



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής Και Υπολογιστών

Design and Implementation of Heuristic Techniques for Power Consumption Reduction in Integrated Circuits

Διπλωματική Εργασία Του Μιχαήλ Μουρτζάκη

Αριθμός Ειδικού Μητρώου : 9742

Επιβλέπων: Κωνσταντίνος Παπαλάμπρου
Επίκουρος Καθηγητής Α.Π.Θ.

Σεπτέμβριος 2024

Περίληψη

Στην παρούσα, εργασία ασχολούμαστε με την επίλυση του προβλήματος της ελαχιστοποίησης κατανάλωσης ισχύος σε deep submicron technology buses. Αυτό το κάνουμε με την χρήση κάποιων ευρετικών μεθόδων που χρησιμοποιούνται για την επίλυση quadratic assignment problems. Αρχικά μιλάμε για την επιστήμη της Επιχειρησιακής Έρευνας και τις κατηγορίες προβλημάτων που ασχολείται μια εκ των οποίων είναι και τα quadratic assignment problems. Βλέπουμε κάποιες πληροφορίες για αυτό και στην συνέχεια περνάμε στην ανάλυση του προβλήματος της ελαχιστοποίησης κατανάλωσης ισχύος σε deep submicron technology buses το οποίο όπως και δείχνουμε μπορεί να διατυπωθεί ως ένα quadratic assignment problem. Αναφέρουμε τις ευρετικές μεθόδους που χρησιμοποιήθηκαν, τον τρόπο λειτουργίας τους καθώς και τις παραμέτρους του προβλήματος. Τέλος παρουσιάζουμε τα αποτελέσματα των δοκιμών που έγιναν και σχολιάζουμε ποιες ευρετικές μέθοδοι είναι αποτελεσματικές στην επίλυση του προβλήματος μας.

Abstract

In this diploma thesis, we deal with the problem of minimizing power consumption in deep submicron technology buses. We do this by using certain heuristic methods that are used to solve quadratic assignment problems. First, we discuss the field of Operations Research and the categories of problems it addresses, one of which is quadratic assignment problems. We review some information about this and then proceed to the analysis of the problem of minimizing power consumption in deep submicron technology buses, which, as we show, can be formulated as a quadratic assignment problem. We mention the heuristic methods used, how they work, and the variables of the problem. Finally, we present the results of the tests conducted and discuss which methods are effective in solving our problem.

Ευχαριστίες

Ολοκληρώνοντας την διπλωματική μου εργασία, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ σε όλους όσους συνέβαλαν στην επιτυχή υλοποίησή της. Αρχικά, ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου Κωνσταντίνο Παπαλάμπρου για την πολύτιμη καθοδήγηση και συμβουλές που μου παρείχε κατά την διάρκεια της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου, τους φίλους μου και τους συμφοιτητές μου για την στήριξη και βοήθεια που μου παρείχαν, ο καθένας με τον τρόπο του, καθ' όλη τη διάρκεια των σπουδών μου. Η υποστήριξή όλων σας υπήρξε ανεκτίμητη και σας είμαι ειλικρινά ευγνώμων.

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ	6
1.1	Ιστορική Αναδρομή	6
1.2	Η Μέθοδος Της Επιχειρησιακής Έρευνας	9
1.3	Γραμμικός Προγραμματισμός και Μέθοδοι Επίλυσης	11
1.4	Γραμμικός Ακέραιος Προγραμματισμός και Μέθοδοι Επίλυσης	12
1.5	Μη Γραμμικός Προγραμματισμός και Μέθοδοι Επίλυσης	12
2	QUADRATIC ASSIGNMENT PROBLEM	14
2.1	Εισαγωγή Στο Quadratic Assignment Problem	14
2.2	Διατυπώσεις Του Quadratic Assignment Problem	15
2.2.1	Quadratic Integer Program Διατύπωση	16
2.2.2	Trace Διατύπωση	17
2.3	Κάτω Όρια	18
2.4	Μέθοδοι Επίλυσης	18
2.4.1	Ακριβείς Μέθοδοι	19
2.4.2	Ευρετικές Μεθόδους	19
3	ΤΟ ΠΡΟΒΛΗΜΑ ΚΑΤΑΝΑΛΩΣΗΣ ΙΣΧΥΟΣ ΣΕ DEEP SUBMICRON TECHNOLOGY BUSES ΩΣ QAP	20
3.1	Εισαγωγή Στο Πρόβλημα	20
3.2	Το Μοντέλο Του Deep Submicron Bus	21
3.3	Η QAP Διατύπωση Της Μετάθεσης Γραμμών	24
4	ΜΕΘΟΔΟΙ ΕΠΙΛΥΣΗΣ ΚΑΙ ΠΑΡΑΜΕΤΡΟΙ	25
4.1	Γενικές Παράμετροι Και Κώδικας	25
4.2	Εξαντλητική Μέθοδος	25
4.3	Ευρετικές Μέθοδοι	28
4.3.1	Simulated Annealing (SA)	28
4.3.2	Tabu Search (TABU)	29
4.3.3	Iterative Local Search Algorithm (ILSA)	30
4.3.4	Fast Ant System (FANT)	31
4.3.5	Greedy Randomized Adaptive Search Procedure (GRASP)	33
5	ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ	34
5.1	Αποτελέσματα Με Είσοδο Τυχαίων Δεδομένων	35
5.1.1	Διαγράμματα για τυχαία είσοδο και $n = 10$	35
5.1.2	Διαγράμματα για τυχαία είσοδο και $n = 13$	36
5.1.3	Διαγράμματα για τυχαία είσοδο και $n = 16$	37

5.1.4	Διαγράμματα για τυχαία είσοδο και $n = 20$	38
5.1.5	Διαγράμματα για τυχαία είσοδο και $n = 25$	39
5.1.6	Διαγράμματα για τυχαία είσοδο και $n = 30$	40
5.1.7	Διαγράμματα για τυχαία είσοδο και $n = 40$	41
5.1.8	Διαγράμματα για τυχαία είσοδο και $n = 50$	42
5.1.9	Διαγράμματα για τυχαία είσοδο και $n = 75$	43
5.1.10	Διαγράμματα για τυχαία είσοδο και $n = 100$	44
5.1.11	Διαγράμματα για τυχαία είσοδο και $n = 150$	45
5.1.12	Σχολιασμός αποτελεσμάτων για τυχαία είσοδο	46
5.2	Αποτελέσματα Με Είσοδο Συγκεκριμένα Dataset	47
5.2.1	Διαγράμματα με είσοδο το Dataset : Sine 16	47
5.2.2	Διαγράμματα με είσοδο το Dataset : Sine 24	48
5.2.3	Διαγράμματα με είσοδο το Dataset : Sine 32	49
5.2.4	Διαγράμματα με είσοδο το Dataset : RGB 24	50
5.2.5	Σχολιασμός αποτελεσμάτων με είσοδο συγκεκριμένα Dataset και σύγκριση τους με εκείνα για τυχαία είσοδο	51
6	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ	52
A'	Ακρωνύμια και Συντομογραφίες	54

Κατάλογος Σχημάτων

1.1 Η μέθοδος της επιχειρησιακής έρευνας [1]	9
3.1 Distributed μοντέλο των γραμμών ενός διαύλου (Συζευγμένες γραμμές μετάδοσης) [2].	21
3.2 Απλοποιημένο (προσεγγιστικό ενεργειακά) DSM μοντέλο [3].	22
4.1 Exhaustive Search ($n = 7, m = 10000, NrTrials = 100$).	26
4.2 Exhaustive Search ($n = 8, m = 10000, NrTrials = 100$).	26
4.3 Exhaustive Search ($n = 9, m = 10000, NrTrials = 100$).	27
4.4 Exhaustive Search ($n = 10, m = 10000, NrTrials = 100$).	27
4.5 Μέθοδος κατασκευής λύσης [4].	31
4.6 Ενημέρωση του πίνακα T.	32
5.1 Αποτελέσματα για το $n = 10$ σε δοκιμή 150 δευτερολέπτων.	35
5.2 Αποτελέσματα για το $n = 13$ σε δοκιμή 600 δευτερολέπτων.	36
5.3 Αποτελέσματα για το $n = 13$ σε δοκιμή 3600 δευτερολέπτων.	36
5.4 Αποτελέσματα για το $n = 16$ σε δοκιμή 600 δευτερολέπτων.	37
5.5 Αποτελέσματα για το $n = 16$ σε δοκιμή 3600 δευτερολέπτων.	37
5.6 Αποτελέσματα για το $n = 20$ σε δοκιμή 600 δευτερολέπτων.	38
5.7 Αποτελέσματα για το $n = 20$ σε δοκιμή 3600 δευτερολέπτων.	38
5.8 Αποτελέσματα για το $n = 25$ σε δοκιμή 600 δευτερολέπτων.	39
5.9 Αποτελέσματα για το $n = 25$ σε δοκιμή 3600 δευτερολέπτων.	39
5.10 Αποτελέσματα για το $n = 30$ σε δοκιμή 600 δευτερολέπτων.	40
5.11 Αποτελέσματα για το $n = 30$ σε δοκιμή 3600 δευτερολέπτων.	40
5.12 Αποτελέσματα για το $n = 40$ σε δοκιμή 600 δευτερολέπτων.	41
5.13 Αποτελέσματα για το $n = 40$ σε δοκιμή 3600 δευτερολέπτων.	41
5.14 Αποτελέσματα για το $n = 50$ σε δοκιμή 600 δευτερολέπτων.	42
5.15 Αποτελέσματα για το $n = 50$ σε δοκιμή 3600 δευτερολέπτων.	42
5.16 Αποτελέσματα για το $n = 75$ σε δοκιμή 600 δευτερολέπτων.	43
5.17 Αποτελέσματα για το $n = 75$ σε δοκιμή 3600 δευτερολέπτων.	43
5.18 Αποτελέσματα για το $n = 100$ σε δοκιμή 600 δευτερολέπτων.	44
5.19 Αποτελέσματα για το $n = 100$ σε δοκιμή 3600 δευτερολέπτων.	44
5.20 Αποτελέσματα για το $n = 150$ σε δοκιμή 600 δευτερολέπτων.	45
5.21 Αποτελέσματα για το $n = 150$ σε δοκιμή 3600 δευτερολέπτων.	45
5.22 Αποτελέσματα για το Sine 16 σε δοκιμή 600 δευτερολέπτων.	47
5.23 Αποτελέσματα για το Sine 16 σε δοκιμή 3600 δευτερολέπτων.	47
5.24 Αποτελέσματα για το Sine 24 σε δοκιμή 600 δευτερολέπτων.	48
5.25 Αποτελέσματα για το Sine 24 σε δοκιμή 3600 δευτερολέπτων.	48
5.26 Αποτελέσματα για το Sine 32 σε δοκιμή 600 δευτερολέπτων.	49

5.27Αποτελέσματα για το Sine 32 σε δοκιμή 3600 δευτερολέπτων.	49
5.28Αποτελέσματα για το RGB 24 σε δοκιμή 600 δευτερολέπτων.	50
5.29Αποτελέσματα για το RGB 24 σε δοκιμή 3600 δευτερολέπτων.	50

Κεφάλαιο 1

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ

Η επιχειρησιακή έρευνα προσπαθεί να παρέχει σε αυτούς που διαχειρίζονται οργανωμένα συστήματα μια αντικειμενική και ποσοτική βάση για τη λήψη αποφάσεων. Συνήθως διεξάγεται από ομάδες επιστημόνων και μηχανικών που προέρχονται από διάφορους επιστημονικούς κλάδους. Η επιχειρησιακή έρευνα δεν αποτελεί από μόνη της μια επιστήμη, αλλά την εφαρμογή των επιστημών στη λύση διαχειριστικών και διοικητικών προβλημάτων, εστιάζοντας στην απόδοση των συστημάτων ως σύνολο και όχι στα μέρη τους ξεχωριστά. Στοχεύει στη βελτίωση των λειτουργιών των υπαρχόντων συστημάτων και αποφεύγει την αντικατάστασή τους από νέα. Κάνει εκτεταμένη χρήση παλαιών επιστημών όπως τα μαθηματικά και την στατιστική, καθώς και πιο πρόσφατων επιστημών όπως η θεωρία επικοινωνιών, η θεωρία αποφάσεων, η θεωρία οργάνωσης, οι συμπεριφορικές επιστήμες και η γενική θεωρία συστημάτων. Το αντικείμενο της επιχειρησιακής έρευνας ασχολείται με τις αποφάσεις που ελέγχουν τις λειτουργίες των συστημάτων. Πιο συγκεκριμένα ασχολείται με τις διαχειριστικές αποφάσεις και τον τρόπο με τον οποίο λαμβάνονται. Επίσης ασχολείται με την επεξεργασία των δεδομένων και των πληροφοριών που απαιτούνται για την αποτελεσματική λήψη αποφάσεων. Η επιχειρησιακή έρευνα λοιπόν εφαρμόζει την επιστημονική μέθοδο και παρέχει τη γνώση και την κατανόηση που απαιτείται για την αποτελεσματική χρήση των διαθέσιμων πόρων [5].

1.1 Ιστορική Αναδρομή

Οι ρίζες της επιχειρησιακής παρατηρούνται για πρώτη φορά στις αρχές του 1800, πολύ πριν την αρχή του Β' Παγκοσμίου Πολέμου. Βέβαια αισθητή γίνεται η εμφάνιση της από το 1885 όταν ο Ferderick W. Taylor τόνισε την εφαρμογή της επιστημονικής ανάλυσης στις μεθόδους παραγωγής. Ένας άλλος άνθρωπος της πρώιμης επιστημονικής εποχής ο οποίος χρησιμοποίησε την επιχειρησιακή έρευνα ήταν ο Henry L. Gantt. Οι περισσότερες μέθοδοι προγραμματισμού εργασιών εκείνη την εποχή ήταν τυχαίες. Για παράδειγμα μια δουλειά, μπορεί να είχε υποβληθεί σε επεξεργασία σε ένα μηχανήμα χωρίς προβλήματα, αλλά στη συνέχεια να περίμενε για μέρες μέχρι την αποδοχή από το επόμενο μηχανήμα. Ο Gantt χαρτογράφησε κάθε εργασία από μηχανή σε μηχανή, ελαχιστοποιώντας κάθε καθυστέρηση. Πλέον με τη διαδικασία Gantt είναι

δυνατό να προγραμματιστούν οι διεργασίες μηχανών μήνες νωρίτερα και παρόλα αυτά να τηρηθούν με ακρίβεια οι ημερομηνίες παράδοσης. Το 1917, ο Α.Κ.Εrlang, ένας Δανός μαθηματικός, δημοσίευσε την εργασία του για το πρόβλημα συμφόρησης της τηλεφωνικής κίνησης. Η δυσκολία ήταν ότι σε περιόδους έντονης δραστηριότητας, οι τηλεφωνητές ήταν πολλοί, με αποτέλεσμα οι κλήσεις να καθυστερούν. Λίγα χρόνια μετά από αυτό, το έργο του έγινε αποδεκτό από τα Βρετανικά Ταχυδρομεία ως βάση για την ρύθμιση των εγκαταστάσεων των δικτύων τους.

Η Πρώτη Βιομηχανική Επανάσταση συνέβαλε κυρίως στην ανάπτυξη της επιχειρησιακής έρευνας. Πριν από αυτή, οι περισσότερες βιομηχανίες ήταν μικρής κλίμακας, απασχολώντας μόνο λίγους άνδρες. Η έλευση των εργαλειομηχανών, η αντικατάσταση ανθρώπου από μηχανή ως πηγή ισχύος καθώς και τα βελτιωμένα μέσα μεταφοράς και επικοινωνιών είχαν ως αποτέλεσμα τη γρήγορη άνθηση της βιομηχανίας. Έτσι έγινε ολόένα και πιο δύσκολο για έναν μόνο άνθρωπο να εκτελεί όλες τις διευθυντικές λειτουργίες (προγραμματισμός, πώληση, αγορά, παραγωγή κ.λπ.). Κατά συνέπεια, πραγματοποιήθηκε μια διαίρεση της λειτουργίας διαχείρισης. Διευθυντές παραγωγής, μάρκετινγκ, οικονομικών, προσωπικού, έρευνας και ανάπτυξης κ.λπ., άρχισαν να εμφανίζονται. Με περαιτέρω βιομηχανική ανάπτυξη, έλαβαν χώρα περαιτέρω υποδιαιρέσεις λειτουργιών διαχείρισης.

Κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου, η στρατιωτική διοίκηση στην Αγγλία κάλεσε μια ομάδα επιστημόνων για να μελετήσουν τα στρατηγικά και τακτικά προβλήματα της εναέριας και χερσαίας άμυνας. Αυτή η ομάδα ήταν υπό τη διεύθυνση του καθηγητή P.M.S. Blackett του Πανεπιστημίου της Μάντσεστερ και πρώην αξιωματικό ναυτικού. Η ομάδα ονομάστηκε «Blackett Circus», και είχε ως στόχο την αποτελεσματικότερη κατανομή των περιορισμένων στρατιωτικών πόρων στις στρατιωτικές επιχειρήσεις (π.χ. την αποτελεσματική χρήση ραντάρ, την κατανομή των Αεροπλάνων της Βρετανικής Πολεμικής Αεροπορίας σε αποστολές, τον καθορισμό των βέλτιστων διαδρομών πτήσης για αναζήτηση υποβρυχίων).

Αυτή η ομάδα επιστημόνων ήταν ουσιαστικά η πρώτη ομάδα επιχειρησιακής έρευνας. Το όνομα επιχειρησιακή έρευνα επινοήθηκε επειδή η ομάδα διεξήγαγε έρευνα για τη (στρατιωτική) επιχείρηση. Τα ενθαρρυντικά αποτελέσματα αυτών των προσπαθειών οδήγησαν στην δημιουργία περισσότερων τέτοιων ομάδων στη Βρετανία. Αυτό σύντομα εξαπλώθηκε και στους συμμάχους (Ηνωμένες Πολιτείες, Καναδά, Γαλλία). Έτσι η χρήση της Επιχειρησιακής Έρευνας η οποία άρχισε στην Βρετανία συνέβαλε αρκετά στην εξέλιξη των στρατιωτικών επιχειρήσεων του Β' Παγκοσμίου Πολέμου.

Αμέσως μετά τον πόλεμο, η επιτυχία των στρατιωτικών ομάδων τράβηξε την προσοχή πολλών μάνατζερ βιομηχανιών, οι οποίοι αναζητούσαν λύσεις στα προβλήματά τους. Η βιομηχανική επιχειρησιακή έρευνα αναπτύχθηκε με διαφορετικό τρόπο στο Ηνωμένο Βασίλειο και τις ΗΠΑ. Στο Ηνωμένο βασίλειο η κρίσιμη οικονομική κατάσταση απαιτούσε την δραστική αύξηση της αποδοτικότητας της παραγωγής και την δημιουργία νέων αγορών. Η εθνικοποίηση μερικών βασικών βιομηχανιών αύξησε περαιτέρω το περιθώριο για την χρήση επιχειρησιακής έρευνας. Κατά συνέπεια η επιχειρησιακή έρευνά σύντομα εξαπλώθηκε από το στρατιωτικό στον κυβερνητικό, βιομηχανικό, κοινωνικό και οικονομικό σχεδιασμό.

Στις ΗΠΑ η κατάσταση ήταν διαφορετική. Μετά από τη τεράστια επιτυχία της επιχειρησιακής έρευνας στο Ηνωμένο Βασίλειο, η έρευνα για τις αμυντικές στρατιωτικές επιχειρήσεις στις Η.Π.Α. αυξήθηκε. Το μεγαλύτερο μέρος των εργατών που ασχολήθη-

καν με την επιχειρησιακή έρευνα στις στρατιωτικές επιχειρήσεις κατά την διάρκεια του πολέμου παρέμειναν στη στρατιωτική τους θητεία. Τα στελέχη της βιομηχανίας δεν ζήτησαν βοήθεια γιατί επέστρεφαν σε κατάσταση ειρήνης και πολλά από αυτά πίστευαν ότι ήταν απλώς μια νέα εφαρμογή μιας παλιάς τεχνικής. Η επιχειρησιακή έρευνα εμφανίστηκε με διάφορα ονόματα στη χώρα αυτή, όπως επιχειρησιακή ανάλυση, αξιολόγηση λειτουργίας, ανάλυση συστημάτων, αξιολόγηση συστήματος, έρευνα συστήματος και διοικητική επιστήμη.

Η πρόοδος της βιομηχανικής επιχειρησιακής έρευνας στις ΗΠΑ οφειλόταν στην έλευση της δεύτερης βιομηχανικής επανάστασης που είχε ως αποτέλεσμα την αυτοματοποίηση-αντικατάσταση του ανθρώπου. Η νέα επανάσταση ξεκίνησε γύρω στη δεκαετία του 1940 όταν οι ηλεκτρονικοί υπολογιστές έγιναν εμπορικά διαθέσιμοι. Οι ηλεκτρονικοί εγκέφαλοι επεξεργάζονταν με μεγάλη υπολογιστική ταχύτητα και αποθήκευαν τεράστιο αριθμό πληροφοριών [6].

Στη δεκαετία του 1950 η επιχειρησιακή έρευνα εξελίχθηκε σε επάγγελμα με τη δημιουργία εθνικών εταιρειών, την ίδρυση περιοδικών και ακαδημαϊκών τμημάτων στα πανεπιστήμια. Η χρήση της επεκτάθηκε πέρα από τον στρατό τόσο σε ιδιωτικές εταιρείες όσο και σε άλλους κυβερνητικούς οργανισμούς. Η πετροχημική βιομηχανία ήταν μια από τις πρώτες που αγκάλιασε ευρέως την επιχειρησιακή έρευνα για τη βελτίωση της απόδοσης των εργοστασίων, την ανάπτυξη φυσικών πόρων και τον σχεδιασμό στρατηγικής. Σήμερα, η επιχειρησιακή έρευνα διαδραματίζει σημαντικό ρόλο σε μια ποικιλία βιομηχανιών όπως:

- Αεροπορία : Προγραμματισμός αεροπλάνων και πληρωμάτων, τιμολόγηση εισιτηρίων, λήψη κρατήσεων και προγραμματισμός του μεγέθους του στόλου
- Αεροπορία : Προγραμματισμός αεροπλάνων και πληρωμάτων, τιμολόγηση εισιτηρίων, λήψη κρατήσεων και προγραμματισμός του μεγέθους του στόλου
- Φαρμακοβιομηχανία
- Λογιστικά
- Χρηματοοικονομικές υπηρεσίες : Πιστοληπτική αξιολόγηση, μάρκετινγκ και εσωτερικές λειτουργίες,
- Ξυλεία και προϊόντα ξύλου : Διαχείριση δασών και κοπή ξυλείας,
- Τοπική αυτοδιοίκηση : Ανάπτυξη υπηρεσιών έκτακτης ανάγκης
- Μελέτες πολιτικής και ρύθμισης

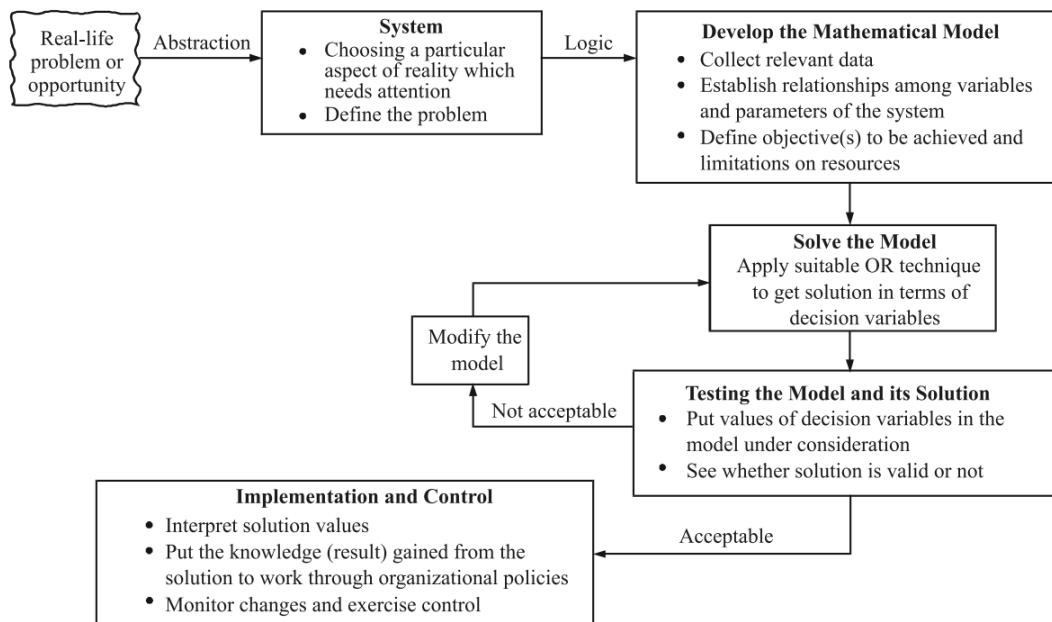
Αναλυτικότερα όλες οι εφαρμογές της επιχειρησιακής έρευνας στις μέρες μας μπορούν να βρεθούν στο [1].

Καθώς το πεδίο εξελισσόταν, ο πυρήνας απομακρύνθηκε από τις διεπιστημονικές ομάδες για να επικεντρωθεί στην ανάπτυξη μαθηματικών μοντέλων που μπορούν να χρησιμοποιηθούν για τη μοντελοποίηση, τη βελτίωση και ακόμη και τη βελτιστοποίηση συστημάτων πραγματικού κόσμου. Αυτά τα μαθηματικά μοντέλα περιλαμβάνουν ντετερμινιστικά μοντέλα όπως ο μαθηματικός προγραμματισμός, η δρομολόγηση ή ροές δικτύου και πιθανοτικά μοντέλα όπως τα δέντρα ουράς, της προσομοίωσης και αποφάσεων. Αυτές οι τεχνικές μαθηματικής μοντελοποίησης αποτελούν το βασικό πρόγραμμα σπουδών των μεταπτυχιακών και διδακτορικών προγραμμάτων στην επιχειρησιακή

έρευνα που μπορούν να βρεθούν είτε σε σχολές μηχανικής είτε σε σχολές επιχειρήσεων [7].

1.2 Η Μέθοδος Της Επιχειρησιακής Έρευνας

Κάθε ειδικός επιχειρησιακής έρευνας μπορεί να έχει τον δικό του τρόπο επίλυσης προβλημάτων. Ωστόσο, η αποτελεσματική χρήση των τεχνικών της επιχειρησιακής έρευνας απαιτεί να ακολουθηθούν μια σειρά βασικών βημάτων [1]:



Σχήμα 1.1: Η μέθοδος της επιχειρησιακής έρευνας [1]

Βήμα 1 - (Καθορισμός του προβλήματος) : Ο ορισμός του προβλήματος περιλαμβάνει τη διαδικασία αναγνώρισης, κατανόησης και περιγραφής του προβλήματος που αντιμετωπίζει ένας οργανισμός. Η διαδικασία ξεκινά με τη συλλογή γενικών πληροφοριών οι οποίες θα βοηθήσουν στην αξιολόγηση της δυσκολίας που ενδέχεται να αντιμετωπιστεί ως προς το κόστος, το χρόνο, τους πόρους, την πιθανότητα επιτυχίας της μελέτης κ.λπ.

Βήμα 2 - (Διατύπωση μαθηματικού μοντέλου) : Αφού το πρόβλημα οριστεί και κατανοηθεί με σαφήνεια, το επόμενο βήμα είναι η συλλογή των απαιτούμενων δεδομένων και στη συνέχεια η διαμόρφωση ενός μαθηματικού μοντέλου. Η διατύπωση μοντέλου απαιτεί τον καθορισμό σχέσεων μεταξύ των μεταβλητών απόφασης. Ορισμένα βασικά κομμάτια που θα δούμε συνήθως σε κάθε πρόβλημα είναι:

- Ελεγχόμενες μεταβλητές (απόφασης) : Είναι ορισμένοι παράγοντες (ή μεταβλητές) που σχετίζονται με το πρόβλημα, των οποίων οι τιμές βρίσκονται με την επίλυση του μοντέλου.
- Μη ελεγχόμενες (εξωγενείς) μεταβλητές : Οι τιμές αυτών των μεταβλητών δεν βρίσκονται υπό τον έλεγχο του λήπτη-αποφάσεων.

- Αντικειμενική συνάρτηση : Αντιπροσωπεύει το κριτήριο αξιολόγησης εναλλακτικών τρόπων δράσης από άποψη αξίας των μεταβλητών απόφασης έτσι ώστε να βελτιστοποιηθεί (ελαχιστοποιηθεί ή μεγιστοποιηθεί) η επιθυμητή απόδοση.
- Πολιτικές και περιορισμοί : Υπάρχουν ορισμένοι περιορισμοί στη χρήση των πόρων οι οποίοι προκύπτουν λόγω οργανωτικής πολιτικής, νομικών περιορισμών ή περιορισμένων πόρων όπως χώρος, χρήματα, ανθρώπινο δυναμικό, υλικό κλπ. Οι περιορισμοί στη χρήση των πόρων εκφράζονται είτε με τη μορφή εξίσωσης ή ανισότητας.
- Λειτουργικές σχέσεις : Σε ένα πρόβλημα απόφασης, οι μεταβλητές απόφασης μεταβάλλονται στην αντικειμενική συνάρτηση και οι περιορισμοί συνδέονται με μια συγκεκριμένη λειτουργική σχέση.
- Παράμετροι : Πρόκειται για σταθερές στις συναρτησιακές σχέσεις μεταξύ των μεταβλητών απόφασης. Οι παράμετροι μπορούν είτε να είναι ντετερμινιστικές είτε πιθανολογικές . Μια ντετερμινιστική παράμετρος είναι αυτή της οποίας η τιμή θεωρείται πως θα συμβεί με βεβαιότητα. Διαφορετικά, είναι πιθανολογική.

Βήμα 3 - (Επίλυση του μαθηματικού μοντέλου) : Μόλις το μαθηματικό μοντέλο του προβλήματος έχει διατυπωθεί, το επόμενο βήμα είναι να ληφθούν οι αριθμητικές τιμές των μεταβλητών απόφασης. Η λήψη αυτών των τιμών εξαρτάται από τη συγκεκριμένη μορφή ή τον τύπο του μαθηματικού μοντέλου. Σε γενικές γραμμές, χρησιμοποιούνται αυτές οι δυο μέθοδοι για την επίλυση ενός μοντέλου :

- Αναλυτικές μέθοδοι : Αυτές οι μέθοδοι εφαρμόζονται για την επίλυση τόσο περιορισμένων όσο και μη περιορισμένων μαθηματικών μοντέλων. Σε περιορισμένα προβλήματα, οι τιμές των μεταβλητών απόφασης ικανοποιούν όλους τους περιορισμούς και ταυτόχρονα παρέχουν μια βέλτιστη τιμή για την αντικειμενική συνάρτηση. Ωστόσο, σε προβλήματα χωρίς περιορισμούς, οι τιμές των μεταβλητών απόφασης παρέχουν μια αποδεκτή τιμή για την αντικειμενική συνάρτηση.
- Ευρετικές Μέθοδοι : Αυτές οι μέθοδοι χρησιμοποιούνται όταν η λήψη της βέλτιστης λύσης είναι είτε πολύ χρονοβόρα είτε το μοντέλο είναι πολύ περίπλοκο. Με τέτοιες μεθόδους θα ασχοληθούμε στην παρούσα εργασία.

Βήμα 4 - (Επικύρωση της λύσης) : Μετά την επίλυση του μαθηματικού μοντέλου, είναι σημαντικό να επανεξεταστεί η λύση για να δούμε αν η τιμή των μεταβλητών έχει νόημα και ότι οι αποφάσεις που προκύπτουν μπορούν να εφαρμοστούν. Μερικοί από τους λόγους για τους οποίους είναι απαραίτητη η επικύρωση της λύσης είναι:

- Το μαθηματικό μοντέλο ίσως να μην έχει λάβει υπόψη όλους τους περιορισμούς που υπάρχουν στη μελέτη επειδή μπορεί είτε να παραλειφθούν είτε να είναι εσφαλμένοι.
- Ορισμένες πτυχές του προβλήματος ενδέχεται να έχουν παραληφθεί ή απλοποιηθεί.
- Τα δεδομένα ενδέχεται να έχουν εκτιμηθεί ή καταγραφεί λανθασμένα.

Βήμα 5 - (Εφαρμογή της λύσης) : Πριν από την εφαρμογή της λύσης, ο υπεύθυνος της λήψης αποφάσεων θα πρέπει να διασφαλίσει την επιλογή εναλλακτικών λύσεων που μπορούν να εφαρμοστούν. Επιπλέον, είναι σημαντικό να διασφαλιστεί ότι κάθε λύση που εφαρμόζεται επανεξετάζεται και ενημερώνεται συνεχώς υπό το πρίσμα ενός μεταβαλλόμενου περιβάλλοντος.

Βήμα 6 - (Τροποποίηση του μοντέλου) : Για να είναι χρήσιμο ένα μαθηματικό μοντέλο, ο βαθμός στον οποίο πραγματικά αντιπροσωπεύει το σύστημα ή το πρόβλημα που μοντελοποιείται πρέπει να διερευνηθεί. Εάν κατά την επικύρωση η λύση δεν μπορεί να εφαρμοστεί τότε πρέπει να εντοπιστούν οι περιορισμοί που παραλείφθηκαν ή ήταν λάθος κατά την διατύπωση του μοντέλου. Σε όλες αυτές τις περιπτώσεις, πρέπει να γίνει επιστροφή στην διατύπωση του προβλήματος και να γίνουν προσεκτικά οι κατάλληλες τροποποιήσεις για να αναπαρασταθεί το δεδομένο πρόβλημα με μεγαλύτερη ακρίβεια. Ένα μοντέλο πρέπει να είναι εφαρμόσιμο για εύλογο χρονικό διάστημα και θα πρέπει να ενημερώνεται από καιρό σε καιρό, λαμβάνοντας υπόψη τις παρελθούσες, τις παρούσες και τις μελλοντικές πτυχές του προβλήματος.

Βήμα 7 - (Καθιέρωση ελέγχων στη λύση) : Οι αλλαγές στο περιβάλλον (κοινωνία ή επιχειρήσεις) επηρεάζουν τη συνεχιζόμενη εγκυρότητα των μοντέλων και των λύσεων τους. Επομένως, χρειάζεται να αναπτυχθεί μια διαδικασία ελέγχου για τον εντοπισμό σημαντικών αλλαγών στις μεταβλητές απόφασης του προβλήματος, χωρίς να χρειάζεται να δημιουργείται ένα νέο μοντέλο κάθε φορά που συμβαίνει μια σημαντική αλλαγή.

1.3 Γραμμικός Προγραμματισμός και Μέθοδοι Επίλυσης

Ο γραμμικός προγραμματισμός ασχολείται κυρίως με προβλήματα κατανομής, δηλαδή στοχεύει στην κατανομή των διαθέσιμων πόρων σε εργασίες. Το αντικείμενο αυτών των προβλημάτων είναι η βελτιστοποίηση της συνολικής αποδοτικότητας, η οποία μπορεί να είναι είτε η ελαχιστοποίηση του συνολικού κόστους είτε η μεγιστοποίηση του συνολικού κέρδους. Τα προβλήματα αυτά απαιτούν την ύπαρξη γραμμικών σχέσεων (εξισώσεων ή ανισοτήτων) τόσο για τη αντικειμενική συνάρτηση όσο και για τους επιβαλλόμενους περιορισμούς. Σήμερα, η επίλυση των προβλημάτων γραμμικού προγραμματισμού θεωρείται μια εύκολη διαδικασία, τόσο λόγω των τεχνικών επίλυσης που έχουν αναπτυχθεί, όσο και λόγω της υπολογιστικής δύναμης των σημερινών υπολογιστικών συστημάτων. Η μέθοδος Simplex αποτελεί πλέον την πιο διαδεδομένη μέθοδο για την επίλυση τέτοιου είδους προβλημάτων. Η μέθοδος αυτή αναπτύχθηκε από τον Dantzig το 1947 [8] και πλέον είναι από τις πιο γνωστές μεθόδους επίλυσης γραμμικών προβλημάτων [9]. Αυτό συμβαίνει, καθώς είναι ιδιαίτερα αποδοτική για την επίλυση πολύπλοκων μοντέλων, αφού υπάρχει πλήθος υπολογιστικών πακέτων Γραμμικού Προγραμματισμού με βάση αυτήν ή παραλλαγές της (π.χ. Dual Simplex). Ο αλγόριθμος Simplex αν και ιδιαίτερα αποδοτικός στην πράξη μπορεί σε συγκεκριμένες περιπτώσεις να μη λύνει τα αντίστοιχα προβλήματα σε πολυωνυμικό χρόνο. Πάραυτα έχουν αναπτυχθεί και άλλοι αλγόριθμοι, οι οποίοι επιλύουν τα προβλήματα γραμμικού προγραμματισμού σε πολυωνυμικό χρόνο [10].

1.4 Γραμμικός Ακέραιος Προγραμματισμός και Μέθοδοι Επίλυσης

Οι εφαρμογές του ακέραιου γραμμικού προγραμματισμού (Integer Linear Programming (ILP)) εμπίπτουν γενικά σε δύο κατηγορίες [5]:

- Κατηγορία των άμεσων εφαρμογών, όπου η φύση του προβλήματος καθιστά αδύνατη την εκχώρηση συνεχών τιμών στις μεταβλητές του μοντέλου.
- Μετασχηματισμένη κατηγορία, στην οποία χρησιμοποιούνται βοηθητικές ακέραιες μεταβλητές για το μετασχηματισμό αναλυτικώς 'δυσεπίλυτων ή δυσχειρίσιμων' προβλημάτων σε μοντέλα που μπορούν να επιλυθούν μέσω διαθέσιμων αλγορίθμων βελτιστοποίησης.

Ένα πρόβλημα ορίζεται ως αμιγώς ακέραιο όταν όλες οι μεταβλητές είναι ακέραιες. Σε αντίθετη περίπτωση, όπου περιλαμβάνει ακέραιες και συνεχείς μεταβλητές, το πρόβλημα χαρακτηρίζεται ως μικτό ακέραιο.

Γενικά οι αλγόριθμοι για προβλήματα ILP εκμεταλλεύονται την τεράστια υπολογιστική επιτυχία του γραμμικού προγραμματισμού. Για να γίνει κάτι τέτοιο εφικτό, το αρχικό πρόβλημα μετατρέπεται κατάλληλα. Οι δυο πιο διαδεδομένες μέθοδοι που χρησιμοποιούνται σε προβλήματα ILP είναι:

- Η μέθοδος διακλάδωσης και φραγής (Branch – and – Bound) [11]
- Η μέθοδος των τεμνόντων επιπέδων (Cutting – Plane) [12]

Καμία από τις δύο μεθόδους δεν εγγυάται σταθερή αποτελεσματικότητα. Ωστόσο, γενικά η μέθοδος διακλάδωσης και φραγής είναι πιο επιτυχημένη σε σχέση με τη μέθοδο των τεμνόντων επιπέδων για την επίλυση των ανωτέρω προβλημάτων [10].

1.5 Μη Γραμμικός Προγραμματισμός και Μέθοδοι Επίλυσης

Τα προβλήματα μη γραμμικού προγραμματισμού διατηρούν τη δομή αυτών του γραμμικού αφού και αυτά αποτελούνται από την αντικειμενική συνάρτηση και τους περιορισμούς. Η διαφορά έγκειται στο γεγονός ότι το μη γραμμικό πρόβλημα έχει τουλάχιστον μία μη γραμμική σχέση μεταξύ των μεταβλητών, η οποία μπορεί να εμφανίζεται στην αντικειμενική συνάρτηση ή/και στους περιορισμούς. Τα περισσότερα συστήματα που απαντώνται στη φύση είναι μη γραμμικά επομένως είναι σημαντικό οι αλγόριθμοι βελτιστοποίησης να μπορούν να διαχειριστούν και μη γραμμικά προβλήματα. Το πρόβλημα όμως προκύπτει από το γεγονός ότι αυτά τα μοντέλα μη γραμμικού προγραμματισμού είναι πολύ πιο δύσκολο να επιλυθούν. Αυτό οφείλεται στους εξής λόγους [13]:

- Είναι δύσκολος ο διαχωρισμός ενός τοπικού βέλτιστου σημείου από ένα ολικό.
- Τα βέλτιστα σημεία δεν περιορίζονται σε ακραία σημεία.
- Μπορεί να υπάρχουν πολλές, μη συνδεδεμένες μεταξύ τους, περιοχές εφικτών λύσεων.

- Διαφορετικά σημεία εκκίνησης μπορεί να δώσουν διαφορετικό αποτέλεσμα.
- Η εύρεση εφικτού σημείου εκκίνησης μπορεί να είναι δύσκολη.
- Οι περιορισμοί ισότητας είναι δύσκολο να ικανοποιηθούν.
- Δεν υπάρχει σαφής προσδιορισμός του αποτελέσματος.
- Υπάρχει πάρα πολλή διαθέσιμη θεωρία γύρω από αυτούς τους αλγόριθμους, αλλά μόνο λίγοι αλγόριθμοι επίλυσης.
- Είναι δύσκολο να προσδιοριστεί αν οι συνθήκες που εφαρμόζουμε σε έναν συγκεκριμένο solver ικανοποιούνται.
- Διαφορετικοί αλγόριθμοι και solvers καταλήγουν σε διαφορετικές λύσεις για τις ίδιες αρχικές συνθήκες.
- Διαφορετικές αλλά ισοδύναμες μοντελοποιήσεις του προβλήματος μπορεί να παράγουν διαφορετικό αποτέλεσμα ακόμη και με τη χρήση του ίδιου solver.
- Η χρήση των διαθέσιμων solver μπορεί να είναι περίπλοκη.

Υπάρχουν πολλοί τύποι προβλημάτων μη γραμμικού προγραμματισμού και σε αντίθεση με τα γραμμικά, αυτά δεν είναι δυνατό να επιλυθούν με έναν συγκεκριμένο αλγόριθμο. Παρακάτω αναφέρουμε κάποιους από τους τύπους αυτών των προβλημάτων:

- Προβλήματα χωρίς περιορισμούς (Unconstrained Optimization), όπου αποτελούνται μόνο από την αντικειμενική συνάρτηση.
- Προβλήματα που έχουν μη γραμμική αντικειμενική συνάρτηση, αλλά γραμμικούς περιορισμούς (Linearly Constrained Optimization).
- Τετραγωνικά προβλήματα (Quadratic Programming), όπου οι περιορισμοί είναι και πάλι γραμμικοί, αλλά η αντικειμενική συνάρτηση πρέπει να είναι τετραγωνική. Στην παρούσα διπλωματική θα ασχοληθούμε με ένα πρόβλημα που ανήκει σε αυτό το είδος.
- Προβλήματα, που επιλύονται με κυρτό προγραμματισμό (Convex Programming). Σε αυτά είναι ένα μεγάλο εύρος προβλημάτων τα οποία περιλαμβάνουν και την ειδική περίπτωση εκείνων όπου έχουν κυρτή αντικειμενική συνάρτηση καθώς και περιορισμούς.
- Προβλήματα τα οποία έχουν συναρτήσεις που είναι διαχωρίσιμες, δηλαδή μπορούν να εκφραστούν ως άθροισμα συναρτήσεων μιας μεταβλητή. Αυτά επιλύονται με την χρήση διαχωρίσιμου προγραμματισμού (Seperable Programming).
- Μη κυρτά προβλήματα που δεν καλύπτουν τους περιορισμούς του κυρτού προγραμματισμού (Nonconvex Programming).
- Γεωμετρικά προβλήματα (Geometric Programming), όπου η αντικειμενική συνάρτηση και οι περιορισμοί είναι γεωμετρικές ακολουθίες.
- Προβλήματα όπου η αντικειμενική συνάρτηση αποτελεί κλάσμα δύο άλλων συναρτήσεων. Αυτά τα προβλήματα λύνονται με την βοήθεια του κλασματικού προγραμματισμού (Fractional Programming) [10].

Κεφάλαιο 2

QUADRATIC ASSIGNMENT PROBLEM

2.1 Εισαγωγή Στο Quadratic Assignment Problem

Ας θεωρήσουμε το πρόβλημα της κατανομής ενός συνόλου εγκαταστάσεων σε ένα σύνολο τοποθεσιών, με το κόστος να είναι συνάρτηση της απόστασης και της ροής μεταξύ των εγκαταστάσεων, συν το κόστος που σχετίζεται με την τοποθέτηση μιας εγκατάσταση σε μια συγκεκριμένη τοποθεσία. Ο στόχος είναι να εκχωρηθεί κάθε εγκατάσταση σε θέση τέτοια ώστε να ελαχιστοποιείται το συνολικό κόστος. Αυτό είναι το Quadratic Assignment Problem (QAP) και εισήχθη από τους Koopmans και Beckmann το 1957 ως μαθηματικό μοντέλο για την κατανομή ενός συνόλου αδιαίρετων οικονομικών δραστηριοτήτων. Από τότε έχει εμφανιστεί σε διάφορες πρακτικές εφαρμογές. Ενδεικτικά κάποιες από τις εφαρμογές στις οποίες μπορούμε να το συναντήσουμε πλέον είναι η διάταξη εγκαταστάσεων, οι καλωδιώσεις ταμπλό, η κατασκευή υπολογιστών, ο προγραμματισμός, η επεξεργασία επικοινωνιών, η εξισορρόπηση τουρμπίνων και πολλά άλλα [14].

Από την πρώτη του διατύπωση, το QAP "τράβηξε" την προσοχή των ερευνητών σε όλο τον κόσμο, όχι μόνο λόγω της πρακτικής και θεωρητικής σημασίας του, αλλά και λόγω της πολυπλοκότητας του. Είναι ένα από τα πιο δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης: γενικά, περιπτώσεις τάξης $n > 30$ δεν μπορούν να λυθούν σε εύλογο χρόνο. Οι Sahni και Gonzales (1976) είχαν δείξει ότι το QAP είναι NP-δύσκολο και ότι ακόμη και η εύρεση μιας κατά προσέγγιση λύσης μέσα σε κάποιο σταθερό παράγοντα από τη βέλτιστη λύση δεν μπορεί να γίνει σε πολυωνυμικό χρόνο εκτός εάν $P = NP$. Υπάρχουν αρκετά NP-δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης τα οποία μπορούν να διατυπωθούν ως QAP. Κάποια από αυτά είναι το traveling salesman problem, το bin-packing problem, το max clique problem και το isomorphism of graphs problem [15].

2.2 Διατυπώσεις Του Quadratic Assignment Problem

Για πολλά προβλήματα συνδυαστικής βελτιστοποίησης, υπάρχουν διαφορετικές αλλά ισοδύναμες μαθηματικές διατυπώσεις, οι οποίες υπογραμμίζουν διαφορετικά δομικά χαρακτηριστικά του προβλήματος. Αυτές οι διατυπώσεις μπορεί να οδηγήσουν σε διαφορετικές προσεγγίσεις επίλυσης. Αρκετές διατυπώσεις υπάρχουν για το Quadratic Assignment Problem. Θα αναφέρουμε τις πιο γνωστές και στην συνέχεια θα μιλήσουμε αναλυτικά για κάποιες. Οι πιο γνωστές διατυπώσεις είναι [14, 15]:

- Quadratic Integer Program Διατύπωση.
- Trace Διατύπωση.
- Concave Quadratic Διατύπωση.
- Graph Διατύπωση.
- Semidefinite Programming Relaxation Διατύπωση.

Όπως αναφέραμε και πριν το QAP είναι το πρόβλημα της κατανομής ενός συνόλου εγκαταστάσεων σε ένα σύνολο τοποθεσιών, με το κόστος να είναι συνάρτηση της απόστασης και της ροής μεταξύ των εγκαταστάσεων, συν το κόστος που σχετίζεται με την τοποθέτηση μιας εγκατάστασης σε μια συγκεκριμένη τοποθεσία. Ο στόχος είναι να εκχωρηθεί κάθε εγκατάσταση σε μια τοποθεσία έτσι ώστε το συνολικό κόστος να ελαχιστοποιηθεί. Συγκεκριμένα, μας δίνονται τρεις πίνακες εισόδου ($n \times n$) με πραγματικά στοιχεία $F = (f_{ij})$, $D = (d_{kl})$, και $B = (b_{ik})$, όπου f_{ij} είναι η ροή μεταξύ της εγκατάστασης i και της εγκατάστασης j , d_{kl} είναι η απόσταση μεταξύ της τοποθεσίας k και της τοποθεσίας l , και b_{ik} είναι το κόστος τοποθέτησης της εγκατάστασης i στην τοποθεσία k . Η εκδοχή του QAP από τους Koopmans-Beckmann μπορεί να διατυπωθεί ως εξής: Έστω n ο αριθμός των εγκαταστάσεων και τοποθεσιών και συμβολίζουμε με N το σύνολο $N = \{1, 2, \dots, n\}$.

$$\min_{\phi \in S_n} \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\phi(i)\phi(j)} + \sum_{i=1}^n b_{i\phi(i)} \quad (2.1)$$

όπου S_n είναι το σύνολο όλων των μεταθέσεων $\phi : N \rightarrow N$. Κάθε επιμέρους γινόμενο $f_{ij} d_{\phi(i)\phi(j)}$ είναι το κόστος ανάθεσης της εγκατάστασης i στην τοποθεσία $\phi(i)$ και της εγκατάστασης j στην τοποθεσία $\phi(j)$. Στο πλαίσιο της τοποθέτησης εγκαταστάσεων, οι πίνακες F και D είναι συμμετρικοί με μηδενικά στην διαγώνιο, και όλοι οι πίνακες είναι μη αρνητικοί. Ένα παράδειγμα του QAP με πίνακες εισόδου F , D , και B θα συμβολίζεται με $\text{QAP}(F, D, B)$, ενώ θα συμβολίζεται με $\text{QAP}(F, D)$ αν δεν υπάρχει γραμμικός όρος (δηλ., $B = 0$).

Μια πιο γενική εκδοχή του QAP εισήχθη από τον Lawler. Σε αυτήν την εκδοχή, έχουμε μια τετραδιάστατη διάταξη $C = (c_{ijkl})$ συντελεστών αντί των δύο πινάκων F και D , και το πρόβλημα μπορεί να διατυπωθεί ως εξής:

$$\min_{\phi \in S_n} \sum_{i=1}^n \sum_{j=1}^n c_{ij\phi(i)\phi(j)} + \sum_{i=1}^n b_{i\phi(i)} \quad (2.2)$$

Σαφώς, ένα πρόβλημα Koopmans-Beckmann $\text{QAP}(F, D, B)$ μπορεί να διατυπωθεί ως QAP του Lawler ορίζοντας $c_{ijkl} = f_{ij}d_{kl}$ για όλα τα i, j, k, l με $i \neq j$ ή $k \neq l$ και $c_{iikl} = f_{ii}d_{kl} + b_{ik}$ αλλιώς.

Επιπλέον ας κάνουμε την παρατήρηση ότι κάθε μετάθεση ϕ του συνόλου $N = \{1, 2, \dots, n\}$ μπορεί να αναπαρασταθεί από έναν πίνακα $(n \times n)$, $X = (x_{ij})$ τέτοιο ώστε:

$$x_{ij} = \begin{cases} 1 & \text{εάν } \phi(i) = j, \\ 0 & \text{αλλιώς.} \end{cases}$$

Ο πίνακας X ονομάζεται *πίνακας μεταθέσεων* και χαρακτηρίζεται από τους εξής περιορισμούς ανάθεσης:

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= 1, \quad j = 1, 2, \dots, n, \\ \sum_{j=1}^n x_{ij} &= 1, \quad i = 1, 2, \dots, n, \\ x_{ij} &\in \{0, 1\}, \quad i, j = 1, 2, \dots, n. \end{aligned}$$

Ορίζουμε το σύνολο όλων των πινάκων μεταθέσεων ως X_n . Σύμφωνα με ένα διάσημο θεώρημα του Birkhoff, οι πίνακες μεταθέσεων αντιστοιχούν με μοναδικό τρόπο στις κορυφές του πολυέδρου ανάθεσης. Αυτό οδηγεί στην ακόλουθη περιγραφή ενός Προβλήματος Τετραγωνικής Ανάθεσης (QAP) ως τετραγωνικό ακέραιο πρόγραμμα.

2.2.1 Quadratic Integer Program Διατύπωση

Χρησιμοποιώντας πίνακες μεταθέσεων αντί για μεταθέσεις, το QAP (2.2) μπορεί να διατυπωθεί ως το ακόλουθο ακέραιο πρόγραμμα με τετραγωνική αντικειμενική συνάρτηση [16]:

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ijkl} x_{ik} x_{jl} + \sum_{i,j=1}^n b_{ij} x_{ij} \quad (2.3)$$

υπό τους περιορισμούς

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n, \quad (2.4)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (2.5)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n. \quad (2.6)$$

Από εδώ και στο εξής, όποτε γράφουμε $(x_{ij}) \in X_n$, θα υπονοείται ότι τα x_{ij} ικανοποιούν τους περιορισμούς ανάθεσης (2.4), (2.5) και (2.6).

Πολλοί συγγραφείς έχουν προτείνει μεθόδους για τη γραμμικοποίηση της τετραγωνικής μορφής της αντικειμενικής συνάρτησης (2.3) με την εισαγωγή επιπλέον μεταβλητών.

Ένα QAP σε μορφή Koormans-Beckmann μπορεί να διατυπωθεί με έναν πιο συμπαγή τρόπο εάν ορίσουμε ένα εσωτερικό γινόμενο μεταξύ πινάκων. Έστω το εσωτερικό γινόμενο δύο πραγματικών πινάκων $n \times n$ A, B ορίζεται ως

$$\langle A, B \rangle := \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}.$$

Δεδομένου ενός $(n \times n)$ πίνακα A , μιας μετάθεσης $\phi \in S_n$ και του αντίστοιχου πίνακα μεταθέσεων $X \in X_n$, τότε οι AX^T και XA αλλάζουν τις στήλες και τις γραμμές του A , αντίστοιχα, σύμφωνα με τη μετάθεση ϕ και επομένως

$$XAX^T = (a_{\phi(i)\phi(j)}).$$

Έτσι, μπορούμε να διατυπώσουμε ένα QAP Koormans-Beckmann εναλλακτικά ως

$$\min \langle F, XDX^T \rangle + \langle B, X \rangle \quad (2.7)$$

υπό τον περιορισμό $X \in X_n$.

2.2.2 Trace Διατύπωση

Το ίχνος ενός πίνακα $(n \times n)$ B ορίζεται ως το άθροισμα των στοιχείων της διαγωνίου του, δηλαδή

$$\text{tr}(B) := \sum_{i=1}^n b_{ii}.$$

Εξετάστε ένα παράδειγμα QAP Koormans-Beckmann με εισερχόμενους πίνακες F, D και B . Αφήνοντας $\bar{D} = XDX^T$, τότε

$$\text{tr}(F\bar{D}) = \sum_{i=1}^n \sum_{j=1}^n f_{ij} \bar{d}_{ji} = \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\phi(i)\phi(j)}, \quad (2.8)$$

εφόσον $\bar{d}_{ji} = d_{\phi(i)\phi(j)}$, $i, j = 1, \dots, n$, όπου $\phi \in S_n$ είναι η μετάθεση που σχετίζεται με τον X . Εφόσον $\text{tr}(BX^T) = \sum_{i=1}^n b_{i\phi(i)}$, το QAP στο (2.7) μπορεί να διατυπωθεί ως

$$\min(\text{tr}(FXD^T + B)X^T) \quad (2.9)$$

με περιορισμό $X \in X_n$.

Η Trace διατύπωση του QAP εμφανίστηκε πρώτα στον Edwards [[17], [18]], και χρησιμοποιήθηκε από τους Finke, Burkard, και Rendl [19] για την εισαγωγή τεχνικών κατώτερου φράγματος με ιδιοτιμές για συμμετρικά QAP. Δεδομένων δύο πραγματικών πινάκων $n \times n$ A, B , θυμηθείτε τις γνωστές ιδιότητες $\text{tr}(AB) = \text{tr}(BA)$, $(AB)^T = B^T A^T$, και $\text{tr}A = \text{tr}A^T$. Για $F = F^T$, μπορούμε τότε να γράψουμε τον τετραγωνικό όρο στο (2.9) ως

$$\text{tr}(FXD^T X^T) = \text{tr}(FXDX^T),$$

όπου το D δεν είναι απαραίτητα συμμετρικό. Συνεπώς, δεδομένου ενός παραδείγματος QAP όπου μόνο ένας από τους πίνακες είναι συμμετρικός (ας πούμε F), μπορούμε να το μετατρέψουμε σε ένα παράδειγμα QAP όπου και οι δύο πίνακες είναι συμμετρικοί. Αυτό γίνεται με την εισαγωγή ενός νέου συμμετρικού πίνακα $E = \frac{1}{2}(D + D^T)$:

$$\text{tr}(FXEX^T) = \frac{1}{2} \text{tr}(FXD^T X^T + FXDX^T) = \text{tr}(FXD^T X^T).$$

2.3 Κάτω Όρια

Η μελέτη των κάτω ορίων είναι πολύ σημαντική για την ανάπτυξη αλγορίθμων με σκοπό την επίλυση μαθηματικών προβλημάτων προγραμματισμού και συνδυαστικής βελτιστοποίησης. Γενικά, οι ακριβείς μέθοδοι χρησιμοποιούν έμμεση απαρίθμηση, σε μια προσπάθεια να εξασφαλιστεί η βέλτιστη λύση και, ταυτόχρονα, να αποφευχθεί η συνολική απαρίθμηση όλων των εφικτές λύσεις. Χρησιμοποιώντας λοιπόν τα κάτω όρια είναι δυνατή η προσέγγιση της βέλτιστης λύσης. Μια πιο ειδική χρήση των τεχνικών των κάτω ορίων αφορά την αξιολόγηση της απόδοσης των ευρετικών αλγορίθμων παρέχοντας ένα σχετικό μέτρο της εγγύτητας της υποβέλτιστης λύσης στο βέλτιστο. Κατά τη σύγκριση των κατώτατων ορίων, θα πρέπει να ληφθούν υπόψη τα ακόλουθα κριτήρια:

- Πολυπλοκότητα υπολογισμού του κάτω ορίου.
- "Στενότητα" του κάτω ορίου (δηλαδή, «μικρό» κενό μεταξύ του ορίου και της βέλτιστης λύσης).
- Αποδοτικότητα στον υπολογισμό των κάτω ορίων για υποσύνολα του αρχικού συνόλου.

Δεδομένου ότι δεν υπάρχει σαφής κατάταξη της απόδοσης των κατώτερων ορίων που αναφέρονται παρακάτω, όλα τα παραπάνω κριτήρια θα πρέπει να ληφθούν υπόψη κατά την επιλογή κατώτατου ορίου. Επιπλέον λαμβάνοντας υπόψη την ασυμπτωτική συμπεριφορά του QAP θα πρέπει να υποθέσουμε ότι η "στενότητα" του κάτω ορίου είναι η σημαντικότερη από τα παραπάνω κριτήρια. Με άλλα λόγια, εάν υπάρχει μεγάλος αριθμός εφικτών λύσεων κοντά στο βέλτιστο, τότε ένα κάτω όριο που δεν είναι αρκετά "στενό", θα αποτύχει να εξαλείψει έναν μεγάλο αριθμό υποπροβλημάτων. Στην συνέχεια αναφέρονται κάποια είδη κάτω ορίων [14, 15]:

- Gilmore-Lawler Κάτω Όρια.
- Όρια βασισμένα σε Linear Programming Relaxations.
- Κάτω όρια βασισμένα σε Variance Reduction.
- Κάτω όρια βασισμένα σε Eigenvalue.
- Όρια βασισμένα σε Semidefinite Relaxations.
- Όρια βασισμένα σε Interior Points Methods.
- Όρια βασισμένα σε Graph Formulation.
- Spectral Όρια.

2.4 Μέθοδοι Επίλυσης

Οι μέθοδοι που χρησιμοποιούνται σε προβλήματα συνδυαστικής βελτιστοποίησης μπορούν να είναι είτε ακριβείς είτε ευρετικές. Παρακάτω θα αναφέρουμε κάποιες από αυτές.

2.4.1 Ακριβείς Μέθοδοι

Οι ακριβείς μέθοδοι που χρησιμοποιούνται για την επίτευξη ενός παγκόσμιου βέλτιστου για το QAP περιλαμβάνουν τις Branch-and-Bound, Cutting-Planes [15], [14] ή συνδυασμούς αυτών των μεθόδων όπως Branch-and-Cut και Dynamic Programming [15]. Οι Branch-And-Bound είναι οι πιο γνωστοί και χρησιμοποιούμενοι αλγόριθμοι και ορίζονται από κανόνες κατανομής και κοπής, οι οποίοι ορίζουν τα κάτω όρια για το QAP. Τα τελευταία χρόνια, οι διαδικασίες που συνδυάζουν Branch-And-Bound με παράλληλη υλοποίηση χρησιμοποιούνται ευρέως. Εξαιτίας τους έχουν επιτευχθεί τα καλύτερα αποτελέσματα για το QAP. Ωστόσο, είναι σημαντικό να παρατηρήσουμε ότι η επιτυχία αυτή σχετίζεται επίσης και με τις τεχνολογικές βελτιώσεις του hardware.

2.4.2 Ευρετικές Μεθόδους

Αν και έχουν γίνει ουσιαστικές βελτιώσεις στην ανάπτυξη ακριβών αλγορίθμων για το QAP, προβλήματα διάστασης $n > 20$ δεν είναι ακόμα πρακτικά να λυθούν λόγω των υψηλών απαιτήσεων χρόνου από τον υπολογιστή. Αυτό έκανε αναγκαία την ανάπτυξη των ευρετικών αλγορίθμων που παρέχουν λύσεις καλής ποιότητας σε εύλογο χρόνο. Πολλές έρευνες έχουν αφιερωθεί στην ανάπτυξη τέτοιων προσεγγίσεων. Γενικά οι ευρετικές μέθοδοι χωρίζονται σε τρεις κατηγορίες:

- Μεθόδους απαρίθμησης : Αυτές οι μέθοδοι επιλέγουν μια λύση σε κάθε κύκλο και "δοκιμάζουν" για να δουν αν είναι η βέλτιστη.
- Κατασκευαστικές Μεθόδους : Σε αυτές τις μεθόδους η τελική λύση κατασκευάζεται σταδιακά επιλέγοντας με κάποιο τρόπο κάθε φορά ένα μέρος της τελικής λύσης.
- Μεθόδους Βελτίωσης : Αυτές οι μέθοδοι χρησιμοποιούν αλγορίθμους τοπικής εύρεσης για να βελτιώσουν μια λύση που έχει προ επιλεγεί.

Στην εργασία αυτή θα χρησιμοποιήσουμε ευρετικές μεθόδους για να αντιμετωπίσουμε το πρόβλημα που θα εξετάσουμε. Επομένως θα δούμε αναλυτικά στα επόμενα κεφάλαια κάποιες ευρετικές μεθόδους. Πάραυτα θα αναφέρουμε ενδεικτικά κάποιες από τις πιο γνωστές ευρετικές μεθόδους [14, 15]:

- Tabu Search (TS).
- Simulated Annealing (SA)
- Genetic Algorithms (GA).
- Scatter Search
- Greedy Randomized Adaptive Search Procedure (GRASP).
- Variable Neighborhood Search (VNS)
- Ant Systems (AS).

Κεφάλαιο 3

ΤΟ ΠΡΟΒΛΗΜΑ ΚΑΤΑΝΑΛΩΣΗΣ ΙΣΧΥΟΣ ΣΕ DEEP SUBMICRON TECHNOLOGY BUSES ΩΣ QAP

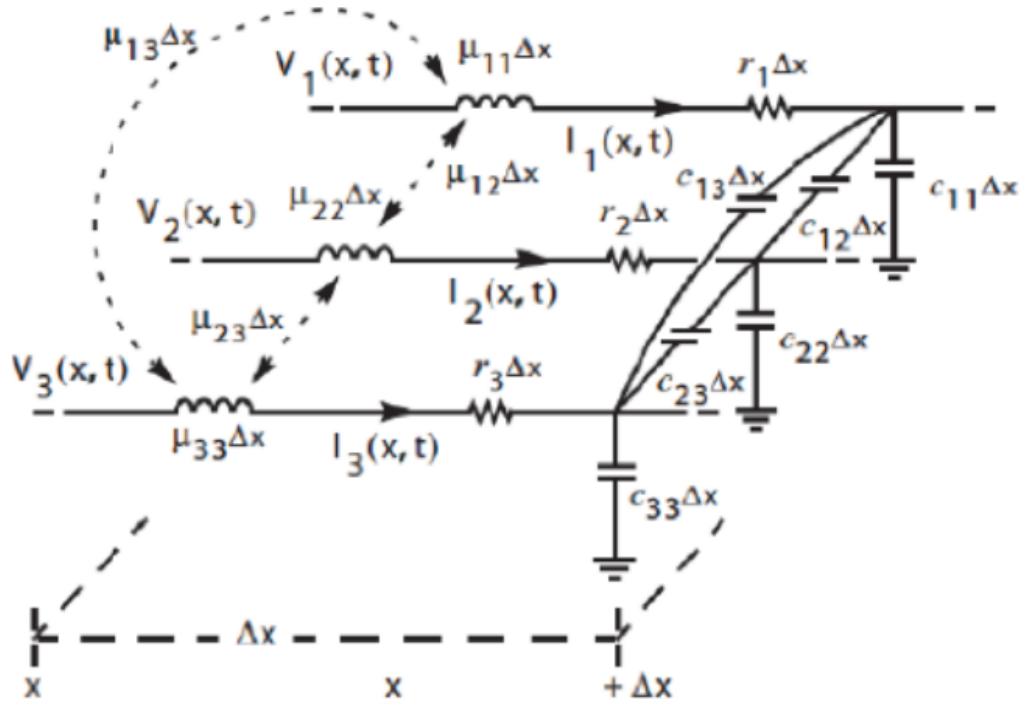
3.1 Εισαγωγή Στο Πρόβλημα

Η ελαχιστοποίηση της κατανάλωσης ισχύος σε deep submicron διασυνδέσεις έχει γίνει απαραίτητη λόγω της αυξανόμενης διάχυσης ενέργειας από τα παρασιτικά φαινόμενα που εμφανίζονται σε μικρές κλίμακες. Όσο οι διαστάσεις των χαρακτηριστικών μειώνονται, οι απώλειες ισχύος από τα ρεύματα διαρροής και οι συζεύξεις των διασυνδέσεων αυξάνονται απότομα. Παρόλο που υψηλότερες πυκνότητες και μεγαλύτερες ταχύτητες είναι εφικτές, οι τεχνικές για τον περιορισμό της περιττής κατανάλωσης ενέργειας είναι απαραίτητες για την βιώσιμη πρόοδο της τεχνολογίας αυτής. Με αφορμή το πρόβλημα αυτό έχει προταθεί μια λύση η οποία βασίζεται στην αναδιάταξη καλωδίων στους διαύλους των deep submicron τεχνολογιών [20, 21]. Καθώς οι δίαυλοι αποτελούν ένα μεγάλο τμήμα του chip και οι διασυνδέσεις καταλαμβάνουν ένα μεγάλο ποσοστό της επιφάνειας του, οι δίαυλοι έχουν αναδειχθεί ως ένας βασικός συντελεστής στη συνολική διάχυση ενέργειας.

Σε αυτή την εργασία θα εξετάσουμε το πρόβλημα της αναδιάταξης καλωδίων ως ένα QAP και θα προσπαθήσουμε να το αντιμετωπίσουμε με μεθόδους επίλυσης των QAP. Όπως είδαμε και στο προηγούμενο κεφάλαιο τα QAP είναι προβλήματα που αφορούν την κατανομή εγκαταστάσεων σε συγκεκριμένες θέσεις με τον βέλτιστο τρόπο, όπως για παράδειγμα η κατανομή τμημάτων σε κτίρια λαμβάνοντας υπόψη αποστάσεις και αλληλεπιδράσεις μεταξύ τμημάτων. Είναι από τις υπολογιστικά πιο δύσκολες κατηγορίες βελτιστοποίησης. Θα δείξουμε αρχικά πως η αναδιάταξη καλωδίων μπορεί να διατυπωθεί σαν ένα QAP και στην συνέχεια θα χρησιμοποιήσουμε γνωστούς αλγόριθμους, οι οποίοι έχουν σχεδιαστεί για QAP, για την επίλυση του προβλήματος, δημιουργώντας διαμορφώσεις καλωδίων με αναδιάταξη. Θα δούμε πως αυτές οι διαμορφώσεις μας παρέχουν μειώσεις ενέργειας σε σύγκριση με μια μη αναδιατεταγμένη βάση [22].

3.2 Το Μοντέλο Του Deep Submicron Bus

Στην deep submicron τεχνολογία, πολλοί παράγοντες συμβάλουν στην αλλαγή των χαρακτηριστικών των γραμμών. Αυτές οι τεχνολογίες παρουσιάζουν μικρότερα πλάτη άλλα μεγαλύτερες αναλογίες (ύψος/μήκος) και στις περισσότερες περιπτώσεις πλησιέστερες τοποθετήσεις γραμμών σε σύγκρισή με παλαιότερες τεχνολογίες. Επιπλέον, υπάρχει αύξηση στην επιφάνεια του chip, καθώς και αύξηση στην αναλογία του μήκους του διαύλου προς τη διατομή των γραμμών. Αυτοί οι παράγοντες δημιουργούν μεγαλύτερο αριθμό παρασιτικών στοιχείων σε σύγκριση με αυτά που υπάρχουν στη απλή διαμόρφωση η οποία είναι με γειωμένο πυκνωτή. Επομένως, ένα μοντέλο που ενσωματώνει τη χωρητική και επαγωγική σύζευξη μεταξύ των γραμμών είναι καλύτερο. Το μοντέλο που παρουσιάζεται παρακάτω έχει υιοθετηθεί ευρέως για την αξιολόγηση της καθυστέρησης και της ακεραιότητας του σήματος [23–26], χρησιμεύοντας ως απλοποιημένο ηλεκτρικό ισοδύναμο στην εκτίμηση των διαύλων deep submicron τεχνολογιών.



Σχήμα 3.1: Distributed μοντέλο των γραμμών ενός διαύλου (Συζευγμένες γραμμές μετάδοσης) [2].

Οι γραμμές είναι διατεταγμένες παράλληλα κατά μήκος του άξονα x και έχουν μήκος L . Έχουν αντίσταση, $r_i(x)$, όπου $i = 1, 2, \dots, n$. Η χωρητικότητα μεταξύ της γραμμής i και της γείωσης είναι $C_{ii}(x)$ και αυτή μεταξύ των γραμμών i και j είναι $C_{ij}(x)$. Επιπλέον, $\mu_{ii}(x)$ είναι η πυκνότητα της αυτεπαγωγής της γραμμής i και $\mu_{ij}(x)$ είναι η πυκνότητα της αμοιβαίας αυτεπαγωγής μεταξύ των γραμμών i και j . Οι πυκνότητες μπορεί να εξαρτώνται από το x . Λεπτομέρειες σχετικά με τα ηλεκτρικά χαρακτηριστικά των γραμμών του δίαυλου και το γενικό ενεργειακό μοντέλο των DSM διαύλων μπορούν να βρεθούν στις [27–33].

Εμείς θα εξετάσουμε τον δίαυλο κατά τη διάρκεια του κύκλου ρολογιού $[0, T]$, όπου T είναι η περίοδος του ρολογιού του δίαυλου. Ας υποθέσουμε ότι $V_r^{(i)}$ και $V_r^{(f)}$ είναι οι τάσεις κατά μήκος της γραμμής i στην αρχή και στο τέλος του κύκλου αντίστοιχα, όπου $r = 1, 2, \dots, n$. Υποθέτουμε ότι το T είναι επαρκές ώστε οι τάσεις κατά μήκος των γραμμών να κατασταλάξουν στις τελικές τους τιμές. Το διάνυσμα των αρχικών τάσεων είναι $V^{(i)} = (V_1^{(i)}, V_2^{(i)}, \dots, V_n^{(i)})^\top$, και το διάνυσμα των τελικών τάσεων είναι $V^{(f)} = (V_1^{(f)}, V_2^{(f)}, \dots, V_n^{(f)})^\top$.

Όπως φαίνεται στις [20, 21], η ενέργεια που αντλείται από την πηγή ισχύος, V_{DD} , κατά τη διάρκεια του κύκλου, συμβολίζεται ως E_{VDD} , και είναι:

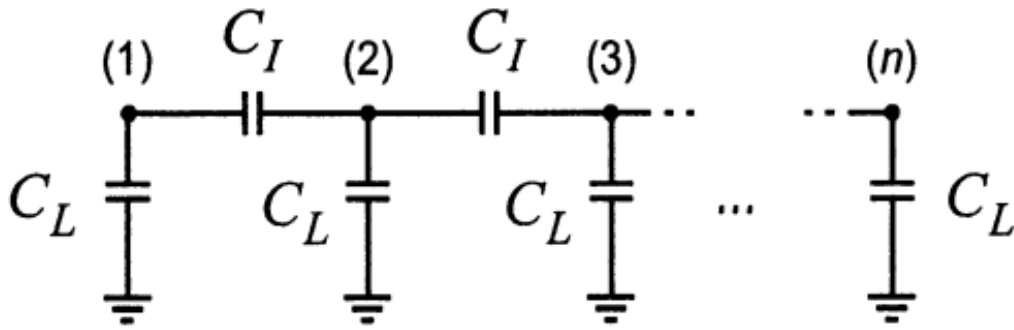
$$E_{VDD}(V^{(i)}, V^{(f)}) = (V^{(f)})^\top A(V^{(f)} - V^{(i)})$$

Ο πίνακας $A = [A_{ij}]_{i,j=1}^n$ δίνεται από:

$$A_{ij} = \begin{cases} \sum_{k=1}^n C_{ik}, & \text{εάν } i = j, \\ -C_{ij}, & \text{εάν } i \neq j, \end{cases} \quad (3.1)$$

όπου C_{ij} , για $i \neq j$, είναι η συνολική χωρητικότητα μεταξύ των γραμμών i και j του δίαυλου, και C_{ii} είναι η συνολική χωρητικότητα μεταξύ της γραμμής i και της γείωσης.

Σε πολλά DSM, η χωρητική σύζευξη μεταξύ μη γειτονικών γραμμών είναι πολύ ασθενής σε σύγκριση με αυτήν μεταξύ γειτονικών γραμμών. Επομένως, το γενικό μοντέλο μπορεί να απλοποιηθεί στο κύκλωμα που βλέπουμε στο (Σχήμα 3.2)



Σχήμα 3.2: Απλοποιημένο (προσεγγιστικό ενεργειακά) DSM μοντέλο [3].

όπου το C_I είναι η συνολική χωρητικότητα μεταξύ των γραμμών, και το C_L είναι η συνολική χωρητικότητα γραμμής-προς-γείωση. Για ευκολία, ορίζουμε τον παράγοντα [3]:

$$\beta = C_I/C_L \quad (3.2)$$

Στο απλοποιημένο μοντέλο ο πίνακας A ορίζεται ως εξής:

$$A = \begin{bmatrix} 1 + 2\beta & -\beta & 0 & \cdots & 0 & 0 \\ -\beta & 1 + 2\beta & -\beta & \cdots & 0 & 0 \\ 0 & -\beta & 1 + 2\beta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 + 2\beta & -\beta \\ 0 & 0 & 0 & \cdots & -\beta & 1 + 2\beta \end{bmatrix} \cdot C_L \quad (3.3)$$

Στην παρούσα εργασία θα χρησιμοποιηθεί το απλοποιημένο μοντέλο ως βάση για την επίλυση του προβλήματος [22]. Κατά συνέπεια το λ , C_L και ο πίνακας A θα είναι αυτοί που είδαμε παραπάνω (3.2),(3.3).

Σύμφωνα με την [21], η οποία επεκτείνει τον ορισμό της δραστηριότητας μετάβασης σε ένα στατιστικό μέτρο το οποίο μπορεί να εφαρμοστεί σε διαύλους με συζευγμένες γραμμές, θεωρούμε έναν δίαυλο με n γραμμές, όπως στο Σχήμα 3.1. Στην συνέχεια θεωρούμε ως $l_r(k)$ την δυαδική τιμή της γραμμής i στο τέλος του k -οστού κύκλου, όπου $r = 1, 2, \dots, n$. Τότε μπορούμε να ορίσουμε την ακολουθία των διανυσμάτων ως εξής:

$$L(k) = (l_1(k), l_2(k), \dots, l_n(k))^T, \quad k = 1, 2, \dots$$

τα οποία είναι τυχαία καθώς τα στοιχεία που εισάγονται είναι τυχαίες μεταβλητές. Επίσης, ορίζουμε ως $V(k) = (V_1(k), V_2(k), \dots, V_n(k))^T$ το διάνυσμα των τάσεων των γραμμών στο τέλος του k -οστού κύκλου. Όπως και πριν, έχουμε $V(k) = V_{dd}L(k)$. Επιπλέον, η ενέργεια που καταναλώθηκε κατά τη διάρκεια του k -οστού κύκλου δίνεται από:

$$E(k) = \frac{1}{2}(V(k) - V(k-1))^T A (V(k) - V(k-1)),$$

όπου ο A δίνεται από την (3.3). Εάν θεωρήσουμε πως οι ακολουθίες των bits $l_r(1), l_r(2), \dots, l_r(k)$ για $r = 1, 2, \dots, n$ είναι από κοινού στάσιμες με την ευρεία έννοια, τότε ο πίνακας αυτοσυσχέτισης, $R(r)$, της ακολουθίας διανυσμάτων $L(1), L(2), L(3), \dots$ μπορεί να οριστεί ως εξής:

$$R(r) = [R_{ij}(r)]_{i,j=1}^n = \overline{L(k+r) \cdot L^T(k)}$$

και για $i, j = 1, 2, \dots, n$

$$R_{ij}(r) = \overline{l_i(k+r)l_j(k)}.$$

Τότε, όπως φαίνεται στο [20], η ενέργεια που καταναλώνεται σε έναν δίαυλο με συζευγμένες γραμμές, κατά τη διάρκεια ενός κύκλου, είναι:

$$\overline{E} = V_{dd}^2 \cdot \text{trace}(A \cdot T^a), \quad (3.4)$$

όπου ο T^a ονομάζεται πίνακας μετάβασης του διαύλου και ορίζεται ως:

$$T^a = R(0) - \frac{1}{2}(R(1) + R^T(1)).$$

3.3 Η QAP Διατύπωση Της Μετάθεσης Γραμμών

Στο [34], παρουσιάζεται η σταθερή μετάθεση των bit ως μια προσέγγιση για τη μείωση της κατανάλωσης ενέργειας. Συγκεκριμένα, αντί να μεταδίδεται η ακολουθία των διανυσμάτων $\mathbf{L}(k) = (l_1(k), l_2(k), \dots, l_n(k))^T$, $k = 1, 2, \dots$, μπορεί να μεταδοθεί η ακολουθία διανυσμάτων με μετατεθειμένες εισόδους $\mathbf{L}'(k) = (l_{\pi(1)}(k), l_{\pi(2)}(k), \dots, l_{\pi(n)}(k))^T$, $k = 1, 2, \dots$ όπου το π είναι μια μετάθεση των δεικτών $1, 2, \dots, n$. Ο στόχος αυτής της προσέγγισης είναι να ελαχιστοποιηθεί η αντίθετη μετάβαση και να μεγιστοποιηθεί η ταυτόχρονη μετάβαση σε παρακείμενες γραμμές επιλέγοντας την κατάλληλη μετάθεση.

Χρησιμοποιώντας τη διατύπωση που αναφέρθηκε στο προηγούμενο κεφάλαιο, θα δείξουμε ότι το πρόβλημα της μετάθεσης των γραμμών μπορεί να θεωρηθεί και να επιλυθεί ως ένα QAP. Συγκεκριμένα, έστω Π ο πίνακας μετάθεσης ($n \times n$) που αντιστοιχεί στη μετάθεση π , τότε $\mathbf{L}_\pi(k) = \Pi \cdot \mathbf{L}(k)$ είναι ο πίνακας δραστηριότητας του αρχικού διαύλου. Έτσι, ο πίνακας δραστηριότητας μετάβασης του διαύλου με τα μετατεθειμένα bit είναι:

$$\mathbf{T}_\Pi^a = \Pi \cdot \mathbf{T}^a \cdot \Pi^T.$$

Χρησιμοποιώντας την (3.4), η αναμενόμενη κατανάλωση ενέργειας δίνεται από την έκφραση:

$$\overline{E}_\pi = V_{DD}^2 \cdot \text{trace}(\mathbf{A} \cdot \Pi \cdot \mathbf{T}^a \cdot \Pi^T). \quad (3.5)$$

Επομένως, η έκφραση που θέλουμε να ελαχιστοποιήσουμε σε συνάρτηση με τον πίνακά μετάθεσης Π , είναι η:

$$\min_{\Pi \in \Pi_n} V_{DD}^2 \cdot \text{trace}(\mathbf{A} \cdot \Pi \cdot \mathbf{T}^a \cdot \Pi^T), \quad (3.6)$$

όπου το Π_n είναι το σύνολο όλων των πινάκων μετάθεσης $n \times n$.

Θα συνδέσουμε τώρα το πρόβλημα μας με το QAP και πιο συγκεκριμένα με την trace διατύπωση του QAP [22]. Όπως αναφέραμε και στο προηγούμενο κεφάλαιο το QAP είναι το πρόβλημα της κατανομής ενός συνόλου εγκαταστάσεων σε ένα σύνολο τοποθεσιών, με το κόστος να είναι συνάρτηση της απόστασης και της ροής μεταξύ των εγκαταστάσεων, συν τα κόστη που σχετίζονται με την τοποθέτηση μιας εγκατάστασης σε μια συγκεκριμένη τοποθεσία. Ο στόχος είναι να ανατεθεί κάθε εγκατάσταση σε μια τοποθεσία έτσι ώστε το συνολικό κόστος να ελαχιστοποιηθεί [14]. Εμείς θα εξετάσουμε την πιο ειδική περίπτωση όπου, δεν υπάρχουν κόστη τα οποία σχετίζονται με την τοποθέτηση μιας εγκατάστασης σε μια συγκεκριμένη τοποθεσία. Αν υποθέσουμε ότι υπάρχουν n εγκαταστάσεις και n τοποθεσίες, τότε έχουμε δύο πίνακες εισόδου $n \times n$ με πραγματικά στοιχεία $\mathbf{F} = (f_{ij})$ και $\mathbf{D} = (d_{kl})$, όπου f_{ij} είναι η ροή μεταξύ της εγκατάστασης i και της εγκατάστασης j και d_{kl} είναι η απόσταση μεταξύ της τοποθεσίας k και της τοποθεσίας l . Είδαμε στο προηγούμενο κεφάλαιο πως η trace διατύπωση για το QAP έχει την ακόλουθη μορφή:

$$\min_{\Pi \in \Pi_n} \text{trace}(\mathbf{F} \cdot \Pi \cdot \mathbf{D}^T \cdot \Pi^T) \quad (3.7)$$

όπου με Π_n , όπως και στην (3.6), συμβολίζεται το σύνολο όλων των πινάκων μετάθεσης ($n \times n$). Παρατηρώντας την trace διατύπωση (3.7) για το QAP και την διατύπωση (3.6) για την ελαχιστοποίηση της κατανάλωσης ενέργειας σε ένα δίαυλο μέσω της μετάθεσης των γραμμών, αντιλαμβανόμαστε πως μπορούμε να χειριστούμε το πρόβλημα μας ως ένα QAP [22].

Κεφάλαιο 4

ΜΕΘΟΔΟΙ ΕΠΙΛΥΣΗΣ ΚΑΙ ΠΑΡΑΜΕΤΡΟΙ

4.1 Γενικές Παράμετροι Και Κώδικας

Σε αυτό το κεφάλαιο θα αναλυθούν οι μέθοδοι που χρησιμοποιήθηκαν για την αντιμετώπιση του προβλήματος που αναφέραμε στο προηγούμενο κεφάλαιο. Ο κώδικας γράφτηκε σε γλώσσα προγραμματισμού python και μπορεί να βρεθεί στο εξής link : <https://github.com/MixalisMourtzakis/ENERGY-CONSUMPTION-OF-DSM-SOLVED-AS-QAP.git>. Η ανάγνωση του κώδικα συνιστάται για την πλήρη εμβάθυνση των μεθόδων που χρησιμοποιήθηκαν.

Οι παράμετροι του προβλήματος που παρέμειναν ίδιες και αφορούν όλες τις μεθόδους που χρησιμοποιήθηκαν είναι οι :

$$V_{DD} = 1$$

$$C_L = 1$$

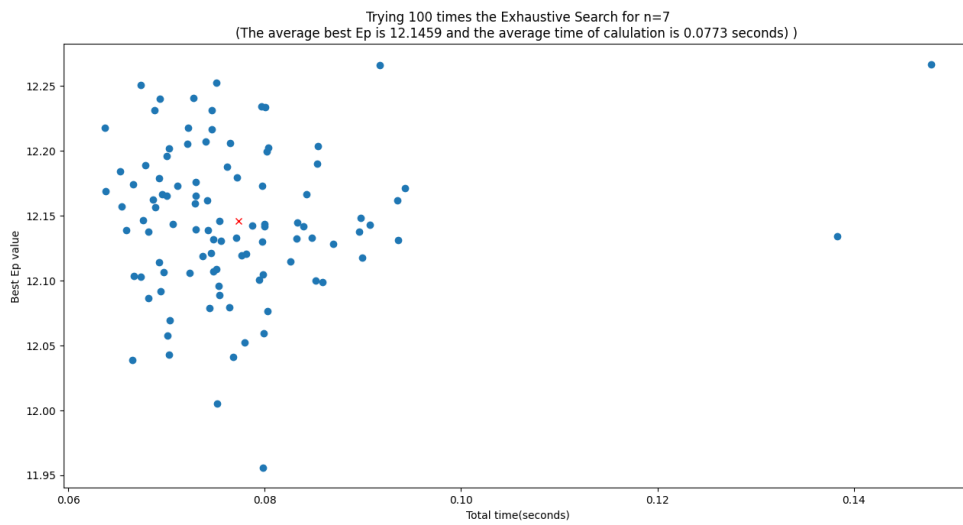
$$\text{Lambda} = 3$$

Οι παράμετροι που αφορούν συγκεκριμένες μεθόδους θα αναφέρονται στο τέλος του μέρους όπου περιγράφεται η αντίστοιχη μέθοδος. Οι παράμετροι που αφορούν τις διαφορές δοκιμές και αποτελέσματα θα αναφέρονται κατά τον σχολιασμό των αποτελεσμάτων.

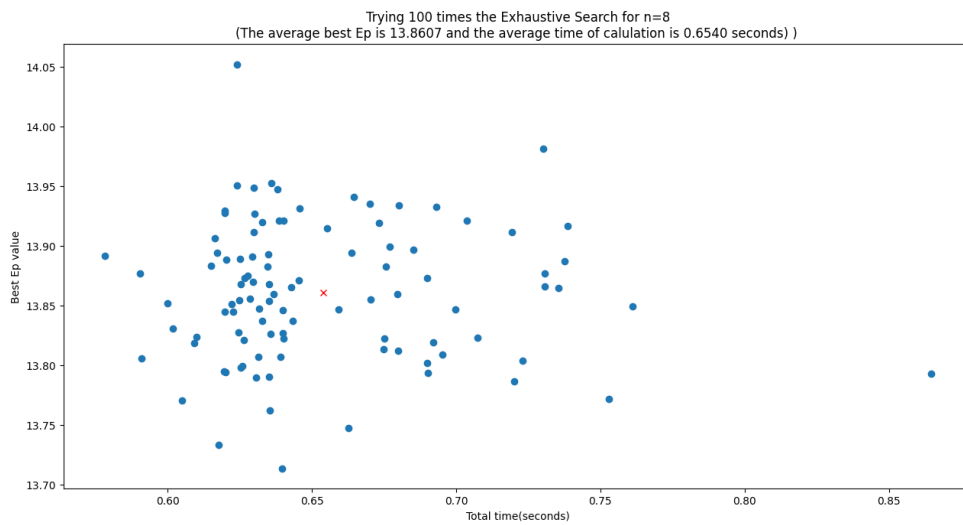
4.2 Εξαντλητική Μέθοδος

Η πρώτη μέθοδος που δοκιμάστηκε είναι η εξαντλητική μέθοδος. Σε αυτή την μέθοδο δοκιμάζονται όλες οι πιθανές μεταθέσεις και επιλέγεται αυτή που μας δίνει το επιθυμητό αποτέλεσμα, δηλαδή την ελάχιστη κατανάλωση ενέργειας. Αυτή η μέθοδος αν και μας εξασφαλίζει το βέλτιστο αποτέλεσμα είναι πολύ χρονοβόρα και απαιτεί και την δέσμευση αρκετού αποθηκευτικού χώρου από το μηχάνημα που την "τρέχουμε". Αυτό αυξάνεται σε πολύ μεγάλο βαθμό όσο αυξάνεται και το n (αριθμός γραμμών). Πιο συγκεκριμένα οι πιθανές μεταθέσεις που μπορούμε να έχουμε για αριθμό γραμμών n είναι $(n!)$. Παρακάτω θα δούμε δοκιμές που έγιναν με την εξαντλητική μέθοδο. Οι δοκιμές έγιναν για $m=10000$ (αριθμός κύκλων) και $n=7,8,9,10$. Για το κάθε n έγιναν 100 δοκιμές και

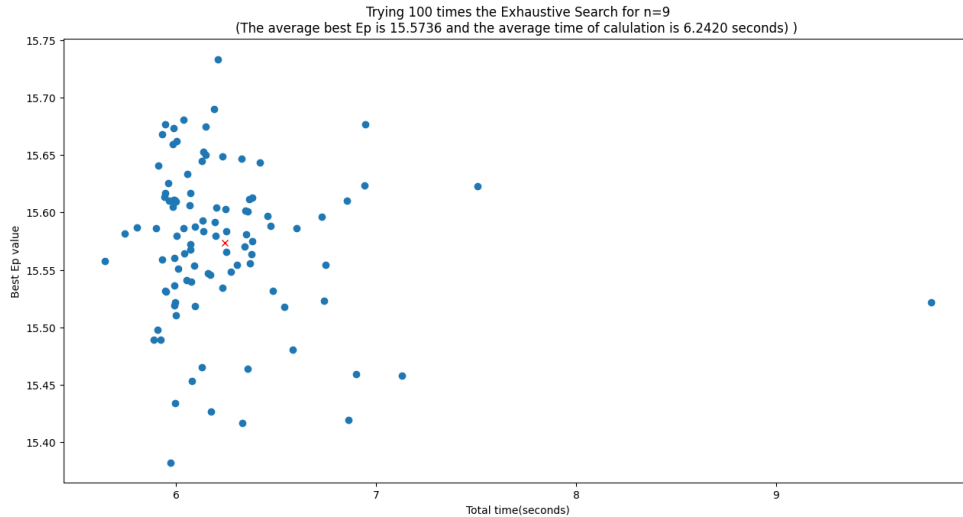
τα δεδομένα που χρησιμοποιήθηκαν ήταν τυχαία και διαφορετικά σε κάθε δοκιμή. Τα δεδομένα παρήχθησαν από την βιβλιοθήκη random της python.



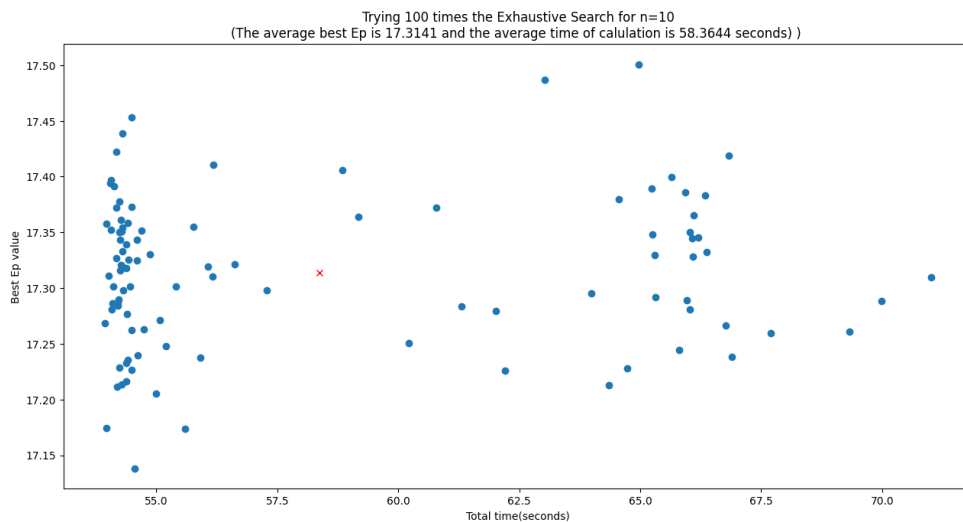
Σχήμα 4.1: Exhaustive Search ($n = 7, m = 10000, NrTrials = 100$).



Σχήμα 4.2: Exhaustive Search ($n = 8, m = 10000, NrTrials = 100$).



Σχήμα 4.3: Exhaustive Search ($n = 9, m = 10000, NrTrials = 100$).



Σχήμα 4.4: Exhaustive Search ($n = 10, m = 10000, NrTrials = 100$).

Όπως παρατηρούμε από τα παραπάνω διαγράμματα ο χρόνος που χρειάστηκε για να εκτελεστεί η εξαντλητική μέθοδος εξαρτάται από το (n) και ακολουθεί το $n!$ όπως ήταν και αναμενόμενο. Πράγμα που σημαίνει πως όσο αυξάνεται το n η χρήση αυτής της μεθόδου θα είναι όλο και χειρότερη από άποψη χρόνου. Επίσης να σημειωθεί πως πέρα από το ότι η μέθοδος αυτή είναι χρονοβόρα όταν έγινε η προσπάθεια για να ελεγχθεί η περίπτωση για $n = 11$ το μηχάνημα στο οποίο εκτελέστηκε η εργασία δεν μπόρεσε να αντεπεξέλθει λόγω της μνήμης που απαιτήθηκε να δεσμευτεί για να "τρέξει" το πρόγραμμά. Επομένως βλέπουμε πως για $n > 10$ η χρήση της εξαντλητικής μεθόδου δεν συνιστάται και είναι σχεδόν αδύνατη. Πάραυτα για $n < 10$ με την εξαντλητική μέθοδος μπορούμε να ελέγχουμε γρήγορα και αποτελεσματικά όλες τις πιθανές μεταθέσεις.

4.3 Ευρετικές Μέθοδοι

Παραπάνω είδαμε πως η εξαντλητική μέθοδος αν και μας δίνει την βέλτιστη λύση δεν είναι ιδανική για $n > 10$. Επομένως πρέπει να βρούμε έναν άλλο τρόπο να αναζητούμε την βέλτιστη λύση στο πρόβλημα μας. Για να το κάνουμε αυτό θα χρησιμοποιήσουμε την σύνδεση που κάναμε στο πρόβλημα μας με την γενική κατηγορία προβλημάτων QAP. Πιο συγκεκριμένα θα χρησιμοποιήσουμε ευρετικές μεθόδους, οι οποίες χρησιμοποιούνται για την επίλυση QAP, έτσι ώστε να αντιμετωπίσουμε το πρόβλημα που αναλύσαμε στο προηγούμενο κεφάλαιο. Παρακάτω θα αναλύσουμε τις ευρετικές μεθόδους που χρησιμοποιήθηκαν στην παρούσα εργασία.

4.3.1 Simulated Annealing (SA)

Η Simulated Annealing είναι μια ευρετική μέθοδος που βασίζεται στον ορισμό της “γειτονιάς”. Για το QAP σχεδόν όλες οι υλοποιήσεις του SA χρησιμοποιούν την μετάθεση μέσα σε μία γειτονία δηλαδή την ανταλλαγή θέσης δύο στοιχείων. Για να μειωθεί όμως ο κίνδυνος η λύση να μας παγιδευτεί σε ένα τοπικό βέλτιστο η βασική ιδέα της SA είναι η εξής: σε κάθε επανάληψη, επιλέγεται τυχαία μια γειτονική λύση και ελέγχεται εάν είναι καλύτερη από το ολικό βέλτιστο. Σε περίπτωση που είναι γίνεται αποδεκτή η μετάβαση σε αυτήν. Εάν όμως είναι χειρότερη τότε η γειτονική λύση γίνεται αποδεκτή με πιθανότητα $e^{-\Delta/T}$, όπου T είναι μια παράμετρος που ονομάζεται θερμοκρασία. Η θερμοκρασία αλλάζει (συνήθως μειώνεται) σε κάθε επανάληψη. Το σύνολο των θερμοκρασιών ονομάζεται cooling scheme. Ο Connolly (1990) πρότεινε το cooling scheme να προσαρμόζεται στο κάθε πρόβλημα και στον αριθμό των επαναλήψεων που καθορίζει ο χρήστης για αυτό [4].

Η ειδική παράμετρος που χρησιμοποιήθηκε για την μέθοδο SA είναι η:

$$InitialDelta = 1 \times 10^{-5}$$

4.3.2 Tabu Search (TABU)

Ο TABU είναι μια μέθοδος τοπικής αναζήτησης που εισήγαγε ο Glover ως μια τεχνική για να ξεπεραστούν τα τοπικά βέλτιστα στην αναζήτηση βέλτιστης λύσης. Η βασική ιδέα στην οποία στηρίζεται ο TABU είναι να "θυμάται" ποιες λύσεις έχουν επισκεφθεί κατά τη διάρκεια του αλγορίθμου, έτσι ώστε η αναζήτηση να κατευθυνθεί στις περισσότερες υποσχόμενες. Με αυτό τον τρόπο η εύρεση της βέλτιστης λύσης βασίζεται και στην μνήμη εκτός από την τοπική αναζήτηση.

Τα κύρια συστατικά του TABU είναι η δομή της γειτονιάς, οι κινήσεις, η tabu λίστα και το aspiration κριτήριο. Η κίνηση είναι μια λειτουργία η οποία, όταν εφαρμοστεί σε μια συγκεκριμένη λύση, δημιουργεί μια νέα γειτονική λύση. Στην περίπτωση μας η κίνηση είναι η εναλλαγή στοιχείων. Η tabu λίστα είναι μια λίστα απαγορευμένων ή "tabu" κινήσεων, δηλαδή κινήσεων που δεν επιτρέπεται να εφαρμοστούν στην τρέχουσα λύση. Η κατάσταση tabu των κινήσεων καθώς και η λίστα tabu ενημερώνεται κατά τη διάρκεια της αναζήτησης. Το tabu κριτήριο είναι μια συνθήκη που, όταν ικανοποιηθεί για μια tabu κίνηση, ακυρώνει την κατάσταση tabu της.

Συνήθως ο TABU ξεκινά με μια αρχική λύση και στην συνέχεια επιλέγει μια καλύτερη από τις γειτονικές λύσεις που δεν προκύπτουν από tabu κινήσεις. Έπειτα, η αρχική λύση αντικαθίσταται από την γειτονική καλύτερη λύση που βρέθηκε. Αυτή η διαδικασία μπορεί να επαναλαμβάνεται, δηλαδή να επισκέπτεται την ίδια λύση περισσότερες από μία φορές. Για να αποφευχθεί λοιπόν χρησιμοποιείται ένα κριτήριο tabu για τον εντοπισμό κινήσεων που αναμένεται να οδηγήσουν σε επαναλήψεις. Αυτές οι κινήσεις τότε χαρακτηρίζονται tabu και προστίθενται στη λίστα tabu. Επειδή, ωστόσο, η απαγόρευση ορισμένων κινήσεων είναι πιθανόν να εμποδίσει την επίσκεψη "ενδιαφέρουσων" λύσεων, ένα aspiration κριτήριο διακρίνει τις "ενδιαφέρουσες" κινήσεις από τις απαγορευμένες. Η αναζήτηση σταματά όταν εκτελεστούν όλες οι επαναλήψεις που έχουμε ορίσει. Γενικά η απόδοση του TABU εξαρτάται σε μεγάλο βαθμό από τις τιμές που δίνουμε στις μετρικές του, και δεν υπάρχουν κάποιες αποδεδειγμένα "καλύτερες" τιμές [14].

Οι ειδικές παράμετροι που χρησιμοποιήθηκαν για την μέθοδο TABU είναι οι:

$$Aspiration = 5 \times n \times n$$

$$TabuDuration = 8 \times n$$

4.3.3 Iterative Local Search Algorithm (ILSA)

Η ευρετική μέθοδος ILSA βασίζεται κυρίως στην τοπική αναζήτηση μιας σειράς τροποποιημένων λύσεων και έχει ως στόχο την αποφυγή τοπικών βέλτιστων και την πιο αποτελεσματική εξερεύνηση του χώρου των λύσεων. Κατά την εκτέλεση του γίνεται μια τοπική αναζήτηση σε κάθε επανάληψη η οποία χαρακτηρίζεται από τον αριθμό αλλαγών και την πιθανότητα αλλαγής. Πιο συγκεκριμένα σε κάθε επανάληψη η αλλαγή στην μετάθεση έχει πιθανότητα εκτέλεσης ίση με την πιθανότητα αλλαγής και εάν εκτελεστεί αυτό σημαίνει πως θα έχουμε τυχαίες εναλλαγές στοιχείων στην μετάθεση μας. Ο αριθμός αυτός των αλλαγών είναι ο αριθμός αλλαγών που αναφέραμε παραπάνω. Αναφέραμε πως ο ILSA εκτελεί αυτή την τοπική αναζήτηση σε μια σειρά τροποποιημένων λύσεων. Αυτό το κάνει με τον εξής τρόπο. Μετά από την τοπική αναζήτηση σε κάθε επανάληψη εξετάζει εάν η νέα μετάθεση μας δίνει χαμηλότερη κατανάλωση ισχύος από την έως τότε ελάχιστη. Εάν ναι τότε η τιμή του μετρητή μας γίνεται ίση με μηδέν και η νέα βέλτιστη τιμή αποθηκεύεται, εάν όχι τότε ο μετρητής μας αυξάνεται κατά μια μονάδα και συνεχίζεται η τοπική αναζήτηση. Όταν η τιμή του μετρητή μας περάσει την τιμή του Shuffle Tolerance που έχει οριστεί η μετάθεση που έχουμε αλλάξει εντελώς. Γίνεται ένα τυχαίο "ανακάτεμα" των στοιχείων και έτσι δημιουργείται μια νέα λύση η οποία θα ερευνηθεί με την τοπική αναζήτηση που αναφέραμε παραπάνω [35].

Οι ειδικές παράμετροι που χρησιμοποιήθηκαν για την μέθοδο ILSA είναι οι:

$$ShuffleTolerance = 100$$

$$PropabilityChange = 0.5$$

$$NumberSwapsLI = 3$$

4.3.4 Fast Ant System (FANT)

Το σκεπτικό πίσω από το σύστημα FANT είναι να αποδίδει ένα "ενδιαφέρον" για κάθε στοιχείο μιας λύσης. Ένα μέρος αυτού του "ενδιαφέροντος" μπορεί να υπολογιστεί εκ των προτέρων και ένα άλλο μέρος υπολογίζεται εκ των υστέρων, όταν η διαδικασία έχει παράγει και αξιολογήσει λύσεις. Αυτό το "εκ των υστέρων ενδιαφέρον" ονομάζεται "επίπεδο ιχνών φερομόνης" σε ένα σύστημα μυρμηγκιών (Coloni, Dorigo, and Maniezzo, 1992). Στα FANT, οι λύσεις κατασκευάζονται τυχαία, με την πιθανότητα επιλογής ενός στοιχείου να είναι ανάλογη του "ενδιαφέροντός" του. Για το QAP, έχουν σχεδιαστεί διάφορα είδη FANT. Τα περισσότερα από αυτά δεν λαμβάνουν υπόψη το "εκ των προτέρων ενδιαφέρον". Το "εκ των υστέρων ενδιαφέρον" εμφανίζεται ως ένας πίνακας T μεγέθους $(n \times n)$, του οποίου το στοιχείο τ_{ij} είναι μια στατιστική επί της ρύθμισης του $\pi_i = j$ για τις προηγούμενες λύσεις π που κατασκευάστηκαν.

Στον FANT (Taillard, 1998), όλα τα στοιχεία του πίνακα T ορίζονται αρχικά σε 1. Για τη δημιουργία μιας νέας (προσωρινής) λύσης μ , χρησιμοποιείται μια κατασκευαστική μέθοδος που επιλέγει τα στοιχεία της μ διαδοχικά, με τυχαία σειρά και με πιθανότητα ανάλογη των τιμών που περιέχονται στον πίνακα T . Πιο συγκεκριμένα, η κατασκευαστική μέθοδος παρουσιάζεται στο Σχήμα 4.5. Η προσωρινή λύση μ που παράγεται στο πρώτο βήμα, όπως είναι και αναμενόμενο, δεν είναι γενικά πολύ καλή γιατί, στην πρώτη επανάληψη, η μ είναι απλά μια τυχαία μετάθεση. Μια διαδικασία πρώτης βελτίωσης γειτονικής λύσης εφαρμόζεται στη μ . Αυτή η διαδικασία μπορεί να εκτελεστεί σε $O(n^3)$ και δεν είναι σίγουρο πως θα μας επιστρέψει τοπικό βέλτιστο. Πάραυτα είναι γρήγορη και μπορεί να παράγει διαφορετικές λύσεις όταν ξεκινά με την ίδια αρχική λύση, με την προϋπόθεσή βέβαια πως αυτή δεν είναι τοπικό βέλτιστο. Η βελτιωμένη λύση ονομάζεται π .

- 1) $I = \emptyset, J = \emptyset$
- 2) While $|I| < n$ repeat:
 - 2a) Choose i , randomly, uniformly, $1 \leq i \leq n, i \notin I$.
 - 2b) Choose j , randomly, $1 \leq j \leq n, j \notin J$, with probability $\frac{\tau_{ij}}{\sum_{1 \leq k \leq n, k \notin J} \tau_{ik}}$ and set $\mu_i = j$.
 - 2c) $I = I \cup \{i\}, J = J \cup \{j\}$

Σχήμα 4.5: Μέθοδος κατασκευής λύσης [4].

Οι τιμές που αποθηκεύονται στον T βασίζονται σε δύο παραμέτρους (r και r^*) οι οποίες καθορίζουν την αύξηση των στοιχείων του T που αντιστοιχούν στη λύση π και την λύση π^* (γενική καλύτερη λύση έως εκείνη την στιγμή) αντίστοιχα. Το r^* είναι μια σταθερή παράμετρος που ορίζουμε εμείς ενώ το r μεταβάλλεται. Ορίζουμε $r = 1$ και $\tau_{ij} = r$, όπου $1 \leq i$ και $j \leq n$, πράγμα που σημαίνει ότι η μνήμη δεν περιέχει αρχικά καμία πληροφορία. Συνήθως, τα στοιχεία του πίνακα T ενημερώνονται με τον τρόπο που φαίνεται στο Σχήμα 4.6.

```

1)          For i = 1 to n do:
1a)    $\tau_{i\pi_i} = \tau_{i\pi_i} + r$ 
1b)    $\tau_{i\pi_i}^* = \tau_{i\pi_i}^* + r^*$ 

```

Σχήμα 4.6: Ενημέρωση του πίνακα T.

Το π είναι η λύση που παράγεται στην τρέχουσα επανάληψη και π^* είναι η καλύτερη λύση που έχει παραχθεί μέχρι στιγμής. Η ενημέρωση του πίνακα T μπορεί να γίνει με 2 διαφορετικούς τρόπους αναλόγως την κατάσταση:

1. Αν η π^* έχει βελτιωθεί, τότε το r επαναφέρεται στο 1 και όλα τα στοιχεία του T γίνονται 1. Σκοπός αυτής της επαναφοράς είναι να ενταθεί η αναζήτηση γύρω από την π^* δίνοντας λιγότερη σημασία στο παρελθόν της αναζήτησης.
2. Αν η προσωρινή λύση μ που παράγεται στο βήμα 26(Σχήμα 4.5) είναι ίση με την π^* , τότε το r αυξάνεται κατά μία μονάδα και όλα τα στοιχεία του πίνακα T γίνονται ίσα με r . Αυτή η κατάσταση εμφανίζεται όταν η π^* δεν έχει βελτιωθεί για μεγάλο αριθμό επαναλήψεων, πράγμα που σημαίνει ότι η ενίσχυση των καταχωρήσεων που αντιστοιχούν στην π^* είναι υπερβολικά υψηλή. Σκοπός αυτής της στρατηγικής είναι να αλλάξει κατεύθυνση η αναζήτηση όταν οι πληροφορίες που περιέχονται στον T δεν διαφέρουν πολύ από την π^* . [4]

Η ειδική παράμετρος που χρησιμοποιήθηκε για την μέθοδο FANT είναι η ($R = r$):

$$R = 1$$

4.3.5 Greedy Randomized Adaptive Search Procedure (GRASP)

Ο Greedy Randomized Adaptive Search Procedure (GRASP) εισήχθη από τους Feo και Resende και έχει εφαρμοστεί με επιτυχία σε διάφορα δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης μεταξύ των οποίων είναι το QAP και το BiQAP. Διαθέτει τέσσερα βασικά συστατικά: μια άπληστη συνάρτηση, μια προσαρμοστική στρατηγική αναζήτησης, μια διαδικασία επιλογής βασισμένη στις πιθανότητες και μια τεχνική τοπικής αναζήτησης. Χωρίζετε σε δύο στάδια τα οποία επαναλαμβάνονται για ένα επιθυμητό αριθμό επαναλήψεων. Το πρώτο είναι το κατασκευαστικό στάδιο στο οποίο κατασκευάζεται μια λύση και το δεύτερο είναι το στάδιο τοπικής αναζήτησης στο οποίο γίνεται μια τοπική αναζήτηση στην λύση που κατασκευάστηκε στο πρώτο στάδιο με σκοπό την βελτίωση της. Μετά την βελτίωση της ελέγχεται εάν η τελική λύση του σταδίου αυτού είναι καλύτερη από την γενική καλύτερη λύση. [36]

Το πρώτο στάδιο στο οποίο γίνονται οι αναθέσεις και κατασκευάζεται η λύση χωρίζεται σε δύο φάσεις. Αρχικά γίνονται οι πρώτες 2 αναθέσεις και στην συνέχεια οι υπόλοιπες. Οι πρώτες δύο αναθέσεις γίνονται ταυτόχρονα στην πρώτη φάση του κατασκευαστικού σταδίου, ενώ οι υπόλοιπες γίνονται μία ανάθεση ανά βήμα στην δεύτερη φάση του κατασκευαστικού σταδίου. Η άπληστη συνάρτηση που επιλέχθηκε στην εργασία αυτή ταξινομεί τις επιτρεπόμενες αναθέσεις με βάση τις αντίστοιχες τιμές τους στον πίνακα T^a . Για την επιλογή του πρώτου ζευγαριού αρχικά ταξινομούμε τις τιμές του πίνακα T^a . Επίσης θεωρούμε μια μεταβλητή beta για την οποία ισχύει $0 < \text{beta} < 1$. Ο ταξινομημένος πίνακας μας έχει $n^2 - n$ τιμές και εμείς κρατάμε τις $\lfloor \beta(n^2 - n) \rfloor$ καλύτερες. Στην συνέχεια γίνεται μια τυχαία επιλογή από τις εναπομείναντες τιμές του πίνακα η οποία αντιστοιχεί σε κάποιο ζευγάρι. Να σημειωθεί πως ο πίνακας αυτός σχηματίζεται μια φορά μόνο καθώς δεν μεταβάλλεται όσο "τρέχει" ο αλγόριθμος. Για την δεύτερη φάση χρησιμοποιούμε μια νέα μεταβλητή alpha για την οποία ισχύει $0 < \text{alpha} < 1$. Σε αυτή την φάση επιλέγεται ένα στοιχείο ανά βήμα μέχρι τον σχηματισμό μιας μετάθεσης. Σε κάθε επανάληψη σχηματίζεται και ταξινομείται ένας πίνακας με τις επιτρεπτές τιμές του πίνακα T^a . Απο αυτές κρατάμε τις $(\text{alpha}) * (\text{αριθμός τιμών})$ και στην συνέχεια επιλέγουμε τυχαία μια απο αυτές. Έτσι βήμα-βήμα σχηματίζεται μια μετάθεση.

Στο δεύτερο στάδιο γίνεται μια τοπική αναζήτηση με σκοπό την βελτίωση της λύσης που έχουμε κατασκευάσει από το πρώτο στάδιο. Η τοπική αναζήτηση που επιλέχθηκε στην συγκεκριμένη μελέτη βασίζεται στην επιλογή μιας "γειτονίας" αναζήτησης. Πιο συγκεκριμένα εφόσον βρισκόμαστε σε ένα στοιχείο της μετάθεσης μας αναζητούμε-κάνουμε αλλαγές μέσα στην "γειτονία" του στοιχείου μας με σκοπό την εύρεση καλύτερης λύσης. Η γειτονία που χρησιμοποιήθηκε είναι τριών στοιχείων επομένως δοκιμάζονται όλες οι πιθανές αλλαγές τριών γειτονικών στοιχείων, και αυτό γίνεται για όλες τις "γειτονιές" τριών στοιχείων στην μετάθεση μας. Όταν τελειώσει η τοπική αναζήτηση ελέγχουμε εάν η μετάθεση που προέκυψε μας δίνει την συνολική ελάχιστη κατανάλωση ισχύος.

Οι ειδικές παράμετροι που χρησιμοποιήθηκαν για την μέθοδο GRASP είναι οι:

$$\text{alpha} = 0.5$$

$$\text{beta} = 0.5$$

Κεφάλαιο 5

ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ

Σε αυτό το κεφάλαιο θα παρουσιάσουμε και θα σχολιάσουμε την απόδοση των 5 ευρετικών μεθόδων που χρησιμοποιήσαμε για να αντιμετωπίσουμε το πρόβλημα μας. Ισχύουν όλες οι γενικές μετρικές που αναφέραμε καθώς και οι ειδικές μετρικές της κάθε μεθόδου. Οι δοκιμές έγιναν σε 2 στάδια. Αρχικά οι μέθοδοι δοκιμάστηκαν σε τυχαία dataset και στην συνέχεια σε συγκεκριμένα dataset. Επιπλέον η ανάλυση των μεθόδων για την κάθε περίπτωση έγιναν σε 2 στάδια. Έγινε μελέτη κατά την οποία η κάθε μέθοδος “έτρεχε” για 600 δευτερόλεπτα και σκοπό είχε να συγκριθεί η απόδοση των μεθόδων. Επίσης έγινε και μελέτη κατά την οποία η κάθε μέθοδος “έτρεχε” 3600 δευτερόλεπτα η οποία είχε σαν σκοπό να μελετήσει το τρόπο που “συμπεριφέρεται” η κάθε μέθοδος εάν χρησιμοποιείται για μεγάλο αριθμό επαναλήψεων. Προφανώς και μεγαλύτερη βαρύτητα έχουν οι μελέτες που έγιναν για 600 δευτερόλεπτα καθώς με την χρήση των ευρετικών μεθόδων στοχεύουμε στην γρήγορη προσέγγιση της βέλτιστης λύσης, πάραυτα έχει ενδιαφέρον να δούμε εάν οι μέθοδοί μας μετά από κάποιο χρόνο τείνουν να μην βρίσκουν καλύτερη λύση από την τελευταία που έχουν βρει. Τα διαγράμματα όλων των δοκιμών που έγιναν μπορούν να βρεθούν στο εξής link : <https://github.com/MixalisMourtzakis/ENERGY-CONSUMPTION-OF-DSM-SOLVED-AS-QAP.git>. Επειδή ο αριθμός των διαγραμμάτων είναι πολύ μεγάλος στην εργασία αυτή θα παρουσιάσουμε για κάθε περίπτωση τα 2 πιο αντιπροσωπευτικά διαγράμματα, δηλαδή το διάγραμμα που προσεγγίζει τον μέσο όρο των αποτελεσμάτων για την εκάστοτε περίπτωση. Ένα για την δοκιμή των 600 δευτερολέπτων και ένα για την περίπτωση των 3600 δευτερολέπτων. Πέρα από τα διαγράμματα που υπάρχουν για την σύγκριση των μεθόδων που υπάρχουν στο link υπάρχουν επίσης και διαγράμματα μεμονωμένα για την κάθε μέθοδο τα οποία έχουν και ένα έξτρα άξονα, παράλληλο στον χρόνο, με τον αριθμό επαναλήψεων. Ο αριθμός των επαναλήψεων που “τρέχει” ένα πρόγραμμα μέσα σε ένα χρονικό διάστημα εξαρτιέται σε πολύ μεγάλο βαθμό από τα χαρακτηριστικά του μηχανήματος. Η κάθε δοκιμή έγινε για ένα συγκεκριμένο χρονικό διάστημα όπως αναφέραμε παραπάνω. Για αυτό λοιπόν σε αυτό το σημείο θα αναφέρουμε τα χαρακτηριστικά του μηχανήματος στο οποίο έγιναν οι δοκιμές τα οποία είναι τα εξής:

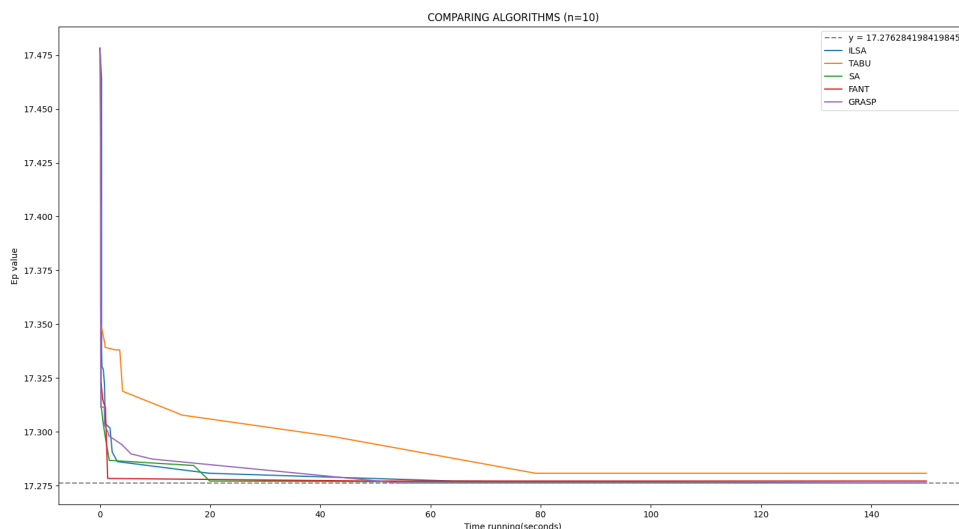
- Επεξεργαστής : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- Μνήμη RAM : 8.00 GB

5.1 Αποτελέσματα Με Είσοδο Τυχαίων Δεδομένων

Όπως αναφέραμε και παραπάνω το πρώτο στάδιο δοκιμών έγινε σε τυχαία dataset. Τα dataset αυτά ήταν για $m = 10000$ (αριθμός κύκλων ρολογιού) και τα δεδομένα δημιουργήθηκαν με την συνάρτηση random της random βιβλιοθήκης της γλώσσας προγραμματισμού python. Σε κάθε δοκιμή το dataset ανανεωνόταν. Οι δοκιμές με τυχαία είσοδο έγιναν έτσι ώστε τα αποτελέσματα μας να είναι ανεξάρτητα από κάποιο συγκεκριμένο dataset. Επιπλέον οι δοκιμές έγιναν συναρτήσει του n (αριθμός γραμμών). Οι τιμές του n που επιλέχθηκαν είναι ($n=10, 13, 16, 20, 25, 30, 40, 50, 75, 100, 150$). Επιλέχθηκαν να αυξάνονται εκθετικά για να παρατηρηθεί το πως αλλάζουν τα αποτελέσματα μας από το ένα n στο επόμενο χωρίς αυτό να εξαρτάται από ένα συγκεκριμένο βήμα- n . Επίσης βλέπουμε πως τα τελευταία n έχουν αρκετά μεγάλες τιμές που μπορεί να μην ανταποκρίνονται στην πραγματικότητα. Πρακτικά ο αριθμός γραμμών είναι στις περισσότερες περιπτώσεις μικρός. Πάραυτα έγιναν δοκιμές και σε μεγαλύτερα n κυρίως για εγκυκλοπαιδικό ενδιαφέρον. Άλλωστε όπως αναφέραμε και στα προηγούμενα κεφάλαια το QAP είναι ένα πρόβλημα με πάρα πολλές εφαρμογές εκ των οποίων κάποιες απαιτούν την χρήση ευρετικών μεθόδων και για μεγάλα n .

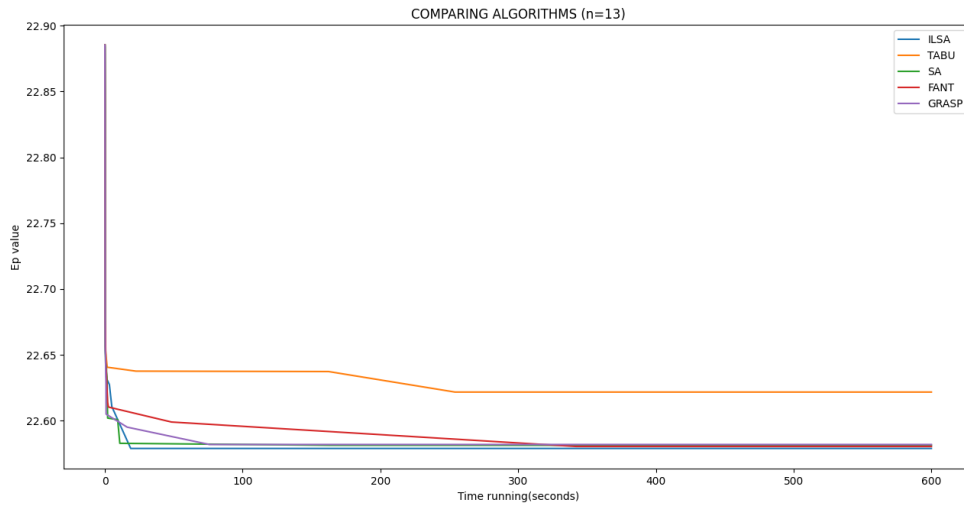
5.1.1 Διαγράμματα για τυχαία είσοδο και $n = 10$

Η συγκεκριμένη δοκιμή έγινε για 150 δευτερόλεπτα καθώς όπως είδαμε στο προηγούμενο κεφάλαιο για $n = 10$ η εξαντλητική μέθοδος βρίσκει την βέλτιστη λύση σε περίπου 60 δευτερόλεπτα. Στο συγκεκριμένο διάγραμμα προστέθηκε και με διακεκομμένη γραμμή η βέλτιστη λύση η οποία βρέθηκε από την εξαντλητική μέθοδο. Παρατηρούμε πως όλες οι μέθοδοι εκτός την Tabu φτάνουν στην βέλτιστη λύση ή αρκετά κοντά της πριν φτάσουν τα 60 δευτερόλεπτα.

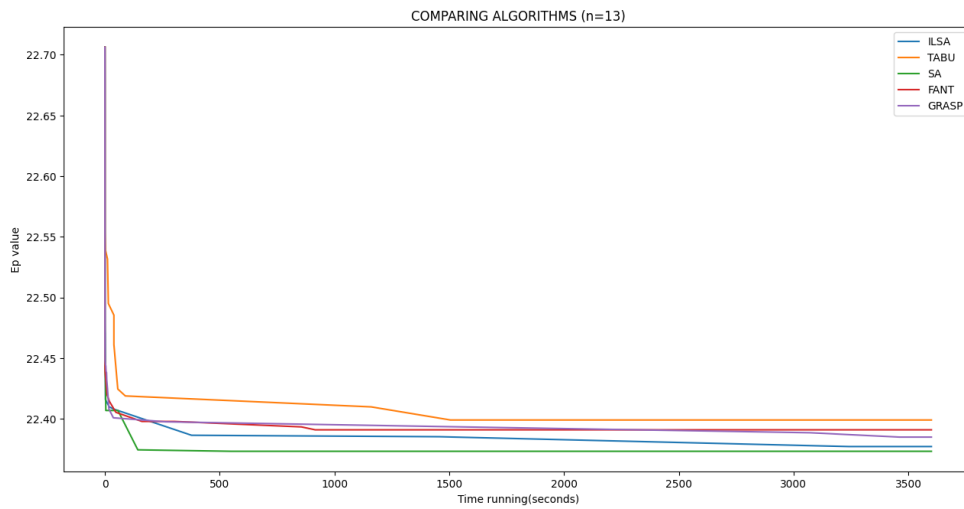


Σχήμα 5.1: Αποτελέσματα για το $n = 10$ σε δοκιμή 150 δευτερολέπτων.

5.1.2 Διαγράμματα για τυχαία είσοδο και $n = 13$

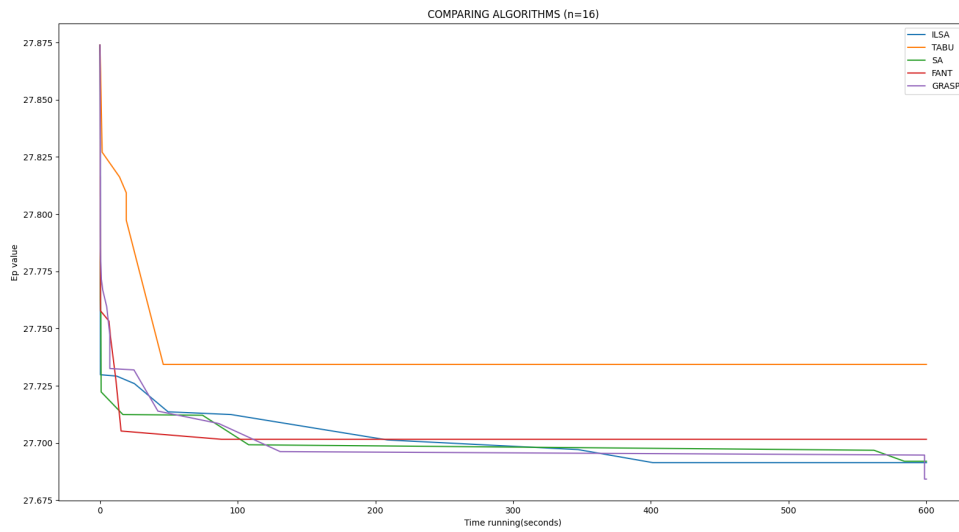


Σχήμα 5.2: Αποτελέσματα για το $n = 13$ σε δοκιμή 600 δευτερολέπτων.

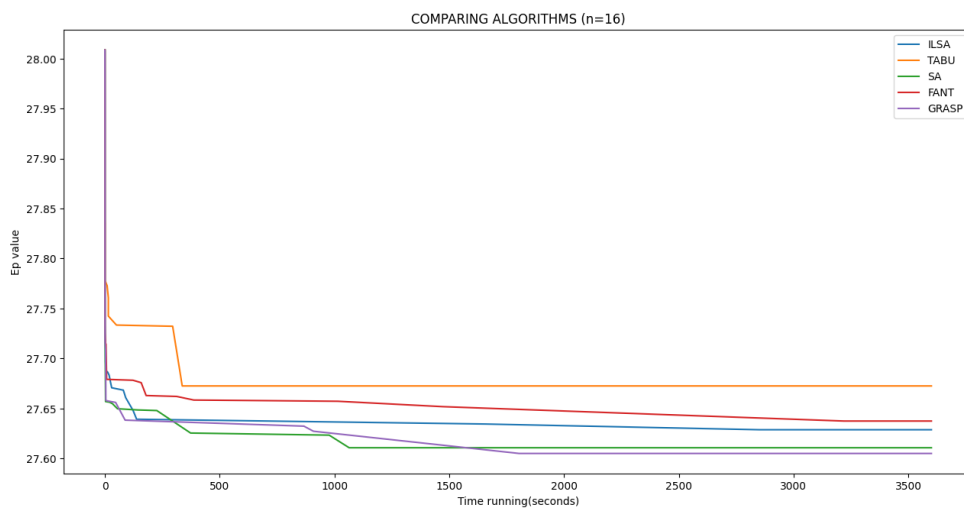


Σχήμα 5.3: Αποτελέσματα για το $n = 13$ σε δοκιμή 3600 δευτερολέπτων.

5.1.3 Διαγράμματα για τυχαία είσοδο και $n = 16$

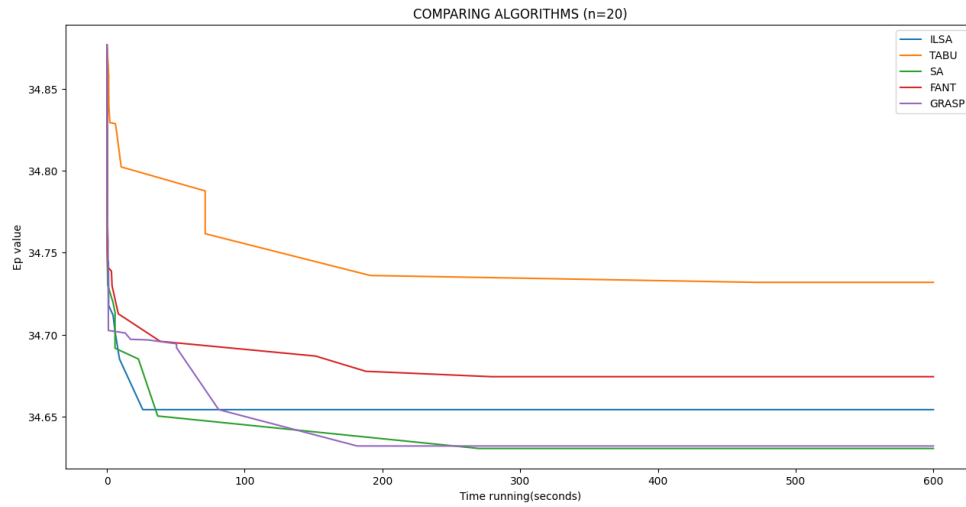


Σχήμα 5.4: Αποτελέσματα για το $n = 16$ σε δοκιμή 600 δευτερολέπτων.

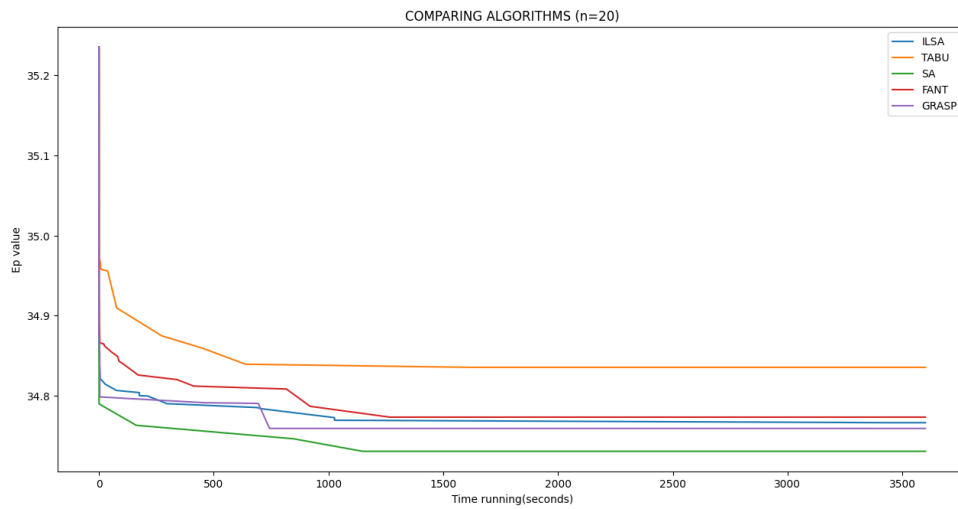


Σχήμα 5.5: Αποτελέσματα για το $n = 16$ σε δοκιμή 3600 δευτερολέπτων.

5.1.4 Διαγράμματα για τυχαία είσοδο και $n = 20$

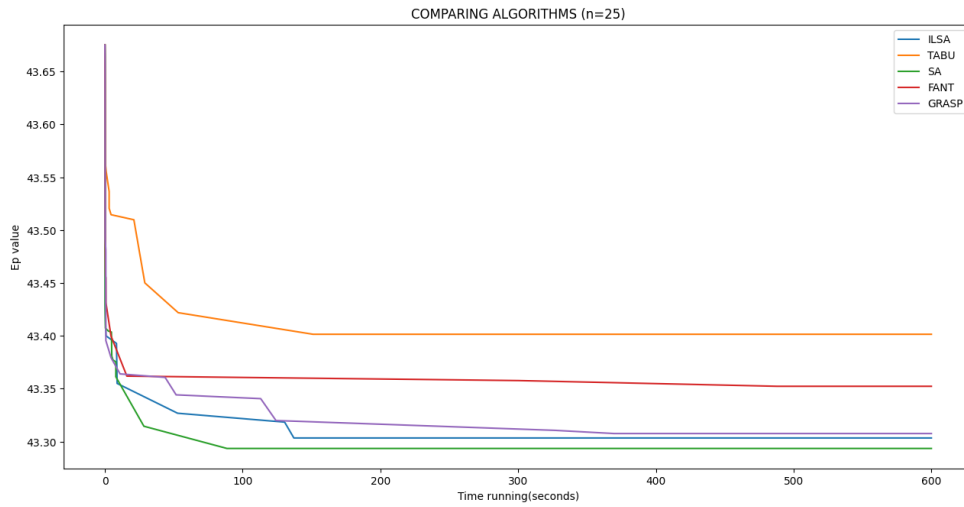


Σχήμα 5.6: Αποτελέσματα για το $n = 20$ σε δοκιμή 600 δευτερολέπτων.

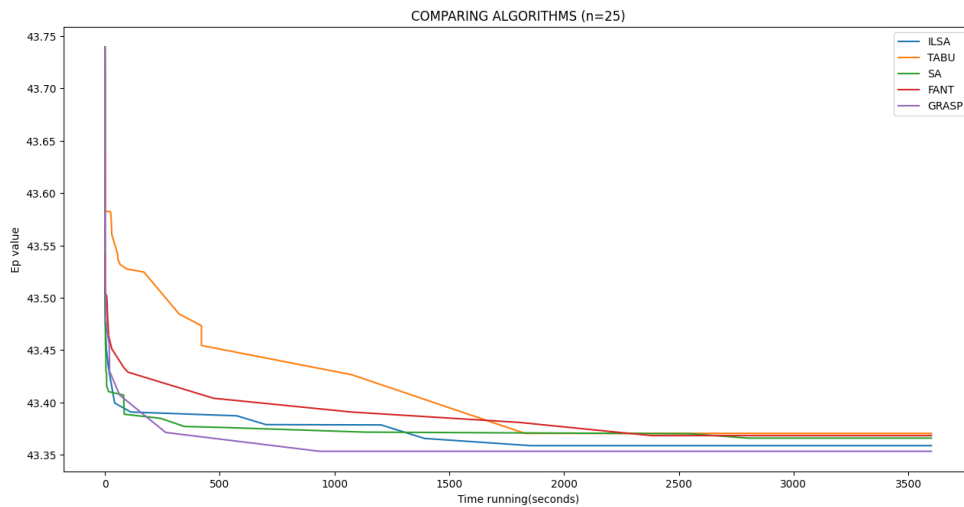


Σχήμα 5.7: Αποτελέσματα για το $n = 20$ σε δοκιμή 3600 δευτερολέπτων.

5.1.5 Διαγράμματα για τυχαία είσοδο και $n = 25$

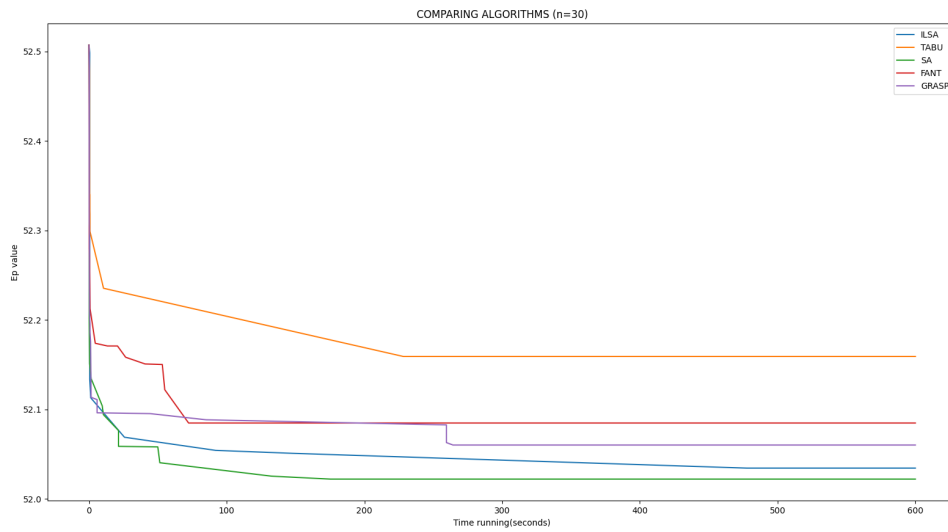


Σχήμα 5.8: Αποτελέσματα για το $n = 25$ σε δοκιμή 600 δευτερολέπτων.

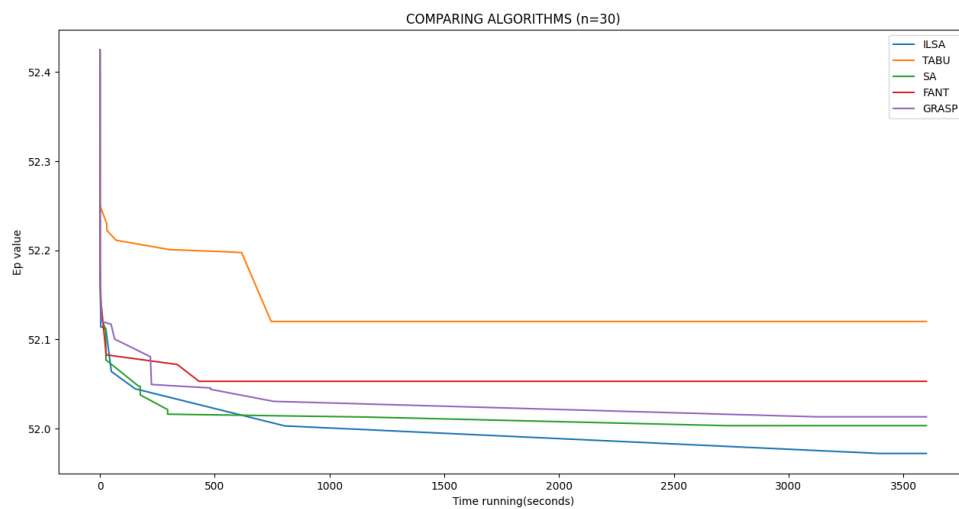


Σχήμα 5.9: Αποτελέσματα για το $n = 25$ σε δοκιμή 3600 δευτερολέπτων.

5.1.6 Διαγράμματα για τυχαία είσοδο και $n = 30$

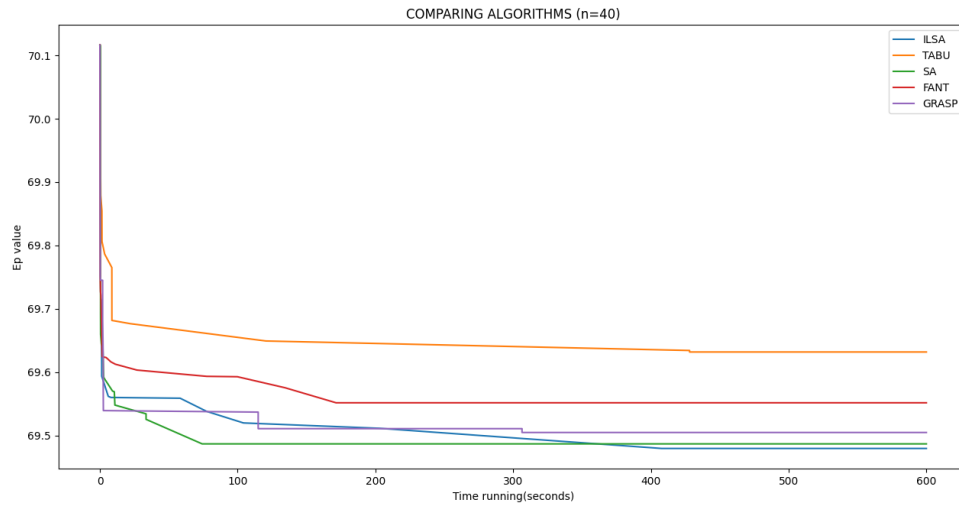


Σχήμα 5.10: Αποτελέσματα για το $n = 30$ σε δοκιμή 600 δευτερολέπτων.

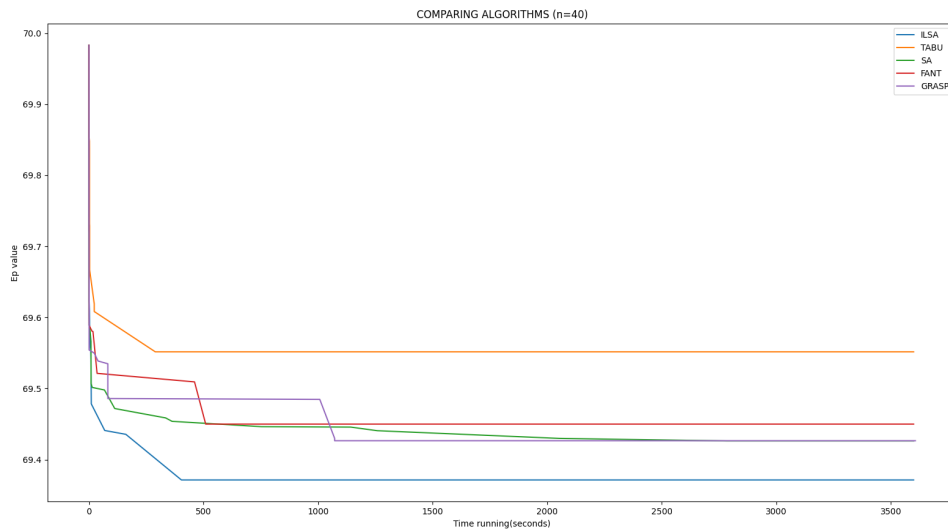


Σχήμα 5.11: Αποτελέσματα για το $n = 30$ σε δοκιμή 3600 δευτερολέπτων.

5.1.7 Διαγράμματα για τυχαία είσοδο και $n = 40$

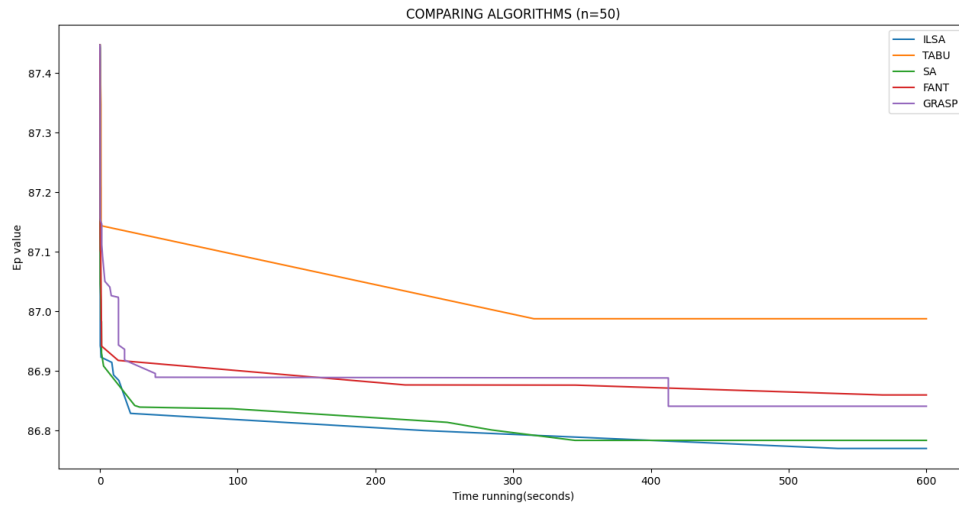


Σχήμα 5.12: Αποτελέσματα για το $n = 40$ σε δοκιμή 600 δευτερολέπτων.

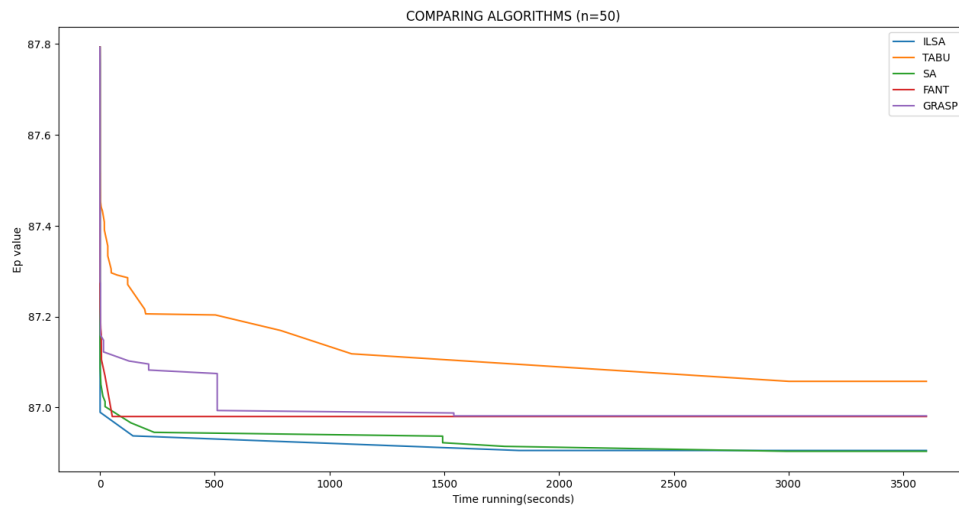


Σχήμα 5.13: Αποτελέσματα για το $n = 40$ σε δοκιμή 3600 δευτερολέπτων.

5.1.8 Διαγράμματα για τυχαία είσοδο και $n = 50$

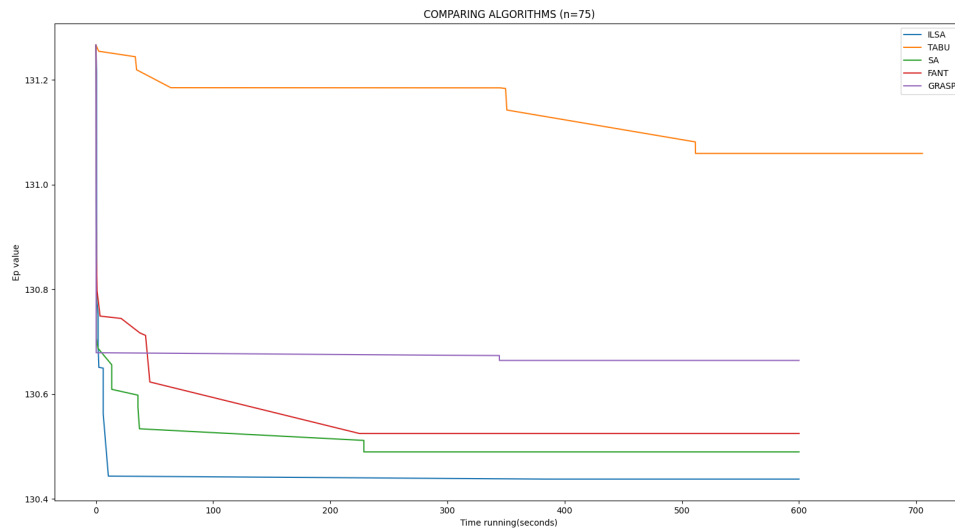


Σχήμα 5.14: Αποτελέσματα για το $n = 50$ σε δοκιμή 600 δευτερολέπτων.

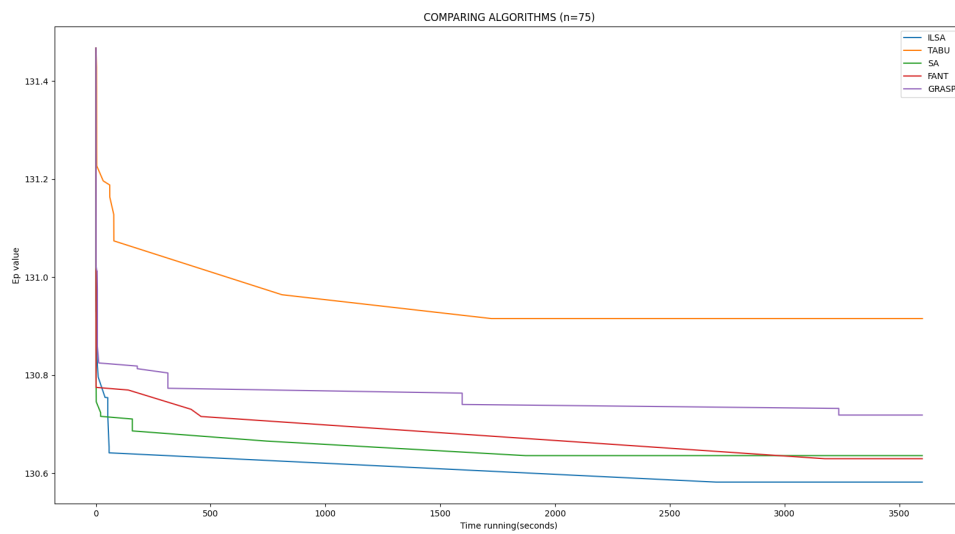


Σχήμα 5.15: Αποτελέσματα για το $n = 50$ σε δοκιμή 3600 δευτερολέπτων.

5.1.9 Διαγράμματα για τυχαία είσοδο και $n = 75$

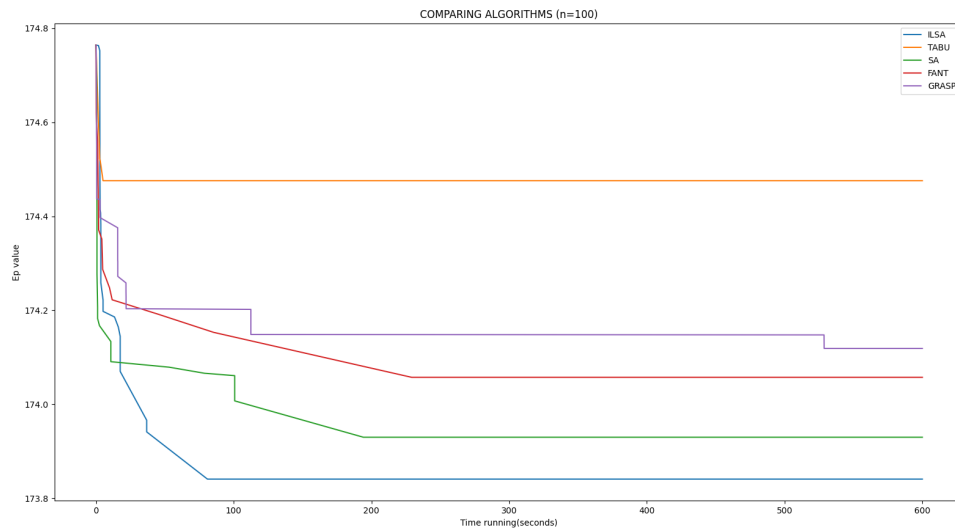


Σχήμα 5.16: Αποτελέσματα για το $n = 75$ σε δοκιμή 600 δευτερολέπτων.

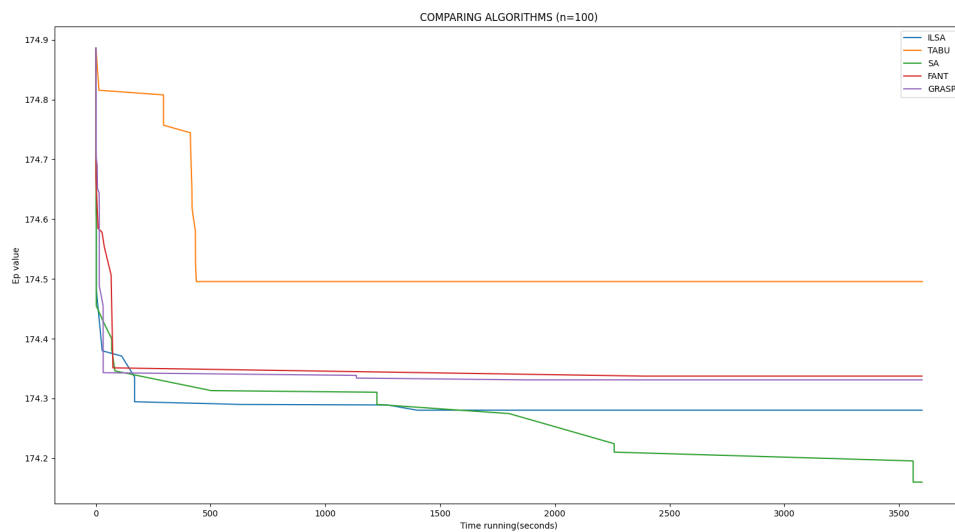


Σχήμα 5.17: Αποτελέσματα για το $n = 75$ σε δοκιμή 3600 δευτερολέπτων.

5.1.10 Διαγράμματα για τυχαία είσοδο και $n = 100$

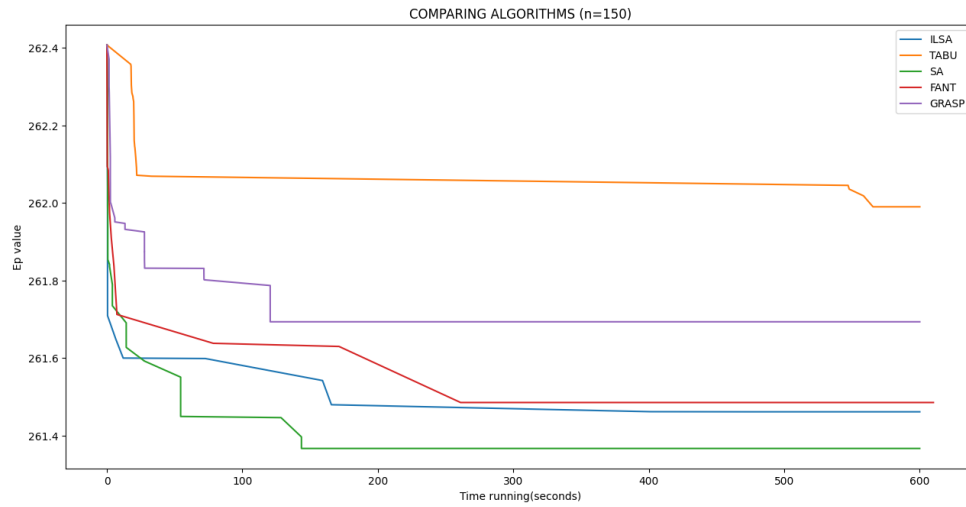


Σχήμα 5.18: Αποτελέσματα για το $n = 100$ σε δοκιμή 600 δευτερολέπτων.

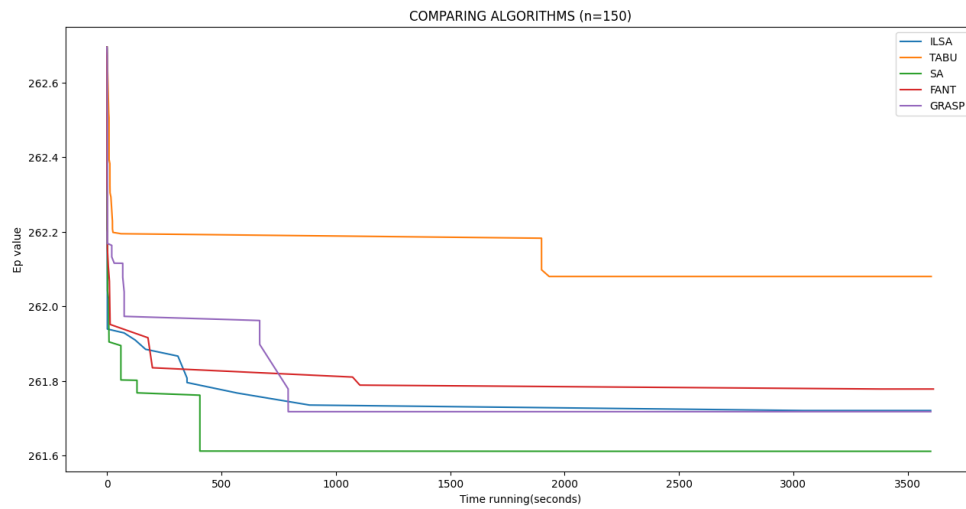


Σχήμα 5.19: Αποτελέσματα για το $n = 100$ σε δοκιμή 3600 δευτερολέπτων.

5.1.11 Διαγράμματα για τυχαία είσοδο και $n = 150$



Σχήμα 5.20: Αποτελέσματα για το $n = 150$ σε δοκιμή 600 δευτερολέπτων.



Σχήμα 5.21: Αποτελέσματα για το $n = 150$ σε δοκιμή 3600 δευτερολέπτων.

5.1.12 Σχολιασμός αποτελεσμάτων για τυχαία είσοδο

Σε αυτό το σημείο θα σχολιάσουμε τα αποτελέσματα των δοκιμών με τυχαία είσοδο. Αρχικά παρατηρούμε πως οι μέθοδοι μας έχουν μια μεγάλη πτώση στην αρχή και στην συνέχεια τείνουν να μένουν στην ίδια τιμή ή να βρίσκουν κάποια καλύτερη πάρα πολύ κοντά σε αυτή που είναι ήδη. Όπως είναι και αναμενόμενο όσο αυξάνεται το n αυξάνονται και οι φορές που παρατηρούμε τις μεθόδους να βρίσκουν νέα χαμηλότερη τιμή μετά από την αρχική μεγάλη πτώση αλλά πάραυτα η γενική μορφή που παρατηρούμε στα αποτελέσματα μας σε γενικές γραμμές είναι σταθερή.

Όσο άφορα την σύγκριση των μεθόδων παρατηρούμε αρχικά πως ο TABU είναι ο χειρότερος για όλα τα n . Ελάχιστες είναι οι φορές τις οποίες με τον TABU είχαμε καλύτερα αποτελέσματα σε σχέση με κάποια άλλη μέθοδο. Για τις άλλες 4 μεθόδους θα παρατηρήσουμε πως "συμπεριφέρονται" για κάθε n . Αρχικά να πούμε πως για τα $n = (13,16,20,25,30,40)$ ο FANT είναι η 2η χειρότερη μέθοδος μετά τον TABU. Για αυτά τα n στις χαμηλότερες τιμές του $n = (13,16,20)$ παρόλο που και οι 3 εναπομείνουσες μέθοδοι έχουν πάνω κάτω την ίδια απόδοση, φαίνεται πως ο GRASP έχει ένα μικρό "προβάδισμα". Στα μεγαλύτερα $n = (25,30,40)$ φαίνεται πως έχουμε την εξής σειρά βάση απόδοσης (SA - ILSA - GRASP - FANT - TABU) με τον SA να προσεγγίζει τον ILSA και τον GRASP να προσεγγίζει τον FANT όσο αυξάνεται το n . Για τα μεγαλύτερα $n = (50,75,100,150)$ παρατηρούμε πως η σειρά βάση απόδοσης φαίνεται να είναι (ILSA - SA - FANT - GRASP - TABU) με μόνη εξαίρεση το $n = 150$ όπου έχουμε μια αλλαγή στην θέση των (ILSA - SA).

Συμπερασματικά παρατηρούμε πως για τα μικρά $n = (13,16,20)$ οι καλύτερες μέθοδοι είναι οι (GRASP - SA - ILSA) με μικρό "προβάδισμα" του GRASP. Για τα $n = (25,30,40)$ η καλύτερη μέθοδος είναι ο SA. Για τα $n = (50,75,100)$ ο καλύτερος είναι ο ILSA και για $n = 150$ έχουμε ξανά τον SA.

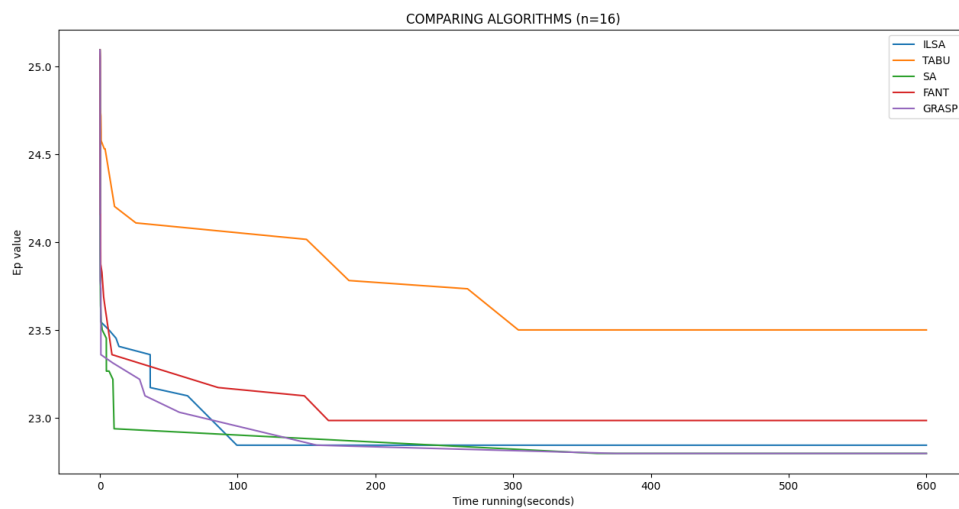
5.2 Αποτελέσματα Με Είσοδο Συγκεκριμένα Dataset

Στο δεύτερο στάδιο της μελέτης μας εργαστήκαμε με 4 πραγματικά dataset. Θα δο-
ύμε αν οι μέθοδοι μας "συμπεριφέρονται" με τον ίδιο τρόπο σε τυχαία και σε πραγ-
ματικά dataset. Τα 4 dataset που χρησιμοποιήθηκαν βρίσκονται σε αυτό το link:
<https://github.com/MixalisMourtzakis/ENERGY-CONSUMPTION-OF-DSM-SOLVED-AS-QAP.git>

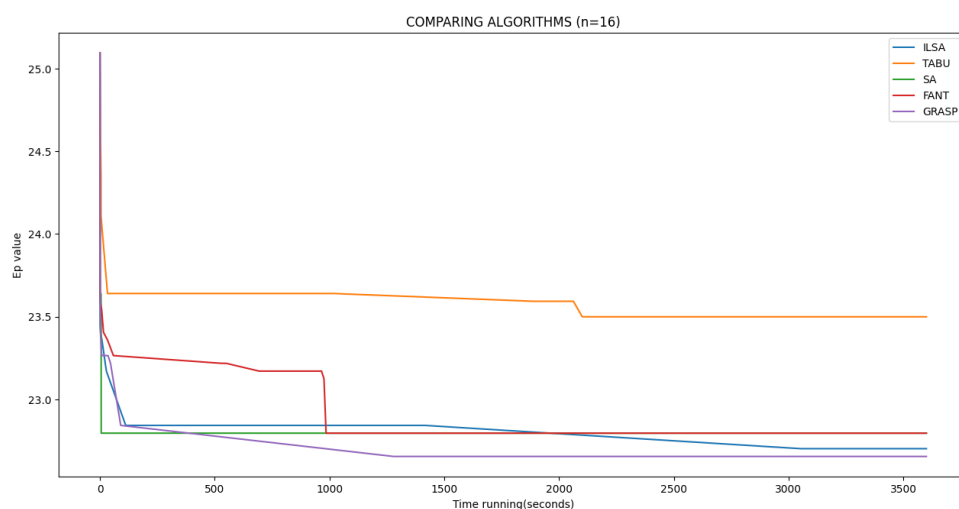
5.2.1 Διαγράμματα με είσοδο το Dataset : Sine 16

Το Dataset:Sine 16 έχει τα εξής χαρακτηριστικά :

$n = 16$ (Αριθμός Γραμμών) , $m = 30721$ (Αριθμός Κύκλων Ρολογιού)



Σχήμα 5.22: Αποτελέσματα για το Sine 16 σε δοκιμή 600 δευτερολέπτων.

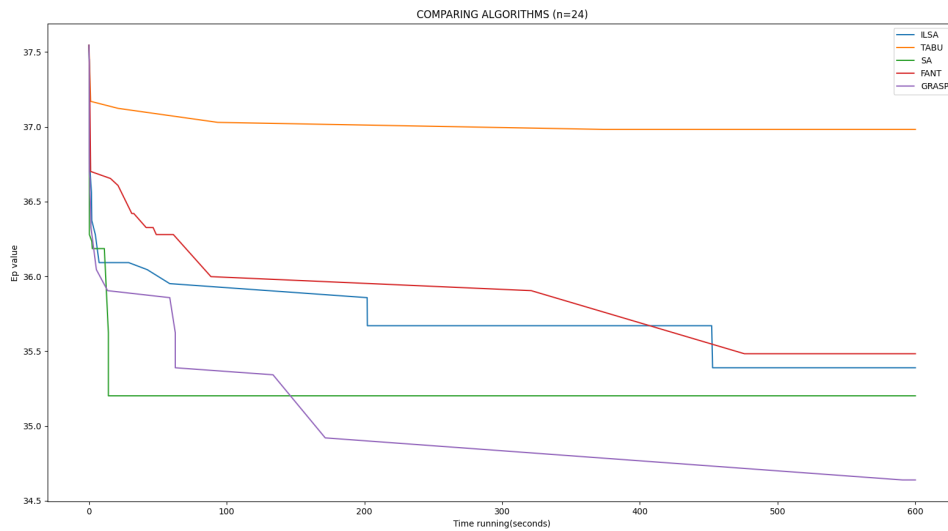


Σχήμα 5.23: Αποτελέσματα για το Sine 16 σε δοκιμή 3600 δευτερολέπτων.

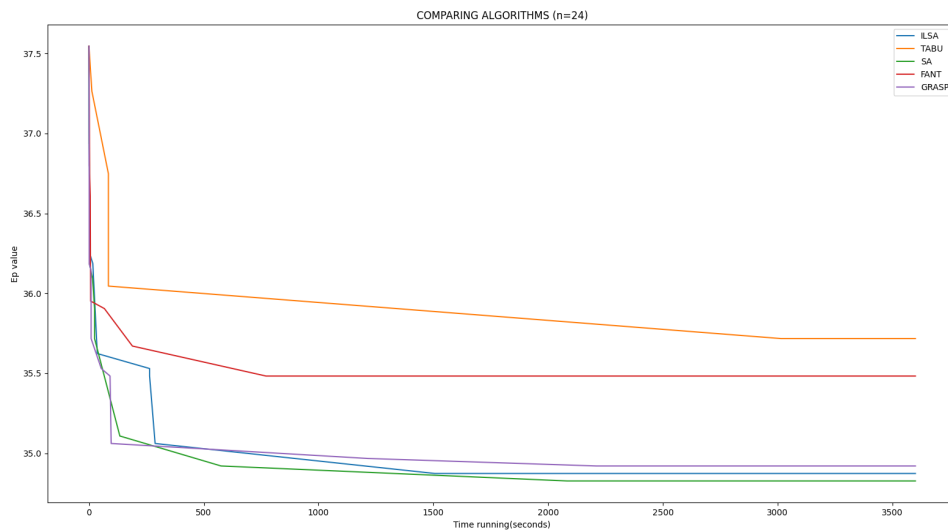
5.2.2 Διαγράμματα με είσοδο το Dataset : Sine 24

Το Dataset:Sine 24 έχει τα εξής χαρακτηριστικά :

$n = 24$ (Αριθμός Γραμμών) , $m = 30721$ (Αριθμός Κύκλων Ρολογιού)



Σχήμα 5.24: Αποτελέσματα για το Sine 24 σε δοκιμή 600 δευτερολέπτων.

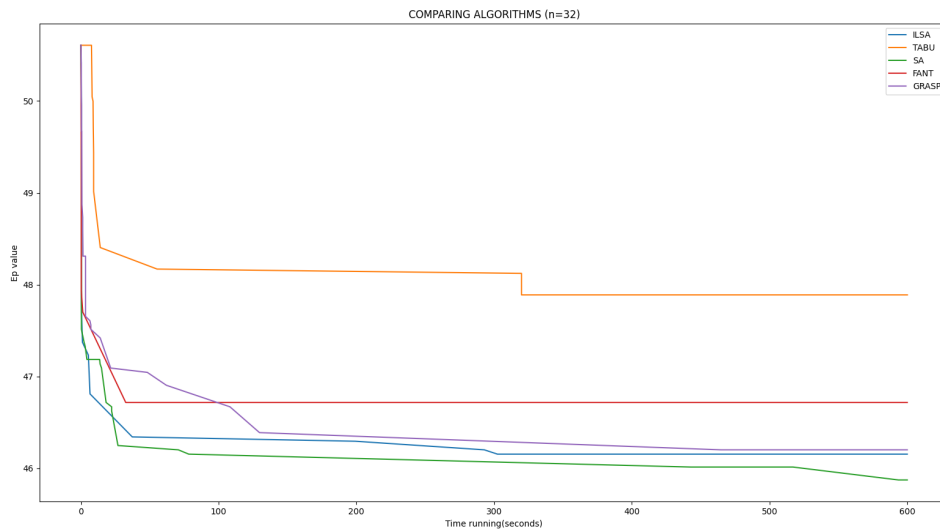


Σχήμα 5.25: Αποτελέσματα για το Sine 24 σε δοκιμή 3600 δευτερολέπτων.

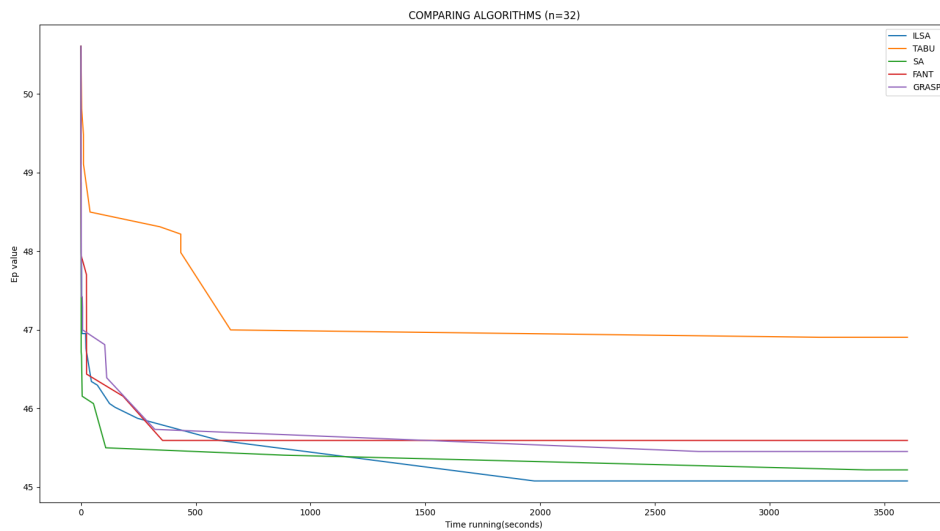
5.2.3 Διαγράμματα με είσοδο το Dataset : Sine 32

Το Dataset:Sine 32 έχει τα εξής χαρακτηριστικά :

$n = 32$ (Αριθμός Γραμμών) , $m = 30721$ (Αριθμός Κύκλων Ρολογιού)



Σχήμα 5.26: Αποτελέσματα για το Sine 32 σε δοκιμή 600 δευτερολέπτων.

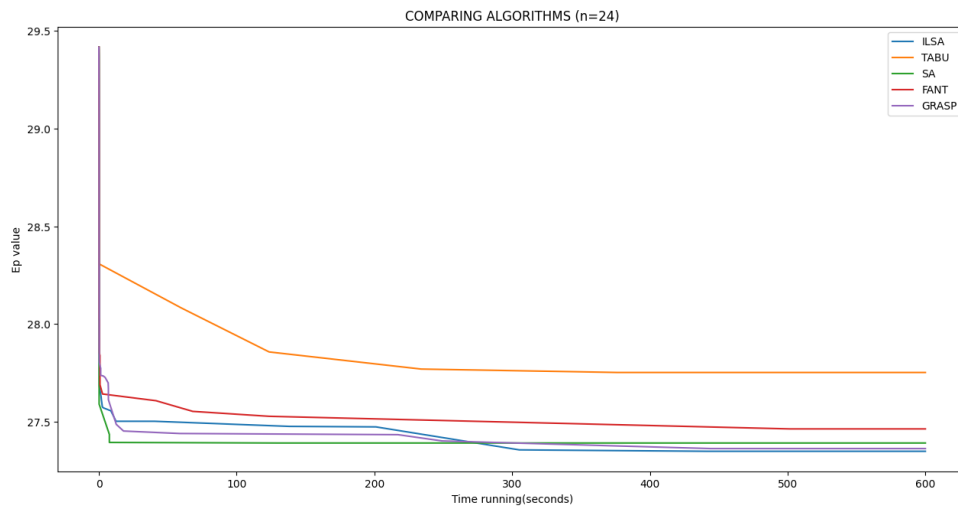


Σχήμα 5.27: Αποτελέσματα για το Sine 32 σε δοκιμή 3600 δευτερολέπτων.

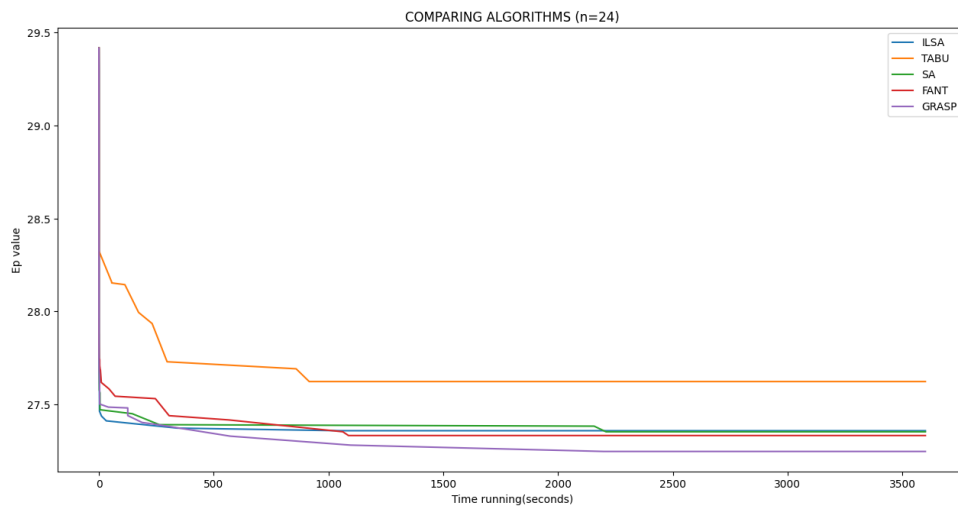
5.2.4 Διαγράμματα με είσοδο το Dataset : RGB 24

Το Dataset:RGB 24 έχει τα εξής χαρακτηριστικά:

$n = 24$ (Αριθμός Γραμμών) , $m = 81920$ (Αριθμός Κύκλων Ρολογιού)



Σχήμα 5.28: Αποτελέσματα για το RGB 24 σε δοκιμή 600 δευτερολέπτων.



Σχήμα 5.29: Αποτελέσματα για το RGB 24 σε δοκιμή 3600 δευτερολέπτων.

5.2.5 Σχολιασμός αποτελεσμάτων με είσοδο συγκεκριμένα Dataset και σύγκριση τους με εκείνα για τυχαία είσοδο

Αρχικά παρατηρούμε πως και εδώ τα αποτελέσματα μας ακολουθούν την μορφή που είδαμε παραπάνω. Δηλαδή έχουμε μια μεγάλη αρχική πτώση της τιμής και στην συνέχεια είτε παραμένουν στην ίδια τιμή, το οποίο είναι και το πιο σύνηθες, είτε βρίσκουν μια καλύτερη λύση πολύ κοντά στην προηγούμενη. Μια ενδιαφέρουσα παρατήρηση επίσης είναι πως με τυχαία είσοδο παρατηρούμε μια πτώση τιμής συνήθως μικρότερη της 1 μονάδας ενώ στα πραγματικά dataset που χρησιμοποιούμε βλέπουμε πτώση κάποιες φορές έως και 5 μονάδες. Αυτό είναι πολύ πιθανόν να οφείλεται πρώτον στο ότι τα πραγματικά dataset έχουν πολύ μεγαλύτερο m (αριθμός κύκλων ρολογιού) και δεύτερον στο ότι στην 1η περίπτωση έχουμε τυχαία κατασκευή δεδομένων ενώ στην 2η τα δεδομένα έχουν κάποια συσχέτιση μεταξύ τους εφόσον είναι πραγματικά. Επίσης άλλη μια ομοιότητα με τα αποτελέσματα που είδαμε παραπάνω είναι πως ο TABU είναι η χειρότερη μέθοδος και τον ακολουθεί ο FANT, ο οποίος είδαμε ότι είναι ο 2ος χειρότερος για τα χαμηλά n .

Τώρα ας σχολιάσουμε μεμονωμένα το κάθε dataset. Για το Sine 16 παρατηρούμε πως ο (SA - GRASP - ILSA) έχουν πάνω κάτω την ίδια απόδοση όπως είδαμε και παραπάνω στα αποτελέσματα των τυχαίων εισόδων για $n = (13,16,20)$. Για το Sine 32 έχουμε σχεδόν τα αποτελέσματα που είδαμε και παραπάνω για τα n κοντά στο $n = 32$ δηλαδή (SA - ILSA - GRASP - FANT - TABU) με μόνη διαφορά το γεγονός ότι ο (ILSA - GRASP) είναι πολύ κοντά στην απόδοσή. Για το RGB 24 βλέπουμε πως προσεγγίζει τα αποτελέσματα των χαμηλότερων n δηλαδή έχουμε τους (GRASP - ILSA - SA) να έχουν παρόμοια απόδοση. Τέλος στο Sine 24 έχουμε πάνω κάτω τα αναμενόμενα αποτελέσματα, με βάση τα προηγούμενα αποτελέσματα, με μόνη διαφορά το ότι ο GRASP έχει ένα αισθητό "προβάδισμα" στην απόδοση σε σύγκριση με τους άλλους.

Συμπερασματικά παρατηρούμε πως τα αποτελέσματα με είσοδο τυχαίων dataset είναι σχεδόν τα ίδια με αυτά που έχουν ως είσοδο πραγματικά dataset. Μετά από την ανάλυση όλων των αποτελεσμάτων συμπεραίνουμε πως οι πιο αξιόπιστες μέθοδοι για την επίλυση του προβλήματος μας είναι οι (GRASP - ILSA - SA) με τον καθένα να ξεχωρίζει ελάχιστα σε κάποιες περιπτώσεις.

Κεφάλαιο 6

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ

Συμπερασματικά, σε αυτή την εργασία αναλύσαμε το πρόβλημα της ελαχιστοποίησης κατανάλωσης ισχύος σε deep submicron technology buses και την διατύπωση του ως ένα quadratic assignment problem με την χρήση της οποίας προσπαθήσαμε να το επιλύσουμε. Όπως είδαμε στο 2ο κεφάλαιο το quadratic assignment problem είναι το πρόβλημα κατανομής εγκαταστάσεων σε ένα σύνολο τοποθεσιών με το κόστος να είναι συνάρτηση της απόστασης και της ροής μεταξύ των εγκαταστάσεων, συν το κόστος που σχετίζεται με την τοποθέτηση μιας εγκατάσταση σε μια συγκεκριμένη τοποθεσία. Χρησιμοποιήσαμε την trace διατύπωση του, θεωρώντας μηδενικό κόστος τοποθέτησης και το παραλληλίσουμε με το πρόβλημα ελαχιστοποίησης κατανάλωσης ισχύος σε deep submicron technology buses. Στην συνέχεια ελέγξαμε εάν το πρόβλημα μας μπορεί να επιλυθεί με την εξαντλητική μέθοδο και συμπεράναμε πως η εξαντλητική μέθοδος είναι εύχρηστη μόνο για τα $n < 10$ (αριθμός γραμμών). Για να αντιμετωπίσουμε λοιπόν το πρόβλημα μας και για τα $n > 10$ χρησιμοποιήσαμε μεθόδους επίλυσης που χρησιμοποιούνται για δύσκολες κατηγορίες προβλημάτων, όπως τα qap, τις λεγόμενες ευρετικές μεθόδους. Πιο συγκεκριμένα χρησιμοποιήσαμε τις εξής πέντε ευρετικές μεθόδους:

- Simulated Annealing (SA)
- Tabu Search (TS)
- Iterative Local Search Algorithm (ILSA)
- Greedy Randomized Adaptive Search Procedure (GRASP)
- Fast Ant System (FANT)

Η μελέτη μας έγινε σε 2 στάδια. Αρχικά μελετήθηκαν τυχαία dataset διαφορετικά για κάθε δοκιμή σε διάφορα n και στην συνέχεια έγινε μελέτη σε πραγματικά dataset. Επιπλέον η κύρια μελέτη μας έγινε για χρονικό διάστημα 10 λεπτών "run-time" κάθε μεθόδου για την σύγκριση της απόδοσης των μεθόδων. Πάραυτα έγινε και για χρονικό διάστημα 1 ώρας με σκοπό να παρατηρηθεί πως "συμπεριφέρονται" οι μέθοδοι. Τα αποτελέσματα της μελέτης μας ήταν τα εξής. Αρχικά παρατηρήσαμε πως τα τυχαία και τα πραγματικά dataset οι μέθοδοι "συμπεριφέρονται" με τον ίδιο τρόπο, δηλαδή έχουμε μια μεγάλη πτώση της τιμής στην αρχή και στην συνέχεια τείνει να μένει σταθερή σε μια

τιμή ή να μειώνεται ελάχιστα συγκριτικά με την αρχική πτώση. Τώρα όσο αφορά την απόδοση των αλγορίθμων είδαμε σε γενικές γραμμές πως οι 3 καλύτερες μέθοδοι είναι οι (GRASP - ILSA - SA) με τον καθένα να έχει ένα μικρό αβαντάζ σε συγκεκριμένες περιπτώσεις.

Είδαμε λοιπόν μετά την μελέτη μας και την ανάλυση των αποτελεσμάτων πως οι 3 καλύτερες μέθοδοι για το συγκεκριμένο πρόβλημα είναι οι (GRASP - ILSA - SA). Στο κεφάλαιο 4 που αναλύσαμε τις μεθόδους και τις παραμέτρους είδαμε πως η κάθε μέθοδος επηρεάζεται από κάποιες, ειδικές για αυτή, παραμέτρους. Στην παρούσα εργασία δεν έγινε εκτενής μελέτη για την εύρεση των βέλτιστων τιμών αυτών των παραμέτρων έτσι ώστε να έχουμε την μέγιστη πιθανή ελαχιστοποίηση της κατανάλωσης ισχύος στο πρόβλημα μας με την χρήση των ευρετικών αυτών μεθόδων. Επομένως προτείνεται να γίνει η χρήση κάποιας τεχνικής (π.χ. Μηχανική Μάθηση, (Βαθιά) Ενισχυμένη Μάθηση) έτσι ώστε να βρεθούν οι τιμές αυτές για την κάθε περίπτωση. Με χρήση τέτοιων τεχνικών είναι εφικτό να εκπαιδεύσουμε ένα μοντέλο, να κάνει τις βέλτιστες επιλογές αυτών των παραμέτρων ή έστω μια αρκετά καλή προσέγγισή τους. Είδαμε από την μελέτη που κάναμε πως τα αποτελέσματα μας είναι σχεδόν ανεξάρτητα από την επιλογή του dataset και επηρεάζονται κυρίως από την τιμή του n . Αυτό αποτελεί ένα μεγάλο "αβαντάζ" καθώς μπορούμε να εκπαιδεύσουμε το μοντέλο μας για ένα συγκεκριμένο dataset σε κάθε n και στην συνέχεια να το χρησιμοποιούμε για όλα τα παρόμοια dataset. Πέρα από αυτό όμως θα μπορούσε επίσης να μελετηθεί η χρήση υβριδικών ευρετικών μεθόδων. Εφόσον είδαμε από την παρούσα μελέτη ποιες μέθοδοι είναι πιο αποτελεσματικές θα μπορούσαμε να δοκιμάσουμε το συνδυασμό τους για την κατασκευή μιας ακόμα πιο αποτελεσματικής υβριδικής μεθόδου. Για παράδειγμα κατά την χρήση του GRASP μια ιδέα θα ήταν να αντικαταστήσουμε ή να χρησιμοποιήσουμε παράλληλα το κατασκευαστικό στάδιο του με τον ILSA ή SA.

Παράρτημα Α΄

Ακρωνύμια και Συντομογραφίες

LAN Local Area Network

QAP Quadratic Assignment Problem

ILP Integer Linear Programming

NP Nondeterministic Polynomial

TABU Tabu Search

SA Simulated Annealing

GA Genetic Algorithms

GRASP Greedy Randomized Adaptive Search Procedure

VNS Variable Neighborhood Search

AS Ant Systems

DSM Deep Submicron Model

ILSA Iterative Local Search Algorithm

FANT Fast Ant System

RAM Random Access Memory

BiQAP Biquadratic Assignment Problem

CPU Central Processing Unit

Bibliography

- [1] J. Sharma, *"Operations Research Theory and Application"*. MACMILAN Publishers, 2009.
- [2] P. P. Sotiriadis, *"Power Reduction Coding for Buses"*. Boston, MA: Springer US, 2004, pp. 177–205. [Online]. Available: https://doi.org/10.1007/1-4020-7836-6_7
- [3] P. Sotiriadis and A. Chandrakasan, "Bus energy reduction by transition pattern coding using a detailed deep submicrometer bus model," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 10, pp. 1280–1295, 2003.
- [4] Z. Drezner, P. M. Hahn, and E. D. Taillard, "Recent Advances for the Quadratic Assignment Problem with Special Emphasis on Instances that are Difficult for Meta-Heuristic Methods," *Annals of Operations Research*, vol. 139, no. 1, pp. 65–94, Oct. 2005. [Online]. Available: <http://link.springer.com/10.1007/s10479-005-3444-z>
- [5] S. Eilon, W. K. Holstein, R. L. Ackoff, and M. Tanenbaum, "Operations research," *Encyclopedia Britannica*, 2024. [Online]. Available: <https://www.britannica.com/topic/operations-research>
- [6] H. Lyeme and M. Selemani, *"Introduction to Operations Research: Theory and Applications"*. LAMBERT Academic Publishing, 2012.
- [7] [Online]. Available: https://www.hsor.org/what_is_or.cfm
- [8] G. B. Dantzig, "Origins of the simplex method," 1987. [Online]. Available: <https://www.osti.gov/biblio/6340929>
- [9] H. A. Taha, *"Operations Research: An Introduction"*. Pearson Education India, 2013.
- [10] C. VOGIATZHS, "The impact of the use of nonlinear programming techniques to the estimation of credit risk," Master's thesis, Aristotle University of Thessaloniki, 2019.
- [11] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960. [Online]. Available: <http://www.jstor.org/stable/1910129>
- [12] R. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, pp. 275–278, 09 1958.

- [13] J. W. Chinneck, "Practical optimization: A gentle introduction," *Systems and Computer Engineering*, 2015.
- [14] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, "*The Quadratic Assignment Problem*". Boston, MA: Springer US, 1998, pp. 1713–1809. [Online]. Available: https://doi.org/10.1007/978-1-4613-0303-9_27
- [15] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 657–690, 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221705008337>
- [16] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica: journal of the Econometric Society*, pp. 53–76, 1957.
- [17] C. Edwards, "The derivation of a greedy approximator for the koopmans-beckmann quadratic assignment problem," in *Proceedings of the 77-th Combinatorial Programming Conference (CP77)*, 1977, pp. 55–86.
- [18] C. S. Edwards, "A branch and bound algorithm for the koopmans-beckmann quadratic assignment problem," *Combinatorial optimization II*, pp. 35–52, 1980.
- [19] G. Finke, R. E. Burkard, and F. Rendl, "Quadratic assignment problems," in *North-Holland Mathematics Studies*. Elsevier, 1987, vol. 132, pp. 61–82.
- [20] P.-P. Sotiriadis and A. P. Chandrakasan, "A bus energy model for deep submicron technology," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 10, no. 3, pp. 341–350, 2002.
- [21] P. P. Sotiriadis and A. Chandrakasan, "Power estimation and power optimal communication in deep submicron buses: Analytical models and statistical measures," *Journal of Circuits, Systems, and Computers*, vol. 11, no. 06, pp. 637–658, 2002.
- [22] K. Papalamprou and L. Pitsoulis, "A quadratic assignment formulation for minimizing power consumption in deep submicron technology buses," *ORIJ*, 2023.
- [23] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in vlsis," *IEEE Transactions on Electron Devices*, vol. 40, no. 1, pp. 118–124, 1993.
- [24] K. T. Tang and E. G. Friedman, "Peak noise prediction in loosely coupled interconnect [vlsi circuits]," in *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1. IEEE, 1999, pp. 541–544.
- [25] T. Sakurai, S. Kobayashi, and M. Noda, "Simple expressions for interconnection delay, coupling and crosstalk in vlsi's," in *1991 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1991, pp. 2375–2378.
- [26] M. Becer and I. N. Hajj, "An analytical model for delay and crosstalk estimation in interconnects under general switching conditions," in *ICECS 2000. 7th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2. IEEE, 2000, pp. 831–834.

- [27] S. Das, W. Smith, and C. Paul, "Modeling of data bus structures using numerical methods," in *1993 International Symposium on Electromagnetic Compatibility*. IEEE, 1993, pp. 409–414.
- [28] L. Pileggi, "Coping with rcl interconnect design headaches," in *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*. IEEE, 1995, pp. 246–253.
- [29] J. A. Davis and J. D. Meindl, "Compact distributed rlc interconnect models-part ii: Coupled line transient expressions and peak crosstalk in multilevel networks," *IEEE Transactions on Electron Devices*, vol. 47, no. 11, pp. 2078–2087, 2000.
- [30] T. Sakurai, "Design challenges for 0.1 um and beyond: Embedded tutorial," in *Proceedings of the 2000 Asia and South Pacific Design Automation Conference*, 2000, pp. 553–558.
- [31] S. Lin and J. P. Lillis, *"Interconnect Analysis and Synthesis"*. John Wiley & Sons, Inc., 1999.
- [32] K. Yamashita and S. Odanaka, "Interconnect scaling scenario using a chip level interconnect model," *IEEE transactions on electron devices*, vol. 47, no. 1, pp. 90–96, 2000.
- [33] N. Menezes and L. Pileggi, "Analyzing on-chips interconnect effects," *Design of High Performance Microprocessor Circuits*, 2001.
- [34] L. Macchiarulo, E. Macii, and M. Poncino, "Low-energy for deep-submicron address buses," in *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, ser. ISLPED '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 176–181. [Online]. Available: <https://doi.org/10.1145/383082.383127>
- [35] H. R. Lourenço, O. C. Martin, and T. Stützle, *"Iterated Local Search: Framework and Applications"*. Cham: Springer International Publishing, 2010, pp. 363–397. [Online]. Available: https://doi.org/10.1007/978-3-319-91086-4_5
- [36] Y. Li, P. M. Pardalos, and M. G. Resende, "A greedy randomized adaptive search procedure for the quadratic assignment problem," *Quadratic assignment and related problems*, vol. 16, pp. 237–261, 1993.
- [37] E. Taillard, "Robust taboo search for the quadratic assignment problem," *Parallel Computing*, vol. 17, no. 4, pp. 443–455, 1991. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167819105801474>
- [38] R. Battiti and G. Tecchiolli, "Simulated annealing and tabu search in the long run: A comparison on qap tasks," *Computers & Mathematics with Applications*, vol. 28, no. 6, pp. 1–8, 1994. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0898122194001472>
- [39] M. G. C. Resende, P. M. Pardalos, and Y. Li, "Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problems using grasp," *ACM Trans. Math. Softw.*, vol. 22, no. 1, p. 104–118, 1996. [Online]. Available: <https://doi.org/10.1145/225545.225553>