

Ατομική Διπλωματική Εργασία

**ΑΛΓΟΡΙΘΜΟΣ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ ΕΙΣΩΡΡΟΠΗΣΗΣ
ΦΟΡΤΙΟΥ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ
ΕΜΠΝΕΥΣΜΕΝΟΣ ΑΠΟ ΤΗ ΦΥΣΗ**

Χριστόφορος Χριστοφίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αλγόριθμος για το πρόβλημα εξισορρόπησης φορτίου εμπνευσμένος από τη φύση

Χριστόφορος Χριστοφίδης

Επιβλέπουσα Καθηγήτρια

Άννα Φιλίππου

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

Μάιος 2016

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στην επιβλέπουσα καθηγήτρια μου Δρ. Άννα Φιλίππου, για τη βοήθειά της, την υποστήριξή της και για την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές της, οι οδηγίες της αλλά και κατευθύνσεις που με συμβούλευε να ακολουθήσω, συνέβαλαν στο μέγιστο, για την ολοκλήρωση της εργασίας αυτής.

Θα ήθελα να αφιερώσω τη διπλωματική μου εργασία στους γονείς και την αδελφή μου για τη στήριξη, τη συμπαράσταση, την αγάπη που μου έχουν δείξει, καθώς και για το ότι με είχαν ενθαρρύνει να συνεχίσω τη δουλειά μου επιτυγχάνοντας τους στόχους μου.

Περίληψη

Υπολογιστικό Νέφος ονομάζεται η κατ' αίτηση διαδικτυακή κεντρική διάθεση υπολογιστικών πόρων (όπως δίκτυο, εξυπηρετητές, εφαρμογές και υπηρεσίες) με υψηλή ευελιξία, ελάχιστη προσπάθεια από τον χρήστη και υψηλή αυτοματοποίηση.

Το Υπολογιστικό Νέφος φέρνει πλεονεκτήματα όσον αφορά το κόστος, την ευελιξία και τη διαθεσιμότητα των υπηρεσιών στους χρήστες. Αυτά τα πλεονεκτήματα αυξάνουν τη ζήτηση για τις υπηρεσίες Υπολογιστικού Νέφους, με αποτέλεσμα να υπάρχουν προβλήματα όπως η υψηλή διαθεσιμότητα και επεκτασιμότητα. Η εξισορρόπηση φορτίου έχει σημαντικό ρόλο, αφού επιτρέπει στα Υπολογιστικά Νέφη να ανταπεξέλθουν στην αυξανόμενη ζήτηση, μέσω αποτελεσματικής κατανομής φόρτου εργασίας ομοιόμορφα σε όλους τους κόμβους.

Έχουν προταθεί πολλοί αλγόριθμοι εξισορρόπησης φόρτου εργασίας για το πρόβλημα αυτό. Στόχος τους είναι τα υπολογιστικά νέφη να αποφέρουν επιθυμητά αποτελέσματα με μικρότερο κόστος. «Κόστος» για ένα αλγόριθμο εξισορρόπησης φόρτου μπορεί να είναι η υπολογιστική ισχύς που χρειάζεται για να εφαρμοστεί, η μνήμη που θα χρειαστεί για να δουλέψει και οι επιπτώσεις που μπορεί να έχει (καθυστέρηση/απώλεια εργασιών).

Η παρούσα διπλωματική εργασία έχει ως στόχο τη δημιουργία ενός τέτοιου αλγορίθμου, εμπνευσμένου από τη φύση, την υλοποίηση και την πειραματική σύγκριση του με αλγορίθμους που έχουν προταθεί για το πρόβλημα αυτό στη βιβλιογραφία.

Ο αλγόριθμος αυτός είναι εμπνευσμένος από τον αλγόριθμο που χρησιμοποιεί ένα είδος πουλιών κατά τη μετανάστευση τους [1]. Υλοποιήθηκε στη γλώσσα προγραμματισμού Java και συγκρίνεται με αλγόριθμο εμπνευσμένο από τη συμπεριφορά των μελισσών κατά την αναζήτηση τροφής [2]. Σημείο αναφοράς για τη σύγκριση αποτέλεσε το ποσοστό εργασιών που οι δύο αλγόριθμοι μπορούσαν να ολοκληρώσουν με επιτυχία.

Με το πέρας της πειραματικής μελέτης, παρατηρήθηκε πως ο αλγόριθμος που προτάθηκε δεν είναι πιο γρήγορος από τον αλγόριθμο με τον οποίο συγκρίθηκε, αλλά είναι αποτελεσματικότερος, αφού μέσα από πειραματική μελέτη έχει διαφανεί πως τα ποσοστά εργασιών που δεν εκτελούνται με επιτυχία είναι πολύ μικρότερα.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Κίνητρο	1
	1.2 Στόχοι Εργασίας	2
	1.3 Μεθοδολογία και Αποτελέσματα	2
	1.4 Οργάνωση Κεφαλαίων	3
Κεφάλαιο 2	Υπολογιστικό Νέφος.....	4
	2.1 Υπολογιστικό Νέφος, Σύμπλεγμα και Πλέγμα	4
	2.2 Συνιστώσες Υπολογιστικού Νέφους	5
	2.3 Εξισορρόπηση Φόρτου σε Περιβάλλον Υπολογιστικού Νέφους	7
	2.4 Κατανομή Πόρων	7
Κεφάλαιο 3	Εισαγωγή στο Πρόβλημα Εξισορρόπησης Φόρτου Εργασίας	8
	3.1 Μελέτη Τεχνικών και Αλγορίθμων	8
	3.2 Εξισορρόπηση Φόρτου στο Υπολογιστικό Νέφος	12
Κεφάλαιο 4	Πρόβλημα Διασκορπισμού Πουλιών.....	17
	4.1 Περιγραφή Έρευνας	17
	4.2 Σύγκριση Προβλήματος Εξισορρόπησης Φορτίου στο Υπολογιστικό Νέφος με το Πρόβλημα Διασκορπισμού Πουλιών	19
	4.3 Προτεινόμενος αλγόριθμος με βάση τη μετανάστευση των πουλιών	20
Κεφάλαιο 5	Υλοποίηση και πειραματική Αξιολόγηση Αλγορίθμων.....	21
	5.1 Αλγόριθμος Εμπνευσμένος από τη Μετανάστευση των Πουλιών - Αλγόριθμος 1	21
	5.2 Παραλλαγή Αλγορίθμου Honey Bee Behaviour Based Load Balancing of Tasks in Cloud Computing – Αλγόριθμος 2	25
	5.3 Σύγκριση Αλγορίθμων 1 και 2	27
	5.4 Σενάρια	28
	5.5 Σχολιασμός Αποτελεσμάτων – Συμπεράσματα	115

ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα	116
6.1 Γενικά Συμπεράσματα	116
6.2 Τελική Ανασκόπηση	117
6.3 Μελλοντική Έρευνα	118
 Βιβλιογραφία	 119

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο	1
1.2 Στόχοι Εργασίας	2
1.3 Μεθοδολογία και Αποτελέσματα	2
1.4 Οργάνωση Κεφαλαίων	3

1.1 Κίνητρο

Η εξισορρόπηση φόρτου εργασίας είναι ένα θέμα το οποίο αναπτύχθηκε μαζί με τη δραματική αύξηση των δυνατοτήτων της επιστήμης της Πληροφορικής. Καθώς αναπτύσσονται τα πληροφοριακά συστήματα, προστίθεται επιπλέον φόρτος εργασίας σε αυτά, απαιτώντας την κατάλληλη κατανομή του, έτσι ώστε να μπορούμε να επωφελομαστε πλήρως από αυτά.

Καθώς τα τελευταία χρόνια οι απαιτήσεις των χρηστών έχουν αυξηθεί, ο κλάδος της Πληροφορικής κινήθηκε προς τη χρήση του υπολογιστικού νέφους. Οι τεράστιες ποσότητες δεδομένων που χρήζουν επεξεργασίας καθημερινά απαιτούν μεγάλους, πολλούς και περίπλοκους υπολογισμούς. Θετικές επιστήμες όπως η Βιολογία και η Γενετική βρίσκουν τη λύση τους στο υπολογιστικό νέφος, το οποίο όμως χωρίς την κατάλληλη εξισορρόπηση φόρτου εργασίας δεν είναι όσο αποδοτικό όσο θα έπρεπε.

Το υπολογιστικό νέφος είναι μία υπηρεσία-ανά-ζήτηση, στην οποία κοινοί πόροι, υπηρεσίες και λογισμικό δίνονται στους χρήστες με βάση τις απαιτήσεις τους σε συγκεκριμένη στιγμή. Πλέον η εξισορρόπηση φόρτου εργασίας στα υπολογιστικά νέφη αποτελεί μεγάλη πρόκληση, αφού δεν είναι πρακτικό, ούτε οικονομικό να υπάρχουν υπηρεσίες που μένουν αδρανείς λόγω κακής κατανομής υπηρεσιών στο Νέφος.

Αρκετοί αλγόριθμοι έχουν προταθεί μέχρι τώρα με σκοπό την εξισορρόπηση φόρτου εργασίας στα υπολογιστικά νέφη, χωρίς όμως να αποτελεί κανένας τη βέλτιστη λύση για το πρόβλημα. Μερικές κλασσικές προσεγγίσεις αναφέρονται στην εργασία αυτή.

1.2 Στόχοι Εργασίας

Στην παρούσα διπλωματική εργασία γίνεται μελέτη του προβλήματος εξισορρόπησης φόρτου εργασίας στο υπολογιστικό νέφος, με σκοπό τη δημιουργία αλγορίθμου μέσω του οποίου να επιτυγχάνεται η βέλτιστη εκτέλεση εργασιών (γρήγορα αποτελέσματα, χωρίς απώλεια εργασιών). Για τη δημιουργία του αλγορίθμου, μελετούμε Αλγόριθμους Μετανάστευσης Πληθυσμών [1] και χρησιμοποιούμε στοιχεία της μεταναστευτικής διαδικασίας που ακολουθούν, με στόχο να εξακριβώσουμε εάν αυτοί μπορούν να χρησιμοποιηθούν για Εξισορρόπηση Φόρτου στο Υπολογιστικό Νέφος. Ακόμη μελετούμε πειραματικά τον αλγόριθμο που δημιουργήσαμε. Τέλος συγκρίνουμε τον αλγόριθμο που κατασκευάσαμε, με ένα δεύτερο αλγόριθμο εξισορρόπησης φόρτου, εμπνευσμένο από τη συμπεριφορά που έχουν οι μέλισσες κατά την αναζήτηση τροφής.

1.3 Μεθοδολογία και αποτελέσματα

Κατά την πρώτη φάση της εργασίας έγινε θεωρητική μελέτη αλγορίθμων που προτάθηκαν μέχρι στιγμής για επίλυση του προβλήματος της εξισορρόπησης φόρτου εργασίας σε υπολογιστικά νέφη. Επίσης έγινε προσεκτική μελέτη του αλγορίθμου που χρησιμοποιεί ένας πληθυσμός πουλιών με όνομα wood thrush (*Hylocichla mustelina*) για τις μεταναστεύσεις του. Η συμπεριφορά των πουλιών αυτών κατά τη μετανάστευση σε άλλες περιοχές παρουσιάζει μεγάλο ενδιαφέρον όσον αφορά τον «αλγόριθμο» που χρησιμοποιούν.

Ιδιαίτερα κάνει εντύπωση η πιθανότητα μετανάστευσης, αφού παρατηρήσαμε ότι τα πουλιά με οικογένεια μεταναστεύουν με μικρότερη πιθανότητα από τα άλλα, ακόμη κι αν η περιοχή που είναι εγκατεστημένα δεν τους ικανοποιεί. Το σημείο αυτό μελετάται σε δεύτερο στάδιο κατά την πειραματική αξιολόγηση του αλγορίθμου μας.

Στη δεύτερη φάση της εργασίας, ο αλγόριθμος των πουλιών προσαρμόστηκε στο πρόβλημα εξισορρόπησης φόρτου στο Υπολογιστικό Νέφος. Εντοπίστηκαν ομοιότητες και διαφορές με τα περιβάλλοντα και τις οντότητες των δύο προβλημάτων.

Στην τρίτη φάση της εργασίας έγινε πειραματική μελέτη, του προτεινόμενου αλγορίθμου εξισορρόπησης φόρτου εργασίας βασισμένο στον Αλγόριθμο Μετανάστευσης Πληθυσμών [1]. Έπειτα ο αλγόριθμος αυτός συγκρίθηκε με την παραλλαγή ενός αλγορίθμου που πρότειναν οι Harshit Gupta και Kalicharan Sahu [3], ο οποίος ήταν και κοντά στο μοντέλο που μελετούμε.

Από τις προσομοιώσεις που υλοποιήσαμε καταλήξαμε στο συμπέρασμα πως ο δικός μας αλγόριθμος ήταν σαφώς καλύτερος για το περιβάλλον του υπολογιστικού νέφους, αφού η εκτέλεση εργασιών χρησιμοποιώντας το δικό μας αλγόριθμο έχει ως αποτέλεσμα τη μικρότερη απώλεια εργασιών.

Για την πειραματική μελέτη δημιουργήσαμε διάφορα σενάρια διαφοροποιώντας παραμέτρους όπως η χωρητικότητα των κόμβων του υπολογιστικού νέφους, ο αριθμός των κόμβων σε αυτό και ο αριθμός των εργασιών που εισέρχονται στο νέφος.

1.4 Οργάνωση Κεφαλαίων

Στο Κεφάλαιο 2 θα μελετήσουμε τί είναι το υπολογιστικό νέφος, όπως και τί είναι εξισορρόπηση φόρτου εργασίας σε αυτό μέσω αποτελεσματικής κατανομής φόρτου. Στο Κεφάλαιο 3 βλέπουμε τεχνικές και αλγορίθμους που ήδη έχουν προταθεί για εξισορρόπηση φόρτου εργασίας στο υπολογιστικό νέφος, χωρίζοντας τους σε κατηγορίες και μελετώντας τους στο υπολογιστικό νέφος. Στο Κεφάλαιο 4 παρουσιάζουμε το πρόβλημα διασκορπισμού πουλιών, περιγράφοντας την έρευνα, συγκρίνοντας το πρόβλημα διασκορπισμού πουλιών με το πρόβλημα εξισορρόπησης φόρτου στο υπολογιστικό νέφος και εφαρμόζοντας αλγόριθμο που δημιουργήσαμε από την έρευνα αυτή. Στο Κεφάλαιο 5 βλέπουμε υλοποιημένους δύο αλγόριθμους, τον ένα που δημιουργήσαμε εμείς και τον άλλο τον οποίο μελετήσαμε στο Κεφάλαιο 3. Συγκρίνουμε τους δύο αλγόριθμους βλέποντας ποιός υπερέχει. Στο Κεφάλαιο 6 καταλήγουμε με γενικά συμπεράσματα τα οποία βγήκαν καθόλη τη διάρκεια της έρευνας αυτής και ανασκοπούμε στο μέλλον για μελλοντική έρευνα.

Κεφάλαιο 2

Υπολογιστικό Νέφος

2.1 Υπολογιστικό Νέφος, Σύμπλεγμα και Πλέγμα	4
2.2 Συνιστώσες Υπολογιστικού Νέφους	5
2.3 Εξισορρόπηση Φόρτου σε Περιβάλλον Υπολογιστικού Νέφους	7
2.4 Κατανομή Πόρων	7

Ο όρος υπολογιστικό νέφος δημιουργήθηκε με τη σύμπτυξη δύο όρων του κλάδου της Πληροφορικής: του όρου του υπολογισμού και του όρου του «νέφους». Το νέφος είναι μία πηγή από πόρους. Η διάστρωση του αναφέρεται τόσο στις εφαρμογές που καταφθάνουν στους χρήστες ως υπηρεσίες μέσω διαδικτύου, όσο και στο υλικό και λογισμικό των κέντρων πληροφοριών που είναι υπεύθυνα για να μεταβιβάσουν αυτές τις πληροφορίες. Για να χρησιμοποιηθούν κατάλληλα αυτοί οι πόροι και να εξασφαλιστεί η διαθεσιμότητά τους στους χρήστες, ο υπολογισμός πάνω στο «νέφος» γίνεται με συγκεκριμένα κριτήρια τα οποία ορίστηκαν σε συμφωνία με τους χρήστες σε επίπεδο υπηρεσιών. Η διάστρωση αυτή του υπολογιστικού νέφους γίνεται διαθέσιμη στους χρήστες κατά παραγγελία σε μορφή του «pay-as-you-go», δηλαδή πληρώνουν καθώς προχωρούν. Οι υπολογισμοί γίνονται στα υπολογιστικά νέφη με σκοπό να επιδιώξουν τη μέγιστη αξιοποίηση πόρων με μεγαλύτερη διαθεσιμότητα με όσο το λιγότερο χαμηλό κόστος.

2.1 Υπολογιστικό Νέφος, Σύμπλεγμα και Πλέγμα

Οι συλλογές (Clusters) είναι παράλληλα καταναμεμημένα συστήματα, κάτω από τον έλεγχο ενός διοικητικού τομέα. Ο κόμβος, σε μία συλλογή, ενσωματώνεται με σκοπό να σχηματιστεί ένας ενιαίος πόρος υπολογιστών. Το Πλέγμα (Grid) είναι μία συσσωμάτωση από αυτόνομους πόρους που κατανέμονται γεωγραφικά. Οι κόμβοι στο πλέγμα επιτρέπουν την ανταλλαγή και τη δυναμική επιλογή κατά το χρόνο εκτέλεσης. Το υπολογιστικό νέφος είναι η εξελιγμένη, νέα γενιά των συλλογών και των πλεγμάτων. Παρόμοια με τα προηγούμενα δύο, το υπολογιστικό νέφος είναι και αυτό μία συλλογή από παράλληλα καταναμεμημένα συστήματα. Σε αντίθεση με τις συλλογές και τα πλέγματα, το υπολογιστικό νέφος δεν ορίζεται από ένα μοναδικό πεδίο, αλλά αποτελείται από πολλούς τομείς και οι κόμβοι του είναι «εικονικοί».

2.2 Συνιστώσες Υπολογιστικού Νέφους [2]

Πελάτες υπολογιστικού νέφους: Χρησιμοποιούν τις διάφορες υπηρεσίες που τους παρέχονται από αυτό. Προτού χρησιμοποιήσουν οποιαδήποτε λειτουργία του, πρέπει να συμφωνήσουν στους όρους (Service Level Agreement - SLA) που ορίζονται από τους παροχείς. Χρησιμοποιούν τις υπηρεσίες αυτές κατά παραγγελία, και πληρώνουν ανάλογα με τη χρήση που κάνουν κάθε φορά. Το νέφος παρέχει στους χρήστες την ευελιξία παρέχοντας τις υπηρεσίες του, ενσωματώνοντας την πληροφοριακή χρησιμότητα.

Πριν συμφωνήσουν οι χρήστες στους όρους που τους βάζει ο κάθε παροχέας, πρέπει να βεβαιωθούν πως οι όροι αυτοί περιέχουν συγκεκριμένη ποιότητα υπηρεσιών, όπως: ασφάλεια, προέλευση, προστασία δεδομένων, υψηλή διαθεσιμότητα, μειωμένο κόστος και ευκολία στη χρήση.

Άρα, για τους πελάτες, η χρήση του υπολογιστικού νέφους είναι ένα σενάριο όπου εκείνοι μπορούν να έχουν πρόσβαση σε κάθε είδος υποδομής, λογισμικού ή πλατφόρμας με ασφαλή τρόπο, με μειωμένο κόστος, όποτε το ζητήσουν και με ευκολία στη χρήση.

Παροχέας υπολογιστικού νέφους: Ο παροχέας του υπολογιστικού νέφους μπορεί να το παρέχει ως δημόσια, προσωπική ή υβριδική υπηρεσία. Είναι υπεύθυνος στο να δημιουργήσει το νέφος. Ο παροχέας του υπολογιστικού νέφους για να θεωρείται καλός στη δουλειά του, πρέπει να πετύχει σωστή τροφοδοσία πόρων η οποία περιέχει δύο κύριες εργασίες. Η διαχείριση τεράστιων πακέτων από πόρους, οι οποίοι και καθιστούν το νέφος, και σωστή διαμοίραση των πόρων αυτών στους χρήστες.

Προγραμματιστής υπολογιστικού νέφους: Ο προγραμματιστής του υπολογιστικού νέφους βρίσκεται μεταξύ του χρήστη και του παροχέα. Είναι υπεύθυνος να λάβει υπόψη του τις ανάγκες και των δύο και να επιλαμβάνεται όλων των τεχνικών λεπτομερειών που χρειάζονται για να τηρούνται οι απαιτήσεις και των δύο. Ο προγραμματιστής έχει ως κίνητρο να εξαλείψει το κενό μεταξύ των χρηστών και του παροχέα.

Ρυθμίσεις υπολογιστικού νέφους: Τα *προσωπικά* υπολογιστικά νέφη ελέγχονται από εταιρείες ή επιχειρήσεις για την εσωτερική τους χρήση. Το χρησιμοποιούν είτε για να διαχειρίζονται μεγάλα αρχεία δεδομένων, είτε για να παρέχουν αρκετούς πόρους κατά παραγγελία στους υπαλλήλους τους ή τους πελάτες τους. Παράδειγμα μερικών προσωπικών υπολογιστικών νεφών είναι τα: OpenStack, VMware και CloudStack.

Τα δημόσια υπολογιστικά νέφη χρησιμοποιούνται από μεμονωμένα άτομα, ή από οργανισμούς με βάση τις απαιτήσεις τους. Είναι πολύ αποδοτικά αλλά η εχεμύθεια είναι το μεγαλύτερο πρόβλημα που αντιμετωπίζει κάποιος σε αυτά, αφού είναι πιο ευάλωτα από τα προσωπικά υπολογιστικά νέφη. Παράδειγμα μερικών προσωπικών υπολογιστικών νεφών είναι τα: Amazon web services, Google Computer Engine, Microsoft Azure και HP cloud.

Ένα υβριδικό υπολογιστικό νέφος είναι ένας συνδυασμός προσωπικού και δημόσιου υπολογιστικού νέφους. Επιτρέπει σε επιχειρήσεις να διαχειρίζονται μερικές πληροφορίες εσωτερικά και μερικές εξωτερικά. Το κύριο μειονέκτημα είναι ότι η πολυπλοκότητα της όλης διαχείρισης αυξάνεται μαζί με τις ανησυχίες περί μέτρων ασφαλείας.

Περιβάλλον υπολογιστικού νέφους: Ένα τυπικό περιβάλλον υπολογιστικού νέφους αποτελείται από: κέντρα δεδομένων (datacenters), ξενιστές (hosts), εικονικές μηχανές (virtual machines), εφαρμογές (applications) όπως και λογισμικό (system software). Τα κέντρα δεδομένων είναι υπεύθυνα στο να παρέχουν υπηρεσίες εσωτερικής διάστροφης στους χρήστες. Καθίστανται «σπίτι» σε πολλούς ξενιστές, ή ξένες οντότητες. Οι ξενιστές σε υπολογιστικά νέφη είναι υλικοί ξενιστές (physical servers) οι οποίοι έχουν προκαθορισμένες δυνατότητες επεξεργασίας. Είναι υπεύθυνοι στο να παρέχουν υπηρεσίες σε επίπεδο λογισμικού στους χρήστες και έχουν δική τους κύρια και δευτερεύουσα μνήμη. Καθίστανται «σπίτι» σε εικονικές μηχανές ή πολλές απεικονίσεις εικονικών μηχανών. Οι εικονικές μηχανές επιτρέπουν την ανάπτυξη όπως και την εγκατάσταση ειδικών προσαρμοσμένων εφαρμογών. Είναι αντιστοιχισμένες σε ξενιστή ο οποίος ικανοποιεί τα χαρακτηριστικά τους, όπως κύρια και δευτερεύουσα μνήμη, ταχύτητα επεξεργασίας και λογισμικό.

2.3 Εξισορρόπηση Φόρτου σε Περιβάλλον Υπολογιστικού Νέφους

Λόγω υψηλής ζήτησης των υπηρεσιών του υπολογιστικού νέφους, πολλές φορές αυτό δεν μπορεί να αντεπεξέλθει, έχοντας έτσι ως αποτέλεσμα την καθυστέρηση των υπηρεσιών, ή και την καθόλου εξυπηρέτηση των πελατών του. Η εξισορρόπηση φόρτου στα υπολογιστικά νέφη παρέχει αποδοτικές λύσεις σε διάφορα προβλήματα που παρουσιάζονται σε αυτά και τα κάνει ικανά να αντεπεξέλθουν στην αυξανόμενη ζήτηση, μέσω αποτελεσματικής κατανομής φόρτου εργασίας ομοιόμορφα σε όλους τους κόμβους. Συγκεκριμένα, η εξισορρόπηση φόρτου στα υπολογιστικά νέφη είναι μία μέθοδος η οποία κατανέμει ομοιόμορφα το φόρτο εργασίας των υπερφορτωμένων κόμβων. Επιτυγχάνει καλή χρήση των πόρων, την καλύτερη ικανοποίηση των χρηστών και την αποτελεσματική βελτίωση της συνολικής απόδοσης του συστήματος. Όταν οι εργασίες καταφθάνουν από διαφορετικές τοποθεσίες, ο εξισορροπιστής φόρτου εργασίας αναλαμβάνει να τις κατανέμει ομοιόμορφα. Κατά την εξισορρόπηση φόρτου πρέπει να λαμβάνονται υπόψη η τροφοδοσία πόρων (ή κατανομή πόρων) και ο χρονοπρογραμματισμός σε κατανεμημένα συστήματα. Αυτό θα εξασφαλίσει ότι 1) οι πόροι θα είναι εύκολα προσβάσιμοι κατά παραγγελία, 2) οι πόροι χρησιμοποιούνται πάντα βέλτιστα άσχετα αν το νέφος είναι υπερφορτωμένο, ή υποφορτωμένο, 3) η ενέργεια εξοικονομείται σε περίπτωση χαμηλού φόρτου και 4) το κόστος χρησιμοποίησης πόρων είναι το ελάχιστο δυνατόν.

2.4 Κατανομή Πόρων

Η κατανομή πόρων είναι η αντιστοίχιση των πόρων σε διαφορετικές οντότητες του υπολογιστικού νέφους κατά τη ζήτησή τους. Οι πόροι πρέπει να κατανέμονται με τέτοιο τρόπο έτσι ώστε κανένας κόμβος στο υπολογιστικό νέφος να μην είναι υπερφορτωμένος, αλλά ούτε να υποβάλλονται σε σπατάλη. Οι διεργασίες εκτελούνται στις εικονικές μηχανές. Κάθε διεργασία έχει συγκεκριμένες απαιτήσεις για την επιτυχημένη ολοκλήρωσή της. Η εικονική μηχανή στην οποία είναι κατανεμημένη η διεργασία οφείλει να παρέχει τους απαραίτητους πόρους σε αυτή και οι διεργασίες πρέπει να κατανεμηθούν σε εικονική μηχανή ανάλογα με τις ρυθμίσεις και τη διαθεσιμότητά της.

Κεφάλαιο 3

Εισαγωγή στο Πρόβλημα Εξισορρόπησης Φόρτου Εργασίας

3.1 Μελέτη Τεχνικών και Αλγορίθμων	8
3.3 Εξισορρόπηση Φόρτου στο Υπολογιστικό Νέφος	12

Στην Επιστήμη της Πληροφορικής, εξισορρόπηση φόρτου εργασίας ονομάζεται η διανομή φόρτου εργασίας μεταξύ πολλών υπολογιστικών πόρων, όπως ηλεκτρονικοί υπολογιστές, συμπλέγματα υπολογιστών, δικτύων, κεντρικών μονάδων επεξεργασίας ή σκληροί δίσκοι. Η εξισορρόπηση φόρτου εργασίας στοχεύει στη μέγιστη αξιοποίηση πόρων, απόδοσης, ελαχιστοποίηση χρόνου ανταπόκρισης και στην αποφυγή υπερφόρτωσης σε οποιοδήποτε από τους υπολογιστικούς πόρους.

Η σωστή εξισορρόπηση φόρτου εργασίας ίσως να αποτελεί ένα από τα μεγαλύτερα και σημαντικότερα προβλήματα σε πολλούς, αν όχι όλους τους κλάδους της Πληροφορικής.

Κατά την παρούσα μελέτη έχει γίνει μελέτη τεχνικών και αλγορίθμων εξισορρόπησης φόρτου εργασίας με σκοπό την εύρεση κατάλληλου αλγορίθμου, ο οποίος να δίνει μια βέλτιστη λύση στο πρόβλημα αυτό, με βάση τα επιθυμητά αποτελέσματα.

3.1 Μελέτη Τεχνικών και Αλγορίθμων

Ο μηχανισμός εξισορρόπησης φόρτου εργασίας, έχει σκοπό να διαμοιράσει το φόρτο εργασίας σε κάθε υπολογιστικό κόμβο, μεγιστοποιώντας έτσι την αξιοποίηση των πόρων και ελαχιστοποιώντας το συνολικό χρόνο εκτέλεσης.

Έτσι, ο μηχανισμός εξισορρόπησης φόρτου πρέπει να είναι «δίκαιος» στο διαμοιρασμό του φόρτου μεταξύ υπολογιστικών κόμβων, δηλαδή η διαφορά μεταξύ του πιο βαρυφορτωμένου κόμβου και του πιο ελαφρά-φορτωμένου κόμβου να είναι η ελάχιστη δυνατή.

Για να είναι αυτό δυνατόν, ένας αλγόριθμος πρέπει να είναι λειτουργικός, δηλαδή να μπορεί να εξυπηρετεί κάθε αίτημα που καταφθάνει και να το τοποθετεί στην καταλληλότερη μηχανή, χωρίς όμως να ξέρει τη λεπτομερή κατάσταση όλου του δικτύου. Επιπρόσθετα, ο αλγόριθμος αυτός θα πρέπει να είναι προσαρμοστικός, δηλαδή να μπορεί να προσαρμόζεται γρήγορα σε αλλαγές του περιβάλλοντος κατά το χρόνο εκτέλεσης, λόγω της ραγδαίας αυξομειώσης των αιτημάτων.

Μπορούμε να κατατάξουμε τους αλγόριθμους αναλόγως του περιβάλλοντος που βρίσκονται και τους μηχανισμούς συμπεριφοράς τους.

Η εξισορρόπηση φόρτου εργασίας αποτελείται από τρεις φάσεις, τη συλλογή πληροφοριών, την επιλογή απόφασης και την ανακατανομή φόρτου.

Κατά τη συλλογή πληροφοριών, καταγράφονται πληροφορίες για την κατανομή του φόρτου και την κατάσταση του περιβάλλοντος και το σύστημα ανιχνεύει κατά πόσο υπάρχει ισορροπία ή όχι. Στην επιλογή απόφασης υπολογίζεται η ιδανική κατανομή φορτίου και στην ανακατανομή φόρτου μετακινείται ο φόρτος από ένα υπερφορτωμένο κόμβο σε ένα άλλο, υποφορτωμένο.

Όλοι οι αλγόριθμοι ισορρόπησης φόρτου μπορούν να κατηγοριοποιηθούν σε τρεις κατηγορίες: στατικοί, δυναμικοί και προσαρμοστικοί.

Στους στατικούς αλγόριθμους οι αποφάσεις παίρνονται κατά την εκτέλεσή τους, σε αντίθεση με τους δυναμικούς οι οποίοι ανακατανέμουν το φόρτο ανάλογα με την αλλαγή της κατάστασης του συστήματος που τρέχουν. Οι προσαρμοστικοί αλγόριθμοι, προσαρμόζουν τις δράσεις τους, αλλάζοντας δυναμικά τις παραμέτρους τους, ή ακόμη και τις πολιτικές τους για να ταιριάζουν στο σύστημα στο οποίο τρέχουν. Σε κάθε περιοχή του δικτύου εφαρμόζεται ο αλγόριθμος που είναι πιο αποδοτικός.

Επιπρόσθετα, οι μέθοδοι για ισορρόπηση φόρτου μπορούν να κατηγοριοποιηθούν σε τρεις κλάσεις: κεντρικοί, κατανεμημένοι (αποκεντρωμένοι) και ιεραρχικοί.

- Σε μια κεντρική προσέγγιση, όλες οι εργασίες υποβάλλονται σε ένα ενιαίο χρονοδιάγραμμα, το οποίο είναι υπεύθυνο για τον προγραμματισμό των εργασιών στους διαθέσιμους πόρους.
- Σε ένα κατανεμημένο μοντέλο δεν υπάρχουν χρονοδιαγράμματα και ο προγραμματισμός εργασιών γίνεται από τις αιτήσεις των πόρων.
- Σε ένα ιεραρχικό μοντέλο οι προγραμματιστές οργανώνονται ιεραρχικά. Οι πόροι υψηλής σημασίας προγραμματίζονται σε ψηλότερο επίπεδο, ενώ οι πόροι χαμηλής σημασίας προγραμματίζονται σε χαμηλότερο επίπεδο. Αυτό το μοντέλο είναι συνδυασμός των δύο πιο πάνω μοντέλων.

Ακολούθως καταγράφονται μερικές κλασσικές προσεγγίσεις εξισορρόπησης φόρτου εργασίας, οι οποίες αναφέρονται στο [2].

Τυχαία Προσέγγιση: Είναι η απλούστερη προσέγγιση εξισορρόπησης φόρτου, η οποία αναθέτει εργασίες σε πόρους με ένα τυχαίο τρόπο. Οι προδιαγραφές των πόρων ή των εργασιών δεν παίζουν κανένα ρόλο. Αυτή η μέθοδος αν και δεν είναι ιδιαίτερα αποδοτική, έχει πολύ μικρό χρόνο απόκρισης.

Κυκλική Ανάθεση: Το σύστημα αναθέτει τις εργασίες σε πόρους κυκλικά. Τοποθετεί τις διαθέσιμες μηχανές σε ουρά και επιλέγει να αναθέσει το αίτημα που καταφθάνει στη μηχανή που βρίσκεται στην πρώτη θέση. Έπειτα η μηχανή αυτή μετακινείται στο τέλος της ουράς. Οι προδιαγραφές των πόρων ή των εργασιών δεν παίζουν κανένα ρόλο. Είναι αρκετά αποδοτικός αλγόριθμος στα περισσότερα συστήματα. Υπάρχουν περιπτώσεις όμως στις οποίες ο αλγόριθμος δημιουργεί προβλήματα απόδοσης αφού ορισμένες πιο «απαιτητικές» εργασίες θα ήταν καλύτερο να ανατεθούν σε μηχανές με καλύτερες επιδόσεις παρά σε αυτές στις οποίες κατανέμονται.

Κυκλική Ανάθεση με Βάρη: Ο αλγόριθμος κυκλικής ανάθεσης με βάρη λειτουργεί όπως ο απλός αλγόριθμος κυκλικής ανάθεσης με μόνη διαφορά το ότι κάθε μηχανή έχει «βάρος» το οποίο και αντιπροσωπεύει την επίδοση της κάθε μηχανής. Στις περιπτώσεις όπου ο απλός αλγόριθμος κυκλικής ανάθεσης δημιουργεί προβλήματα, δηλαδή στις περιπτώσεις όπου ορισμένες πιο «απαιτητικές» εργασίες θα ήταν καλύτερο να ανατεθούν σε μηχανές με καλύτερες επιδόσεις παρά σε αυτές που κατανέμονται, ο αλγόριθμος τις κατανέμει σε μηχανές με μεγαλύτερο βάρος, και άρα επίδοση. Παρόλα αυτά, τα βάρη που ανατίθενται σε κάθε μηχανή παραμένουν τα ίδια καθ' όλη τη διάρκεια του αλγορίθμου, χωρίς να γίνεται έλεγχος αν εξακολουθούν να ισχύουν.

Δυναμικός Αλγόριθμος Κυκλικής Ανάθεσης με Βάρη: Αυτός ο αλγόριθμος, λειτουργεί όπως ο αλγόριθμος κυκλικής ανάθεσης με βάρη. Η διαφορά είναι πως τα βάρη κατά τη διάρκεια εκτέλεσης του αλγορίθμου αλλάζουν, αναλόγως του φόρτου κάθε μηχανής και άρα της αποδοτικότητάς της. Κάθε φορά που μπαίνει ή βγαίνει φόρτος σε μία μηχανή, ο φόρτος της, όπως και η αποδοτικότητά της αυξάνεται ή μειώνεται ανάλογα (μεγαλύτερος φόρτος συνεπάγεται μικρότερη αποδοτικότητα). Έτσι, κάθε μηχανή προτού μπει στη ουρά αναμονής, εξετάζεται για να ρυθμιστεί αναλόγως το βάρος της. Αυτό λύει το πρόβλημα του αλγορίθμου κυκλικής ανάθεσης με βάρη, προσθέτοντας όμως κόστος στον αλγόριθμο, λόγω των συνεχών υπολογισμών επανεξέτασης βαρών.

Μικρότερου Χρόνου Εκτέλεσης (Minimum Execution Time): Ο αλγόριθμος αυτός αναθέτει κάθε εργασία στη μηχανή η οποία θα του αποδώσει το μικρότερο χρόνο εκτέλεσης,

ανεξάρτητα εάν ο πόρος αυτός είναι διαθέσιμος ή όχι εκείνη τη στιγμή. Προσπαθεί δηλαδή να βρει το καλύτερο «ταίριασμα» μεταξύ εργασίας και μηχανής, χωρίς να έχει υπόψη του το φόρτο κάθε μηχανής. Ο αλγόριθμος ελαχιστοποιεί το χρόνο αναμονής των εργασιών, αλλά δημιουργεί ανισορροπία στο δίκτυο.

MCT (Minimum Completion Time): Σε αυτόν τον αλγόριθμο, κάθε εργασία ανατίθεται στη μηχανή η οποία θα της αποδώσει το μικρότερο χρόνο ολοκλήρωσης της. Ο χρόνος ολοκλήρωσης υπολογίζεται ως το άθροισμα του χρόνου αναμονής της εργασίας προτού ανατεθεί σε κάποια μηχανή, με το χρόνο εκτέλεσης της εργασίας. Αυτό έχει ως αποτέλεσμα εργασίες να ανατίθενται σε πόρους που δεν έχουν τον ελάχιστο χρόνο εκτέλεσης.

Min-Min: Αρχικά ο αλγόριθμος αρχίζει με το να υπολογίζει το χρόνο ολοκλήρωσης κάθε εργασίας. Έπειτα επιλέγεται η εργασία η οποία έχει το μικρότερο χρόνο εκτέλεσης και η μηχανή που έχει το μικρότερο χρόνο ολοκλήρωσης για όλες τις εργασίες. Τελικά η εργασία και η μηχανή που επιλέχθηκε, αντιστοιχούνται και ο χρόνος ετοιμότητας της μηχανής ανανεώνεται. Ο αλγόριθμος επαναλαμβάνεται μέχρι όλες οι εργασίες να αντιστοιχηθούν με μηχανές. Το όφελος του αλγορίθμου έγκειται στο ότι η εργασία με το μικρότερο χρόνο εκτέλεσης εκτελείται. Το μειονέκτημα του αλγορίθμου είναι ότι μερικές εργασίες μπορεί να μην επιλέγονται και ο χρόνος αναμονής τους να αυξάνεται δραματικά.

Max-Min: Αρχικά ο αλγόριθμος υπολογίζει το χρόνο ολοκλήρωσης κάθε εργασίας. Έπειτα επιλέγεται η εργασία η οποία έχει το μεγαλύτερο χρόνο εκτέλεσης και η μηχανή που έχει το μικρότερο χρόνο ολοκλήρωσης για όλες τις εργασίες. Τελικά η εργασία και η μηχανή που επιλέχθηκε, αντιστοιχούνται και ο χρόνος ετοιμότητας της μηχανής ανανεώνεται. Ο αλγόριθμος επαναλαμβάνεται μέχρι όλες οι εργασίες να αντιστοιχηθούν με μηχανές. Το όφελος του αλγορίθμου έγκειται στο ότι η εργασία με το μεγαλύτερο χρόνο εκτέλεσης εκτελείται. Το μειονέκτημα του αλγορίθμου είναι ότι μερικές εργασίες μπορεί να μην επιλέγονται και ο χρόνος αναμονής τους να αυξάνεται δραματικά.

Suffrage: Κάθε εργασία έχει συγκεκριμένη ή συγκεκριμένες μηχανές στις οποίες θα έχει τον ελάχιστο-καλύτερο χρόνο εκτέλεσης. Εάν μία εργασία δεν ανατεθεί σε συγκεκριμένη μηχανή, θα «υποφέρει», εννοώντας πως ο χρόνος εκτέλεσης της αυξάνεται δραματικά. Για να λυθεί αυτό, η εργασία θα πάρει ψηλή προτεραιότητα, «suffrage value». Το suffrage value για κάθε εργασία υπολογίζεται ως η διαφορά μεταξύ του πρώτου καλύτερου μικρότερου χρόνου εκτέλεσης και του δεύτερου καλύτερου μικρότερου χρόνου εκτέλεσης.

3.2 Εξισορρόπηση Φόρτου στο Υπολογιστικό Νέφος

Στο Κεφάλαιο 2, μιλήσαμε για το περιβάλλον του υπολογιστικού νέφους. Δώσαμε τον ορισμό του, τα χαρακτηριστικά του, από τί αποτελείται και ποιές λειτουργίες προσφέρει. Επίσης στο Κεφάλαιο 2 αναφερθήκαμε στην εξισορρόπηση φόρτου εργασίας, τα είδη αλγορίθμων που υπάρχουν, αλλά και συγκεκριμένους, πιο ευρέως διαδεδομένους αλγόριθμους. Σε αυτό το κεφάλαιο αναφερόμαστε σε εξισορρόπηση φόρτου εργασίας στο περιβάλλον του υπολογιστικού νέφους. Διαχωρίζουμε την περιγραφή μας με βάση το περιβάλλον του υπολογιστικού νέφους, το οποίο μπορεί να είναι είτε δυναμικό είτε στατικό αναλόγως των απαιτήσεων που θα έχουν οι χρήστες του. Ακολούθως καταγράφονται προσεγγίσεις εξισορρόπησης φόρτου με βάση το περιβάλλον του νέφους, οι οποίες αναφέρονται στο [6].

Εξισορόπηση σε Στατικό περιβάλλον: Σε στατικό περιβάλλον ο παροχέας του νέφους εγκαθιστά ομογενείς πόρους οι οποίοι δεν είναι ευέλικτοι και άρα πρέπει το σύστημα να έχει γνώση των χαρακτηριστικών τους (χωρητικότητα, ταχύτητα επεξεργασίας, κύρια και δευτερεύουσα μνήμη). Τα χαρακτηριστικά αυτά δεν μπορούν να αλλάξουν καθ' όλη τη διάρκεια της εκτέλεσης του αλγορίθμου εξισορρόπησης. Επίσης οι αλγόριθμοι που εφαρμόζονται σε στατικό περιβάλλον δεν μπορούν να αλλάξουν για να προσαρμοστούν ανάλογα με το φόρτο εργασίας.

Ο πιο γνωστός αλγόριθμος εξισορρόπησης φόρτου εργασίας υπολογιστικού νέφους σε στατικό περιβάλλον είναι ο αλγόριθμος κυκλικής ανάθεσης, ο οποίος και περιγράφηκε πιο πάνω.

Εξισορρόπηση φόρτου με βάση τη χωρική κατανομή των κόμβων: Οι κόμβοι σε ένα υπολογιστικό νέφος είναι σε μεγάλο βαθμό κατανεμημένοι. Ως εκ τούτου, ο τρόπος με τον οποίο κατανέμονται οι κόμβοι ορίζουν και τον τύπο του αλγόριθμου που πρέπει να χρησιμοποιείται. Υπάρχουν τέσσερα είδη που καθορίζουν ποιοί κόμβοι είναι υπεύθυνοι για την εξισορρόπηση του φορτίου:

1. **Κεντρική εξισορρόπηση φόρτου:** Σε κεντρικές τεχνικές εξισορρόπησης φόρτου, η κατανομή φόρτου και οι αποφάσεις χρονοπρογραμματισμού γίνονται από ένα κόμβο. Ο κόμβος αυτός είναι υπεύθυνος για την αποθήκευση όλης της γνώσης του δικτύου και μπορεί να εφαρμόσει στατικές ή δυναμικές προσεγγίσεις εξισορρόπησης φόρτου. Η κεντρική εξισορρόπηση φόρτου μειώνει το χρόνο που χρειάζεται το δίκτυο να αναλύσει την κατάσταση του δικτύου, αλλά δημιουργεί μεγάλη καθυστέρηση στον κεντρικό κόμβο.

Επιπλέον το σύστημα δεν είναι ανεκτικό σε σφάλματα, αφού η πιθανότητα για να υπερφορτωθεί ο κεντρικός κόμβος είναι μεγάλη και η ανάκαμψη του είναι δύσκολη.

2. Κατανεμημένη εξισορρόπηση φόρτου: Κατά την κατανεμημένη εξισορρόπηση φόρτου, κανένας κόμβος δεν είναι αρμόδιος για την κατανομή φόρτου και το χρονοπρογραμματισμό. Κάθε κόμβος στο δίκτυο διατηρεί τοπική γνώση για να εξασφαλίσει αποδοτική κατανομή των διεργασιών σε στατικό περιβάλλον και ανακατανομή σε δυναμικό περιβάλλον. Σε σενάριο το οποίο εφαρμόζεται κατανεμημένη εξισορρόπηση φόρτου, το σύστημα είναι ανεκτικό σε λάθη και σχεδόν πάντα ισορροπημένο αφού κανένας κόμβος δεν υπερφορτώνεται για να λάβει αποφάσεις εξισορρόπησης φόρτου.

Ένας αλγόριθμος εμπνευσμένος από τη φύση περιγράφεται πιο κάτω, ο οποίος συγκρίνεται με άλλο αλγόριθμο εμπνευσμένο από τη φύση, τον Honey Bee Behaviour Based Load Balancing Algorithm. Και οι δύο αλγόριθμοι διαγράφουν τη βάση μιας κατανεμημένης εξισορρόπησης φόρτου σε υπολογιστικά νέφη.

Η βεβαιωμένη τυχαία δειγματοληψία είναι ακόμη μία μέθοδος κατανεμημένης εξισορρόπησης φόρτου.

3. Ιεραρχική εξισορρόπηση φόρτου: Η ιεραρχική εξισορρόπηση φόρτου εργασίας περιλαμβάνει διάφορα επίπεδα του υπολογιστικού νέφους κατά την λήψη αποφάσεων της. Τέτοιες τεχνικές συνήθως λειτουργούν με τον τρόπο του «master-slave». Αυτά μπορούν να μοντελοποιηθούν χρησιμοποιώντας δέντρα, στα οποία κάθε κόμβος είναι υπό τον έλεγχο του κόμβου-πατέρα του που βρίσκεται πιο πάνω. Οι κόμβοι-πατέρες καθώς γνωρίζουν την κατάσταση των κόμβων-παιδιών, παίρνουν τις κατάλληλες αποφάσεις εξισορρόπησης φόρτου.

4. Εξισορρόπηση φόρτου βασισμένη σε εξαρτήσεις διεργασιών: Εξαρτώμενες διεργασίες είναι εκείνες που εξαρτώνται από άλλες. Δηλαδή μπορούν να εκτελεσθούν, μόνο αν εκτελεσθούν άλλες συγκεκριμένες διεργασίες πρώτα, από τις οποίες και εξαρτώνται. Για αυτού του είδους εργασίες προαπαιτείται ο χρονοπρογραμματισμός όλων των εργασιών έτσι ώστε να προηγούνται οι εργασίες οι οποίες δεν έχουν εξαρτήσεις. Τέτοιες περιπτώσεις λύνονται με αλγορίθμους εξισορρόπησης φόρτου προγραμματισμού ροής εργασιών.

3.2.1 Αλγόριθμοι Κατανεμημένης Εξισορρόπησης Φόρτου στο Υπολογιστικό Νέφος

Γρήγορη προσαρμοστική μέθοδος εξισορρόπησης φόρτου εργασίας [6]: Για να υπολογιστεί ο τοπικός φόρτος εργασίας, ο κύριος φόρτος εργασίας επικεντρώνεται σε συγκεκριμένα κελιά και έτσι η διαδικασία προσαρμογής των κορυφών του δικτύου είναι πολύ μεγάλη.

Για την επίλυση αυτού του προβλήματος προτείνεται μια δομή δυαδικού δέντρου έτσι ώστε να διαιρεθεί η προσομοιωμένη περιοχή σε διάφορες υποπεριοχές. Στόχος αυτού είναι να εξισορροπηθεί ο φόρτος από τις τοπικές στις καθολικές περιοχές. Ανάλογα με τη διαφορά του φόρτου εργασίας στα διάφορα κελιά, λαμβάνονται οι διάφορες αποφάσεις. Η καθολική περιοχή κατανέμεται βάσει του δυαδικού δέντρου. Τα δέντρα αποτελούνται από κόμβους τριών τύπων: κόμβοι φύλλα, κόμβοι παιδί, κόμβοι γονέας.

Ο αλγόριθμος αυτός έχει γρηγορότερη ταχύτητα εξισορρόπησης, μικρότερο χρόνο απόκρισης και μικρότερο κόστος επικοινωνίας κατά τη διάρκεια της προσομοίωσης.

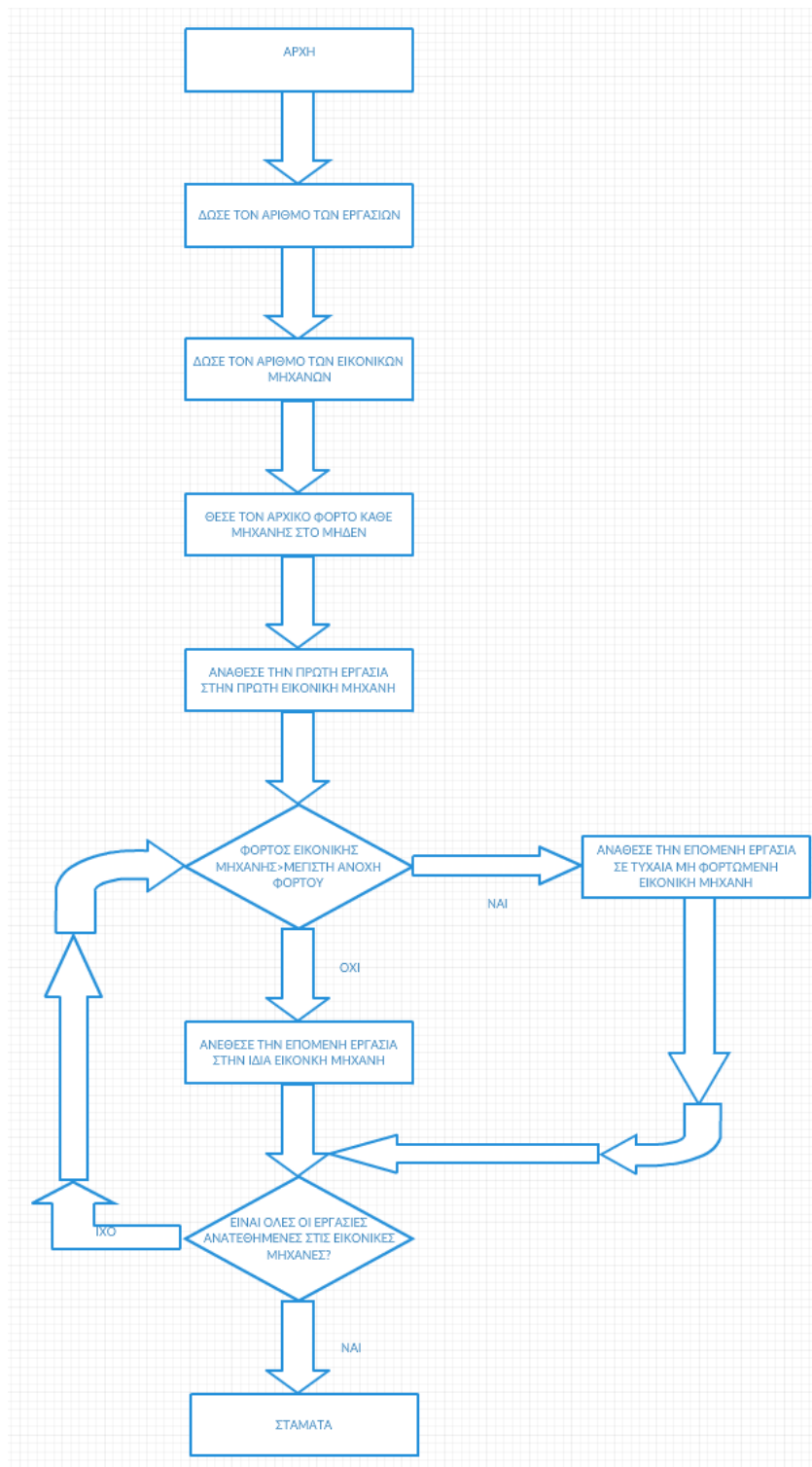
Δυναμική εξισορρόπηση φορτίου βασισμένη στη θερμική διάχυση [6]: Κατά τη θερμική διάχυση, δύο γειτονικά αντικείμενα, έχοντας διαφορά θερμότητας ‘εξισορροπούνται’ με το να μεταφέρει το πιο ζεστό αντικείμενο θερμότητα στο λιγότερο ζεστό. Ο αλγόριθμος αυτός θεωρεί το φόρτο ως ‘θερμότητα’ και τον αντιμετωπίζει με τον ίδιο τρόπο. Όλοι οι γειτονικοί κόμβοι μέσα στο ίδιο κελί μεταφέρουν το φόρτο τους από τον περισσότερο στο λιγότερο φορτωμένο κόμβο μέχρι οι κόμβοι να έχουν τον ίδιο φόρτο. Πρώτα γίνεται ο προγραμματισμός εξισορρόπησης φόρτου σε καθολικό και έπειτα σε τοπικό επίπεδο.

Συμπεριφορά μελισσών κατά την αναζήτηση τροφής [4]: Κατά την αναζήτηση τροφής μερικές μέλισσες ψάχνουν νέα μέρη με πηγές τροφής. Όταν βρουν τροφή οι ανιχνευτές πηγαίνουν στον τόπο αυτό με σκοπό να μελετήσουν την ποιότητα τροφής που βρέθηκε. Σε ένα ειδικό χώρο στη φωλιά των μελισσών οι μέλισσες που βρήκαν μια ποιοτική πηγή τροφής χορεύουν με σκοπό να ενημερώσουν τις άλλες μέλισσες για την ανακάλυψή τους. Μετά το πέρας του χορού ο ανιχνευτής επιστρέφει στο μέρος που βρέθηκε η τροφή για να δει αν υπάρχει περισσότερη. Εφόσον υπάρχει τροφή, ο ανιχνευτής θα ενημερώνει τη φωλιά για την τοποθεσία εκείνη. Τέλος, νέοι ανιχνευτές κατά την ένταξή τους στη φωλιά, ενημερώνουν τους υπόλοιπους με τον ίδιο χορό.

Προτεινόμενο διάγραμμα ροής: Το προτεινόμενο διάγραμμα στην Εικόνα 1 βασίζεται στη συμπεριφορά των μελισσών κατά τη διάρκεια αναζήτησης καλύτερης τοποθεσίας για τροφή.

Αρχίζει με την εισαγωγή αριθμού εργασιών στο σύστημα και έπειτα την εισαγωγή αριθμού εικονικών μηχανών. Μετά γίνεται η αρχικοποίηση του συστήματος, θέτοντας το φόρτο κάθε εικονικής μηχανής στο μηδέν. Αρχίζει η ανάθεση των εργασιών στις εικονικές μηχανές με την ανάθεση της πρώτης εργασίας στην πρώτη εικονική μηχανή. Μετά από την πρώτη ανάθεση και πριν από κάθε ανάθεση, γίνεται έλεγχος για το αν η εικονική μηχανή στην οποία έγινε η τελευταία ανάθεση είναι υπερφορτωμένη – δηλαδή ο φόρτος εργασίας που έχει ξεπερνά το μέγιστο φόρτο εργασιών που μπορεί να διαχειριστεί. Εάν δεν είναι

υπερφορτωμένη - δηλαδή ο φόρτος εργασίας που έχει δεν ξεπερνά το μέγιστο φόρτο εργασιών που μπορεί να διαχειριστεί –συνεχίζει με την εισαγωγή εργασιών. Εάν είναι υπερφορτωμένη - δηλαδή ο φόρτος εργασίας που έχει ξεπερνά το μέγιστο φόρτο εργασιών που μπορεί να διαχειριστεί, τότε απλά επιλέγεται τυχαία άλλη εικονική μηχανή η οποία δεν είναι υπερφορτωμένη και συνεχίζει με την εισαγωγή εργασιών μέχρι να μουν όλες στο σύστημα.



Εικόνα 1: Διάγραμμα Ροής αλγορίθμου εμπνευσμένου από τη συμπεριφορά των μελισσών κατά τη διάρκεια αναζήτησης καλύτερης τοποθεσίας για τροφή [4].

Παρ' όλο που και οι τέσσερις αλγόριθμοι μπορούν να εφαρμοστούν για να επιλύσουμε το πρόβλημα εξισορρόπησης φόρτου, επιλέξαμε να συγκρίνουμε το δικό μας αλγόριθμο με τον αλγόριθμο που χρησιμοποιούν οι μέλισσες κατά την αναζήτηση τροφής [4]. Και οι δύο αλγόριθμοι είναι εμπνευσμένοι από τη φύση, οπότε η σύγκριση θα έχει περισσότερο ενδιαφέρον. Επιπλέον και οι δύο αλγόριθμοι, σε συγκεκριμένα σημεία χρησιμοποιούν την τυχαιοποίηση για να πάρουν αποφάσεις. Ο αλγόριθμος εμπνευσμένος από τη συμπεριφορά των μελισσών επιλέγει τυχαία σε ποιόν κόμβο θα κατανέμει εργασίες μέχρι αυτός να γεμίσει, ενώ ο αλγόριθμος εμπνευσμένος από τη συμπεριφορά πουλιών σε κάθε εισαγωγή εργασίας επιλέγει τυχαία σε ποιόν κόμβο θα κατανεμηθεί η εργασία αυτή.

Κεφάλαιο 4

Πρόβλημα Διασκορπισμού Πουλιών

4.1 Περιγραφή Έρευνας	17
4.2 Σύγκριση Προβλήματος Εξισορρόπησης Φορτίου στο Υπολογιστικό Νέφος με το Πρόβλημα Διασκορπισμού των Πουλιών	19
4.3 Προτεινόμενος αλγόριθμος με βάση τη μετανάστευση των πουλιών	20

4.1 Περιγραφή Έρευνας

Χρησιμοποιήσαμε τη μελέτη που περιγράφεται στο [1] ως πηγή έμπνευσης των αλγορίθμων που ακολουθούν πιο κάτω. Χρησιμοποιείται το περιβάλλον ενός είδους πουλιού, wood thrush, του *Hylcocichla mustelina*. Οι εργασίες προσομοιώνονται στα πουλιά, και οι κόμβοι/εικονικές μηχανές στις διάφορες περιοχές.

Στη μελέτη αυτή υπάρχουν τρία είδη πουλιών wood thrush. Οι floaters, οι breeders, και οι juveniles. Οι πρώτοι είναι ενήλικες οι οποίοι δεν γεννούν αυγά και τείνουν προς τη μετανάστευση. Οι δεύτεροι, είναι ενήλικες που γεννούν αυγά και τείνουν να ζουν σε μια σταθερή περιοχή. Οι τρίτοι είναι ανήλικα πουλιά, τα οποία με το πέρασμα του χρόνου θα γίνουν είτε floaters, είτε breeders.

Αρχικά όλο το σύστημα αρχικοποιεί το μισό πληθυσμό με breeders, ο οποίος γεννά με κάποια Α αναλογία. Κάθε κάτοικος έχει πιθανότητα να πεθάνει την επόμενη χρονική στιγμή ανάλογα με την μακροζωία του. Οι floaters και οι juveniles έχουν μεγαλύτερη πιθανότητα να πεθάνουν παρά οι breeders.

Η διασπορά (ή διασκορπισμός) συμβαίνει όταν μια περιοχή έχει περισσότερα πουλιά από ότι πρέπει. Τότε λαμβάνοντας υπόψη τόσο την απόσταση που το πουλί μπορεί να διανύσει όσο και το μέγιστο αριθμό φορών που μπορεί να επιχειρήσει τη διασπορά, το πουλί αποφασίζει ποιά θα είναι η μόνιμη κατοικία του. Εάν αυτή η κατοικία έχει διαθέσιμο χώρο, τότε το πουλί μένει εκεί. Αν η κατοικία δεν έχει διαθέσιμο χώρο, τότε το πουλί θα επιχειρήσει τη διασπορά ξανά, εφ όσον δεν έχει ξεπεράσει το μέγιστο αριθμό φορών που μπορεί να διασκορπιστεί, ειδάλλως θα μετατραπεί σε floater. Οι breeders, ως πιο “πιστοί” στην κατοικία τους, έχουν μικρότερη πιθανότητα διασκορπισμού από ότι οι floaters και οι juveniles και έτσι τείνουν να

παραμένουν στην ίδια τοποθεσία και στον επόμενο χρόνο.

Όλα τα πουλιά, ανεξαρτήτως είδους, έχουν μια συγκεκριμένη ανοχή για το πόσο γεμάτος μπορεί να είναι ο τόπος διαμονής τους. Η διασπορά ενός πουλιού συμβαίνει όταν μία περιοχή έχει περισσότερα πουλιά από ό,τι πρέπει. Τότε, το πουλί αυτό μεταναστεύει σε άλλη περιοχή. Εάν αυτή η περιοχή το ικανοποιεί, τότε μένει εκεί, διαφορετικά μεταναστεύει ξανά σε άλλη περιοχή. Κάθε πουλί έχει όριο για την απόσταση την οποία μπορεί να διανύσει κάθε φορά, αλλά και όριο για τον αριθμό των μεταναστεύσεων που μπορεί να κάνει.

Ψευδοκώδικας:

1. Αρχικοποίησε το σύστημα με το μισό πληθυσμό.
2. Για κάθε κατοικία
3. {
4. Αν το πουλί είναι *breeder*
5. {
6. Παρήγαγε απόγονους
7. Θάνατος πουλιών
8. Έστω $K(\Pi)$ η κατοικία του πουλιού Π
9. Έστω $\Phi K(\Pi)$ ο φόρτος εργασίας του $K(\Pi)$
10. Έστω $\Pi(\Pi)$ η πιθανότητα μετανάστευσης του Π
11. Έστω $M(K)$ ο μέγιστος φόρτος της κατοικίας K
12. Αν $\Phi K(\Pi) > M(K)$ τότε
13. {
14. Αποφάσισε εάν θα διασκορπιστεί βάσει του $\Pi(\Pi)$
15. }
16. Εάν θα διασκορπιστεί και είναι *juvenile*
17. {
18. μετατροπή από *juvenile* σε *breeder*
19. }
20. }
21. Ενημέρωσε το όλο σύστημα με βάση τα νέα δεδομένα.
22. }

4.2 Σύγκριση Προβλήματος Εξισορρόπησης Φορτίου στο Υπολογιστικό Νέφος με το Πρόβλημα Διασκορπισμού των Πουλιών

Αντιπαραβάλλοντας το πρόβλημα των πουλιών διακρίνουμε ομοιότητες και διαφορές με το πρόβλημα εξισορρόπησης φόρτου.

Ομοιότητες μοντέλων

Και στα δύο μοντέλα, καθ' όλη τη διάρκεια ζωής τους οι περιοχές/κόμβοι πρέπει να είναι ισορροπημένοι, δηλαδή κάθε περιοχή να μην είναι συνωστισμένη και κάθε κόμβος να μην είναι υπερφορτωμένος.

Αντιστοιχία Χαρακτηριστικών

#	Πρόβλημα διασποράς πουλιών	Πρόβλημα υπολογιστικού νέφους
1	Πουλί	Εργασία
3	Φωλιά	Κόμβος/Εικονική μηχανή
4	Μέγεθος Περιοχής	Χωρητικότητα εικονικής μηχανής
5	Συνωστισμός φωλιάς	Υπερφόρτωση κόμβου
6	Μετανάστευση πουλιού	Μετακίνηση εργασίας σε άλλη εικονική μηχανή

Οι αντιστοιχίες είναι ξεκάθαρες, αφού βγαίνουν από μόνες τους, μέσα από τη δομή των δύο προβλημάτων.

Διαφορές μοντέλων

#	Πρόβλημα διασποράς πουλιών	Πρόβλημα υπολογιστικού νέφους
1	Νέα πουλιά δημιουργούνται μέσα από το σύστημα	Νέες εργασίες έρχονται έξω από το σύστημα.
2	Δεν υπάρχει έννοια του deadline	Έννοια του deadline
3	Υπάρχει όριο στον αριθμό διασποράς των πουλιών	Δεν υπάρχει όριο στον αριθμό μετακίνησης εργασίας
4	Υπάρχει η έννοια του θανάτου των πουλιών	Δεν υπάρχει η έννοια του θανάτου των εργασιών
5	Ένα πουλί τύπου juvenile μπορεί να μετατραπεί σε πουλί τύπου floater ή breeder	Μία εργασία δεν μπορεί να αλλάξει τύπο

Στο πρόβλημα του υπολογιστικού νέφους, οι εργασίες έρχονται όλες από έξω από το σύστημα. Ακόμη μια σημαντική διαφορά, είναι η ύπαρξη της έννοιας του deadline, που στο πρόβλημα διασποράς πουλιών δεν υπάρχει, η οποία παίζει καθοριστικό ρόλο στην αντιμετώπιση του προβλήματος. Το σύστημα πρέπει να λάβει υπόψη και αυτόν τον παράγοντα για να μπορέσει να εφαρμόσει σωστά την εξισορρόπηση φόρτου.

4.3 Προτεινόμενος αλγόριθμος με βάση τη μετανάστευση των πουλιών

Ο ψευδοκώδικας που δίδεται στο 4.2, αν και εφαρμόζεται με επιτυχία από τα πουλιά, δεν είναι εφαρμόσιμος από ένα υπολογιστικό νέφος ως έχει. Ένα υπολογιστικό νέφος δεν αρχικοποιείται όπως φαίνεται στην 1^η γραμμή, δεν «αλλάζουν είδος» όπως στην 6^η γραμμή, δεν υπάρχει «παραγωγή εργασιών» όπως γίνεται στην 11^η γραμμή. Ο πιο πάνω ψευδοκώδικας μπορεί να τροποποιηθεί έτσι ώστε να ταιριάζει περισσότερο σε περιβάλλον υπολογιστικού νεφους.

Ψευδοκώδικας:

1. Αρχικοποίησε το σύστημα με μηδεν πληθυσμό.
2. Εισήγαγε πουλιά στο σύστημα
3. Έστω $K(\Pi)$ η κατοικία του πουλιού Π
4. Έστω $\Phi K(\Pi)$ ο φόρτος εργασίας του K
5. Έστω $M(K)$ ο μέγιστος φόρτος της κατοικίας K
6. Αν $\Phi K(\Pi) > M(K)$ τότε
7. {
8. Αποφάσισε εάν θα διασκορπιστεί
9. }
10. Ενημέρωσε το όλο σύστημα με βάση τα νέα δεδομένα.

Κεφάλαιο 5

Υλοποίηση και πειραματική αξιολόγηση

5.1 Αλγόριθμος Εμπνευσμένος από τη Μετανάστευση των Πουλιών -Αλγόριθμος 1	21
5.2 Παραλλαγή Αλγορίθμου Honey Bee Behaviour Based Load Balancing of Tasks in Cloud Computing – Αλγόριθμος 2	25
5.3 Σύγκριση Αλγορίθμων 1 και 2	27
5.4 Σενάρια	28
5.4 Σχολιασμός Αποτελεσμάτων - Συμπεράσματα	115

Σε αυτή τη φάση της Διπλωματικής Εργασίας ασχολήθηκα με την πρακτική μελέτη του προβλήματος εξισορρόπησης φορτίου σε υπολογιστικά νέφη.

Προτείνονται δύο αλγόριθμοι για επίλυση αυτού του προβλήματος: Ο πρώτος αλγόριθμος αποτελεί παραλλαγή του αλγορίθμου του [3], ενώ ο δεύτερος αλγόριθμος είναι εμπνευσμένος από τον αλγόριθμο μετανάστευσης των πουλιών που μελετήσαμε στο προηγούμενο κεφάλαιο στον οποίο διαμορφώνεται η αρχικοποίηση των κόμβων και εισάγεται η έννοια του deadline.

5.1 Αλγόριθμος Εμπνευσμένος από τη Μετανάστευση των Πουλιών -Αλγόριθμος 1

Ο Αλγόριθμος 1, είναι εμπνευσμένος από τη διαδικασία που ακολουθούν τα πουλιά για να μεταναστεύσουν, όπως παρουσιάζεται στο άρθρο «Uncertainty in Spatially Explicit Population Models» [1].

Αρχικά ο γράφος είναι άδειος. Έπειτα ακολουθεί επαναληπτική διαδικασία. Επιλέγεται ένας κόμβος τυχαία, στον οποίο εισάγεται ένας αριθμός από εργασίες. Για κάθε εργασία που εισάγεται στο σύστημα: προσθέτουμε το φόρτο εργασίας στον κόμβο που της ανατέθηκε και η ίδια η εργασία τυγχάνει επεξεργασίας από τον κόμβο με βάση την επεξεργαστική του ισχύ. Μετά από κάθε εισαγωγή εργασίας, γίνεται έλεγχος κατά πόσο ο κόμβος βρίσκεται σε ισορροπία, αλλά και η κάθε εργασία είναι ικανοποιημένη στον κόμβο αυτό. Εάν δεν βρίσκεται σε ισορροπία, ή η εργασία δεν είναι ικανοποιημένη*, τότε ξεκινά η διαδικασία μετακίνησης των εργασιών “Dispersal” με σκοπό την αποφόρτωση του κόμβου, ή την ικανοποίηση της εργασίας.

*(Μία εργασία δεν είναι ικανοποιημένη εάν με βάση την υπολογιστική ταχύτητα του κόμβου σε συγκεκριμένο χρόνο θα χρειαστεί περισσότερο χρόνο να τύχει επεξεργασίας από ότι είναι το deadline της.)

Η διαδικασία του Dispersal έχει ως εξής:

Εάν η διαδικασία του Dispersal ενεργοποιήθηκε λόγω υπερφορτωμένου κόμβου, τότε επιλέγεται μία εργασία τυχαία. Εάν της το επιτρέψει η πιθανότητα μετακίνησής της, τότε βγαίνει από τον κόμβο και μπαίνει στη λίστα διασποράς. Ο κόμβος αποφορτώνεται φόρτο ίσο με το φόρτο της εργασίας που αφαιρέθηκε. Έπειτα γίνεται έλεγχος εάν ο κόμβος είναι ακόμη υπερφορτωμένος. Εάν ναι, τότε επαναλαμβάνεται η διαδικασία του dispersal. Εάν δεν είναι υπερφορτωμένος, τότε η διαδικασία του dispersal σταματά και το πρόγραμμα συνεχίζει με την κανονική του ροή. Στον επόμενο χρόνο, το πρόγραμμα ελέγχει κατά πόσο η λίστα διασποράς είναι άδεια. Αφού δεν είναι άδεια, θα εισαχθούν πρώτα όσες εργασίες βρίσκονται στη λίστα διασποράς.

Εάν η διαδικασία του dispersal ενεργοποιήθηκε από το γεγονός ότι μία εργασία δεν είναι ικανοποιημένη, τότε η ίδια εργασία επιλέγεται και ξεκινά τη διασπορά. Βγαίνει από τον κόμβο που βρίσκεται εκείνη τη χρονική στιγμή και μεταφέρεται σε άλλο τυχαία επιλεγμένο κόμβο. Εάν την ικανοποιεί ο νέος κόμβος, τότε μένει εκεί, διαφορετικά μεταφέρεται εκ νέου σε νέο κόμβο μέχρι να βρει κόμβο που να την ικανοποιεί, ή μέχρι να περάσει το deadline της και αναγκαστικά να μεταφερθεί στη λίστα χαμένων εργασιών.

```
struct job {  
    int load  
    int deadline  
    int arrival  
    float prob  
}
```

```
struct node {  
    int load  
    int maxload  
    int speed  
    struct job [] jobs  
}
```

Κλάση Job:

Η κλάση Job περιλαμβάνει στοιχεία μίας εργασίας όπως το load, που είναι ο φόρτος της εργασίας, το deadline, που είναι η μέγιστη χρονική διορία που η εργασία μπορεί να διεκπεραιωθεί, το arrival που είναι ο χρόνος εισαγωγής της εργασίας στο σύστημα και το prob, που είναι η πιθανότητα μιας εργασίας να μεταναστεύσει.

Κλάση Node:

Η κλάση Node περιλαμβάνει στοιχεία ενός κόμβου, όπως το load όπου αντιπροσωπεύει το φόρτο που έχει ένας κόμβος μία χρονική στιγμή, το maxload όπου αντιπροσωπεύει το μέγιστο φόρτο που μπορεί να εξυπηρετήσει ο κόμβος αυτός, το job [] το οποίο είναι η λίστα με τις εργασίες που εξυπηρετεί ο κόμβος και το speed το οποίο είναι η ταχύτητα με την οποία ο κόμβος επεξεργάζεται τις εργασίες.

```

ob_queue : jobs [max]
1. I=0
2. While (i<max) {
3. While (jobs_queue[i].arrival == Time) {
    i. n = choose_node[nodelist]
    ii. n.jobs = n.jobs U jobs_queue[i]
    iii. n.load=n.load+jobs_queue[i].load
    iv. i++
4. }
5. J=0
6. While (dispersal_queue != empty) {
    i. n = choose_node[nodelist]
    ii. n.jobs = n.jobs U dispersal_queue[j]
    iii. n.load=n.load+dispersal_queue[j].load
    iv. j++
7. }
8. }
9. for ( k=0; k< nodelist.length; k++) {
10.   if (nodelist[k] > nodelist[k].maxload)
        i. dispersal(nodelist[k]);
11.   if (nodelist[k].jobs[0].load/(nodelist[k].speed/
        nodelist[k].jobs.length))
        i. dispersal(nodelist[k].jobs[0], nodelist[k]);
12. }

13. RUN
14. For each node n {
15.   For each job j {
        i. j.load = j.load - (n.speed/n.jobs.length)
        i. j.deadline--;
        ii. if ((j.load > 0) && (j.deadline == 0)) {
        iii. n.jobs = n.jobs - {j};
        iv. lostJobsQueue = lostJobsQueue U {j};
        v. }
        vi. If (j.load == 0) {
        vii. n.jobs = n.jobs - {j};
        viii. finishedJobsQueue = finishedJobsQueue U {j};
        ix. }
16.   n.load = n.load - speed;
17. }
18. }
19. Time++;
20. }

21. Dispersal (node n)          (triggered by node n)
22. While (n.load>n.maxLoad){
23.   Random j = Random(n.joblist)
24.   Random p = Random()
        i. If (j.prob<p){
        ii. dispersalQueue=dispersalQueue U {j}
        iii. n.load=n.load-j.load
        iv. n.jobs = n.jobs - {j};
        v. j.deadline--;
        vi. }
25. }

```

```

26. Dispersal (Job j, Node n)      (triggered by Job j)
27. Node newNode=n;
28. While(j.deadline < j.load/(newNode.speed/newNode.jobs.length)) {
29.   newNode.load=newNode.load-j.load
30.   n.jobs = n.jobs - {j};
31.   Random n = Random(nodelist)
32.   newNode=n;
33.   newNode.load=newNode.load+j.load;
34.   newNode.jobs=newNode.jobs U {j}
35.   j.deadline--;
36. }
37.

```

Στη γραμμή 4 βλέπουμε την εισαγωγή εργασιών από τη λίστα εισαγωγής εργασιών και στη γραμμή 7 την εισαγωγή εργασιών από τη λίστα διασποράς. Στις γραμμές 10 με 13 γίνεται έλεγχος κατά πόσο οι κόμβοι είναι υπερφορτωμένοι έτσι ώστε να αρχίσει η διαδικασία διασποράς. Στις γραμμές 15 με 21 βλέπουμε τη διαδικασία αποφόρτωσης εργασιών ίσο με την επεξεργαστική ισχύ του κάθε κόμβου και τον τερματισμό των εργασιών εάν έχουν ξεπεράσει το χρονικό τους όριο. Στις γραμμές 22 με 26 βλέπουμε τη διαδικασία του Dispersal που ενεργοποιήθηκε λόγω υπερφορτωμένου κόμβου ενώ στις γραμμές 27 με 37 βλέπουμε τη διαδικασία του dispersal που ενεργοποιήθηκε από το γεγονός ότι μία εργασία δεν είναι ικανοποιημένη.

Παράδειγμα εκτέλεσης:

Το σύστημα μας αποτελείται από 10 κόμβους με υπολογιστική ταχύτητα 3 ο κάθε ένας (δηλαδή ανά κύκλο αφαιρεί φόρτο ίσο με 3/(αριθμό εργασιών) από κάθε εργασία). Υπάρχουν 50 εργασίες που πρέπει να εισαχθούν στο σύστημα για να εκτελεστούν. Η κάθε εργασία έχει φόρτο 5 και όριο εκτέλεσης 10 (δηλαδή μετά από 10 χρονικούς κύκλους εάν δεν εκτελεστεί, θα μεταφερθεί στη λίστα χαμένων εργασιών). Αρχικά ο γράφος είναι άδειος. Έπειτα ακολουθεί επαναληπτική διαδικασία. Επιλέγεται ο κόμβος 3 τυχαία, στον οποίο εισάγονται 3 εργασίες. Ο φόρτος του κόμβου 3 γίνεται 15. Μετά την εισαγωγή αυτών των εργασιών, γίνεται έλεγχος κατά πόσο ο κόμβος είναι υπερφορτωμένος, αλλά και η κάθε εργασία είναι ικανοποιημένη στον κόμβο αυτό. Οι εργασίες είναι ικανοποιημένες, αλλά ο κόμβος έχει υπερφορτωθεί, με αποτέλεσμα να ξεκινήσει η διαδικασία μετακίνησης των εργασιών “Dispersal” με σκοπό την αποφόρτωση του κόμβου. Τότε επιλέγεται η εργασία 2 τυχαία και η πιθανότητα μετακίνησής της, της επιτρέπει να διασκορπιστεί, βγαίνει από τον κόμβο και μπαίνει στη λίστα διασποράς. Ο κόμβος αποφορτώνεται φόρτο ίσο με το φόρτο της εργασίας που αφαιρέθηκε δηλαδή φόρτο ίσο με 5. Έπειτα γίνεται έλεγχος εάν ο κόμβος είναι ακόμη υπερφορτωμένος. Αφού τώρα έχει φόρτο ίσο με 10, το οποίο είναι και το όριο του κόμβου, η διαδικασία του dispersal σταματά και το πρόγραμμα συνεχίζει με την κανονική του ροή. Στον επόμενο χρόνο, το πρόγραμμα ελέγχει κατά πόσο η λίστα διασποράς είναι άδεια. Αφού δεν είναι άδεια, θα εισαχθεί πρώτα η εργασία 2.

5.2 Παραλλαγή Αλγορίθμου Honey Bee Behaviour Based Load Balancing of Tasks in Cloud Computing – Αλγόριθμος 2 [3]

Όπως έχουμε αναφέρει, ο πρώτος αλγόριθμος που θα μελετήσουμε, και τον οποίο αποκαλούμε Αλγόριθμο 2, αποτελεί παραλλαγή του αλγορίθμου που παρουσιάζεται στο [3].

Ο Αλγόριθμος, θεωρεί ένα άδειο δίκτυο από κόμβους το οποίο και γεμίζει μέχρι να εισαχθούν όλες οι εργασίες και ακολουθεί επαναληπτική διαδικασία έως ότου όλες οι εργασίες εκτελεσθούν. Κατά τη διάρκεια της επαναληπτικής διαδικασίας προστίθενται εργασίες σε εικονική μηχανή, μέχρι να υπερφορτωθεί. Όταν γίνει αυτό, δεν προστίθενται επιπλέον εργασίες στην ίδια εικονική μηχανή, αλλά επιλέγεται τυχαία μια άλλη.

Αρχικά ο γράφος είναι άδειος. Επιλέγεται ένας κόμβος ο οποίος είναι και ο κόμβος στον οποίο θα αρχίσει και η εισαγωγή εργασιών, defaultNode. Έπειτα ακολουθεί επαναληπτική διαδικασία στην οποία σε κάθε επανάληψη εισάγεται ένας αριθμός εργασιών. Για κάθε εργασία που εισάγεται στο σύστημα: προσθέτουμε το φόρτο εργασίας στον κόμβο που της ανατέθηκε και η ίδια η εργασία τυγχάνει επεξεργασίας από τον κόμβο με βάση την επεξεργαστική του ισχύ. Μετά από κάθε εισαγωγή εργασίας, γίνεται έλεγχος κατά πόσο ο κόμβος βρίσκεται σε ισορροπία. Εάν δεν βρίσκεται σε ισορροπία, τότε ο κόμβος defaultNode αλλάζει τυχαία, έτσι ώστε οι επόμενες εργασίες να μουν σε μη υπερφορτωμένο κόμβο.

Ψευδοκώδικας

```
struct job {  
  int load  
  int deadline  
  int arrival  
}
```

```
struct node {  
  int load  
  int maxload  
  int speed  
  struct job [] jobs  
}
```

Κλάση Job:

Η κλάση Job περιλαμβάνει στοιχεία μίας εργασίας όπως το load, που είναι ο φόρτος της εργασίας, το deadline, που είναι η μέγιστη χρονική διορία που η εργασία μπορεί να διεκπεραιωθεί και το arrival που είναι ο χρόνος εισαγωγής της εργασίας στο σύστημα.

Κλάση Node:

Η κλάση Node περιλαμβάνει στοιχεία ενός κόμβου, όπως το load όπου αντιπροσωπεύει το φόρτο που έχει ένας κόμβος μία χρονική στιγμή, το maxload όπου αντιπροσωπεύει το μέγιστο φόρτο που μπορεί να εξυπηρετήσει ο κόμβος αυτός, το job [] το οποίο είναι η λίστα με τις εργασίες που εξυπηρετεί ο κόμβος και το speed το οποίο είναι η ταχύτητα με την οποία ο κόμβος επεξεργάζεται τις εργασίες.

```
1. Job_queue : jobs [max]
2. I=0
3. While (i<max) {
4. While (jobs_queue[i].arrival == Time) {
    i. n = nodelist[0];
    ii. n.jobs = n.jobs U jobs_queue[i]
    iii. n.load=n.load+jobs_queue[i].load
    iv. i++;
    v. if (nodelist[k] > nodelist[k].maxload)
    vi. changeDefaultNode( );

5. }
6. }
7. CHANGE DEFAULT NODE:
8. Random r(nodelist.size());
9. defaultNode=r;

10. RUN:
11. For each node n {
12. For each job j {
13. j.load = j.load - (n.speed/n.jobs.length)
14. j.deadline--;
15. if ((j.load > 0) && (j.deadline == 0)) {
    i. n.jobs = n.jobs - {j};
    ii. lostJobsQueue = lostJobsQueue U {j};
16. }
17. If (j.load == 0) {
    i. n.jobs = n.jobs - {j};
    ii. finishedJobsQueue = finishedJobsQueue U {j};
18. }
19. n.load = n.load - speed;
20. }
21. }
22. Time++;
23. }
```

Στη γραμμή 4 βλέπουμε την εισαγωγή εργασιών από τη λίστα εισαγωγής εργασιών όπως και τον έλεγχο κατά πόσο υπάρχει υπερφόρτωση στους κόμβους.

Στις γραμμές 8 και 9 βλέπουμε την αλλαγή του κύριου κόμβου.

Στις γραμμές 11 με 23 βλέπουμε τη διαδικασία αποφόρτωσης εργασιών ίσο με την επεξεργαστική ισχύ του κάθε κόμβου και τον τερματισμό των εργασιών εάν έχουν ξεπεράσει το χρονικό τους όριο.

Για την υλοποίηση του προβλήματος και τη δημιουργία προσομοιώσεων προς εξαγωγή συμπερασμάτων, δημιουργήθηκαν οι παρακάτω αλγόριθμοι όπως έχουν περιγραφεί και παρουσιάζεται στο Κεφάλαιο 4.

Για την υλοποίηση των αλγορίθμων δημιουργήθηκαν οι ακόλουθες κλάσεις-αντικείμενα:

5.3 Σύγκριση Αλγορίθμων 1 και 2

Οι δύο αλγόριθμοι αρχικοποιούνται, εισάγουν και επεξεργάζονται εργασίες με τον ίδιο τρόπο. Η διαφορά τους έγκειται στη συμπεριφορά τους κατά την υπερφόρτωσή τους. Ο Αλγόριθμος 1, εμπνευσμένος από τη συμπεριφορά των πουλιών, αφού ανιχνεύσει υπερφόρτωση σε κόμβο, ή βρει πως μία εργασία δεν είναι ικανοποιημένη από τον κόμβο που βρίσκεται, τότε αρχίζει διαδικασία διασποράς, όπως περιγράφηκε πιο πάνω. Ο Αλγόριθμος 2, εμπνευσμένος από τη συμπεριφορά μελισσών, μόλις ανιχνεύσει υπερφόρτωση σε κόμβο, τότε απλά σταματά να εισάγει εργασίες στον κόμβο αυτό, και αλλάζει κόμβο.

Η κύρια διαφορά των δύο αλγορίθμων και το γεγονός που καθιστά τον Αλγόριθμο 1 ξεχωριστό από κάθε άλλο αλγόριθμο, είναι η χρήση πιθανοτήτων για την αντιμετώπιση της υπερφόρτωσης ενός κόμβου. Επιλέγοντας τυχαία μία εργασία από τον κόμβο, αποφασίζει με αυτή την πιθανότητα αν ο κόμβος θα μετακινηθεί, ή αν θα παραμείνει.

5.3.1 Δημιουργία Συντελεστή Σεναρίων

Ο συντελεστής σεναρίων θα καθορίζει πόσο ελαφρά ή βαρετά φορτωμένα θα είναι τα σενάρια αξιολόγησης του αλγορίθμου μας. Θα ορίζεται από ένα πηλίκο με βάση τα εξής χαρακτηριστικά: την ταχύτητα επεξεργασίας του κόμβου, τον αριθμό των κόμβων, το φόρτο κάθε εργασίας και τον αριθμό εργασιών. Ο διαιρέτης του πηλίκου αυτού θα αποτελείται από το γινόμενο των συντελεστών οι οποίοι επιβαρύνουν το όλο σύστημα, δηλαδή του φόρτου που έχει η κάθε εργασία με τον αριθμό των εργασιών που βρίσκονται στο σύστημα, ενώ ο διαιρέτης θα είναι το γινόμενο των συντελεστών οι οποίοι ελαφραίνουν το όλο σύστημα, δηλαδή η ταχύτητα επεξεργασίας του κάθε κόμβου με τον αριθμό των κόμβων.

L = συνολικός φόρτος εργασιών

Ο συνολικός φόρτος εργασιών είναι το άθροισμα του φόρτου κάθε εργασίας

S = συνολική υπολογιστική ταχύτητα

Η συνολική υπολογιστική ταχύτητα, είναι το άθροισμα της υπολογιστικής ταχύτητας κάθε κόμβου.

W = παράθυρο εισαγωγής εργασιών

Το παραθυρο εισαγωγής εργασιών, είναι το χρονικό όριο που μπορούν οι εργασίες να εισαχθούν στο σύστημα.

N = μέγιστο φορτίο κόμβων

Το μέγιστο φορτίο κόμβων είναι ο μέγιστος φόρτος που μπορεί να ανεχτεί ένας κόμβος

Συντελεστής:

$$\frac{L}{W \times N \times S}$$

Με βάση κάποια όρια του συντελεστή θα καθορίζονται τα σενάρια ως ελαφρά, μέτρια ή βαριά φορτωμένα.

Για παράδειγμα, σε ένα σενάριο όπου ο φόρτος κάθε εργασίας είναι 5 και ο αριθμός εργασιών είναι 50 ο συνολικός φόρτος είναι 250. Εάν έχουμε 10 κόμβους και ο κάθε κόμβος έχει 3 υπολογιστική ταχύτητα, τότε η συνολική υπολογιστική ταχύτητα του δικτύου είναι 30. Έχοντας το παράθυρο εισαγωγής εργασιών και το μέγιστο φορτίο κόμβων ίσο με 10, τότε ο συντελεστής φόρτου έχει ως εξής: $\frac{250}{30 \times 10 \times 10}$ και ισούται με 0.0833.

5.4 Σενάρια

Σκοπός της εξισορρόπησης φόρτου εργασίας στο υπολογιστικό νέφος, όπως έχει προαναφερθεί, είναι η εκτέλεση των εργασιών, ελαχιστοποιώντας το χρόνο εκτέλεσης, όπως και τον αριθμό απώλειας εργασιών.

Η κυριότερη διαφορά μεταξύ των δύο αλγορίθμων είναι η χρήση πιθανοτήτων από τον Αλγόριθμο 1. Όπως στον Αλγόριθμο των πουλιών [1] έτσι και στον Αλγόριθμο 1 οι εργασίες μεταναστεύουν κατά τη εκτέλεση του αλγορίθμου όταν δεν είναι ικανοποιημένες. Δηλαδή εάν με βάση την υπολογιστική ταχύτητα του κόμβου σε μια συγκεκριμένη στιγμή θα χρειαστεί περισσότερο χρόνο να τύχει επεξεργασίας από ότι είναι το deadline της. Η πιθανότητα που επιλέχθηκε στον αλγόριθμο των πουλιών [1] είναι η 0.37. Εμείς στον Αλγόριθμο 1 χρησιμοποιήσαμε τις πιθανότητες 0.2, 0.35, 0.37, 0.4, 0.6 .

Πρωτού συγκριθούν οι δύο αλγόριθμοι μεταξύ τους, πειραματιστήκαμε με τις πιθανότητες που χρησιμοποιά ο Αλγόριθμος 1 κατά τη διαδικασία διασποράς εργασιών. Βρήκαμε τη βέλτιστη σε κάθε σενάριο και έπειτα τη συγκρίναμε με τα αποτελέσματα του Αλγορίθμου 2.

Όλες οι εκτελέσεις έτρεξαν σε δίκτυο όπου ο κάθε κόμβος είναι συνδεδεμένος με όλους τους κόμβους γύρω του. Αυτό έγινε λόγω της υπερ συνδεσιμότητας των υπολογιστικών νεφών.

Πριν από τα αποτελέσματα – γραφικές παραστάσεις κάθε σεναρίου (προσομοίωσης) καταγράφονται σε πίνακα οι μεταβλητές οι οποίες παραμένουν σταθερές. Ως αποτέλεσμα, στις γραφικές παραστάσεις βλέπουμε τον αριθμό των χαμένων εργασιών κάθε σεναρίου, αφού ο χρόνος που χρειάστηκαν τα σενάρια είναι σε όλα ο ίδιος.

Οι πιθανότητες σε κάθε γραφική αναγράφονται με βάση το 10.

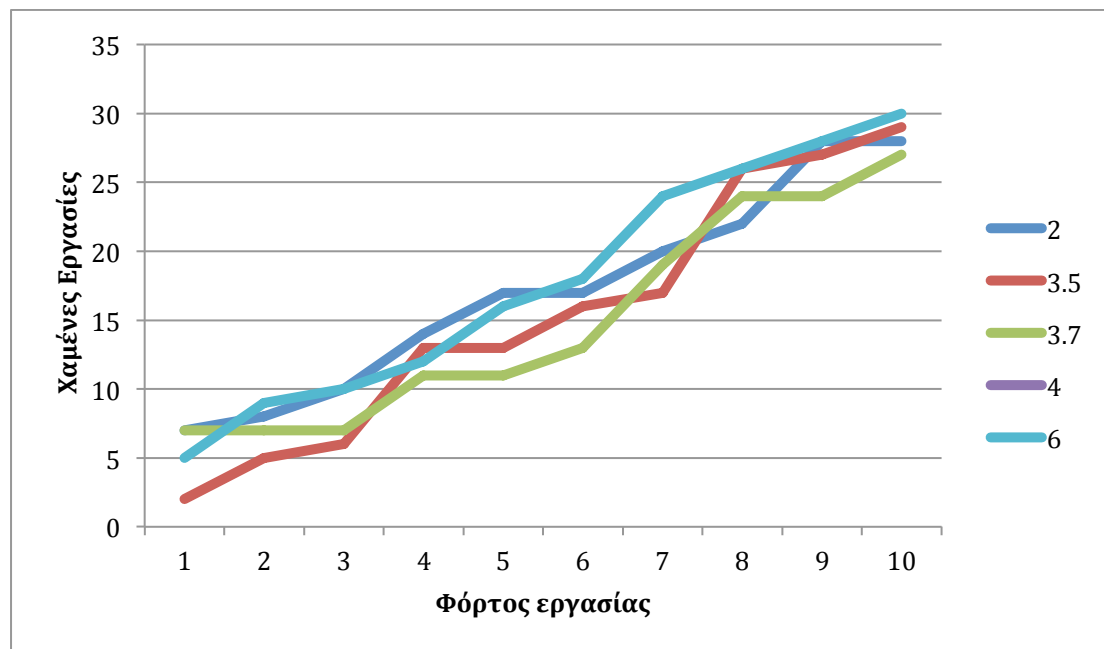
5.4.1 Ελαφρά Φορτωμένα Σενάρια

Τα ελαφρά φορτωμένα σενάρια έχουν συντελεστή μέχρι και 0.416

Παρουσιάζονται πρώτα οι εκτελέσεις με τον Αλγόριθμο 1.

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
50	3	10	10	10



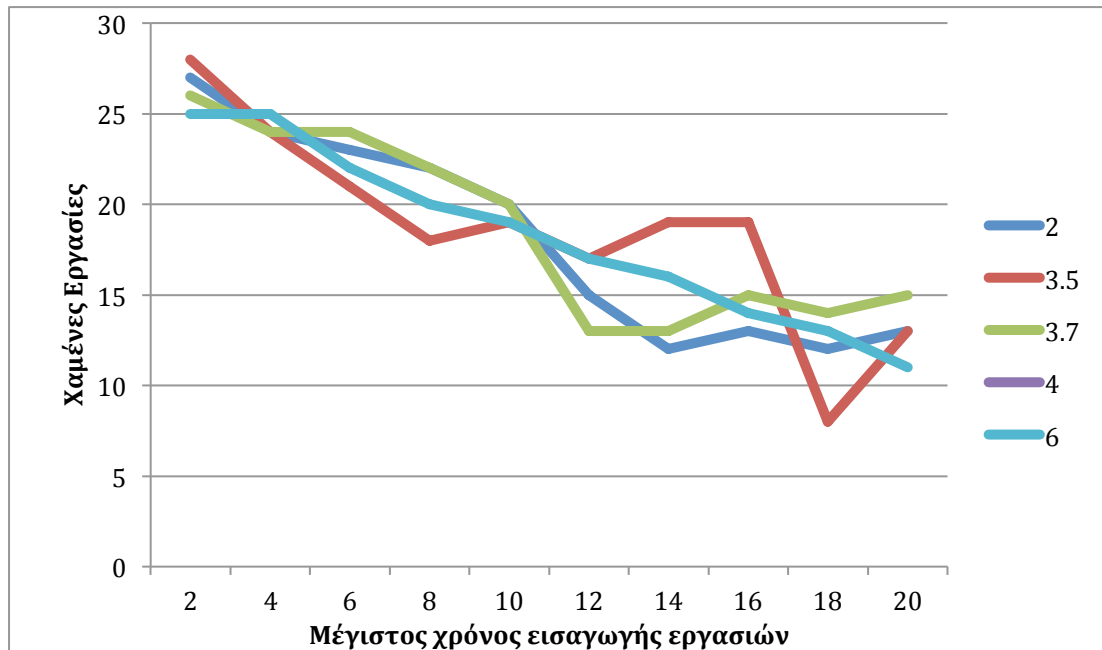
Μέγιστος Φόρτος: 0.166

Χαμένες: 4% - 60%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως ο αριθμός χαμένων εργασιών αυξάνεται σταδιακά από 2 εργασίες σε 30 και άρα το ποσοστό χαμένων εργασιών κυμαίνεται από 4% μέχρι 60%. Η πιθανότητα που έφερε τα καλύτερα αποτελέσματα είναι η 3.7.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
50	3	10	10	5



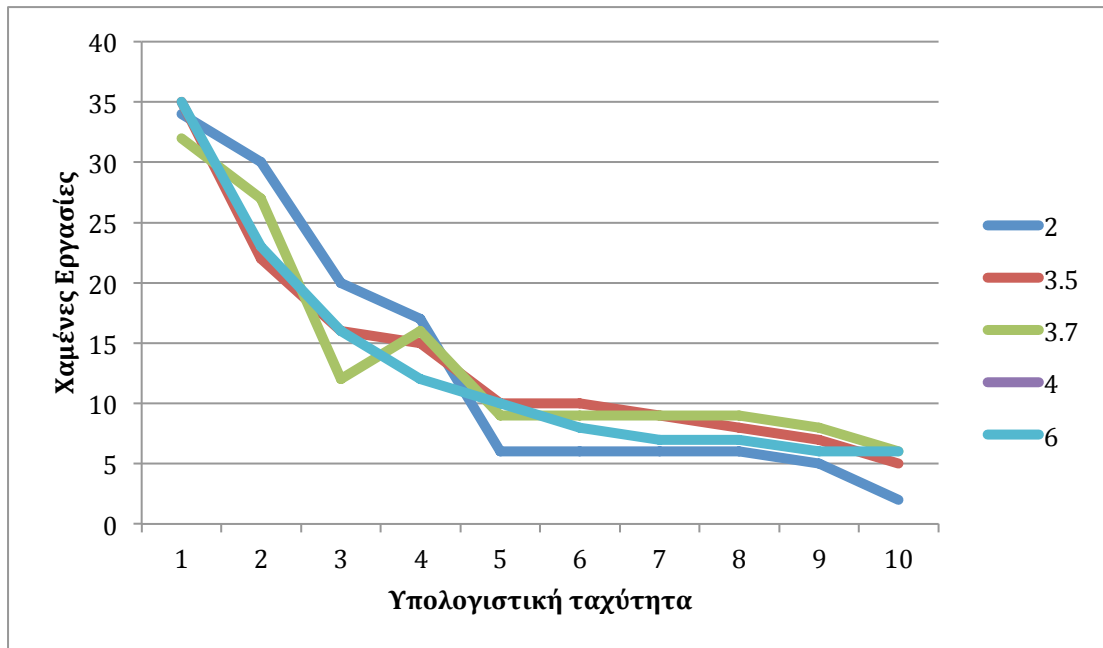
Μέγιστος Φόρτος: 0.416

Χαμένες: 14% - 54%

Σε αυτό το σενάριο μεταβάλαμε το μέγιστο χρονικό όριο εισδοχής εργασιών. Το όριο κυμαίνεται από 2 μέχρι 20. Όπως περιμέναμε, καθώς αυξάνεται το όριο εισδοχής εργασιών, ο αριθμός χαμένων εργασιών μειώνεται, αφού οι εργασίες εισέρχονται πιο αραιά και το σύστημα μπορεί να τις χειριστεί καλύτερα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 14% μέχρι 54%. Οι πέντε πιθανότητες συμπεριφέρθηκαν περίπου το ίδιο.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
50	5	10	10	10



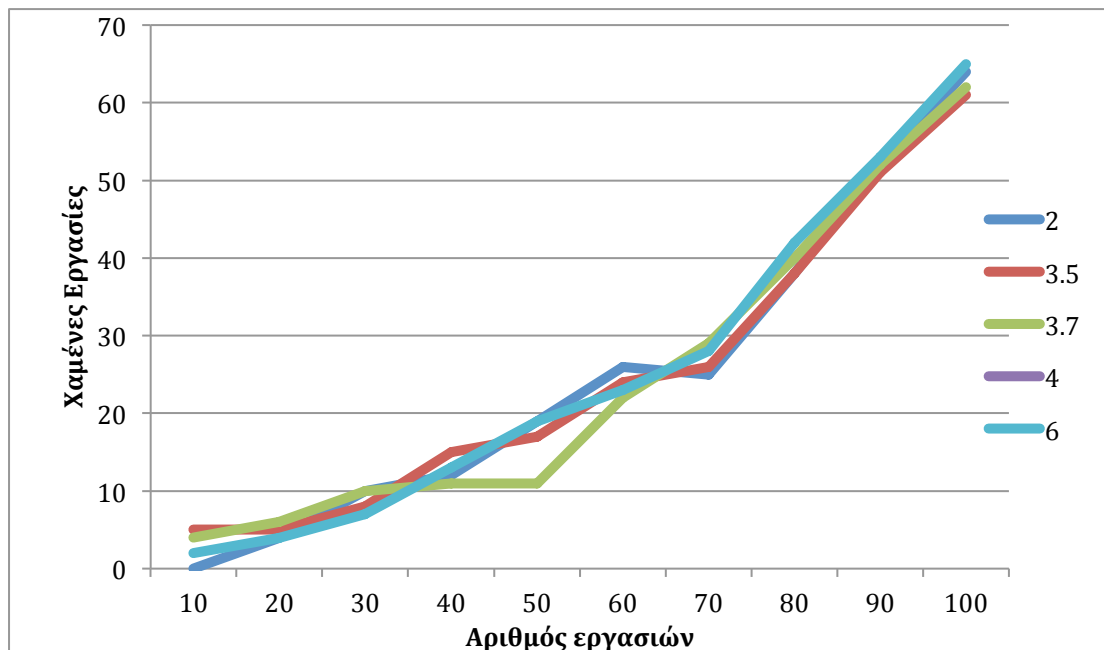
Μέγιστος Φόρτος: 0.25

Χαμένες: 4% - 70%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 4% μέχρι 70%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με μικρή υπολογιστική ταχύτητα οι μεγάλες πιθανότητες θα είχαν καλύτερα αποτελέσματα, αφού θα επέτρεπαν καλύτερη κατανομή των εργασιών σε εκτελέσεις με μικρή υπολογιστική ταχύτητα, αλλά και το αντίθετο οι μικρές πιθανότητες θα έπρεπε να είχαν καλύτερα αποτελέσματα στις εκτελέσεις με μεγάλη υπολογιστική ταχύτητα, αφού οι κόμβοι θα είχαν την δυνατότητα να χειριστούν ευκολότερα τις εργασίες χωρίς κάποια κατανομή. Παρόλα αυτά όλες οι πιθανότητες είχαν περίπου ίδια αποτελέσματα.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



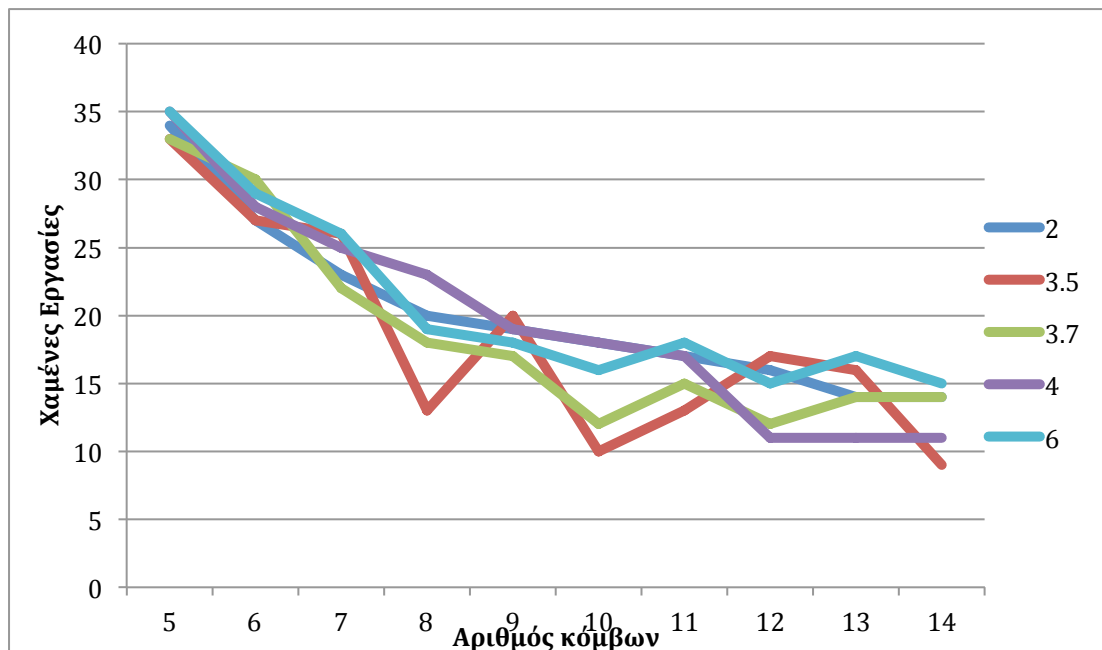
Μέγιστος Φόρτος: 0.166

Χαμένες: 0% - 65%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 10 εργασίες στο σύστημα, τελειώνοντας στις 100 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 0% μέχρι 65%. Οι πιθανότητες και σε αυτό το σενάριο συμπεριφέρθηκαν το ίδιο.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
50	5	3	10	10



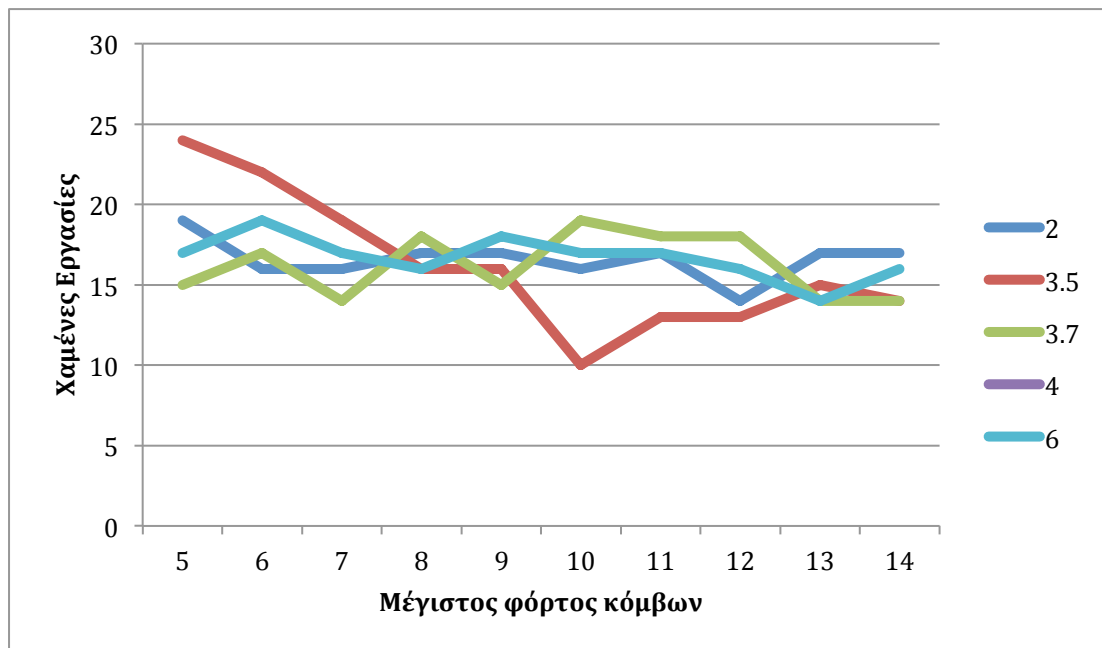
Μέγιστος Φόρτος 0.166

Χαμένες: 18% - 70%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 18% μέχρι 70%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με λιγότερους κόμβους, οι μεγάλες πιθανότητες (4,6) θα είχαν καλύτερα αποτελέσματα, αφού θα βελτιώναν την κατανομή των εργασιών στους κόμβους. Οι πιθανότητες όμως, συμπεριφέρθηκαν με τον ίδιο τρόπο σε όλες τις εκτελέσεις.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
50	5	3	10	10



Μέγιστος Φόρτος: 0.166

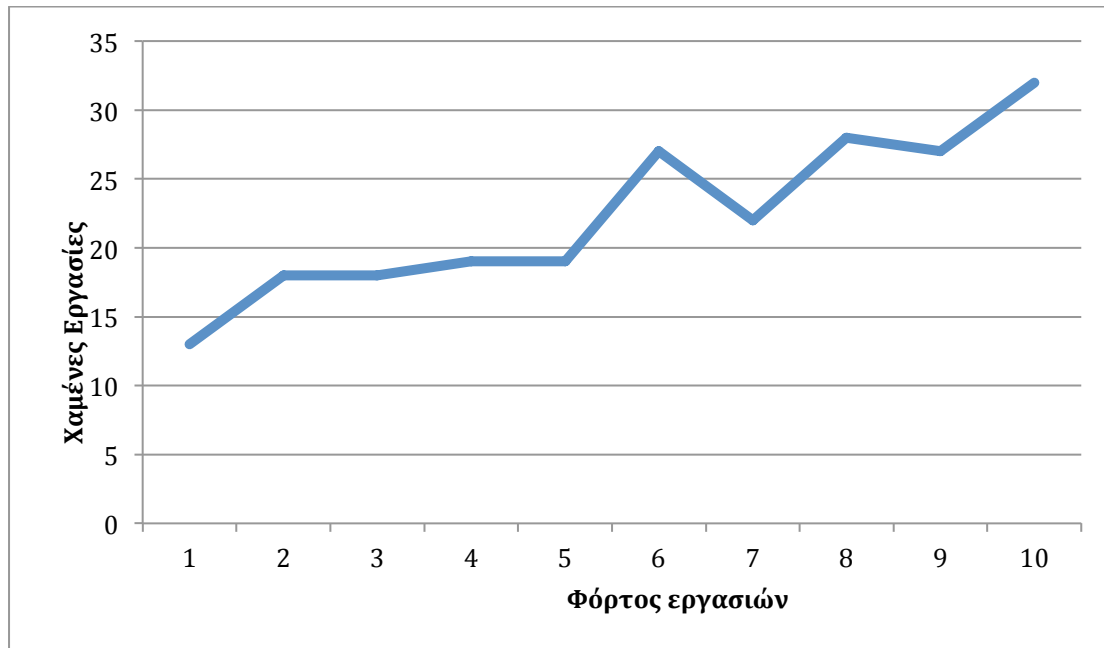
Χαμένες Εργασίες: 20% - 48%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών, ούτε οι διαφορετικές πιθανότητες στον αλγόριθμο έπαιξαν κάποιο ρόλο. Το ποσοστό χαμένων εργασιών κυμαίνεται από 20% μέχρι 48%.

Παρουσιάζονται οι εκτελέσεις με τον Αλγόριθμο 2

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
50	3	10	10	10



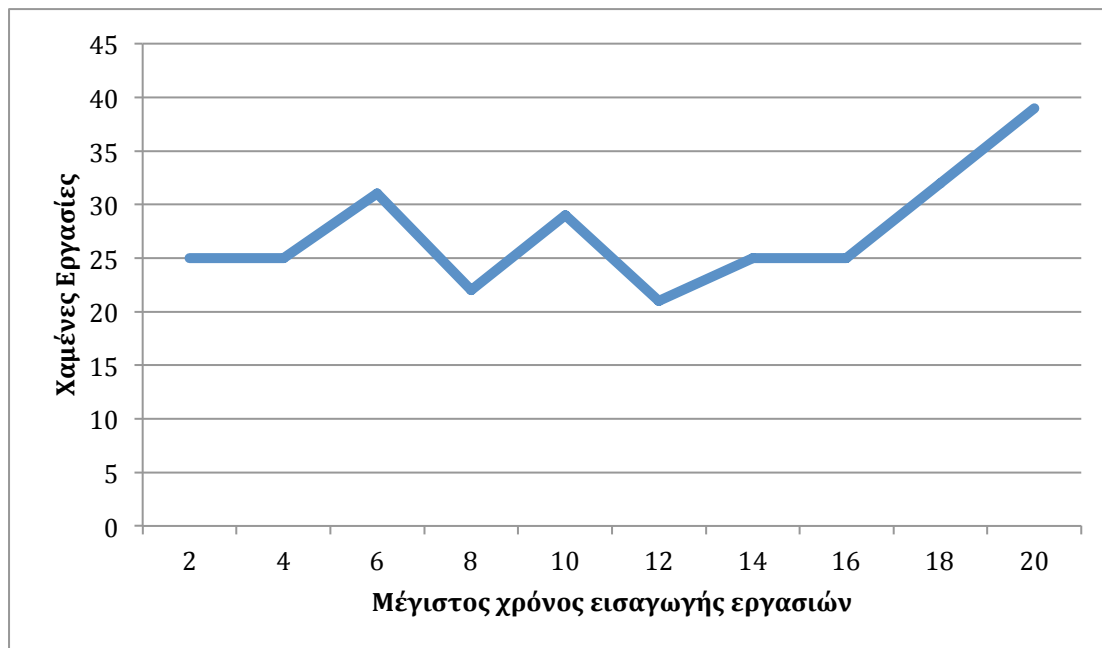
Μέγιστος Φόρτος: 0.166

Χαμένες Εργασίες: 26% - 64%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται ο φόρτος εργασίας, αυξάνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26% - 64%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
50	3	10	10	5



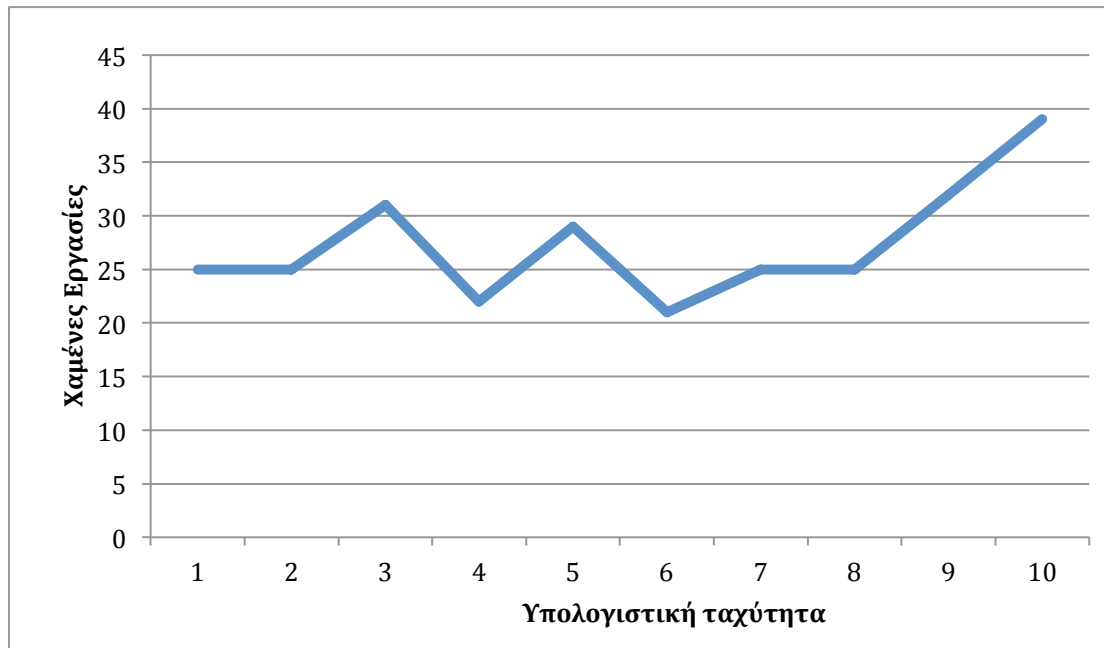
Μέγιστος Φόρτος: 0.416

Χαμένες Εργασίες: 42% - 78%

Σε αυτό το σενάριο αυξάνεται το μέγιστο όριο εισδοχής εργασιών, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται το όριο εισδοχής εργασιών, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 42% - 78%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
50	5	10	10	10



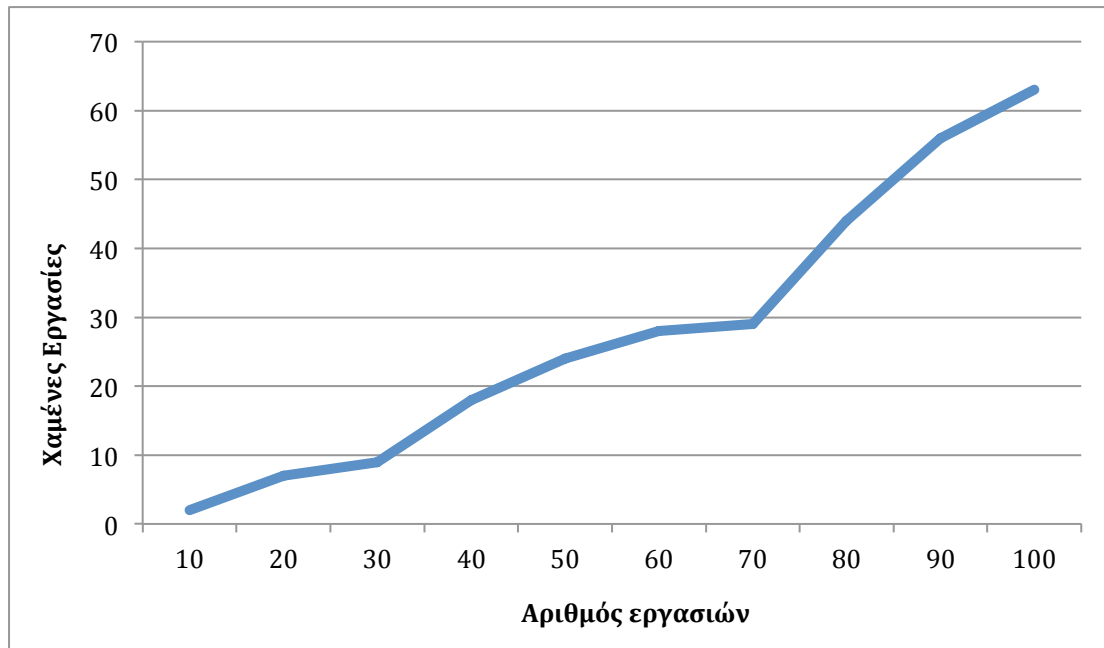
Μέγιστος Φόρτος: 0.25

Χαμένες Εργασίες: 42% - 78%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 42% μέχρι 78%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



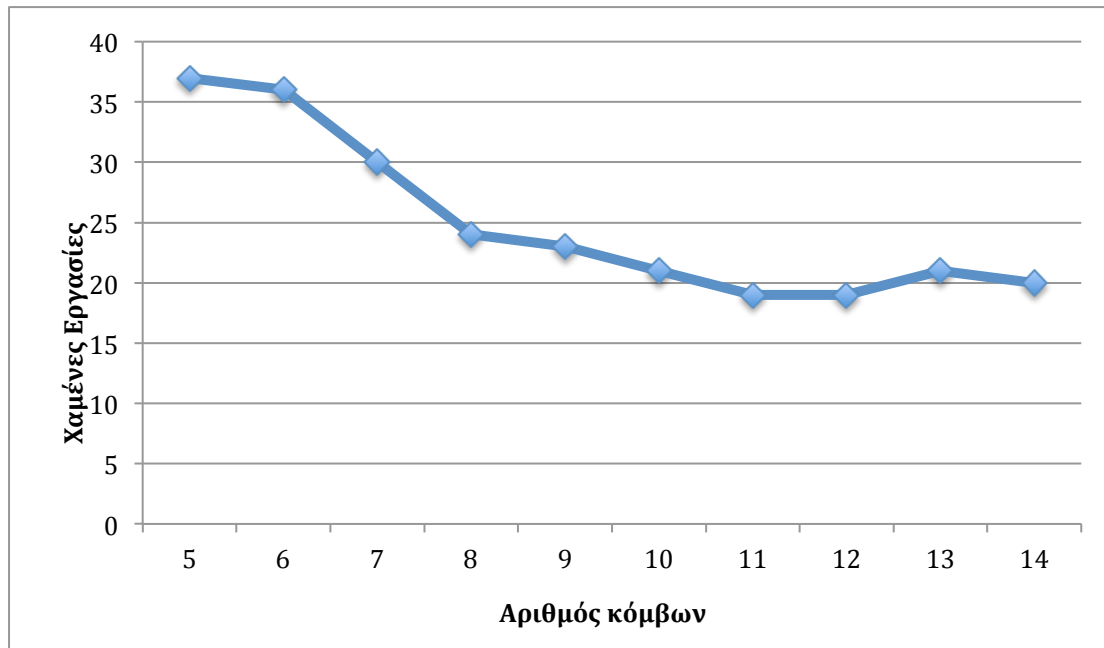
Μέγιστος Φόρτος: 0.166

Χαμένες Εργασίες: 20% - 63%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 10 εργασίες στο σύστημα, τελειώνοντας στις 100 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 20% μέχρι 63%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
50	5	3	10	10



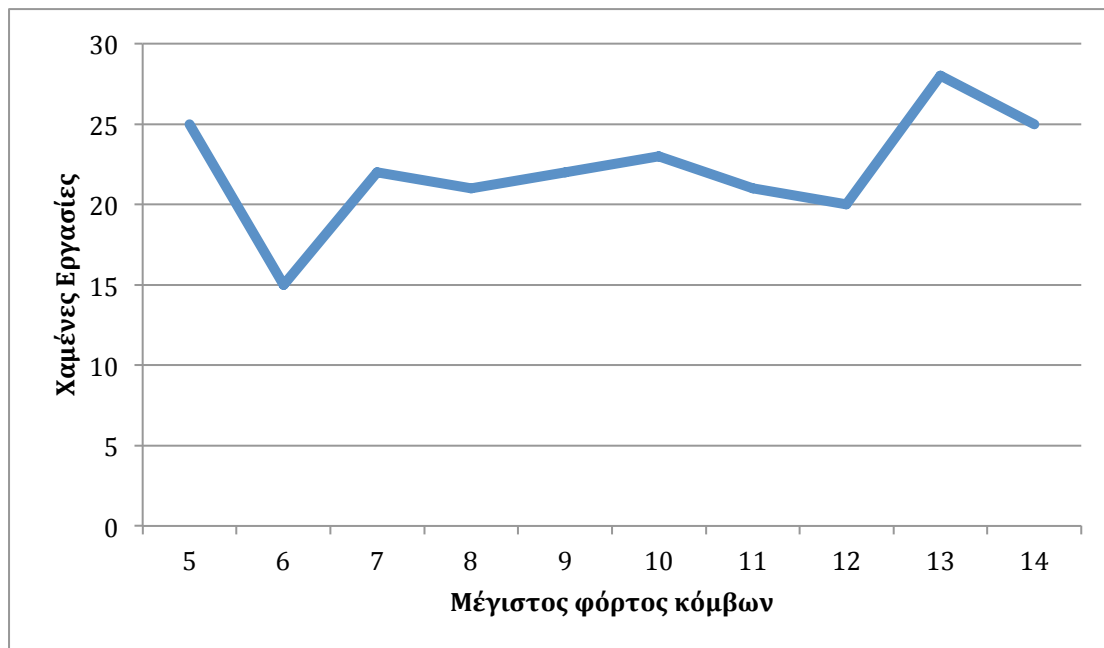
Μέγιστος Φόρτος: 0.166

Χαμένες Εργασίες: 38% - 68%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 38% - 68%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
50	5	3	10	10



Μέγιστος Φόρτος: 0.166

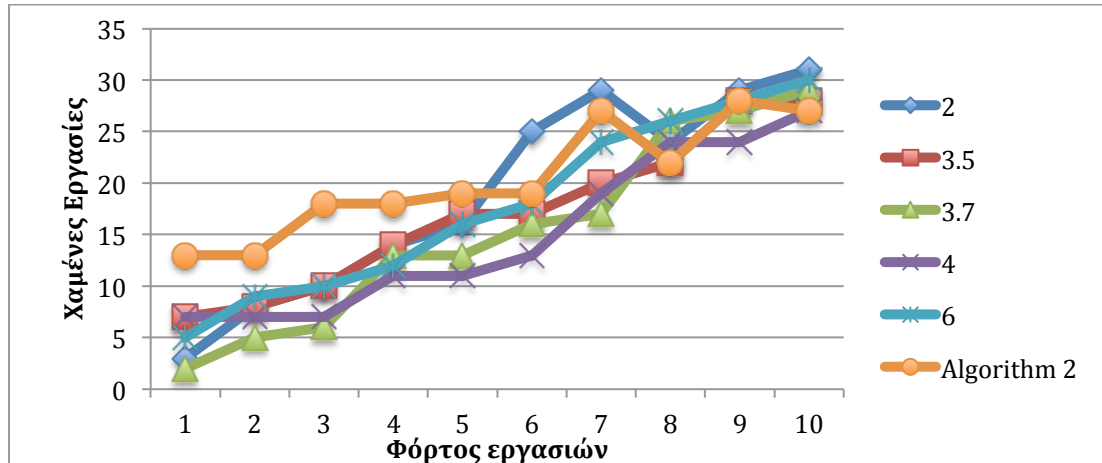
Χαμένες Εργασίες: 30% - 56%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 30% - 56%.

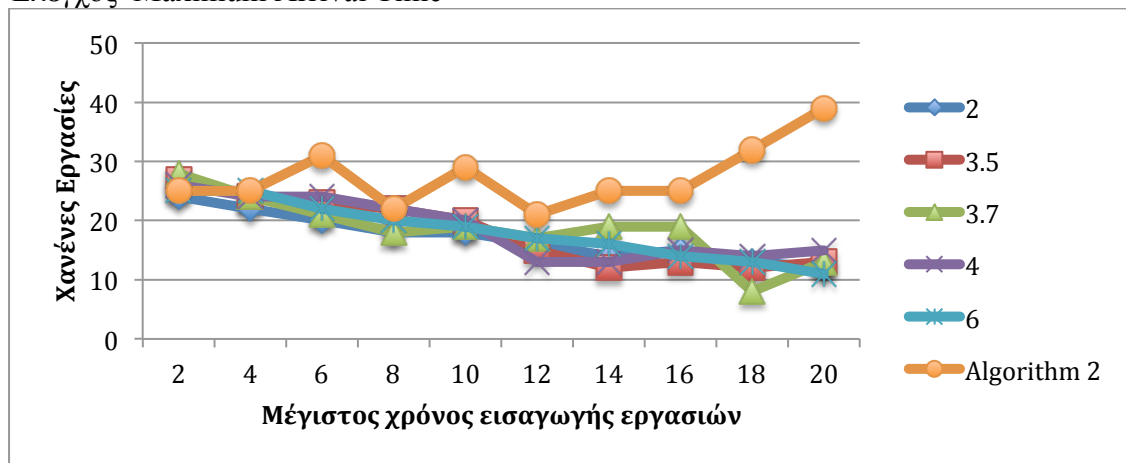
Σύγκριση Αλγορίθμων

Οι δύο αλγόριθμοι κινήθηκαν στα ίδια πλαίσια όσον αφορά το χρόνο εκτέλεσης των εργασιών. Η διαφορά τους έγκειται στον αριθμό χαμένων εργασιών, η οποία φαίνεται καλύτερα εάν αντιπαραβάλουμε τα αποτελέσματα των δύο αλγορίθμων μεταξύ τους.

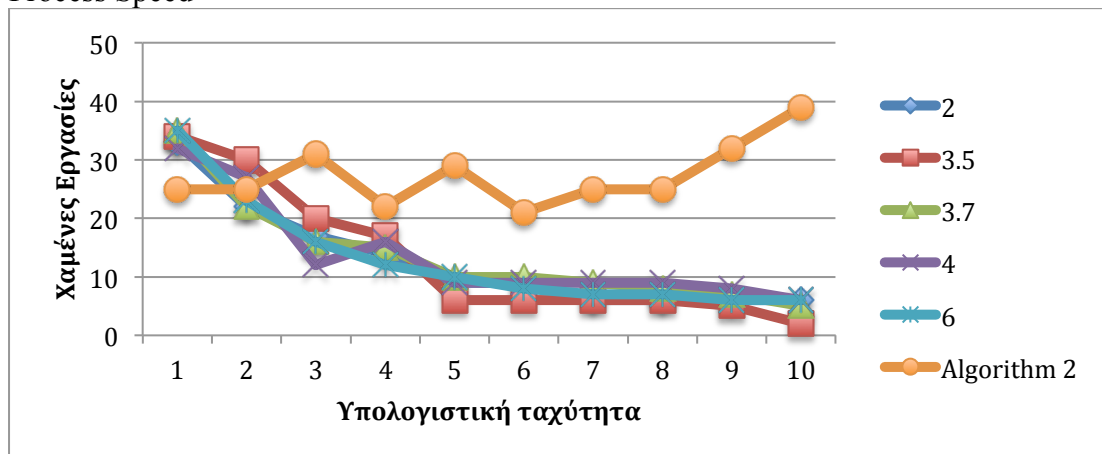
Έλεγχος Job Load



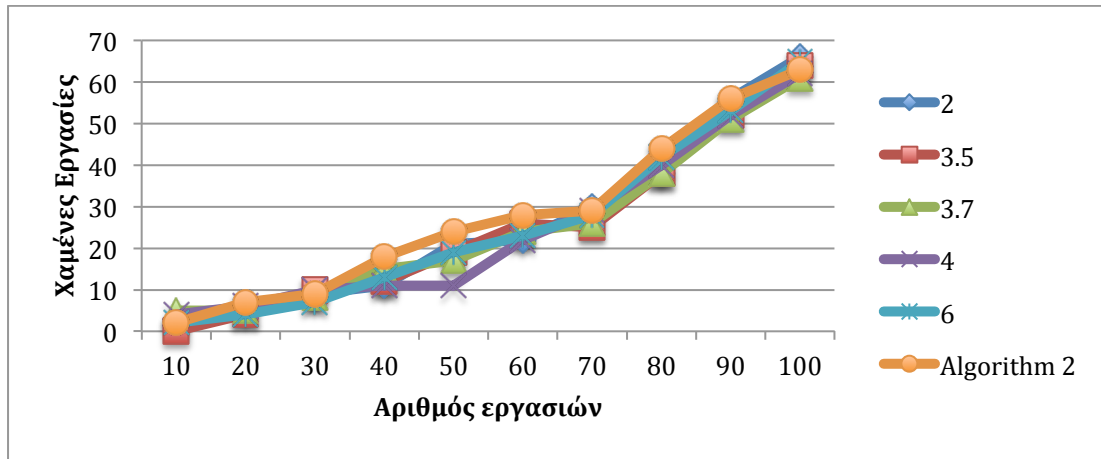
Έλεγχος Maximum Arrival Time



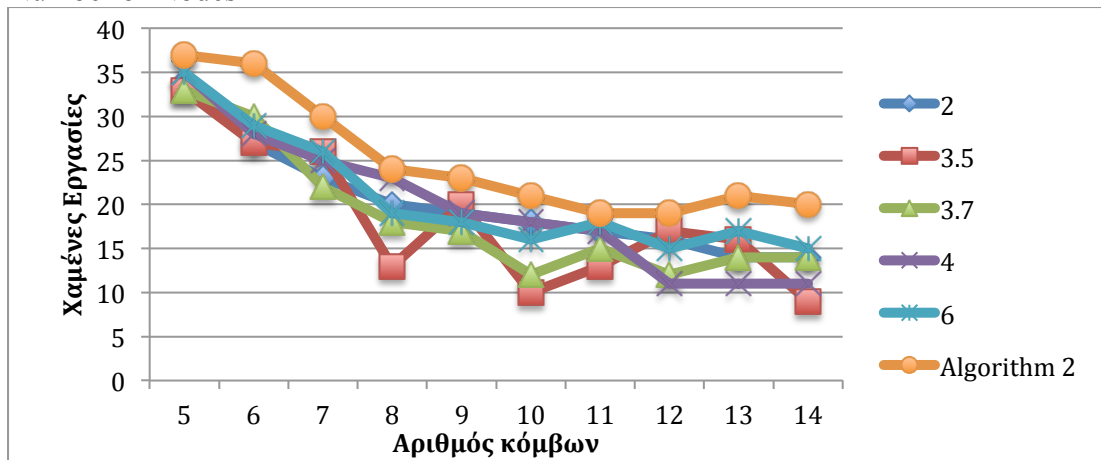
Process Speed



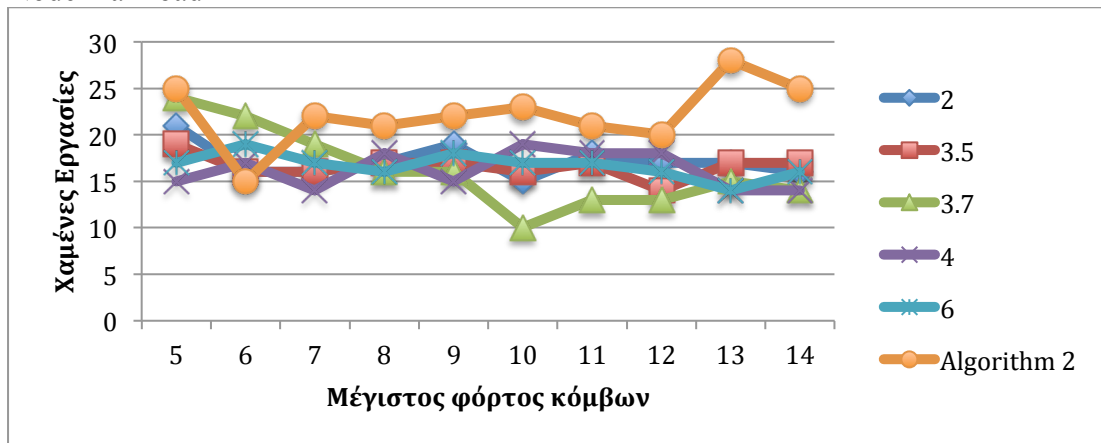
Number of Jobs



Number of Nodes



Node max load



Όπως φαίνεται καθαρά από τις γραφικές παραστάσεις, ο Αλγόριθμος 1 υπερτερεί. Όλες οι πιθανότητες του Αλγορίθμου 1 φέρουν καλύτερα αποτελέσματα εαν τις συγκρίνουμε με τα αποτελέσματα του Αλγορίθμου 2.

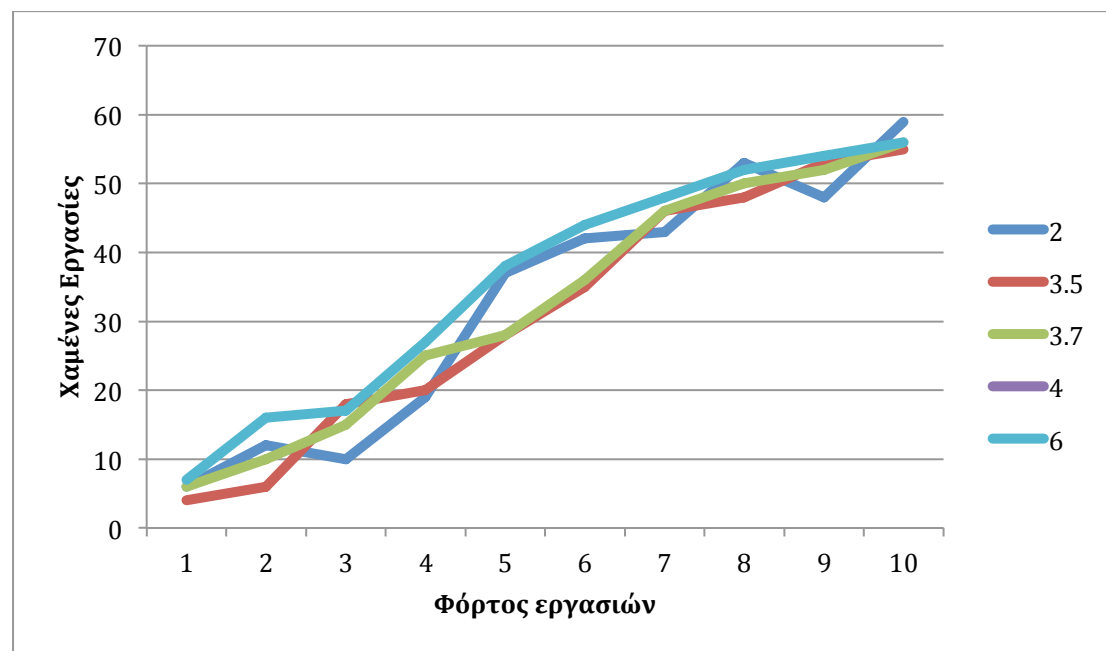
5.4.2 Μέτρια Φορτωμένα Σενάρια

Τα μέτρια φορτωμένα σενάρια έχουν συντελεστή μέχρι και 0.624.

Παρουσιάζονται πρώτα οι εκτελέσεις με τον Αλγόριθμο 1.

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
75	3	10	10	10



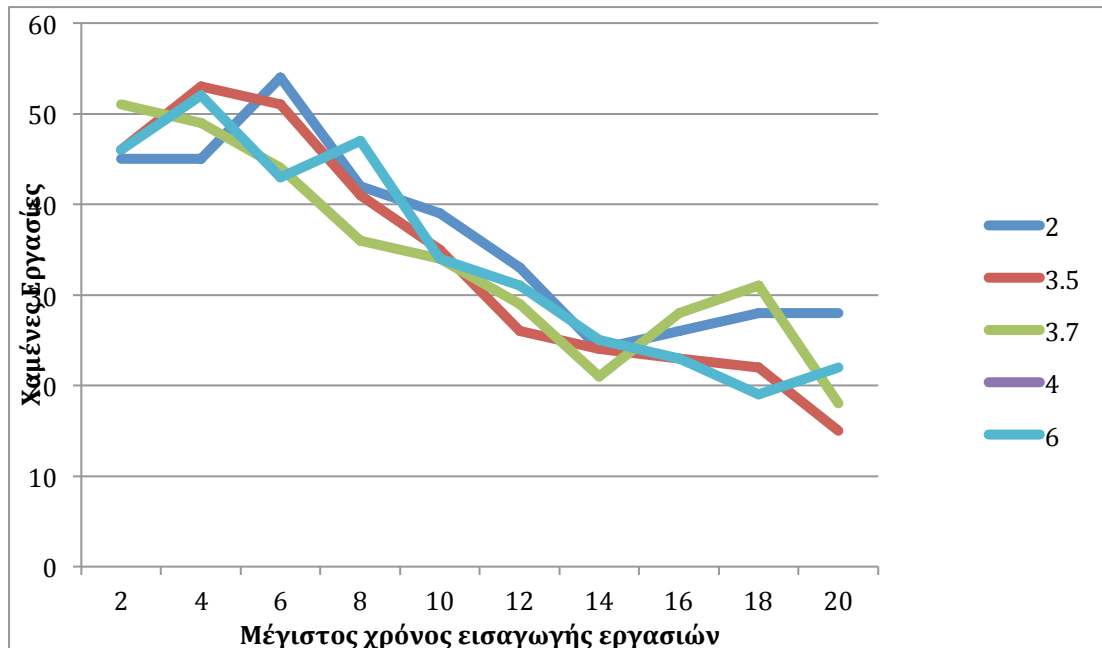
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 12% - 78%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως ο αριθμός χαμένων εργασιών αυξάνεται σταδιακά. Το ποσοστό χαμένων εργασιών κυμαίνεται από 4% μέχρι 60%. Η πιθανότητα που έφερε τα καλύτερα αποτελέσματα είναι η 3.5.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
75	3	10	10	5



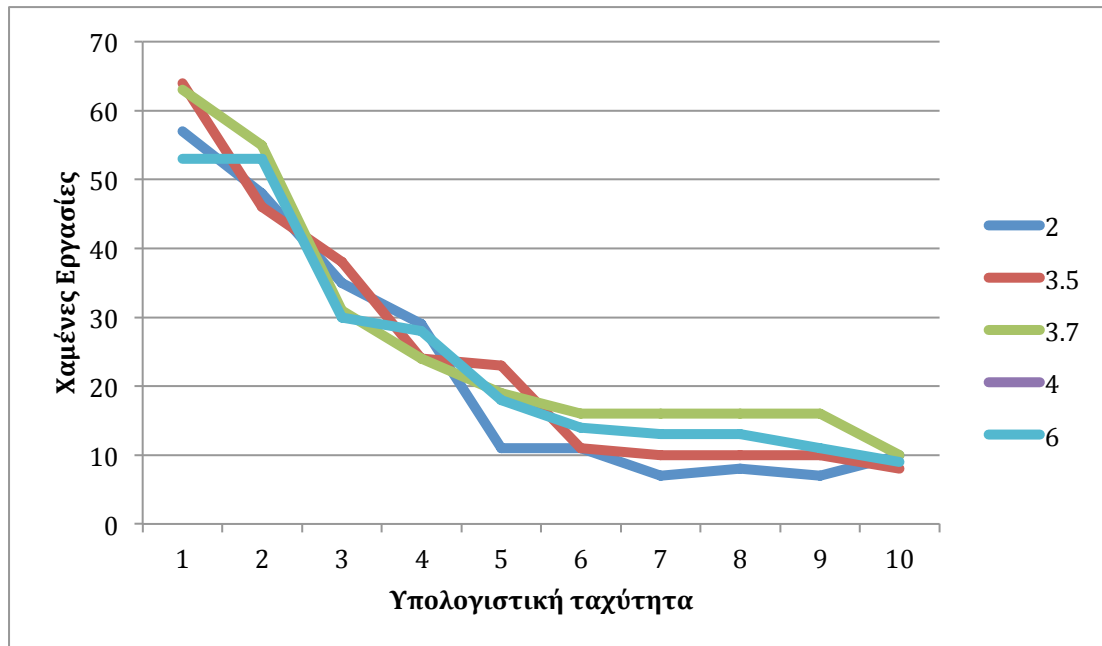
Μέγιστος Φόρτος: 0.624

Χαμένες Εργασίες: 20% - 72%

Σε αυτό το σενάριο μεταβάλαμε το μέγιστο χρονικό όριο εισδοχής εργασιών. Το όρο κυμαίνεται από 5 μέχρι 15. Όπως περιμέναμε, καθώς αυξάνεται το όριο εισδοχής εργασιών, ο αριθμός χαμένων εργασιών μειώνεται, αφού οι εργασίες εισέρχονται πιο αραιά και το σύστημα μπορεί να τις χειριστεί καλύτερα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 20% - 72%. Η πιθανότητα που έφερε τα καλύτερα αποτελέσματα είναι η 3.7.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
75	5	10	10	10



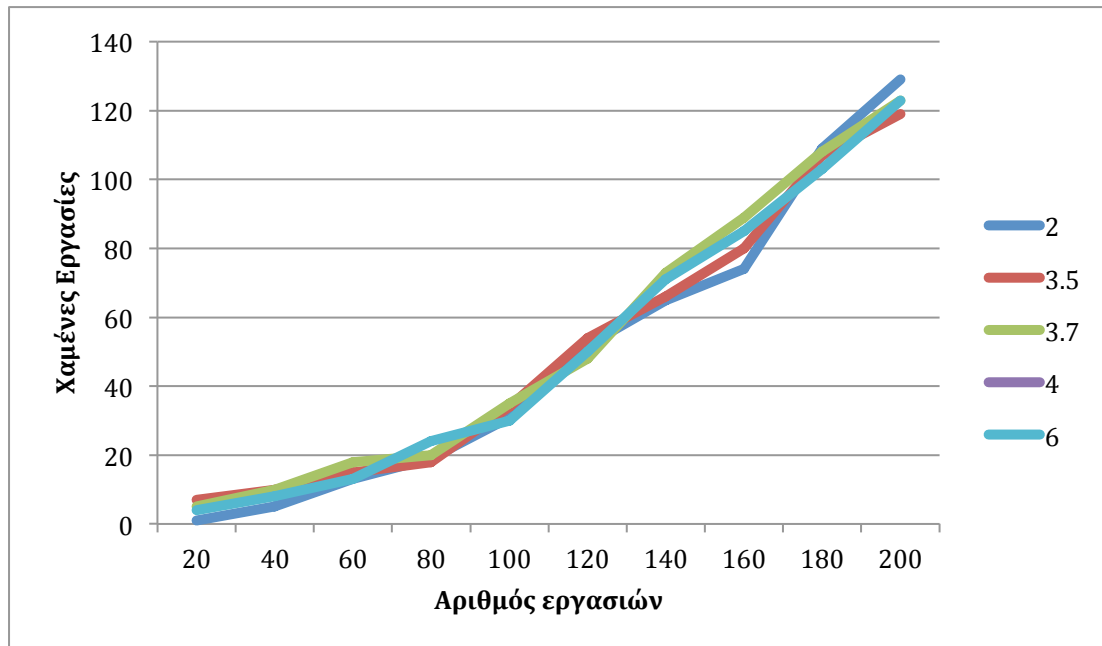
Μέγιστος Φόρτος: 0.375

Χαμένες Εργασίες: 10.6% - 86.6%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 10.6% - 86.6%. Σε αυτό το σενάριο, όπως και πριν περιμέναμε πως στις εκτελέσεις με μικρή υπολογιστική ταχύτητα οι μεγάλες πιθανότητες θα είχαν καλύτερα αποτελέσματα, αφού θα επέτρεπαν καλύτερη κατανομή των εργασιών σε αδύνατους υπολογιστικά κόμβους, αλλά και το αντίθετο οι μικρές πιθανότητες θα έπρεπε να είχαν καλύτερα αποτελέσματα στις εκτελέσεις με μεγάλη υπολογιστική ταχύτητα, αφού οι κόμβοι θα είχαν τη δυνατότητα να χειριστούν ευκολότερα τις εργασίες χωρίς κάποια κατανομή. Παρόλα αυτά όλες οι πιθανότητες είχαν περίπου ίδια αποτελέσματα.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



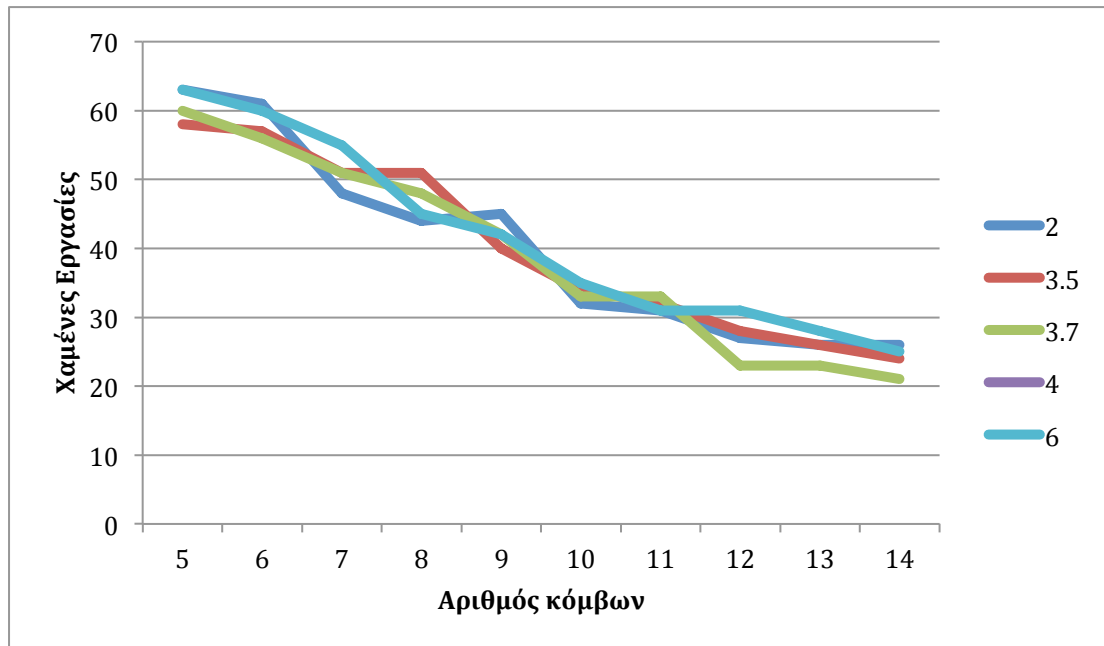
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 0% - 65%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 10 εργασίες στο σύστημα, τελειώνοντας στις 100 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 0% μέχρι 65%, όπως στο αντίστοιχο ελαφρά φορτωμένο σενάριο. Οι πιθανότητες και σε αυτό το σενάριο συμπεριφέρθηκαν το ίδιο.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
75	5	3	10	10



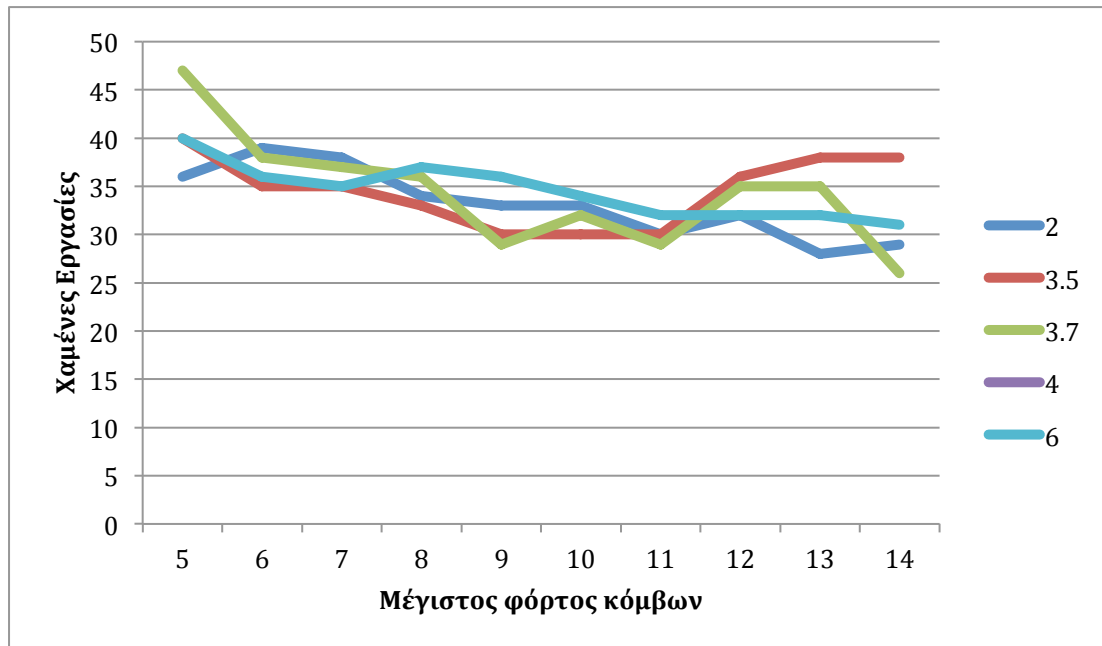
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 28% - 84%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 28% - 84%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με λιγότερους κόμβους, οι μεγάλες πιθανότητες (4,6) θα είχαν καλύτερα αποτελέσματα, αφού θα βελτίωναν την κατανομή των εργασιών στους κόμβους. Οι πιθανότητες όμως, συμπεριφέρθηκαν με τον ίδιο τρόπο σε όλες τις εκτελέσεις.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
75	5	3	10	10



Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 34.6% - 64%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό έχει ελάχιστο αντίκτυπο στον αριθμό χαμένων εργασιών, και οι διαφορετικές πιθανότητες στον αλγόριθμο δεν έπαιξαν κάποιο ιδιαίτερο ρόλο. Το ποσοστό χαμένων εργασιών κυμαίνεται από 34.6% - 64%.

Παρουσιάζονται οι εκτελέσεις με τον Αλγόριθμο 2

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
75	3	10	10	10



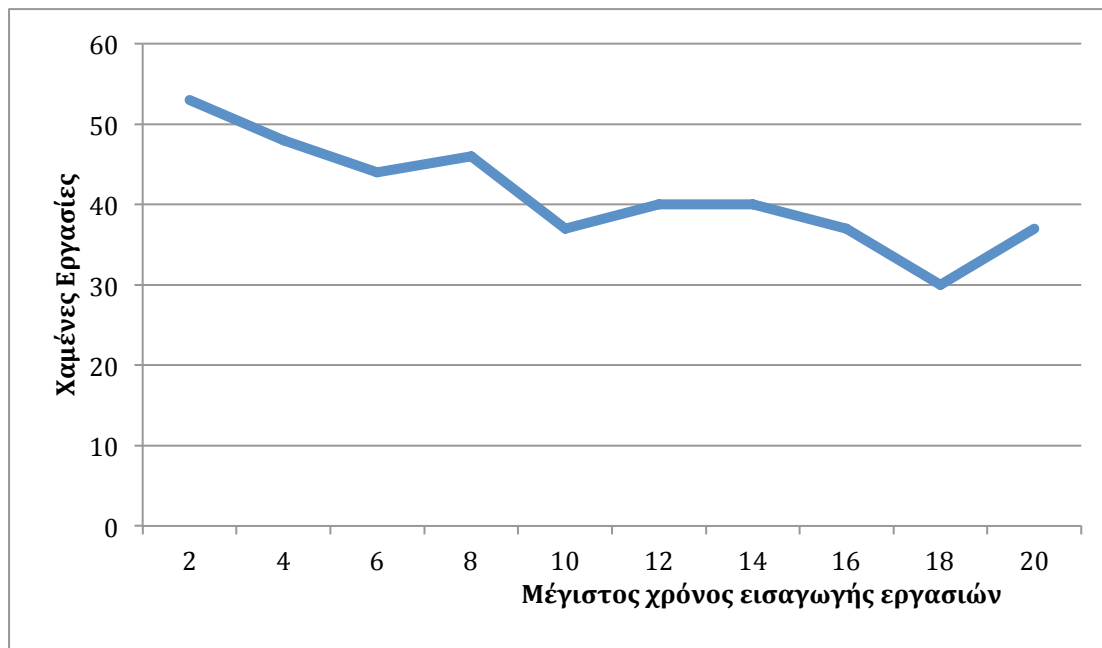
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 36% - 77.33%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται ο φόρτος εργασίας, αυξάνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 36% - 77.33%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
75	3	10	10	5



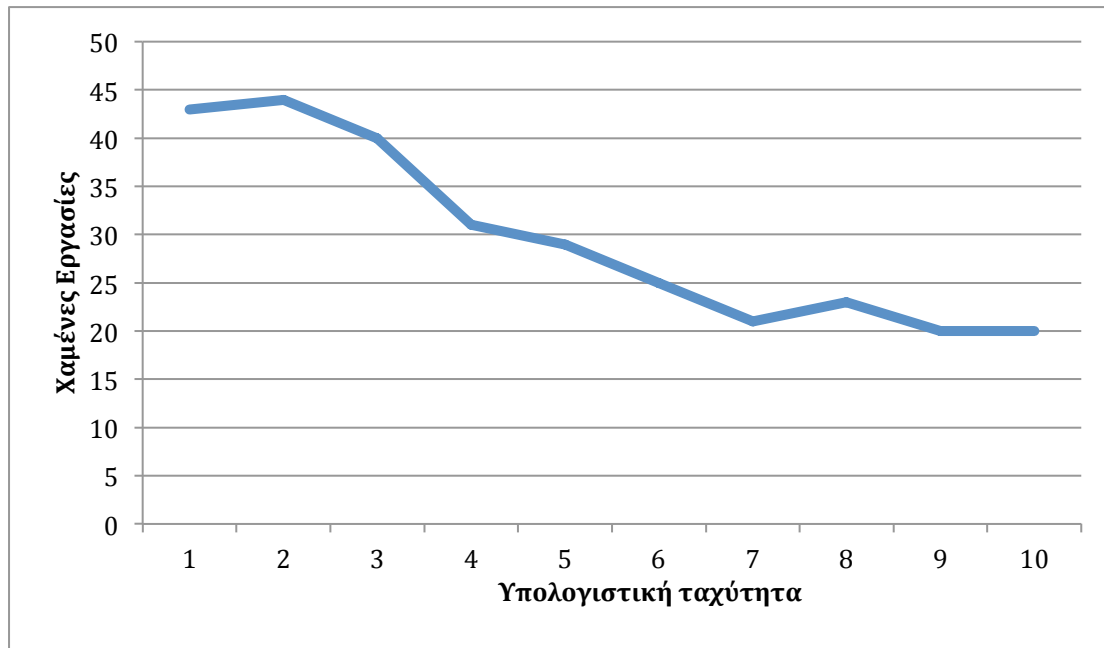
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 47% - 57%

Σε αυτό το σενάριο αυξάνεται το μέγιστο όριο εισδοχής εργασιών, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται το όριο εισδοχής εργασιών, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 47% - 57%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
75	5	10	10	10



Μέγιστος Φόρτος: 0.375

Χαμένες Εργασίες: 26.66% - 58.66%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26.66% - 58.66%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



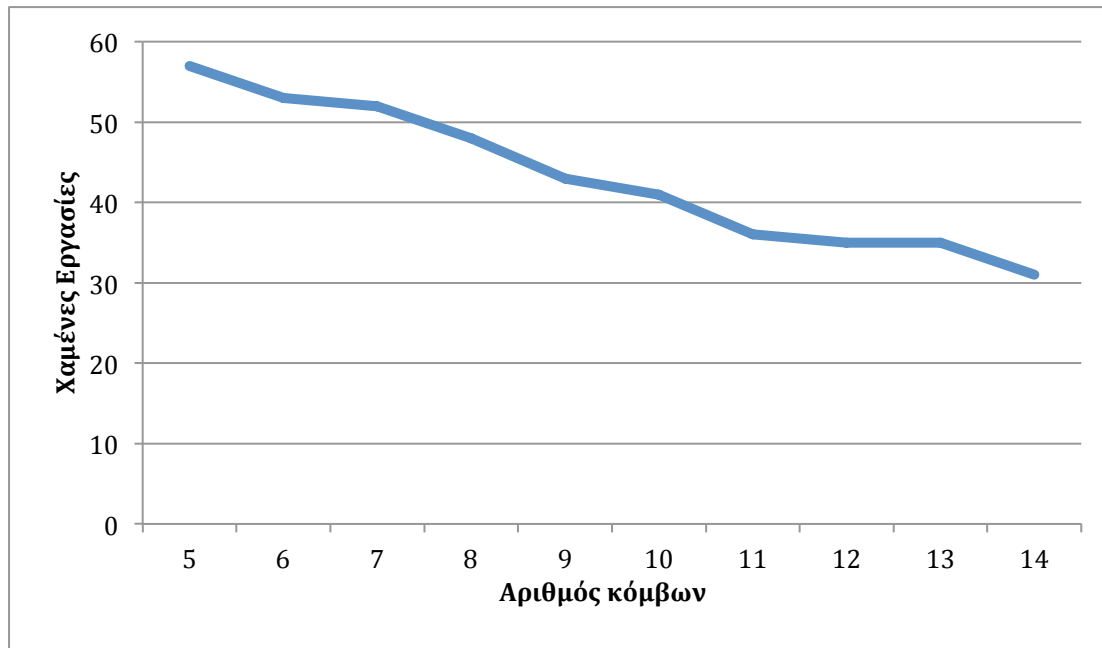
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 8% - 62%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 15 εργασίες στο σύστημα, τελειώνοντας στις 150 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 8% - 62%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
75	5	3	10	10



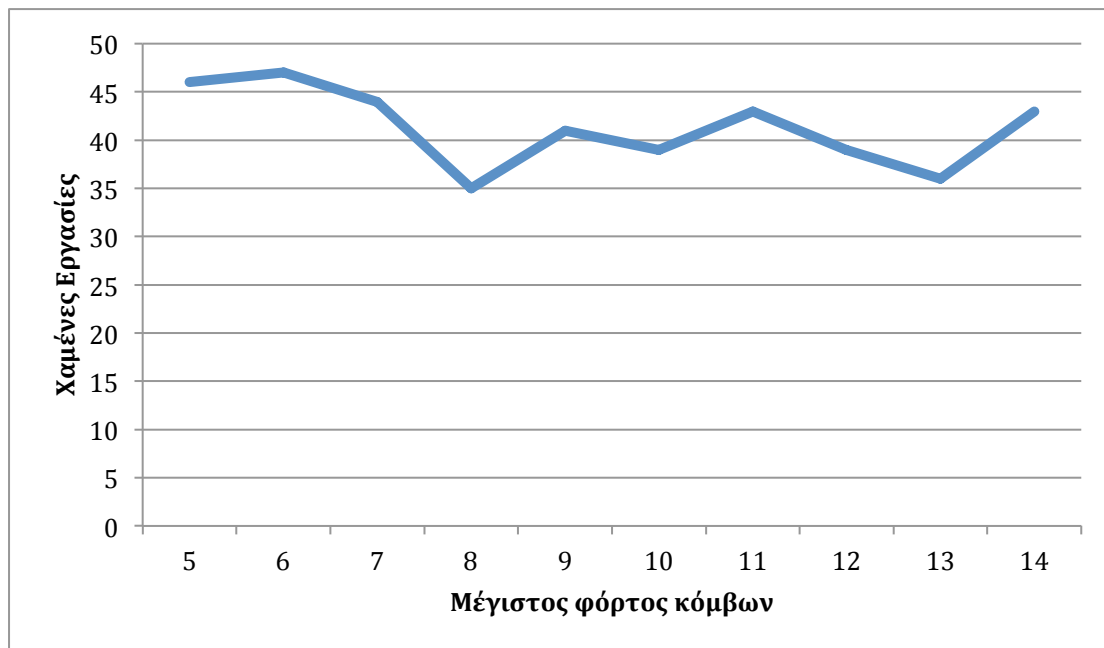
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 41.33% - 77.33%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 41.33% - 77.33%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
75	5	3	10	10



Μέγιστος Φόρτος: 0.249

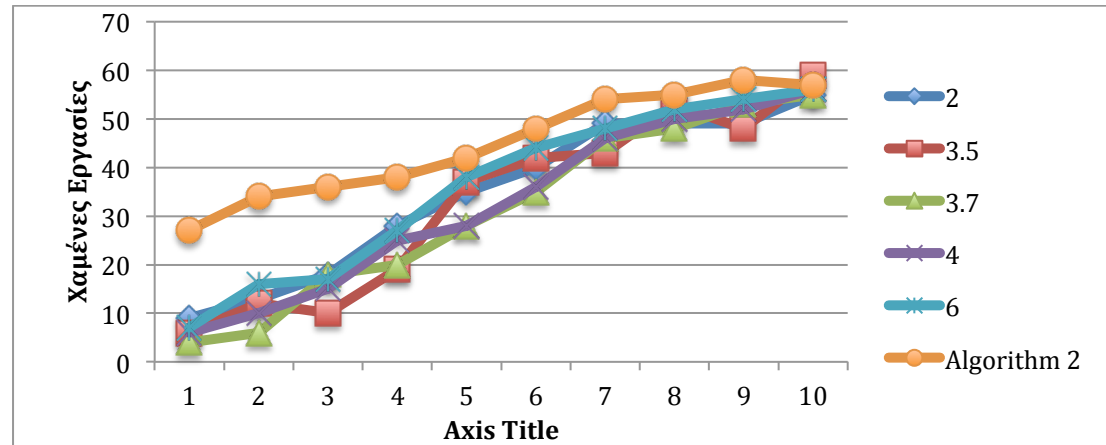
Χαμένες Εργασίες: 46.66% - 62.66%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 46.66% - 62.66%.

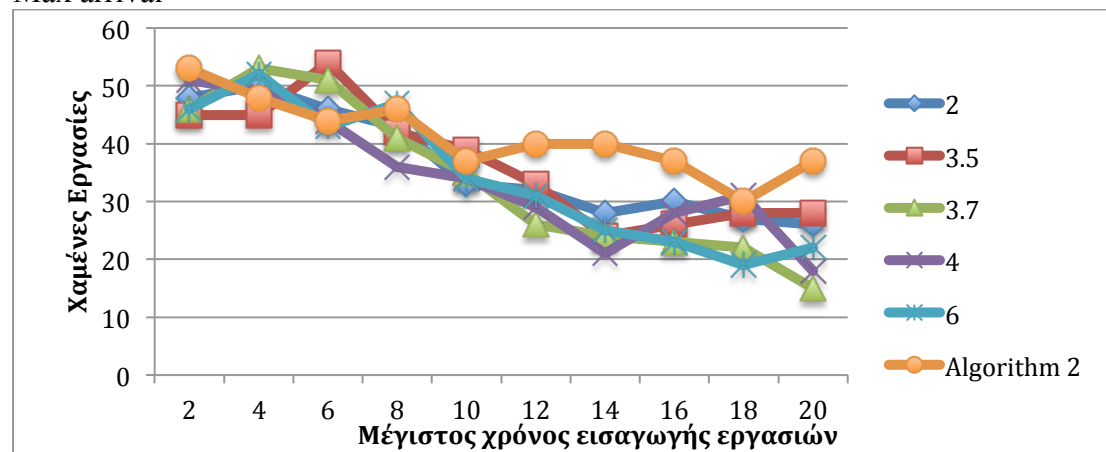
Σύγκριση Αλγορίθμων

Οι δύο αλγόριθμοι κινήθηκαν στα ίδια πλαίσια όσον αφορά το χρόνο εκτέλεσης των εργασιών. Η διαφορά τους έγκειται στον αριθμό χαμένων εργασιών, η οποία φαίνεται καλύτερα εάν αντιπαραβάλουμε τα αποτελέσματα των δύο αλγορίθμων μεταξύ τους.

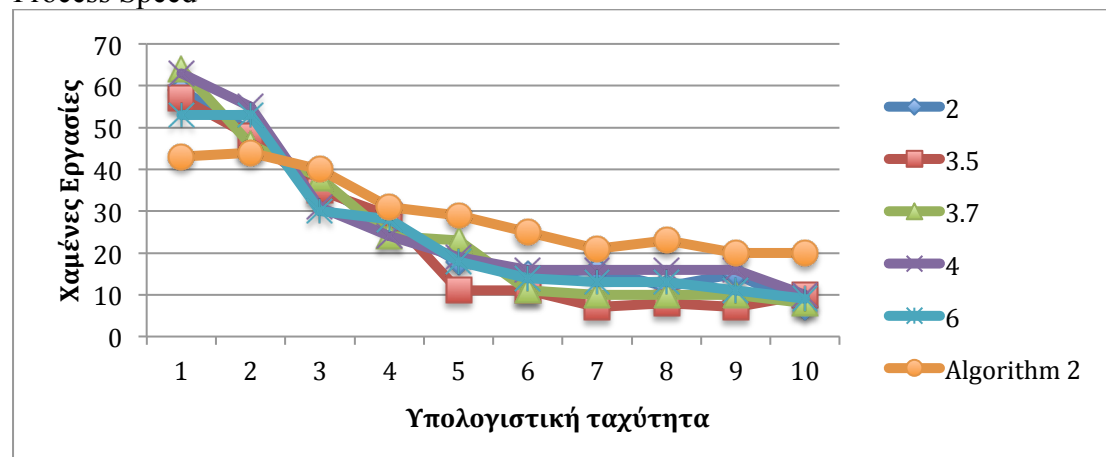
Job Load



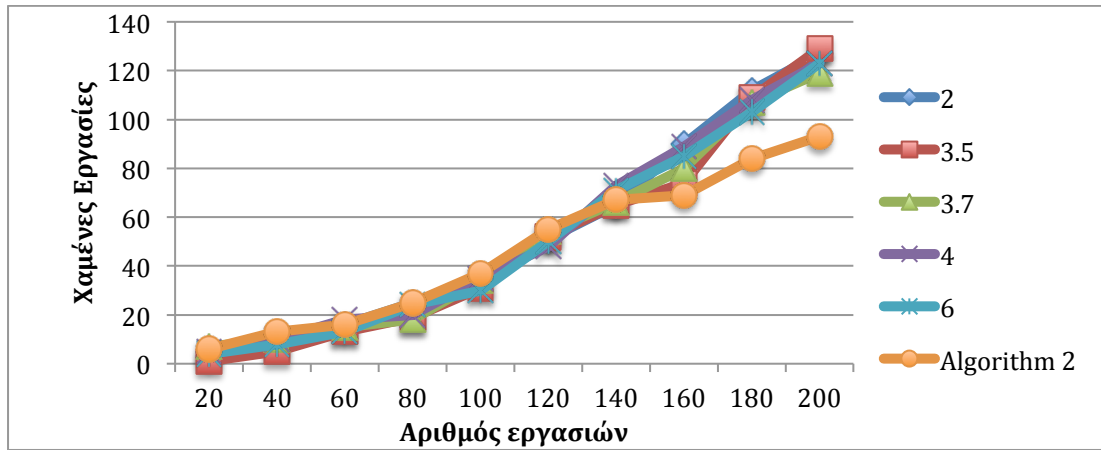
Max arrival



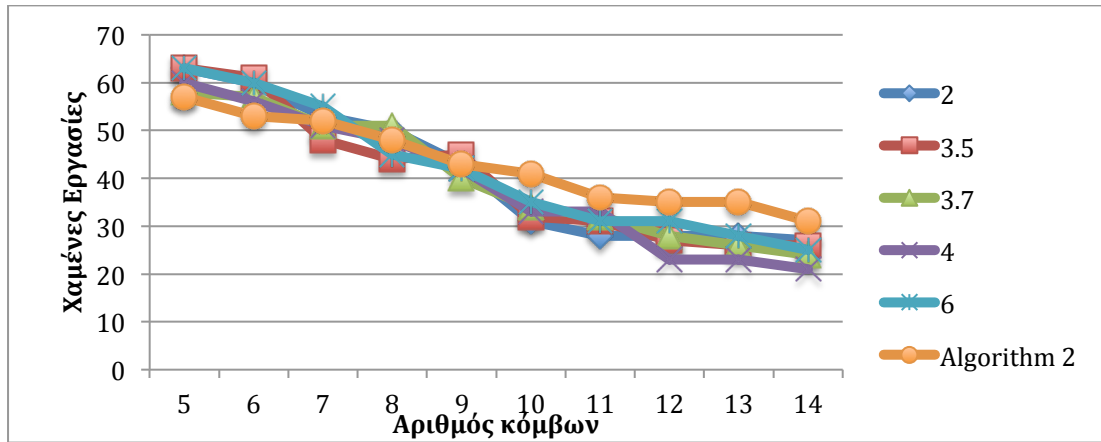
Process Speed



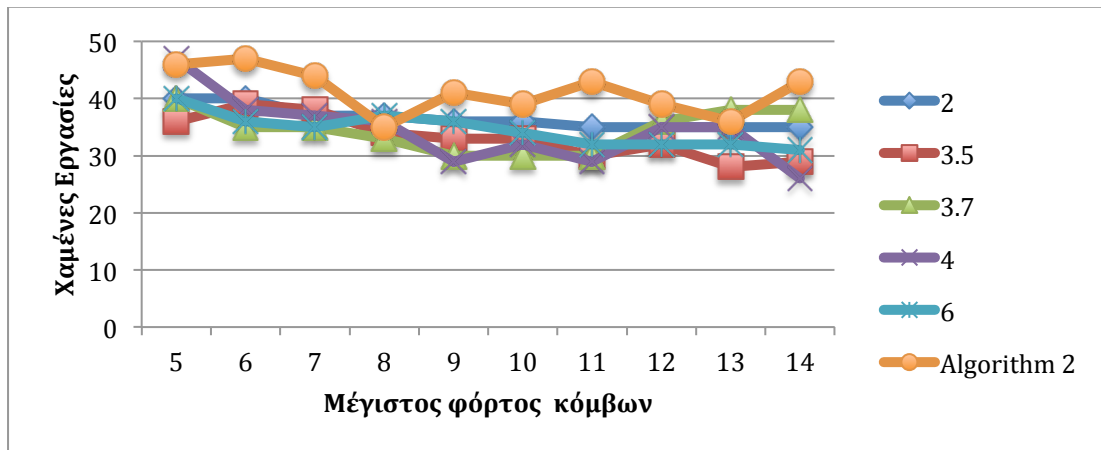
Number of Jobs



Number of Nodes



Node max Load



Στα μέτρια φορτωμένα σενάρια φαίνεται καθαρά από τις γραφικές παραστάσεις πως ο Αλγόριθμος 1 υπερτερεί με εξαίρεση το σενάριο στο οποίο μεταβάλλεται ο αριθμός των εργασιών (σενάριο 4). Όλες οι πιθανότητες του Αλγορίθμου 1 φέρουν καλύτερα αποτελέσματα εαν τις συγκρίνουμε με τα αποτελέσματα του Αλγορίθμου 2.

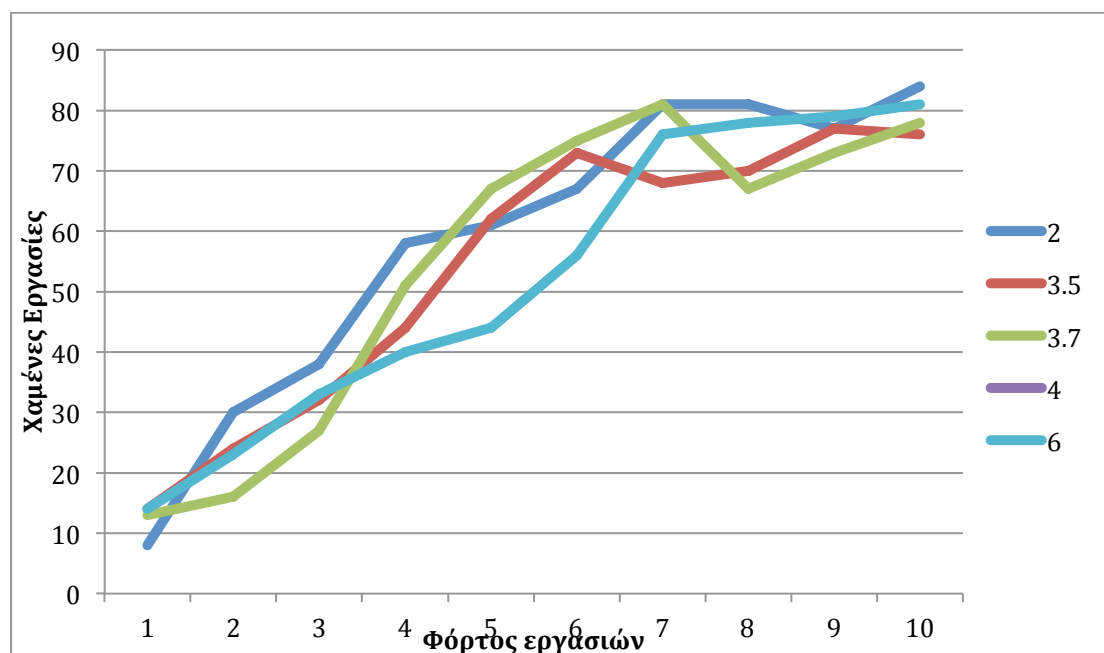
5.4.3 Βαριά Φορτωμένα Σενάρια

Τα βαριά φορτωμένα σενάρια έχουν συντελεστή μέχρι και 0.832.

Παρουσιάζονται πρώτα οι εκτελέσεις με τον Αλγόριθμο 1.

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
100	3	10	10	10



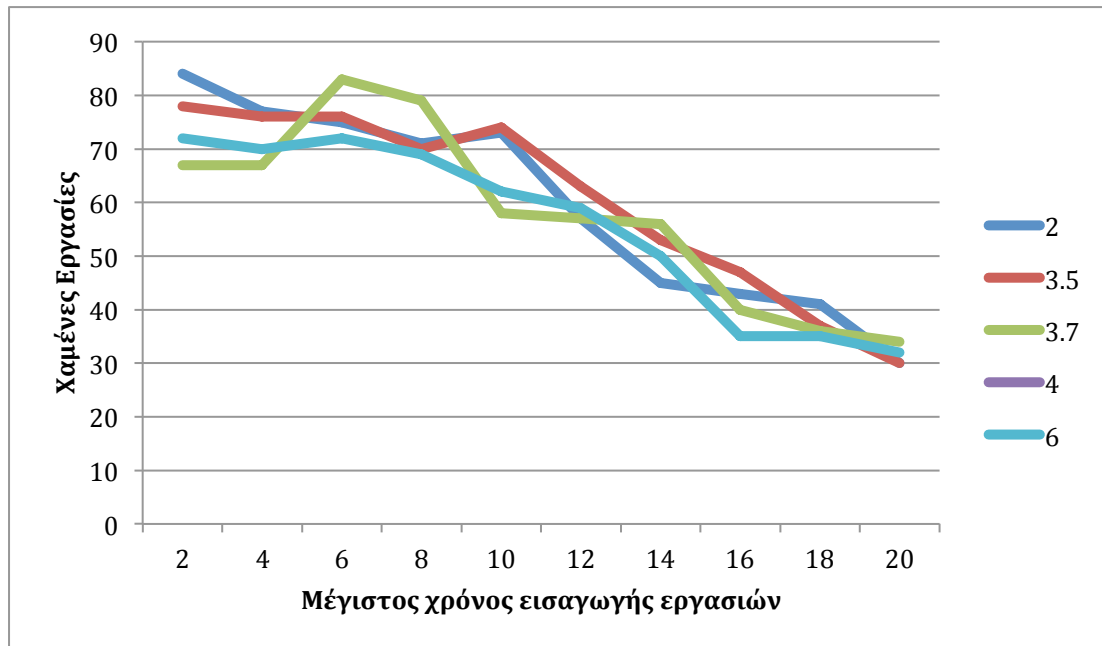
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 9% - 84%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως ο αριθμός χαμένων εργασιών αυξάνεται σταδιακά. Το ποσοστό χαμένων εργασιών κυμαίνεται από 9% - 84%. Η πιθανότητα που έφερε τα καλύτερα αποτελέσματα είναι η 6.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
100	3	10	10	5



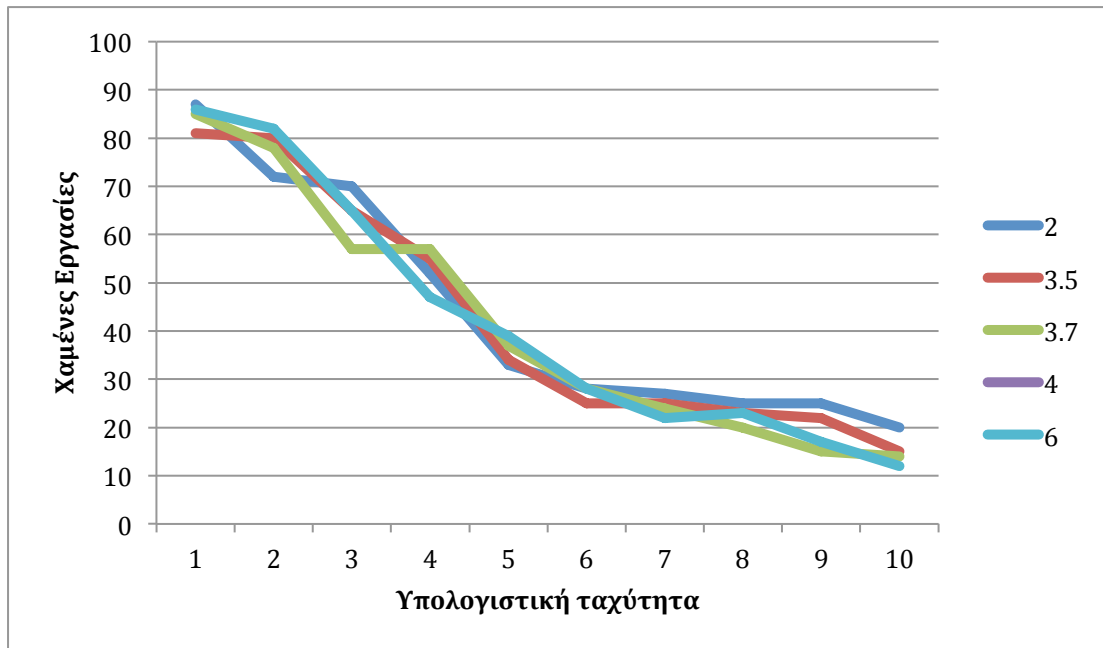
Μέγιστος Φόρτος: 0.832

Χαμένες Εργασίες: 30% - 84%

Σε αυτό το σενάριο μεταβάλαμε το μέγιστο χρονικό όριο εισδοχής εργασιών. Το όρο κυμαίνεται από 2 μέχρι 20. Όπως περιμέναμε, καθώς αυξάνεται το όριο εισδοχής εργασιών, ο αριθμός χαμένων εργασιών μειώνεται, αφού οι εργασίες εισέρχονται πιο αραιά και το σύστημα μπορεί να τις χειριστεί καλύτερα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 30% - 84%. Η πιθανότητα που έφερε τα καλύτερα αποτελέσματα είναι η 6.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
100	5	10	10	10



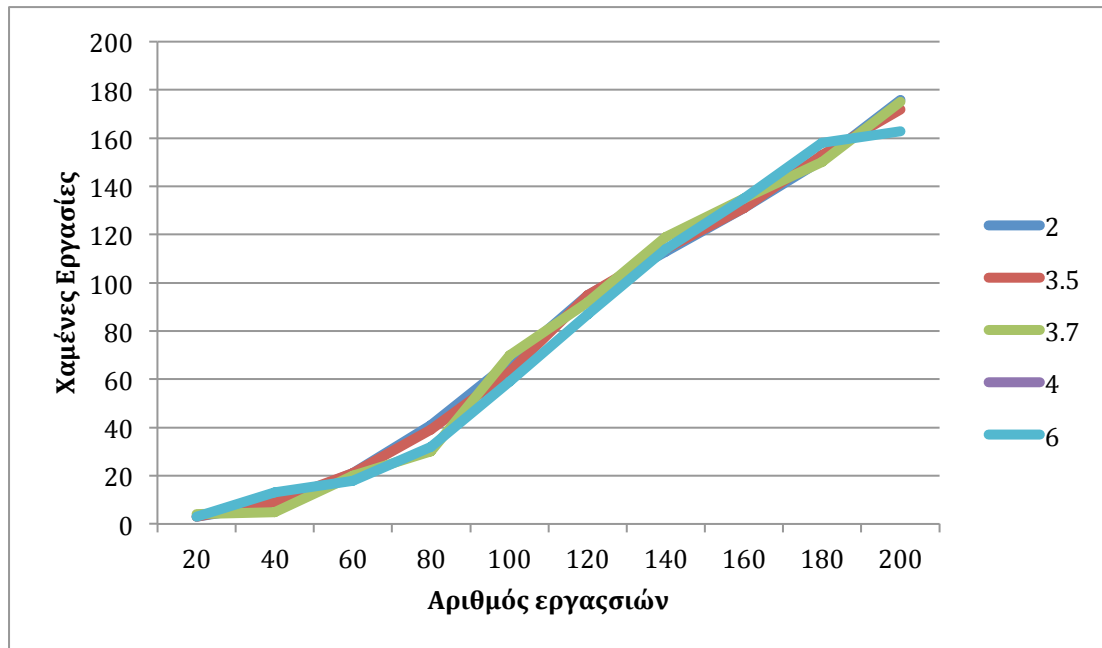
Μέγιστος Φόρτος: 0.5

Χαμένες Εργασίες: 11% - 88%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 11% - 88%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με μικρή υπολογιστική ταχύτητα οι μεγάλες πιθανότητες (4,6) θα είχαν καλύτερα αποτελέσματα, αφού θα επέτρεπαν καλύτερη κατανομή των εργασιών σε αδύνατους υπολογιστικά κόμβους, αλλά και το αντίθετο οι μικρές πιθανότητες θα έπρεπε να είχαν καλύτερα αποτελέσματα στις εκτελέσεις με μεγάλη υπολογιστική ταχύτητα, αφού οι κόμβοι θα είχαν την δυνατότητα να χειριστούν ευκολότερα τις εργασίες χωρίς κάποια κατανομή. Παρόλα αυτά η πιθανότητα που έδωσε τα καλύτερα αποτελέσματα ήταν η 3.7.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



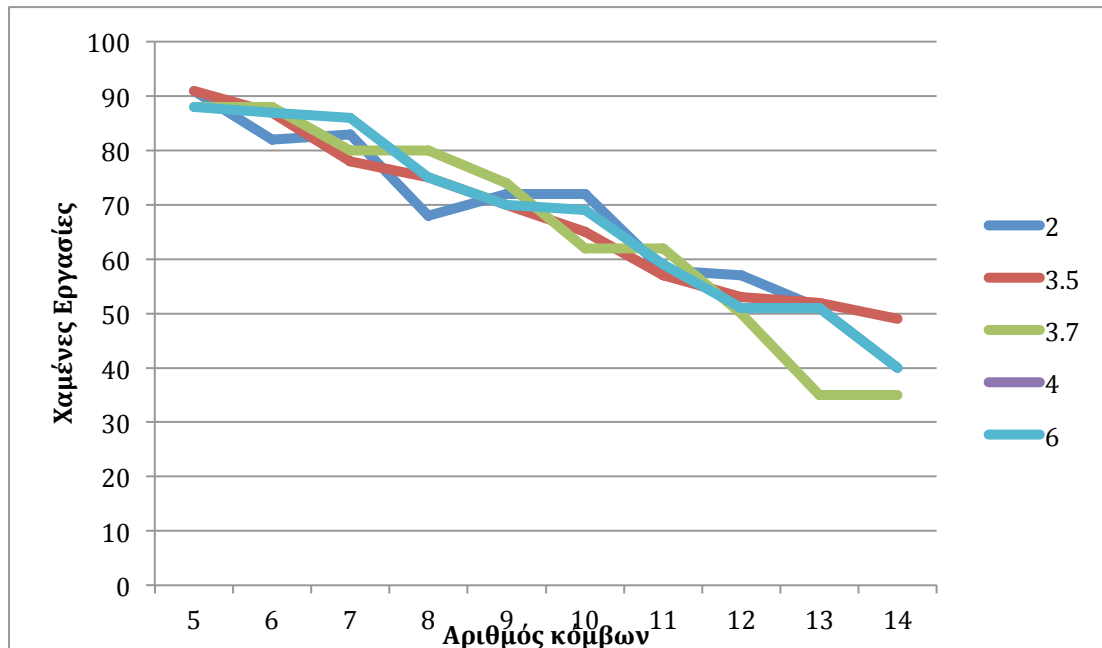
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 2% - 86%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά είκοσι, αρχίζοντας με 20 εργασίες στο σύστημα, τελειώνοντας στις 200 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 2% - 86%. Με μικρή διαφορά η πιθανότητα 6 έδωσε καλύτερα αποτελέσματα.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
100	5	3	10	10



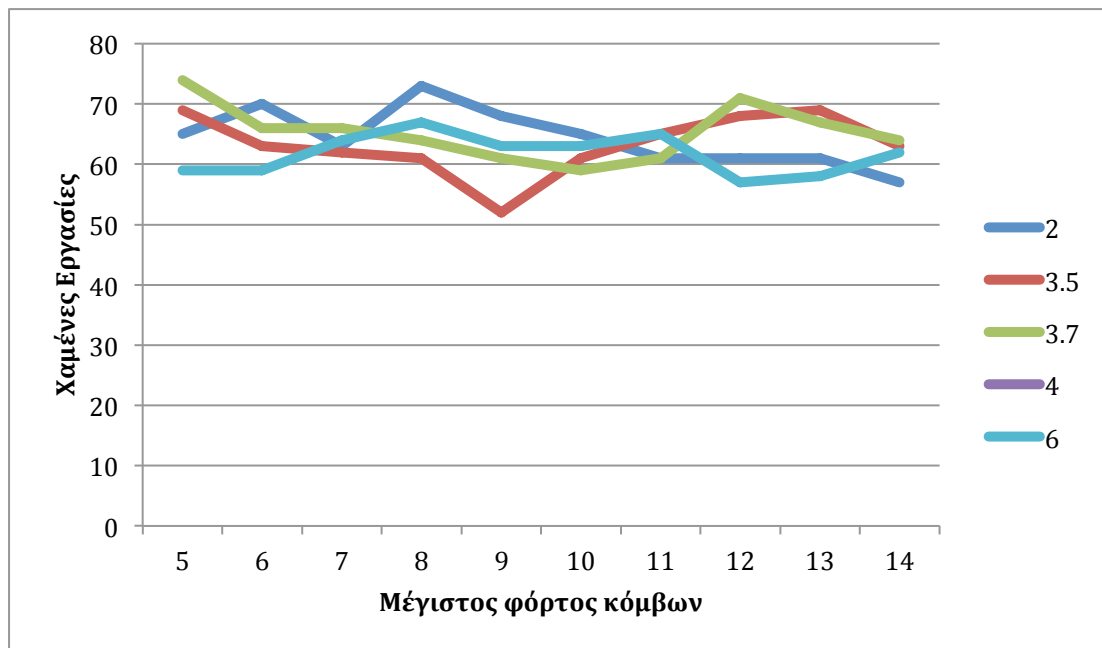
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 35% - 91%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 35% - 91%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με λιγότερους κόμβους, οι μεγάλες πιθανότητες (4,6) θα είχαν καλύτερα αποτελέσματα, αφού θα βελτίωναν την κατανομή των εργασιών στους κόμβους. Παρόλα αυτά η πιθανότητα που είχε καλύτερα αποτελέσματα είναι η 3.5.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
100	5	3	10	10



Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 52% - 73%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών, ούτε οι διαφορετικές πιθανότητες στον αλγόριθμο έπαιξαν κάποιο ρόλο. Το ποσοστό χαμένων εργασιών κυμαίνεται από 52% - 73%.

Παρουσιάζονται οι εκτελέσεις με τον Αλγόριθμο 2

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
100	3	10	10	10



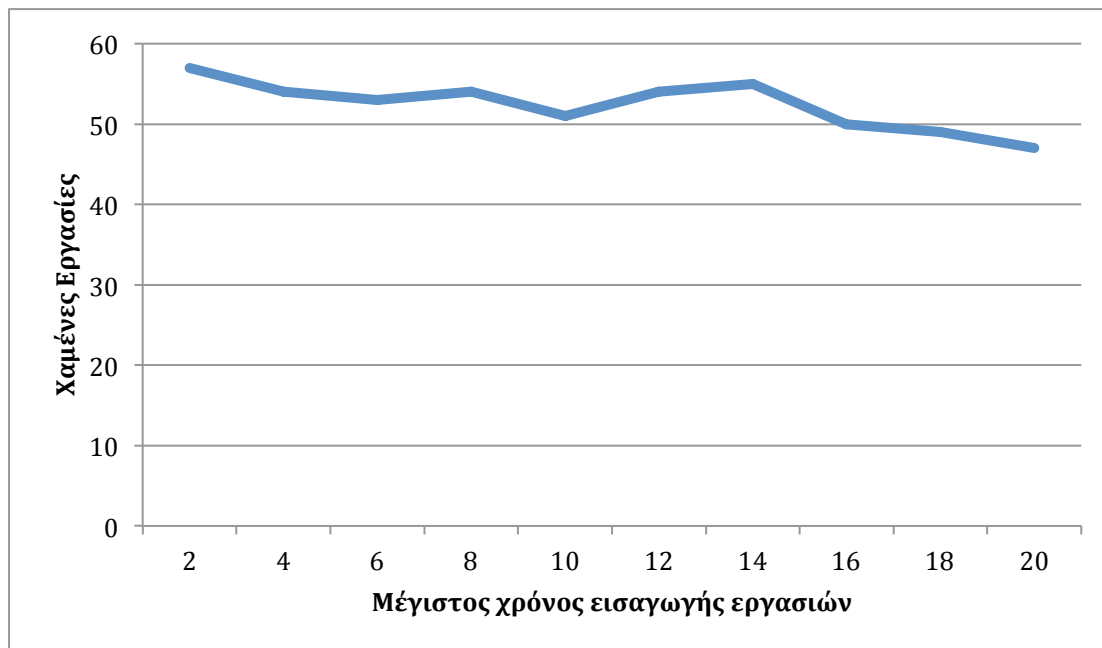
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 28% - 71%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται ο φόρτος εργασίας, αυξάνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 28% - 71%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
100	3	10	10	5



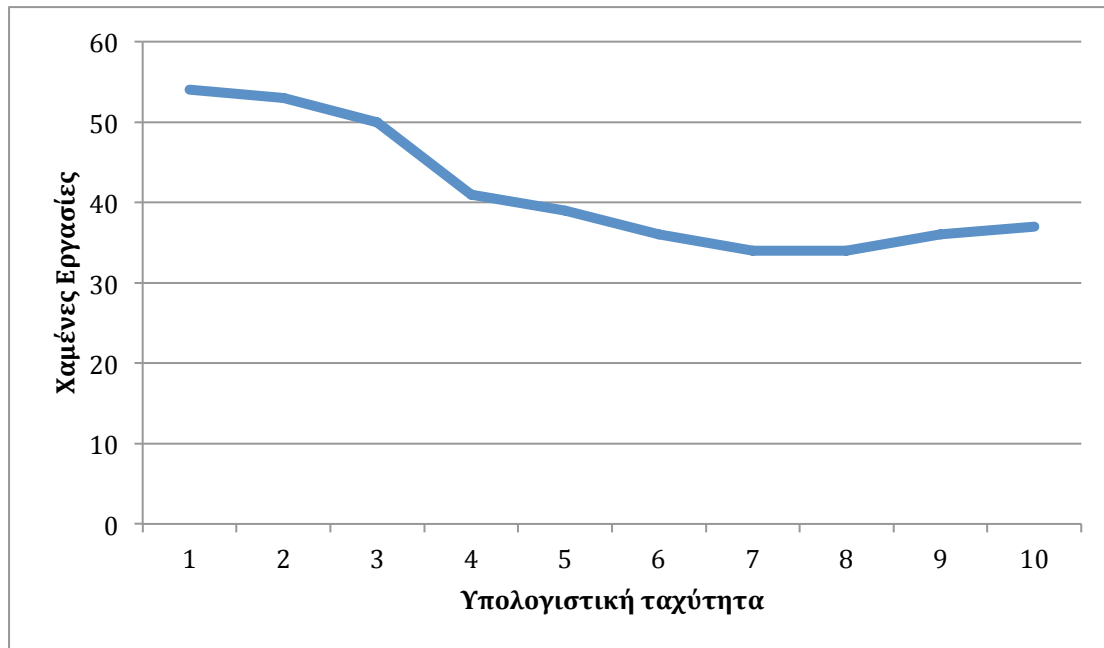
Μέγιστος Φόρτος: 0.832

Χαμένες Εργασίες: 30% - 84%

Σε αυτό το σενάριο αυξάνεται το μέγιστο όριο εισδοχής εργασιών, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται το όριο εισδοχής εργασιών, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 30% - 84%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
100	5	10	10	10



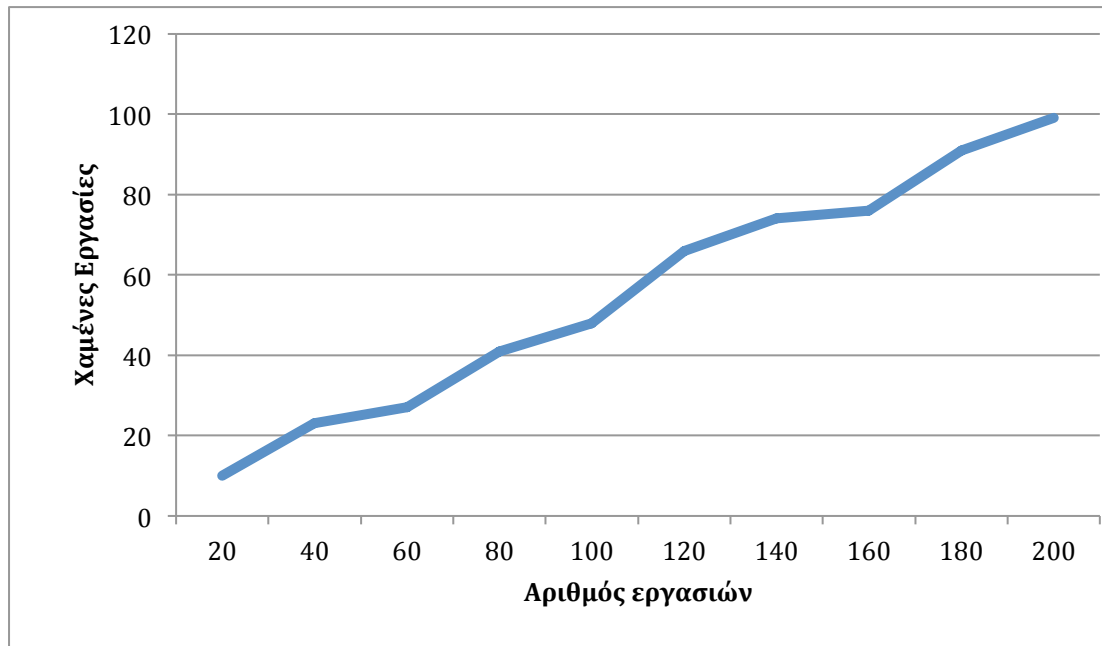
Μέγιστος Φόρτος: 0.5

Χαμένες Εργασίες: 34% - 54%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 34% - 54%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



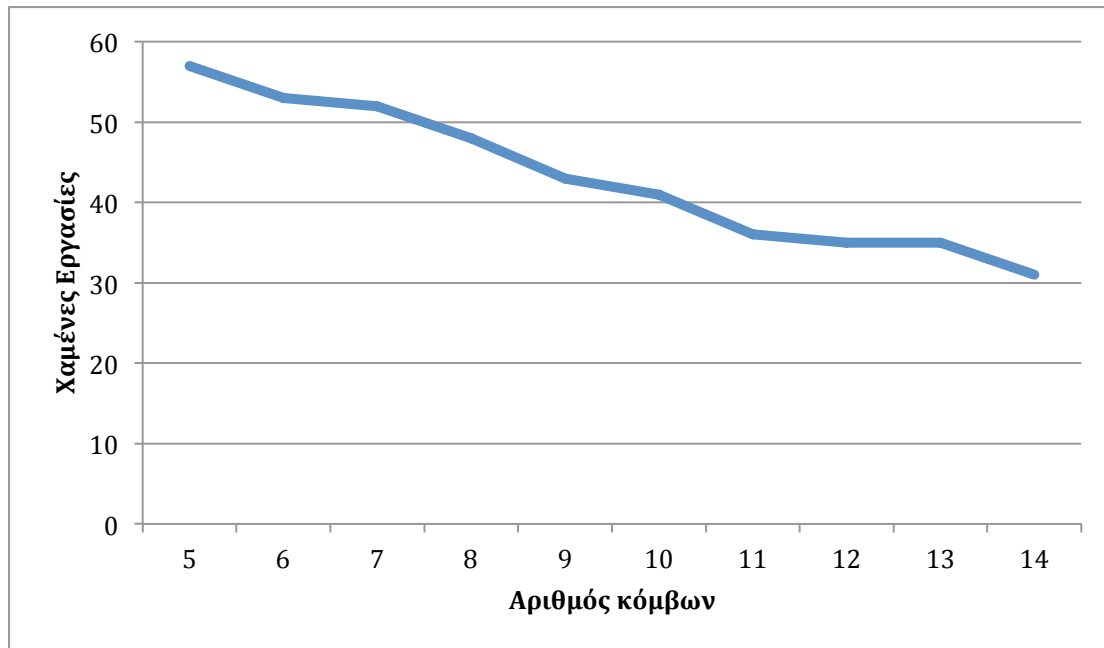
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 50% - 65%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 10 εργασίες στο σύστημα, τελειώνοντας στις 100 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 50% - 65%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
100	5	3	10	10



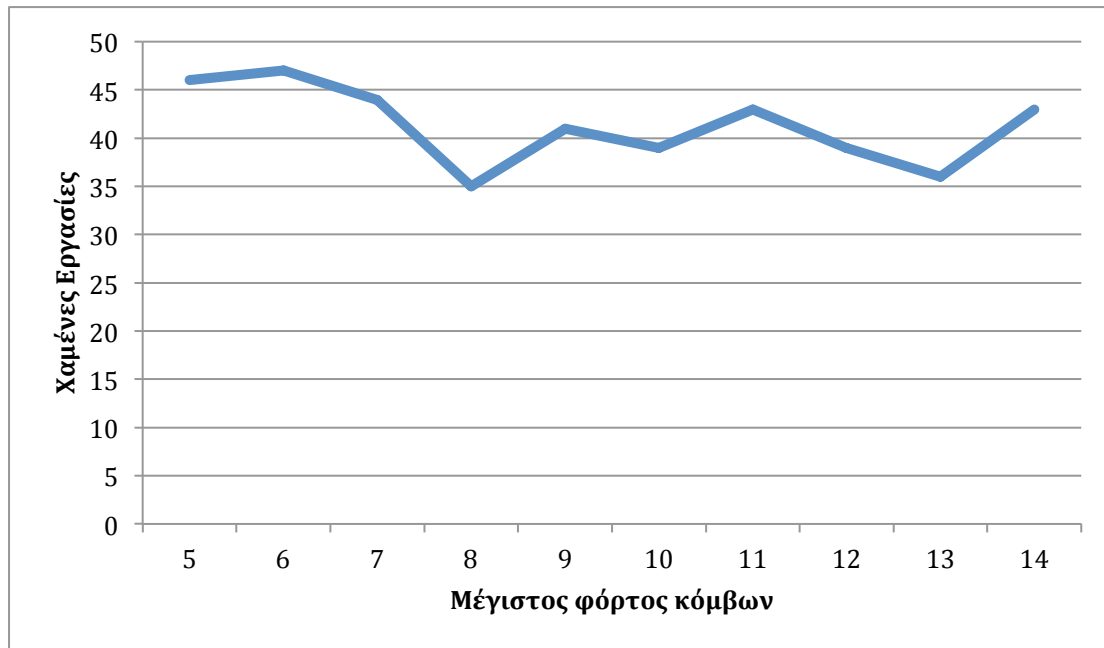
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 31% - 57%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 31% - 57%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
100	5	3	10	10



Μέγιστος Φόρτος: 0.332

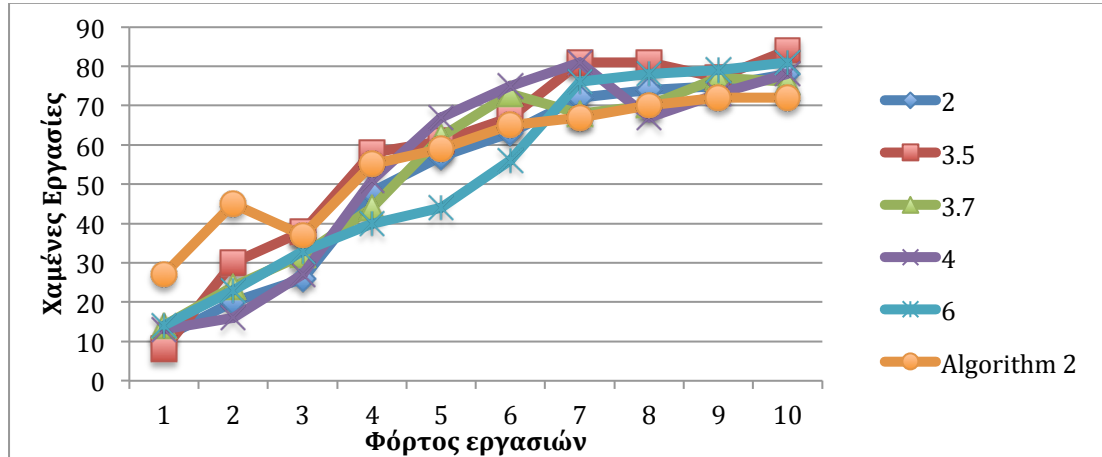
Χαμένες Εργασίες: 35% - 47%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 35% - 47%.

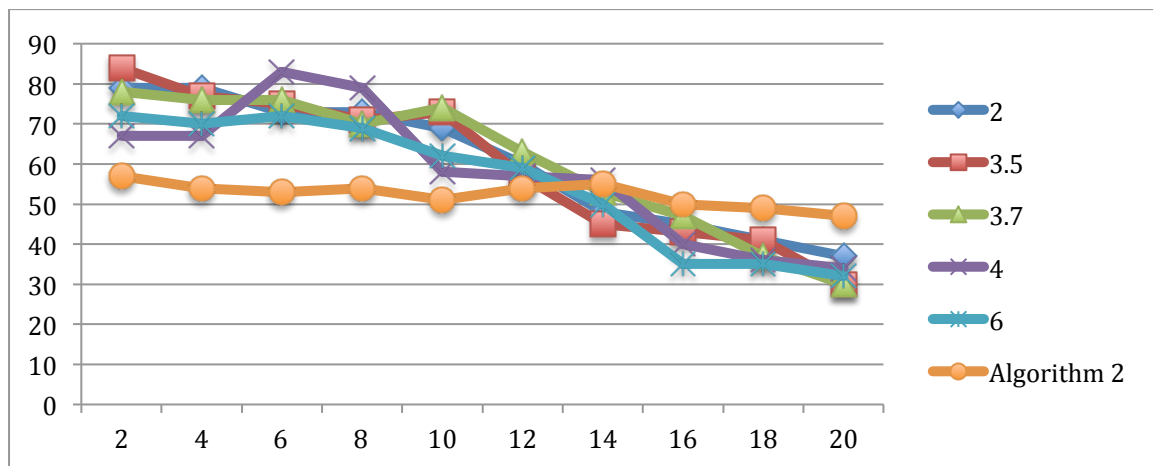
Σύγκριση Αλγορίθμων

Οι δύο αλγόριθμοι κινήθηκαν στα ίδια πλαίσια όσον αφορά το χρόνο εκτέλεσης των εργασιών. Η διαφορά τους έγκειται στον αριθμό χαμένων εργασιών, η οποία φαίνεται καλύτερα εάν αντιπαραβάλουμε τα αποτελέσματα των δύο αλγορίθμων μεταξύ τους.

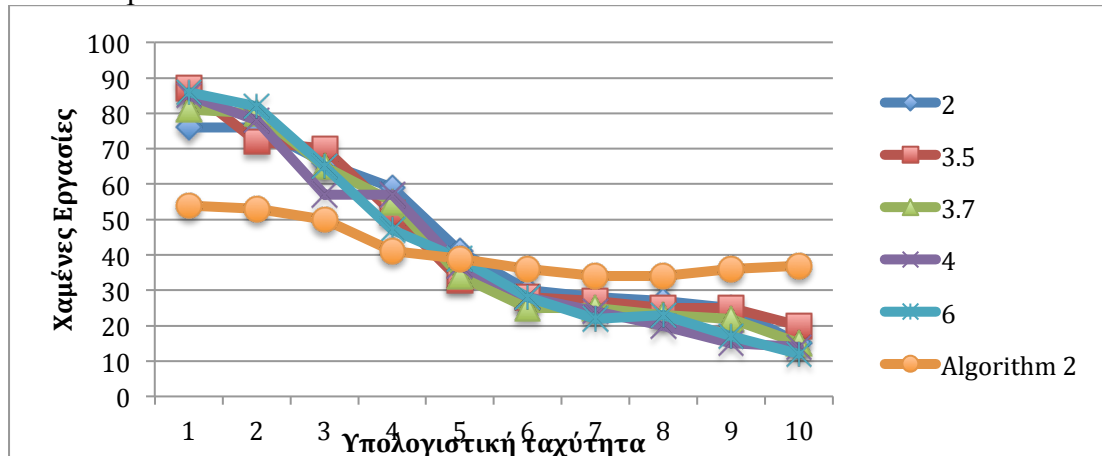
Job Load



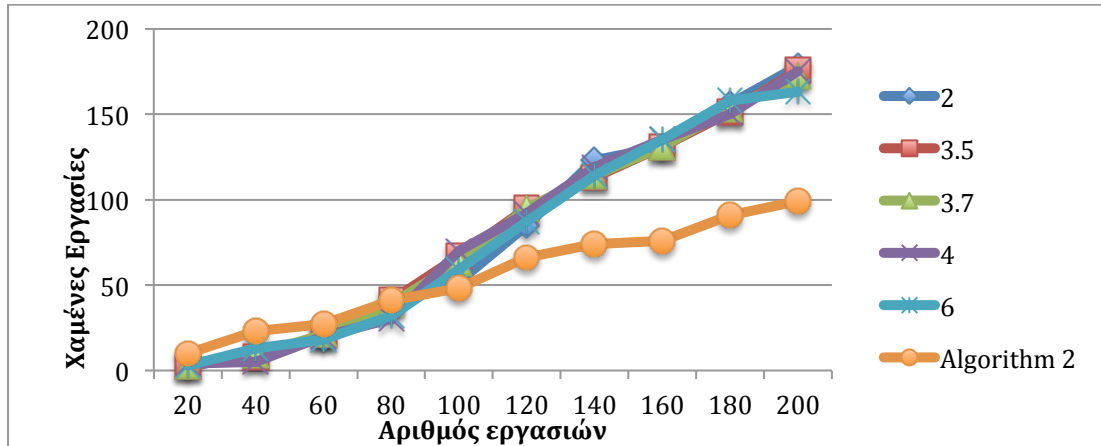
Max arrival Time



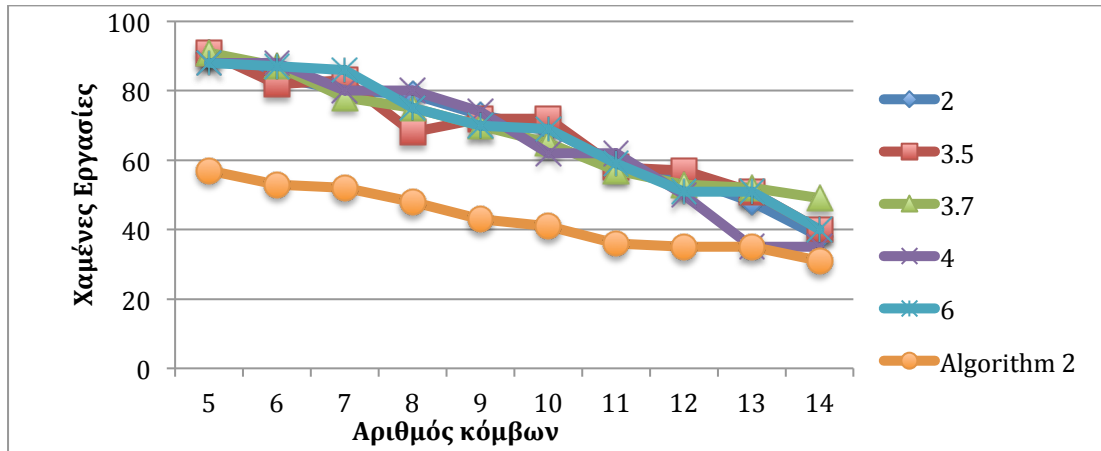
Process speed



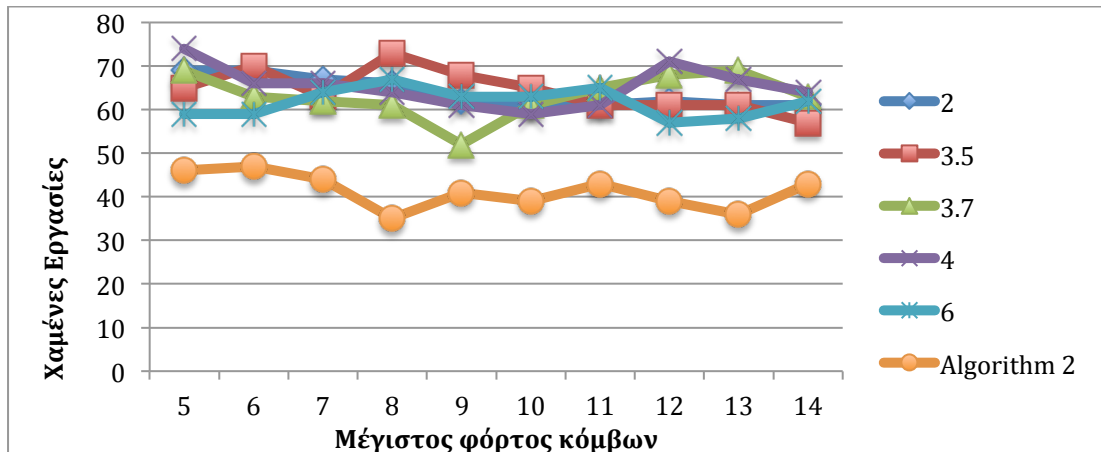
Number of Jobs



Number of Nodes



Node Maximum Load



Στο πρώτο σενάριο ο Αλγόριθμος 1 αρχίζει με λιγότερες χαμένες εργασίες και καταλήγει στο ίδιο σημείο περίπου με τον Αλγόριθμο 2. Στα σενάρια 2 και 3 παρόλο που ο Αλγόριθμος 1 αρχίζει με περισσότερες χαμενες εργασίες, καταλήγει με λιγότερες. Στα τελευταία τρία σενάρια ο Αλγόριθμος 2 έχει αρκετά λιγότερες χαμένες εργασίες.

Στη συνέχεια θέλοντας να ενσωματώσουμε περισσότερα στοιχεία της μελέτης Uncertainty in Explicit Population Models στον αλγόριθμο εμπνευσμένο από αυτή, αποφασίσαμε να κάνουμε αλλαγές και στους δύο αλγόριθμους και να τους συγκρίνουμε ξανά.

Όπως προαναφέρθηκε στο άρθρο ‘Uncertainty in Explicit Population Models’ υπάρχουν τρία είδη πουλιών, οι Breeders, οι Floaters και οι Juveniles. Στη σύγκριση του προβλήματος εξισορρόπησης φόρτου εργασίας με το πρόβλημα μετανάστευσης πουλιών αντιστοιχήθηκαν τα πουλιά του άρθρου, μαζί με τις εργασίες του υπολογιστικού νέφους. Τέθηκε σε εφαρμογή η αντιστοιχία αυτή με το να διαμορφώσουμε τα πιο πάνω σενάρια αυξάνοντας τον αριθμό των τύπων εργασίας σε 3, με βάση τον πίνακα αντιστοίχισης.

Οι εργασίες χωρίστηκαν σε 3 ίσα μέρη, προσπαθώντας να διατηρηθεί ο συνολικός φόρτος των εργασιών στο σύστημα.

Για τον Αλγόριθμο 1, η πιθανότητα μετακίνησης των εργασιών έχει ως εξής:

Breeder: 0.2

Floater: 0.8

Juvenile: 0.4

Για τον Αλγόριθμο 1 πήραμε τις πιθανότητες που έδωσαν τα καλύτερα αποτελέσματα και τα χρησιμοποιήσαμε. Στα σενάρια στα οποία δεν είχαμε πιθανότητα με καλύτερα αποτελέσματα εξακολουθούν να προβάλλονται 5 διαφορετικές πιθανότητες.

5.5.1 Ελαφρά Φορτωμένα Σενάρια

Τα ελαφρά φορτωμένα σενάρια έχουν συντελεστή μέχρι και 0.166.

Παρουσιάζονται πρώτα οι εκτελέσεις με τον Αλγόριθμο 1.

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
50	3	10	10	10



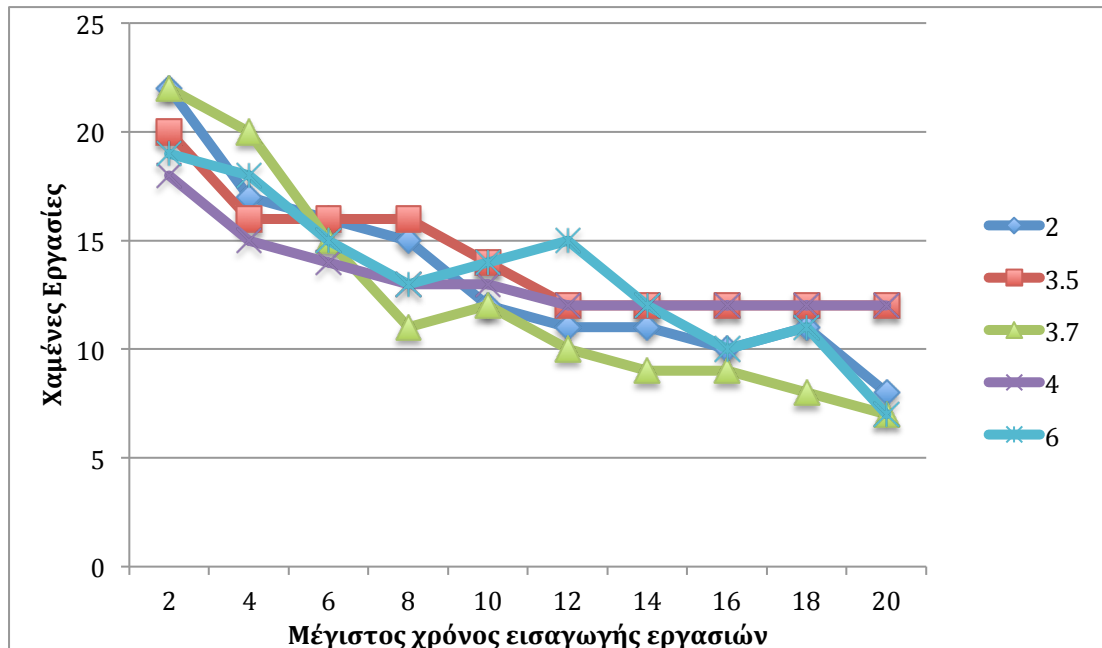
Μέγιστος Φόρτος: 0.166

Χαμένες: 10% - 42%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως ο αριθμός χαμένων εργασιών αυξάνεται σταδιακά. Το ποσοστό χαμένων εργασιών κυμαίνεται από 4% μέχρι 60%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
50	3	10	10	5



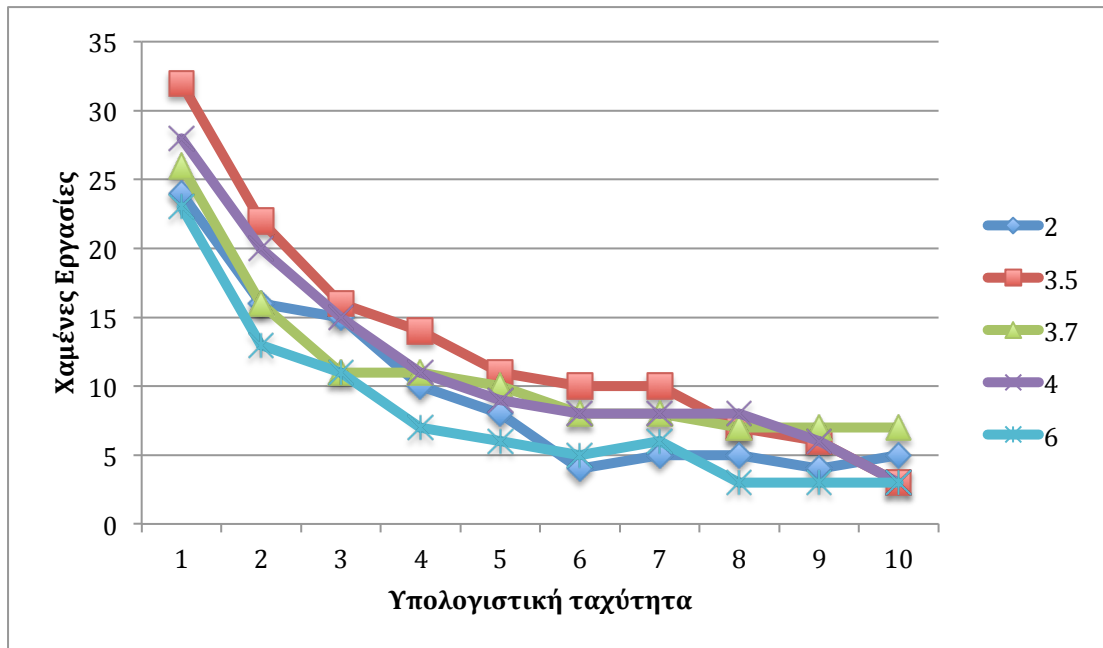
Μέγιστος Φόρτος: 0.416

Χαμένες Εργασίες: 14% - 44%

Σε αυτό το σενάριο μεταβάλαμε το μέγιστο χρονικό όριο εισδοχής εργασιών. Το όρο κυμαίνεται από 5 μέχρι 15. Όπως περιμέναμε, καθώς αυξάνεται το όριο εισδοχής εργασιών, ο αριθμός χαμένων εργασιών μειώνεται, αφού οι εργασίες εισέρχονται πιο αραιά και το σύστημα μπορεί να τις χειριστεί καλύτερα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 14% μέχρι 44%. Οι πέντε πιθανότητες συμπεριφέρθηκαν περίπου το ίδιο.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
50	5	10	10	10



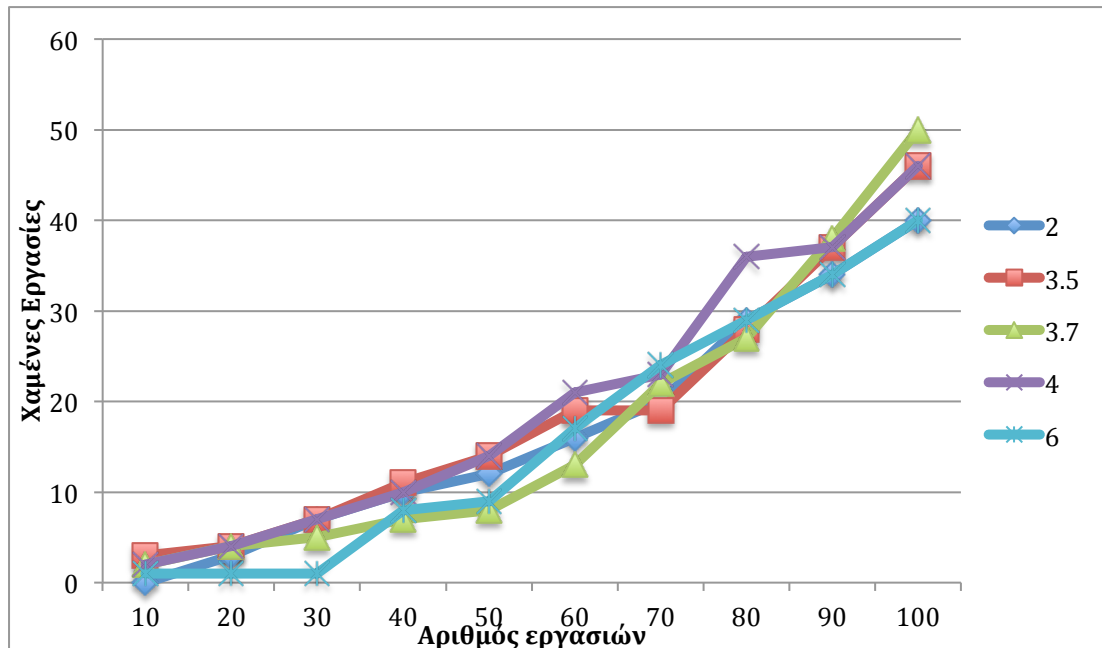
Μέγιστος Φόρτος: 0.25

Χαμένες: 6% - 46%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 4% μέχρι 46%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με μικρή υπολογιστική ταχύτητα οι μεγάλες πιθανότητες θα είχαν καλύτερα αποτελέσματα, αφού θα επέτρεπαν καλύτερη κατανομή των εργασιών σε αδύνατους υπολογιστικά κόμβους, αλλά και το αντίθετο οι μικρές πιθανότητες θα έπρεπε να είχαν καλύτερα αποτελέσματα στις εκτελέσεις με μεγάλη υπολογιστική ταχύτητα, αφού οι κόμβοι θα είχαν την δυνατότητα να χειριστούν ευκολότερα τις εργασίες χωρίς κάποια κατανομή. Παρόλα αυτά η πιθανότητα 6 υπερείχε σε όλες τις εκτελέσεις.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



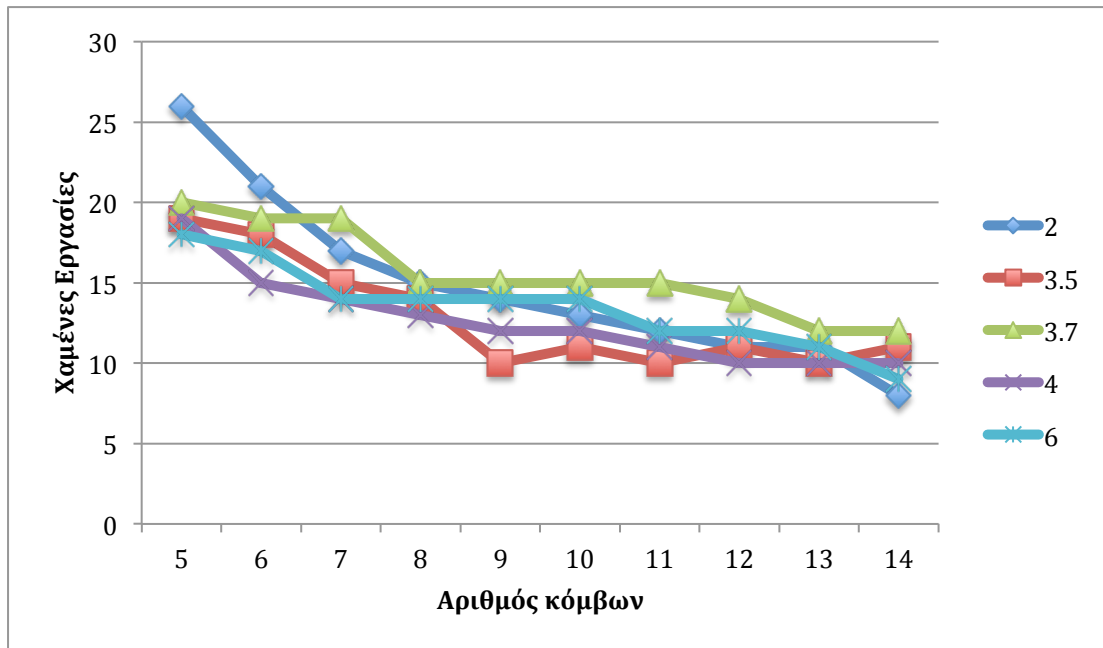
Μέγιστος Φόρτος: 0.166

Χαμένες: 2% - 40%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 10 εργασίες στο σύστημα, τελειώνοντας στις 100 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 2% μέχρι 40%. Οι πιθανότητες και σε αυτό το σενάριο συμπεριφέρθηκαν το ίδιο.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
50	5	3	10	10



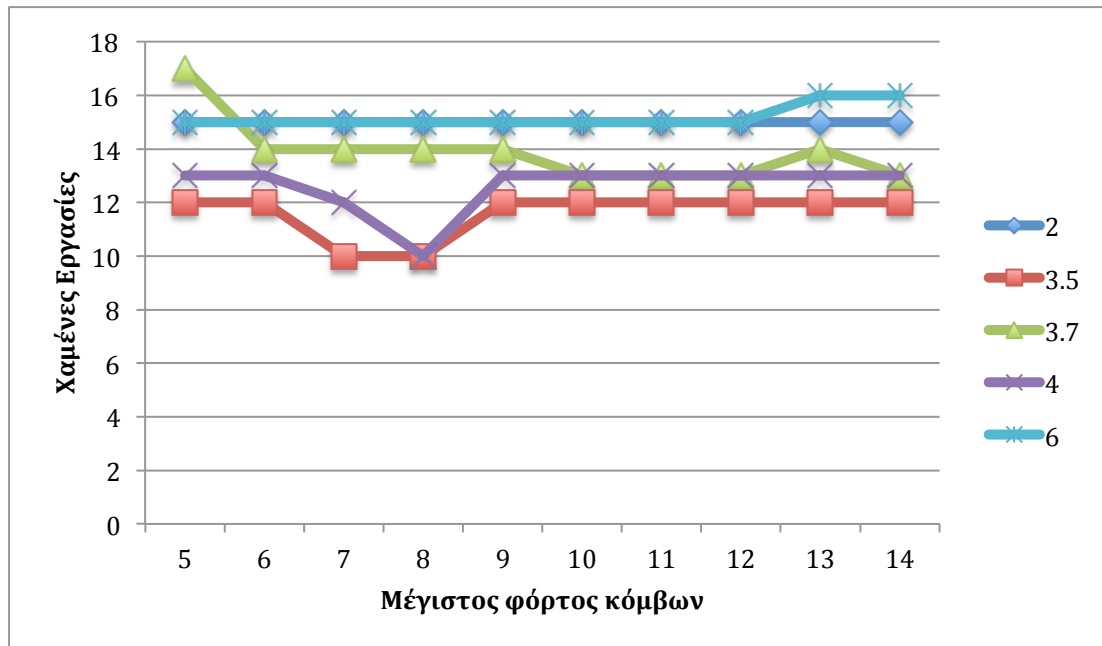
Μέγιστος Φόρτος 0.166

Χαμένες: 18% - 36%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 18% μέχρι 36%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με λιγότερους κόμβους, οι μεγάλες πιθανότητες (4,6) θα είχαν καλύτερα αποτελέσματα, αφού θα βελτίωναν την κατανομή των εργασιών στους κόμβους, αλλά και στους εκτελέσεις με περισσότερους κόμβους θα είχαν καλύτερα αποτελέσματα οι πιο μικρές πιθανότητες (2, 3.5). Όντως στα πρώτα σενάρια όπου ο αριθμός των κόμβων είναι μικρός έχει καλύτερα αποτελέσματα η πιθανότητα 6, ενώ έπειτα όπου ο αριθμός των κόμβων είναι μεγαλύτερος, η πιθανότητα 3.5 έχει καλύτερα αποτελέσματα.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
50	5	3	10	10



Μέγιστος Φόρτος: 0.166

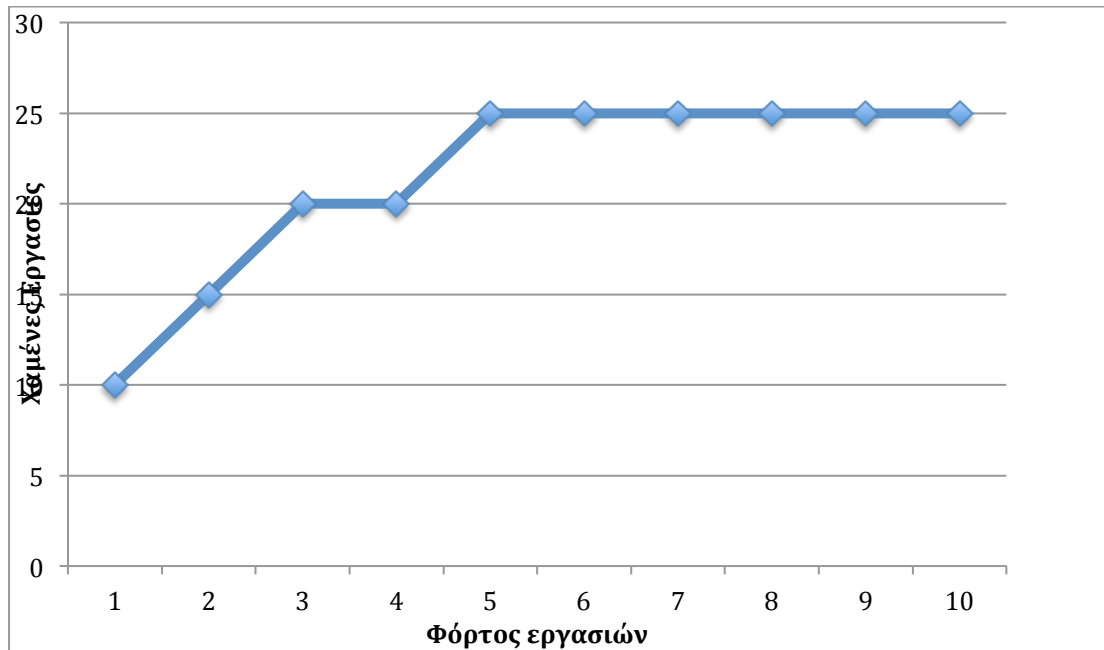
Χαμένες Εργασίες: 34% - 44%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών, ούτε οι διαφορετικές πιθανότητες στον αλγόριθμο έπαιξαν κάποιο ρόλο. Το ποσοστό χαμένων εργασιών κυμαίνεται από 34% μέχρι 44%.

Παρουσιάζονται οι εκτελέσεις με τον Αλγόριθμο 2

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
50	3	10	10	10



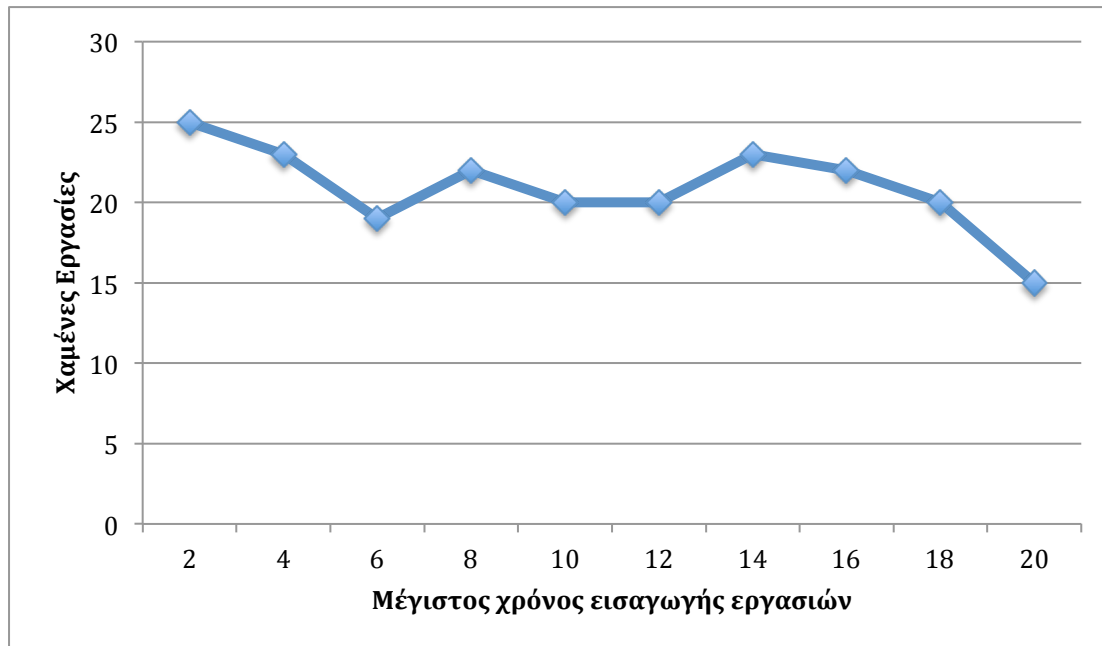
Μέγιστος Φόρτος: 0.166

Χαμένες Εργασίες: 20% - 50%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται ο φόρτος εργασίας, αυξάνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26% - 64%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
50	3	10	10	5



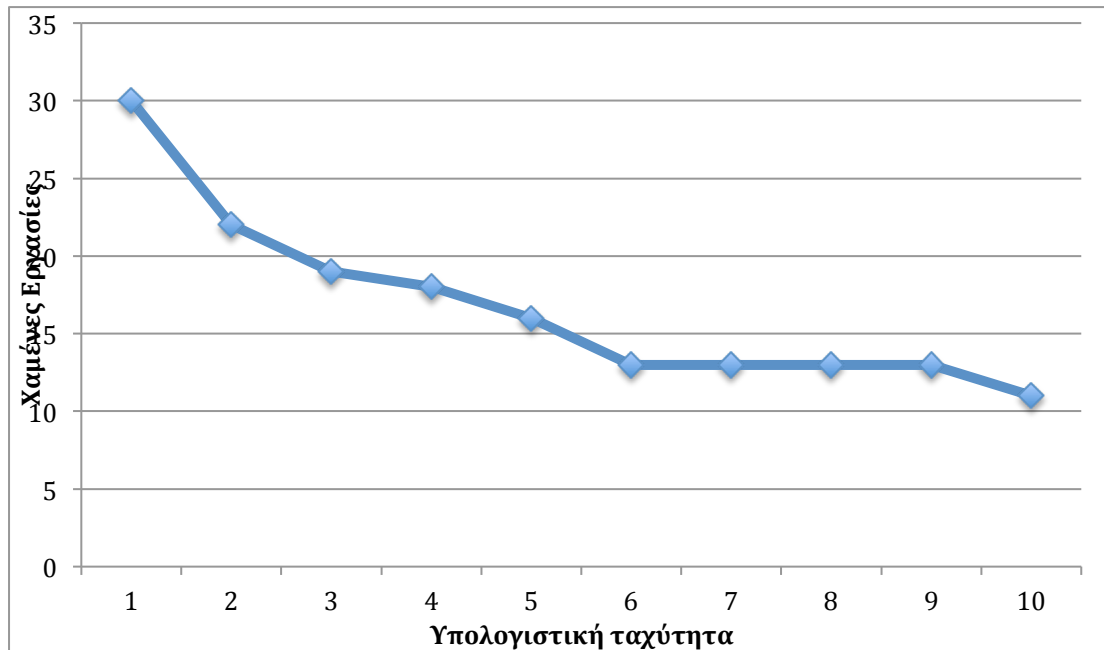
Μέγιστος Φόρτος: 0.416

Χαμένες Εργασίες: 30% - 50%

Σε αυτό το σενάριο αυξάνεται το μέγιστο όριο εισδοχής εργασιών, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται το όριο εισδοχής εργασιών, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 30% - 50%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
50	5	10	10	10



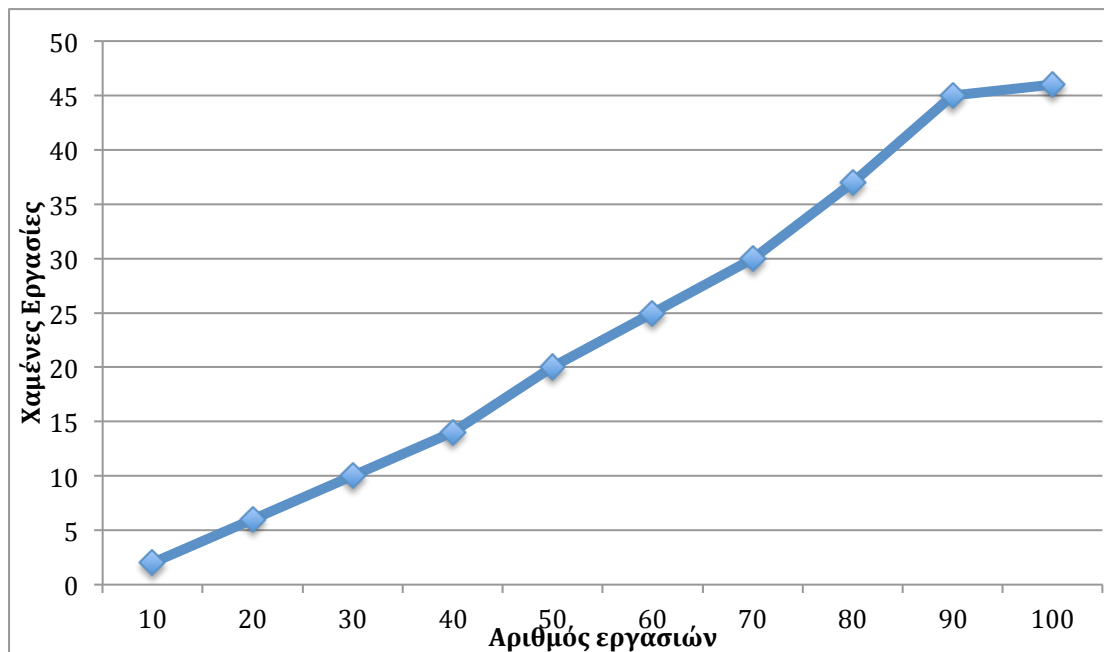
Μέγιστος Φόρτος: 0.25

Χαμένες Εργασίες: 22% - 60%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 22% μέχρι 60%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



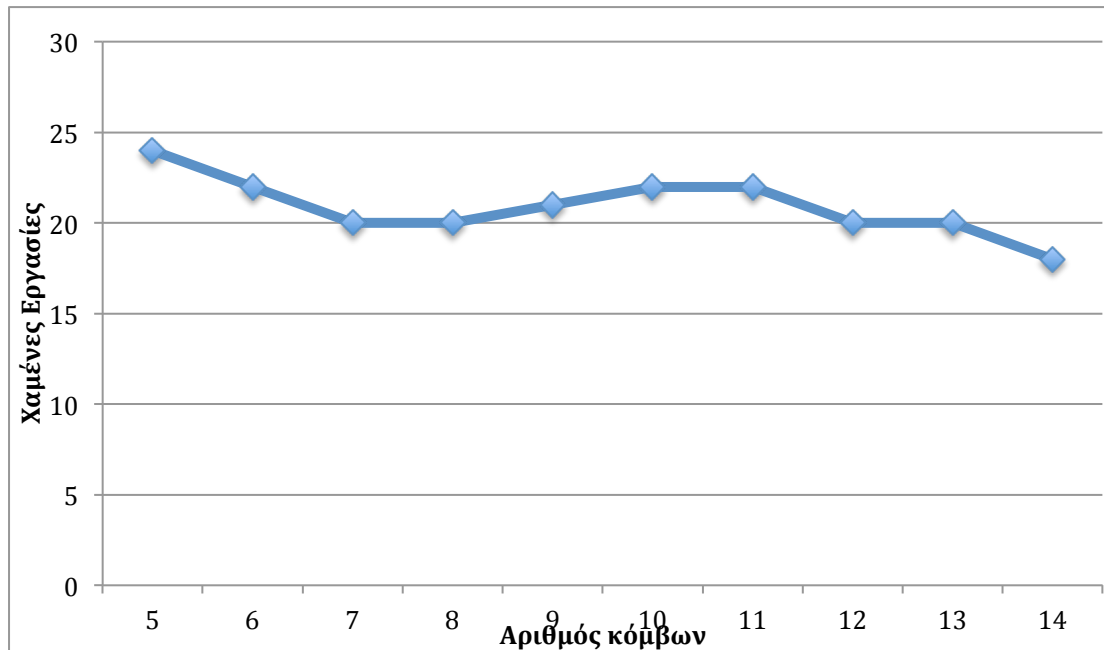
Μέγιστος Φόρτος: 0.166

Χαμένες Εργασίες: 2% - 40%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δέκα, αρχίζοντας με 10 εργασίες στο σύστημα, τελειώνοντας στις 100 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 2% μέχρι 40%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
50	5	3	10	10



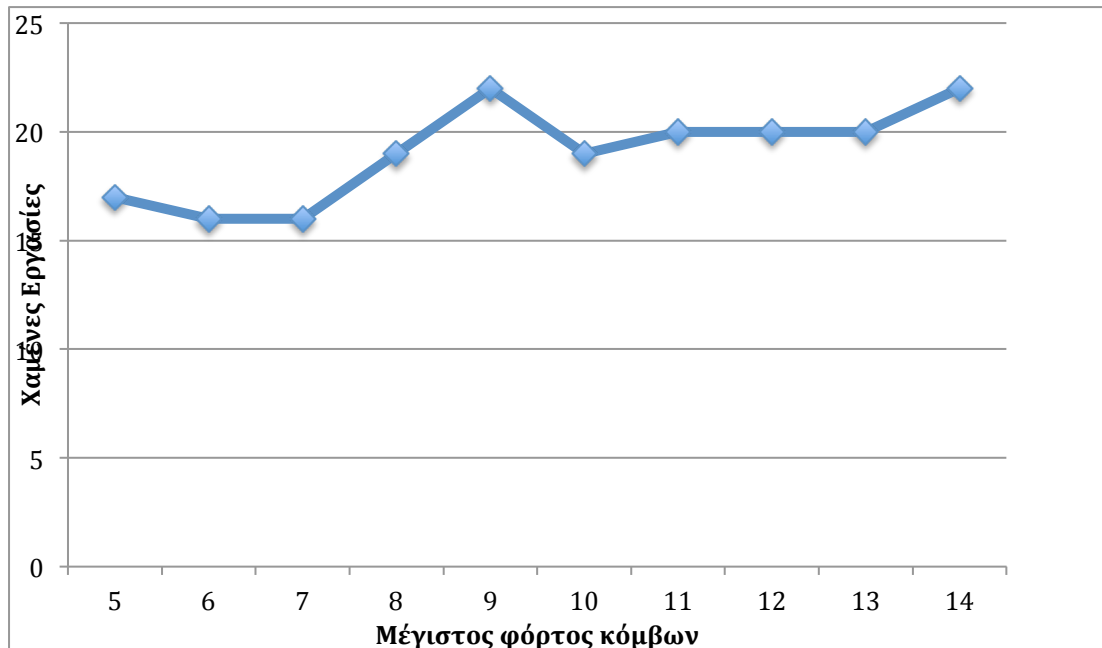
Μέγιστος Φόρτος: 0.166

Χαμένες Εργασίες: 36% - 48%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 38% - 68%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
50	5	3	10	10



Μέγιστος Φόρτος: 0.166

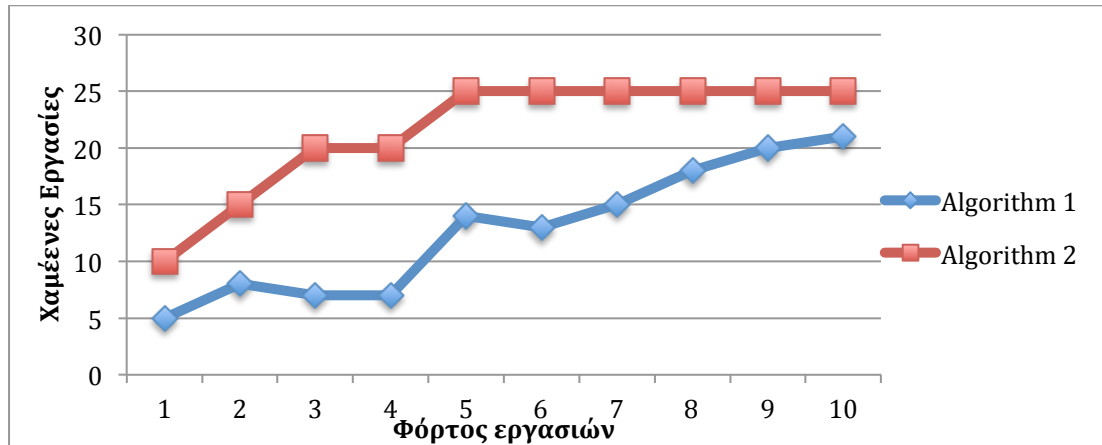
Χαμένες Εργασίες: 34% - 44%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Καθώς αυξάνεται η χωρητικότητα των κόμβων, θα έπρεπε να μειώνεται και ο αριθμός των χαμένων εργασιών. Παρόλα αυτά ο αριθμός των χαμένων εργασιών αυξάνεται. Αυτό μπορεί να συμβαίνει λόγω του ότι ο κάθε κόμβος αναλαμβάνει περισσότερες εργασίες από όσες μπορεί να διαχειριστεί. Το ποσοστό χαμένων εργασιών κυμαίνεται από 30% - 56%.

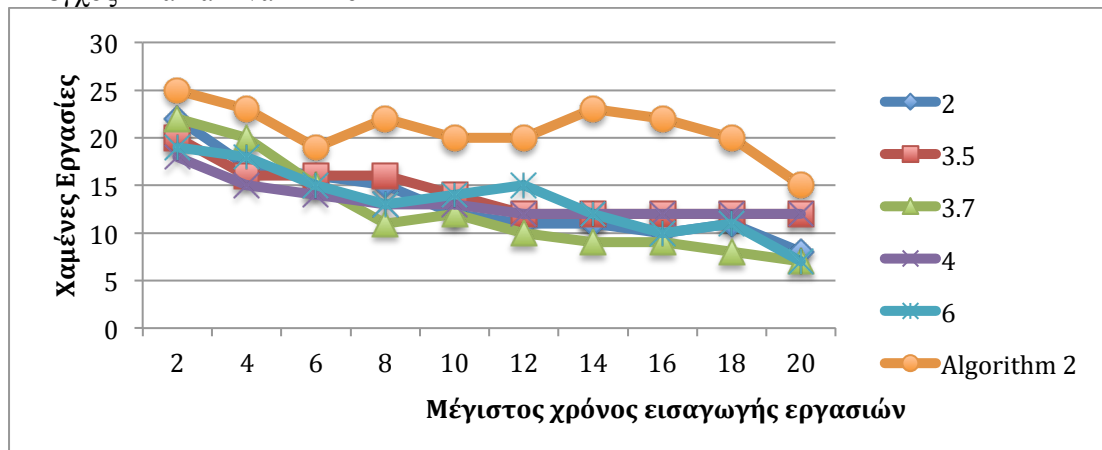
Σύγκριση Αλγορίθμων

Οι δύο αλγόριθμοι κινήθηκαν στα ίδια πλαίσια όσον αφορά το χρόνο εκτέλεσης των εργασιών. Η διαφορά τους έγκειται στον αριθμό χαμένων εργασιών, η οποία φαίνεται καλύτερα εάν αντιπαραβάλουμε τα αποτελέσματα των δύο αλγορίθμων μεταξύ τους.

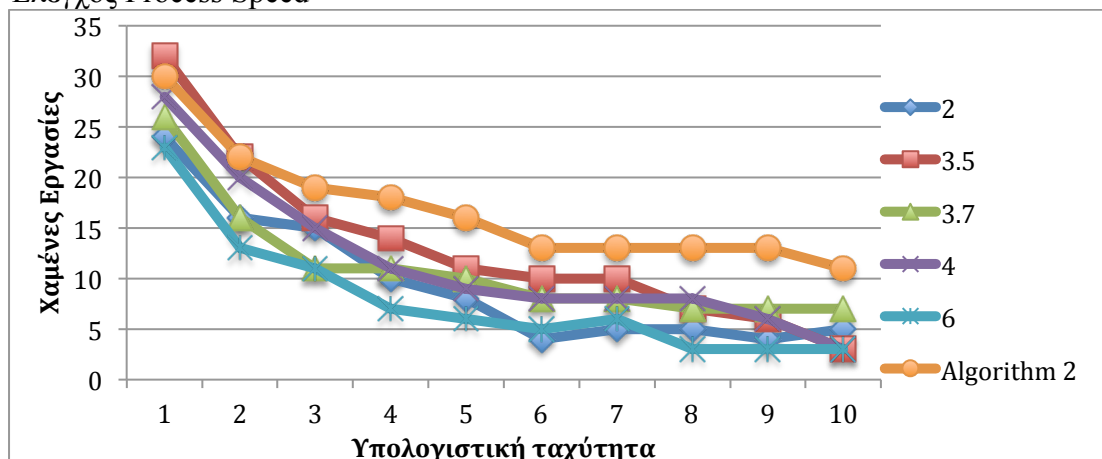
Έλεγχος Job Load



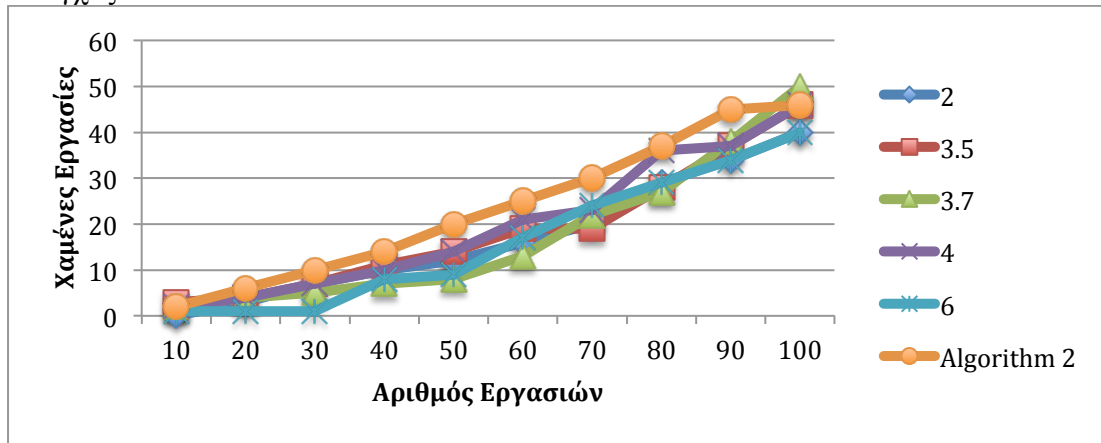
Έλεγχος Max arrival Time



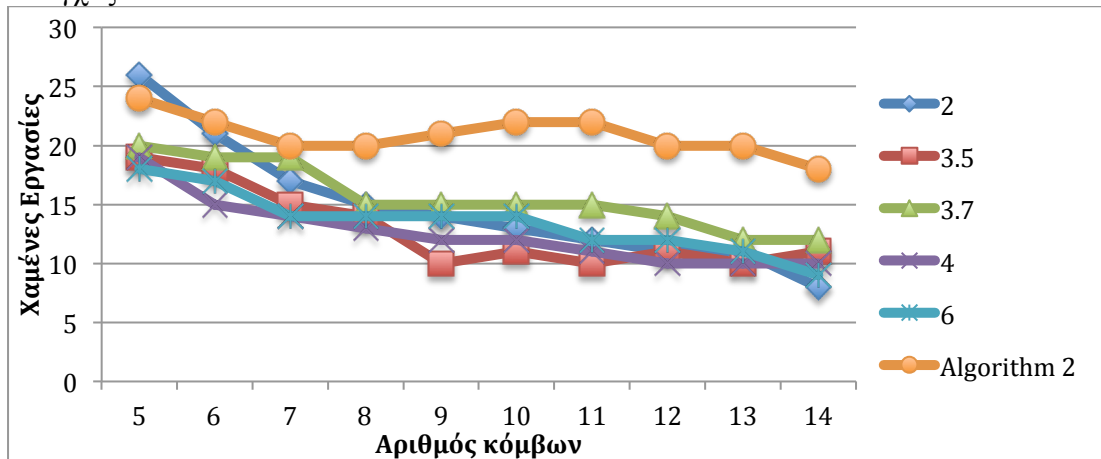
Έλεγχος Process Speed



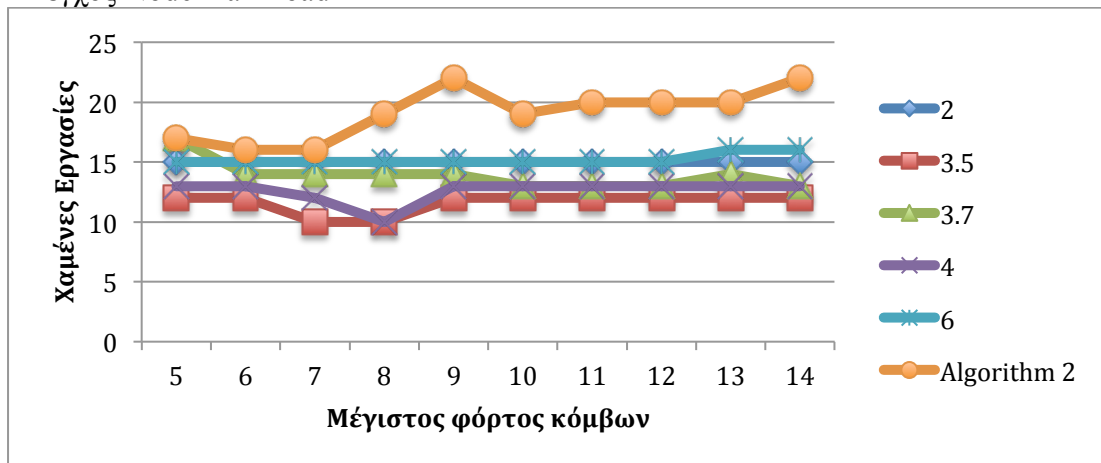
Έλεγχος Number of Jobs



Έλεγχος Number of Nodes



Έλεγχος Node max Load



Όπως φαίνεται καθαρά από τις γραφικές παραστάσεις, ο Αλγόριθμος 1 υπερτερεί. Όλες οι πιθανότητες του Αλγορίθμου 1 φέρουν καλύτερα αποτελέσματα εαν τις συγκρίνουμε με τα αποτελέσματα του Αλγορίθμου 2.

5.4.2 Μέτρια Φορτωμένα Σενάρια

Τα μέτρια φορτωμένα σενάρια έχουν συντελεστή μέχρι και 0.1

Παρουσιάζονται πρώτα οι εκτελέσεις με τον Αλγόριθμο 1.

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
75	3	10	10	10



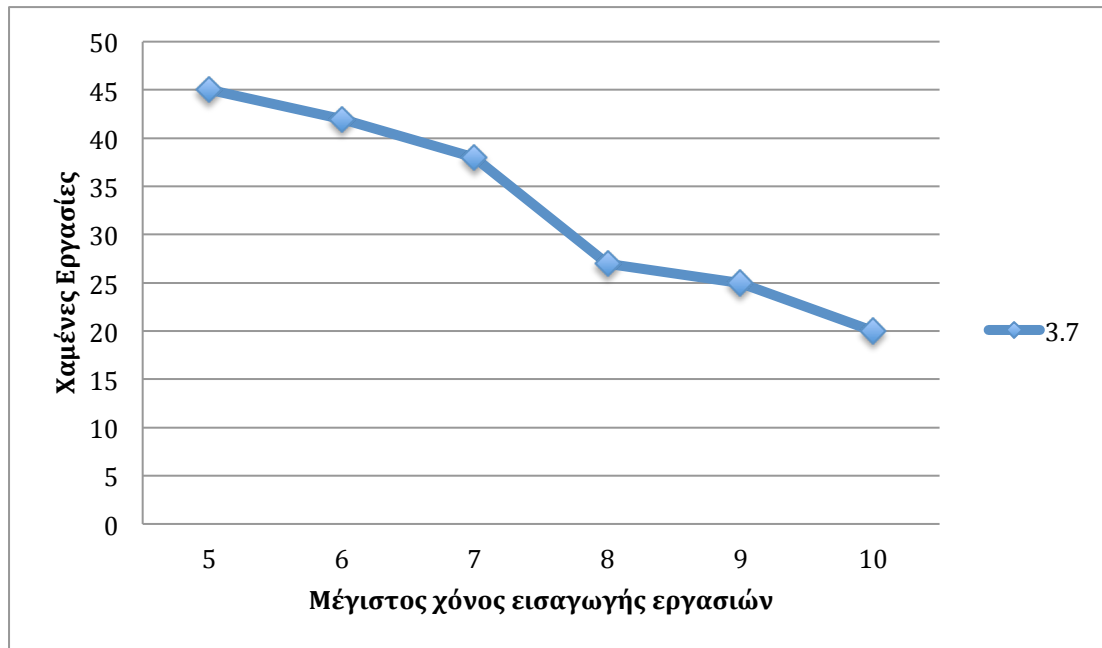
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 8% - 66.6%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως ο αριθμός χαμένων εργασιών αυξάνεται σταδιακά. Το ποσοστό χαμένων εργασιών κυμαίνεται από 8% μέχρι 66.6%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
75	3	10	10	5



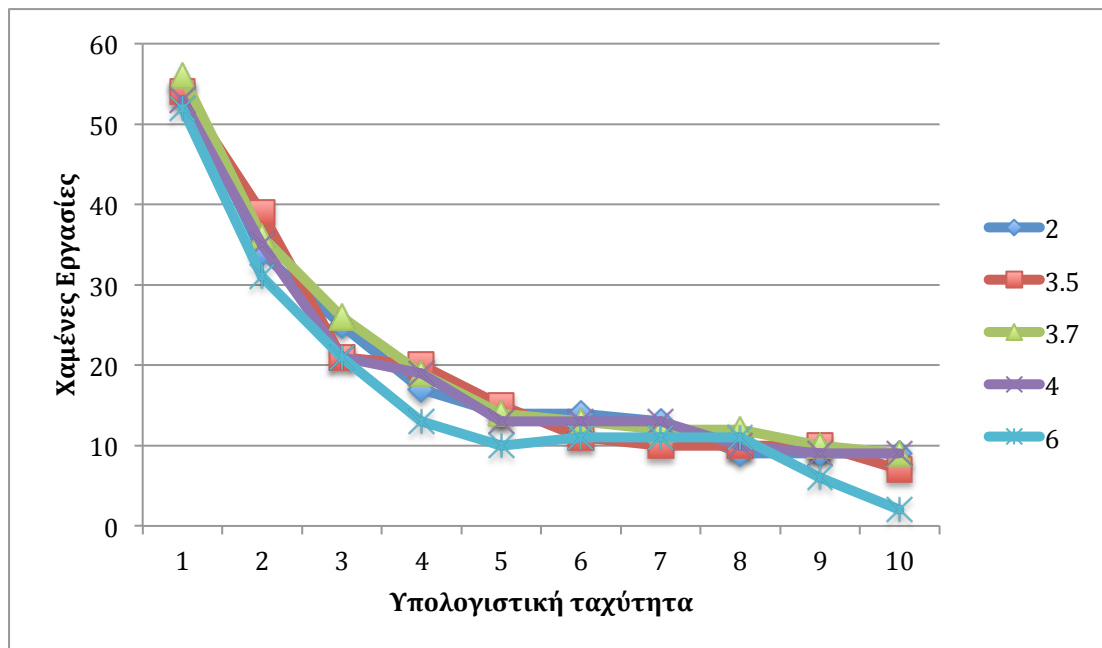
Μέγιστος Φόρτος: 0.624

Χαμένες Εργασίες: 26.66% - 60%

Σε αυτό το σενάριο μεταβάλαμε το μέγιστο χρονικό όριο εισδοχής εργασιών. Το όρο κυμαίνεται από 5 μέχρι 15. Όπως περιμέναμε, καθώς αυξάνεται το όριο εισδοχής εργασιών, ο αριθμός χαμένων εργασιών μειώνεται, αφού οι εργασίες εισέρχονται πιο αραιά και το σύστημα μπορεί να τις χειριστεί καλύτερα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26.66% - 60%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
75	5	10	10	10



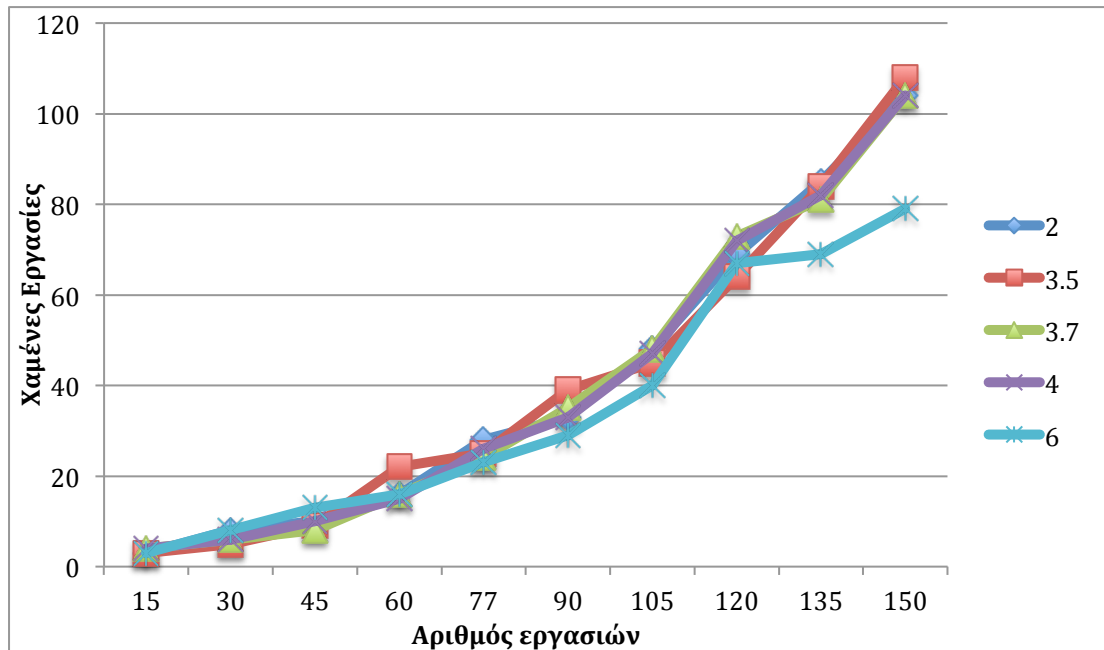
Μέγιστος Φόρτος: 0.375

Χαμένες Εργασίες: 2.6% - 69.33%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 2.6% - 69.33%. Σε αυτό το σενάριο, όπως και πριν περιμέναμε πως στις εκτελέσεις με μικρή υπολογιστική ταχύτητα οι μεγάλες πιθανότητες θα είχαν καλύτερα αποτελέσματα, αφού θα επέτρεπαν καλύτερη κατανομή των εργασιών σε αδύνατους υπολογιστικά κόμβους, αλλά και το αντίθετο οι μικρές πιθανότητες θα έπρεπε να είχαν καλύτερα αποτελέσματα στις εκτελέσεις με μεγάλη υπολογιστική ταχύτητα, αφού οι κόμβοι θα είχαν την δυνατότητα να χειριστούν ευκολότερα τις εργασίες χωρίς κάποια κατανομή. Παρόλα αυτά η πιθανότητα 6 έχει σε όλες τις εκτελέσεις καλύτερα αποτελέσματα.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



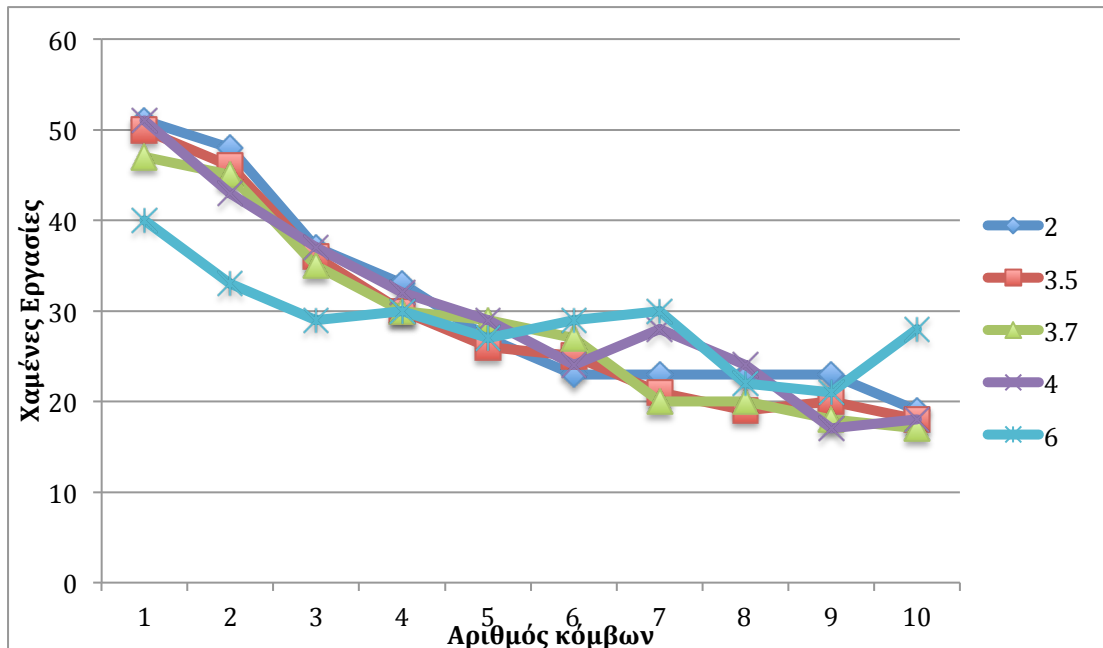
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 26.66% - 52.66%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά δεκαπέντε, αρχίζοντας με 15 εργασίες στο σύστημα, τελειώνοντας στις 150 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26.66% μέχρι 52.66%. Η πιθανότητα 6 έφερε τα καλύτερα αποτελέσματα.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
75	5	3	10	10



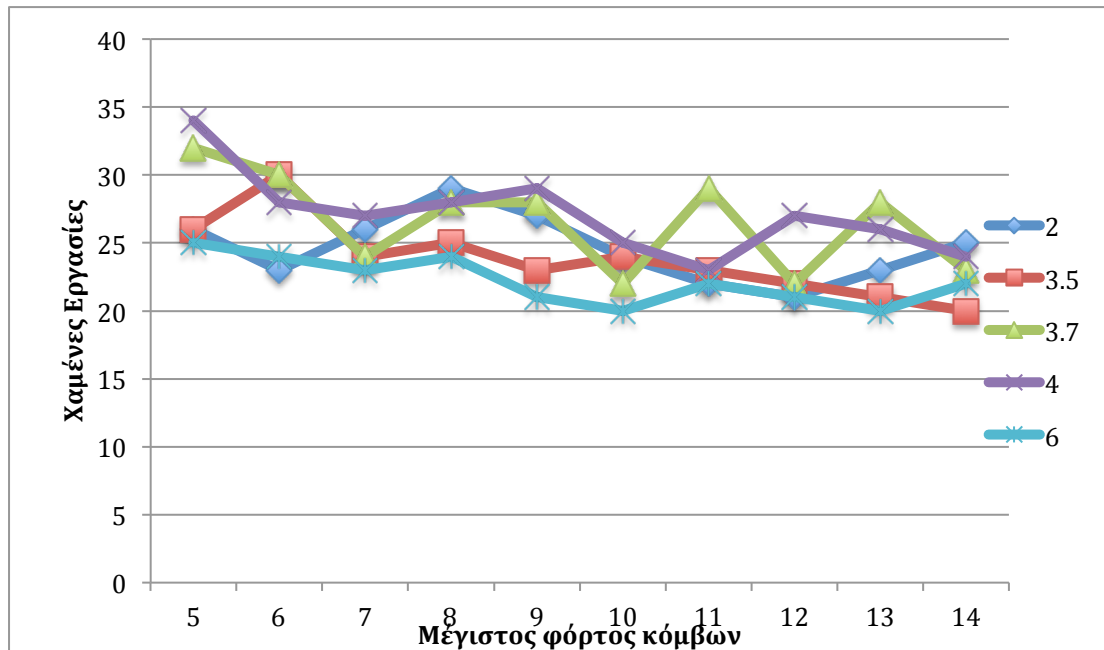
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 28% - 53.33%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 28%-53.33%. Σε αυτό το σενάριο περιμέναμε πως στις εκτελέσεις με λιγότερους κόμβους, οι μεγάλες πιθανότητες (4,6) θα είχαν καλύτερα αποτελέσματα, αφού θα βελτίωναν την κατανομή των εργασιών στους κόμβους. Οι πιθανότητες όμως, συμπεριφέρθηκαν με παρόμοιο τρόπο σε όλες τις εκτελέσεις.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
75	5	3	10	10



Μέγιστος Φόρτος: 0.249

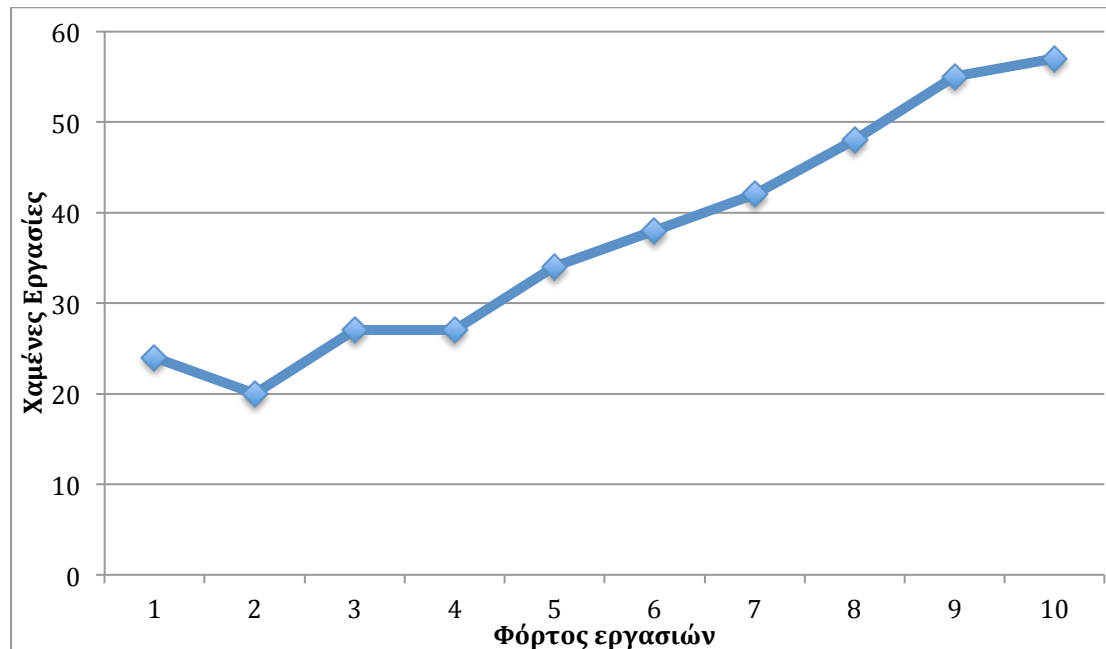
Χαμένες Εργασίες: 26.66% - 33.33%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό έχει ελάχιστο αντίκτυπο στον αριθμό χαμένων εργασιών. Η πιθανότητα 6 έχει τα καλύτερα αποτελέσματα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26.66% - 33.33%.

Παρουσιάζονται οι εκτελέσεις με τον Αλγόριθμο 2

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
75	3	10	10	10



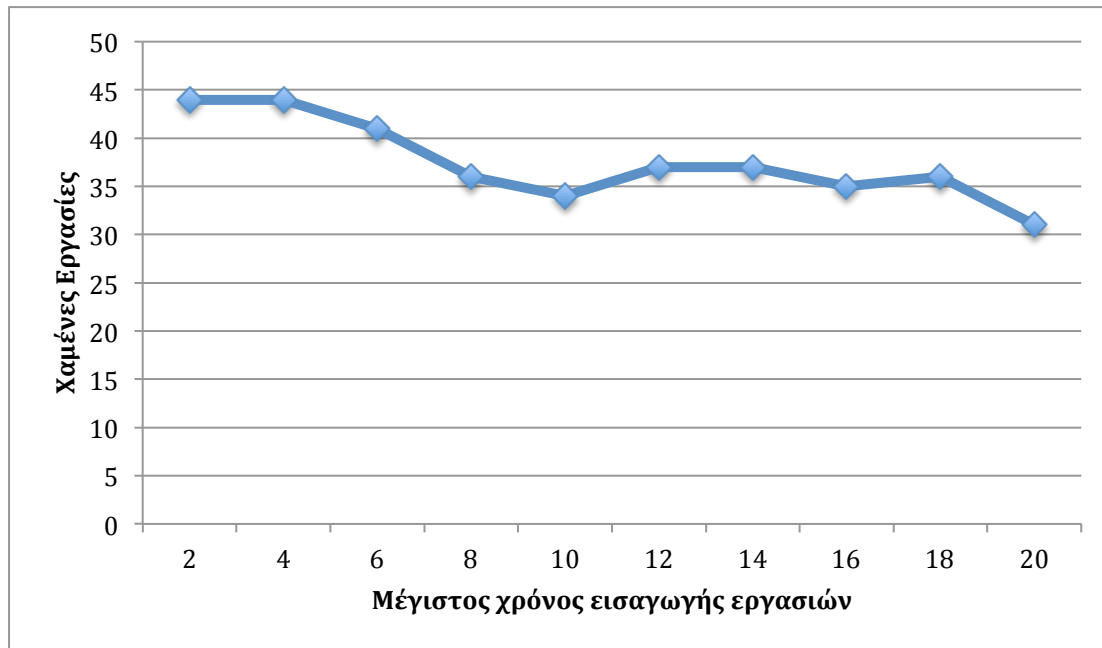
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 26.66% - 76%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται ο φόρτος εργασίας, αυξάνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 26.66% - 76%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
75	3	10	10	5



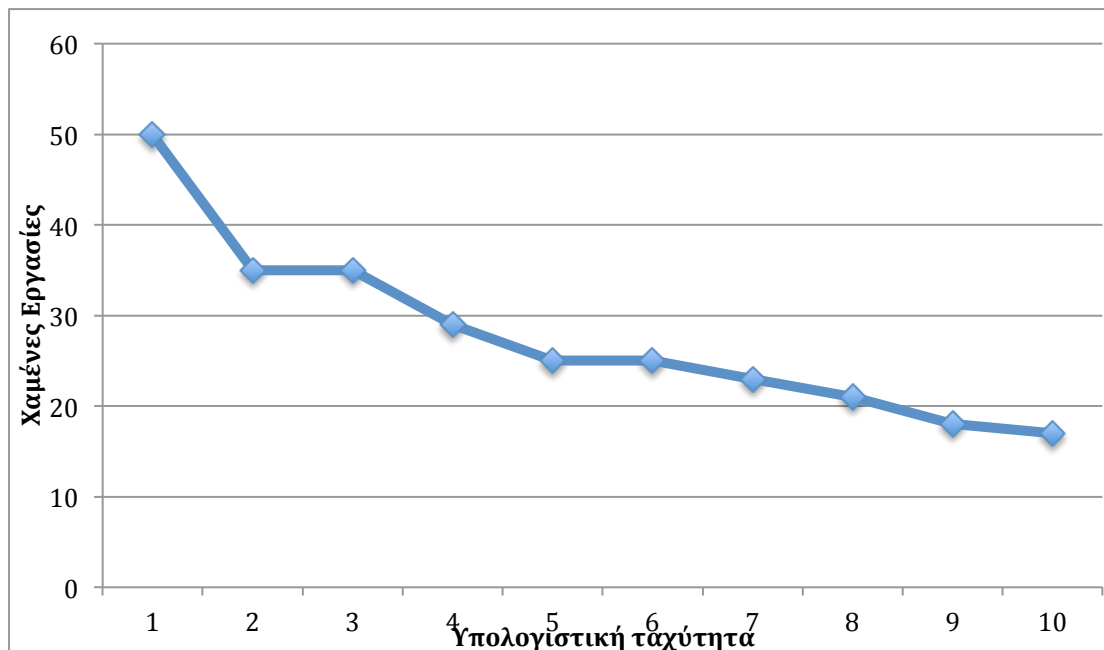
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 41.33% - 58.66%

Σε αυτό το σενάριο αυξάνεται το μέγιστο όριο εισδοχής εργασιών, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται το όριο εισδοχής εργασιών, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 41.33% - 58.66%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
75	5	10	10	10



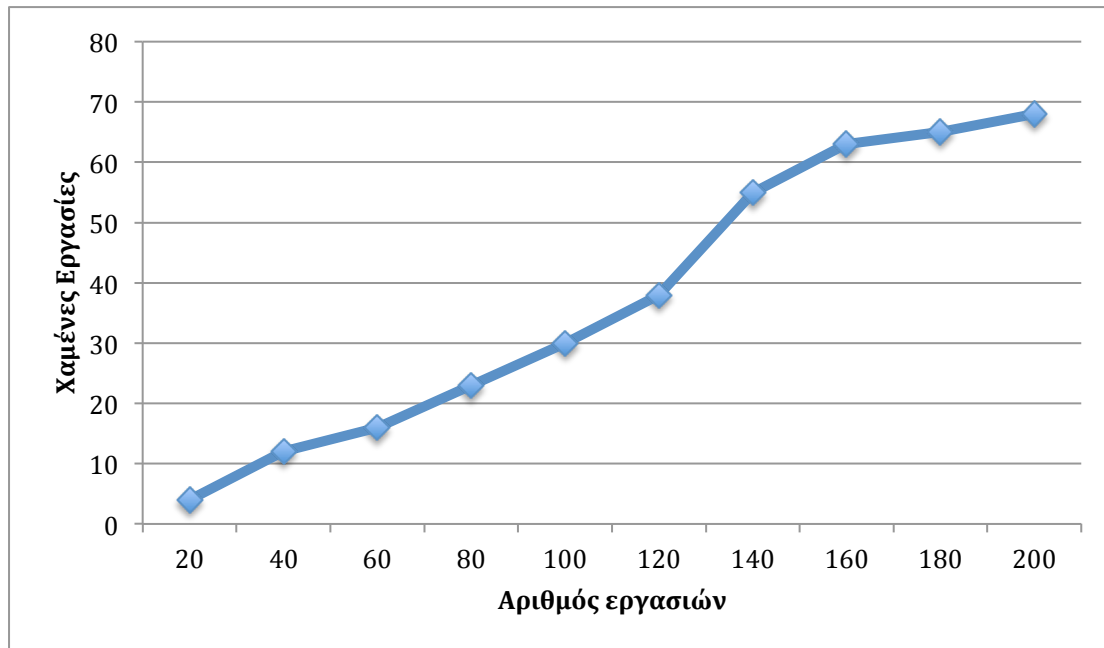
Μέγιστος Φόρτος: 0.375

Χαμένες Εργασίες: 22.66% - 66.66%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 22.66% - 66.66%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



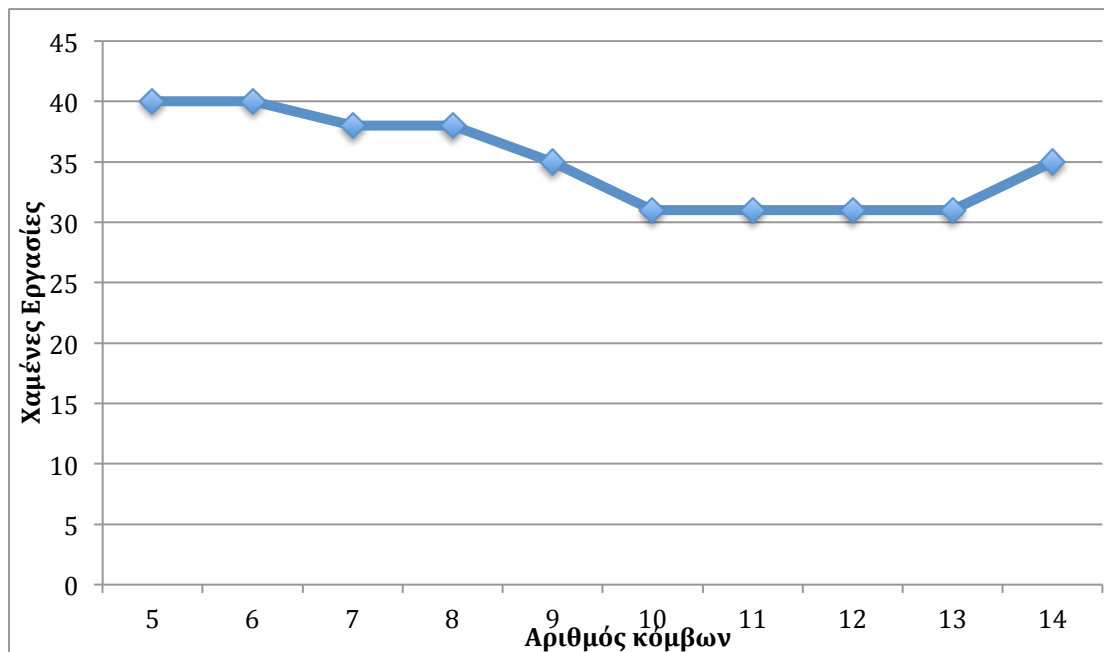
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 5.33% - 39.28%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά είκοσι, αρχίζοντας με 20 εργασίες στο σύστημα, τελειώνοντας στις 200 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 5.33% - 39.28%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
75	5	3	10	10



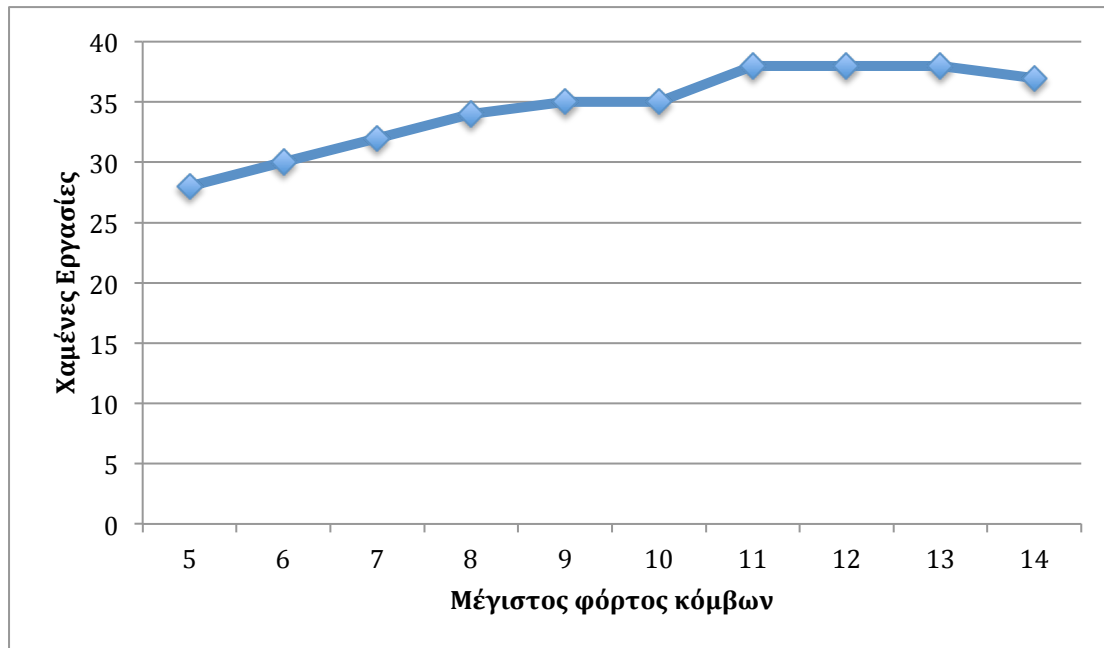
Μέγιστος Φόρτος: 0.249

Χαμένες Εργασίες: 41.33% - 53.33%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 41.33% - 53.33%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
75	5	3	10	10



Μέγιστος Φόρτος: 0.249

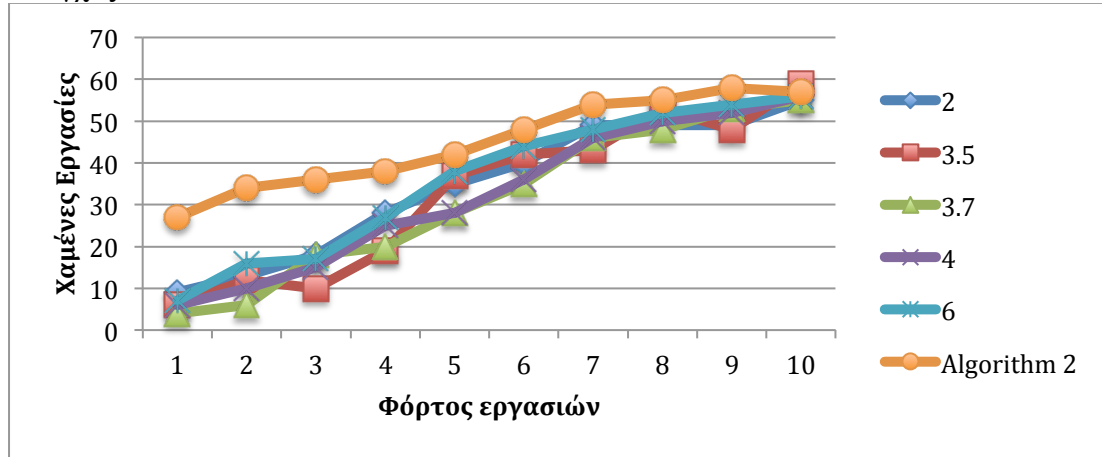
Χαμένες Εργασίες: 37.33% - 50.66%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 37.33% - 50.66%.

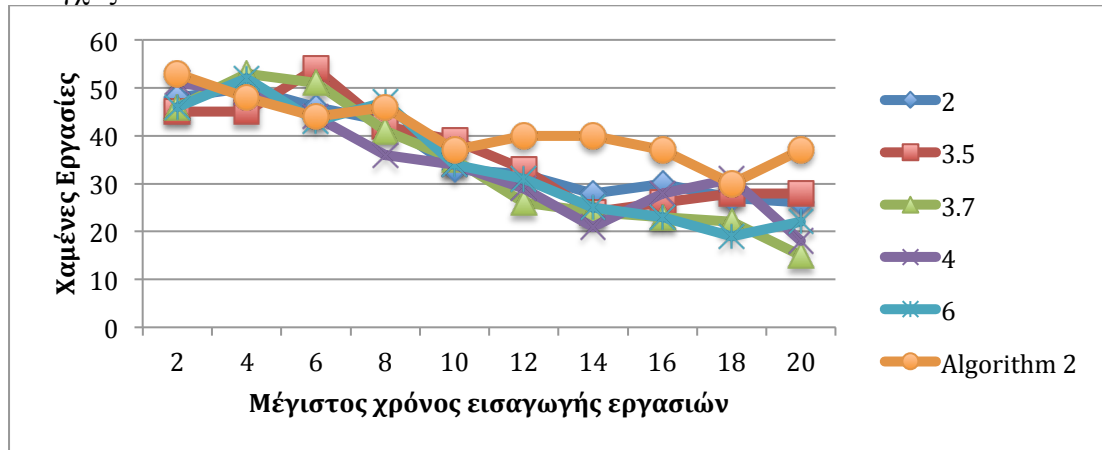
Σύγκριση Αλγορίθμων

Οι δύο αλγόριθμοι κινήθηκαν στα ίδια πλαίσια όσον αφορά το χρόνο εκτέλεσης των εργασιών. Η διαφορά τους έγκειται στον αριθμό χαμένων εργασιών, η οποία φαίνεται καλύτερα εάν αντιπαραβάλουμε τα αποτελέσματα των δύο αλγορίθμων μεταξύ τους.

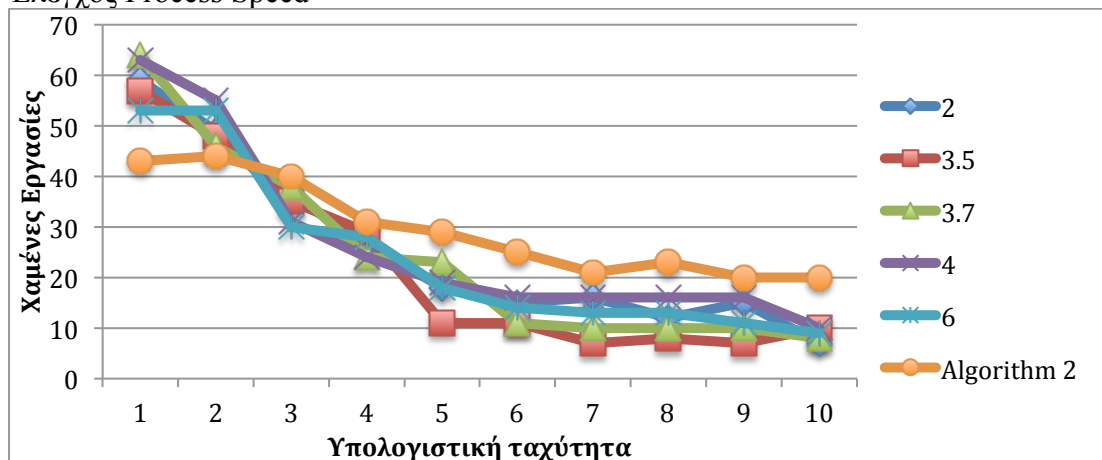
Έλεγχος Job Load



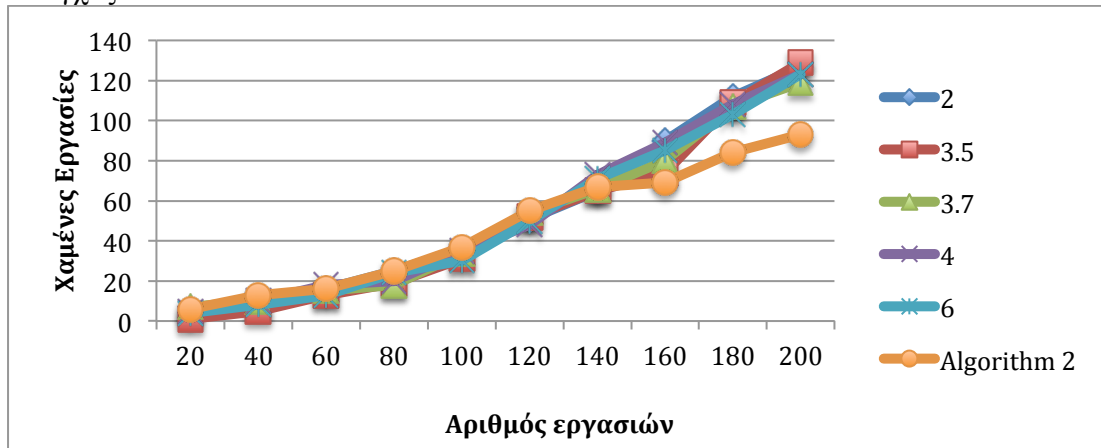
Έλεγχος Max arrival



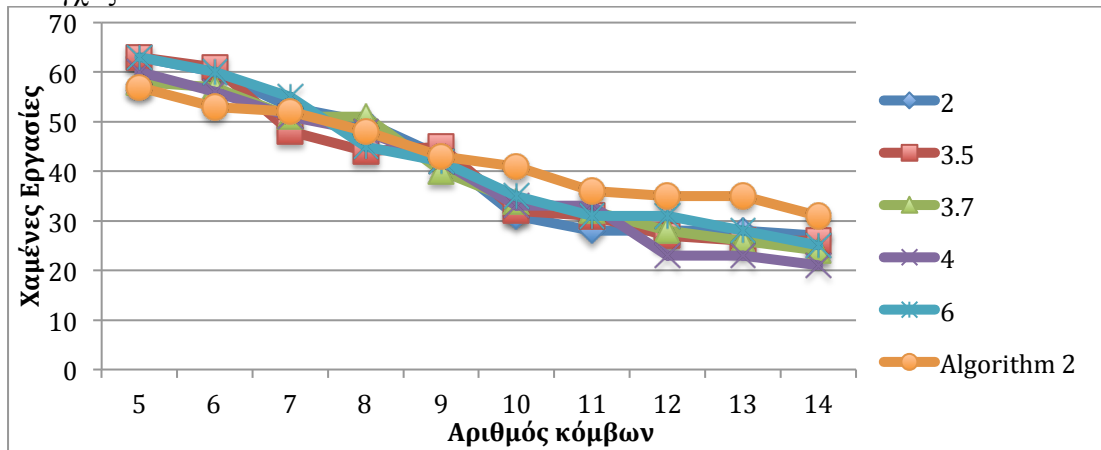
Έλεγχος Process Speed



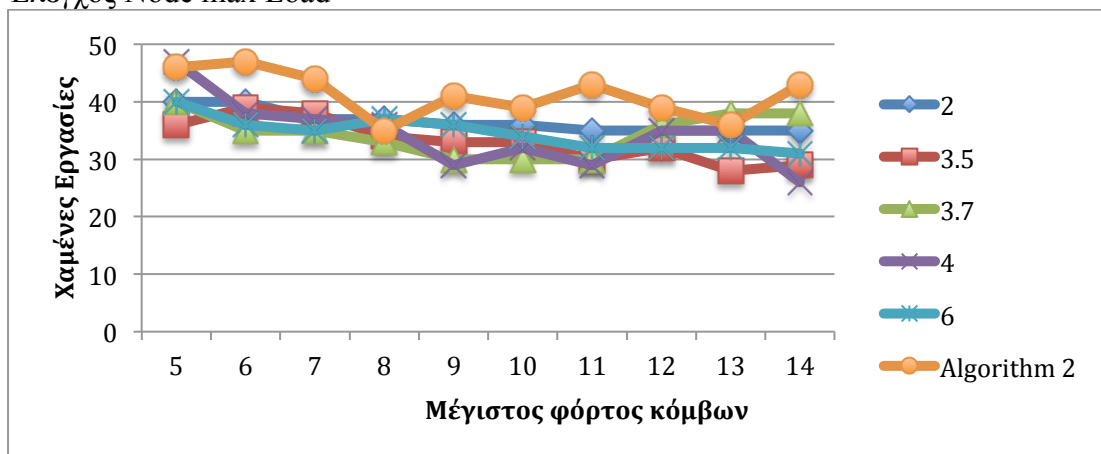
Έλεγχος Number of Jobs



Έλεγχος Number of Nodes



Έλεγχος Node max Load



Στα μέτρια φορτωμένα σενάρια φαίνεται καθαρά από τις γραφικές παραστάσεις πως ο Αλγόριθμος 1 υπερτερεί με εξαίρεση το σενάριο στο οποίο μεταβάλλεται ο αριθμός των εργασιών (σενάριο 4). Όλες οι πιθανότητες του Αλγορίθμου 1 φέρουν καλύτερα αποτελέσματα εάν τις συγκρίνουμε με τα αποτελέσματα του Αλγορίθμου 2.

5.4.3 Βαριά Φορτωμένα Σενάρια

Τα βαριά φορτωμένα σενάρια έχουν συντελεστή μέχρι και 0.832.

Παρουσιάζονται πρώτα οι εκτελέσεις με τον Αλγόριθμο 1.

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
100	3	10	10	10



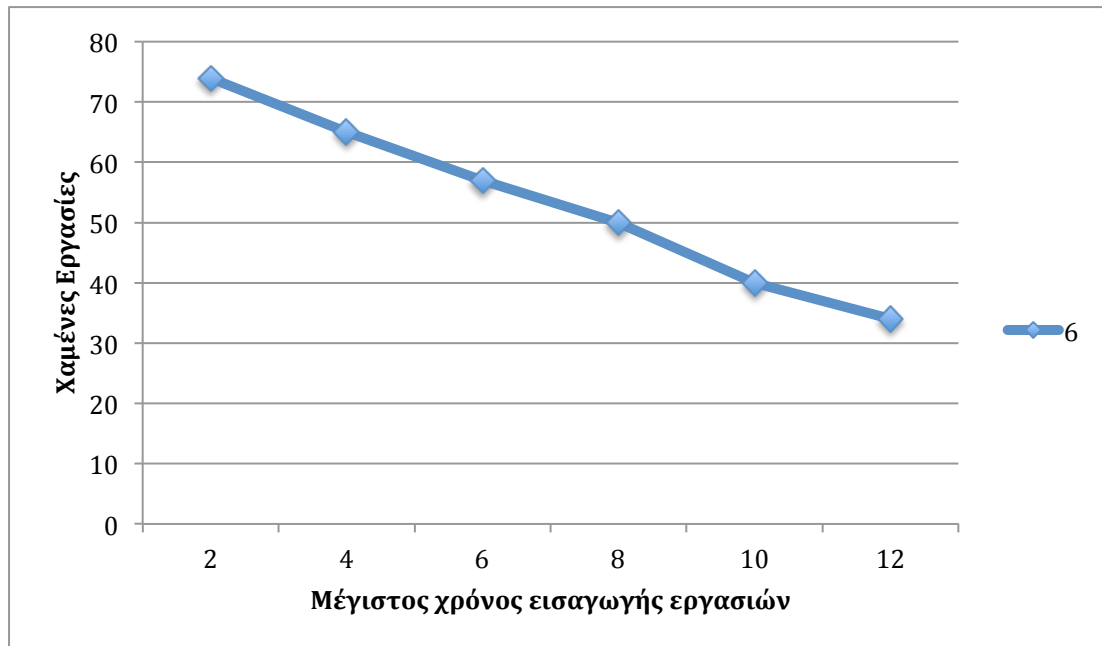
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 9% - 75%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως ο αριθμός χαμένων εργασιών αυξάνεται σταδιακά. Το ποσοστό χαμένων εργασιών κυμαίνεται από 9% - 75%. Η πιθανότητα που έφερε τα καλύτερα αποτελέσματα είναι η 6.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
100	3	10	10	5



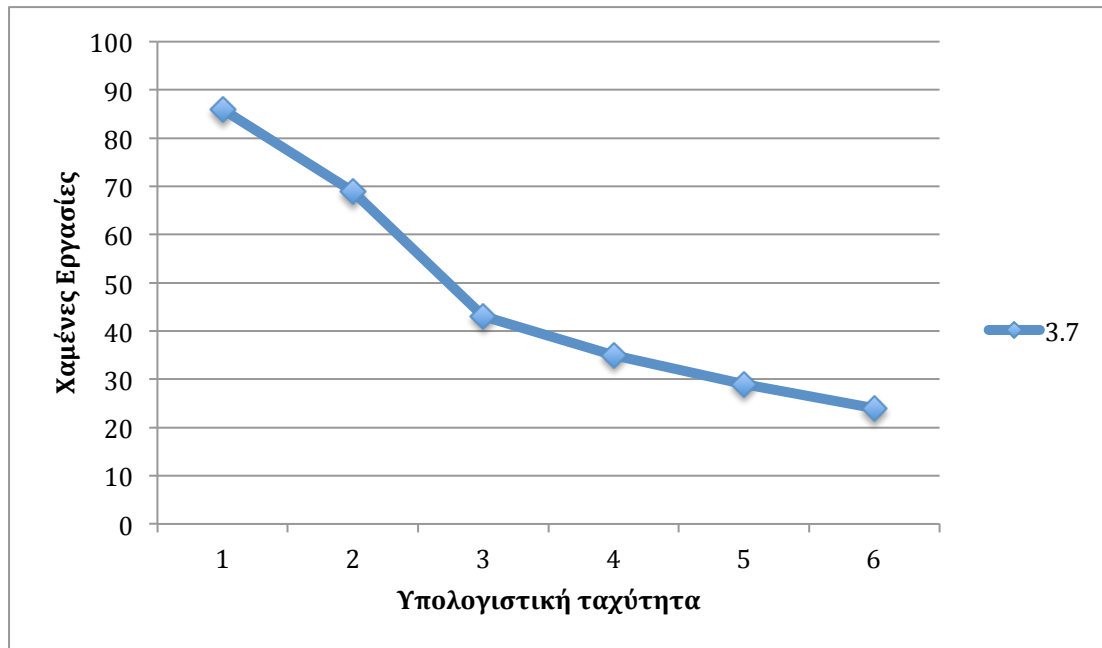
Μέγιστος Φόρτος: 0.832

Χαμένες Εργασίες: 34% - 74%

Σε αυτό το σενάριο μεταβάλαμε το μέγιστο χρονικό όριο εισδοχής εργασιών. Το όρο κυμαίνεται από 2 μέχρι 20. Όπως περιμέναμε, καθώς αυξάνεται το όριο εισδοχής εργασιών, ο αριθμός χαμένων εργασιών μειώνεται, αφού οι εργασίες εισέρχονται πιο αραιά και το σύστημα μπορεί να τις χειριστεί καλύτερα. Το ποσοστό χαμένων εργασιών κυμαίνεται από 34% - 74%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
100	5	10	10	10



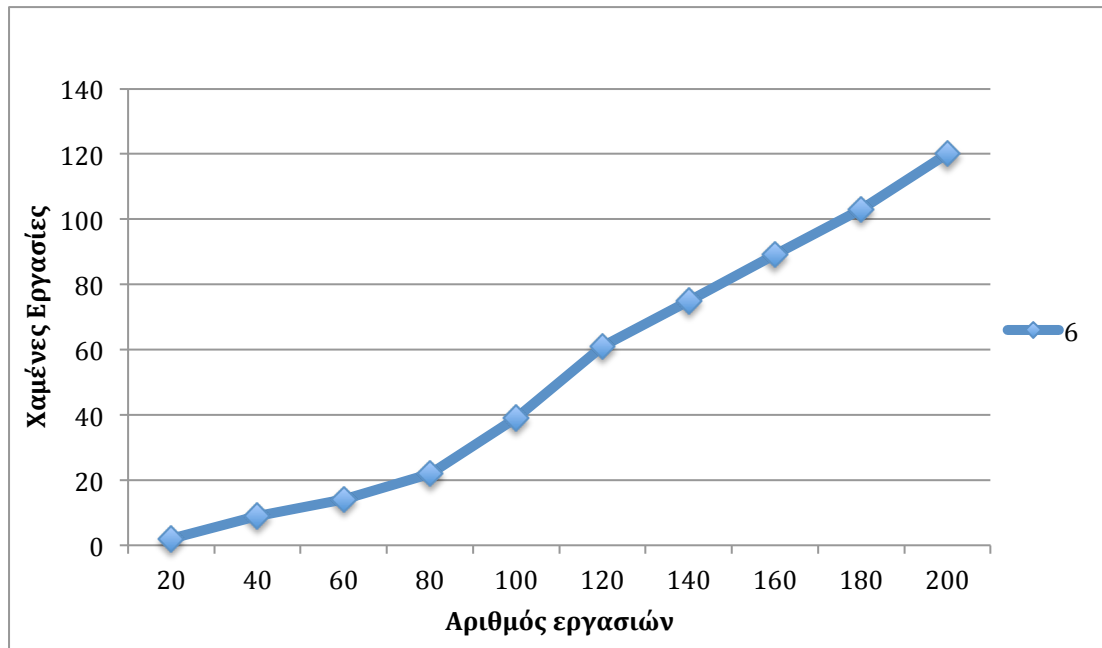
Μέγιστος Φόρτος: 0.5

Χαμένες Εργασίες: 24% - 86%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 24% - 86%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



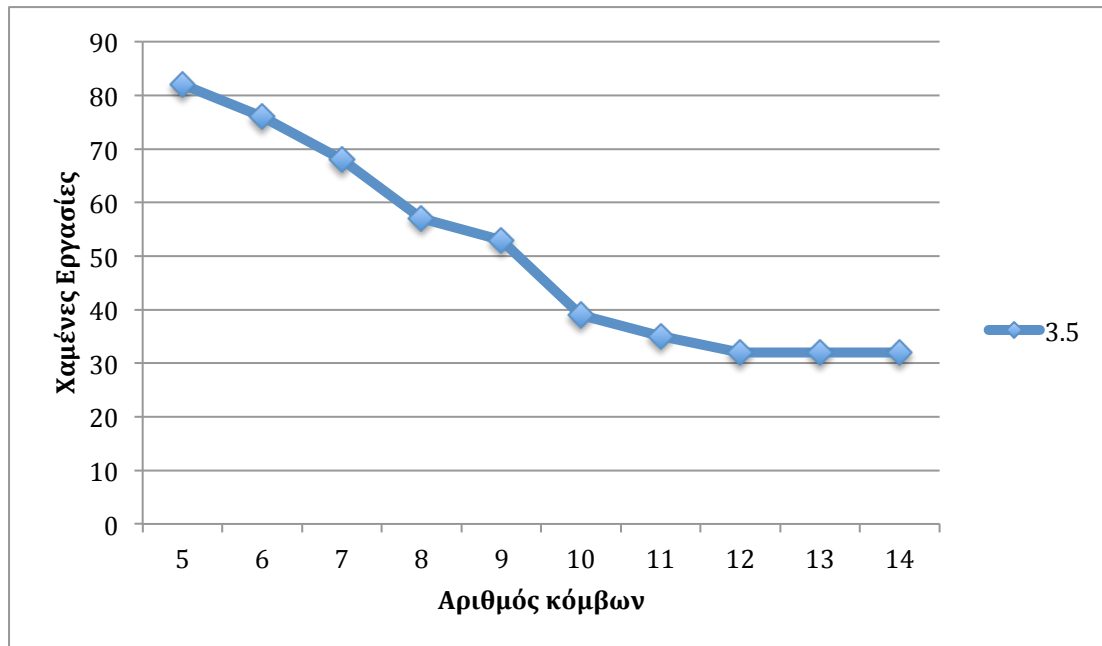
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 10% - 60%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά είκοσι, αρχίζοντας με 20 εργασίες στο σύστημα, τελειώνοντας στις 200 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 10% - 60%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
100	5	3	10	10



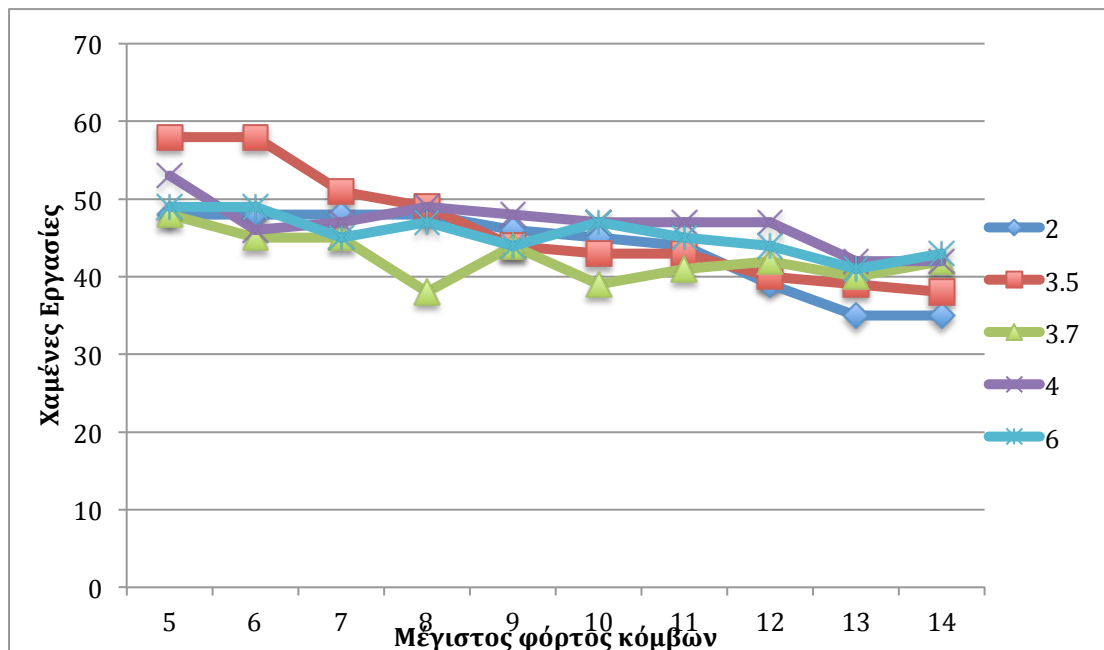
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 32% - 82%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 32% - 82%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
100	5	3	10	10



Μέγιστος Φόρτος: 0.332

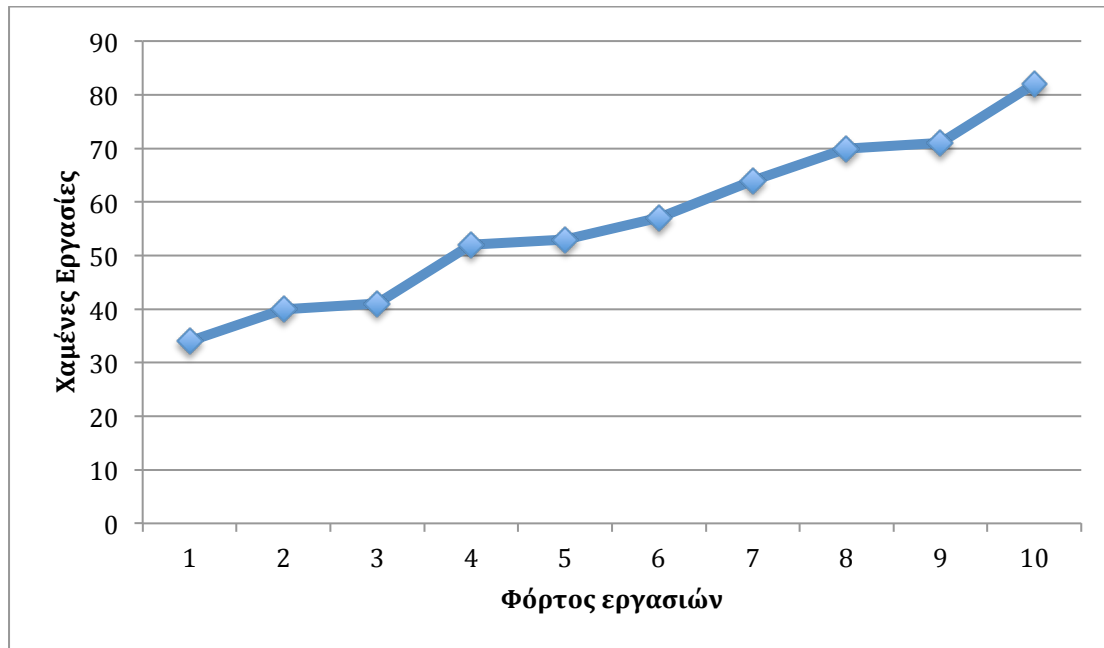
Χαμένες Εργασίες: 38% - 49%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει μεγάλο αντίκτυπο στον αριθμό χαμένων εργασιών, ούτε οι διαφορετικές πιθανότητες στον αλγόριθμο έπαιξαν κάποιο ρόλο. Το ποσοστό χαμένων εργασιών κυμαίνεται από 38% - 49%.

Παρουσιάζονται οι εκτελέσεις με τον Αλγόριθμο 2

Σενάριο 1: Έλεγχος Job Load

# of Jobs	Process speed	# of Nodes	Node max Load	Max arrival Time
100	3	10	10	10



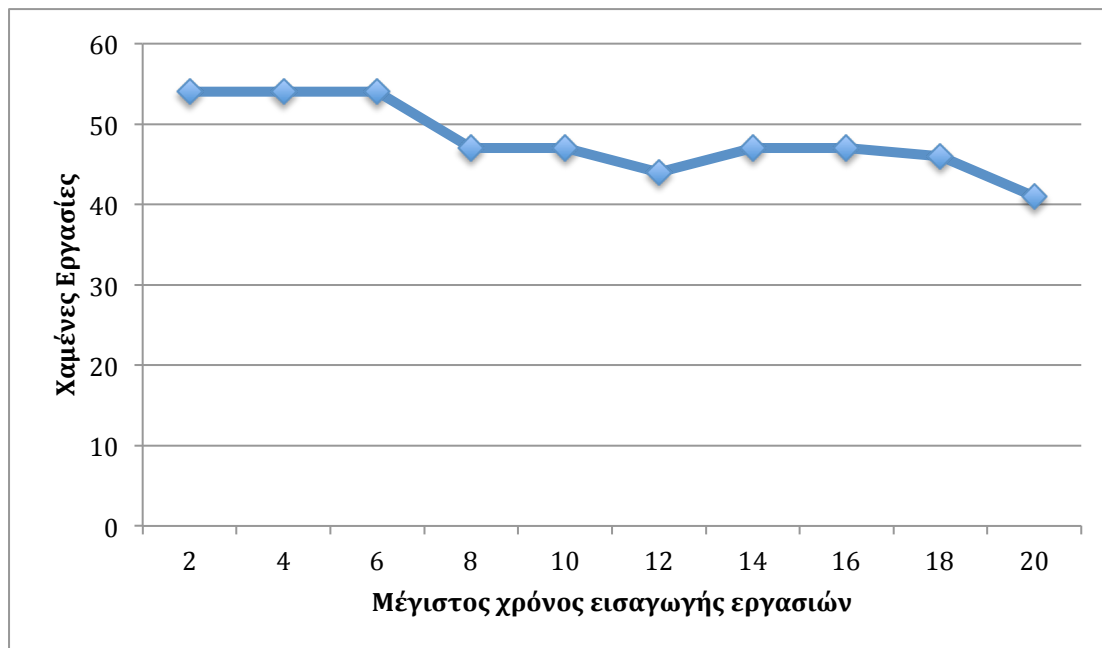
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 34% - 82%

Σε αυτό το σενάριο αυξάνεται σταδιακά ο φόρτος εργασίας, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται ο φόρτος εργασίας, αυξάνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 34% - 82%.

Σενάριο 2: Έλεγχος Maximum Arrival Time

# of Jobs	Process speed	# of Nodes	Node max Load	Job Load
100	3	10	10	5



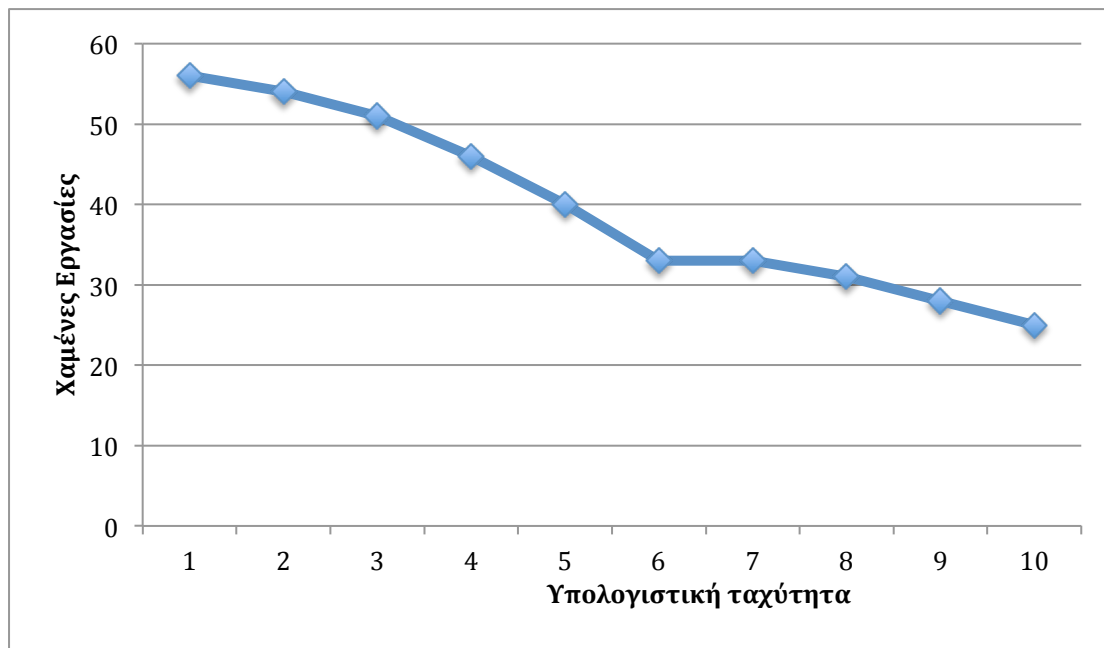
Μέγιστος Φόρτος: 0.832

Χαμένες Εργασίες: 41% - 54%

Σε αυτό το σενάριο αυξάνεται το μέγιστο όριο εισδοχής εργασιών, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Καθώς αυξάνεται το όριο εισδοχής εργασιών, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 41% - 54%.

Σενάριο 3: Έλεγχος Process Speed

# of Jobs	Job Load	# of Nodes	Node max Load	Max arrival Time
100	5	10	10	10



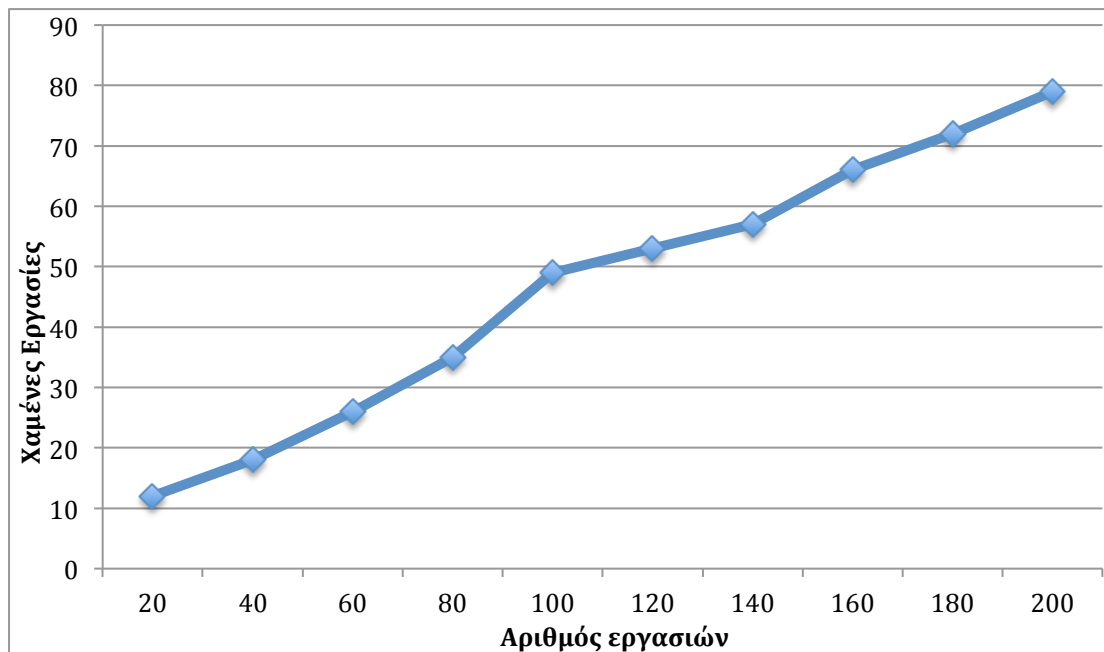
Μέγιστος Φόρτος: 0.5

Χαμένες Εργασίες: 25% - 56%

Σε αυτό το σενάριο αυξάνεται σταδιακά η υπολογιστική ταχύτητα, κρατώντας σταθερές τις υπόλοιπες μεταβλητές. Ως αποτέλεσμα βλέπουμε πως καθώς αυξάνεται η υπολογιστική ταχύτητα, μειώνεται και ο αριθμός χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 25% - 56%.

Σενάριο 4: Έλεγχος Number of Jobs

Process Speed	Job Load	# of Nodes	Node max Load	Max arrival Time
3	5	10	10	10



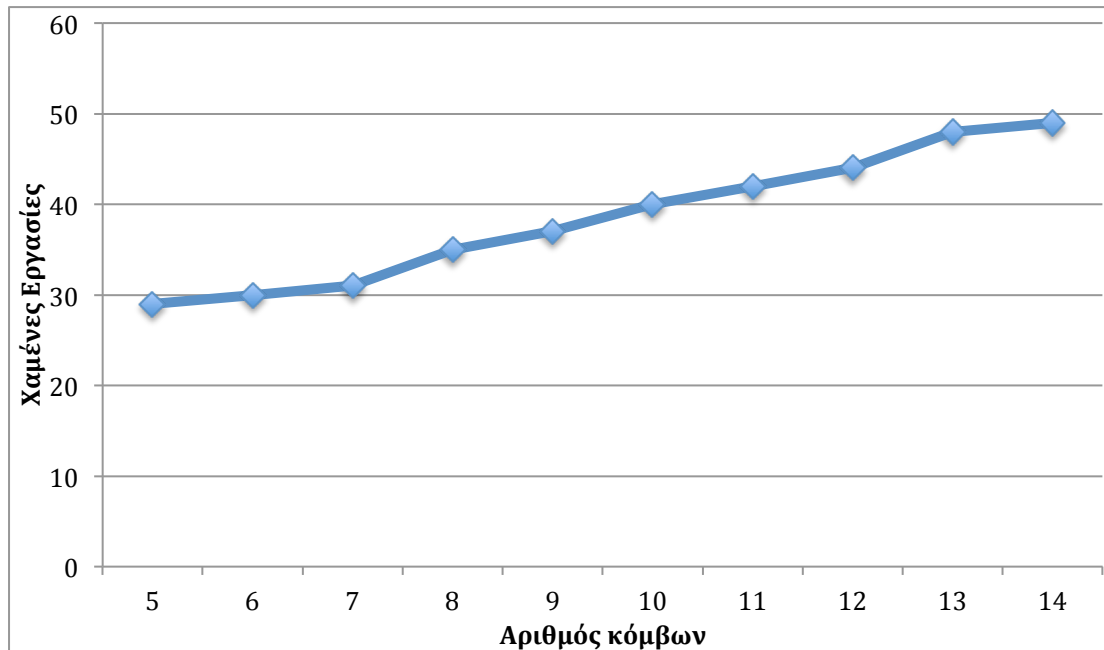
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 39.5% - 60%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση των εργασιών. Σε κάθε περίπτωση ο αριθμός των εργασιών αυξάνεται κατά είκοσι, αρχίζοντας με 20 εργασίες στο σύστημα, τελειώνοντας στις 200 εργασίες. Όπως είναι λογικό με την αύξηση των εργασιών αυξάνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 39.5% - 60%.

Σενάριο 5: Έλεγχος Number of Nodes

# of Jobs	Job Load	Process Speed	Node max Load	Max arrival Time
100	5	3	10	10



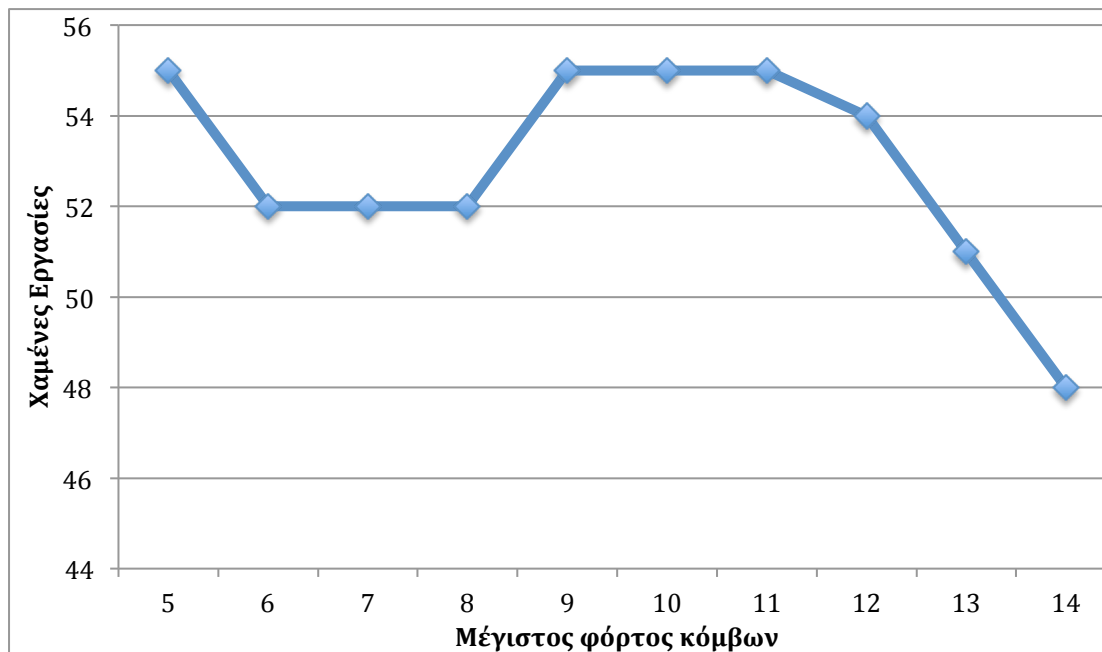
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 32% - 82%

Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα η αύξηση του αριθμού των κόμβων. Σε κάθε περίπτωση ο αριθμός των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στους 14 κόμβους. Από τα αποτελέσματα βλέπουμε πως αυτό έχει αντίκτυπο στον αριθμό χαμένων εργασιών. Καθώς αυξάνεται ο αριθμός των κόμβων, μειώνεται και ο αριθμός των χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 32% - 82%.

Σενάριο 6: Έλεγχος Node Maximum Load

# of Jobs	Job Load	Process Speed	# of Nodes	Max arrival Time
100	5	3	10	10



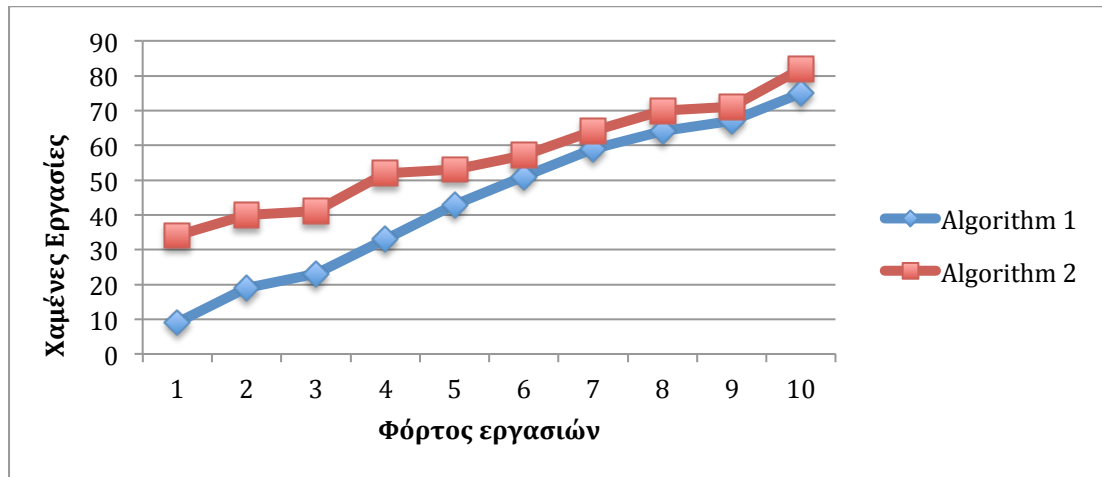
Μέγιστος Φόρτος: 0.332

Χαμένες Εργασίες: 48% - 55%

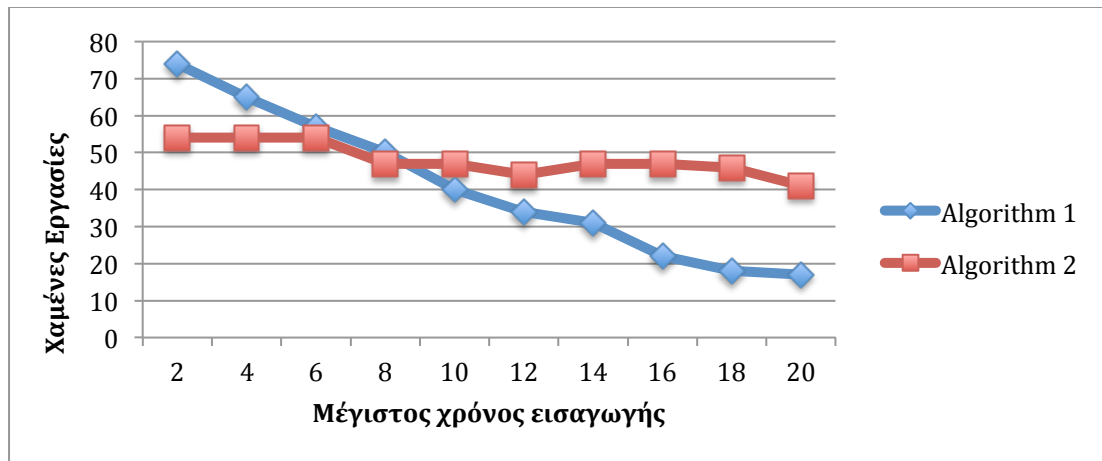
Στο σενάριο αυτό ελέγχεται πόσο επηρεάζει το σύστημα ο μέγιστος φόρτος των κόμβων. Σε κάθε περίπτωση ο αριθμός φόρτου των κόμβων αυξάνεται κατά ένα, αρχίζοντας με 5 και τελειώνοντας στο 14. Από τα αποτελέσματα βλέπουμε πως αυτό δεν έχει μεγάλο αντίκτυπο στον αριθμό χαμένων εργασιών. Το ποσοστό χαμένων εργασιών κυμαίνεται από 48% - 55%.

Σύγκριση Αλγορίθμων

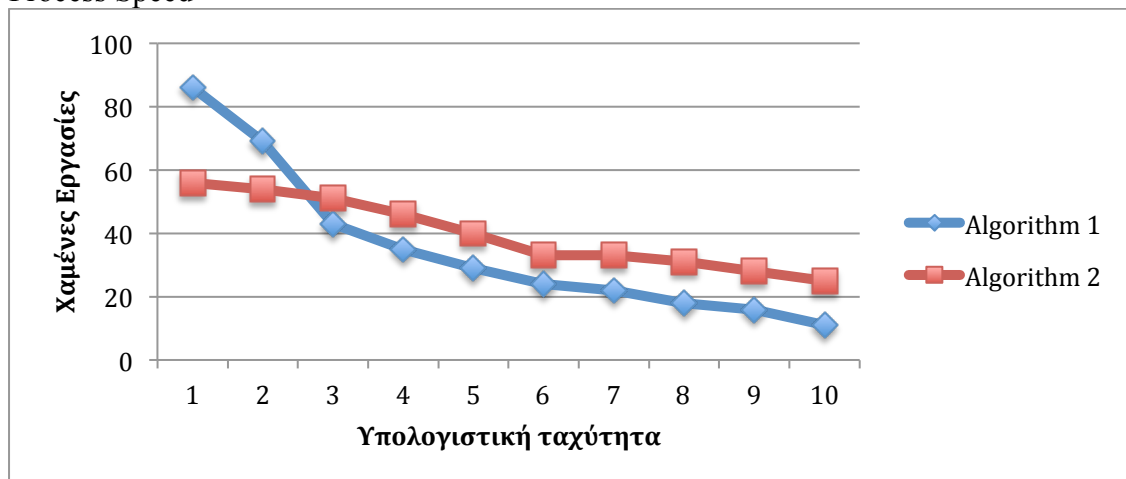
Job Load



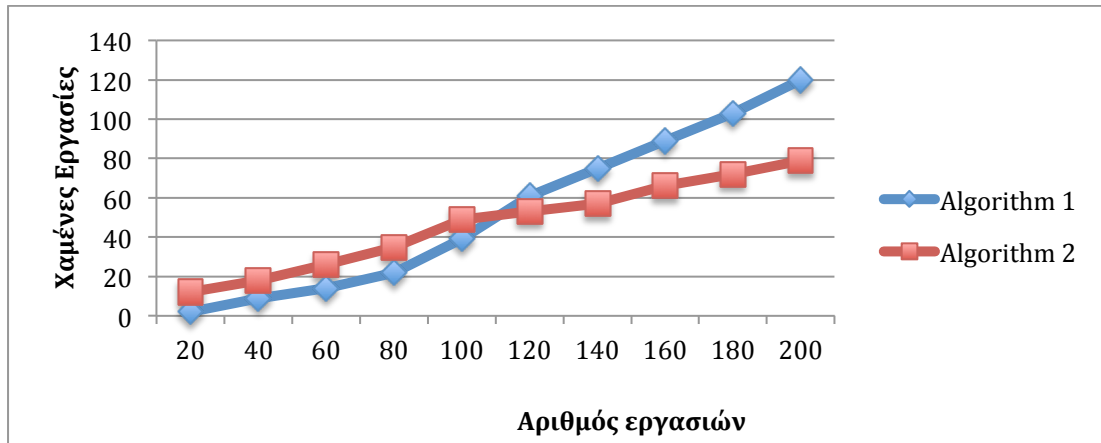
Max Arrival Time



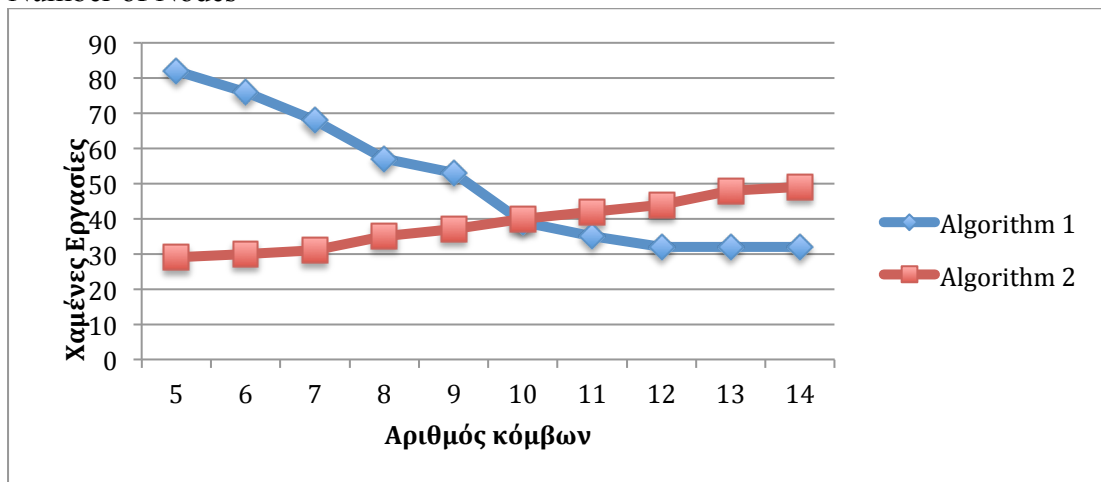
Process Speed



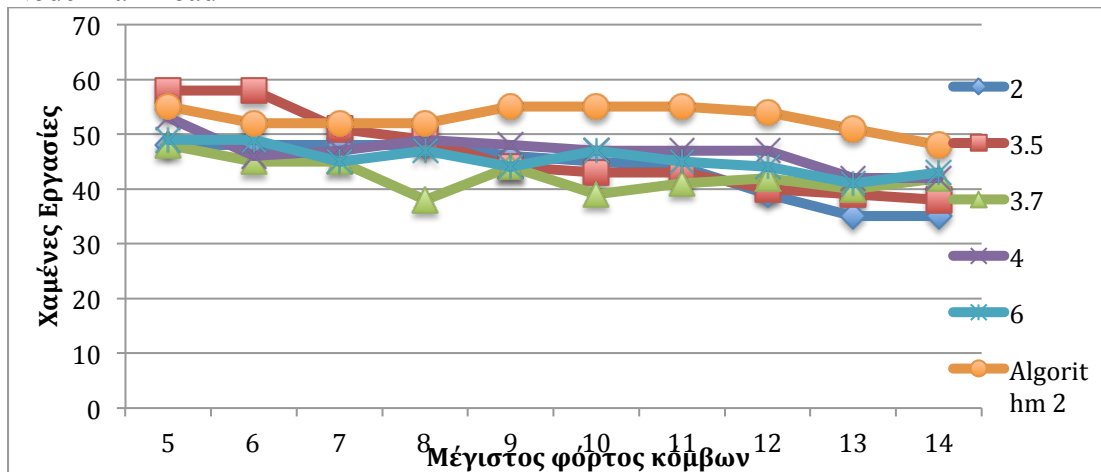
Number of Jobs



Number of Nodes



Node Max Load



Στα πρώτα τρία σενάρια όπως και στο τελευταίο βλέπουμε τον Αλγόριθμο 1 να υπερέχει έναντι του Αλγορίθμου 2. Στο τέταρτο σενάριο ο Αλγόριθμος 1 αρχίζει με πιο λίγες χαμένες εργασίες, αλλά καταλήγει με περισσότερες στις τελευταίες εκτελέσεις, σε αντίθεση με το σενάριο 5 όπου συμβαίνει το ακριβώς ανάποδο.

5.5 Σχολιασμός Αποτελεσμάτων – Συμπεράσματα

1. Εάν συγκρίνουμε τους δύο αλγορίθμους μεταξύ τους βγάζουμε το συμπέρασμα πως ο Αλγόριθμος 1 υπερέχει στα περισσότερα σενάρια. Ακόμη, στα σενάρια εκείνα όπου δεν έχει καλύτερα αποτελέσματα, υπάρχουν εκτελέσεις όπου ο Αλγόριθμος 1 υπερέχει.
2. Για να μπορέσουμε να επιλέξουμε κατάλληλες πιθανότητες για τον Αλγόριθμο 1, έπρεπε να αυξήσουμε αρκετά το φόρτο στο σύστημα, έτσι ώστε να αναγκάσουμε τις εργασίες να κάνουν περισσότερες μετακινήσεις.
3. Παρόλο που ξεχώρισαν μερικές πιθανότητες σε συγκεκριμένα σενάρια, η διαφορά στα αποτελέσματά τους δεν είναι μεγάλη, πράγμα που σημαίνει πως ακόμη και χωρίς τη χρήση των πιθανοτήτων ο Αλγόριθμος μπορεί να είχε τα ίδια καλά αποτελέσματα.
3. Δεν υπάρχει μία μοναδική πιθανότητα που φέρει τα καλύτερα αποτελέσματα σε όλα τα σενάρια, παρόλο που η πιθανότητα 6 έδωσε καλύτερα αποτελέσματα στην πλειοψηφία σεναρίων.
4. Στο δεύτερο μέρος της πειραματικής μελέτης, όπου οι εργασίες διέφεραν, παρόλο που ο φόρτος εργασίας ήταν ο ίδιος παρατηρήθηκε βελτίωση στον Αλγόριθμο 1, όπου μειώθηκαν τα ποσοστά χαμένων εργασιών. Από αυτό μπορούμε να συμπεράνουμε πως ο Αλγόριθμος 1 συμπεριφέρεται καλύτερα σε περιπτώσεις όπου οι εργασίες δεν είναι όλες όμοιες. Συγκεκριμένα οι μετακινήσεις των εργασιών στο πρώτο πειραματικό μέρος, όχι μόνο δεν βοηθούν τον αλγόριθμο, αλλά τον επιβαρύνουν. Σε κάθε μετακίνηση μία εργασία όταν θα φύγει από ένα κόμβο τότε απλά μία άλλη εργασία θα πάρει τη θέση της με ακριβώς τον ίδιο φόρτο, προκαλώντας αλυσιδωτή αντίδραση στο σύστημα προσθέτοντας καθυστέρηση.

Κεφάλαιο 6

Συμπεράσματα

6.1 Γενικά Συμπεράσματα	117
6.2 Τελική Ανασκόπηση	118
6.3 Μελλοντική Έρευνα	119

6.1 Γενικά Συμπεράσματα

Η εξισορρόπηση φόρτου εργασίας -και ειδικότερα σε υπολογιστικά νέφη- είναι ένα μεγάλο και περίπλοκο θέμα με πολλές πτυχές. Είναι όμως απαραίτητη στα περιβάλλοντα υπολογιστικών νεφών αφού είναι αναγκαία στο να επιτευχθεί η μεγιστοποίηση της ευχρηστίας των πόρων. Κάθε υπολογιστικό νέφος έχει τα δικά του χαρακτηριστικά και ιδιομορφίες τα οποία το καθιστούν διαφορετικό. Καθώς οι απαιτήσεις των χρηστών αυξάνονται, όσο αφορά την ταχύτητα και την επεξεργαστική ισχύ, η ανάγκη για επίλυση του προβλήματος της εξισορρόπησης φόρτου γίνεται ολοένα και πιο επιτακτική, αφού χωρίς τη σωστή διαχείριση οι ανάγκες των χρηστών δεν μπορούν να ικανοποιηθούν.

Οι μέχρι τώρα προτεινόμενοι αλγόριθμοι εξισορρόπησης φόρτου παρουσιάζουν προβλήματα υπερφόρτωσης και υπερχειλίσης σε μεγάλα δίκτυα, ενώ τα πλείστα έχουν μεγάλες απαιτήσεις σε πόρους. Οι αλγόριθμοι που ίσως να δίνουν καλύτερα αποτελέσματα, έχουν πολλές φορές μεγάλη καθυστέρηση (overhead) ή είναι πολύ πιο δύσκολο να εφαρμοστούν.

Ο Αλγόριθμος 1, εμπνευσμένος από τη φύση δίνει μία καλή λύση στο πρόβλημα εξισορρόπησης φόρτου στα υπολογιστικά νέφη. Δεν έχει καθυστέρηση (overhead) αφού δεν προϋποθέτει οποιονδήποτε υπολογισμό πριν την έναρξη του αλγορίθμου και εισάγει την έννοια των πιθανοτήτων με ένα τρόπο πρωτόγνωρο, ελαχιστοποιώντας και τους υπολογισμούς κατά τη διάρκεια του αλγορίθμου. Ακόμη ως αλγόριθμος δεν είναι δύσκολο να εφαρμοστεί είτε ως έχει είτε και να εισαχθεί και σε υπάρχοντα αλγόριθμο δημιουργώντας ένα νέο υβριδικό αλγόριθμο.

6.2 Τελική Ανασκόπηση

Η διπλωματική μου μελέτη ήταν ιδιαίτερα ενδιαφέρουσα. Μέσα από αυτήν μπόρεσα να εφαρμόσω σχεδόν όλα όσα διδάχτηκα στα τέσσερα χρόνια των σπουδών μου στο τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου. Ως αποτέλεσμα είχα και την επέκταση των γνώσεων μου στον τομέα της Πληροφορικής σε μέρη τα οποία δεν γνώριζαν πως υπήρχαν. Η όλη μελέτη με έκανε να αντιληφθώ πως η έρευνα στον κλάδο με τον οποίο επέλεξα να ασχοληθώ και κατ' επέκταση σε όλους τους κλάδους, δεν είναι απλή υπόθεση. Προϋποθέτει πολλές βάσεις οι οποίες κτίζονται με θεωρητική μελέτη, και πολλές πειραματικές προσπάθειες πριν καταλήξουμε σε συμπεράσματα.

Λόγω του ότι η παρούσα διπλωματική εργασία ήταν η πρώτη μεγάλη έρευνα με την οποία ασχολήθηκα, συνάντησα αρκετές δυσκολίες σε όλες τις φάσεις της εργασίας. Αρχικά έπρεπε να αφιερώσω χρόνο στην κατανόηση βασικών όρων που έπαιζαν κρίσιμο ρόλο σε αυτή τη μελέτη, αφού για εμένα η εξισορρόπηση φόρτου εργασίας ήταν άγνωστη. Δυσκολία παρουσιάστηκε και στην επιλογή των κατάλληλων άρθρων προς μελέτη αφού πολλά από τα άρθρα που συνάντησα ήταν παρεμφερή με το θέμα που ασχολούμουν αλλά όχι ιδανικά για σωστή μελέτη, ούτε και αξιόπιστα. Κάποια δυσκολία συνάντησα στη μελέτη του άρθρου που αναφερόταν στη μετανάστευση πληθυσμών, αφού έπρεπε να εξοικειωθώ με όρους της βιολογίας των πουλιών. Ιδιαίτερη δυσκολία συνάντησα κατά τη σύγκριση των αλγορίθμων, αφού προτού συγκριθούν πρέπει στη δομή τους να υπάρχει αντιστοιχία. Χωρίς την αντιστοιχία δεν θα υπήρχε μέτρο σύγκρισης των αλγορίθμων και ούτε ενδιαφέρον στη μελέτη. Τελειώνοντας τη μελέτη αισθάνομαι πιο εξοικειωμένος με τη διαδικασία μιας μελέτης, αφού αντιμετώπισα τις όλες δυσκολίες που βρήκα μπροστά μου. Έχοντας εμπλουτίσει τις γνώσεις μου γύρω από το θέμα της εξισορρόπησης φόρτου εργασίας θα προχωρήσω σε μελλοντικές έρευνες πάνω στο ίδιο θέμα πιο προετοιμασμένος.

6.3 Μελλοντική Έρευνα

Η περαιτέρω αναβάθμιση των συστημάτων απαιτεί και την ανάπτυξη περαιτέρω αλγορίθμων που να λύνουν το πρόβλημα εξισορρόπησης φόρτου εργασίας, ή έστω βελτιστοποίηση των υπαρχόντων αλγορίθμων. Ο αλγόριθμος που προτείναμε αποδείχτηκε πολύ καλός, αλλά αυτό δεν σημαίνει πως δεν μπορεί να επεκταθεί και να βελτιστοποιηθεί περισσότερο.

Μία ιδέα θα ήταν να συγκριθεί και να συνδυαστεί με τον αλγόριθμο ABC (Artificial Bee Colony) Load Balancing Technique in Cloud Computing [4], αφού και εκεί υπάρχουν τρία είδη «εργασιών» διαμορφωμένα με βάση τα τρία είδη μελισσών. Η προσαρμογή του με αυτόν τον αλγόριθμο θα έδινε στον Αλγόριθμο 1 μία μερική γνώση του περιβάλλοντος. Σε αυτή την περίπτωση θα μπορούσε να μελετηθεί το κόστος μιας τέτοιας γνώσης και κατά πόσο αυτό το κόστος εξουδετερώνεται ή όχι από την βελτίωση του χρόνου εκτέλεσης των εργασιών.

Πέραν από επέκταση ή συνδυασμό του αλγορίθμου, θα μπορούσε να μελετηθεί μετανάστευση πληθυσμού είτε άλλου είδους πουλιών, είτε άλλου είδους ζώου-ψαριού-εντόμου. Φαίνεται πως η φύση έχει αρκετά ακόμη να μας διδάξει.

Βιβλιογραφία

- [1] E.S Minor, R.I McDonald, E.A Trembl and D.L Urban, “Uncertainty in spatially explicit population models,” *Biological Conservation* 141 , 2008.
- [2] M. Katyal and A.Mishra, “A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment,” *International Journal of Distributed and Cloud Computing*, 2013.
- [3] H.Gupta and K. Sahu “Honey Bee Behavior Based Load Balancing of Tasks in Cloud Computing”, *International Journal of Science and Research (IJSR)* 2012.
- [4] N.S. Nipane and N.M. Dhande, “abc-load balancing technique-in cloud computing”, *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 2014.
- [5] M. Suresh Z.U Shafi and B.K. Santosh, “An Analysis of Load Balancing in Cloud Computing”, *International Journal of Engineering Research & Technology (IJERT)*, 2013.
- [6] R.R. Kotkondawar, P.A. Khair, M.C. Akewar and Y.N. Patil, “A Study of Effective Load Balancing Approaches in Cloud Computing”, *International Journal of Computer Applications*, 2014.