

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ Η/Υ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**“ΥΛΟΠΟΙΗΣΗ ΕΝΟΣ ΚΑΤΑΝΕΜΗΜΕΝΟΥ
SOFTWARE FRAMEWORK ΓΙΑ ΤΟΝ ΣΥΝΤΟΝΙΣΜΟ
ΠΟΛΥΠΡΑΚΤΟΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΤΗ
ΧΡΗΣΗ ΔΗΜΟΠΡΑΣΙΩΝ”**



ΣΚΟΥΡΤΗΣ ΒΑΣΙΛΕΙΟΣ

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κο Λαγουδάκη για την υποστήριξη και την πολύτιμη βοήθειά του στην πραγματοποίηση αυτής της διπλωματικής εργασίας καθώς και την μεγάλη υπομονή που έδειξε σε κάποιες περιπτώσεις.

Επίσης θα ήθελα να ευχαριστήσω τους καθηγητές της εξεταστικής επιτροπής κο Σαμολαδά και κα Μανιά για τον χρόνο που αφιέρωσαν για την ανάγνωση αυτού του κειμένου καθώς και για τις εποικοδομητικές παρατηρήσεις τους.

Ακόμα θα ήθελα να ευχαριστήσω το προσωπικό και τους συμφοιτητές μου στο εργαστήριο για την βοήθειά τους.

Με την διπλωματική αυτή εργασία κλείνει το κεφάλαιο της φοιτητικής μου ζωής. Για τον λόγο αυτό θα ήθελα να κάνω μία αναδρομή και να ευχαριστήσω όλους τους ανθρώπους που με βοήθησαν στην προσπάθειά μου να γίνω μηχανικός.

Στο ακαδημαϊκό κομμάτι θα ήθελα να εκφράσω τις ευχαριστίες μου στον κο Δόλλα επειδή μας έδωσε για πρώτη φορά την ευκαιρία να δοκιμάσουμε τις δυνάμεις μας σε μία μεγάλη και ιδιαίτερα ενδιαφέρουσα εργασία, η οποία είχε ως αποτέλεσμα τα Contagibooooooooots!!!!

Δεν θα μπορούσα να μην αναφερθώ ονομαστικά στις palioseires, τους 7 μύθους που θα ζουν για πάντα και θα κατακτούν συνεχώς το πρωτάθλημα στις καρδιές των φιλάθλων, όπως και στους los griegos, τους συνοδοιπόρους μου στην μηνιαία πολιορκία και τελική κατάκτηση της Ιβηρικής χερσονήσου. Επίσης θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που έκαναν την διαμονή μου στα Χανιά πιο ευχάριστη καθώς και τον σύλλογο ΒΕΣΤ που μου έδωσε την ευκαιρία να ζήσω αξέχαστες στιγμές και να αποκτήσω απίστευτες εμπειρίες. Κάποιοι άνθρωποι λένε ότι τα φοιτητικά χρόνια είναι τα καλύτερα. Η συνισταμένη των παραπάνω με κάνει να το πιστεύω.

Τέλος, το μεγαλύτερο ευχαριστώ στην οικογένειά μου, τους γονείς μου Σωτήρη και Δέσποινα και την αδελφή μου Γεωργία, για την αμέριστη συμπαράσταση και υποστήριξη τους καθώς και για όλες τις θυσίες που έκαναν για να καταφέρω να ολοκληρώσω τις σπουδές μου στο τμήμα των Ηλεκτρονικών Μηχανικών & Μηχανικών Η/Υ.

Περιεχόμενα

1	Εισαγωγή	10
1.1	Γενικά	10
1.2	Περιγραφή και προσέγγιση προβλήματος	12
1.3	Σχετική εργασία	12
2	Ο αλγόριθμος δημοπρασίας	14
2.1	Σχετική ορολογία	14
2.2	Multi-Agent coordination	15
2.3	Team Objectives for Multi-Agent Task Assignment	16
2.4	Auction Framework	17
2.5	Ο αλγόριθμος	18
2.6	Παραγωγή Bidding κανόνων	19
2.7	Οι φάσεις του γύρου δημοπρασίας	21
3	Ο Πυρήνας	23
3.1	Καθορισμός των bidders	24
3.2	Network set-up	24
3.3	Πρωτόκολλο Δημοπρασίας	25
3.3.1	Διαδικασία σύνδεσης	26
3.3.2	Πραγματοποίηση Δημοπρασιών	29
3.4	Η Φάση της Ανάθεσης	32
3.5	Η Ρουτίνα του Auctioneer	32
3.6	Η ρουτίνα του bidder	36
4	Εφαρμογές	42
4.1	Καθορισμός διαδρομών ομάδας από robots	42
4.1.1	Περιγραφή σεναρίου	43
4.1.2	Υλοποίηση	44
4.1.3	Παράδειγμα ανάθεσης κόμβων	48
4.2	Computer File Sharing	53
4.2.1	Περιγραφή σεναρίου	54
4.2.2	Υλοποίηση	55

5	Αποτελέσματα πειραμάτων	58
5.1	Καθορισμός διαδρομών ομάδας robots	58
5.1.1	Ανάθεση 100 κόμβων	63
5.1.2	Ανάθεση 500 κόμβων	71
5.1.3	Ανάθεση 1000 κόμβων	79
5.2	Computer File Sharing	87
5.2.1	Πείραμα Α	88
5.2.2	Πείραμα Β	90
5.2.3	Πείραμα Γ	92
6	Συμπεράσματα	95
6.1	Ανακεφαλαίωση και Συμπεράσματα	95
6.2	Μελλοντική Εργασία	96

Κατάλογος Πινάκων

2.1	Auction Algorithm's pseudocode	18
4.1	Όλα τα βήματα της δημοπρασίας μέχρι την τελική ανάθεση του σχήματος 4.4	53
5.1	Πίνακας τιμών κόστους για τις διαδρομές των σχημάτων 5.2, 5.3 και 5.4	60
5.2	Αποτελέσματα 1ου πειράματος	64
5.3	Αποτελέσματα 2ου πειράματος	68
5.4	Αποτελέσματα 3ου πειράματος	72
5.5	Αποτελέσματα 4ου πειράματος	76
5.6	Αποτελέσματα 5ου πειράματος	80
5.7	Αποτελέσματα 6ου πειράματος	84
5.8	Τα στοιχεία της δημοπρασίας του πειράματος Α	88
5.9	Οι τιμές latency στο πείραμα Α	89
5.10	Ανάθεση αρχείων και χρόνοι μεταφοράς τους στο πείραμα Α	89
5.11	Τα στοιχεία της δημοπρασίας του πειράματος Β	91
5.12	Οι τιμές latency στο πείραμα Β	91
5.13	Ανάθεση αρχείων και χρόνοι μεταφοράς τους στο πείραμα Β	92
5.14	Τα στοιχεία της δημοπρασίας του πειράματος Γ	93
5.15	Οι τιμές latency στο πείραμα Γ	93
5.16	Ανάθεση αρχείων και χρόνοι μεταφοράς τους στο πείραμα Γ	94

Κατάλογος Σχημάτων

3.1	Ο auctioneer και οι bidders είναι συνδεδεμένοι σε ένα δίκτυο Αστέρα με τον auctioneer να είναι ο κεντρικός κόμβος	25
3.2	Τρόπος λειτουργίας του Auction Request Manager	29
3.3	Data overlap	30
3.4	Η ουρά εξυπηρέτησης του bidder	31
3.5	Το χρονοδιάγραμμα της συνολικής διαδικασίας	33
3.6	Ο auctioneer δημιουργεί 4 threads για τους 4 bidders της δημοπρασίας του. Τα threads συνδέονται στους bidders και αρχίζουν να αλληλεπιδρούν μαζί τους.	34
3.7	Τα auctioneer threads εκτελούνται παράλληλα και χρησιμοποιούν κοινές μεταβλητές και δομές δεδομένων για να συγχρονίζονται και να καταλήγουν σε κοινά αποτελέσματα	35
3.8	Η συνολική λειτουργία του bidder. Η συνεχής γραμμή που συνδέει το Auction Request Manager thread με το actual bidder thread σημαίνει ότι το bidder thread εκτελείται ενώ η διακεκομμένη ότι βρίσκεται σε αναμονή.	38
3.9	Ένα στιγμιότυπο του προγράμματος του πράκτορα σε επίπεδο thread. Η συνεχόμενη γραμμή σε κάποιο thread σημαίνει ότι αυτό εκτελείται ενώ η διακεκομμένη ότι είναι ‘παγωμένο’ και βρίσκεται σε αναμονή.	40
3.10	Το διάγραμμα ροής του προγράμματος του πράκτορα	41
4.1	Διαδικασία υπολογισμού του bid για το MINISUM team objective	46
4.2	Διαδικασία υπολογισμού του bid για το MINIMAX team objective	47
4.3	Διαδικασία υπολογισμού του bid για το MINIAVE team objective	48
4.4	Η επιθυμητή ανάθεση των 10 κόμβων στην ομάδα που αποτελείται από τα robots R_1, R_2 και R_3	49
4.5	Ο πρώτος γύρος της δημοπρασίας για την ανάθεση των κόμβων	50
4.6	Σε αυτό το γύρο τα robots R_1 και R_3 συναγωνίζονται για τον κόμβο 7	51
4.7	Τα robots R_1 και R_2 διεκδικούν τον κόμβο 1 με νικητή το R_2	51
4.8	Το τελικό αποτέλεσμα	52

4.9 Όλοι οι υπολογιστές του δικτύου λειτουργούν σαν ένα τεράστιο αποθηκευτικό μέσο	54
5.1 Οι θέσεις των 15 κόμβων και του robot στο χάρτη διαστάσεων 20x20.	61
5.2 Η διαδρομή που σχηματίζεται σύμφωνα με το MiniSum κανόνα	61
5.3 Η διαδρομή που σχηματίζεται σύμφωνα με το MiniMax κανόνα	62
5.4 Η διαδρομή που σχηματίζεται σύμφωνα με το MiniAve κανόνα .	62
5.5 Οι τιμές των τριών αναθέσεων του πειράματος 1 για τα πεδία SUM, MAX και AVE.	64
5.6 Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 1. .	64
5.7 Οι θέσεις των 100 κόμβων και των 3 robots στο χάρτη για το πείραμα 1.	65
5.8 Οι διαδρομές των 3 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.7.	65
5.9 Οι διαδρομές των 3 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.7.	66
5.10 Οι διαδρομές των 3 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.7.	66
5.11 Οι τιμές των τριών αναθέσεων του πειράματος 2 για τα πεδία SUM, MAX και AVE.	68
5.12 Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 2. .	68
5.13 Οι θέσεις των 100 κόμβων και των 4 robots στο χάρτη για το πείραμα 2.	69
5.14 Οι διαδρομές των 4 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.13.	69
5.15 Οι διαδρομές των 4 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.13.	70
5.16 Οι διαδρομές των 4 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.13.	70
5.17 Οι τιμές των τριών αναθέσεων του πειράματος 3 για τα πεδία SUM, MAX και AVE.	72
5.18 Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 3. .	72
5.19 Οι θέσεις των 500 κόμβων και των 3 robots στο χάρτη για το πείραμα 3.	73
5.20 Οι διαδρομές των 3 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.19.	73
5.21 Οι διαδρομές των 3 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.19.	74
5.22 Οι διαδρομές των 3 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.19.	74
5.23 Οι τιμές των τριών αναθέσεων του πειράματος 4 για τα πεδία SUM, MAX και AVE.	76
5.24 Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 4. .	76

5.25	Οι θέσεις των 500 κόμβων και των 4 robots στο χάρτη για το πείραμα 4.	77
5.26	Οι διαδρομές των 4 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.25.	77
5.27	Οι διαδρομές των 4 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.25.	78
5.28	Οι διαδρομές των 4 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.25.	78
5.29	Οι τιμές των τριών αναθέσεων του πειράματος 5 για τα πεδία SUM, MAX και AVE.	80
5.30	Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 5. .	80
5.31	Οι θέσεις των 1000 κόμβων και των 3 robots στο χάρτη για το πείραμα 5.	81
5.32	Οι διαδρομές των 3 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.31.	81
5.33	Οι διαδρομές των 3 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.31.	82
5.34	Οι διαδρομές των 3 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.31.	82
5.35	Οι τιμές των τριών αναθέσεων του πειράματος 6 για τα πεδία SUM, MAX και AVE.	84
5.36	Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 6. .	84
5.37	Οι θέσεις των 1000 κόμβων και των 4 robots στο χάρτη για το πείραμα 6.	85
5.38	Οι διαδρομές των 4 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.37.	85
5.39	Οι διαδρομές των 4 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.37.	86
5.40	Οι διαδρομές των 4 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.37.	86

Γενικά

Μία πολύ καλή και αποδοτική μέθοδος για τον συντονισμό μιας ομάδας πρακτόρων θεωρείται η ανάθεση εργασιών μέσω εικονικών δημοπρασιών. Στα πλαίσια της διπλωματικής εργασίας αυτής υλοποιήθηκε ένα γενικευμένο πλαίσιο λογισμικού για την υποστήριξη μεθόδων συντονισμού με τη χρήση δημοπρασιών σε κατανεμημένα πολυπρακτορικά συστήματα, όπου η επικοινωνία επιτυγχάνεται μέσω ενσύρματου ή ασύρματου TCP/IP δικτύου. Ο πυρήνας του πλαισίου μπορεί να χρησιμοποιηθεί αυτούσιος σε πολλές εφαρμογές για να αναπτυχθούν δημοπρατικές μέθοδοι συντονισμού μόνο με την προσθήκη εξειδικευμένων ρουτίνων που σχετίζονται με το είδος των εργασιών και το κόστος εκτέλεσής τους στην εκάστοτε εφαρμογή.

Η λειτουργικότητα του πλαισίου επιδεικνύεται σε δύο ενδεικτικές εφαρμογές. Στην πρώτη, οι πράκτορες (κόμβοι του δικτύου) είναι κινητά ρομποτικά οχήματα που πρέπει να επισκεφθούν ένα σύνολο θέσεων σε κάποια χαρτογραφημένη περιοχή. Η διαμόρφωση των διαδρομών τους γίνεται μέσω δημοπρασιών και έχουν ως στόχο τη βελτιστοποίηση μίας συνάρτησης ομαδικού κόστους. Στη δεύτερη εφαρμογή, οι πράκτορες είναι υπολογιστές που λειτουργούν ταυτόχρονα ως αποθηκευτικές μονάδες και τερματικά πρόσβασης. Η μεταφορά αρχείων από τον ένα στον άλλο ανάλογα με τη ζήτηση των χρηστών συντονίζεται μέσω δημοπρασιών ώστε να επιτυγχάνεται βελτιστοποίηση του μέσου χρόνου μετάδοσης.

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά

Οι ομάδες πρακτόρων έχουν αρχίσει να θεωρούνται ως μία πολύ καλή εναλλακτική επιλογή έναντι των απλών πρακτόρων για την επίτευξη ποικίλων δύσκολων αποστολών όπως την εξερεύνηση κάποιας άγνωστης περιοχής. Σε πολλές εφαρμογές μπορεί να χρησιμοποιηθεί πιο αποδοτικά ομάδα πρακτόρων σε σχέση με ένα απλό πράκτορα. Η ομάδα των πρακτόρων μπορεί να ολοκληρώσει μία αποστολή πολύ ταχύτερα από ένα πράκτορα διαιρώντας την αποστολή σε επιμέρους εργασίες και εκτελώντας αυτές παράλληλα. Οι ομάδες πρακτόρων πλεονεκτούν σε αρκετά σημεία συγκριτικά με τους απλούς πράκτορες, όπως σε σταθερότητα και ανθεκτικότητα λόγω πληθώρας πρακτόρων, σε αποδοτικότητα λόγω παράλληλης πραγματοποίησης των εργασιών και σε ευελιξία λόγω δυνατότητας ανασχηματισμού των ομάδων. Πολύ σημαντικός παράγοντας όμως για την επιτυχία της ομάδας των πρακτόρων είναι ο αποδοτικός συντονισμός τους.

Ο συντονισμός των πρακτόρων μπορεί να πραγματοποιηθεί είτε μέσω κεντροποιημένου (centralised) είτε μέσω αποκεντροποιημένου (decentralised) ελέγχου. Ο κεντροποιημένος έλεγχος θεωρείται μη αποδοτικός εξαιτίας της

μεγάλης απαιτούμενης υπολογιστικής ισχύος καθώς και της μεγάλης ευαισθησίας του συστήματος λόγω εξάρτησης σε ένα κεντρικό συντονιστή, η απώλεια του οποίου οδηγεί σε αποτυχία της ομάδας. Επίσης, ο κεντρικοποιημένος έλεγχος μειονεκτεί ιδιαίτερα και εξαιτίας της κακής του απόκρισης σε οποιεσδήποτε αλλαγές. Ο αποκεντρικοποιημένος έλεγχος δεν έχει αυτά τα μειονεκτήματα αλλά μπορεί να οδηγήσει σε μη βέλτιστες λύσεις. Σε αρκετές έρευνες εξετάζονται τα μοντέλα των δημοπρασιών σαν αποδοτικές μέθοδοι για αποκεντρικοποιημένο έλεγχο πολυπρακτορικών συστημάτων. Στις δημοπρασίες οι απαιτήσεις υπολογιστικής ισχύος είναι αρκετά μικρότερες από τον κεντρικοποιημένο έλεγχο αφού η πληροφορία συμπυκνώνεται σε αριθμητικά bids, τα οποία υπολογίζονται παράλληλα από τους πράκτορες. Επιπλέον, οι δημοπρασίες δίνουν λύσεις αρκετά κοντά στις βέλτιστες.

Οι μετρήσεις και οι μελέτες που επαληθεύουν τα πλεονεκτήματα του μοντέλου των δημοπρασιών για συντονισμό πολυπρακτορικών συστημάτων, ενθαρρύνοντας έτσι την αντίστοιχη χρήση των δημοπρασιών, πραγματοποιούνται κυρίως σε ειδικούς προσομοιωτές όπως ο TeamBots [7]. Επίσης, οι όποιες υλοποιήσεις συστημάτων κατανομής επιμέρους εργασιών μίας αποστολής σε ομάδα πρακτόρων με τη χρήση δημοπρασιών βασίζονται σε ένα συγκεκριμένο κεντρικό υπολογιστή που λειτουργεί ως auctioneer. Στη συγκεκριμένη περίπτωση διατηρείται το μειονέκτημα του κεντρικοποιημένου ελέγχου όπου η απώλεια του auctioneer οδηγεί σε αποτυχία της ομάδας.

Στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε ένα software framework που συντονίζει ομάδα πρακτόρων με τη χρήση δημοπρασιών χωρίς να βασίζεται σε κάποιον συγκεκριμένο κεντρικό auctioneer. Επιπλέον έχει την δυνατότητα να υποστηρίζει την πραγματοποίηση δημοπρασιών για διάφορες εφαρμογές.

1.2 Περιγραφή και προσέγγιση προβλήματος

Η επιλογή του αποκεντριοποιημένου ελέγχου των πρακτόρων δημιουργεί την ανάγκη για την υλοποίηση ενός συστήματος ανθεκτικού σε διάφορες μεταβολές και βλάβες, το οποίο εγγυάται την επίτευξη των αποστολών που αναλαμβάνει η ομάδα των πρακτόρων. Δημιουργείται δηλαδή η ανάγκη για ένα σύστημα που θα μπορεί να πραγματοποιήσει την αποστολή που αναλαμβάνει κάτω από οποιεσδήποτε συνθήκες και ασχέτως από αλλαγές καταστάσεων. Μία επιπλέον συνθήκη που θέλουμε να ικανοποιείται από το σύστημα συντονισμού των πρακτόρων είναι η δυνατότητα υποστήριξης δημοπρασιών διαφόρων εφαρμογών.

Το σύστημα που υλοποιήθηκε για τον συντονισμό της ομάδας των πρακτόρων στα πλαίσια αυτής της διπλωματικής ικανοποιεί τα παραπάνω ζητούμενα. Η ανθεκτικότητα του συστήματος επιτυγχάνεται με τον σχεδιασμό των πρακτόρων ώστε να μπορούν να λειτουργήσουν και ως bidders αλλά και ως auctioneers ταυτόχρονα. Με αυτό τον τρόπο επιτυγχάνεται μία ισοδυναμία των μελών της ομάδας και μία ανεξαρτητοποίηση από κάποιον κεντρικό πράκτορα, η οποία οδηγεί σε πραγματοποίηση της αποστολής. Ο καθορισμός του auctioneer για την δημοπράτηση των επιμέρους εργασιών μίας αποστολής εξαρτάται από την αντίστοιχη εφαρμογή.

Τα γνωρίσματα που χαρακτηρίζουν το παραπάνω σύστημα είναι τα πλεονεκτήματα της χρήσης ομάδας πρακτόρων, δηλαδή αποδοτικότητα και ευελιξία, καθώς και ανθεκτικότητα σε βλάβες και μεταβολές και γενικότητα στη χρήση με την δυνατότητα χρησιμοποίησής του σε πλήθος εφαρμογών.

1.3 Σχετική εργασία

Γενικά δεν έχουν πραγματοποιηθεί πολλές υλοποιήσεις κατανεμημένων πολυπρακτορικών συστημάτων. Από τις υλοποιήσεις πάντως που έχουν πραγματο-

ποιηθεί, οι περισσότερες στηρίζονται σε κάποιο κεντρικό υπολογιστή που λειτουργεί ως auctioneer λαμβάνοντας τα bids των πρακτόρων και ενημερώνοντάς τους για την εξέλιξη της δημοπρασίας. Επίσης, αρκετή έρευνα γίνεται σχετικά με την κατανομή του φόρτου εργασίας του auctioneer περί της ανάδειξης του νικητή στους bidders. Ο σχεδιασμός δηλαδή του συστήματος ως κατανεμημένου επικεντρώνεται στον διαμοιρασμό των υπολογισμών και του συνολικού έργου για την ανάδειξη του νικητή σε όλους τους πράκτορες επιβαρύνοντας λιγότερο τον auctioneer.

Η πλησιέστερη εργασία στην δική μας υλοποίηση είναι μία εφαρμογή ενός κατανεμημένου μοντέλου scheduler που χρησιμοποιεί δημοπρασίες για τον χρονοπρογραμματισμό εργασιών (auction-based distributed scheduling model for a dynamic job shop environment) [8]. Σε αυτή την υλοποίηση έχουμε ένα σύστημα χρονοπρογραμματισμού (scheduling system) που αποτελείται από δύο είδη πρακτόρων, τα jobs και τα μηχανήματα. Κάθε πράκτορας είναι μία διεργασία με την δική της τοπική μνήμη και χωρίς shared memory και αναπαριστά είτε ένα job είτε ένα μηχανήμα στο σύστημα.

Κάθε μονάδα ή θυρίδα χρόνου (time slot) σε κάθε μηχανήμα θεωρείται σαν δημοπρατούμενο αντικείμενο και κάθε μηχανήμα θεωρείται σαν ένας auctioneer. Κάθε job είναι ένας bidder και συμμετέχει στις δημοπρασίες που πραγματοποιούν τα μηχανήματα, τα οποία απαιτούνται για τις λειτουργίες του job. Όποτε κάποιο μηχανήμα είναι idle και διαθέσιμο, πραγματοποιεί μία δημοπρασία για τις χρονικές θυρίδες από εκείνη τη στιγμή μέχρι το τέλος του χρονικού ορίζοντα. Σκοπός κάθε πράκτορα job είναι να βρει τον τρόπο πραγματοποίησής του μέσα από το κατάστημα που αποτελείται από πράκτορες μηχανήματα ελαχιστοποιώντας το πιθανό κόστος.

Κεφάλαιο 2

Ο αλγόριθμος δημοπρασίας

Σε αυτό το κεφάλαιο παρουσιάζεται ο αλγόριθμος που χρησιμοποιήθηκε στη συγκεκριμένη διπλωματική εργασία για την ανάθεση εργασιών σε ομάδα πρακτόρων μέσω δημοπρασίας. Αρχικά εξηγείται συνοπτικά η σχετική ορολογία και στην συνέχεια αναλύεται ο αλγόριθμος.

2.1 Σχετική ορολογία

Tasks ή εργασίες είναι τα επιμέρους κομμάτια στα οποία χωρίζεται μία αποστολή και η συνολική πραγματοποίηση τους αποτελεί ολοκλήρωση της αποστολής.

Auction ή Δημοπρασία είναι η διαδικασία κατά την οποία ανατίθενται M εργασίες - tasks σε N πράκτορες - agents. Οι πράκτορες κάνουν προσφορές ακολουθώντας κανόνες δημοπρασίας ώστε να αναλάβουν τις εργασίες.

Auction round είναι ένας γύρος της δημοπρασίας κατά τον οποίο ανατίθεται μία εργασία σε έναν πράκτορα.

Bid είναι η προσφορά που κάνει ένας πράκτορας που συμμετέχει στη δημοπρασία για να αναλάβει κάποια εργασία.

Auctioneer είναι ο πράκτορας, ο οποίος είναι υπεύθυνος για την πραγματοποίηση της δημοπρασίας και ο οποίος διευθύνει την όλη διαδικασία.

Bidder ονομάζεται κάθε πράκτορας που συμμετέχει στην δημοπρασία και κάνει προσφορές ώστε να αναλάβει εργασίες.

Cost Function είναι η συνάρτηση που καθορίζει το κόστος κάθε ενέργειας για τους πράκτορες.

Agent Objective Function για κάποιον πράκτορα είναι η συνάρτηση που υπολογίζει το κόστος πραγματοποίησης των εργασιών που του ανατέθηκαν.

Team Objective Function είναι η συνάρτηση που καθορίζει το κόστος της πραγματοποίησης όλων των εργασιών σε μία ομαδική ανάθεση.

Team Objective είναι ο στόχος μίας ομάδας πρακτόρων και συνήθως εκφράζεται ως η ελαχιστοποίηση της συνάρτησης Team Objective Function για μία αποστολή.

2.2 Multi-Agent coordination

Έστω ότι υπάρχει μία ομάδα N πρακτόρων $R = \{r_1, r_2, \dots, r_N\}$ που θέλουν να καταναείμουν μεταξύ τους M εργασίες $T = \{t_1, t_2, \dots, t_M\}$ ¹. Για κάθε πράκτορα r_i υπάρχουν τα σύνολα $S_{r_i} \subseteq T$ και $A_{r_i} \subseteq S_{r_i}$. Το S_{r_i} περιέχει τις εργασίες, τις οποίες μπορεί να αναλάβει ο πράκτορας r_i ενώ το A_{r_i} περιέχει τις εργασίες που έχουν ήδη ανατεθεί στον r_i .

¹μπορεί να είναι είτε $N \geq M$ ή $N < M$

Έστω η συνάρτηση $g_{r_i}(r_i, A_{r_i})$, η οποία μετράει την απόδοση του r_i για το σύνολο των εργασιών A_{r_i} και επιστρέφει το κόστος της πραγματοποίησης των εργασιών του A_{r_i} από τον πράκτορα r_i (Agent Objective Function). Επίσης η $f(g_{r_1}(r_1, A_{r_1}), \dots, g_{r_N}(r_N, A_{r_N}))$ υπολογίζει την συνολική απόδοση της ομάδας σαν συνάρτηση των επιμέρους συναρτήσεων των πρακτόρων και επιστρέφει το συνολικό κόστος για την υλοποίηση όλων των εργασιών στο σύνολο $A_{r_1} \cup A_{r_2} \cup \dots \cup A_{r_N}$ (Team Objective Function). Ο στόχος είναι όλες οι εργασίες που βρίσκονται στο T να ανατεθούν σε κάποιο πράκτορα με τέτοιο τρόπο ώστε η συνολική αυτή ανάθεση να ελαχιστοποιεί το ομαδικό κόστος. Αν $A = \{A_{r_1}, A_{r_2}, \dots, A_{r_N}\}$ είναι μία πλήρης ανάθεση που ικανοποιεί την προϋπόθεση ότι όλες οι εργασίες έχουν ανατεθεί και επομένως $A_{r_1} \cup A_{r_2} \cup \dots \cup A_{r_N} = T$, τότε το team objective εκφράζεται ως

$$\min_A f(g_{r_1}(r_1, A_{r_1}), \dots, g_{r_N}(r_N, A_{r_N})) \quad (2.1)$$

για όλες τις πιθανές πλήρεις αναθέσεις A .

2.3 Team Objectives for Multi-Agent Task Assignment

Στην εργασία αυτή εξετάζουμε τρία team objectives :

- MINISUM : $\min_A \sum_j g_{r_j}(r_j, A_{r_j})$
- MINIMAX : $\min_A \max_j g_{r_j}(r_j, A_{r_j})$
- MINIAVE : $\min_A \frac{1}{M} \sum_j STC(r_j, A_{r_j})$

όπου $STC(r_i, A_{r_i})$ είναι το ελάχιστο άθροισμα των επιμέρους κοστών ανά εργασία για όλες τις εργασίες στο A_{r_i}

Η ανάθεση που συμβαίνει σύμφωνα με το MINISUM team objective ελαχιστοποιεί το άθροισμα όλων των ατομικών κοστών των πρακτόρων, ενώ με το MINIMAX team objective ελαχιστοποιεί τη μέγιστη agent objective function g και τέλος με το MINIAVE ελαχιστοποιεί το μέσο κόστος ανά εργασία.

Η εύρεση βέλτιστων αναθέσεων για τα 3 team objectives είναι NP-hard πρόβλημα [1]. Εξαιτίας της μεγάλης υπολογιστικής πολυπλοκότητας των αλγορίθμων εύρεσης βέλτιστων αναθέσεων, στην πράξη χρησιμοποιούνται αποδοτικοί προσεγγιστικοί αλγόριθμοι για μεγάλης κλίμακας προβλήματα. Οι προσεγγιστικοί αλγόριθμοι καταλήγουν σε λύσεις κοντά στις βέλτιστες (near-optimal). Για μικρής κλίμακας προβλήματα μπορούν να δοθούν βέλτιστες λύσεις με την χρήση μικτού ακέραιου μαθηματικού προγραμματισμού.

2.4 Auction Framework

Το auction-based multi-agent coordination system θεωρεί τους πράκτορες σαν bidders και τις εργασίες σαν δημοπρατούμενα αντικείμενα. Στην αρχή καμία εργασία δεν έχει ανατεθεί σε κάποιον πράκτορα. Σε κάθε γύρο προσφορών, όλοι οι πράκτορες κάνουν bid για όλες τις μη ανατετιμένες εργασίες. Ο πράκτορας που δίνει τη χαμηλότερη προσφορά για κάποια εργασία σε κάποιο γύρο, κερδίζει στο γύρο αυτό και αναλαμβάνει την συγκεκριμένη εργασία. Στη συνέχεια ξεκινάει ένας νέος γύρος δημοπρασίας με όλους τους πράκτορες να κάνουν bid σε όλες τις εναπομείναντες μη ανατετιμένες εργασίες. Η ίδια διαδικασία ακολουθείται μέχρι να ανατεθούν όλες οι εργασίες. Αξίζει να σημειωθεί ότι κάποιος πράκτορας δεν χρειάζεται να κάνει προσφορές για περισσότερες από μία εργασίες σε κάθε γύρο, αφού όλα τα υπόλοιπα bids αυτού του πράκτορα δεν έχουν πιθανότητα να κερδίσουν.

Για την ομαλή διεξαγωγή της δημοπρασίας χρειάζεται ένας auctioneer που θα συγχρονίζει τους bidders, θα συλλέγει όλα τα bids και θα προσδιορίζει

Πίνακας 2.1: Auction Algorithm’s pseudocode

```
 $W \leftarrow T$ 
while( $W$  is not empty)
(
- ask each bidder to bid for one task from set  $W$ 
- gather the bids from all bidders
- determine the winner agent  $r$  and winner task  $t$ 
- assign  $t$  to  $r$  ( $W \leftarrow W - \{t\}$ )
)
```

τον νικητή. Σε κάθε γύρο ο auctioneer ανακοινώνει την αρχή του και κάθε πράκτορας υπολογίζει το bid του τοπικά και παράλληλα με τους υπόλοιπους πράκτορες. Εν συνεχεία, οι πράκτορες στέλνουν τις προσφορές τους στον auctioneer, ο οποίος αναλαμβάνει να καθορίσει το νικητήριο bid καθώς και τον νικητή πράκτορα. Στο τέλος του γύρου, ο auctioneer ανακοινώνει τα αποτελέσματα σε όλους τους πράκτορες και αναθέτει στο νικητή bidder την αντίστοιχη εργασία.

2.5 Ο αλγόριθμος

Ο αλγόριθμος της δημοπρασίας σε μορφή ψευδοκώδικα βρίσκεται στον πίνακα 2.1.

Το σύνολο W περιέχει τις εργασίες που δεν έχουν ανατεθεί σε κάποιον πράκτορα. Όταν λοιπόν το σετ W αδειάσει, τότε όλες οι εργασίες από το T έχουν ανατεθεί. Κάθε επανάληψη του βρόγχου στον πίνακα 2.1 αποτελεί έναν γύρο δημοπρασίας. Κάθε γύρος δημοπρασίας χωρίζεται σε δύο φάσεις, την bidding φάση και την φάση ανάθεσης. Αρχικά, οι πράκτορες κάνουν bid στην εργασία που θεωρούν ως πιο πιθανή να αναλάβουν. Για κάποιον πράκτορα, η πιο πιθανή εργασία για να αναλάβει είναι αυτή με το μικρότερο bid σε σύγκριση με τις υπόλοιπες εργασίες στο σύνολο W . Όταν όλοι οι πράκτορες τελειώσουν με τον

υπολογισμό και την επιλογή των bids , τότε αρχίζει η φάση της ανάθεσης. Σε αυτή τη φάση καθορίζεται ο νικητής bidder και αναλαμβάνει την αντίστοιχη εργασία.

2.6 Παραγωγή Bidding κανόνων

Σε κάθε γύρο της δημοπρασίας οι πράκτορες χρησιμοποιούν ένα bidding κανόνα για να καθορίσουν το κατάλληλο bid για κάθε εργασία ανάλογα με το τρέχον team objective.

Έστω ότι $(A_{r_1}, A_{r_2}, \dots, A_{r_N})$ είναι η τρέχουσα ανάθεση των εργασιών στους πράκτορες σε κάποιο γύρο της δημοπρασίας και t μία εργασία που δεν έχει ανατεθεί σε κάποιον πράκτορα. Οι πράκτορες χρησιμοποιούν ένα γενικό κανόνα για να κάνουν bid για οποιοδήποτε team objective.

Bidding Rule : Για ένα συγκεκριμένο team objective, ο πράκτορας r κάνει bid στην μη ανατετιμένη εργασία t την διαφορά στην ομαδική απόδοση μεταξύ της τρέχουσας ανάθεσης και της ανάθεσης που προκύπτει αν στην τρέχουσα ανάθεση προστεθεί η εργασία t .

Επομένως, ο πράκτορας r_i θα κάνει bid για την εργασία t τη διαφορά

$$f(g_{r_1}(r_1, A'_{r_1}), \dots, g_{r_N}(r_N, A'_{r_N})) - f(g_{r_1}(r_1, A_{r_1}), \dots, g_{r_N}(r_N, A_{r_N})) \quad (2.2)$$

όπου $A'_{r_i} = A_{r_i} \cup \{t\}$ και $A'_{r_j} = A_{r_j}$ για $i \neq j$

Σύμφωνα με την εξίσωση 2.2, κάθε πράκτορας πρέπει να γνωρίζει όλες τις αναθέσεις εργασιών στους υπόλοιπους πράκτορες ώστε να υπολογίσει το bid του. Παρακάτω δείχνουμε ότι αυτό δεν είναι απαραίτητο για τα τρία team objectives που εξετάζουμε και απλά κάθε πράκτορας χρειάζεται να γνωρίζει μόνο τις δικές του ανατετιμένες εργασίες.

- Για το MINISUM team objective, ο πράκτορας r_i θα έπρεπε να κάνει bid για το task t

$$\begin{aligned}
& \sum_j g_{r_j}(r_j, A'_{r_j}) - \sum_j g_{r_j}(r_j, A_{r_j}) \\
&= (g_{r_1}(r_1, A'_{r_1}) + \dots + g_{r_N}(r_N, A'_{r_N})) - (g_{r_1}(r_1, A_{r_1}) + \dots + g_{r_N}(r_N, A_{r_N})) \\
&= g_{r_i}(r_i, A'_{r_i}) - g_{r_i}(r_i, A_{r_i}) \\
&\text{αφού } A'_{r_i} = A_{r_i} \cup \{t\} \text{ και } A'_{r_j} = A_{r_j} \text{ για } i \neq j.
\end{aligned}$$

Επομένως, ο πράκτορας r_i τελικά κάνει bid για το task t

$$b_{r_i t} = g_{r_i}(r_i, A_{r_i} \cup \{t\}) - g_{r_i}(r_i, A_{r_i}) \quad (2.3)$$

η οποία σχέση δεν εξαρτάται από τα σύνολα A_r άλλων πρακτόρων.

- Για το MINIMAX team objective, ο πράκτορας r_i θα έπρεπε να κάνει bid για το task t

$$\max_j g_{r_j}(r_j, A'_{r_j}) - \max_j g_{r_j}(r_j, A_{r_j}) = g_{r_i}(r_i, A_{r_i} \cup \{t\}) - \max_j g_{r_j}(r_j, A_{r_j})$$

Για την εξαγωγή αυτής της σχέσης λαμβάνουμε υπόψιν μας ότι $\max_j g_{r_j}(r_j, A'_{r_j}) = g_{r_j}(r_j, A'_{r_j})$, ειδικά η εργασία t θα είχε ήδη ανατεθεί σε προηγούμενο γύρο. Ο όρος $\max_j g_{r_j}(r_j, A_{r_j})$ μπορεί να παραλειφθεί καθώς τα αποτελέσματα της δημοπρασίας παραμένουν τα ίδια εάν όλα τα bids αλλάζουν κατά μία προσθετική σταθερά. Συνεπώς, ο πράκτορας r_i μπορεί να κάνει bid την τιμή της agent objective function $g_{r_i}(r_i, A_{r_i} \cup \{t\})$ για την εργασία t

$$b_{r_i t} = g_{r_i}(r_i, A_{r_i} \cup \{t\}) \quad (2.4)$$

- Για το MINIAVE team objective, ο πράκτορας r_i θα έπρεπε να κάνει bid

για το task t

$$\begin{aligned}
& \frac{1}{M} \sum_i STC(r_i, A'_{r_i}) - \frac{1}{M} \sum_i STC(r_i, A_{r_i}) \\
&= \frac{1}{M} ((STC(r_1, A'_{r_1}) + \dots + STC(r_N, A'_{r_N})) - (STC(r_1, A_{r_1}) + \dots + STC(r_N, A_{r_N}))) \\
&= \frac{1}{M} (STC(r_i, A'_{r_i}) - STC(r_i, A_{r_i})) \\
&\text{αφού } A'_{r_i} = A_{r_i} \cup \{t\} \text{ και } A'_{r_j} = A_{r_j} \text{ για } i \neq j
\end{aligned}$$

Ο όρος $\frac{1}{M}$ μπορεί να παραλειφθεί καθώς τα αποτελέσματα της δημοπρασίας παραμένουν αναλλοίωτα εάν όλα τα bids πολλαπλασιαστούν με ένα σταθερό παράγοντα. Το bid του πράκτορα r_i για την εργασία t είναι

$$b_{r_i t} = STC(r_i, A_{r_i} \cup \{t\}) - STC(r_i, A_{r_i}) \quad (2.5)$$

2.7 Οι φάσεις του γύρου δημοπρασίας

Bidding phase: Για την ολοκλήρωση μίας δημοπρασίας M εργασιών χρειάζεται να πραγματοποιηθούν M γύροι δημοπρασίας. Σε κάθε γύρο, κάθε πράκτορας κάνει bid για την εργασία που θεωρεί ως πιο πιθανή να αναλάβει. Οι πράκτορες υπολογίζουν πρώτα το bid για κάθε εργασία t που βρίσκεται στο W σύμφωνα με τον γενικό bidding κανόνα

$$b_{r_i t} = \begin{cases} f(g_{r_1}(r_1, A'_{r_1}), \dots, g_{r_N}(r_N, A'_{r_N})) - f(g_{r_1}(r_1, A_{r_1}), \dots, g_{r_N}(r_N, A_{r_N})) & \text{if } t \in \mathbf{S}_{r_i} \\ \infty & \text{if } t \notin \mathbf{S}_{r_i} \end{cases} \quad (2.6)$$

Εάν ο πράκτορας r_i δεν μπορεί να αναλάβει την εργασία t επειδή αυτή δεν περιέχεται στο σύνολο S_{r_i} , τότε θέτει την τιμή του bid στο άπειρο για την εργασία αυτή. Με αυτόν τον τρόπο εξασφαλίζει ότι δεν θα αναλάβει την συγκεκριμένη εργασία.

Αφού οι πράκτορες υπολογίσουν τα bids για όλες τις εργασίες στο W , επιλέγουν την εργασία με το ελάχιστο bid και ενημερώνουν τον auctioneer για την εργασία που ενδιαφέρονται να αναλάβουν και την προσφορά που κάνουν.

$$b_{r_i} = \min_{j \in W} (b_{r_j})$$

$$t_{r_i} = \arg \min_{j \in W} (b_{r_j})$$

Οι bidders στέλνουν στον auctioneer το ζεύγος τιμών $\{b_{r_i}, t_{r_i}\}$.

Φάση ανάθεσης: Ο auctioneer λαμβάνει ένα σύνολο με N ζεύγη τιμών $\{b_{r_i}, t_{r_i}\}$ από όλους τους bidders. Αρχικά, ο auctioneer προσδιορίζει το bid με την ελάχιστη τιμή.

$$p = \min_{r_i \in R} \{b_1, b_2, \dots, b_N\}$$

όπου p η τιμή του νικητήριου bid.

Μετά τον προσδιορισμό του χαμηλότερου bid, ο auctioneer καθορίζει την εργασία στην οποία αντιστοιχεί το bid, καθώς και τον πράκτορα r_i στον οποίο ανήκει το συγκεκριμένο bid. Στη συνέχεια ο auctioneer αναθέτει στον πράκτορα r_i την εργασία t_{r_i} . Η ανάθεση της t_{r_i} στον r_i περιλαμβάνει την προσθήκη της t_{r_i} στο σετ A_{r_i} και την αφαίρεσή της από το W .

Κεφάλαιο 3

Ο Πυρήνας

Βασισμένοι στον αλγόριθμο του κεφαλαίου 2 υλοποιήσαμε ένα κατανεμημένο software framework που μπορεί να χρησιμοποιηθεί για την επίλυση ποικίλων προβλημάτων ανάθεσης εργασιών σε πολυπρακτορικά συστήματα μέσω δημοπρασιών. Σε αυτό το κεφάλαιο αναλύουμε τον τρόπο υλοποίησης του πυρήνα, ο οποίος παραμένει αναλλοίωτος για διάφορες εφαρμογές.

Κάθε πράκτορας θεωρείται σαν ένας ξεχωριστός κόμβος σε ένα ενσύρματο ή ασύρματο δίκτυο υπολογιστών με την δική του IP διεύθυνση που μπορεί να συμμετάσχει σε μία δημοπρασία είτε ως bidder είτε ως auctioneer ή ως και τα δύο. Για να συμβαίνει αυτό πρέπει οι πράκτορες να πραγματοποιούν δύο διαφορετικές λειτουργίες παράλληλα. Το πρόγραμμα σύμφωνα με το οποίο λειτουργούν οι πράκτορες χωρίζεται σε δύο ρουτίνες, την ρουτίνα του auctioneer και την ρουτίνα του bidder. Πρώτα περιγράφονται τα βασικά βήματα πραγματοποίησης μιας δημοπρασίας και οι κανόνες επικοινωνίας των πρακτόρων. Στη συνέχεια εξετάζουμε τις δύο ρουτίνες του προγράμματος των πρακτόρων.

3.1 Καθορισμός των bidders

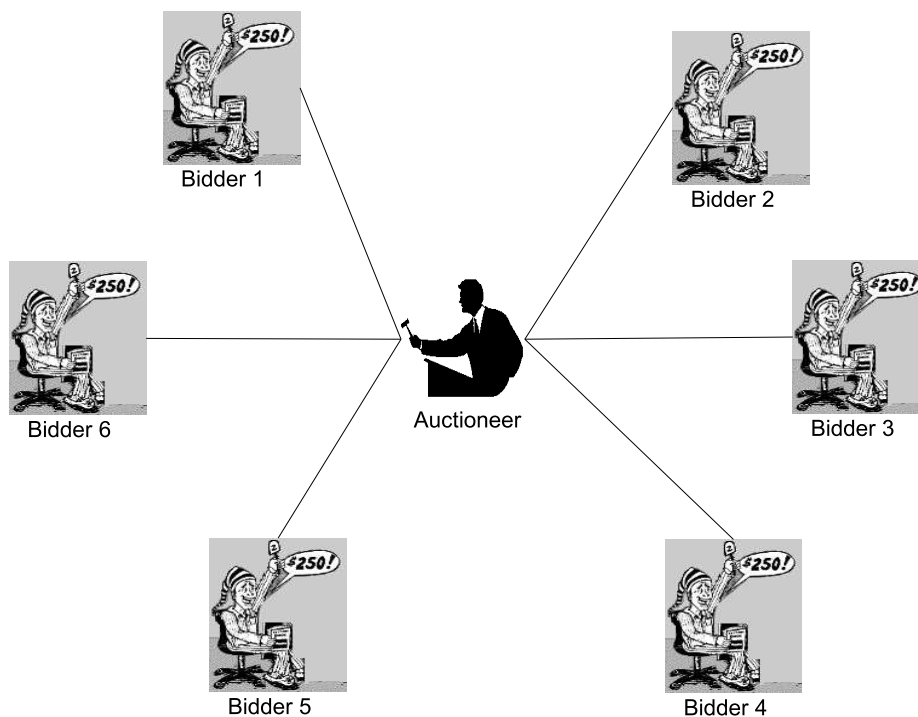
Όλοι οι πράκτορες διαθέτουν ένα κατάλογο, ο οποίος περιέχει τις IP διευθύνσεις για τους πράκτορες που είναι δηλωμένοι στην ομάδα διαμοιρασμού των εργασιών. Υπό φυσιολογικές συνθήκες όλοι οι δηλωμένοι πράκτορες θα έπαιρναν μέρος στις δημοπρασίες που πραγματοποιούνται από μέλος της ομάδας τους. Ωστόσο, υπάρχει πάντα η πιθανότητα κάποιος πράκτορας να μην μπορεί να συμμετάσχει σε μία δημοπρασία για διάφορους λόγους, όπως να βρίσκεται σε μακρινή απόσταση ή να έχει τεθεί εκτός λειτουργίας. Τέτοιους πράκτορες τους θεωρούμε μη προσιτούς (non-reachable).

Όποτε ένας πράκτορας έχει ένα ή περισσότερα tasks για δημοπρασία, τότε θα πρέπει να συμπεριλάβει όλους τους διαθέσιμους πράκτορες στην δημοπρασία και να εξαιρέσει τους non-reachable από αυτήν. Ο παραπάνω πράκτορας, ο οποίος είναι ο auctioneer της συγκεκριμένης δημοπρασίας, καθορίζει ποιοι θα είναι οι bidders κάνοντας “ping” σε όλες τις IP διευθύνσεις που είναι καταχωρημένες στον αντίστοιχο κατάλογο. Ο auctioneer κρατάει/διατηρεί τις IP διευθύνσεις των πρακτόρων που ανταποκρίθηκαν στο “ping” που επιχείρησε και προχωράει στην πραγματοποίηση της δημοπρασίας λαμβάνοντας μόνο αυτούς υπόψιν του ως bidders.

3.2 Network set-up

Έχοντας καθορίσει τους bidders της δημοπρασίας, ο auctioneer εγκαθιστά ένα δίκτυο για να επικοινωνεί μαζί τους. Επειδή ο auctioneer διευθύνει την διαδικασία της δημοπρασίας, πρέπει να είναι συνδεδεμένος με όλους τους bidders. Αντίθετα, οι bidders δεν χρειάζεται να επικοινωνούν μεταξύ τους. Επομένως υπάρχουν συνδέσεις μόνο μεταξύ του auctioneer και των bidders και όχι μεταξύ των bidders. Το δίκτυο που δημιουργείται έχει τοπολογία ‘Αστέρα’ με

τον auctioneer σαν κεντρικό κόμβο. Στο σχήμα 3.1 παρουσιάζεται ένα τέτοιο δίκτυο.



Σχήμα 3.1: Ο auctioneer και οι bidders είναι συνδεδεμένοι σε ένα δίκτυο Αστέρα με τον auctioneer να είναι ο κεντρικός κόμβος

3.3 Πρωτόκολλο Δημοπρασίας

Όλοι οι auctioneers ακολουθούν κάποιους κανόνες για να εξασφαλίσουν την σωστή διεξαγωγή κάθε δημοπρασίας. Ως σωστή διεξαγωγή ορίζουμε ότι η δημοπρασία θα ολοκληρωθεί επιτυγχάνοντας κάποια βελτιστοποίηση των αποτελεσμάτων. Οι παραπάνω κανόνες αποτελούν το Πρωτόκολλο Δημοπρασίας. Το πρωτόκολλο αυτό χωρίζεται σε δύο ενότητες, την διαδικασία σύνδεσης του auctioneer στους bidders και την ακολουθιακή πραγματοποίηση των δημοπρασιών.

3.3.1 Διαδικασία σύνδεσης

Γενικά σε μία σύνδεση μεταξύ δύο κόμβων, ο ένας λειτουργεί ως server και ο άλλος ως client. Στην δική μας υλοποίηση ο bidder λειτουργεί ως server και ο auctioneer ως client, ο οποίος συνδέεται στον bidder για να εξυπηρετηθεί.

Εκτός του καταλόγου με τις IP διευθύνσεις, κάθε πράκτορας διατηρεί ακόμα ένα κατάλογο που έχει καταχωρημένες τις IP διευθύνσεις με τα αντίστοιχα port numbers. Κάθε bidder περιμένει τον auctioneer σε συγκεκριμένη προκαθορισμένη θύρα. Ο αριθμός της θύρας, στην οποία ‘ακούει’ ο bidder, λειτουργεί ως ID για αυτόν. Ο auctioneer μπορεί να αναγνωρίσει κάθε bidder από τον αριθμό της θύρας του bidder στην οποία είναι συνδεδεμένος.

Ο auctioneer για να συνδεθεί με κάποιον πράκτορα, συνδέεται αρχικά στη θύρα που του υποδεικνύει το αντίστοιχο αρχείο, χωρίς όμως να ξεκινά την πραγματοποίηση της δημοπρασίας. Αφού συνδεθεί στον πράκτορα, ο auctioneer κάνει αίτηση για ένα νέο αριθμό θύρας για σύνδεση. Όταν αποκτήσει το νέο αριθμό, ο auctioneer συνδέεται στον ίδιο πράκτορα μέσω της νέας θύρας όπου και αρχίζει την δημοπρασία. Ο πράκτορας στον οποίο συνδέθηκε ο auctioneer εξακολουθεί να περιμένει για επιπλέον auctioneers στην αντίστοιχη προκαθορισμένη θύρα. Με αυτό τον τρόπο αποφεύγονται οι αποτυχημένες προσπάθειες για πραγματοποίηση δημοπρασίας από κάποιον auctioneer.

Όσο περισσότεροι είναι οι πράκτορες, τόσο πιο πιθανό είναι να συμπέσουν χρονικά οι δημοπρασίες που θέλουν να υλοποιήσουν. Στην παράγραφο 3.3.2 εξηγείται γιατί δεν επιτρέπεται να πραγματοποιούνται παράλληλα οι δημοπρασίες. Εφόσον λοιπόν μόνο μία δημοπρασία μπορεί να διενεργείται σε κάθε χρονική στιγμή, κάθε auctioneer που θα επιχειρούσε να αρχίσει μία δημοπρασία ενώ κάποιος άλλος auctioneer πραγματοποιεί την δική του, θα απορριπτόταν από τους πράκτορες bidders. Το γεγονός αυτό αποτελεί αποτυχημένη προσπάθεια για πραγματοποίηση δημοπρασίας.

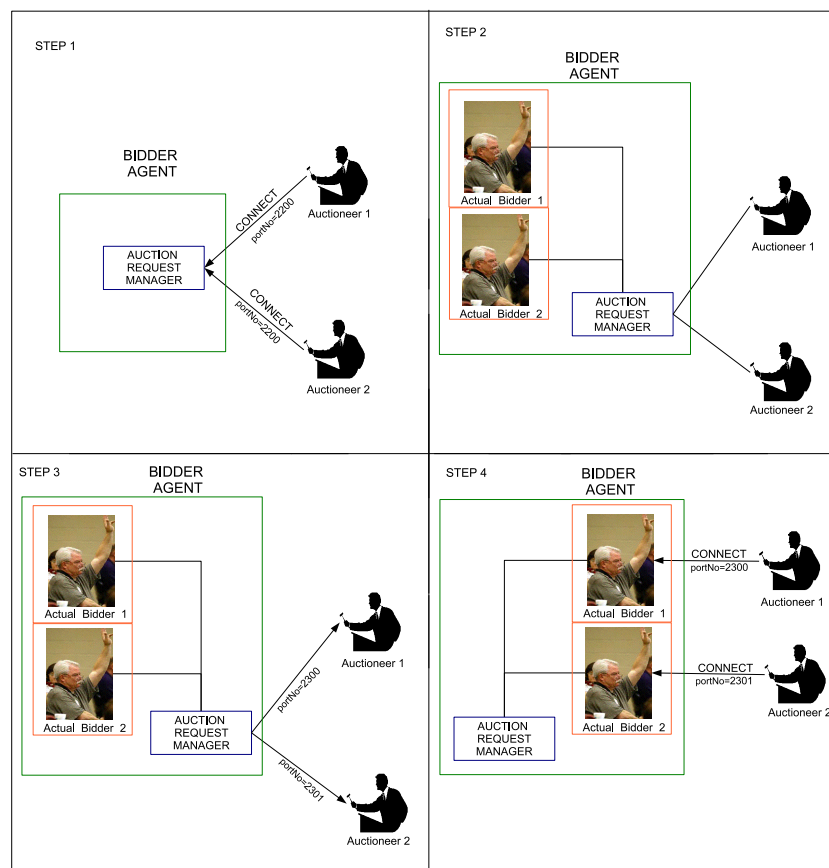
Στην περίπτωση μας οι πράκτορες bidders δεν απορρίπτουν τους auctioneers που επιχειρούν να αρχίσουν μία δημοπρασία ενώ πραγματοποιείται κάποια άλλη, αλλά τους κρατούν στην αναμονή. Όταν ολοκληρωθεί η τρέχουσα δημοπρασία επιλέγεται ο επόμενος auctioneer για να ξεκινήσει την δική του δημοπρασία. Έτσι κάθε πράκτορας συμμετέχει σαν bidder σε μία δημοπρασία ενώ ταυτόχρονα διατηρεί και μία λίστα αναμονής με τους auctioneers που έχουν κάνει αίτηση για δημοπρασία.

Για την καλύτερη κατανόηση της εξυπηρέτησης των auctioneers από έναν bidder φανταστείτε ένα γιατρό και N ασθενείς σε δύο διαφορετικές περιπτώσεις. Στην πρώτη περίπτωση υπάρχει μόνο ένα δωμάτιο εξέτασης στο ιατρείο χωρίς χώρο αναμονής. Ο πρώτος ασθενής που θα επισκεφθεί το ιατρείο θα εξεταστεί από τον γιατρό. Κατά την διάρκεια εξέτασης αυτού του ασθενή υπάρχει υψηλή πιθανότητα κάποιος άλλος ασθενής να επισκεφθεί το ιατρείο. Αυτός ο ασθενής θα πρέπει να αποχωρήσει από το ιατρείο και να επιστρέψει αργότερα. Φυσικά, όταν αυτός ο ασθενής επιστρέψει στο ιατρείο είναι πιθανόν να βρει ένα νέο ασθενή να εξετάζεται. Στην δεύτερη περίπτωση το ιατρείο διαθέτει N δωμάτια εξέτασης και μία γραμματέα. Η γραμματέας διατηρεί πληροφορίες για όλα τα δωμάτια και όλους τους ασθενείς που βρίσκονται στο ιατρείο. Όταν ένας νέος ασθενής επισκέπτεται το ιατρείο, η γραμματέας αναλαμβάνει να του βρει κάποιο άδειο δωμάτιο εξέτασης ώστε να περιμένει να εξεταστεί από τον γιατρό. Επιπλέον, η γραμματέας είναι υπεύθυνη για να ενημερώσει τον γιατρό σχετικά με τον επόμενο ασθενή για εξέταση και το δωμάτιο στο οποίο περιμένει.

Συγκρίνοντας τις δύο περιπτώσεις, παρατηρούμε ότι ο δεύτερος τρόπος λειτουργίας με τα N δωμάτια καταναλώνει περισσότερους πόρους αλλά είναι αποδοτικότερος από πλευράς χρόνου, καθώς δεν υπάρχουν διαστήματα ανεκμετάλλευτου χρόνου όσο υπάρχουν ασθενείς για εξέταση. Ο πρώτος τρόπος λειτουργίας είναι παραπλήσιος με το πρωτόκολλο ALOHA εάν παρομοιάσουμε

τους auctioneers με κόμβους δικτύου, τον bidder με τον κεντρικό server-node και τα auction requests με πακέτα προς μετάδοση. Η μόνη διαφορά είναι ότι εάν συγκρουστούν δύο πακέτα στο ALOHA πρέπει να επαναμεταδοθούν και τα δύο μετά από τυχαίο χρόνο, ενώ στην παραπάνω περίπτωση ο ένας εκ των δύο auctioneers θα εξυπηρετηθεί. Η διαδικασία σύνδεσης των auctioneers στην υλοποίησή μας προσομοιάζει την δεύτερη περίπτωση αφού δεν τίθεται θέμα μη επάρκειας των hardware πόρων.

Η εξυπηρέτηση του auctioneer αποτελείται από δύο επίπεδα, ο τρόπος υλοποίησης των οποίων περιγράφεται αναλυτικά στην παράγραφο 3.6. Το πρώτο επίπεδο είναι το “Auction Request Manager” και το δεύτερο είναι το “Actual Bidder”. Στο σχήμα 3.2 απεικονίζεται ένα παράδειγμα για την καλύτερη κατανόηση της παραπάνω διαδικασίας. Οι auctioneers 1 και 2 συνδέονται σχεδόν ταυτόχρονα σε κάποιον πράκτορα, ζητώντας του να συμμετάσχει στις δημοπρασίες τους. Ο auctioneer 1 έχει συνδεθεί λίγο πριν από τον 2 και προηγείται. Ο auction request manager, ο οποίος ενεργεί σαν την γραμματέα παραπάνω, δέχεται τα auction requests από τους δύο auctioneers και δημιουργεί δύο actual bidders, ένα για κάθε auctioneer. Στη συνέχεια, ο request manager στέλνει στους auctioneers τους αριθμούς των θυρών, στις οποίες περιμένουν οι actual bidders. Τέλος, οι auctioneers συνδέονται στους αντίστοιχους actual bidders όπου μπορούν να πραγματοποιήσουν τις δημοπρασίες τους. Εφόσον ο auctioneer 1 συνδέθηκε πρώτος θα ξεκινήσει την δημοπρασία του πριν τον auctioneer 2, ο οποίος μπαίνει στην αναμονή και περιμένει τον 1 να τελειώσει. Στην πραγματικότητα ο actual bidder 2 περιμένει τον actual bidder 1 να τελειώσει.



Σχήμα 3.2: Τρόπος λειτουργίας του Auction Request Manager

3.3.2 Πραγματοποίηση Δημοπρασιών

Όπως είδαμε και στο κεφάλαιο 2, ο στόχος της ανάθεσης εργασιών μέσω δημοπρασιών σε πολυπρακτορικά συστήματα είναι η βελτιστοποίηση των αποτελεσμάτων και η ελαχιστοποίηση της team objective συνάρτησης. Η πραγματοποίηση δημοπρασιών που εκτελούνται παράλληλα δεν μπορεί να εγγυηθεί βελτιστοποίηση των αποτελεσμάτων καθώς δεν προλαβαίνουν να ενημερωθούν οι τιμές των δεδομένων. Αντιθέτως, η σειριακή πραγματοποίηση εγγυάται την ποιότητα των αποτελεσμάτων που αναμένονται από τον αλγόριθμο του κεφαλαίου 2. Για το λόγο αυτό στην υλοποίησή μας οι δημοπρασίες εκτελούνται σειριακά.

Στο σχήμα 3.3 παρουσιάζεται ένα σενάριο παράλληλης πραγματοποίησης δημοπρασιών με τους κινδύνους επικάλυψης δεδομένων που εμπεριέχει, οδηγώντας έτσι σε μη επιθυμητά αποτελέσματα. Τα bids b_{15}, b_{25}, b_{34} δεν υπολογίζονται σωστά αφού κατά τον υπολογισμό τους αγνοούνται τα αποτελέσματα προηγούμενων auction rounds. Θεωρώντας ότι το κόστος για τον πράκτορα r_1 να φέρει σε πέρας τα tasks t_1, t_2 είναι $g(A_1=\{t_1, t_2\}) = 13$, το bid b_{15} σύμφωνα με το MINISUM team objective θα έπρεπε να είναι 9 και όχι 5. Σύμφωνα με τον αλγόριθμο δημοπρασίας το task t_1 θα έπρεπε να ανατεθεί στον πράκτορα r_1 και το task t_2 στον r_2 ενώ στο χρονοδιάγραμμα του σχήματος 3.3 και τα δύο tasks ανατίθενται στον r_1 .

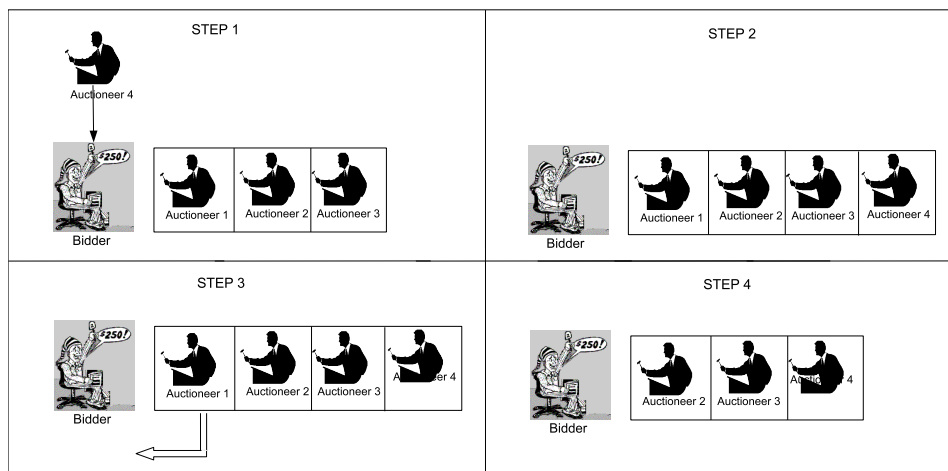
Auction 1	Auction 2
Tasks (t1,t2,t3)	Tasks (t4,t5)
$b_{11} = 4$ $b_{21} = 7$ $b_{33} = 9$	$b_{15} = 5$ $b_{25} = 7$ $b_{34} = 10$
$\min(b_{rt})=b_{11}=4$	$\min(b_{rt})=b_{15}=5$
Assign t1 to r1	Assign t5 to r1
$b_{13} = 20$ $b_{22} = 10$ $b_{33} = 9$	

Σχήμα 3.3: Data overlap

Μετά την επιλογή της σειριακής πραγματοποίησης των δημοπρασιών, χρειάζεται να καθοριστεί η σειρά με την οποία θα λαμβάνουν χώρα οι δημοπρασίες. Εφόσον δεν υπάρχει κάποιος special πράκτορας ή κάποιο special auction και όλα τα auctions θεωρούνται ισοδύναμα, ο πιο δίκαιος τρόπος πραγματοποίη-

σης των δημοπρασιών είναι σύμφωνα με τη σειρά άφιξης τους στον bidder. Ο bidder χρησιμοποιεί σαν auctions' IDs τα port numbers, στα οποία είναι συνδεδεμένοι οι auctioneers. Τα port numbers εισάγονται σε μία FIFO λίστα, την οποία ελέγχει ο auction request manager για να καθορίσει την επόμενη δημοπρασία για πραγματοποίηση.

Στο σχήμα 3.4 βλέπουμε την ουρά εξυπηρέτησης των auctioneers στους bidders και την διαμόρφωσή της ανάλογα με τα events στην διάρκεια του χρόνου. Στην πρώτη γραμμή ο auctioneer 4 κάνει ένα auction request προς τον bidder και στην συνέχεια μπαίνει τελευταίος στην ουρά εξυπηρέτησης. Στην δεύτερη γραμμή ο auctioneer 1 ολοκληρώνει την πραγματοποίηση της δημοπρασίας του και φεύγει από την ουρά αφήνοντας την θέση του ελεύθερη. Ο auctioneer 2 που είναι επόμενος στην ουρά, παίρνει την θέση του auctioneer 1 και αρχίζει να εξυπηρετείται, ενώ όλοι οι auctioneers μετακινούνται κατά μία θέση πιο κοντά στον bidder.



Σχήμα 3.4: Η ουρά εξυπηρέτησης του bidder

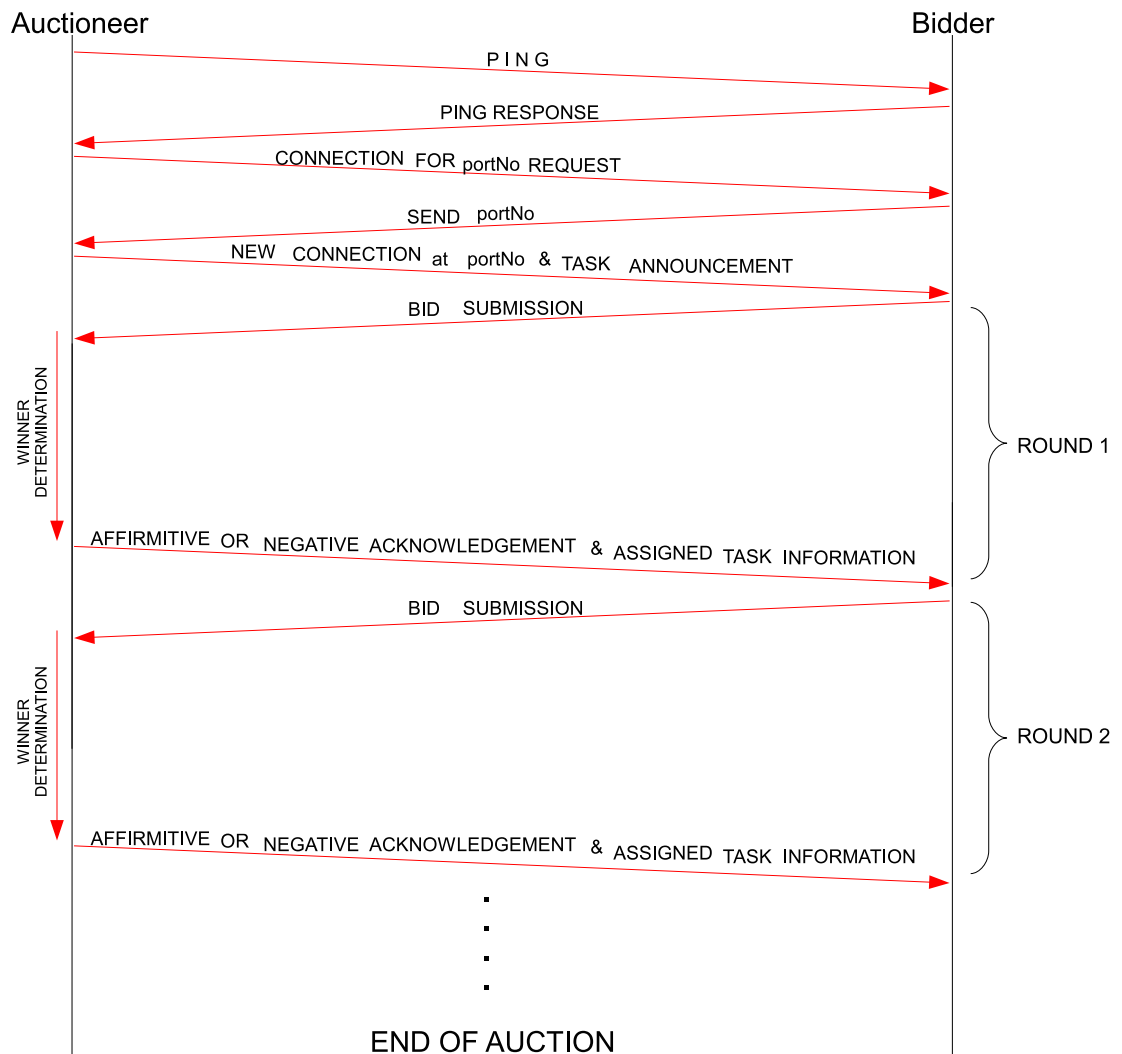
3.4 Η Φάση της Ανάθεσης

Ο auctioneer είναι υπεύθυνος για τον καθορισμό του νικητή bidder σε κάθε auction round. Αφού εγκαταστήσει την επικοινωνία μεταξύ των πρακτόρων, ο auctioneer προχωρά στην πραγματοποίηση της δημοπρασίας ακολουθώντας τους κανόνες του Auction Protocol. Σε κάθε γύρο ο auctioneer συλλέγει τα bids από όλους τους bidders και στη συνέχεια καθορίζει το μικρότερο καθώς και τον αντίστοιχο πράκτορα και task. Στο τέλος του γύρου ενημερώνει όλους τους bidders σχετικά με τα αποτελέσματα στέλνοντας μία θετική επιβεβαίωση στον νικητή bidder, η οποία τον ειδοποιεί ότι κέρδισε τον γύρο και πρέπει να ενημερώσει δεδομένα των set του και μία αρνητική επιβεβαίωση στους υπόλοιπους bidders. Χρησιμοποιώντας αρνητικές επιβεβαιώσεις (negative acknowledgements) αντί για timeout signals, επιταχύνουμε την όλη διαδικασία αφού εξαλείφουμε τις περιόδους χαμένου χρόνου ανάμεσα στους γύρους. Έτσι οι bidders μπορούν να ξεκινούν άμεσα την συμμετοχή τους σε ένα νέο γύρο δημοπρασίας. Στο σχήμα 3.5 παρουσιάζεται το χρονοδιάγραμμα της συνολικής διαδικασίας για την πραγματοποίηση μία δημοπρασίας.

3.5 Η Ρουτίνα του Auctioneer

Το πρόγραμμα κάθε πράκτορα είναι χωρισμένο σε δύο ρουτίνες που εκτελούνται παράλληλα, την ρουτίνα του auctioneer και την ρουτίνα του bidder. Κάθε πράκτορας μπορεί να δημοπρατήσει περισσότερα από ένα tasks ανά δημοπρασία αλλά δεν μπορεί να πραγματοποιήσει περισσότερες από μία δημοπρασίες ταυτόχρονα. Όταν ένας πράκτορας έχει κάποια tasks προς δημοπρασία ενεργοποιείται η ρουτίνα του auctioneer και λαμβάνει χώρα μία σειρά από ενέργειες.

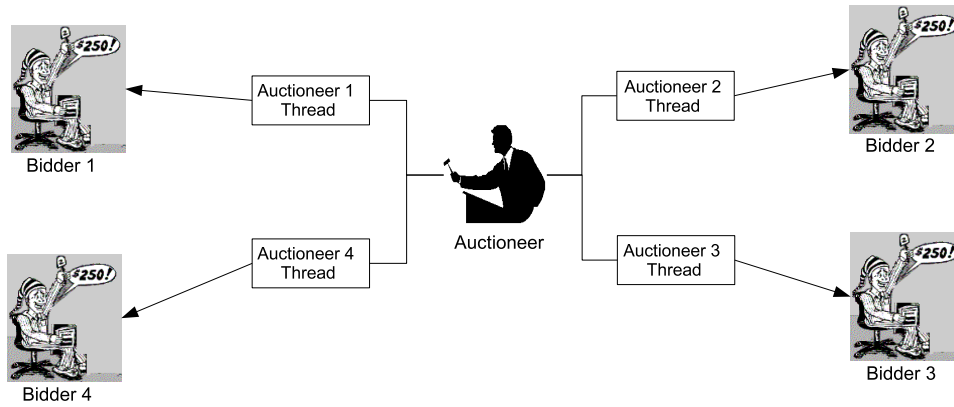
Αρχικά ο auctioneer ελέγχει για το ποιοι πράκτορες είναι διαθέσιμοι να συμμετάσχουν στην δημοπρασία του σύμφωνα με την φάση καθορισμού των



Σχήμα 3.5: Το χρονοδιάγραμμα της συνολικής διαδικασίας

bidders. Μετά τον καθορισμό των bidders δημιουργείται ένα thread για κάθε πράκτορα που θεωρείται ότι είναι bidder στην συγκεκριμένη δημοπρασία. Αν για παράδειγμα ανταποκρίθηκαν K πράκτορες στο “ping” που επιχείρησε ο auctioneer, τότε θα δημιουργηθούν K threads. Κάθε thread θα συνδεθεί στον αντίστοιχο bidder για τον οποίο δημιουργήθηκε και σύμφωνα με τους κανόνες του Auction Protocol θα ζητήσει από τον bidder ένα νέο port number για να συνεχίσει την διαδικασία. Στην συνέχεια θα συνδεθεί στη νέα θύρα που του

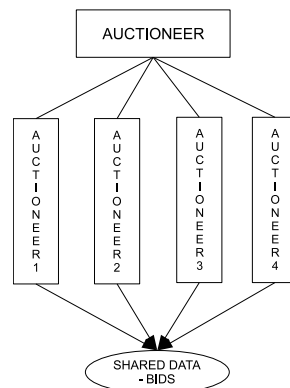
ορίζει ο bidder και θα ξεκινήσει την πραγματοποίηση της δημοπρασίας. Στο σχήμα 3.6 παρουσιάζεται σχηματικά η παραπάνω διαδικασία.



Σχήμα 3.6: Ο auctioneer δημιουργεί 4 threads για τους 4 bidders της δημοπρασίας του. Τα threads συνδέονται στους bidders και αρχίζουν να αλληλεπιδρούν μαζί τους.

Στην αρχή κάθε δημοπρασίας όλα τα threads ενημερώνουν τους bidders σχετικά με τα tasks που δημοπρατούνται. Εν συνεχεία, αναλόγως τον αριθμό των δημοπρατούμενων tasks πραγματοποιούν τον αντίστοιχο αριθμό από auction rounds. Για παράδειγμα, μία δημοπρασία για K tasks χρειάζεται K γύρους για να ολοκληρωθεί. Σε κάθε γύρο, κάθε thread λαμβάνει το bid από τον αντίστοιχο bidder και το αποθηκεύει σε ένα shared data structure. Αφού όλοι οι bidders έχουν υποβάλλει τα bids τους, κάθε thread καθορίζει τον πράκτορα και το task που κέρδισε στο συγκεκριμένο γύρο δημοπρασίας, ενώ μεταφέρει το task t από το set με τα auctioned tasks στο set με τα assigned tasks. Στο τέλος του γύρου, κάθε thread ενημερώνει τον bidder που είναι συνδεδεμένο για τον αν κέρδισε ή όχι στέλνοντας μία θετική ή αρνητική επιβεβαίωση αντίστοιχα. Επιπλέον στέλνει και το ID του task που ανατέθηκε στο συγκεκριμένο γύρο. Η διαδικασία αυτή συνεχίζεται μέχρι όλα τα tasks να μετακινηθούν στο set με τα assigned tasks. Σε αυτό το σημείο ολοκληρώνεται η δημοπρασία και ο auctioneer αποσυνδέεται από τους bidders.

Όλα τα threads λειτουργούν σαν ένας auctioneer. Τα threads συγχρονίζονται έτσι ώστε να βλέπουν το ίδιο στιγμιότυπο των shared μεταβλητών καταλήγοντας στα ίδια αποτελέσματα. Επιπλέον τα threads διασφαλίζουν ότι τα bids περιέχουν τις ενημερωμένες τιμές του τρέχοντος γύρου κάθε φορά. Για παράδειγμα, ένα thread που προηγείται θα μπορούσε να βγάζει αποτέλεσμα βλέποντας τις τιμές των bids του προηγούμενου γύρου για κάποιους bidders. Για τον συγχρονισμό τους και την σωστή διεξαγωγή της δημοπρασίας τα threads χρησιμοποιούν κάποιες shared μεταβλητές ελέγχου, όπως counters και flags, για να καταλαβαίνουν πότε όλα τα threads έχουν φτάσει στα κρίσιμα σημεία και να συνεχίζουν την εκτέλεση του κώδικά τους. Τα κρίσιμα σημεία όπου όλα τα threads 'ευθυγραμμίζονται' είναι η αρχή του auction round και το σημείο μετά την υποβολή των bids και πριν τον καθορισμό του νικητή. Στο σχήμα 3.7 απεικονίζεται η ρουτίνα του auctioneer με τα threads να χρησιμοποιούν shared data για τον συντονισμό τους και τον καθορισμό του νικητή bidder.



Σχήμα 3.7: Τα auctioneer threads εκτελούνται παράλληλα και χρησιμοποιούν κοινές μεταβλητές και δομές δεδομένων για να συγχρονίζονται και να καταλήγουν σε κοινά αποτελέσματα

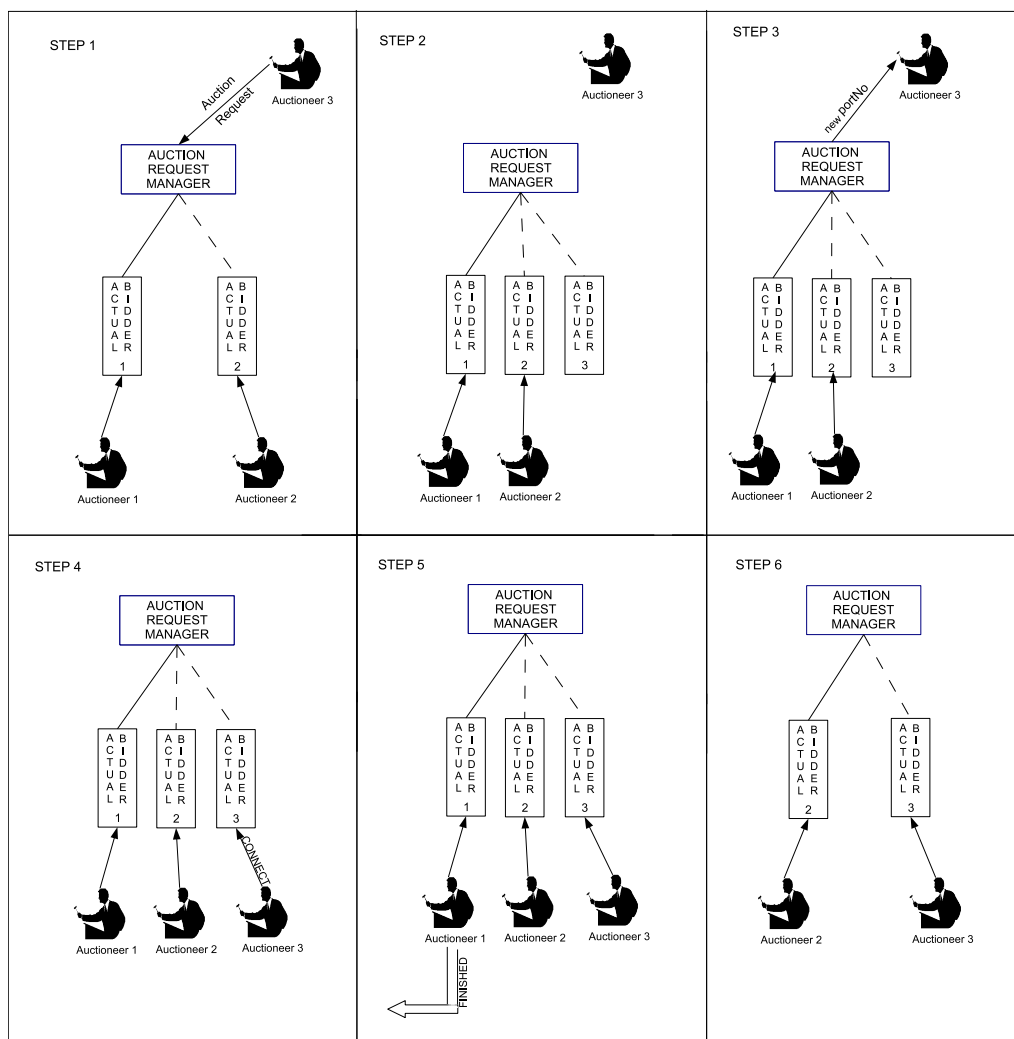
3.6 Η ρουτίνα του bidder

Η ρουτίνα του bidder αποτελεί ένα server που περιμένει τον auctioneer να συνδεθεί και στη συνέχεια αφού επεξεργαστεί το αίτημα του auctioneer για δημοπρασία, προχωρά στην πραγματοποίηση της δημοπρασίας και της υποβολής των bids.

Η εξυπηρέτηση του auctioneer από τον bidder χωρίζεται σε δύο επίπεδα. Το πρώτο επίπεδο είναι το “Auction Request Manager”. Σε αυτό το επίπεδο ο auction request manager του κάθε πράκτορα περιμένει (listens) σε μία συγκεκριμένη προκαθορισμένη θύρα σύμφωνα με το Auction Protocol. Όποτε ένας auctioneer συνδέεται σε κάποιον πράκτορα ζητώντας του να ξεκινήσει μία δημοπρασία, ο auction request manager του πράκτορα στέλνει στον auctioneer ένα νέο port number στο οποίο θα πρέπει να συνδεθεί ο auctioneer για να πραγματοποιήσει την δημοπρασία. Πριν στείλει το port number στον auctioneer, ο auction request manager δημιουργεί ένα νέο thread που περιμένει τον auctioneer να συνδεθεί στην επιλεγμένη θύρα. Το νέο thread λειτουργεί ως bidder αφού αυτό υπολογίζει και στην συνέχεια υποβάλλει το bid του πράκτορα σε κάθε γύρο. Ο request manager στέλνει το port number στον auctioneer μόνο εφόσον το νέο thread είναι έτοιμο για να κάνει accept τον auctioneer. Ενεργώντας κατά αυτόν τον τρόπο, ο request manager εξασφαλίζει ότι ο auctioneer δεν θα επιχειρήσει να συνδεθεί σε κάποιο “ghost” port. Μία προσπάθεια για σύνδεση σε κάποιο “ghost” port θα χαλούσε τον συγχρονισμό των bidders, ενώ είναι πολύ πιθανό να οδηγήσει και σε deadlock καθώς αυτό το “actual bidder” thread δεν θα τερμάτιζε εμποδίζοντας την έναρξη της εκτέλεσης νέων thread.

Το δεύτερο επίπεδο εξυπηρέτησης είναι το “Actual Bidder”. Στο δεύτερο επίπεδο το “actual bidder” thread αρχικά περιμένει τον auctioneer να συνδεθεί. Ο auctioneer αφού συνδεθεί, στέλνει στον bidder όλα τα tasks που

δημοπρατούνται. Το actual bidder thread αφού υπολογίσει τα bids για όλα τα δημοπρατούμενα tasks χρησιμοποιώντας τον generic bidding rule, επιλέγει το ελάχιστο και το στέλνει μαζί με το αντίστοιχο task ID στον auctioneer. Στην συνέχεια περιμένει να του σταλούν τα αποτελέσματα. Στο τέλος κάθε γύρου δημοπρασίας οι bidders ενημερώνουν τα δεδομένα των auctioned tasks, αφαιρώντας το task που ανατέθηκε στο συγκεκριμένο γύρο από το set με τα auctioned tasks, ενώ ο νικητής πράκτορας προσθέτει το task που κέρδισε στο set με τα assigned σε αυτόν tasks. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να ανατεθούν όλα τα tasks, με τα actual bidder threads να υπολογίζουν τα bids για τα εναπομείναντα auctioned tasks σε κάθε γύρο. Μετά την ολοκλήρωση της δημοπρασίας, το actual bidder thread για να τερματίσει την εκτέλεσή του πρέπει να ανανεώσει τα δεδομένα σχετικά με την σειρά εξυπηρέτησης έτσι ώστε να μπορεί να πραγματοποιηθεί το επόμενο auction στην ουρά αναμονής. Στο σχήμα 3.8 φαίνεται η λειτουργία του bidder σε επίπεδο thread και δείχνει την σειρά πραγματοποίησης των δημοπρασιών.



Σχήμα 3.8: Η συνολική λειτουργία του bidder. Η συνεχής γραμμή που συνδέει το Auction Request Manager thread με το actual bidder thread σημαίνει ότι το bidder thread εκτελείται ενώ η διακεκομμένη ότι βρίσκεται σε αναμονή.

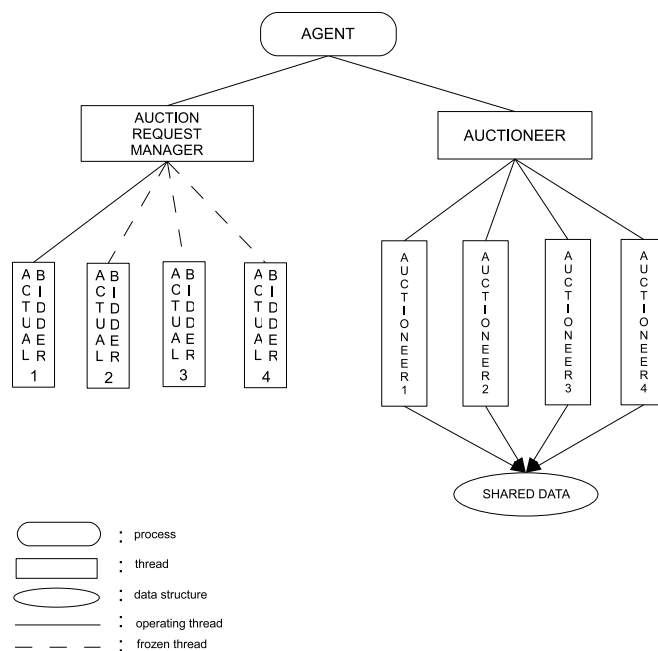
Επειδή τα actual bidder threads που αντιστοιχούν σε διαφορετικούς auctioneer εξυπηρετούνται στη σειρά με κανένα να μην μπορεί να προσπεράσει και επειδή ο auctioneer διαχειρίζεται μία δημοπρασία την φορά και χρειάζεται όλα τα bids σε κάθε γύρο, υπάρχει ο κίνδυνος εμφάνισης deadlock. Εάν ένας auctioneer A_1 είναι πρώτος στην ουρά εξυπηρέτησης του bidder B_1 και δεύτερος στην ουρά του bidder B_2 ενώ ένας άλλος auctioneer A_2 είναι δεύτερος στην

ουρά του B_1 και πρώτος στην ουρά του B_2 , τότε οδηγούμαστε σε deadlock. Το actual bidder thread του B_1 δεν θα τερματίσει αφού δεν θα λάβει αποτελέσματα από τον auctioneer A_1 . Το ίδιο θα συμβεί και με το thread του B_2 . Οι δύο auctioneers θα περιμένουν ο ένας τον άλλο για πάντα και οι auctioneers που ακολουθούν στην ουρά δεν θα μπορούν να ξεκινήσουν τις δικές τους δημοπρασίες.

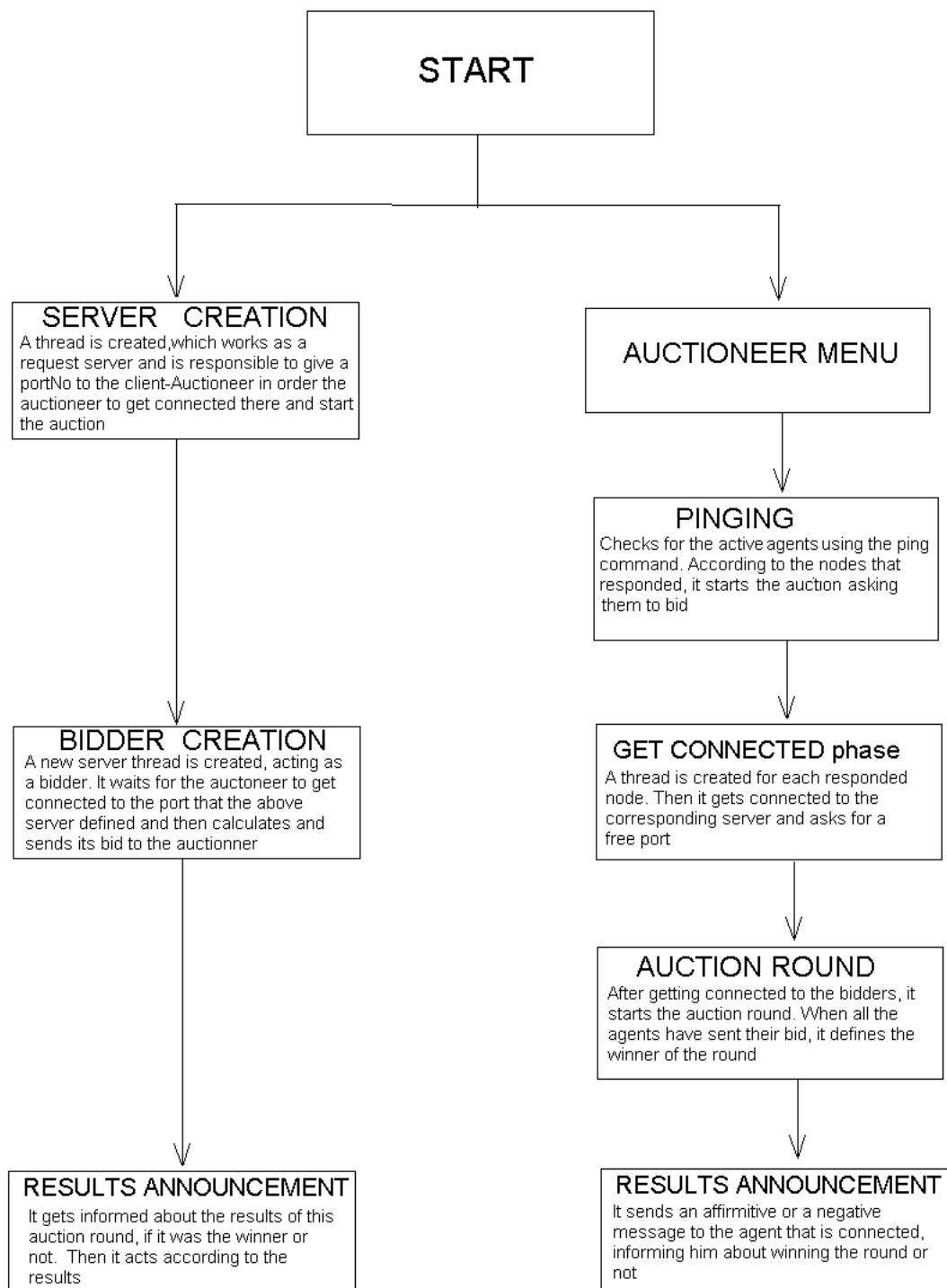
Για να συμβεί τέτοιο deadlock θα πρέπει το μηχανήμα στο οποίο εκτελείται η ρουτίνα του auctioneer A_2 να είναι αρκετά γρηγορότερο από το μηχανήμα του A_1 . Ο auctioneer A_1 θα συνδεθεί πρώτος στον bidder B_1 και στη συνέχεια θα συνδεθεί ο A_2 , αλλά το μηχανήμα του A_2 θα εκτελέσει πολύ γρηγορότερα τις ίδιες εντολές με τον A_1 υπερνικώντας την χρονική διαφορά που είχαν στον B_1 κι έτσι ο A_2 θα συνδεθεί στον B_2 πριν τον A_1 . Αυτή η περίπτωση deadlock μπορεί να αντιμετωπιστεί με την χρήση timeout signals. Με timeout signals αντιμετωπίζεται και η περίπτωση της προσπάθειας σύνδεσης σε “ghost” port.

Τα threads και του auctioneer αλλά και των bidders ενεργοποιούν ένα timer στην αρχή κάθε auction round. Τα threads του auctioneer χρησιμοποιούν ένα κοινό timer και όποτε λάβουν κάποιο timeout signal, καθορίζουν τον νικητή του γύρου υπολογίζοντας μόνο τα bids που έχουν υποβληθεί ως εκείνη την στιγμή. Η πραγματοποίηση της δημοπρασίας συνεχίζεται κανονικά. Κάθε actual bidder thread χρησιμοποιεί ένα timer και όποτε λάβει ένα timeout signal τερματίζει την εκτέλεσή του με τον πράκτορα να θεωρεί ότι η δημοπρασία ολοκληρώθηκε. Ο πράκτορας αυτός διατηρεί τα tasks που του είχαν ανατεθεί αλλά δεν συμμετέχει στους επόμενους γύρους.

Έχοντας περιγράψει όλα τα επιμέρους κομμάτια ενός αυτόνομου πράκτορα και τον τρόπο αλληλεπίδρασης του με τους υπόλοιπους πράκτορες, γίνεται πλέον κατανοητή η συνολική λειτουργία του πράκτορα. Παρατίθενται δύο σχήματα που παρουσιάζουν την λειτουργία ενός πράκτορα. Στο σχήμα 3.9 απεικονίζεται το στιγμιότυπο ενός πράκτορα σε επίπεδο ρουτίνας και thread. Στην ρουτίνα του bidder, ο πράκτορας έχει δηλώσει ότι συμμετέχει σε τέσσερις δημοπρασίες οπότε έχουν δημιουργηθεί τέσσερα actual bidder threads. Η συνεχής γραμμή υποδηλώνει ότι το συγκεκριμένο thread εκτελείται ενώ η διακεκομμένη σημαίνει ότι το συγκεκριμένο thread είναι σε αναμονή. Ταυτόχρονα, ο πράκτορας πραγματοποιεί μία δημοπρασία. Στην ρουτίνα του auctioneer έχουν δημιουργηθεί τέσσερα threads που σημαίνει ότι συμμετέχουν τέσσερις bidders σε αυτή τη δημοπρασία. Στο σχήμα 3.10 είναι το διάγραμμα ροής του προγράμματος του πράκτορα, με τις δύο διαφορετικές ροές να πραγματοποιούνται ταυτόχρονα.



Σχήμα 3.9: Ένα στιγμιότυπο του προγράμματος του πράκτορα σε επίπεδο thread. Η συνεχόμενη γραμμή σε κάποιο thread σημαίνει ότι αυτό εκτελείται ενώ η διακεκομμένη ότι είναι 'παγωμένο' και βρίσκεται σε αναμονή.



Σχήμα 3.10: Το διάγραμμα ροής του προγράμματος του πράκτορα

Κεφάλαιο 4

Εφαρμογές

Ο σκοπός της υλοποίησης του πυρήνα που περιγράφεται στο κεφάλαιο 3 είναι η αμέριστη χρησιμοποίησή του για την επίλυση προβλημάτων ανάθεσης εργασιών μέσω δημοπρασιών σε διάφορες εφαρμογές. Αναλόγως την εφαρμογή ο σχεδιαστής του συστήματος χρειάζεται να κάνει απλά κάποιες προσθήκες ή κάποιες μικρές αλλαγές σχετικά με τις συναρτήσεις υπολογισμού των bids των πρακτόρων. Σε αυτή την διπλωματική εργασία ο πυρήνας χρησιμοποιήθηκε για την ανάθεση εργασιών σε δύο διαφορετικές εφαρμογές οι οποίες περιγράφονται στο παρόν κεφάλαιο.

4.1 Καθορισμός διαδρομών ομάδας από robots

Στη μία εκ των δύο εφαρμογών υπάρχει μία ομάδα από robots, τα οποία συνεργάζονται μεταξύ τους ώστε να επισκεφθούν ένα σύνολο κόμβων ελαχιστοποιώντας το συνολικό ομαδικό κόστος. Τα robots χρησιμοποιούν τον αλγόριθμο δημοπρασίας για να καθορίσουν τις διαδρομές τους.

4.1.1 Περιγραφή σεναρίου

Στο σενάριο της εφαρμογής αυτής υπάρχουν N mobile robots τα οποία χρειάζεται να επισκεφθούν M κόμβους. Το σενάριο αυτό μπορεί να έχει διάφορες εκδοχές με ναυαγοσωστικά σκάφη, πυροσβεστικά οχήματα, rovers αντί για robots και ναυάγια, εστίες φωτιάς και τοποθεσίες συλλογής δειγμάτων αντί για κόμβους. Στην συγκεκριμένη περιγραφή θεωρούμε τους πράκτορες και τα tasks των προηγούμενων κεφαλαίων σαν robots και κόμβους αντίστοιχα.

Κάθε robot έχει την δική του IP διεύθυνση, είναι αυτόνομο και μπορεί να κινείται σε φυσικό περιβάλλον. Επιπλέον, τα robots είναι εξοπλισμένα με τον αρχικό χάρτη του περιβάλλοντος στο οποίο κινούνται ενώ γνωρίζουν τις συντεταγμένες των κόμβων και μπορούν να ορίσουν την θέση τους στο χάρτη. Ο αρχικός χάρτης τον οποίο διαθέτουν τα robots μπορεί να αναπτύσσεται δυναμικά ανάλογα με τις πληροφορίες που συλλέγουν τα robots κατά την πλοήγησή στο χώρο δράσης τους και οι οποίες πληροφορίες δεν επαληθεύουν τα δεδομένα του χάρτη. Για παράδειγμα, εάν στο χάρτη υπάρχει προσβάσιμο μονοπάτι και στην πραγματικότητα έχει κλείσει από κάποιο κορμό δέντρου, τότε τα δεδομένα του χάρτη πρέπει να ενημερωθούν.

Κάθε κόμβος πρέπει να προσπελαστεί από τουλάχιστον ένα robot. Αρχικά τα robots διαμοιράζουν το σύνολο των κόμβων και κάθε robot αναπτύσσει μία διαδρομή ανάλογα με τους κόμβους που έχει αναλάβει. Μετά την ανάθεση όλων των κόμβων, τα robots ξεκινούν να τους προσεγγίζουν ανάλογα με τις διαδρομές που έχουν ορίσει. Ωστόσο, υπάρχει η πιθανότητα λάθους εκτίμησης στην ανάληψη των κόμβων και συνεπώς στην δημιουργία των διαδρομών είτε εξαιτίας ελλειπούς πληροφορίας του χάρτη είτε λόγω φυσικών παραγόντων. Για παράδειγμα κάποιο robot θα μπορούσε να κολλήσει στη λάσπη. Στην περίπτωση αυτή το robot αντιλαμβάνεται ότι η προσέγγιση ενός ή παραπάνω κόμβων οδηγεί σε μεγαλύτερο κόστος από όσο είχε αρχικά εκτιμηθεί και ζητά την ανάθεση

αυτών των κόμβων σε άλλα robots.

Το σενάριο της συγκεκριμένης εφαρμογής βασίζεται στην δημοσίευση [1]. Η κατανομή των κόμβων πραγματοποιείται σύμφωνα με κάποιο team objective. Στην ίδια δημοσίευση αναλύονται τρία team objectives για την κατανομή των κόμβων στα robots, το MINISUM, το MIMIMAX και το MINIAVE. Το MINISUM team objective είναι καταλληλότερο εάν θέλουμε να ελαχιστοποιήσουμε την συνολική κατανάλωση ενέργειας. Αν θεωρούμε τον χρόνο επίσκεψης κάποιου κόμβου σαν κόστος, το MINIMAX είναι πιο κατάλληλο για να ελαχιστοποιήσουμε τον μέγιστο χρόνο αναμονής κάποιου κόμβου και το MINIAVE για να ελαχιστοποιήσουμε τον μέσο χρόνο αναμονής των κόμβων.

4.1.2 Υλοποίηση

Αφού αποκτήσουν τον χάρτη με τις θέσεις των κόμβων, τα robots καθορίζουν ποια από αυτά είναι διαθέσιμα για την αποστολή. Μετά την συγκρότηση της ομάδας που θα αναλάβει την αποστολή, τα robots αρχίζουν να επικοινωνούν μεταξύ τους έτσι ώστε να καθορίσουν τις διαδρομές τους. Για να κατανείμουν τους κόμβους και να ορίσουν τις διαδρομές τους, τα robots πραγματοποιούν ψευδοκεντρικοποιημένες δημοπρασίες.

Σύμφωνα με το κεφάλαιο 2 σε κάθε δημοπρασία χρειάζεται ένας auctioneer που θα διευθύνει την διαδικασία και θα συγχρονίσει τους bidders. Ωστόσο, αποτελεί σημαντικό μειονέκτημα για την ομάδα των robots να εξαρτάται από ένα κεντρικό robot διότι εάν αυτό δεν λειτουργεί ή έχει υποστεί κάποια ζημιά, τα υπόλοιπα robots θα αποτύχουν να φέρουν σε πέρας την αποστολή. Εφόσον τα robots θεωρούνται 'ίσα' μεταξύ τους και διατηρούν την ίδια πληροφορία, οποιοδήποτε από αυτά μπορεί να γίνει auctioneer και να υλοποιήσει δημοπρασία για τους κόμβους. Έτσι, αν διαιρέσουν το αρχικό set από τους δημοπρατούμενους κόμβους σε K μικρότερα sets, τότε K robots θα πραγμα-

τοποιάσουν K δημοπρασίες με M/K κόμβους η καθεμία και στο τέλος όλοι οι κόμβοι θα έχουν ανατεθεί. Η τιμή του K κυμαίνεται από 1 μέχρι N . Τα robots ορίζουν ποιοι θα είναι οι auctioneers και με αυτό τον τρόπο εξασφαλίζεται ότι η αποστολή θα επιτευχθεί αφού τα robots είναι ανεξάρτητα από κάποιον προκαθορισμένο αρχηγό. Για παράδειγμα, θα μπορούσαν να επιλέγονται για auctioneers τα robots που βρίσκονται στις πιο κεντρικές θέσεις.

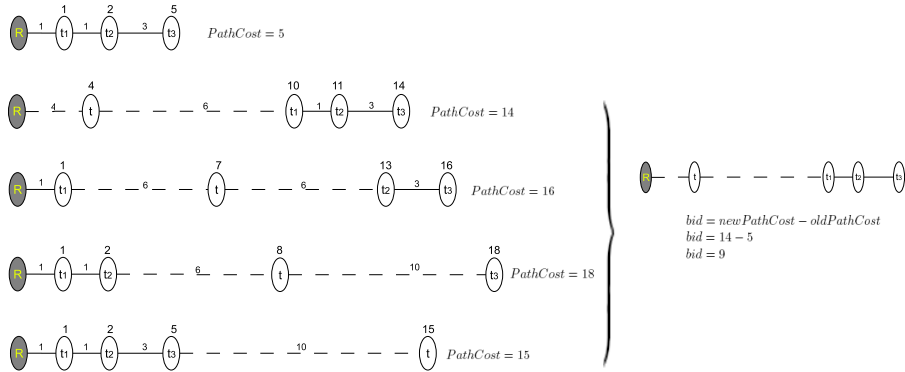
Η τιμή του K επιλέγεται από τον σχεδιαστή του συστήματος. Χαμηλότερη τιμή του K δίνει πιο ποιοτικά αποτελέσματα αφού τα robots διαθέτουν περισσότερες πληροφορίες στην αρχή της δημοπρασίας και κάνουν ορθότερες επιλογές σαν bidders. Στην περίπτωση μας προτεραιότητα έχει η μεγιστοποίηση της απόδοσης του αλγορίθμου δημοπρασίας και η βελτιστοποίηση των αποτελεσμάτων. Για αυτό το λόγο τα robots επιλέγουν ένα από αυτά για να εκτελέσει χρέη auctioneer, δηλαδή $K = 1$. Η διαδικασία της δημοπρασίας για την κατανομή των κόμβων είναι ίδια με αυτή που περιγράφηκε στο κεφάλαιο 3, οπότε δεν θα αναφερθούμε σε αυτή παρά μόνο στον τρόπο υπολογισμού των bids από τα robots.

Υπολογισμός Bid : Τα robots γνωρίζοντας τις θέσεις των κόμβων καθώς και τις δικές τους, έχουν την δυνατότητα να υπολογίσουν όλα τα κόστη μεταβασης είτε μεταξύ των θέσεών τους και των κόμβων είτε μεταξύ των κόμβων. Σε αυτή την υλοποίηση τα robots αποκτούν αυτές τις αποστάσεις μαζί με τον χάρτη στην αρχή, πριν ξεκινήσουν την διαδικασία για δημοπρασία. Τα robots υπολογίζουν τα bids για όλους τους δημοπρατούμενους κόμβους σύμφωνα με τον bidding rule που περιγράφεται στο κεφάλαιο 2.

Τα robots υπολογίζουν το bid για κάθε κόμβο προσθέτοντάς τον στο μονοπάτι που έχουν αναπτύξει με τους ήδη ανατετιμένους σε αυτά κόμβους. Ο κόμβος τοποθετείται σε όλες τις πιθανές θέσεις μεταξύ των κόμβων του μο-

νοπατιού και εκτιμάται το κόστος για κάθε περίπτωση. Για παράδειγμα, για ένα μονοπάτι που αποτελείται από K κόμβους υπολογίζονται τα κόστη $(K+1)$ πιθανών μονοπατιών. Στη συνέχεια επιλέγεται το μονοπάτι με το ελάχιστο κόστος απο το οποίο προκύπτει και η τιμή του bid για τον κόμβο. Αφού έχουν υπολογίσει τα bids για όλους τους δημοπρατούμενους κόμβους, τα robots επιλέγουν το bid με την ελάχιστη τιμή και το στέλνουν στον auctioneer μαζί με το αντίστοιχο ID του κόμβου (targetID).

- MINISUM : Για το MINISUM team objective το bid του κάθε robot είναι η διαφορά μεταξύ του κόστους του νέου μονοπατιού και του κόστους του παλαιού μονοπατιού.

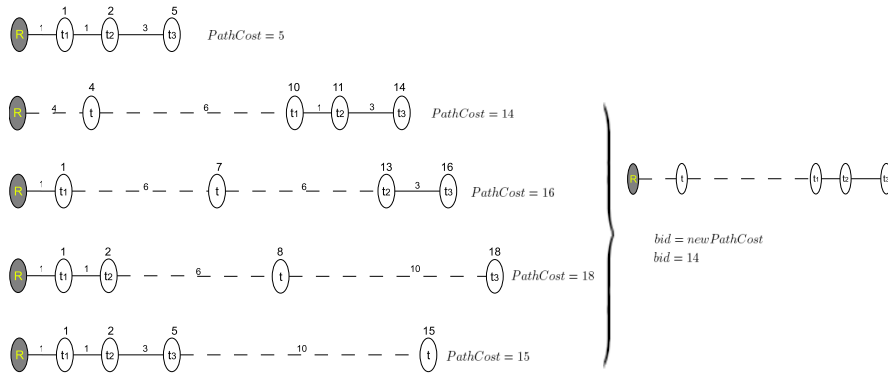


Σχήμα 4.1: Διαδικασία υπολογισμού του bid για το MINISUM team objective

Στο σχήμα 4.1 παρουσιάζεται η διαδικασία υπολογισμού του bid για ένα κόμβο t σύμφωνα με το MINISUM team objective. Δεδομένου ότι το robot r έχει αναλάβει την προσπέλαση των κόμβων t_1, t_2, t_3 , οι οποίοι σχηματίζουν ένα μονοπάτι, εισάγει τον t σε όλες τις πιθανές θέσεις του μονοπατιού και εκτιμάει το κόστος εισαγωγής του t για όλες τις περιπτώσεις. Στη συνέχεια επιλέγει το μονοπάτι με το ελάχιστο κόστος και υπολογίζει το αντίστοιχο bid για τον κόμβο t . Με αυτό τον τρόπο, ένα robot υπολογίζει τα bids για όλους τους

μη αναθετιμένους κόμβους και επιλέγει το ελάχιστο, το οποίο είναι το bid του robot για ένα γύρο δημοπρασίας. Ομοίως και για τα MINIMAX και MINIAVE team objectives τα οποία παρουσιάζονται στα σχήματα 4.2 και 4.3.

- MINIMAX : Για το MINIMAX team objective το bid για κάθε robot είναι το κόστος του νέου μονοπατιού.



Σχήμα 4.2: Διαδικασία υπολογισμού του bid για το MINIMAX team objective

- MINIAVE : Για το MINIMAX team objective, το bid για κάθε robot είναι η διαφορά μεταξύ του νέου αθροίσματος των κοστών προσπέλασης για τους κόμβους του μονοπατιού και του προηγούμενου αθροίσματος των κοστών προσπέλασης για τους κόμβους του μονοπατιού.

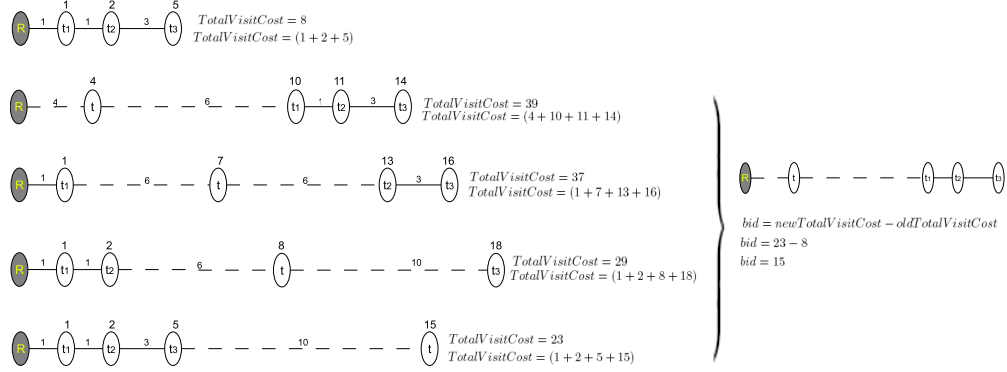
$$new \sum_{A_{r_i} \cup \{t\}} c_j - old \sum_{A_{r_i}} c_j$$

όπου

c_j : το κόστος προσπέλασης για κάθε κόμβο j

A_{r_i} : το set με τους ήδη αναθετιμένους κόμβους στο r_i , το οποίο αποτελεί το μονοπάτι του r_i

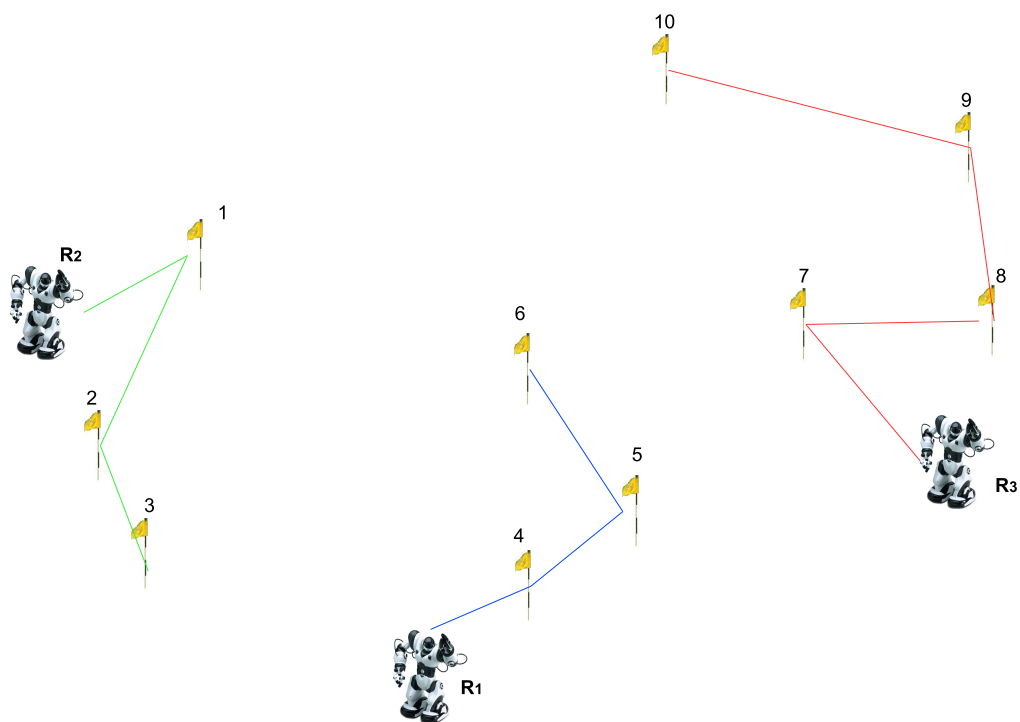
$A_{r_i} \cup \{t\}$: το νέο set που δημιουργείται με την προσθήκη του κόμβου t στο set A_{r_i}



Σχήμα 4.3: Διαδικασία υπολογισμού του bid για το MINIAVE team objective

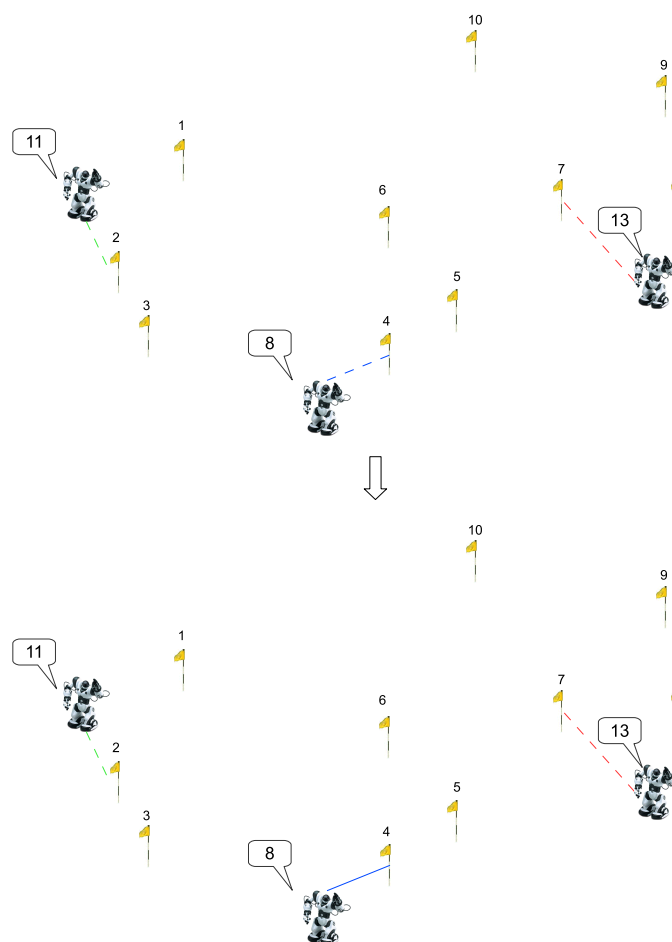
4.1.3 Παράδειγμα ανάθεσης κόμβων

Θεωρούμε τα robots του σχήματος 4.4, τα οποία πρέπει να προσπελάσουν τους κόμβους του ιδίου σχήματος. Στο σχήμα αυτό φαίνεται επίσης η επιθυμητή ανάθεση των κόμβων καθώς και οι αντίστοιχες διαδρομές των robots. Τα robots υλοποιούν μία δημοπρασία για να κατανεύουν όλους τους κόμβους του σχήματος 4.4. Οι δέκα κόμβοι είναι το αρχικό set των δημοπρατούμενων αντικειμένων και τα τρία robots είναι οι bidders της δημοπρασίας. Έστω ότι το robot R_1 είναι ο auctioneer επειδή βρίσκεται στην πιο κεντρική θέση. Η διαδικασία και τα αποτελέσματα της δημοπρασίας είναι τα ίδια ασχέτως απο το ποιο robot είναι ο auctioneer. Σε αυτό το παράδειγμα τα robots υπολογίζουν τα bids τους σύμφωνα με το MINISUM team objective.



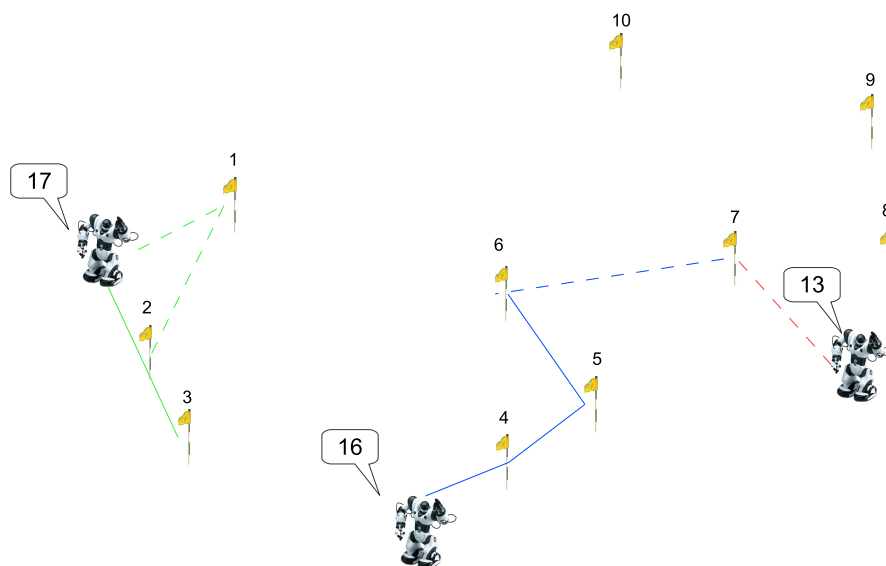
Σχήμα 4.4: Η επιθυμητή ανάθεση των 10 κόμβων στην ομάδα που αποτελείται από τα robots R_1 , R_2 και R_3

Στον πρώτο γύρο τα robots κάνουν bid στους κόμβους 2, 4 και 7 όπως φαίνεται και στο σχήμα 4.5. Το robot R_1 κάνει την καλύτερη προσφορά και αναλαμβάνει τον κόμβο 4. Το R_1 προσθέτει τον κόμβο στο μονοπάτι του, στη θέση που ελαχιστοποιεί το κόστος του μονοπατιού. Στη συγκεκριμένη περίπτωση δεν παίζει ρόλο αφού το μονοπάτι είναι άδειο και ο κόμβος 4 μπαίνει στην πρώτη θέση. Η ίδια διαδικασία ακολουθείται και στους επόμενους γύρους.

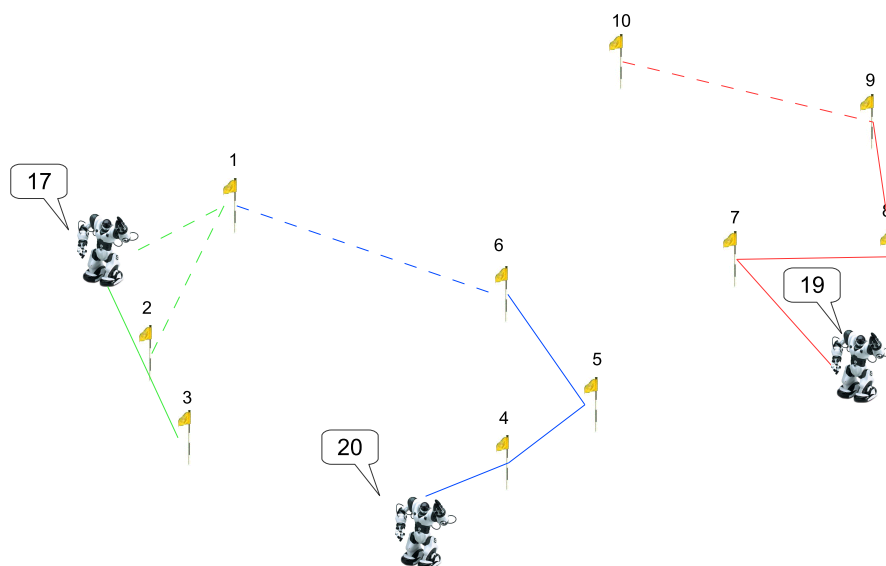


Σχήμα 4.5: Ο πρώτος γύρος της δημοπρασίας για την ανάθεση των κόμβων

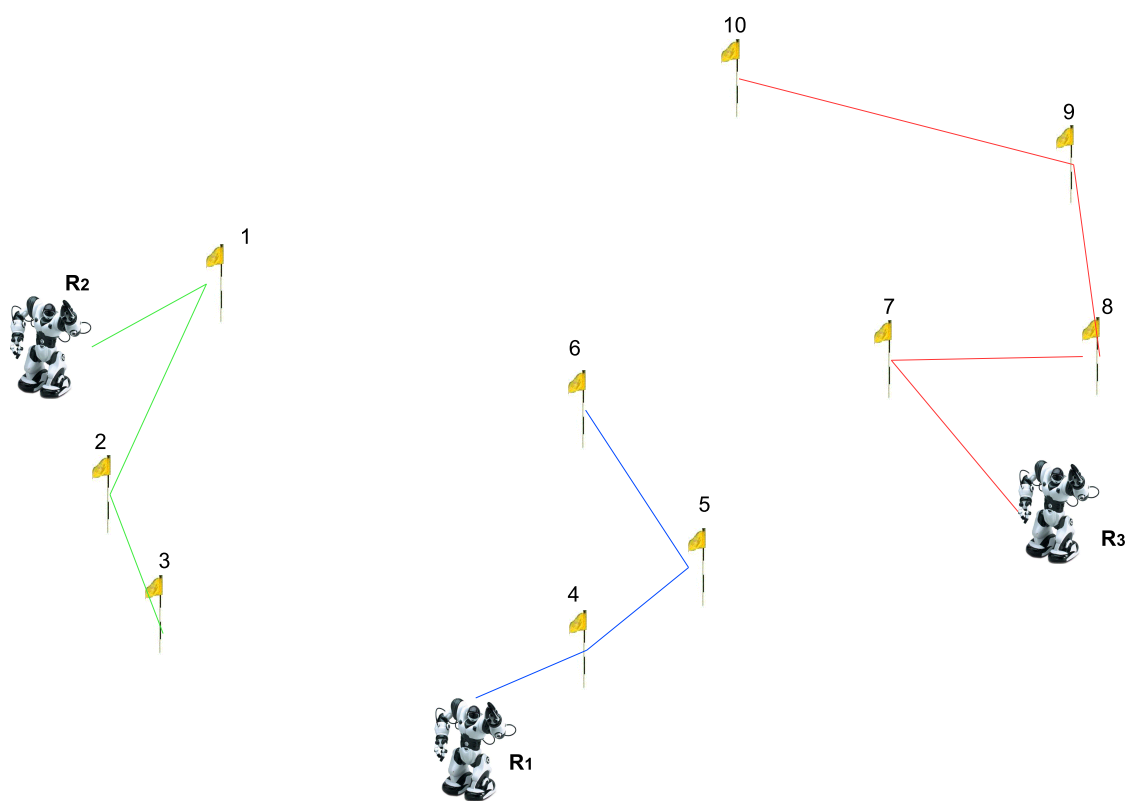
Στο σχήμα 4.7 έχουμε μία περίπτωση διεκδίκησης του ίδιου κόμβου, του 7, από δύο robots, τα R_1 και R_3 . Το R_3 κάνει καλύτερη προσφορά από το R_1 κι έτσι αναλαμβάνει τον κόμβο 7. Επίσης στο σχήμα 4.6 έχουμε περίπτωση διεκδίκησης κόμβου από δύο robots, το R_1 και το R_2 . Βλέπουμε ότι το R_2 αναδιατάσσει το τρέχον μονοπάτι του για να προσθέσει τον κόμβο 1 σε αυτό. Τέλος, στο σχήμα 4.8 φαίνεται το τελικό αποτέλεσμα της ανάθεσης των κόμβων στα robots. Στον πίνακα 4.1 παρουσιάζονται όλοι οι γύροι της δημοπρασίας για την ανάθεση των κόμβων.



Σχήμα 4.6: Σε αυτό το γύρο τα robots R_1 και R_3 συναγωνίζονται για τον κόμβο 7



Σχήμα 4.7: Τα robots R_1 και R_2 διεκδικούν τον κόμβο 1 με νικήτη το R_2



Σχήμα 4.8: Το τελικό αποτέλεσμα

Πίνακας 4.1: Όλα τα βήματα της δημοπρασίας μέχρι την τελική ανάθεση του σχήματος 4.4

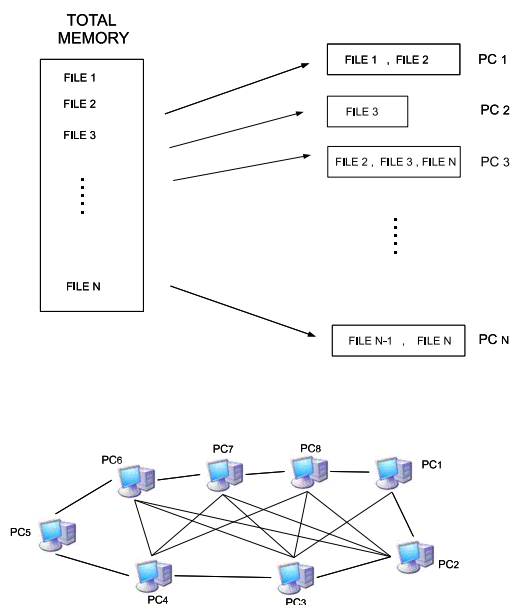
<i>Unallocated Targets</i>	<i>Auctioneer</i>	<i>Robot1</i>	<i>Robot2</i>	<i>Robot3</i>
set $\mathbf{W} = \{t_1, \dots, t_N\}$	$\min B = \{b_1, \dots, b_N\}, \arg \min B$	$(\min_{j \in W} b_{rj}, \arg \min_{j \in W} b_{rj})$ $path = \{t_1, \dots\}$		
$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$	$8 = \min \{8, 11, 13\}, t = 4$	(8, 4)	(11, 2)	(13, 7)
		{4}	{}	{}
$\{1, 2, 3, 5, 6, 7, 8, 9, 10\}$	$7 = \min \{7, 11, 13\}, t = 5$	(7, 5)	(11, 2)	(13, 7)
		{4, 5}	{}	{}
$\{1, 2, 3, 6, 7, 8, 9, 10\}$	$11 = \min \{12, 11, 13\}, t = 2$	(12, 6)	(11, 2)	(13, 7)
		{4, 5}	{2}	{}
$\{1, 3, 6, 7, 8, 9, 10\}$	$9 = \min \{12, 9, 13\}, t = 3$	(12, 6)	(9, 3)	(13, 7)
		{4, 5}	{2, 3}	{}
$\{1, 6, 7, 8, 9, 10\}$	$12 = \min \{12, 17, 13\}, t = 6$	(12, 6)	(17, 1)	(13, 7)
		{4, 5, 6}	{2, 3}	{}
$\{1, 7, 8, 9, 10\}$	$13 = \min \{16, 17, 13\}, t = 7$	(16, 7)	(17, 1)	(13, 7)
		{4, 5, 6}	{2, 3}	{7}
$\{1, 8, 9, 10\}$	$10 = \min \{20, 17, 10\}, t = 8$	(20, 1)	(17, 1)	(10, 8)
		{4, 5, 6}	{2, 3}	{7, 8}
$\{1, 9, 10\}$	$13 = \min \{20, 17, 13\}, t = 9$	(20, 1)	(17, 1)	(13, 9)
		{4, 5, 6}	{2, 3}	{7, 8, 9}
$\{1, 10\}$	$17 = \min \{20, 17, 19\}, t = 1$	(20, 1)	(17, 1)	(19, 10)
		{4, 5, 6}	{1, 2, 3}	{7, 8, 9}
$\{10\}$	$19 = \min \{23, 29, 19\}, t = 10$	(23, 10)	(29, 10)	(19, 10)
		{4, 5, 6}	{1, 2, 3}	{7, 8, 9, 10}
$\{\}$	Allocation Completed			

4.2 Computer File Sharing

Στην δεύτερη εφαρμογή υπολογιστές είναι συνδεδεμένοι σε δίκτυο και μοιράζονται κοινόχρηστα αρχεία. Όταν ένας υπολογιστής χρειάζεται κάποια αρχεία, ενημερώνει τους υπόλοιπους υπολογιστές στο δίκτυο και όποιοι τα διαθέτουν, αναλαμβάνουν να του τα στείλουν. Η επιλογή των υπολογιστών που θα αναλάβουν να μεταδώσουν τα αρχεία γίνεται μέσω δημοπρασίας ώστε να επιτευχθεί η ταχύτερη μεταφορά.

4.2.1 Περιγραφή σεναρίου

Οι συσκευές αποθήκευσης των υπολογιστών του δικτύου λειτουργούν στο σύνολό τους σαν μία τεράστια μνήμη που διατηρεί όλα τα αρχεία που θα μπορούσε να χρειαστεί κάποιος από τους υπολογιστές. Τα αρχεία δεν βρίσκονται σε κάποιο συγκεκριμένο υπολογιστή αλλά μπορούν να βρίσκονται σε οποιοδήποτε υπολογιστή του δικτύου. Κάθε υπολογιστής αποτελεί κόμβο δικτύου και ξεχωριστό σταθμό εργασίας ταυτόχρονα. Σαν σταθμός εργασίας θα μπορούσε να χρειαστεί κάποιο αρχείο από το δίκτυο σε οποιαδήποτε στιγμή. Όταν ένας υπολογιστής χρειάζεται να χρησιμοποιήσει κάποιο αρχείο που δεν διατηρεί στις δικές του συσκευές αποθήκευσης, τότε το αναζητά στο δίκτυο ενημερώνοντας τους υπόλοιπους κόμβους. Το αρχείο μπορεί να βρίσκεται σε περισσότερους από έναν κόμβους. Ο κόμβος - υπολογιστής που μπορεί να κάνει την μεταφορά του αρχείου ταχύτερα από τους υπόλοιπους αναλαμβάνει να το μεταδώσει στον υπολογιστή που το ζήτησε.



Σχήμα 4.9: Όλοι οι υπολογιστές του δικτύου λειτουργούν σαν ένα τεράστιο αποθηκευτικό μέσο

4.2.2 Υλοποίηση

Όταν κάποιος υπολογιστής χρειάζεται να προσπελάσει ένα αρχείο, αρχικά ελέγχει εάν αυτό βρίσκεται σε κάποια από τις τοπικές συσκευές αποθήκευσής του. Στην περίπτωση που το αρχείο δεν βρίσκεται στις συσκευές του υπολογιστή, τότε ο υπολογιστής προσπαθεί να το αποκτήσει όσο το δυνατόν συντομότερα. Για να επιτευχθεί η ταχύτερη μετάδοση, ο υπολογιστής πραγματοποιεί μία δημοπρασία για το συγκεκριμένο αρχείο.

Σε αυτή την εφαρμογή οι υπολογιστές και τα αρχεία αποτελούν τους πράκτορες και τα tasks της δημοπρασίας του κεφαλαίου 3 αντίστοιχα. Ο υπολογιστής που ζητάει κάποιο αρχείο από το δίκτυο θα εκτελέσει χρέη auctioneer στην δημοπρασία που πραγματοποιείται. Αρχικά θα καθορίσει τους διαθέσιμους πράκτορες για την δημοπρασία χρησιμοποιώντας την εντολή “ping” σε όλους τους δηλωμένους κόμβους δικτύου. Οι υπολογιστές αυτοί αποτελούν τους bidders της δημοπρασίας. Όπως και στην περιγραφή της εφαρμογής των robots, δεν θα επεκταθούμε σε περαιτέρω ανάλυση της διαδικασίας της δημοπρασίας και θα αναλύσουμε μόνο τον τρόπο υπολογισμού των bids.

Υπολογισμός Bid : Οι υπολογιστές υπολογίζουν την τιμή των bids τους λαμβάνοντας υπόψη τρεις παράγοντες :

1. το μέγεθος του αρχείου
2. το transfer rate της γραμμής που συνδέει τους δύο κόμβους (αποστολέα - παραλήπτη)
3. το φόρτο μεταφοράς τους στη δεδομένη χρονική στιγμή

Ο auctioneer αφού λάβει τα αποτελέσματα από το ping που επιχείρησε, εξάγει την τιμή του latency και την στέλνει στους αντίστοιχους bidders. Γνωρίζοντας την τιμή του latency, οι bidders μπορούν να εκτιμήσουν το transfer

rate της γραμμής που τους συνδέει με τον auctioneer. Για το μέγεθος του αρχείου ενημερώνονται μέσω μίας κλήσης συστήματος που πραγματοποιούν.

Σχετικά με την πληροφορία του φόρτου μεταφοράς, κάθε υπολογιστής διατηρεί ένα κατάλογο με τα ονόματα και τα μεγέθη των αρχείων που έχει αναλάβει να μεταδώσει και δεν έχει ολοκληρωθεί η μεταφορά τους. Για την αποφυγή σύγχυσης θα αναφερόμαστε στο αρχείο αυτό με τον όρο FTI (File Transfer Information). Το FTI λειτουργεί ως μία λίστα FIFO αφού η σειρά με την οποία εισάγονται σε αυτό τα αρχεία υποδεικνύει τη σειρά μετάδοσής τους. Το FTI ενημερώνεται συνεχώς από τον υπολογιστή του για την εξέλιξη της μεταφοράς των αρχείων. Όταν ο συγκεκριμένος υπολογιστής κερδίσει σε κάποιο γύρο δημοπρασίας και αναλάβει την μετάδοση του ενός αρχείου, εισάγει τα στοιχεία του αρχείου αυτού, όπως όνομα και μέγεθος, στο τέλος του FTI. Επειδή όλα τα αρχεία έχουν την ίδια προτεραιότητα όσον αφορά την μετάδοσή τους, κάθε νέο αρχείο μπαίνει στο τέλος του FTI κι έτσι τα αρχεία μεταδίδονται με την σειρά που ανατέθηκαν στον υπολογιστή. Στην περίπτωση ολοκλήρωσης της μεταφοράς ενός αρχείου, τα στοιχεία του αρχείου αυτού διαγράφονται από το FTI και έχουμε ενημέρωση των δεδομένων του FTI για να αρχίσει η μετάδοση του επόμενου αρχείου.

Στη συγκεκριμένη εφαρμογή μας ενδιαφέρει να ελαχιστοποιήσουμε τον μέσο χρόνο μετάδοσης ενός αρχείου. Για αυτό το λόγο σε μία δημοπρασία δίνεται προτεραιότητα σε μικρού μεγέθους αρχεία, τα οποία χρειάζονται λιγότερο χρόνο για να μεταδοθούν. Κάθε bidder δηλαδή ταξινομεί βάσει μεγέθους τα αρχεία που αναλαμβάνει στα πλαίσια μίας δημοπρασίας. Στην εξίσωση 4.1 οι όροι TransferLoad και latency παραμένουν σταθεροί σε κάθε γύρο δημοπρασίας με αποτέλεσμα η επιλογή του μικρότερου bid να οδηγεί στην επιλογή του μικρότερου σε μέγεθος αρχείου. Αν θέλαμε να παραλληλίσουμε τις δύο εφαρμογές, σε αυτή την εφαρμογή οι υπολογιστές μπορούν να εισάγουν ένα νέο task μόνο

στο τέλος του μονοπατιού με τα ήδη ανατετιμένα σε αυτούς tasks. Με αυτό τον τρόπο ο bidder αναπτύσσει την ουρά αναμονής των αρχείων προς μετάδοση. Η τιμή του bid που θα καταθέσει κάποιος υπολογιστής για ένα αρχείο υπολογίζεται ως εξής :

$$bid = TransferLoad + (filesize * latency) \quad (4.1)$$

όπου

bid : η τιμή του bid για το αρχείο

TransferLoad : ο φόρτος μεταφοράς του υπολογιστή για τα ήδη ανατετιμένα αρχεία σε αυτόν

filesize : το μέγεθος του αρχείου

latency : η τιμή της καθυστέρησης της γραμμής σύνδεσης μεταξύ του υπολογιστή και του auctioneer

Στην περίπτωση που ο υπολογιστής δεν διαθέτει το ζητούμενο αρχείο, η τιμή του bid που καταθέτει στον auctioneer είναι το άπειρο σύμφωνα με την παράγραφο 2.7. Στην υλοποίησή μας τα bids με θεωρητικά άπειρη τιμή παίρνουν αρνητική τιμή και έτσι παραβλέπονται από τον auctioneer κατά τον καθορισμό του νικητή κάποιου γύρου. Με αυτό τον τρόπο αποφεύγεται η ανάθεση μετάδοσης ενός αρχείου σε υπολογιστή που δεν το διαθέτει.

Κεφάλαιο 5

Αποτελέσματα πειραμάτων

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα που προέκυψαν από την δοκιμή του αλγορίθμου δημοπρασίας σε διαφορετικά σενάρια για τις δύο εφαρμογές του κεφαλαίου 4. Στην πρώτη ενότητα του κεφαλαίου παρουσιάζονται τα αποτελέσματα της εφαρμογής για την προσπέλαση πλήθους κόμβων από ομάδα robots και στη δεύτερη ενότητα τα αποτελέσματα και οι αναθέσεις της εφαρμογής για την μεταφορά αρχείων μεταξύ υπολογιστών συνδεδεμένων σε δίκτυο.

5.1 Καθορισμός διαδρομών ομάδας robots

Το κατανεμημένο software framework που υλοποιήθηκε και προσαρμόστηκε στις ανάγκες του πολυπρακτορικού συστήματος της εφαρμογής των robots δοκιμάστηκε σε 3 διαφορετικούς χάρτες και 6 διαφορετικά σενάρια. Στους 3 χάρτες βρίσκονται τοποθετημένοι σε τυχαίες θέσεις 100, 500 και 1000 κόμβοι αντίστοιχα σε κάθε χάρτη. Την προσπέλαση των κόμβων αυτών ανέλαβαν 3 robots στην μία περίπτωση και 4 robots στην άλλη. Για την προσπέλαση των κόμβων, τα robots χρησιμοποίησαν τους κανόνες του MiniSum, του MiniMax και του MiniAve πραγματοποιώντας έτσι 3 διαφορετικές αναθέσεις για κάθε

σενάριο.

Αρχικά, για να εξοικειωθεί ο αναγνώστης σχετικά με τις διαδρομές που αναπτύσσουν τα robots χρησιμοποιώντας τους κανόνες MiniSum, MiniMax και MiniAve παρατίθενται οι 3 τρόποι με τους οποίους ένα robot προσπελαύνει 15 κόμβους σε κάποιο χάρτη. Θεωρητικά αναμένουμε οι διαδρομές που προκύπτουν από τους MiniSum και MiniMax κανόνες να ταυτίζονται καθώς οι 2 κανόνες θέλουν να ελαχιστοποιήσουν το μήκος του μονοπατιού. Η διαδρομή που προκύπτει από τον MiniAve κανόνα μπορεί να διαφέρει ή να συμπίπτει με τις άλλες δύο αναλόγως τις θέσεις των κόμβων και του robot. Για παράδειγμα, εάν οι κόμβοι είναι τοποθετημένοι σε μία ευθεία μπροστά από το robot, οι 3 διαδρομές θα συμπίπτουν.

Στο σχήμα 5.1 φαίνονται οι θέσεις των 15 κόμβων και του robot στο χάρτη. Οι κόμβοι συμβολίζονται με 'x' και το robot με το πράσινο 'o'. Στα σχήματα 5.2, 5.3 και 5.4 απεικονίζονται οι διαδρομές που προκύπτουν σύμφωνα με τους MiniSum, MiniMax και MiniAve κανόνες αντίστοιχα. Όπως αναμενόταν, οι διαδρομές MiniSum και MiniMax ταυτίζονται ενώ η διαδρομή MiniAve είναι αρκετά διαφορετική.

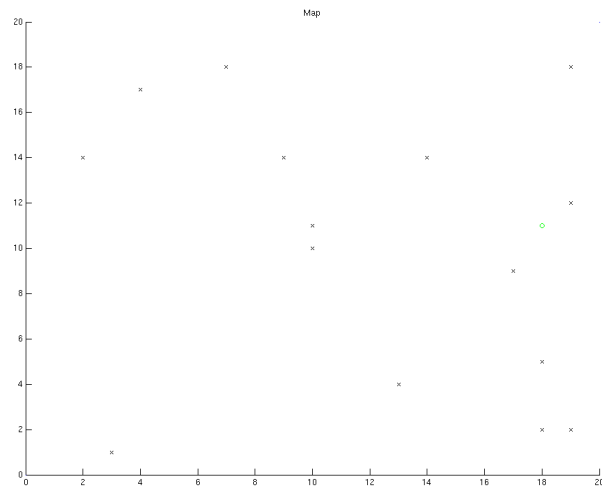
Πιθανόν η διαδρομή που αναπτύχθηκε βάσει του MiniAve κανόνα να παραξενεύει τον αναγνώστη και να του προκαλεί εντύπωση αφού φαίνεται αρκετά 'δαπανηρή' σχηματίζοντας κόλπους και κλειστές επιφάνειες. Ωστόσο, αυτή η εκδοχή προσπέλασης των κόμβων συγκριτικά με τις άλλες δύο ελαχιστοποιεί το μέσο κόστος επίσκεψης κάθε κόμβου, που είναι και το ζητούμενο του MiniAve κανόνα και επαληθεύεται από τις τιμές του πίνακα 5.1. Σύμφωνα με το MiniAve κανόνα, το robot θα προτιμήσει να επισκεφθεί πολλούς κόμβους με μικρό κόστος άμεσα παρά να επιλέξει κάποιον άλλο κόμβο που θα βελτιώσει το κόστος του μονοπατιού του καθυστερώντας την προσπέλαση των παραπάνω κόμβων. Το κόστος επίσκεψης του κόμβου αυτού προστίθεται και στους

Πίνακας 5.1: Πίνακας τιμών κόστους για τις διαδρομές των σχημάτων 5.2, 5.3 και 5.4

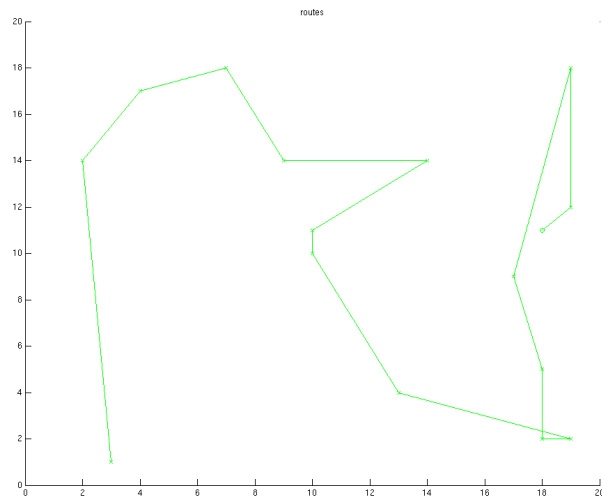
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>Path Cost</i>	73.068	73.068	82.548
<i>Visit Cost per Target</i>	35.85	35.85	29.85

επόμενους κόμβους οδηγώντας σε μεγαλύτερη μέση τιμή ανά κόμβο. Με αυτό τον τρόπο προσπέλασης είναι πιθανό κόμβοι οι οποίοι βρίσκονται σχετικά κοντά στο robot να προσεγγίζονται στο τέλος της διαδρομής. Το robot δηλαδή ‘πληρώνει’ σε κόστος μονοπατιού το κέρδος στο μέσο κόστος επίσκεψης ανά κόμβο, γεγονός το οποίο γίνεται ιδιαίτερα εμφανές σε περιπτώσεις που οι κόμβοι είναι ανομοιόμορφα διεσπαρμένοι στον χάρτη όπως και στην περίπτωση του σχήματος 5.1.

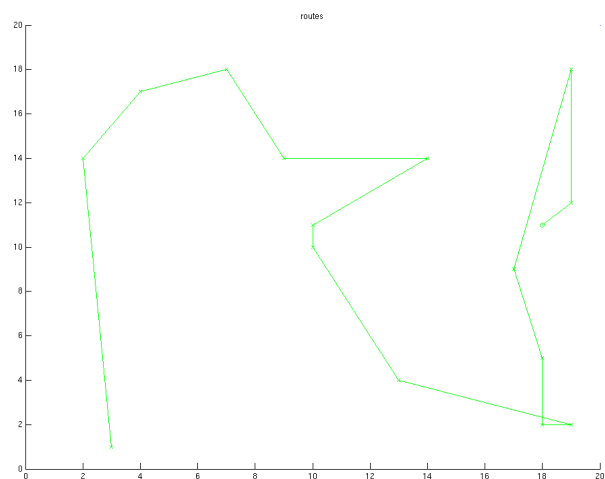
Αξίζει να σχολιαστεί και η διαδρομή των σχημάτων 5.2 και 5.3 η οποία δεν δίνει την βέλτιστη λύση σχηματίζοντας μία μικρή κλειστή επιφάνεια. Τα μονοπάτια που προκύπτουν από τους κανόνες MiniSum και MiniMax δεν θα έπρεπε να δημιουργούν κλειστές επιφάνειες όπως του MiniAve διότι έχουν σαν αποτέλεσμα διαδρομές μεγαλύτερου κόστους. Αυτό συμβαίνει λόγω μη ικανότητας πρόβλεψης ή backtracking [12] του αλγορίθμου σχηματισμού του μονοπατιού. Επομένως θα μπορούσαν να δημιουργηθούν κλειστές επιφάνειες σε διαδρομές που προκύπτουν από MiniSum και MiniMax κανόνες, κάτι το οποίο όμως συμβαίνει σπάνια. Έχοντας δει τα διαφορετικά είδη διαδρομών των τριών κανόνων είναι πιο εύκολο για τον αναγνώστη να αξιολογήσει τα αποτελέσματα και τις αναθέσεις των πειραμάτων που ακολουθούν.



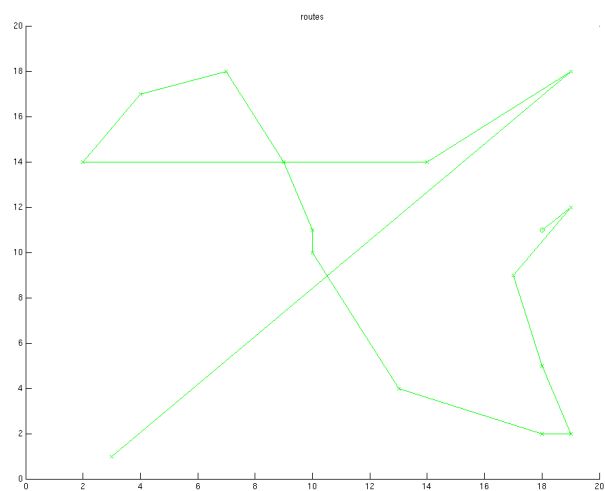
Σχήμα 5.1: Οι θέσεις των 15 κόμβων και του robot στο χάρτη διαστάσεων 20x20.



Σχήμα 5.2: Η διαδρομή που σχηματίζεται σύμφωνα με το MiniSum κανόνα



Σχήμα 5.3: Η διαδρομή που σχηματίζεται σύμφωνα με το MiniMax κανόνα



Σχήμα 5.4: Η διαδρομή που σχηματίζεται σύμφωνα με το MiniAve κανόνα

5.1.1 Ανάθεση 100 κόμβων

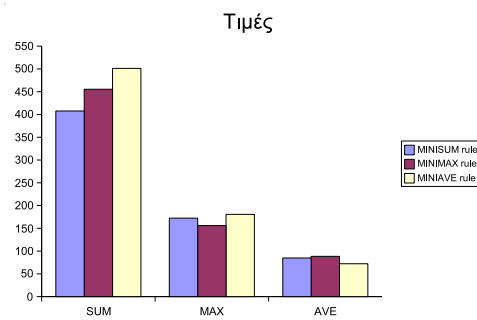
Δίνεται ένας χάρτης διαστάσεων 50x50 ο οποίος διατηρεί πληροφορία για τις θέσεις των robots και για τις θέσεις των κόμβων. Την προσπέλαση των 100 κόμβων αναλαμβάνουν 3 robots στην μία περίπτωση και 4 robots στη δεύτερη, δηλαδή έχουμε δύο διαφορετικές δημοπρασίες με $N = 3$, $M = 100$ και $N = 4$, $M = 100$ αντίστοιχα.

Πείραμα 1: Ανάθεση σε 3 πράκτορες

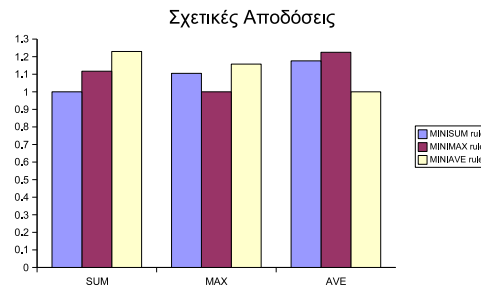
Στον πίνακα 5.2 περιέχονται οι τιμές του συνολικού αθροίσματος των επιμέρους κοστών των διαδρομών, του κόστους της μέγιστης διαδρομής και του μέσου κόστους επίσκεψης ανά κόμβου για κάθε ανάθεση. Ακολουθούν τα διαγράμματα με τις τιμές του πίνακα 5.2 στο σχήμα 5.2 και τις σχετικές αποδόσεις των τριών αναθέσεων ανά πεδίο στο σχήμα 5.6. Το διάγραμμα των σχετικών αποδόσεων προκύπτει συγκρίνοντας τις τιμές των αναθέσεων ανά πεδίο με την καλύτερη τιμή του πεδίου. Τέλος, παρατίθεται το οπτικοποιημένο κομμάτι των αποτελεσμάτων του πειράματος 1. Στο σχήμα 5.7 βρίσκεται ο χάρτης με τις θέσεις των 3 robots και των 100 κόμβων. Τα robots συμβολίζονται με 'ο' διαφορετικών χρωμάτων. Οι διαδρομές που σχηματίζουν τα robots σύμφωνα με τους 3 κανόνες του MiniSum, του MiniMax και του MiniAve φαίνονται στα σχήματα 5.8, 5.9 και 5.10 αντίστοιχα.

Πίνακας 5.2: Αποτελέσματα 1ου πειράματος

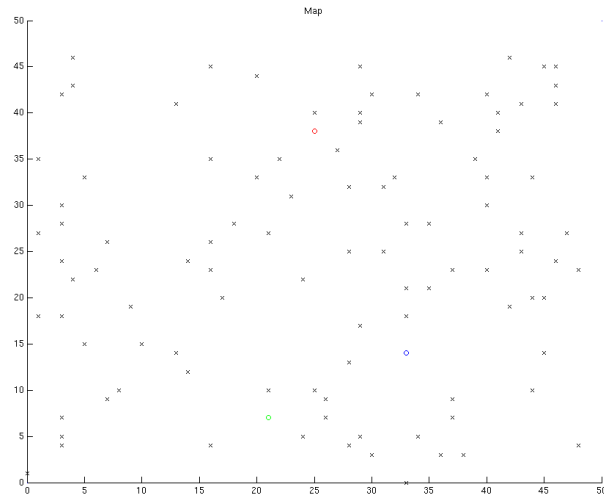
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>SUM</i>	407.64	455.4	501.22
<i>MAX</i>	172.4	156.02	180.62
<i>AVE</i>	84.91	88.43	72.19



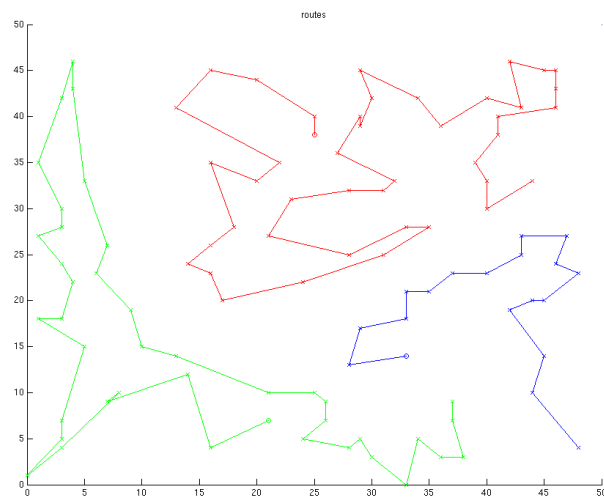
Σχήμα 5.5: Οι τιμές των τριών αναθέσεων του πειράματος 1 για τα πεδία SUM, MAX και AVE.



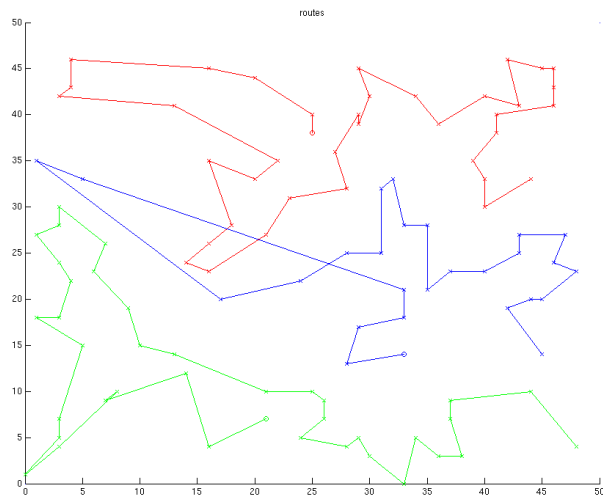
Σχήμα 5.6: Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 1.



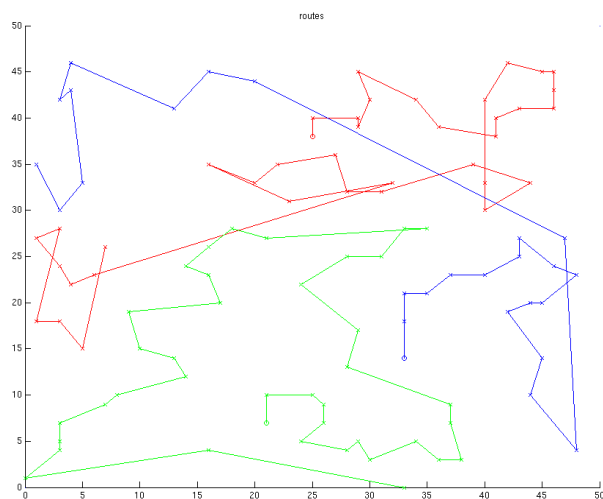
Σχήμα 5.7: Οι θέσεις των 100 κόμβων και των 3 robots στο χάρτη για το πείραμα 1.



Σχήμα 5.8: Οι διαδρομές των 3 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.7.



Σχήμα 5.9: Οι διαδρομές των 3 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.7.



Σχήμα 5.10: Οι διαδρομές των 3 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.7.

Πείραμα 2: Ανάθεση σε 4 πράκτορες

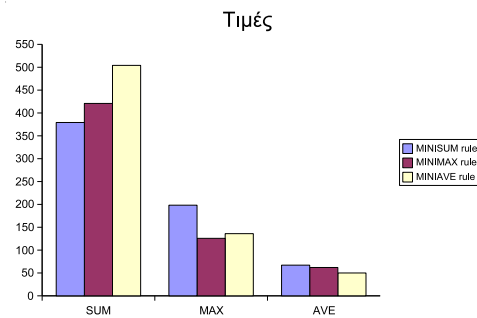
Σε αυτό το πείραμα προστίθεται ένα ακόμα robot στην ομάδα αποστολής ενώ ο χάρτης και οι θέσεις των κόμβων παραμένουν τα ίδια. Αυτό που περιμένουμε να συμβεί είναι μείωση της τιμής του μέσου κόστους επίσκεψης ανά κόμβου καθώς και μείωση της τιμής κόστους του μέγιστου μονοπατιού στις αναθέσεις MiniMax και MiniAve, με τις οποίες κατανέμονται πιο ομοιόμορφα στα robots οι αποστάσεις που χρειάζεται να διανυθούν. Για την ανάθεση MiniSum αποκλείεται να αυξηθούν οι τιμές του συνολικού κόστους διαδρομών της ομάδας καθώς και του μέσου κόστους επίσκεψης ανά κόμβο ενώ το πιο πιθανό είναι να μειωθούν κιόλας. Για τα υπόλοιπα πεδία των αναθέσεων δεν μπορούμε να κάνουμε κάποιες εκτιμήσεις.

Η παρουσίαση των αποτελεσμάτων του πειράματος 2 γίνεται παρόμοια με του πειράματος 1. Αρχικά δίνονται οι τιμές των αποτελεσμάτων, οι οποίες βρίσκονται στον πίνακα 5.2, με τα αντίστοιχα διαγράμματα που φαίνονται στα σχήματα 5.5 και 5.6. Στη συνέχεια, στα σχήματα 5.13, 5.14, 5.15 και 5.16 απεικονίζονται ο χάρτης καθώς και οι διαδρομές που προκύπτουν σύμφωνα με τους κανόνες MiniSum, MiniMax και MiniAve.

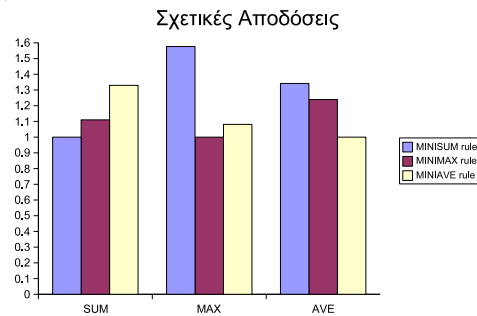
Οι τιμές των αποτελεσμάτων στον πίνακα 5.2 επαληθεύουν τις εκτιμήσεις μας. Όντως οι τιμές κόστους μονοπατιού και μέσου κόστους επίσκεψης ελαττώνονται στις αναθέσεις MiniMax και MiniAve. Το ίδιο συμβαίνει στην τιμή του μέσου κόστους και για την ανάθεση MiniSum, κάτι το οποίο θα μπορούσε και να μην συμβεί στην ακραία περίπτωση όπου η ανάθεση κόμβων στο επιπρόσθετο robot αυξάνει την τιμή του συνολικού κόστους ανάθεσης. Επίσης ελαττώνεται και η τιμή του συνολικού αθροίσματος των επιμέρους κοστών των μονοπατιών ενώ αυξάνεται η τιμή του μέγιστου μονοπατιού της ανάθεσης MiniSum, το οποίο σημαίνει ότι το μεγαλύτερο μονοπάτι σχηματίζεται από τον επιπρόσθετο κόμβο.

Πίνακας 5.3: Αποτελέσματα 2ου πειράματος

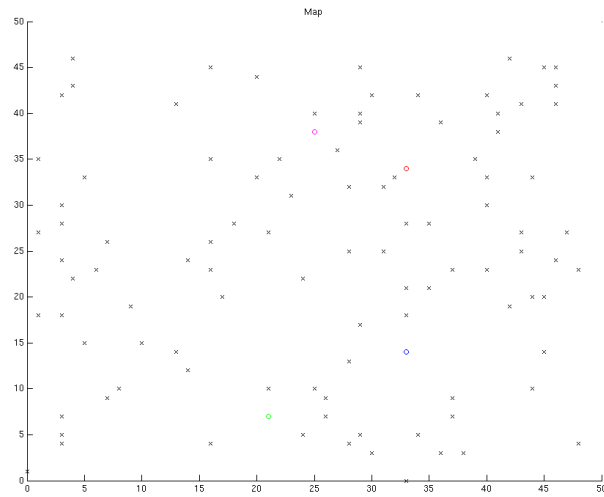
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>SUM</i>	379.19	420.87	504.11
<i>MAX</i>	198.3	125.78	136.02
<i>AVE</i>	67.12	62.01	50.03



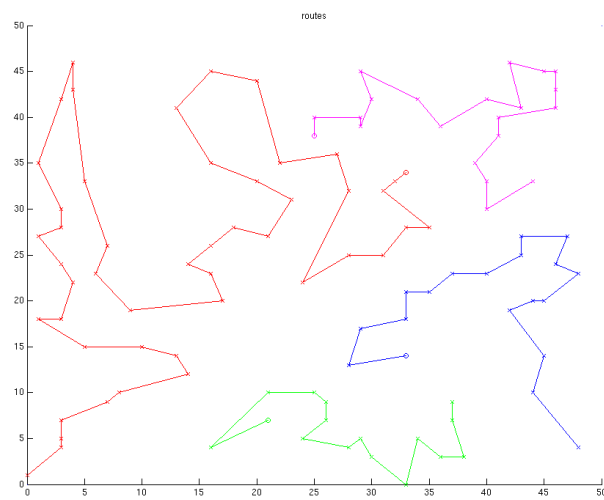
Σχήμα 5.11: Οι τιμές των τριών αναθέσεων του πειράματος 2 για τα πεδία SUM, MAX και AVE.



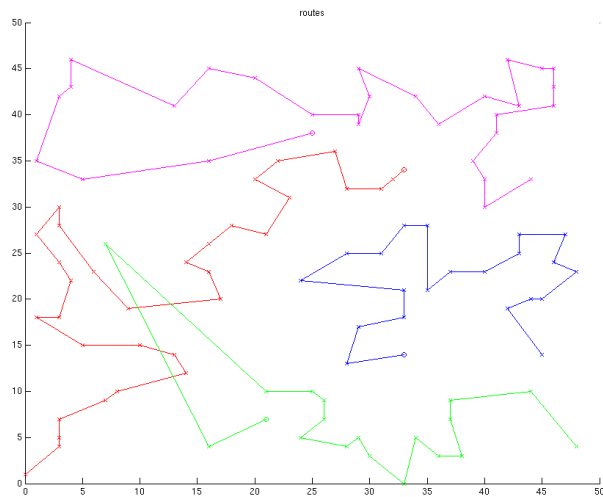
Σχήμα 5.12: Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 2.



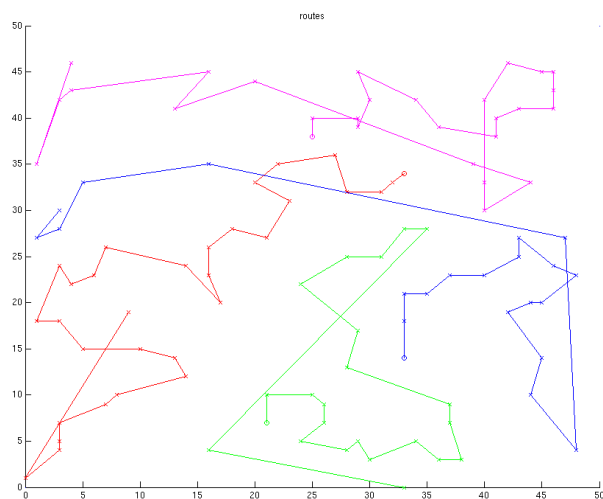
Σχήμα 5.13: Οι θέσεις των 100 κόμβων και των 4 robots στο χάρτη για το πείραμα 2.



Σχήμα 5.14: Οι διαδρομές των 4 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.13.



Σχήμα 5.15: Οι διαδρομές των 4 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.13.



Σχήμα 5.16: Οι διαδρομές των 4 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.13.

5.1.2 Ανάθεση 500 κόμβων

Σε αυτή την περίπτωση υπάρχει ένας χάρτης διαστάσεων 300x300 ο οποίος διατηρεί πληροφορία για τις θέσεις των robots και για τις θέσεις των 500 κόμβων. Πρώτα παρουσιάζονται οι αναθέσεις των κόμβων για τα 3 robots και στη συνέχεια για τα 4 robots.

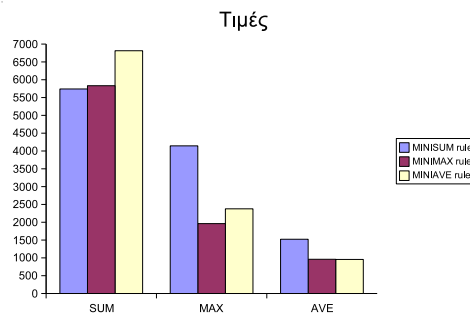
Πείραμα 3: Ανάθεση σε 3 πράκτορες

Εξαιτίας του νέου χάρτη με τους 500 κόμβους οι τιμές των πινάκων 5.4 και 5.5 αναμένεται να αυξηθούν σε σχέση με τους 5.2 και 5.3 αφού αυξάνεται ο αριθμός των κόμβων και οι αποστάσεις μεγαλώνουν λόγω του μεγαλύτερου μεγέθους του χάρτη.

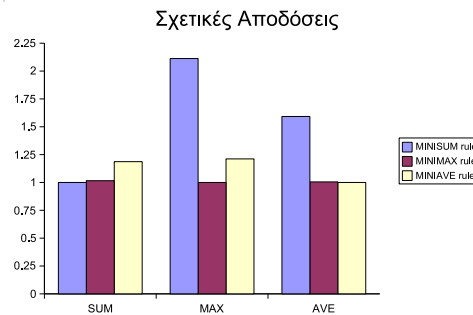
Οι αναθέσεις MiniSum, MiniMax και MiniAve πραγματοποιούν το καλύτερο αποτέλεσμα στα αντίστοιχα πεδία όπως συνηθίζεται. Παρόλα αυτά στα διαγράμματα των σχημάτων 5.17 και 5.18 παρατηρείται μία πολύ χαμηλή απόδοτικότητα της ανάθεσης MiniSum σε σύγκριση με τις άλλες δύο στα πεδία MAX και AVE χωρίς οι αναθέσεις MiniMax και MiniAve να παρουσιάζουν το ίδιο χαμηλή απόδοση στο πεδίο SUM. Επίσης αξίζει να σημειωθεί η πολύ υψηλή απόδοση που εμφανίζει η MiniMax ανάθεση και στα άλλα δύο πεδία και κυρίως στο AVE για αυτό τον χάρτη και την ανάληψη των 500 κόμβων από 3 robot. Ο χάρτης και οι διαδρομές των robots για τις αναθέσεις MiniSum, MiniMax και MiniAve απεικονίζονται στα σχήματα 5.19, 5.20, 5.21 και 5.22 αντίστοιχα.

Πίνακας 5.4: Αποτελέσματα 3ου πειράματος

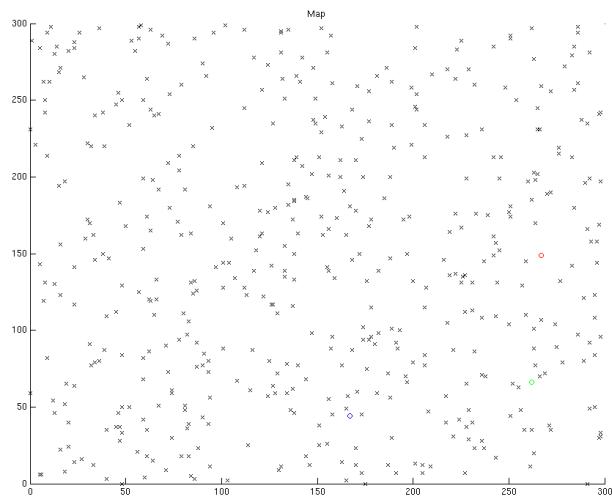
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>SUM</i>	5740.2	5832.02	6813.12
<i>MAX</i>	4144.01	1961.43	2376.83
<i>AVE</i>	1522.98	961.1	956.26



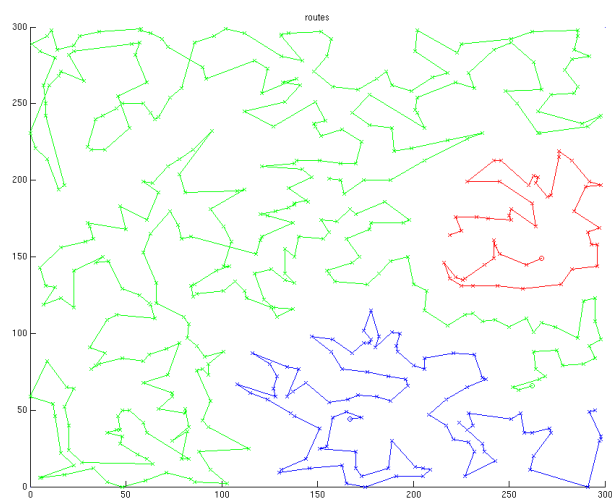
Σχήμα 5.17: Οι τιμές των τριών αναθέσεων του πειράματος 3 για τα πεδία SUM, MAX και AVE.



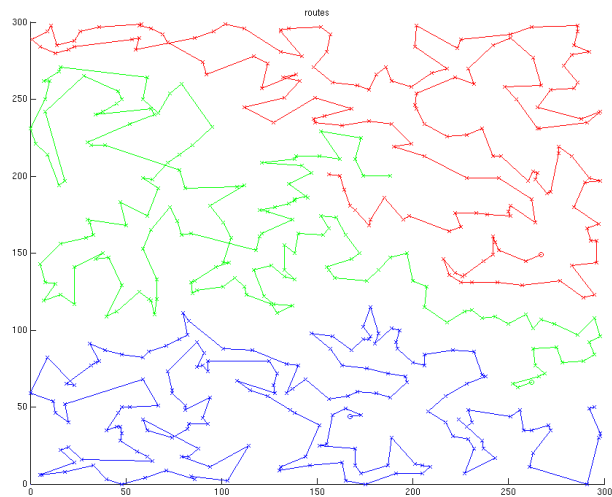
Σχήμα 5.18: Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 3.



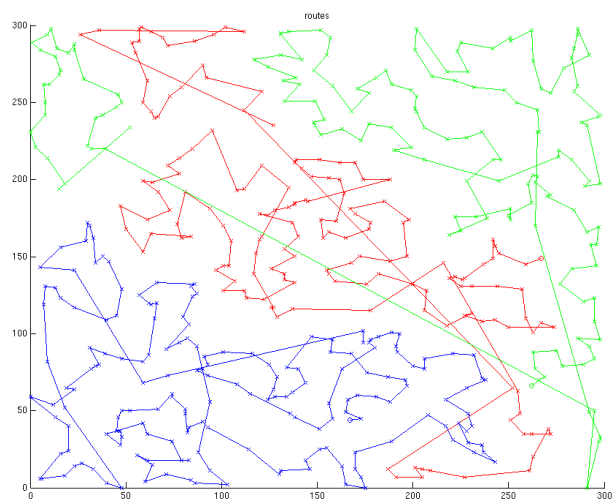
Σχήμα 5.19: Οι θέσεις των 500 κόμβων και των 3 robots στο χάρτη για το πείραμα 3.



Σχήμα 5.20: Οι διαδρομές των 3 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.19.



Σχήμα 5.21: Οι διαδρομές των 3 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.19.



Σχήμα 5.22: Οι διαδρομές των 3 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.19.

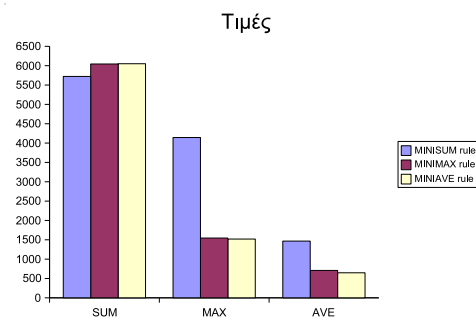
Πείραμα 4: Ανάθεση σε 4 πράκτορες

Όπως και στο πείραμα 2, αναμένεται να μειωθούν οι τιμές των πεδίων MAX και AVE για τις αναθέσεις MiniMax και MiniAve σε σύγκριση με τις τιμές του πειράματος 3 εξαιτίας της προσθήκης ενός ακόμα robot στην ομάδα αποστολής. Η μείωση αυτή όντως παρατηρείται στα δεδομένα του πίνακα 5.5. Επίσης βλέπουμε ότι για την ανάθεση MiniSum η τιμή του πεδίου MAX δεν αλλάζει με την προσθήκη ενός επιπλέον robot. Συμπεραίνουμε λοιπόν ότι το robot με το μονοπάτι μέγιστου κόστους διατηρεί την ίδια ανάθεση και έτσι το νέο robot ανέλαβε κόμβους από τις διαδρομές των άλλων 2 robots διατηρώντας έτσι την τιμή του μέσου κόστους επίσκεψης ανά κόμβο σε υψηλά επίπεδα.

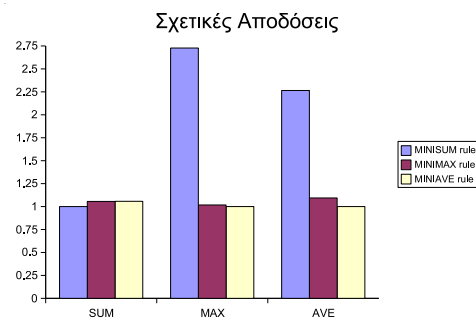
Στον πίνακα 5.5 καθώς και στα διαγράμματα των σχημάτων 5.23 και 5.24 παρατηρούμε ένα αξιοσημείωτο γεγονός όπου η ανάθεση MiniAve παρουσιάζει το καλύτερο αποτέλεσμα και την καλύτερη απόδοση στο πεδίο MAX. Αυτό συμβαίνει διότι ο αλγόριθμος σχηματισμού των μονοπατιών δεν δίνει την βέλτιστη λύση αλλά η λύση που δίνει κυμαίνεται μέσα σε κάποια περιοχή ποσοστού της βέλτιστης. Στην δημοσίευση [1], στην οποία βασίζεται ο αλγόριθμος ανάπτυξης των μονοπατιών των robots, υπάρχει ένας πίνακας με τα όρια της περιοχής στην οποία μπορεί να βρίσκεται η λύση που δίνει ο αλγόριθμος. Σύμφωνα με αυτό τον πίνακα είναι πιθανό ένα είδος ανάθεσης να δώσει καλύτερη λύση από κάποιο άλλο στο πεδίο που η δεύτερη επιδιώκει να ελαχιστοποιήσει.

Πίνακας 5.5: Αποτελέσματα 4ου πειράματος

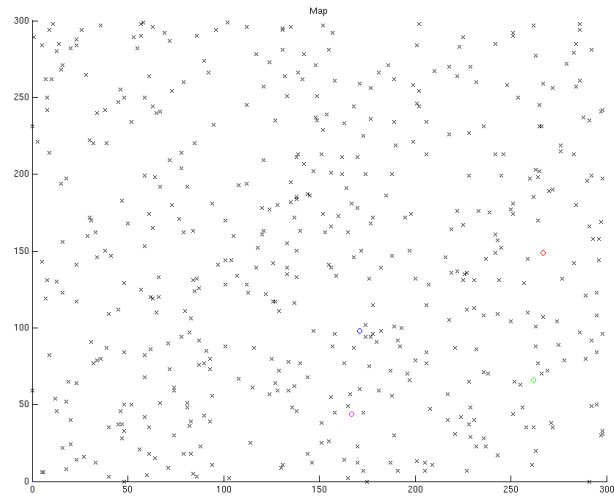
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>SUM</i>	5722.83	6044.01	6050.99
<i>MAX</i>	4144.01	1545.67	1519.39
<i>AVE</i>	1466.42	708.35	647.44



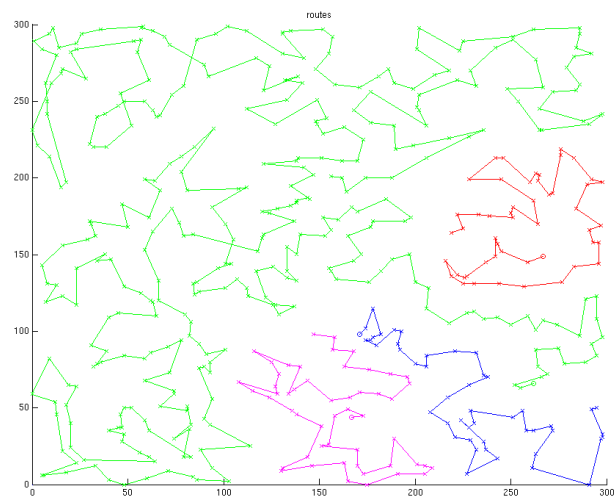
Σχήμα 5.23: Οι τιμές των τριών αναθέσεων του πειράματος 4 για τα πεδία SUM, MAX και AVE.



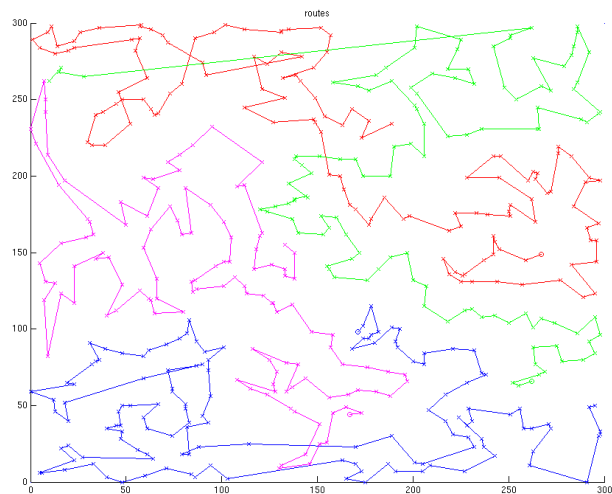
Σχήμα 5.24: Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 4.



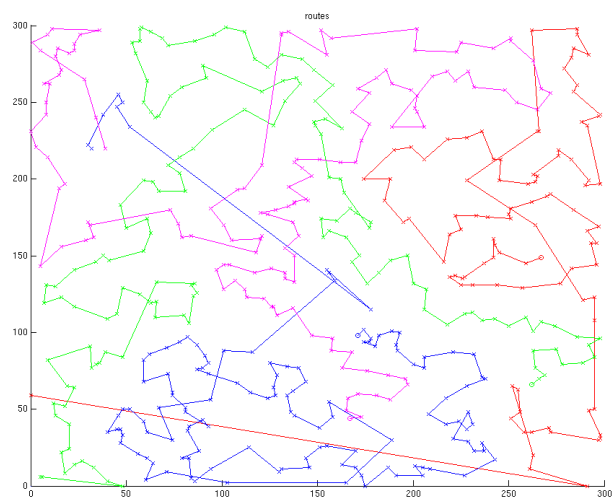
Σχήμα 5.25: Οι θέσεις των 500 κόμβων και των 4 robots στο χάρτη για το πείραμα 4.



Σχήμα 5.26: Οι διαδρομές των 4 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.25.



Σχήμα 5.27: Οι διαδρομές των 4 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.25.



Σχήμα 5.28: Οι διαδρομές των 4 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.25.

5.1.3 Ανάθεση 1000 κόμβων

Ο τελευταίος χάρτης είναι διαστάσεων 500x500 και διατηρεί πληροφορία για τις θέσεις των robots και για τις θέσεις των 1000 κόμβων. Στο πρώτο μέρος του σεναρίου, το οποίο αποτελεί το πείραμα 5, την προσπέλαση των 1000 κόμβων αναλαμβάνουν τα 3 robots και στο δεύτερο μέρος η ανάθεση των κόμβων γίνεται σε 4 robots.

Πείραμα 5: Ανάθεση σε 3 πράκτορες

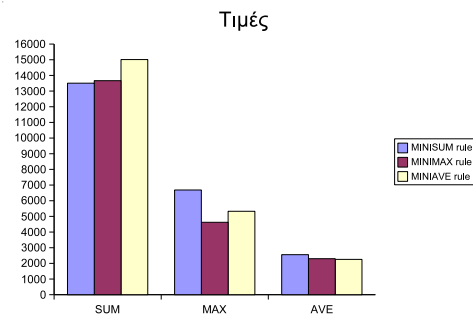
Σε αυτό το πείραμα οι τιμές των αποτελεσμάτων του πίνακα 5.6 είναι αρκετά μεγαλύτερες από τις αντίστοιχες του πειράματος 3 στον πίνακα 5.4 αφού το μέγεθος του χάρτη αυξήθηκε και το πλήθος των κόμβων διπλασιάστηκε ενώ ο αριθμός των robots παρέμεινε ο ίδιος.

Οι τιμές των αποτελεσμάτων στον πίνακα 5.6 και στο διάγραμμα του σχήματος 5.29 θα μπορούσαν να χαρακτηριστούν σαν φυσιολογικές αφού το κάθε είδος ανάθεσης δίνει την καλύτερη λύση στο πεδίο του. Στο διάγραμμα του σχήματος 5.29 παρατηρούμε ότι η MiniMax ανάθεση έχει πάρα πολύ καλή απόδοση σε όλα τα πεδία όπως και στο πείραμα 3.

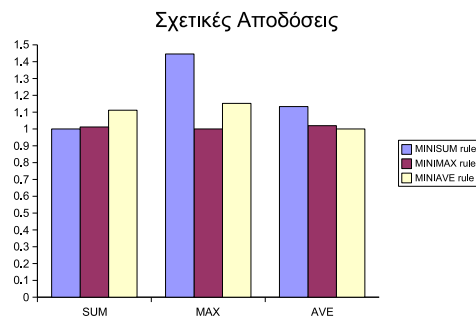
Ο χάρτης και οι διαδρομές των robots για τις αναθέσεις MiniSum, MiniMax και MiniAve απεικονίζονται στα σχήματα 5.31, 5.32, 5.33 και 5.34 αντίστοιχα. Στο σχήμα 5.32 όπου απεικονίζεται η ανάθεση MiniMax το robot που παριστάνεται με το πράσινο κυκλάκι φαίνεται να κάνει πολύ μεγάλη διαδρομή για να επισκεφθεί κάποιους κόμβους, οι οποίοι φαίνεται να είναι στην περιοχή του μπλε και του κόκκινου robot.

Πίνακας 5.6: Αποτελέσματα 5ου πειράματος

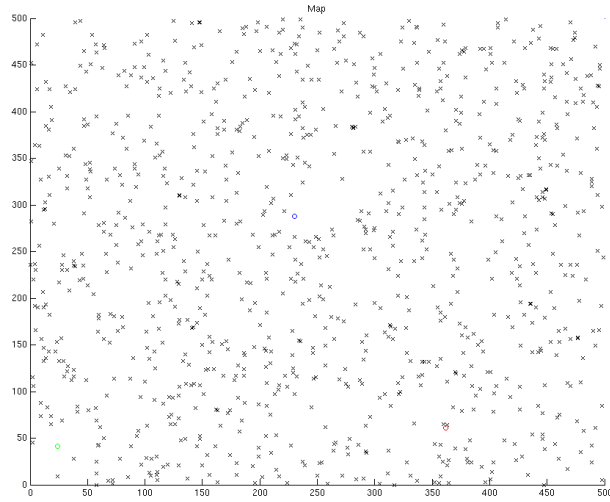
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>SUM</i>	13504.77	13664.29	15011.79
<i>MAX</i>	6684.69	4623.54	5326.54
<i>AVE</i>	2555.27	2299.15	2254.87



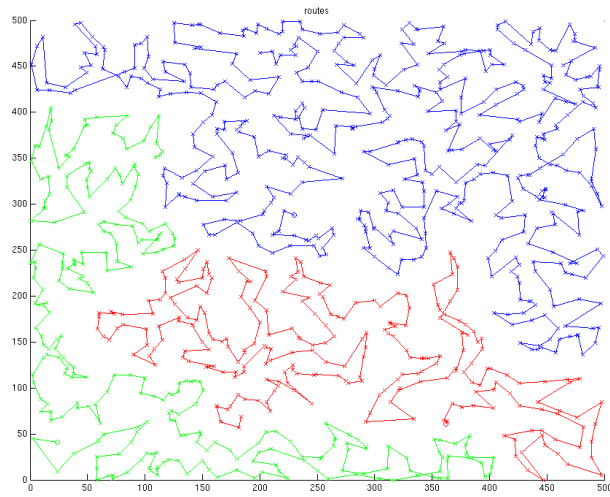
Σχήμα 5.29: Οι τιμές των τριών αναθέσεων του πειράματος 5 για τα πεδία SUM, MAX και AVE.



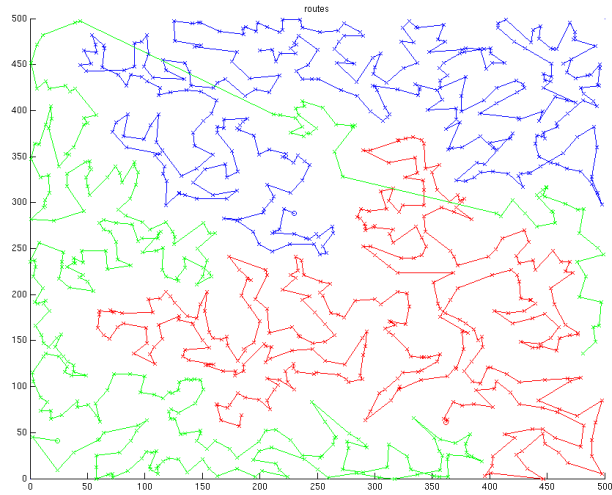
Σχήμα 5.30: Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 5.



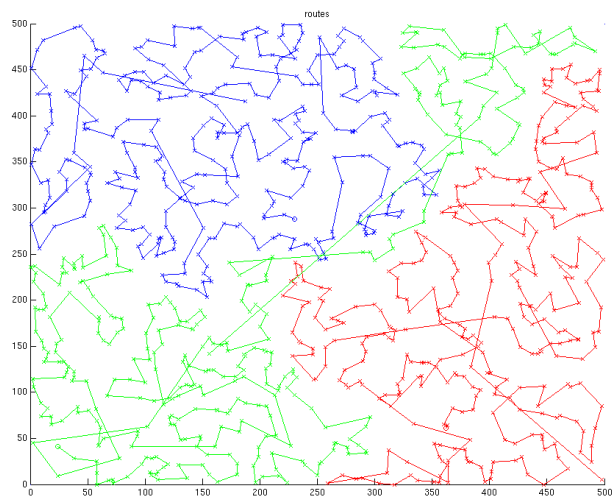
Σχήμα 5.31: Οι θέσεις των 1000 κόμβων και των 3 robots στο χάρτη για το πείραμα 5.



Σχήμα 5.32: Οι διαδρομές των 3 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.31.



Σχήμα 5.33: Οι διαδρομές των 3 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.31.



Σχήμα 5.34: Οι διαδρομές των 3 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.31.

Η παράδοξη αυτή διαδρομή του πράσινου robot συμβαίνει γιατί το team objective είναι η ελαχιστοποίηση του μέγιστου μονοπατιού, που έχει ως αποτέλεσμα την ισοκατανομή των αποστάσεων που πρέπει να διανυθούν, χωρίς όμως τα robots να βρίσκονται σε αρκετά μακρινές αποστάσεις ώστε να έχουν σαφείς περιοχές δράσης. Στην αρχή ανάπτυξης των μονοπατιών τους, το μπλε και το κόκκινο robot αναλαμβάνουν κόμβους σχετικά κοντά στη διαδρομή του πράσινου, τους οποίους κόμβους θα μπορούσε να είχε αναλάβει αυτό αποφεύγοντας την μεγάλη διαδρομή από τη μία άκρη του χάρτη στην άλλη. Όταν προς το τέλος της δημοπρασίας απομένουν οι απομακρυσμένοι από το πράσινο robot κόμβοι, το μπλε και το κόκκινο έχουν ήδη διαμορφώσει μεγαλύτερα μονοπάτια από το πράσινο κι έτσι το πράσινο αναλαμβάνει την προσπέλασή τους.

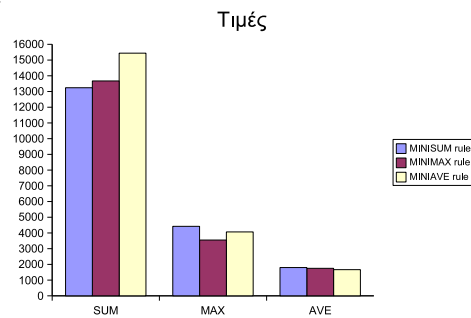
Πείραμα 6: Ανάθεση σε 4 πράκτορες

Παρατηρώντας τον πίνακα 5.7 και το σχήμα 5.35 βλέπουμε ότι με την προσθήκη ενός ακόμα robot στην ομάδα αποστολής, οι τιμές των πεδίων Max και Ave ελαττώνονται για τις αναθέσεις MiniMax και MiniAve όπως και στα προηγούμενα πειράματα. Επίσης οι τιμές των πεδίων αυτών ελαττώνονται αισθητά και για την ανάθεση MiniSum. Μπορούμε να συμπεράνουμε ότι το νέο robot αναλαμβάνει μέρος της μεγαλύτερης διαδρομής του πειράματος 5 για την MiniSum ανάθεση. Η υπόθεσή μας αυτή επιβεβαιώνεται στο σχήμα 5.38, στο οποίο απεικονίζεται η MiniSum ανάθεση.

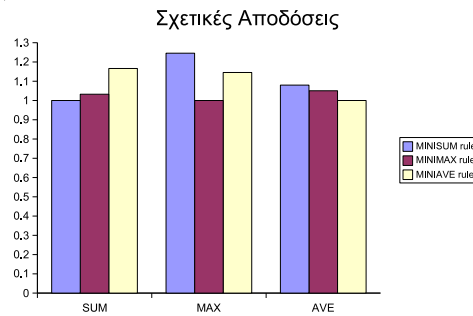
Οι σχετικές αποδόσεις των τριών αναθέσεων βρίσκονται στο διάγραμμα του σχήματος 5.36. Ο χάρτης του πειράματος βρίσκεται στο σχήμα 5.25 και ακολουθούν τα σχήματα 5.39 και 5.40 όπου απεικονίζονται οι αναθέσεις MiniMax και MiniAve αντίστοιχα.

Πίνακας 5.7: Αποτελέσματα του πειράματος

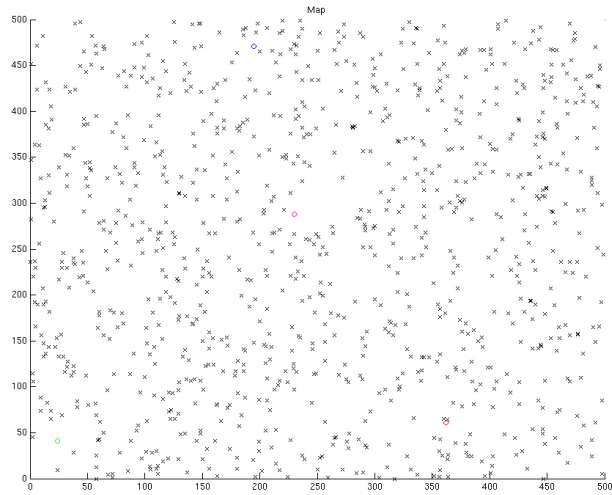
	<i>MiniSum</i>	<i>MiniMax</i>	<i>MiniAve</i>
<i>SUM</i>	13238.97	13670.99	15439.78
<i>MAX</i>	4422.2	3549.81	4066.56
<i>AVE</i>	1800.04	1751.2	1666.79



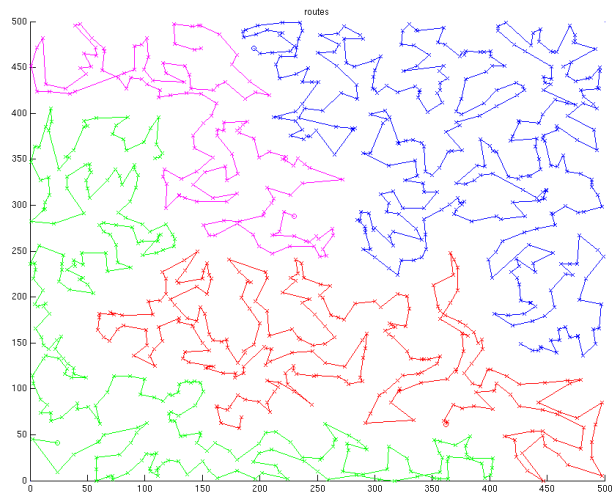
Σχήμα 5.35: Οι τιμές των τριών αναθέσεων του πειράματος 6 για τα πεδία SUM, MAX και AVE.



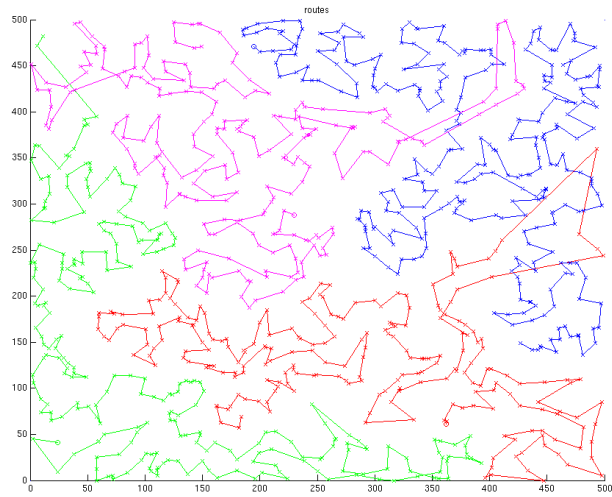
Σχήμα 5.36: Οι σχετικές αποδόσεις των τριών αναθέσεων για το πείραμα 6.



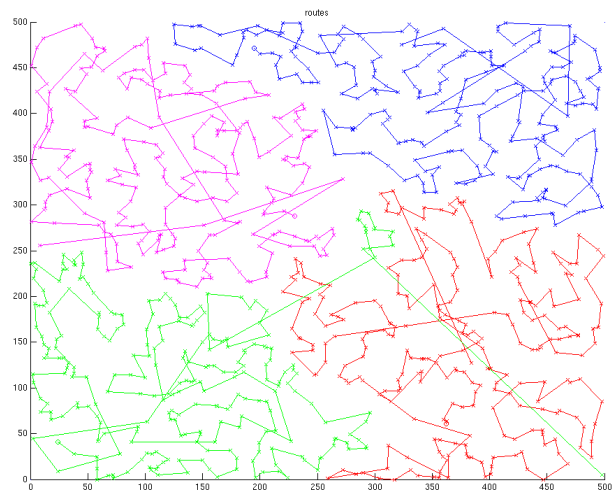
Σχήμα 5.37: Οι θέσεις των 1000 κόμβων και των 4 robots στο χάρτη για το πείραμα 6.



Σχήμα 5.38: Οι διαδρομές των 4 robots σύμφωνα με το MiniSum κανόνα για το χάρτη του σχήματος 5.37.



Σχήμα 5.39: Οι διαδρομές των 4 robots σύμφωνα με το MiniMax κανόνα για το χάρτη του σχήματος 5.37.



Σχήμα 5.40: Οι διαδρομές των 4 robots σύμφωνα με το MiniAve κανόνα για το χάρτη του σχήματος 5.37.

Παρατηρούμε ότι στις αναθέσεις MiniMax και MiniAve λόγω της ισοκατανομής του φόρτου, δεν είναι σπάνιο το φαινόμενο των αλληλοεπικαλύψεων των διαδρομών των robots. Αντίθετα, στη MiniSum ανάθεση δεν συμβαίνει αυτό το φαινόμενο αφού κάποιο robot που θα επισκεφθεί ένα ή περισσότερους κόμβους στην περιοχή άλλου robot θα αυξήσει περισσότερο το συνολικό κόστος σε σχέση με το robot της περιοχής.

Συνοψίζοντας, ο στόχος των πειραμάτων της ενότητας 5.1 ήταν το κατανεμημένο software framework της υλοποίησής μας να έδινε τις λύσεις και τα αποτελέσματα που προβλέπονται από τον αλγόριθμο δημοπρασίας της δημοσίευσης [1]. Τα αποτελέσματα των πειραμάτων πράγματι συμφωνούν με τα προβλεπόμενα του αλγορίθμου αποδεικνύοντας έτσι την ισχύ των πλεονεκτημάτων του αλγορίθμου περί ικανοποιητικών λύσεων για μεγάλο αριθμό κόμβων σε πολυπρακτορικά συστήματα, όπου οι αλγόριθμοι βέλτιστων λύσεων υστερούν λόγω μεγάλης υπολογιστικής πολυπλοκότητας.

5.2 Computer File Sharing

Στην περίπτωση της μεταφοράς αρχείων μεταξύ υπολογιστών σε δίκτυο πραγματοποιήθηκαν τρία πειράματα. Τα σενάρια των πειραμάτων καλύπτουν το μεγαλύτερο εύρος των πιθανών συνθηκών που μπορεί να συναντήσει ένας υπολογιστής κατά την πραγματοποίηση δημοπρασίας για ανάληψη αρχείων. Σε όλα τα πειράματα, ο υπολογιστής που πραγματοποιεί μία δημοπρασία στέλνει μία λίστα με τα ονόματα των αρχείων που χρειάζεται. Η παρουσίαση των πειραμάτων αποτελείται από δύο μέρη, πριν την έναρξη της δημοπρασίας και μετά την ολοκλήρωσή της. Στο πρώτο μέρος δίνεται η λίστα με τα αιτούμενα για μεταφορά αρχεία καθώς και τα περιεχόμενα αρχεία στα αποθηκευτικά μέσα των υπολογιστών που συμμετέχουν στη δημοπρασία. Επίσης για κάθε bidder δίνεται και η τιμή του latency της γραμμής μεταξύ αυτού και του auctioneer. Στο δεύτερο

μέρος βρίσκεται η ανάθεση των αρχείων με τους χρόνους μεταφοράς τους.

Πριν την παρουσίαση των αποτελεσμάτων να υπενθυμίσουμε στον αναγνώστη ότι σε μία δημοπρασία δίνεται προτεραιότητα για μεταφορά στα μικρού μεγέθους αρχεία διότι έτσι ελαχιστοποιείται ο μέσος χρόνος μετάδοσης αρχείου. Μεταξύ δύο ή περισσότερων δημοπρασιών δεν ισχύει αυτή η προτεραιότητα, δηλαδή δεν πρόκειται να προσπεραστεί ένα μεγάλο αρχείο που έχει προγραμματιστεί για μεταφορά από μικρότερα αρχεία επόμενης δημοπρασίας.

5.2.1 Πείραμα A

Στο σενάριο του πειράματος A το δίκτυο αποτελείται από τους υπολογιστές PC_1 , PC_2 , PC_3 και PC_4 . Ο PC_4 συνδέεται στους άλλους τρεις και τους στέλνει μία λίστα ζητώντας να του μεταφερθούν τα αρχεία που περιέχονται σε αυτή. Στον πίνακα 5.8 βρίσκονται τα στοιχεία του πειράματος πριν την έναρξη της δημοπρασίας και στον πίνακα 5.9 οι τιμές του latency μεταξύ των bidders και του PC_4 . Οι υπολογιστές στους οποίους ανατέθηκαν τα αρχεία καθώς και οι χρόνοι μεταφοράς τους βρίσκονται στον πίνακα 5.10. Σε αυτό το παράδειγμα εξετάζεται η περίπτωση στην οποία οι υπολογιστές δεν έχουν κάποια αρχεία προς μετάδοση που να επηρεάζουν τον υπολογισμό των bids τους.

Πίνακας 5.8: Τα στοιχεία της δημοπρασίας του πειράματος A

<i>Auctioneer</i>	<i>Λίστα Αρχείων</i>	<i>Μέγεθος Αρχείου</i>	PC_1	PC_2	PC_3
PC_4	file1	480 KBytes	file1	file3	file1
	file2	40 MBytes	file6	file4	file2
	file3	2 MBytes	file8	file6	file9
	file4	4 Mbytes	file7	file8	file7
	file5	29 MBytes	file5	file7	
	file6	4 MBytes		file5	
	file7	2 MBytes			
	file8	52 MBytes			
	file9	23 MBytes			

Πίνακας 5.9: Οι τιμές latency στο πείραμα A

<i>latency</i>	PC_1	PC_2	PC_3
PC_4	0.106	0.118	0.105

Πίνακας 5.10: Ανάθεση αρχείων και χρόνοι μεταφοράς τους στο πείραμα A

	<i>Assigned files</i>	<i>Transmission time</i>
PC_1	file7	00 : 02 : 484 sec
	file6	00 : 03 : 825 sec
	file5	00 : 11 : 214 sec
PC_2	file3	00 : 02 : 230 sec
	file4	00 : 03 : 421 sec
	file8	00 : 15 : 780 sec
PC_3	file1	00 : 02 : 537 sec
	file9	00 : 09 : 138 sec
	file2	00 : 16 : 965 sec

Οι τιμές του latency είναι παραπλήσιες για τους τρεις υπολογιστές οπότε δεν παρουσιάζεται το φαινόμενο υπερφόρτωσης κάποιου bidder λόγω ταχύτερης γραμμής σύνδεσης με τον auctioneer. Στον πίνακα 5.10 παρατηρείται μία ομοιόμορφη κατανομή των αρχείων στους bidders που οφείλεται κυρίως στις παραπλήσιες τιμές latency. Ο PC_3 αναλαμβάνει το αρχείο file1 και δύο μεγάλου μεγέθους αρχεία, τα file9 και file2, τα οποία μόνο αυτός διαθέτει και μπορεί να μεταδώσει. Οι PC_1 και PC_2 περιέχουν αρκετά κοινά αρχεία με αποτέλεσμα να ανταγωνίζονται για την ανάληψή τους κατά την διάρκεια της δημοπρασίας. Τελικά, οι PC_1 και PC_2 αναλαμβάνουν δύο αρχεία μικρού μεγέθους και ένα αρχείο μεγάλου μεγέθους ο καθένας.

5.2.2 Πείραμα Β

Στο πείραμα αυτό το δίκτυο αποτελείται από 6 υπολογιστές, τους PC_1 , PC_2 , PC_3 , PC_4 , PC_5 και PC_6 . Οι PC_5 και PC_6 χρειάζονται κάποια αρχεία και επομένως λειτουργούν ως auctioneers ενώ οι άλλοι τέσσερις είναι οι bidders των δημοπρασιών αυτών. Οι PC_5 και PC_6 δεν περιέχουν κάποια αρχεία και άρα ουσιαστικά δεν συμμετέχει ο ένας στη δημοπρασία του άλλου σαν bidder. Οι δύο αυτοί υπολογιστές πραγματοποιούν μία δημοπρασία έκαστος για να αποκτήσουν τα αρχεία που χρειάζονται. Οι αιτήσεις τους λαμβάνουν χώρα με μικρή χρονική διαφορά με τον PC_5 να προηγείται. Αφού ολοκληρωθεί η δημοπρασία του PC_5 , ξεκινάει η πραγματοποίηση της δημοπρασίας του PC_6 . Η ανάθεση των αρχείων από τη δημοπρασία του PC_5 θα επηρεάσει την ανάθεση των αρχείων του PC_6 αφού οι υπολογιστές έχουν να συνεκτιμήσουν τον χρόνο μετάδοσης των αρχείων του PC_5 στον υπολογισμό των bids. Στους πίνακες 5.11 και 5.12 βρίσκονται τα στοιχεία του πειράματος και οι τιμές latency πριν την έναρξη της δημοπρασίας και στον πίνακα 5.13 περιέχονται οι αναθέσεις των αρχείων με τους χρόνους μετάδοσης.

Πίνακας 5.11: Τα στοιχεία της δημοπρασίας του πειράματος B

<i>Auctioneer</i>	<i>Λίστα Αρχείων</i>	<i>Μέγεθος Αρχείου</i>	<i>PC₁</i>	<i>PC₂</i>	<i>PC₃</i>	<i>PC₄</i>
<i>PC₅</i>	file10	2 Mbytes	file1	file3	file1	file5
	file11	7 MBytes	file5	file4	file2	file8
	file12	4 Mbytes	file6	file5	file9	file12
	file13	17 Mbytes	file7	file6	file13	file13
	file14	18 MBytes	file8	file8	file14	file15
	file15	1 MBytes	file10	file12	file15	file16
	file16	91 MBytes	file11			
<i>PC₆</i>	file1	480 KBytes	file16			
	file2	40 MBytes				
	file3	2 MBytes				
	file4	4 Mbytes				
	file5	29 MBytes				
	file6	4 MBytes				
	file7	2 MBytes				
	file8	52 MBytes				
	file9	23 MBytes				

Πίνακας 5.12: Οι τιμές latency στο πείραμα B

<i>latency</i>	<i>PC₁</i>	<i>PC₂</i>	<i>PC₃</i>	<i>PC₄</i>
<i>PC₅</i>	1.334	0.123	0.872	0.113
<i>PC₆</i>	2.274	0.115	2.538	0.128

Σε αυτό το πείραμα την τελική ανάθεση επηρεάζουν δύο παράγοντες, η μεγάλη διαφορά των τιμών του latency ανάμεσα στους 4 bidders καθώς και τα αρχεία που αναλαμβάνουν οι υπολογιστές στην πρώτη δημοπρασία. Στην πρώτη δημοπρασία, μόνο για τα αρχεία file13 και file15 υπάρχουν περισσότεροι του ενός υποψήφιοι υπολογιστές για να τα μεταδώσουν. Τα αρχεία αυτά τα αναλαμβάνει ο *PC₄* λόγω χαμηλότερου latency από τον *PC₃*. Στη δεύτερη δημοπρασία ο *PC₄* έχει αρκετά μεγάλο όγκο αρχείων προς μετάδοση και για αυτό το λόγο δεν αναλαμβάνει την μεταφορά κάποιου αρχείου. Ο *PC₂* με μικρή

Πίνακας 5.13: Ανάθεση αρχείων και χρόνοι μεταφοράς τους στο πείραμα Β

	<i>Assigned files</i>	<i>Transmission time</i>
PC_1	file10	00 : 02 : 228 sec
	file11	00 : 03 : 453 sec
	file7	00 : 04 : 631 sec
PC_2	file12	00 : 03 : 217 sec
	file3	00 : 05 : 117 sec
	file4	00 : 07 : 975 sec
	file6	00 : 10 : 741 sec
	file5	00 : 15 : 888 sec
	file8	00 : 23 : 913 sec
PC_3	file14	00 : 03 : 670 sec
	file1	00 : 05 : 516 sec
	file9	00 : 10 : 590 sec
	file2	00 : 17 : 077 sec
PC_4	file15	00 : 02 : 532 sec
	file13	00 : 06 : 227 sec
	file16	00 : 14 : 953 sec

τιμή latency και μικρό όγκο αρχείων προς μετάδοση αναλαμβάνει όλα τα αρχεία που περιέχει στο δίσκο του. Οι PC_1 και PC_3 έχουν πολύ μεγάλη τιμή latency και επομένως αναλαμβάνουν τα αρχεία που διατηρούν μόνο αυτοί.

5.2.3 Πείραμα Γ

Στο τελευταίο πείραμα το δίκτυο αποτελείται από τους υπολογιστές PC_1 , PC_2 , PC_3 , PC_4 , PC_5 και PC_6 . Σε αυτό το πείραμα πραγματοποιούνται 3 δημοπρασίες συνολικά. Αρχικά ο PC_1 πραγματοποιεί μία δημοπρασία στην οποία ζητάει κάποια αρχεία. Αφού ολοκληρωθεί η μεταφορά των αρχείων αυτών στον PC_1 , λαμβάνουν χώρα άλλες δύο δημοπρασίες από τον PC_5 και PC_6 . Στη λίστα των αρχείων που ζητάει ο PC_6 περιέχονται και τα αρχεία που έλαβε ο PC_1 . Στο πείραμα αυτό εξετάζεται η αντίδραση του συστήματος σε ανανέωση δεδομένων όπου ένας αρχικά δεν διατηρούσε κάποια αρχεία και στη συνέχεια αφού

τα λάβει, κάνει προσφορά για να αναλάβει την μετάδοσή τους.

Στον πίνακα 5.14 δίνονται τα στοιχεία του πειράματος και στον πίνακα 5.15 οι τιμές latency. Στον πίνακα 5.14, στην στήλη του PC_4 που περιέχονται τα αρχεία

Πίνακας 5.14: Τα στοιχεία της δημοπρασίας του πειράματος Γ

<i>Auctioneer</i>	<i>Λίστα Αρχείων</i>	<i>Μέγεθος Αρχείου</i>	PC_1	PC_2	PC_3	PC_4
PC_4	file1	480 Kbytes	file1	file3	file1	file5
	file2	40 MBytes	file5	file4	file2	file8
	file14	18 Mbytes	file6	file5	file9	file12
PC_5	file10	2 Mbytes	file7	file6	file13	file13
	file11	7 MBytes	file8	file8	file14	file15
	file12	4 Mbytes	file10	file12	file15	file16
	file13	17 Mbytes	file11	file13		
	file14	18 MBytes	file12			
	file15	1 MBytes	file14			
	file16	91 MBytes	file16			
PC_6	file1	480 KBytes				file1
	file2	40 MBytes				file2
	file3	2 MBytes				file14
	file4	4 Mbytes				
	file5	29 MBytes				
	file6	4 MBytes				
	file7	2 MBytes				
	file8	52 MBytes				
	file9	23 MBytes				

Πίνακας 5.15: Οι τιμές latency στο πείραμα Γ

<i>latency</i>	PC_1	PC_2	PC_3	PC_4
PC_4	1.007	0.121	1.317	
PC_5	0.109	0.116	0.103	1.535
PC_6	1.59	0.123	1.472	0.939

Πίνακας 5.16: Ανάθεση αρχείων και χρόνοι μεταφοράς τους στο πείραμα Γ

	<i>Assigned files</i>	<i>Transmission time</i>
PC_1	file1	00 : 01 : 102 sec
	file14	00 : 05 : 214 sec
	file10	00 : 02 : 496 sec
	file11	00 : 04 : 645 sec
	file14	00 : 07 : 870 sec
	file16	00 : 16 : 653 sec
	file7	00 : 19 : 080 sec
PC_2	file12	00 : 02 : 833 sec
	file3	00 : 04 : 660 sec
	file4	00 : 06 : 819 sec
	file6	00 : 09 : 587 sec
	file5	00 : 16 : 270 sec
	file8	00 : 22 : 893 sec
PC_3	file2	00 : 06 : 229 sec
	file15	00 : 01 : 854 sec
	file13	00 : 05 : 811 sec
	file9	00 : 11 : 140 sec
PC_4	file1	00 : 01 : 795 sec
	file2	00 : 10 : 278 sec

Στον πίνακα 5.14 τα αρχεία file1, file2 και file14 ακολουθούν κάποιες κενές θέσεις στην στήλη του PC_4 για να φανεί ότι δεν περιέχονταν αρχικά στον PC_4 . Στην δεύτερη χρονολογικά δημοπρασία, στην οποία auctioneer είναι ο PC_5 , οι υπολογιστές δεν έχουν κάποια αρχεία να μεταδώσουν οπότε δεν επηρεάζονται από κάποιο φόρτο μεταφοράς. Ο PC_4 δεν αναλαμβάνει να μεταφέρει κάποιο αρχείο προς τον PC_5 εξαιτίας της πολύ μεγαλύτερης τιμής του latency συγκριτικά με των άλλων τριών. Στην τελευταία δημοπρασία οι PC_1 και PC_3 έχοντας μεγάλη τιμή latency και φόρτο μεταφοράς, αναλαμβάνουν μόνο τα αρχεία που περιέχουν αποκλειστικά αυτοί. Ο PC_4 αναλαμβάνει τα αρχεία file1 και file2, τα οποία δεν διατηρούσε στην αρχή του πειράματος, αφού οι PC_1 και PC_3 έχουν μεγαλύτερο φόρτο και latency από αυτόν. Ο PC_2 αναλαμβάνει να μεταδώσει όλα τα αρχεία που διατηρεί.

Κεφάλαιο 6

Συμπεράσματα

6.1 Ανακεφαλαίωση και Συμπεράσματα

Η χρήση δημοπρασιών αποτελεί μία από τις πιο διαδεδομένες και αποτελεσματικές μεθόδους για τον συντονισμό ομάδων αποτελούμενων από πράκτορες. Ο σκοπός της διπλωματικής εργασίας αυτής ήταν η υλοποίηση ενός πολυπρακτορικού συστήματος ικανού να ολοκληρώσει με την χρήση δημοπρασιών όποια αποστολή του ανατεθεί. Τα κύρια ζητούμενα κατά τον σχεδιασμό του συστήματος ήταν η ανθεκτικότητα και αξιοπιστία του συστήματος όσον αφορά την επίτευξη της αποστολής και η γενικότητα και ευελιξία του όσον αφορά την χρησιμοποίησή του σε διάφορες εφαρμογές.

Για να εξακριβωθεί το κατά πόσο το μοντέλο του συστήματός μας ικανοποιούσε τις παραπάνω απαιτήσεις, το σύστημα δοκιμάστηκε σε δύο διαφορετικές εφαρμογές. Με τον τρόπο υπολογισμού των bids να εξαρτάται από την κάθε εφαρμογή αλλά τον πυρήνα και την λειτουργία του συστήματος να παραμένουν ίδια αποδεικνύεται η γενικότητα και η ευελιξία του συστήματος.

Η ανθεκτικότητα και η αξιοπιστία του συστήματος αποδεικνύεται μέσα από την χρήση του σε αρκετά και διαφορετικά σενάρια. Η σύσταση της ομάδας των

πρακτόρων καθώς και ο auctioneer της εκάστοτε δημοπρασίας διέφεραν για κάθε αποστολή, επαληθεύοντας την ανεξαρτησία της ομάδας από κάποιον συγκεκριμένο κεντρικό πράκτορα. Επίσης, το σύστημα δοκιμάστηκε στην διαχείριση πολλαπλών αιτήσεων δημοπρασιών και εν συνεχεία στην πραγματοποίησή τους. Σε όλες τις περιπτώσεις τα αποτελέσματα και οι λύσεις που δόθηκαν από το σύστημα ήταν ικανοποιητικές και συμφωνούσαν με τις προβλεπόμενες από τους αλγόριθμους δημοπρασίας.

Ολοκληρώνοντας, η δυσκολία έγκειται στη σωστή επιλογή της μεθόδου επίλυσης ενός προβλήματος. Εάν κριθεί ότι η καταλληλότερη και αποτελεσματικότερη μέθοδος είναι αυτή των δημοπρασιών, τότε η αποστολή πρέπει να χωριστεί σε εργασίες τις οποίες θα αναλάβουν οι πράκτορες. Εφόσον όλα αυτά συμβούν, το σύστημα που υλοποιήσαμε μπορεί να εγγυηθεί την επιτυχή πραγματοποίηση των δημοπρασιών και συνεπώς τον συντονισμό της ομάδας των πρακτόρων.

6.2 Μελλοντική Εργασία

Ο συντονισμός των πολυπρακτορικών ομάδων με την χρήση δημοπρασιών αποτελεί σχετικά νέα μέθοδο. Επομένως υπάρχουν αρκετά πεδία πάνω στις δημοπρασίες για περαιτέρω μελέτη και εργασία. Όσον αφορά την συγκεκριμένη διπλωματική, η μελλοντική μελέτη που θα μπορούσε να γίνει χωρίζεται σε τρία κομμάτια.

1. Την υλοποίηση περισσότερων εφαρμογών και σεναρίων. Ενδεικτικά θα περιγράψουμε δύο πιθανά σενάρια που θα μπορούσαν να προσομοιαστούν από το σύστημα που υλοποιήσαμε.
 - Στο πρώτο σενάριο robots λειτουργούν ως πράκτορες, τα οποία αναλαμβάνουν να συναρμολογήσουν μία διαστημική βάση. Η τελική

μορφή που πρέπει να πάρει η βάση αποτελεί την αποστολή της ομάδας ενώ τα επιμέρους εξαρτήματα και ο τρόπος συναρμολόγησης τους αποτελούν τις δημοπρατούμενες εργασίες.

- Στο δεύτερο σενάριο προσομοιάζεται ο τρόπος ανεφοδιασμού εξερευνητικών υποβρυχίων. Κάθε υποβρύχιο που χρειάζεται ανεφοδιασμό πραγματοποιεί μία δημοπρασία στην οποία συμμετέχουν ως bidders πλοία με δεξαμενές καυσίμων. Τα πλοία υπολογίζουν τα bids τους αναλόγως την απόστασή τους από το υποβρύχιο, το ζητούμενο φορτίο καυσίμου και τον ήδη προγραμματισμένο ανεφοδιασμό άλλων υποβρυχίων.
2. Την χρήση των αλγορίθμων που βελτιώνουν τα αποτελέσματα της εφαρμογής των robots. Στην εφαρμογή με τον καθορισμό των διαδρομών των robots, μία τροποποίηση στον τρόπο υπολογισμού των bids των πρακτόρων θα βελτίωνε τα αποτελέσματα των πειραμάτων.
 3. Τέλος, η χρησιμοποίηση του συστήματος σε αληθινά robots. Μετά τον καθορισμό των διαδρομών από την ομάδα πρακτόρων και εφόσον αυτές κρίθηκαν ικανοποιητικές για διάφορους χάρτες, το επόμενο βήμα είναι η ανάπτυξη του συστήματος σε αληθινά robots ή σε αυτοκινούμενα οχήματα. Βέβαια η δυσκολία είναι πολύ μεγάλη αφού τα οχήματα θα πρέπει να είναι εφοδιασμένα με ένα είδος GPS [14] ενώ θα πρέπει να είναι ικανά να προσανατολίζονται εκτός από το να καθορίζουν την θέση τους στο χάρτη. Ταυτόχρονα θα πρέπει να έχει χαρτογραφηθεί και όλη η περιοχή στην οποία κινούνται.

Βιβλιογραφία

- [1] Michail G. Lagoudakis, Vangelis Markakis, David Kempee, Pinar Keskinocak, Sven Koenig, Craig Tovey, Anton Kleywegt, Adam Meyerson, and Sonal Jain, “Auction-Based Multi-Robot Routing”, *Proceedings of Robotics: Science and Systems, MIT, Boston, MA, June 2005*.
- [2] Craig Tovey, Michail G. Lagoudakis, Sonal Jain, and Sven Koenig, “Generation of Bidding Rules for Auction-Based Robot Coordination”, *Proceedings of the 3rd International Multi-Robot Systems Workshop, Naval Research Laboratory, Washington, DC, March 2005*.
- [3] Michail G. Lagoudakis, Mark Berhault, Sven Koenig, Pinar Keskinocak, and Anton Kleywegt, “Simple Auctions with Performance Guarantees for Multi-Robot Task Allocation”, *Proceedings of the 2004 IEEE/JRS International Conference on Intelligent Robots and Systems (IROS 2004), Sendai, Japan, September 2004*.
- [4] M. Dias and A. Stentz, “A free market architecture for distributed control of a multirobot system”, *Proceedings of the International Conference on Intelligent Autonomous Systems, 2000, pp. 115 - 122*.
- [5] B. Gerkey and M. Mataric, “Sold!: Auction methods for multi-robot coordination”, *IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 758 - 768, 2002*.

- [6] R. Zlot, A. Stentz, M. Dias, and S. Thayer, “ Multi-robot exploration controlled by a market economy”, *Proceedings of the International Conference on Robotics and Automation, 2002*, pp. 3016 - 3023.
- [7] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, “ Robot exploration with combinatorial auctions”, *Proceedings of the International Conference on Intelligent Robots and Systems, October 2003*, pp. 1957 - 1962.
- [8] POOJA DEWAN and SANJAY JOSHI, “ Implementation of an auction -based distributed scheduling model for a dynamic job shop environment”, *INT. J. COMPUTER INTEGRATED MANUFACTURING, 2001, VOL. 14, NO. 5, 446-456*
- [9] Benito Mendoza and Jose M. Vidal, “ Bidding Algorithms for a Distributed Combinatorial Auction”, *Proceeding of the Autonomous Agents and Multi-Agent Systems Conference, 2007*
- [10] Alexandros Salazar and Panagiotis Tsiotras, “ An Auction Algorithm for Optimal Satellite Refueling”, *Georgia Institute of Technology, 2005*
- [11] Parkes, D.C. and Shneidman, J. , “ Distributed implementations of Vickrey-Clarke-Groves mechanisms”, *Proceeding of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2004, AAMAS 2004*.
- [12] Stuart Russell and Peter Norvig , “ Artificial Intelligence: A Modern Approach”, *Prentice Hall series in Artificial Intelligence*.
- [13] “ Little Unix Programmers Group (LUPG)’s Little Site”, “<http://users.actcom.co.il/choo/lupg/tutorials/index.html>”.

[14] “GPS definition”, “<http://scign.jpl.nasa.gov/learn/gps1.htm>”.