέμα μου κέντρισε το ενδιαφέρον και με ώθησε να ασχοληθώ πιο ζεστά με το αντικείμενο. Έγινε κατορθωτό να κατανοήσω ότι μπορώ να αποτελέσω κι εγώ μέρος της ευρύτερης ερευνητικής κοινότητας. Καταλήγοντας μπορώ πλέον να πω με σιγουριά ότι η διπλωματική αυτή εργασία με έχει βοηθήσει σημαντικά σαν επικείμενο επιστήμονα της πληροφορικής.

Βιβλιογραφία

- [1] Davis, E.W.: Project Scheduling under Resource Constraints – a Historical Review and Categorization of Procedures. AIIE Transactions 5, 297 – 313.
- [2] Merkle, D., Middendorf, M., & Schneck, H.: Ant colony optimization for resource constrained project scheduling. IEEE Transactions on Evolutionary Computation 6 (2002), 333 – 346.
- [3] Blazewicz, J., Lenstra, J. K., & Rinnoy Kan, A.H.G.: Scheduling subject to resource constraints: classification and complexity. Discrete Appl. Math. 5 (1983), 11 – 24.
- [4] Carruthers, J.A., & Battersbym A.: Advances in critical path methods. Operational Research Quarterly 17 4 (1966), 359 – 380.
- [5] Patterson, J.H., & Huber, W.D.: A horizon-varying, zero-one approach to project scheduling. Management Science 20 (1974), 990 – 998.
- [6] Deneulemeester, E., & Herrolen, W.: A branch-and-bound procedure for the multiple resource-constraint project scheduling problem. Management Science 38 12 (1992), 1803 – 1818.
- [7] Deneulemeester, E., & Herrolen, W.: New benchmark result for the resource-constrained project scheduling problem. Management Science 43 11 (1997), 1485 – 1489.
- [8] Minozzi, A., Maniezzo, V., Ricciardelli, S., & Bianco, L.: An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. Management Science 44 5 (1998), 714 – 729.

- [9] Brucker, P., Knust, S., Schoo, A., & Thiele, O.: A branch and bound algorithm for the resource constrained project scheduling problem. *European Journal of Operational Research* 107 (1998), 272 - 288
- [10] Sprecher, A.: Solving RCPSP efficiently at modest memory requirements. Technical Report, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel No. 425 (1196)
- [11] Jalilvand, A., Khanmohaddani, S., & Shabaninia, F.: Scheduling of sequence-dependant jobs on parallel multiprocessor systems using branch and bound based Petri net. In *Emerging Technologies, proceedings of the IEEE symposium* (2005), 334 – 339.
- [12] Alcaraz, J., Maroto, C.: A robust genetic algorithm for resource allocation in project scheduling. *Annals of Operations Research* 102 (2001), 83 – 109.
- [13] Kolisch, R., & Hartmann, S.: Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis. In: Weglarz J., editor. *Project scheduling: Recent models, algorithms and applications*. Kluwer Academic Publishers (1999), 147 – 78.
- [14] Kolisch, R., & Hartmann, S.: Experimental investigation of heuristics for resource-constrained project scheduling: an update. *European Journal of Operational Research* 174 1 (2006), 23 – 37.
- [15] Alvarez-Valdes, R., & Tamarit, J.M.: Heuristic Algorithms for resource-constrained project scheduling: A review and an empirical analysis, in Slowinski, R., Weglarz, J. (Eds), *Advances in Project Scheduling*, Elsevier, Amsterdam, the Netherlands, (1989), 113 – 134.

- [16] Tormos, P., & Lova, A.: A competitive heuristic solution technique for resource-constrained project scheduling, *Annals of Operations Research* 102 (2001), 65 – 81.
- [17] Lawrence, S.R.: Resource constrained project scheduling – A computational comparison of heuristic scheduling techniques, Technical report, Carnegie Mellon University, Pittsburgh, Pennsylvania (1985).
- [18] Bell, C.E., & Han, J.: A new heuristic solution method in resource – constrained project scheduling, *Naval Research Logistics* 38 (1991), 315 – 331.
- [19] Kirkpartick, S., Gelatt, C.D, Jr., & Vrecchi, M.P.: Optimization by Simulated Annealing, *Science* 220 (4598) (1983), 671 – 680.
- [20] Valls, V., Ballestin, F., & Quintanilla, M.S.: Justification and RCPSP: A technique that pays, *European Journal of Operational Research* 165 (2005), 375 – 386.
- [21] Bouleimen, K., & Lecocq, H.: A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research* 140 (2) (2003), 268 – 281.
- [22] Glover, F.: Tabu Search – Part I, *ORSA Journal on Computing* 1 (3) (1989), 190 – 206.
- [23] Thomas, P.R., Salhi, S.: A tabu search approach for the resource constrained project scheduling problem, *Journal of heuristics* 4 (1998), 123 – 139.
- [24] Nonobe, K., Ibaraki, T.: Formulation and tabu search algorithm for the resource constrained project scheduling problem, in : Ribeiro, C.C., Hansen, P. (Eds), *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers (2002), 557 – 588.

- [25] Klein, P.: Project scheduling with time-varying resource constraints, *International Journal of Production Research* 38 (16) (2000), 619 – 638.
- [26] Holland, J.H.: Genetic algorithms and classifier systems: foundations and future directions. *Proceedings of the second international conference on genetic algorithms and their application* (1987), 82 – 89.
- [27] Toklu, Y.C.: Application of genetic algorithms to construction scheduling with or without resource constraints, *Canadian Journal of Civil Engineering* 29 (2002), 421 – 429.
- [28] Coelho, J., & Tavares, L.: Comparative analysis of metaheuristics for the resource constrained project scheduling problem, Technical Report, Department of Civil Engineering, Instituto Superior Tecnico, Portugal (2003)
- [29] Liu, Z., & Wang, H.: GA – based resource constrained project scheduling with the objective of minimizing activities' cost, *Lecture Notes in Computer Science* 3644 (2005), 937 – 946.
- [30] Dorigo, M., & Gambardella, L.M.: Ant colony system: A cooperative learning approach to the travelling salesman problem, *IEEE Transactions on Evolutionary Computation* 1 (1) (1997), 53 – 66
- [31] Merkle, D., & Middendorf, M.: An algorithm with a new pheromone evaluation rule for total tardiness problems, In: E.J.-W. Boers, Editor, *Lecture Notes in Computer Science* vol. 2037, Springer-Verlag, (1992), 287 – 296.
- [32] Stuzle, T., Hoos H.H.: Max-Min Ant system, *Future Generat Comput Syst* 16 (8) (2000), 899 – 914.

- [33] Eberhart, R.C., & Kennedy, J.: A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human science, Nayoya, Japan, Piscataway:IEEE (1995), 39 – 45.
- [34] Wolpert, D.H., & Macready, W.G.: No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997), 67 – 82.
- [35] Heppner F., & Grenander, U.: A stochastic nonlinear model for coordinated bird flocks. In: Krasner S (eds) The ubiquity of chaos, AAAS publications, Washington DC (1990)
- [36] Eberhart, R.C., Simpson, P., & Dobbins, R.: Computational Intelligence PC tools. AP Professional, San Diego, CA, (1996), Chapter 6 212 – 226.
- [37] Shi, Y., & Eberhart, R.C.: A new particle swarm optimizer, Proceedings of IEEE international conference on evolutionary computation, IEEE press, Piscataway, NJ (1998), 69 – 73.
- [38] Shi, Y., & Eberhart, R.C.: Parameter selection in particle swarm optimization, Proceedings on Evolutionary Programming VII, Lecture notes in computer science 1447 (1998), 591 – 600.
- [39] Eberhart, R.C, & Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In Proceedings on the IEEE congress on evolutionary computation, San Diego, CA (2000), 84 – 88.
- [40] Eberhart, R.C, & Shi, Y.: Tracking and optimizing dynamic systems using particle swarms. In Proceedings on the IEEE congress on evolutionary computation, Seoul, Korea (2001), 94 – 100.

- [41] Zheng Y.L., Ma, L.H., Zhang, L.Y., & Qian J.X.: On the convergence analysis and parameter selection in particle swarm optimization. In Proceedings of the IEEE international conference on machine learning and cybernetics (2003), 1802 – 1807.
- [42] Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters, 85 (6) (2003), 317 – 325.
- [43] Clerc, M., & Kennedy, J.: The particle swarm: explosion, stability and convergence in a multi-dimensional complex space, IEEE Transaction on Evolutionary Computation 6 (1) (2002), 58 – 73.
- [44] Parsopoulos K.E., & Vrahatis, M.N.: UPSO: a unified particle swarm optimization scheme. In Lecture series on computer and computational sciences, Proceedings of international conference on computational methods in sciences and engineering (ICCMSE 2004), VSP International Science Publishers, Zeist, The Netherlands (2004), 68 – 73.
- [45] Parsopoulos K.E., & Vrahatis, M.N.: Unified Particle Swarm Optimization in Dynamic Environments. In Rothlauf F et al (eds) EvoWorkshops, LNCS 3449 (2005), 590 – 599.
- [46] Parsopoulos K.E., & Vrahatis, M.N.: Unified Particle Swarm Optimization for tackling operations research problems. Proceedings swarm intelligence symposium SIS 2005 (2005), 53 – 59.
- [47] Kennedy, J., & Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. Proceedings of the conference on systems, man and cybernetics, New Jersey (1997), 4104 – 4109.
- [48] Kennedy, J., & Spears, M.: Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem

generator. Proceedings of the international conference on evolutionary computation, NJ IEEE Service Center (1998), 78 – 83.

- [49] Agrafiotis, D. K., & Cedeno, W.: Feature selection for structure-activity correlation using binary particle swarms, *Journal of Medicinal Chemistry* 45 (5) (2002), 1098 – 1107.
- [50] Mohan, C.K., & Al-Kazemi, B.: Discrete particle swarm optimization. In *Proceedings of the workshop on particle swarm optimization*, Indianapolis, IN, Purdue School of Engineering and Technology, IUPUI (2001)
- [51] Laskari, E.C., Parsopoulos, K.E, & Vrahatis, M.N.: Particle swarm optimization for integer programming, *Proceedings of the IEEE* (2002)
- [52] Kennedy, J.: The particle swarm: social adaptation of knowledge, *Proceedings of international conference on evolutionary computation*, IEEE NJ (1997), 303 – 308.
- [53] Kennedy, J.: Small words and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the IEEE congress on evolutionary computation*, IEEE NJ (1999), 1931 – 1938.
- [54] Kennedy, J., & Mendes, R.: Population structure and particle swarm performance. In *Proceedings of the IEEE congress on evolutionary computation*, IEEE Honolulu HI (2002), 1671 – 1676.
- [55] Suganthan, P.N.: Particle swarm optimizer with neighborhood operator. In *Proceedings of the IEEE congress on evolutionary computation*. IEEE (1999), 1958 – 1962.

- [56] Janson, S., & Middendorf, M.: A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on System man and cybernetics B* 35 (6) (2005), 1272 – 1282.
- [57] Liang, J.J., & Suganthan, P.N.: Dynamic multiswarm particle swarm optimizer (DMS-PSO). In *Proceedings of the IEEE swarm intelligence symposium*. IEEE (2005), 124 – 129.
- [58] Mendes, R.: Population topologies and their influence in particle swarm performance. PhD thesis, Departamento de Informatica, Escola de Engenharia, Universidade do Minho, (2004).
- [59] Carlisle, A., & Dozier, G.: Adapting particle swarm optimization to dynamic environments. In *Proceedings of International conference on artificial intelligence*, Las Vegas, NE (2000), 429 – 434.
- [60] Parrot, D., & Li, X.: A particle swarm model for tracking multiple peaks in a dynamic environment using speciation. In *Proceedings of the congress on evolutionary computation*, IEEE service center, NJ (2004), 98 – 103.
- [61] Parsopoulos, K.E., & Vrahatis, N.M.: Particle swarm optimizer in noisy and continuously changing environments. In M. H. Hamza (eds), *Artificial intelligence and soft computing* (2001), 289 – 294.
- [62] Coello Coello, C.A., & Lechuga, S.: MOPSO: A proposal for multiple objective particle swarm optimization. Technical Report EVOCINV-01-201, Evolutionary Computation Groupat CINVESTAV, Seccion de Computation, Departamento de Ingenieria Electrica, Mexico (2001)

- [63] Parsopoulos, K.E., & Vrahatis, M.N.: Particle swarm optimization method in multiobjective problems. In Proceedings of the ACM symposium on Applied Computing (2002), 603 – 607.
- [64] Hu, X., & Eberhart, R.C.: Multiobjective optimization using dynamic neighborhood particle swarm optimization. In Proceedings of the IEEE congress on evolutionary computation, Honolulu, HI, USA (2002), 1677 – 1681.
- [65] Miranda, V., & Fonseca, N.: New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control. In Proceedings of the 14th power systems computation conference, Seville Spain (2002), 21 (5) 1 – 6.
- [66] Loøvbjerg, M., Rasmussen, T.K., & Krink, T.: Hybrid particle swarm optimizer with breeding and subpopulations. In Proceedings of the third genetic and evolutionary computation conference, San Francisco, (2001), 469 – 476.
- [67] Poli, R., & Stephens, C.R.: Constrained molecular dynamics as a search and optimization tool. In M. Keijzer (eds), Lecture notes in computer science: Vol. 3003. In Proceedings of the 7th European conference in genetic programming, Coimbra, Portugal (2004), 150 – 161.
- [68] Poli, R., Kennedy, J., & Blackwell, T.: Particle swarm optimization: an overview. *Swarm Intelligence* (2007), 1(1), 33 – 57.
- [69] van den Bergh, F., & Engelbrecht, A.P.: A new locally convergent particle swarm optimizer. In Proceedings of IEEE conference on systems, man and cybernetics, Hammamet, Tunisia, (2002).
- [70] Riget, J., & Vesterstrøm, J.S.: A diversity guided particle swarm optimizer – the ARPSO. *EVALife Technical report* (2002)

- [71] Ratnaweera, A., Halgamuge, S.K., & Watson, H.C.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions in Evolutionary Computation* 8 (3) (2004), 240 – 255.
- [72] Banks, A., Vincent, J., & Anyakoha, C.: A review of particle swarm optimization, part I: background and development. *Nat Comput* 6(4) (2007,), 467 – 484.