



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Ακριβείς και ευριστικοί αλγόριθμοι σχεδιασμού και
βελτιστοποίησης ροών εργασιών σε περιβάλλον
υπολογιστικού νέφους με πολλαπλές μηχανές**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΝΙΚΟΔΗΜΟΥ ΠΡΟΒΑΤΑ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ακριβείς και ευριστικοί αλγόριθμοι σχεδιασμού και βελτιστοποίησης ροών εργασιών σε περιβάλλον υπολογιστικού νέφους με πολλαπλές μηχανές

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΝΙΚΟΔΗΜΟΥ ΠΡΟΒΑΤΑ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25 Οκτωβρίου 2016.

(Υπογραφή)

.....
Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Γεώργιος Γκούμας
Λέκτορας Ε.Μ.Π.

(Υπογραφή)

.....
Δημήτριος Τσουμάκος
Επίκουρος Καθ. Ιόνιο Πανεπ.

Αθήνα, Οκτώβριος 2016

(Υπογραφή)

.....

ΝΙΚΟΔΗΜΟΣ ΠΡΟΒΑΤΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2016 – All rights reserved

Περίληψη

Οι μοντέρνες ροές εργασιών έχουν γίνει αρκετά μεγάλες και πολύπλοκες. Για κάθε εργασία είναι πιθανόν να υπάρχει η δυνατότητα εκτέλεσης σε διαφορετική μηχανή εκτέλεσης. Επομένως, μία ροή εργασιών έχει πολλά διαφορετικά πλάνα εκτέλεσης. Από όλα τα διαφορετικά πλάνα εκτέλεσης που μπορούν να προκύψουν για ροή εργασιών κρίνεται σκόπιμο να βρεθούν τα βέλτιστα ως προς κάποια κριτήρια. Στην παρούσα διπλωματική εργασία, κατασκευάζεται ένας μετα-σχεδιαστής, ο οποίος έχει ως στόχο τον αυτόματο προσδιορισμό των βέλτιστων λύσεων κατά pareto ως προς τα κριτήρια του χρόνου και του κόστους εκτέλεσης. Ο σχεδιαστής αυτός αρχικά διαβάζει την περιγραφεί μίας ροής εργασιών από ένα XML αρχείο και στη συνέχεια επιχειρεί την εύρεση των κατάλληλων πλάνων με δύο τρόπους, με τη χρήση μοντέλων που προσδιορίζουν το χρόνο και στο κόστος εκτέλεσης κάθε εργασίας σε κάθε μηχανή. Ο πρώτος χρησιμοποιεί έναν βέλτιστο αλγόριθμο που εγγυάται την εύρεση των καλύτερων λύσεων. Ο δεύτερος είναι ο προσεγγιστικός γενετικός αλγόριθμος NSGA-II. Στόχος είναι η αξιολόγηση των δύο μεθόδων, από την οποία προέκυψε ότι μέγαλη επίδραση στην επίδοση και των δύο αλγορίθμων έχει το μέγεθος της ροής εργασιών, σε αντίθεση με το πλήθος των μηχανών όπου επηρεάζει κυρίως το βέλτιστο αλγόριθμο. Επίπλέον, υπάρχουν περιπτώσεις όπου ο προσεγγιστικός αλγόριθμος είναι σημαντικά πιο γρήγορος από τον βέλτιστο (μέχρι και 152 φορές), γεγονός που υποδεικνύει τη χρήση του δεδομένης της μικρής απόστασης των λύσεων. Τέλος, διαπιστώνεται ότι χρησιμοποιώντας τον σχεδιαστή μπορεί να επιτευχθεί εντοπισμός καλύτερου πλάνου εκτέλεσης ως προς κάποιο κριτήριο σε σχέση με το να εκτελούνταν όλη τη ροή στην ίδια μηχανή εκτέλεσης.

Λέξεις Κλειδιά: <<υπολογιστικό νέφος, ροές εργασιών, πολλαπλές μηχανές, βελτιστοποίηση πολλαπλών κριτηρίων, Pareto >>

Abstract

Modern workflows have become extremely large and complex. Each task in the workflow might be able to execute in more than one execution engines. Therefore, a workflow has many different execution plans. From all different execution plans available for the workflow, it is crucial to locate the best plans according to some objectives. In this diploma thesis, a meta-planner is built , which target is to locate the best pareto solutions according to the time and cost objectives. At first, the planner reads the description of the workflow from an XML file and then attempts to find the appropriate execution plans in two ways, using models that describe the time and cost of executing each operator in any engine. The first way uses an optimal algorithm that guarantees the best solutions. The second is the approximate genetic algorithm NSGA-II. The target is the evaluation of the two methods, which showed that the size of the workflow really affects the performance of the two algorithms. Instead, the number of the available execution engines affects mostly the optimal algorithm. Moreover, there are cases that NSGA-II is much faster than the optimal (up to 152 times), which indicates its usage given the small distance of the solution sets. In the end, it is found that using the meta-planner can locate better execution plans according to an objective, instead of executing the whole workflow in the same execution engine.

Keywords: <<cloud computing, workflow, multiengine, multiobjective optimization, Pareto>>

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ.Νεκτάριο Κοζύρη, για την ευκαιρία που μου έδωσε να γνωρίσω καλύτερα τον τομέα των καταναμημένων συστημάτων και να εκπονήσω την παρούσα διπλωματική μέσα σε ένα τόσο οργανωμένο εργαστήριο. Θέλω ακόμη να ευχαριστήσω την ερευνήτρια Κατερίνα Δόκα, της οποίας η συμβολή και η βοήθεια ήταν τεράστια στην ολοκλήρωση της εργασίας. Επίσης, θα ήθελα να ευχαριστήσω και όλα τα μέλη του εργαστηρίου που ήταν διατεθειμένοι να προσφέρουν βοήθεια ή να επιλύσουν απορίες όποτε χρειάστηκε. Τέλος, θέλω να ευχαριστήσω θερμά την οικογένειά μου για τη διαρκή στήριξη της και τους φίλους μου για την κατανόηση και την εμπύχωση καθόλη τη διάρκεια αυτών των χρόνων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κατάλογος Εικόνων	1
Κατάλογος Πινάκων	4
1 Εισαγωγή.....	5
1.1. Κίνητρο της Εργασίας.....	5
1.2. Σκοπός της Εργασίας	6
1.3. Διάρθρωση της Εργασίας.....	7
2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	8
2.1. Γενικοί ορισμοί.....	8
2.2. Ροές εργασιών και πλάνα εκτέλεσης.....	8
2.2.1. Ροές Εργασιών (<i>Workflows</i>).....	8
2.2.2 Πλάνο Εκτέλεσης (<i>Execution Plan</i>).....	11
2.3. Προβλήματα βελτιστοποίησης	12
2.3.1. Βελτιστοποίηση ενός κριτηρίου (<i>Single – objective optimization</i>).....	12
2.3.2 Βελτιστοποίηση περισσότερων κριτηρίων (<i>Multi – objective optimization</i>)	12
2.4. Αλγόριθμοι εύρεσης βέλτιστων λύσεων κατα pareto	14
2.4.1. Δυναμικός Προγραμματισμός.....	14
2.4.2. Εξελικτικοί Αλγόριθμοι – Ο αλγόριθμος <i>NSGA-II</i>	15
2.5. Απόσταση Hausdorff.....	16
3 Περιγραφή Συστήματος	17
3.1. Αρχιτεκτονική Συστήματος.....	17
3.2. Χρησιμοποιούμενες Τεχνολογίες.....	18
3.2.1. <i>Java</i> και <i>Eclipse</i>	18
3.2.2. <i>MOEA framework</i>	18
3.2.3. <i>Pegasus: Γεννήτρια Ροών Εργασιών</i>	19
3.3. Περιγραφή βασικών κλάσεων	20
3.3.1. Οι κλάσεις <i>Job</i> και <i>UsedFile</i>	20
3.3.2. Η κλάση <i>GraphNode</i>	21
3.3.3. Η κλάση <i>Engine</i>	22
3.3.4. Οι κλάσεις <i>Objectives</i> και <i>TreeNode</i>	22

3.4. Ανάγνωση από xml αρχείο.....	23
3.5. Αναζήτηση βέλτιστων λύσεων κατά Pareto	24
3.5.1. Περιγραφή αλγορίθμου εύρεσης βέλτιστης λύσης κατά Pareto	26
3.5.2. Περιγραφή αλγορίθμου εύρεσης προσεγγιστικής λύσης.....	28
4 Πειραματικά Δεδομένα - Παρουσίαση και Σχολιασμός.....	29
4.1. Έλεγχος ορθότητας συστήματος σε μικρούς γράφους	29
4.1.1. Γραμμική ροή εργασιών με 7 κόμβους.....	30
4.1.2. Γραμμική ροή εργασιών με κόμβους που έχουν εξάρτηση από περισσότερους προγόνους.....	32
4.1.3. Ροή εργασιών 7 κόμβων που υποστηρίζει παράλληλη εκτέλεση δύο κόμβων....	34
4.1.4. Ροή Εργασιών 6 κόμβων με πολλές εξαρτήσεις ανάμεσα τους.....	36
4.1.5. Σχολιασμός αποτελεσμάτων.....	38
4.2. Αξιολόγηση των αλγορίθμων σε Γραμμικές Ροές Εργασιών διαφόρων μεγεθών..	38
4.2.1. Ροή με 25 εργασίες.....	39
4.2.2. Ροή με 50 εργασίες.....	41
4.2.3. Ροή με 100 εργασίες.....	44
4.2.4. Ροή με 500 κόμβους.....	47
4.2.5. Συμπεράσματα.....	50
4.3. Μελέτη σε γράφους με βαθμός παραλληλισμού μεγαλύτερο του 1.....	51
4.3.1. Μελέτη Επίδρασης αριθμού κόμβων.....	52
4.3.2. Μελέτη επίδρασης βαθμού παραλληλισμού.....	53
4.4. Μελέτη σε γράφους του Pegasus	54
4.4.1. Epigenomics.....	54
4.4.2. Montage.....	57
4.5. Εύρεση λύσης σε ροή εργασιών από πραγματικά δεδομένα	60
5 Συμπεράσματα και Επεκτάσεις.....	63
Βιβλιογραφία.....	64

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2-1: Παράδειγμα ροής εργασιών με τη μορφή γράφου.....	9
Εικόνα 2-2: Γραμμική ροή εργασιών.....	10
Εικόνα 2-3: Ροές εργασιών με δυνατότητα παράλληλης εκτέλεσης.....	10
Εικόνα 2-4: Ένα παράδειγμα ενός skyline [12].....	13
Εικόνα 2-5: Η διαδικασία που ακολουθεί ο NSGA – II.....	16
Εικόνα 3-1: Αρχιτεκτονική του συστήματος που υλοποιήθηκε.....	17
Εικόνα 3-2: Τμήμα του xml αρχείο που περιγράφει εργασίες κόμβων.....	19
Εικόνα 3-3: Τμήμα του xml αρχείου που συνδέει κάποιο κόμβο με τους πατέρες του.....	19
Εικόνα 3-4: Διάγραμμα κλάσης για τις κλάσεις Job, UsedFile.....	20
Εικόνα 3-5: Η κλάση GraphNode.....	21
Εικόνα 3-6: Η κλάση Engine.....	22
Εικόνα 3-7: Διάγραμμα κλάσεων για τις TreeNode, Objectives.....	22
Εικόνα 3-8: Διάγραμμα των κλάσεων που συμμετέχουν στην ανάγνωση της ροής εργασιών από το xml αρχείο.....	23
Εικόνα 3-9: Αλγοριθμική περιγραφή της διαδικασίας ανάγνωσης από αρχείο.....	24
Εικόνα 3-10: Διάγραμμα των κλάσεων που συμμετέχουν στη διαδικασία εύρεσης της ακριβούς και της προσεγγιστικής λύσης.....	25
Εικόνα 3-11: Αλγοριθμική περιγραφή γενικής λειτουργίας του συστήματος στην πορεία εύρεσης λύσης.....	25
Εικόνα 3-12: Ψευδοκώδικας περιγραφής αλγορίθμου εύρεσης βέλτιστων λύσεων κατά Pareto.....	27
Εικόνα 3-13: Ψευδοκώδικας της συνάρτησης υπολογισμόςΧρόνου.....	27
Εικόνα 3-14: Ψευδοκώδικας της συνάρτησης υπολογισμόςΚόστους.....	28
Εικόνα 3-15: Ψευδοκώδικας της συνάρτησης PARETO.....	28
Εικόνα 4-1: Η γραμμική ροή εργασιών που χρησιμοποιείται στη σειρά των πειράματων.....	30
Εικόνα 4-2: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-1.....	31
Εικόνα 4-3: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-1.....	31
Εικόνα 4-4: Γραμμική Ροή εργασιών με κόμβους με παραπάνω εξαρτήσεις που χρησιμοποιείται στη σειρά των πειράματων.....	32
Εικόνα 4-5: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-4.....	33
Εικόνα 4-6: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-4.....	33
Εικόνα 4-7: Γραμμική Ροή εργασιών με κόμβους με δυνατότητα ταυτόχρονης εκτέλεσης που χρησιμοποιείται στη σειρά των πειράματων.....	34
Εικόνα 4-8: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-7.....	35
Εικόνα 4-9: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-7.....	35
Εικόνα 4-10: Πολύπλοκη ροή εργασιών από θέμα ροής δεδομένων που χρησιμοποιήθηκε στα πειράματα.....	36
Εικόνα 4-11: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-10.....	37
Εικόνα 4-12: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-10.....	37

Εικόνα 4-13: Αποτελέσματα για γράφο 25 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	39
Εικόνα 4-14: Αποτελέσματα για γράφο 25 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	39
Εικόνα 4-15: Αποτελέσματα για γράφο 25 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	40
Εικόνα 4-16: Αποτελέσματα για γράφο 50 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	41
Εικόνα 4-17: Αποτελέσματα για γράφο 50 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	42
Εικόνα 4-18: Αποτελέσματα για γράφο 50 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	43
Εικόνα 4-19: Αποτελέσματα για γράφο 100 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	44
Εικόνα 4-20: Αποτελέσματα για γράφο 100 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	45
Εικόνα 4-21: Αποτελέσματα για γράφο 100 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	46
Εικόνα 4-22: Αποτελέσματα για γράφο 500 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	47
Εικόνα 4-23: Αποτελέσματα για γράφο 500 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	48
Εικόνα 4-24: Αποτελέσματα για γράφο 500 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων.....	49
Εικόνα 4-25: Διάγραμμα Επιτάχυνσης του NSGA-II ως προς τον βέλτιστο και η απόσταση της λύσης του από τη βέλτιστη ως συνάρτηση του πλήθους των κόμβων του γραμμικού γράφου	50
Εικόνα 4-26: Γράφος με βαθμό παραλληλίας (α) 3 και (β) 5.....	51
Εικόνα 4-27: Διάγραμμα χρόνου εκτέλεσης των αλγορίθμων ως συνάρτηση του βαθμού παραλληλισμού του γράφου	53
Εικόνα 4-28: Γράφος της οικογένειας <i>Eriogenomics</i> 24 κόμβων	54
Εικόνα 4-29: Σύνολα λύσεων για ροή εργασιών τύπου <i>Eriogenomics</i> 24 κόμβων με δύο διαθέσιμες μηχανές από τον βέλτιστο και τον προσεγγιστικό αλγόριθμο.	55
Εικόνα 4-30: Χρόνος εκτέλεσης του NSGA-II ανάλογα με το πλήθος των κόμβων του γράφου <i>Eriogenomics</i>	56
Εικόνα 4-31: Γράφος της οικογένειας <i>Montage</i> 25 κόμβων.....	57
Εικόνα 4-32: Σύνολα λύσεων για ροή εργασιών τύπου <i>Montage</i> 25 κόμβων με δύο διαθέσιμες μηχανές από τον βέλτιστο και τον προσεγγιστικό αλγόριθμο.	58

Εικόνα 4-33: Χρόνος εκτέλεσης του NSGA-II ανάλογα με το πλήθος των κόμβων του γράφου <i>Montage</i>	59
Εικόνα 4-34: Πραγματική ροή εργασιών η οποία θα βελτιστοποιηθεί.....	60
Εικόνα 4-35: Ο χρόνος εκτέλεσης του πλάνου εκτέλεσης αν εκτελεστεί όλος ο γράφος σε <i>Spark, Scikit</i> ή σύμφωνα με το πλάνο που πρότεινε ο σχεδιαστής.	62

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 4-1: Αριθμός επαναλήψεων NSGA-II για διάφορα πλήθη κόμβων ροής εργασιών ώστε να ληφθεί ένα ικανοποιητικό αποτέλεσμα.....	50
Πίνακας 4-2: Στοιχεία από την εκτέλεση των αλγορίθμων σε γράφους με βαθμό παραλληλισμού 3.....	52
Πίνακας 4-3: Στοιχεία από την εκτέλεση των αλγορίθμων σε γράφους με βαθμό παραλληλισμού 5.....	52
Πίνακας 4-4: Χρόνοι εκτέλεσης των δύο αλγορίθμων και η απόσταση των λύσεων τους για τη ροή εργασιών <i>Eriogenomixs</i> 24 κόμβων με δύο διαθέσιμες μηχανές.....	55
Πίνακας 4-5: Χρόνοι εκτέλεσης των δύο αλγορίθμων και η απόσταση των λύσεων τους για τη ροή εργασιών <i>Eriogenomixs</i> 24 κόμβων με δύο διαθέσιμες μηχανές.....	58
Πίνακας 4-6: Πειραματικά δεδομένα από τη ροή εργασιών της <i>Εικόνας 4-33</i>	60

1

ΕΙΣΑΓΩΓΗ

1.1. ΚΙΝΗΤΡΟ ΤΗΣ ΕΡΓΑΣΙΑΣ

Η ανάλυση μεγάλου όγκου δεδομένων έχει γίνει αναγκαιότητα για τη πλεοψήφια των βιομηχανιών, λαμβάνοντας την πρώτη θέση στην αξιολόγηση του ρίσκου, στην αποδοτικότητα επιχειρηματικών διαδικασιών κτλ. Για αυτό έχει αυξηθεί σημαντικά η αξιοποίηση μεγάλης ποσότητα κρίσιμων επιχειρηματικών δεδομένων από μηχανικούς, αναλυτές και επιστήμονες. Τέτοιου τύπου ανάλυση δεδομένων απαιτεί υψηλό βαθμό παραλληλισμού τόσο στην αποθήκευση όσο και στον υπολογισμό. Τα μοντέρνα κέντρα δεδομένων φιλοξενούν τεράστιους όγκους δεδομένων, που αποθηκεύονται σε ένα μεγάλο αριθμό κόμβων με πολλαπλές συσκευές αποθήκευσης και τα επεξεργάζονται χρησιμοποιώντας εκατοντάδες ή εκατομμύρια πυρήνες. Η ζήτηση σχεδόν σε πραγματικό χρόνο δεδομενοκεντρικών αναλύσεων έχει προκαλέσει την ανάπτυξη ποικίλων μηχανών εκτέλεσης και καταστημάτων δεδομένων που στοχεύουν σε συγκεκριμένα δεδομένα και τύπους υπολογισμών. Ωστόσο, αυτά υποθέτουν αυστήρα περιβάλλοντα με μία μηχανή εκτέλεσης.

Παρόλα αυτά, οι μοντέρνες ροές εργασιών έχουν γίνει αρκετά μεγάλες και σύνθετες. Συγκεκριμένα, οι ροές εργασιών μπορεί να περιλαμβάνουν ποικίλους τύπους δεδομένων που έχουν παραχθεί από διαφορετικές πηγές. Επιπλέον, εκτελούνται υπό διάφορους περιορισμούς ή πολιτικές (για παράδειγμα, τη βελτιστοποίηση της απόδοσης ή του κόστους, την απαίτηση διαφορετικών βαθμών ανοχής σε σφάλματα, κλπ.). Τέλος, οι τελεστές της ροής εργασιών μπορεί να είναι πολύ διαφορετικοί, από αρκετά απλοί μέχρι τελεστές που σχετίζονται με λειτουργίες σε γράφους.

Οι αναλύσεις σε πολλαπλές μηχανές προτείνονται τελευταία ως μία πολλά υποσχόμενη λύση που μπορεί να βελτιστοποιήσει για αυτή την πολυπλοκότητα [1] και κερδίζουν έδαφος συνεχώς. Πωλητές νέφους προσφέρουν λύσεις λογισμικού που συνεργάζονται με διάφορα framework, αποθήκες δεδομένων και βιβλιοθήκες για να διευκολύνουν την διαχείριση πολλαπλών εγκατάσεων και διαμορφώσεων [2], [3], [4], [5]. Μία από τις πιο επιτακτικές προκλήσεις σε ένα περιβάλλον πολλαπλών μηχανών είναι ο σχεδιασμός και η δημιουργία ενός μετα-σχεδιαστή που αναθέτει αυτόματα εργασίες να εκτελεστούν στη

σωστή μηχανή εκτέλεσης σύμφωνα με πολλαπλά κριτήρια, τις δημιουργεί και τις τρέχει χωρίς χειροκίνητη παρέμβαση.

1.2. ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Υποθέτοντας την ύπαρξη ενός μοντέλου κόστους για κάθε μηχανή, είναι επιθυμητή η δημιουργία ενός μετα-σχεδιαστή, ο οποίος είναι ικανός να αναθέσει ξεχωριστά κομμάτια της ροής εργασιών στο πιο συμφέρον πλάνο εκτέλεσης ώστε να βελτιστοποιηθεί οποιαδήποτε πολιτική που έχει οριστεί από το χρήστη. Η προκύπτουσα βελτιστοποίηση βελτιώνεται από οποιαδήποτε προσπάθεια βελτιστοποίησης εντός μίας μηχανής.

Ουσιαστικά, στην συγκεκριμένη εργασία κατασκευάζεται ένας σχεδιαστής σε περιβάλλον πολλαπλών μηχανών που επιλέγει τα πιο υποσχόμενα πλάνα εκτέλεσης μίας ροής εργασιών ανάμεσα στις διαθέσιμες μηχανές, αφού έχει συμβουλευτεί μοντέλα επίδοσης και κόστους από διάφορους τελεστές.

Το σύστημα είναι ικανό να επιλέξει ανάλογα να εκτελέσει οποιοδήποτε τμήμα της ροής εργασιών σε μία (πιθανόν διαφορετική) μηχανή εκτέλεσης. Ο στόχος είναι να ανατεθούν οι υπο-εργασίες στην πιο υποσχόμενη διαθέσιμη τεχνολογία και να επιβεβαιωθεί ο κατάλληλος προγραμματισμός των πόρων και των ροών δεδομένων ώστε να βελτιστοποιηθεί η επίδοση και το κόστος εκτέλεσης. Έτσι, αποφασίζονται τα καλύτερα πλάνα εκτέλεσης και η καλύτερη τοποθέτηση δεδομένων (για παράδειγμα η υποεργασία A να εκτελεστεί σε Spark, να αποθηκεύσει τα άμεσα δεδομένα σε μία NoSQL μηχανή και μετά να τρέξουν παράλληλα οι υποεργασίες B και C, γράφοντας τα τελικά αποτελέσματα στο HDFS).

Στο χρήστη παρέχεται η δυνατότητα εύρεσης πλάνων εκτέλεσης με δύο τρόπους. Ο πρώτος είναι ένας βέλτιστος αλγόριθμος, ο οποίος εγγυάται την εύρεση των βέλτιστων κατά Pareto πλάνων εκτέλεσης για τη ροή εργασιών. Ο δεύτερος είναι ένας ευριστικός αλγόριθμος, ο NSGA-II, ο οποίος δίνει ένα σύνολο πλάνων εκτέλεσης ως λύσεις, που προσεγγίζει ή και πολλές φορές ταυτίζεται με το σύνολο των βέλτιστων λύσεων.

Στόχος της διπλωματικής είναι η σύγκριση και η αξιολόγηση των παραπάνω δύο αλγορίθμων. Θα διαπιστώθει σε ποιες περιπτώσεις αξίζει η προσπάθεια αναζήτησης της βέλτιστης λύσης (π.χ. σε απλούς γράφους), ενώ θα προκύψουν περιπτώσεις όπου ο προσεγγιστικός αλγόριθμος θα είναι σημαντικά γρηγορότερος από το βέλτιστο, όπως στην περίπτωση του γράφου Eripenomics που ο προσεγγιστικός τρέχει 152 φορές πιο γρήγορα, και εκεί θα είναι προτιμότερος. Τέλος, θα ελεγχθεί με χρήση πραγματικών δεδομένων, κατά πόσο το σύστημα είναι ικανό να βρει λύσεις καλύτερες από την απευθείας εκτέλεση των εργασιών σε κάποια μηχανή εκτέλεσης.

1.3. ΔΙΑΡΘΡΩΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Η εργασία αποτελείται συνολικά από πέντε κεφάλαια. Η διάρθρωση της είναι η ακόλουθη:

- Στο **Κεφάλαιο 2**, καλύπτεται όλο το απαραίτητο θεωρητικό υπόβαθρο που χρειάζεται για την κατανόηση των θεμάτων που καλύπτει η εργασία.
- Στο **Κεφάλαιο 3**, γίνεται αναλυτική περιγραφή του συστήματος που έχει υλοποιηθεί. Αρχικά, αναφέρονται όσα εργαλεία χρησιμοποιήθηκαν. Στη συνέχεια, εξηγείται ο τρόπος λειτουργίας του βέλτιστου αλγορίθμου που υλοποιήθηκε. Στο τέλος του κεφαλαίου παρουσιάζεται ο τρόπος λειτουργίας του προσεγγιστικού αλγορίθμου.
- Στο **Κεφάλαιο 4**, γίνεται παρουσίαση των πειράματων που εκτελέστηκαν με χρήση του συστήματος που περιγράφεται στο Κεφάλαιο 3. Σχολιάζεται τόσο η επίδοση των αλγορίθμων, όσο και η απόδοση του προσεγγιστικού αλγορίθμου σε σχέση με τον βέλτιστο.
- Στο **Κεφάλαιο 5**, γίνεται μια σύνοψη των συμπερασμάτων που προέκυψαν από την ανάλυση στο Κεφάλαιο 4 και αναφέρονται πιθανές μελλοντικές επεκτάσεις της εργασίας.

2

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Στο κεφάλαιο αυτό θα καλύφθει το απαραίτητο θεωρητικό υπόβαθρο για την κατανόηση του κειμένου. Θα αναλύθουν έννοιες, όπως οι ροές εργασιών (workflows), οι μηχανές (engines) και άλλες σχετικά με το πλάνο εκτέλεσης και τη βελτιστότητα του.

2.1. ΓΕΝΙΚΟΙ ΟΡΙΣΜΟΙ

Υπολογιστικό Νέφος (*cloud computing*) [6] ονομάζεται το μοντέλο που καθιστά δυνατή την κατ' απαίτηση διαδικτυακή διάθεση υπολογιστικών πόρων (δικτύων, εξυπηρετητών, χώρου αποθήκευσης, εφαρμογών και υπηρεσιών) και δεδομένων σε υπολογιστές και άλλες συσκευές. Παρέχει στους χρήστες του τη δυνατότητα να αποθηκεύουν και να επεξεργάζονται τα δεδομένα τους σε άλλα κέντρα δεδομένων, τα οποία είναι πιθανό να βρίσκονται μακριά από το χρήστη.

Ροή Εργασιών (*workflow*) [7] είναι μια σειρά από ενέργειες, οι οποίες πρέπει να εκτελεστούν για να ολοκληρωθεί μια εργασία σε ορισμένα δεδομένα επιτυχώς. Κάθε μία από αυτές τις ενέργειες αποτελεί μία στοιχειώδη εργασία, που πρέπει να πραγματοποιηθεί πάνω στα διαθέσιμα δεδομένα.

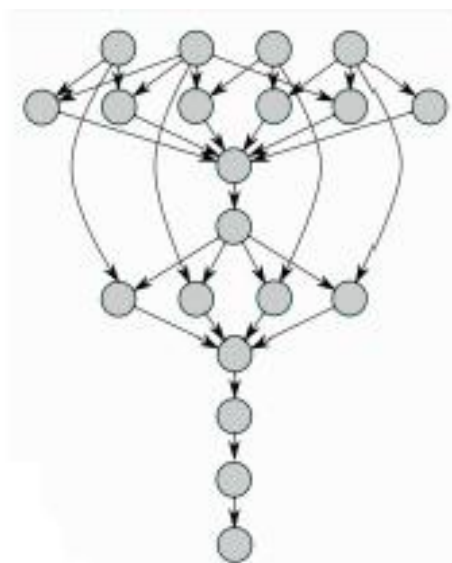
Μηχανή (*engine*) είναι το περιβάλλον στο οποίο μπορεί να εκτελεστεί μια στοιχειώδης εργασία. Κάθε μηχανή μπορεί να περιλαμβάνει διαφορετικές υλοποιήσεις για μια εργασία. Κάθε υλοποίηση καταλήγει στο ίδιο αποτέλεσμα μετά την επεξεργασία, με διαφορετικό όμως χρόνο εκτέλεσης στην κάθε περίπτωση.

2.2. ΡΟΕΣ ΕΡΓΑΣΙΩΝ ΚΑΙ ΠΛΑΝΑ ΕΚΤΕΛΕΣΗΣ

2.2.1. ΡΟΕΣ ΕΡΓΑΣΙΩΝ (WORKFLOWS)

Μια ροή εργασιών [8] μπορεί να παρασταθεί με τη βοήθεια ενός κατευθυνόμενου ακυκλικού γραφήματος (*directed acyclic graph - DAG*). Στο γράφο αυτό, κάθε κόμβος αντιστοιχεί σε μία από τις στοιχειώδεις εργασίες που πρόκειται να εκτελεστούν πάνω στα δεδομένα προς

επεξεργασία. Οι ακμές αυτού δηλώνουν μεταφορά δεδομένων από την μία στοιχειώδη εργασία στην επομένη. Ένα παράδειγμα μιας ροής εργασιών φαίνεται στην Εικόνα 2-1.



Εικόνα 2-1: Παράδειγμα ροής εργασιών με τη μορφή γράφου

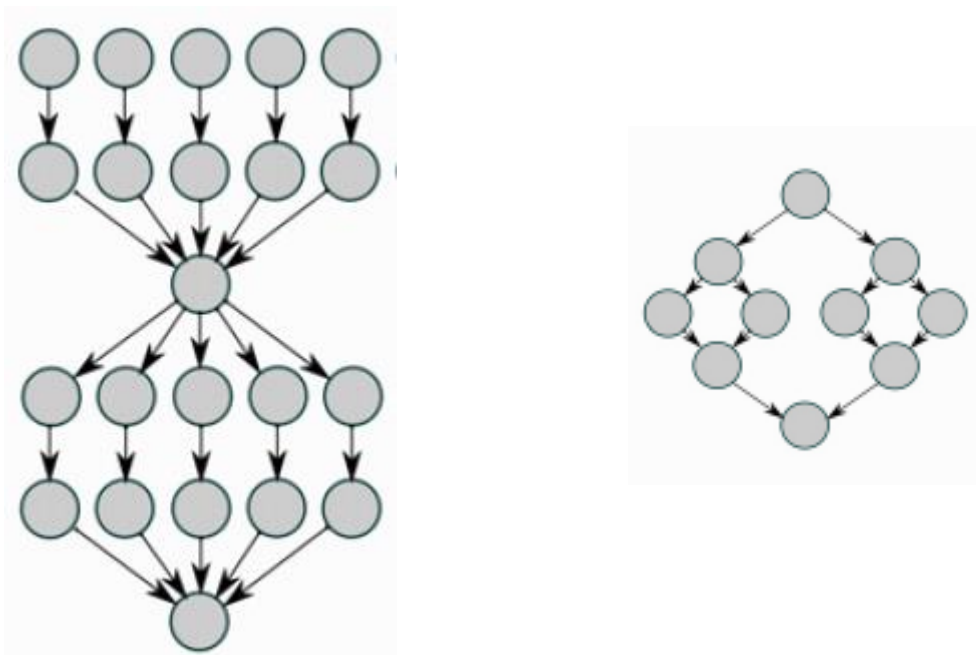
Όπως φαίνεται και από την Εικόνα 2-1, κάθε στοιχειώδης εργασία σε μία ροή εργασιών έχει κάποιες συγκεκριμένες εργασίες που πρέπει να εκτελεστούν υποχρεωτικά πριν από αυτήν. Συνεπώς, στον γράφο η ύπαρξη κάποιας ακμής δεν δηλώνει μόνο την μεταφορά δεδομένων, όπως έχει αναφερθεί παραπάνω, αλλά και την απαίτηση εκτέλεσης ενός βήματος πριν από κάποιο άλλο.

Ανάλογα με την ύπαρξη των ακμών, υπάρχουν διάφορα είδη ροών εργασιών. Η πιο απλή είναι η γραμμική ροή εργασιών, η οποία παρουσιάζεται στην Εικόνα 2-2. Όπως υποδηλώνει το όνομα της, τα επιμέρους βήματα πρέπει να εκτελεστούν σειριακά, δηλαδή απαιτείται να ολοκληρωθεί το προηγούμενο για να ξεκινήσει η εκτέλεση του επόμενου.



Εικόνα 2-2: Γραμμική ροή εργασιών

Ωστόσο, υπάρχουν και ροές εργασιών οι οποίες επιτρέπουν την παράλληλη εκτέλεση κάποιων επιμέρους βημάτων, όπως αυτές που παρουσιάζονται στην Εικόνα 2-3. Ένα σημαντικό μέγεθος που ορίζεται σε τέτοιους τύπους ροών εργασιών είναι ο *βαθμός παραλληλισμού*, δηλαδή το μέγιστο πλήθος των στοιχειώδων βημάτων τα οποία μπορούν να εκτελεστούν ταυτόχρονα.



Εικόνα 2-3: Ροές εργασιών με δυνατότητα παράλληλης εκτέλεσης

Φυσικά, υπάρχουν και πιο πολύπλοκες ροές εργασιών, όπως αυτή της Εικόνας 2-1. Σε αυτές, είναι αναγκαία η μεταφορά δεδομένων από πολλούς κόμβους και ενδεχομένως και από κόμβους που έχουν εκτελεστεί αρκετά βήματα πριν.

2.2.2 ΠΛΑΝΟ ΕΚΤΕΛΕΣΗΣ (EXECUTION PLAN)

Όταν μια ροή εργασιών πρόκειται να εκτελεστεί σε περιβάλλον υπολογιστικού νέφους με πολλαπλές μηχανές, είναι απαραίτητος ο σχεδιασμός ενός πλάνου εκτέλεσης για αυτήν. Με τον όρο *πλάνο εκτέλεσης (execution plan)* εννοούμε την αντιστοίχιση των επιμέρους κόμβων – εργασιών σε μηχανές, έχοντας ορίσει τους διαθέσιμους πόρους για την καθεμία.

Για κάθε μία ροή εργασιών, ανάλογα με τις διαθέσιμες μηχανές, μπορούν να κατασκευαστούν πολλά διαφορετικά πλάνα εκτέλεσης. Το κάθε πλάνο μπορεί να αξιολογηθεί σύμφωνα με διάφορες μετρικές, όπως είναι ο χρόνος εκτέλεσης ή το κόστος να υλοποιηθεί το δεδομένο πλάνο εκτέλεσης.

Ο χρόνος εκτέλεσης (t) του πλάνου μπορεί να υπολογιστεί με χρήση της αναδρομικής σχέσης 2.1.

$$t_p^i = \max_{q:\text{πατέρας } p} \{t_q^k + dtt_i^k\} \quad (2.1)$$

Στην σχέση 2.1. τα σύμβολα ερμηνεύονται ως εξής:

- t_p^i είναι ο χρόνος εκτέλεσης της εργασίας – κόμβου στη μηχανή i με τους επιλεγμένους πόρους.
- t_q^k είναι, αντίστοιχα, ο χρόνος εκτέλεσης του πατέρα q στην μηχανή k .
- dtt_i^k είναι ο χρόνος που απαιτείται για την μεταφορά των δεδομένων που χρειάζεται η εργασία p για να εκτελεστεί από τον πατέρα της q . Σε περίπτωση που οι δύο εργασίες εκτελούνται στις ίδιες μηχανές, ο χρόνος αυτός είναι μηδέν, ενώ σε αντίθετη περίπτωση ορίζεται ως το πηλίκο των bytes που πρόκειται να μεταφερθούν προς την μικρότερη διαθέσιμη ταχύτητα δικτύου (σε Bps) όπως φαίνεται μέσω της σχέσης 2.2.

$$dtt_i^k = \frac{\text{bytesToTransfer}}{\min_{j \in \{i,k\}} \{\text{netSpeed}_j\}} \quad (2.2)$$

Το κόστος του πλάνου εκτέλεσης μπορεί να είναι είτε χρηματικό κόστος είτε κόστος που εκφράζεται συναρτήσει των πόρων που έχουν χρησιμοποιηθεί. Σε κάθε περίπτωση όμως το κόστος μπορεί να υπολογιστεί ως το άθροισμα του κόστους κάθε επιμέρους εργασίας στη μηχανή στην οποία εκτελείται, όπως φαίνεται στη σχέση (2.3).

$$\text{total Cost} = \sum_p c_p^i \quad (2.3)$$

Στη σχέση 2.3 με c_p^i συμβολίζεται το κόστος για την εκτέλεση της εργασίας p στη μηχανή i .

2.3. ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Πρόβλημα βελτιστοποίησης (*optimization problem*) [9] είναι το πρόβλημα εύρεσης της βέλτιστης λύσης ανάμεσα σε όλες τις διαθέσιμες λύσεις. Τα προβλήματα βελτιστοποίησης διακρίνονται σε δύο κατηγορίες αναλόγα με το αν οι μεταβλητές τους είναι διακριτές ή συνεχείς. Ένα πρόβλημα βελτιστοποίησης με διακριτές μεταβλητές ονομάζεται συνδυαστικό πρόβλημα βελτιστοποίησης. Σε τέτοιου τύπου προβλήματα, γίνεται αναζήτηση ενός ή περισσότερων αντικειμένων (ανάλογα με το πλήθος των κριτηρίων προς βελτιστοποίηση) από ένα μετρήσιμο σύνολο πιθανών λύσεων. Προβλήματα με συνεχείς μεταβλητές περιλαμβάνουν προβλήματα με περιορισμούς και πολυτροπικά προβλήματα. Το πρόβλημα βελτιστοποίησης που αντιμετωπίζεται στην παρούσα εργασία είναι η εύρεση των πλάνων εκτέλεσης μιας ροής εργασιών που βελτιστοποιούν ταυτόχρονα τον χρόνο και το κόστος ως κριτήρια.

2.3.1. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΝΟΣ ΚΡΙΤΗΡΙΟΥ (SINGLE – OBJECTIVE OPTIMIZATION)

Στόχος των προβλημάτων αυτών είναι να βρεθεί η βέλτιστη λύση, η οποία αντιστοιχεί στην ελάχιστη ή τη μέγιστη τιμή μιας συνάρτησης – κριτηρίου (*single – objective function*). Για παράδειγμα, στην περίπτωση των ροών εργασιών, πρόβλημα βελτιστοποίησης του κριτηρίου του χρόνου εκτέλεσης είναι η ανάζητηση εκείνου του πλάνου εκτέλεσης, το οποίο εκτελείται στον ελάχιστο χρόνο. Το πρόβλημα αυτού του είδους προβλημάτων βελτιστοποίησης είναι ότι δεν είναι δυνατόν να συνυπολογιστούν περισσότερα από ένα κριτήρια, όταν υπάρχουν και για αυτό υπάρχουν τα προβλήματα βελτιστοποίησης περισσότερων κριτηρίων. [10]

2.3.2 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΠΕΡΙΣΣΟΤΕΡΩΝ ΚΡΙΤΗΡΙΩΝ (MULTI – OBJECTIVE OPTIMIZATION)

Στα προβλήματα βελτιστοποίησης περισσότερων κριτηρίων, [11] στόχος είναι η ταυτόχρονη βελτιστοποίηση περισσότερων του ενός συναρτήσεων – κριτηρίων. Ουσιαστικά, η βέλτιστη απόφαση περιλαμβάνει τη μερική βελτιστοποίηση κάθε μίας εκ των συναρτήσεων, καθώς αυτές είναι μεταξύ τους αντικρουόμενες. Για παράδειγμα, στο πρόβλημα της βελτιστοποίησης των ροών εργασιών, ο χρόνος εκτέλεσης ενός πλάνου είναι αντικρουόμενο κριτήριο σε σχέση με το κόστος αυτού.

Σε ένα τετριμμένο πρόβλημα βελτιστοποίησης πολλών κριτηρίων, δεν υπάρχει μία μεμονωμένη λύση που να βελτιστοποιεί ταυτόχρονα όλα τα κριτήρια. Σε αυτήν την περίπτωση υπάρχει ένας αριθμός (πιθανόν

άπειρων) λύσεων βέλτιστων κατά Pareto. Μία λύση ονομάζεται **βέλτιστη κατά Pareto**, εάν καμία από τις συναρτήσεις – κριτήρια δεν μπορεί να βελτιωθεί χωρίς να επηρεάσει αρνητικά τις τιμές των υπόλοιπων κριτηρίων. Χωρίς επιπλέον συνθήκες για την επιλογή μιας εκ των λύσεων, όλες οι βέλτιστες λύσεις κατά Pareto θεωρούνται εξίσου καλές.

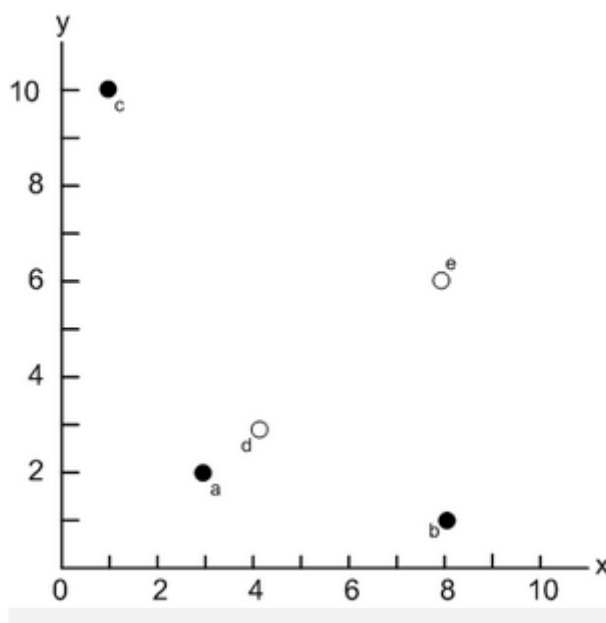
Με μαθηματικούς όρους, ένα πρόβλημα βελτιστοποίησης πολλών κριτηρίων μπορεί να οριστεί ως

$$\min_{x \in X} (f_1(x), f_2(x), \dots, f_k(x))$$

όπου ο ακέραιος $k \geq 2$ είναι το πλήθος των κριτηρίων και το σύνολο X είναι το σύνολο των διανυσμάτων από τις μεταβλητές του προβλήματος. Στην ίδια μοντελοποίηση, μία λύση $x^1 \in X$ **επικρατεί κατά Pareto** (*pareto dominant*) σε μία λύση $x^2 \in X$, εάν

- $f_i(x^1) \leq f_i(x^2)$ για κάθε $i \in \{1, 2, \dots, k\}$ και
- $f_j(x^1) < f_j(x^2)$ για τουλάχιστον ένα $j \in \{1, 2, \dots, k\}$.

Μία λύση ονομάζεται **βέλτιστη κατά Pareto** (*pareto optimal*) αν και μόνο αν δεν υπάρχει κάποια άλλη λύση, η οποία να επικρατεί σε αυτήν κατά Pareto. Στην περίπτωση που το σύνολο X των διανυσμάτων είναι διακριτό, τότε το σύνολο των βέλτιστων λύσεων κατά Pareto αποτελεί τον ορίζοντα (*skyline*) του προβλήματος. Ενδεικτικά, στην Εικόνα 2.4, παρατίθεται το skyline από κάποια σημεία στον 2Δ χώρο.



Εικόνα 2-4: Ένα παράδειγμα ενός skyline [12]

Το πρόβλημα βελτιστοποίησης που επιλύεται στη συγκεκριμένη εργασία είναι δύο κριτηρίων. Συγκεκριμένα, ανάμεσα στο σύνολο όλων

των πλάνων εκτέλεσης, αναζητείται το skyline του προβλήματος, όπου οι συναρτήσεις - κριτήρια είναι ο χρόνος εκτέλεσης και το κόστος του πλάνου.

2.4. ΑΛΓΟΡΙΘΜΟΙ ΕΥΡΕΣΗΣ ΒΕΛΤΙΣΤΩΝ ΛΥΣΕΩΝ ΚΑΤΑ PARETO

Η επίλυση ενός προβλήματος βελτιστοποίησης πολλών κριτηρίων δεν είναι μία απλή διαδικασία, όπως στην περίπτωση του ενός κριτηρίου, καθώς υπάρχει περισσότερες από μία βέλτιστες λύσεις κατά Pareto. Το γεγονός αυτό κάνει αρκετά δύσκολη τη διαδικασία εύρεσης του ακριβούς skyline του προβλήματος. Ως αποτέλεσμα, πολλές φορές είναι αρκετό να χρησιμοποιούμε ένα άλλο σύνολο σημείων, τα οποία να αποτελούν μία καλή προσέγγιση για το skyline του προβλήματος.

Για την ακριβή εύρεση του skyline χρησιμοποιούνται οι αρχές του δυναμικού προγραμματισμού. Για την εύρεση ενός προσεγγιστικού συνόλου υπάρχουν διάφορες μέθοδοι. Ωστόσο, στην παρούσα εργασία θα γίνει αναφορά στους εξελικτικούς αλγόριθμους και συγκεκριμένα στον αλγόριθμο NSGA-II.

2.4.1. ΔΥΝΑΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ο δυναμικός προγραμματισμός [13] είναι μία μέθοδος επίλυσης προβλημάτων μέσω του συνδυασμού των λύσεων κάποιων υποπροβλημάτων. Είναι εφαρμόσιμος όταν τα διάφορα υποπροβλήματα δεν είναι ανεξάρτητα μεταξύ τους, δηλαδή όταν έχουν κάποια κοινά υπό-υποπροβλήματα. Το κάθε υπο-υποπρόβλημα επιλύεται μόνο μία φορά και αποθηκεύεται, αποφεύγοντας έτσι τον εκ νέου υπολογισμό κάθε φορά που συναντάται.

Ο δυναμικός προγραμματισμός εφαρμόζεται κατά κανόνα σε προβλήματα βελτιστοποίησης. Το ζητούμενο είναι η εύρεση των βέλτιστων λύσεων. Η ανάπτυξη ενός αλγορίθμου δυναμικού προγραμματισμού μπορεί να αναλυθεί στα παρακάτω βήματα.

- Χαρακτηρισμός της δομής της βέλτιστης λύσης.
- Αναδρομικός ορισμός της τιμής μιας βέλτιστης λύσης.
- Υπολογισμός της τιμής από «κάτω προς τα πάνω» (bottom - up τεχνική).
- Κατασκευή βέλτιστων λύσεων από τα δεδομένα που έχουν συλλεχθεί.

Σαν τεχνική, ο δυναμικός προγραμματισμός εγγυάται πάντα την εύρεση της βέλτιστης λύσης και είναι πολύ εύκολα εφαρμόσιμος. Ωστόσο, ανάλογα με την πολυπλοκότητα του προβλήματος, είναι πιθανό να χρειάζονται απαγορευτικοί χρόνοι για την εκτέλεση του, καθιστώντας μη κατάλληλη επιλογή σε κάποιες περιπτώσεις.

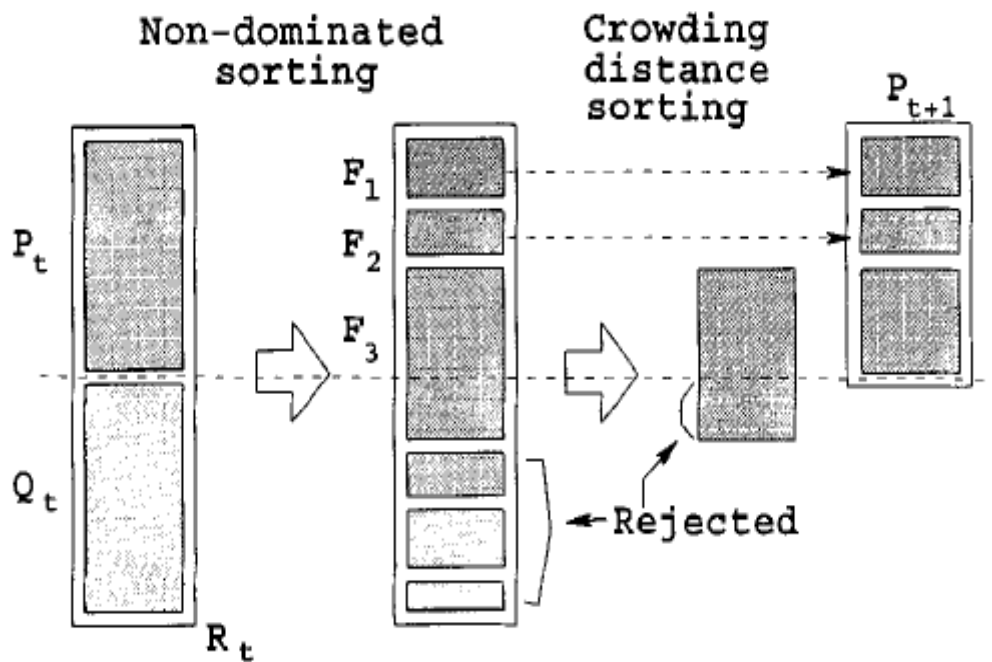
Εξελικτικός αλγόριθμος (*evolutionary algorithm*) [14] είναι ένας γενικός μετα-ευριστικός αλγόριθμος βελτιστοποίησης με βάση τον πληθυσμό. Χρησιμοποιεί μηχανισμούς που έχουν εμπνευστεί από την βιολογική εξέλιξη, όπως η αναπαραγωγή και η επιλογή. Υποψηφίες λύσεις του προβλήματος βελτιστοποίησης παίζουν το ρόλο των ατόμων του πληθυσμού και η συνάρτηση καταλληλότητας (*fitness function*) αποφασίζει την ποιότητα των λύσεων. Η εξέλιξη του πληθυσμού λαμβάνει χώρα μετά από επαναλαμβανόμενες εφαρμογές των παραπάνω τελεστών.

Οι εξελικτικοί αλγόριθμοι συνήθως αποδίδουν καλά στην προσέγγιση λύσεων σε όλους τους τύπους των προβλημάτων επειδή δεν κάνουν κάποια υπόθεση για την καταλληλότητα του υποβάθρου. Στις περισσότερες εφαρμογές, όμως, η υπολογιστική πολυπλοκότητα είναι ένας απαγορευτικός παράγοντας, λόγω του υπολογισμού της συνάρτησης καταλληλότητας. Η εκτίμηση της καταλληλότητας ξεπερνά αυτή τη δυσκολία, επιλύοντας συχνά πολύπλοκα προβλήματα. Επόμενως, δεν υπάρχει άμεση σχέση ανάμεσα στην πολυπλοκότητα του αλγορίθμου και το πρόβλημα.

Ο πιο συνηθισμένος τύπος από τους εξελικτικούς αλγόριθμους είναι οι **γενετικοί αλγόριθμοι** (*genetic algorithms*). Πραγματοποιείται αναζήτηση της λύσης με τη μορφή συμβολοσειρών αριθμών εφαρμόζοντας τελεστές, όπως ανασυνδυασούς ή μεταθέσεις (κάποιες φορές έναν εκ των δύο, κάποιες φορές και τους δύο). Ο τύπος αυτός των εξελικτικών αλγορίθμων χρησιμοποιείται κυρίως σε προβλήματα βελτιστοποίησης.

Ο γενετικός αλγόριθμος που θα χρησιμοποιηθεί στα πλαίσια της συγκεκριμένης εργασίας είναι ο Nondominated Sorting Genetic Algorithm - II (NSGA - II). Στον αλγόριθμο αυτό, [15] δημιουργείται αρχικά ένας αρχικός πατρικός πληθυσμός P_0 μεγέθους N . Ο πληθυσμός αυτός ταξινομείται με βάση τη μη κυριαρχία κατά Pareto. Σε κάθε λύση αντιστοιχεί ένα επίπεδο μη κυριαρχίας (1 για το καλύτερο επίπεδο, 2 για το αμέσως επόμενο). Στην αρχή, εφαρμόζονται διάφοροι τελεστές (όπως μεταθέσεις) για να δημιουργηθεί ένας πληθυσμός απογόνων Q_0 μεγέθους N . Στο σημείο αυτό η περιγραφή του αλγορίθμου συνεχίζεται για το πως γίνεται η μετάβαση από τη γενιά t στη γενιά $t+1$. Αρχικά έχουμε έναν πληθυσμό $R_t = P_t \cup Q_t$. Ο πληθυσμός R_t ταξινομείται με βάση τη μη κυριαρχία. Οι λύσεις που ανήκουν στο καλύτερο μη κυριαρχούμενο σύνολο F_1 είναι από τις καλύτερες πιθανές λύσεις στον πληθυσμό και δίνεται περισσότερη έμφαση σε αυτές. Εάν το μέγεθος του συνόλου F_1 είναι μικρότερο από N , τότε επιλέγεται ολόκληρο το σύνολο αυτό για τον πληθυσμό P_{t+1} . Τα υπόλοιπα μέλη του πληθυσμού P_{t+1} επιλέγονται από τα αμέσως επόμενα καλύτερα σύνολα F_2, F_3, \dots μέχρι να συμπληρωθούν N

μέλη για αυτόν. Στη συνέχεια, με εφαρμογή τελεστών παράγεται ο πληθυσμός των απογόνων Q_{t+1} μεγέθους N . Έπειτα, η διαδικασία επαναλαμβάνεται για τον νέο πληθυσμό R_{t+1} .



Εικόνα 2-5: Η διαδικασία που ακολουθεί ο NSGA – II.

Η συνολική πολύπλοκότητα του αλγορίθμου είναι $O(MN^2)$, που είναι ο πρακτικά ο χρόνος που χρειάζεται για τις συνολικές ταξινομήσεις των $2N$ μελών του πληθυσμού στις M επαναλήψεις που θα εκτελεστεί η διαδικασία.

2.5. ΑΠΟΣΤΑΣΗ HAUSDORFF

Η απόσταση Hausdorff [16] μετράει πόσο μακριά βρίσκεται ένα υποσύνολο ενός μετρικού χώρου από ένα άλλο. Δύο σύνολα είναι κοντά σύμφωνα με την απόσταση Hausdorff εάν κάθε σημείο του ενός συνόλου είναι κοντά με κάποιο σημείο του άλλου συνόλου. Πρόκειται για την μεγαλύτερη από όλες τις αποστάσεις από ένα σημείο στο ένα σύνολο προς το πιο κοντινό του από το άλλο σύνολο. Με τη βοήθεια μαθηματικού τύπου, η απόσταση αυτή ενός συνόλου Q από ένα σύνολο P , εκφράζεται ως:

$$d_h(P, Q) = \max_{\forall p \in P} \left\{ \min_{\forall q \in Q} \{d(p, q)\} \right\}$$

όπου με d συμβολίζεται η Ευκλείδεια απόσταση ανάμεσα στα p και q .

3

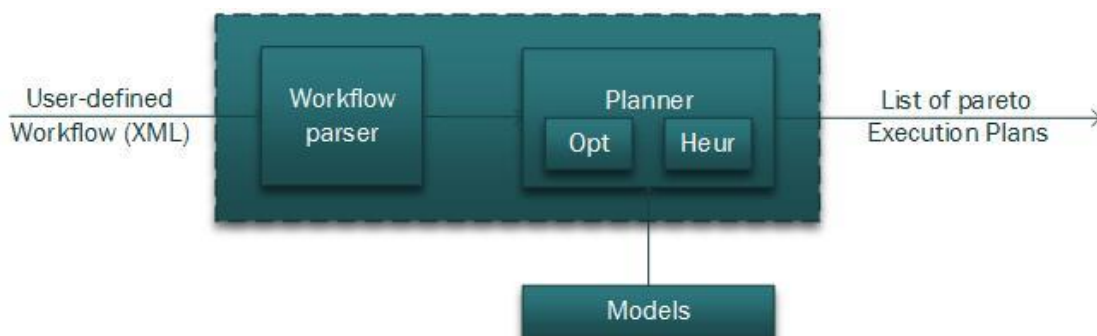
ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό γίνεται περιγραφή του συστήματος που υλοποιήθηκε στα πλαίσια της εργασίας. Πιο αναλυτικά, γίνεται αναφορά στην αρχή στις τεχνολογίες που χρησιμοποιήθηκαν. Στη συνέχεια, αναλύονται τα τρία βασικά μέρη του συστήματος, τα οποία είναι τα ακόλουθα:

- η ανάγνωση των αρχείων που περιγράφουν τη ροή εργασιών.
- ο αλγόριθμος υπολογισμού των βέλτιστων λύσεων κατά Pareto.
- ο προσεγγιστικός τρόπος εύρεσης λύσεων.

3.1. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Στην Εικόνα 3-1 περιγράφεται η αρχιτεκτονική του συστήματος που υλοποιήθηκε στην παρούσα εργασία. Το σύστημα αρχικά λαμβάνει ως είσοδο την περιγραφή μίας ροής εργασιών με τη μορφή ενός XML αρχείου. Αφού λάβει την περιγραφή αυτή, στη συνέχεια από δοσμένα μοντέλα λαμβάνονται οι χρόνοι εκτέλεσης και το κόστος κάθε επιμέρους εργασίας σε κάποια μηχανή και ο μετα-βελτιστοποιητής χρησιμοποιεί όποιον αλγόριθμο έχει επιλεγεί (βέλτιστος ή ευριστικός) για την εύρεση των πλάνων εκτέλεσης που ανήκουν στο σύνολο των βέλτιστων κατά Pareto λύσεων. Αναλυτική περιγραφή των επιμέρους τμημάτων του συστήματος ακολουθεί σε επόμενες υποπαραγράφους.



Εικόνα 3-1: Αρχιτεκτονική του συστήματος που υλοποιήθηκε

3.2. ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

3.2.1. JAVA ΚΑΙ ECLIPSE

Η γλώσσα προγραμματισμού που επιλέχθηκε για την υλοποίηση του συστήματος είναι η Java. Πρόκειται [17] για μία γενικού σκοπού γλώσσα προγραμματισμού, η οποία είναι ταυτόχρονη, βασίζεται σε κλάσεις, αντικειμενοστραφής και έχει σχεδιαστεί ώστε να έχει όσο το δυνατόν λιγότερες εξαρτήσεις υλοποίησης. Η έκδοση της Java που χρησιμοποιήθηκε στην υλοποίηση είναι η Java SE 1.7 της Oracle.

Το περιβάλλον στο οποίο αναπτύχθηκε το σύστημα είναι το Eclipse. Πρόκειται για ένα περιβάλλον ανάπτυξης εφαρμογών και είναι από τα ευρέως χρησιμοποιούμενα για τον προγραμματισμό σε Java. Η έκδοση που χρησιμοποιήθηκε είναι το Eclipse Mars.

3.2.2. MOEA FRAMEWORK

Το MOEA Framework [18] είναι μια ελεύθερη και ανοιχτού κώδικα βιβλιοθήκη Java για ανάπτυξη και πειραματισμό με εξελικτικούς αλγορίθμους πολλαπλών κριτηρίων (MOEAs) και άλλους γενικού σκοπού αλγορίθμους βελτιστοποίησης. Ένα πλήθος αλγορίθμων παρέχονται έτοιμοι για χρήση, όπως ο NSGA-II. Επιπλέον, το framework παρέχει όλα τα απαραίτητα εργαλεία για γρήγορο σχεδιασμό, ανάπτυξη, εκτέλεση και στατιστικό έλεγχο αλγορίθμων βελτιστοποίησης.

Εκτός από κάποια προκαθορισμένα προβλήματα βελτιστοποίησης που παρέχονται μαζί με το framework, μπορούν [19] να οριστούν νέα προβλήματα με μία ή περισσότερες μεταβλητές απόφασης διαφόρων τύπων. Σε αυτούς περιλαμβάνονται συνήθεις αναπαραστάσεις όπως δυαδικές συμβολοσειρές, πραγματικοί αριθμοί και συνδυασμοί. Επιπροσθέτως, υποστηρίζει γραμματικές σε μορφή Backus – Naur. Από τη στιγμή που ένα πρόβλημα θα έχει οριστεί, ο χρήστης μπορεί να βελτιστοποιήσει με χρήση οποιουδήποτε από τους αλγορίθμους που παρέχονται.

Στην παρούσα εργασία, το MOEA framework χρησιμοποιείται για να βελτιστοποιήσει το πρόβλημα των ροών εργασιών με χρήση του αλγορίθμου NSGA-II. Σε επόμενη υποπαράγραφο, περιγράφεται αναλυτικά ο τρόπος που χρησιμοποιήθηκε το framework για την επίλυση του προβλήματος.

3.2.3. PEGASUS: ΓΕΝΝΗΤΡΙΑ ΡΟΩΝ ΕΡΓΑΣΙΩΝ

Το Pegasus περιλαμβάνει μια σειρά από τεχνολογίες που μπορούν να κατασκευάσουν επιστημονικές ροές εργασιών, οι οποίες μπορούν να εκτελεστούν είτε σε ένα υπολογιστή είτε σε κατανεμημένο περιβάλλον. Από αυτό λαμβάνεται σε μορφή xml η περιγραφή μιας ροής εργασιών, η οποία θα χρησιμοποιηθεί για τον έλεγχο του συστήματος που υλοποιείται στην εργασία.

Υπάρχουν διάφοροι τύποι ροών εργασιών που παράγει αυτή η γεννητριά. Οι τύποι αυτοί θα περιγραφούν στο επόμενο κεφάλαιο μαζί με τα πειραματικά δεδομένα που τους αφορούν. Ωστόσο, ανεξαρτήτως της μορφής της ροής, το αρχείο xml που την περιγράφει έχει πάντα την ίδια δομή. Στο πρώτο κομμάτι του, περιέχει την πληροφορία ποια εργασία αντιστοιχεί σε κάθε κόμβο της ροής, ποια αρχεία χρειάζεται για να λειτουργήσει και ποια είναι τα αρχεία εξόδου της (Εικόνα 3-2). Στο δεύτερο κομμάτι του, υπάρχει η πληροφορία για τη μορφή της εργασιών. Συγκεκριμένα, για κάθε κόμβο που έχει κάποιο πατέρα, αναφέρεται ποιος ή ποιοι είναι οι κόμβοι αυτοί στο γράφο (Εικόνα 3-3).

```
<job id="ID00000" namespace="Montage" name="mProjectPP" version="1.0" runtime="13.39">
  <uses file="region.hdr" link="input" register="true" transfer="true" optional="false" type="data" size="304"/>
  <uses file="2mass-atlas-ID00000s-jID00000.fits" link="input" register="true" transfer="true" optional="false" type="data" size="4222080"/>
  <uses file="p2mass-atlas-ID00000s-jID00000.fits" link="output" register="true" transfer="true" optional="false" type="data" size="4167312"/>
  <uses file="p2mass-atlas-ID00000s-jID00000_area.fits" link="output" register="true" transfer="true" optional="false" type="data" size="4167312"/>
</job>
<job id="ID00001" namespace="Montage" name="mProjectPP" version="1.0" runtime="13.83">
  <uses file="region.hdr" link="input" register="true" transfer="true" optional="false" type="data" size="304"/>
  <uses file="2mass-atlas-ID00001s-jID00001.fits" link="input" register="true" transfer="true" optional="false" type="data" size="4222080"/>
  <uses file="p2mass-atlas-ID00001s-jID00001.fits" link="output" register="true" transfer="true" optional="false" type="data" size="4171851"/>
  <uses file="p2mass-atlas-ID00001s-jID00001_area.fits" link="output" register="true" transfer="true" optional="false" type="data" size="4171851"/>
</job>
```

Εικόνα 3-2: Τμήμα του xml αρχείου που περιγράφει εργασίες κόμβων

```
<child ref="ID00005">
  <parent ref="ID00001"/>
  <parent ref="ID00000"/>
</child>
<child ref="ID00006">
  <parent ref="ID00001"/>
  <parent ref="ID00000"/>
</child>
<child ref="ID00007">
  <parent ref="ID00001"/>
  <parent ref="ID00003"/>
</child>
<child ref="ID00008">
  <parent ref="ID00002"/>
  <parent ref="ID00000"/>
</child>
```

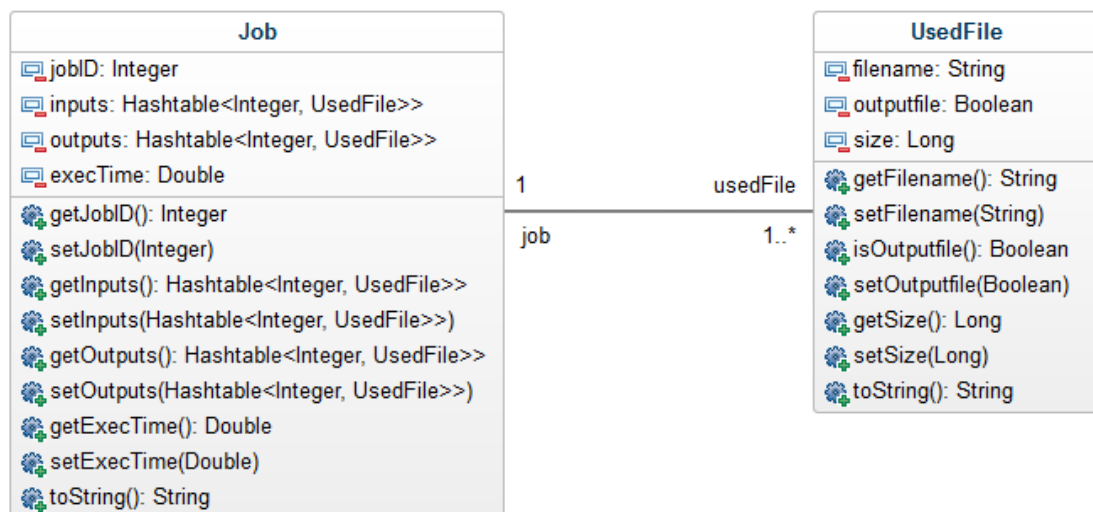
Εικόνα 3-3: Τμήμα του xml αρχείου που συνδέει κάποιο κόμβο με τους πατέρες του

3.3. ΠΕΡΙΓΡΑΦΗ ΒΑΣΙΚΩΝ ΚΛΑΣΕΩΝ

Για την καλύτερη κατανόηση της περιγραφής του συστήματος, που θα ακολουθήσει στις επόμενες υποπαραγράφους, κρίνεται απαραίτητη μια σύντομη αναφορά σε κάποιες βασικές κλάσεις, οι οποίες αντιστοιχούν σε στοιχειώδη οντότητες του προβλήματος. Αυτές οι κλάσεις αντιπροσωπεύουν εργασίες, κόμβους της ροής εργασιών, μηχανές, τα κριτήρια και δομές που αντιστοιχίζουν έναν κόμβο σε μία μηχανή.

3.3.1. ΟΙ ΚΛΑΣΕΙΣ JOB ΚΑΙ USEDFILE

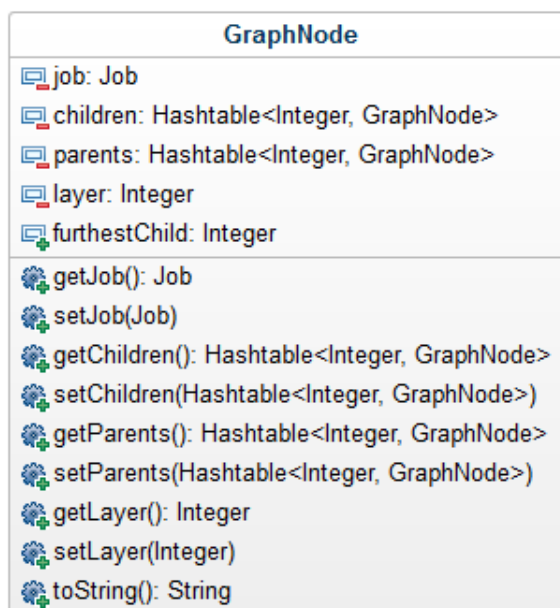
Η κλάση UsedFile χρησιμοποιείται για να περιγράψει ένα αρχείο εισόδου ή εξόδου για κάποια στοιχειώδη εργασία. Η εργασία αυτή περιγράφεται από την κλάση Job. Στην Εικόνα 3-3, δίνεται ένα διάγραμμα κλάσεων για αυτές τις δύο κλάσεις.



Εικόνα 3-4: Διάγραμμα κλάσης για τις κλάσεις *Job*, *UsedFile*

Όπως φαίνεται από το παραπάνω διάγραμμα κλάσεων, η κλάση UsedFile περιέχει όλη την απαραίτητη πληροφορία για ένα αρχείο που χρησιμοποιείται όπως το όνομα του, το αν πρόκειται για αρχείο εισόδου ή εξόδου και το μέγεθος του σε bytes. Μια εργασία αποτελείται από έναν ακέραιο που είναι το αναγνωριστικό της, τα αρχεία εισόδου που περιμένει για να εκτελεστεί, τα αρχεία εξόδου που παράγει και τον χρόνο εκτέλεσης της. Παράλληλα, γίνεται φανερό ότι οι κλάσεις διαθέτουν κατάλληλες μεθόδους για την ανάθεση και την ανάγνωση τιμών στα πεδία τους.

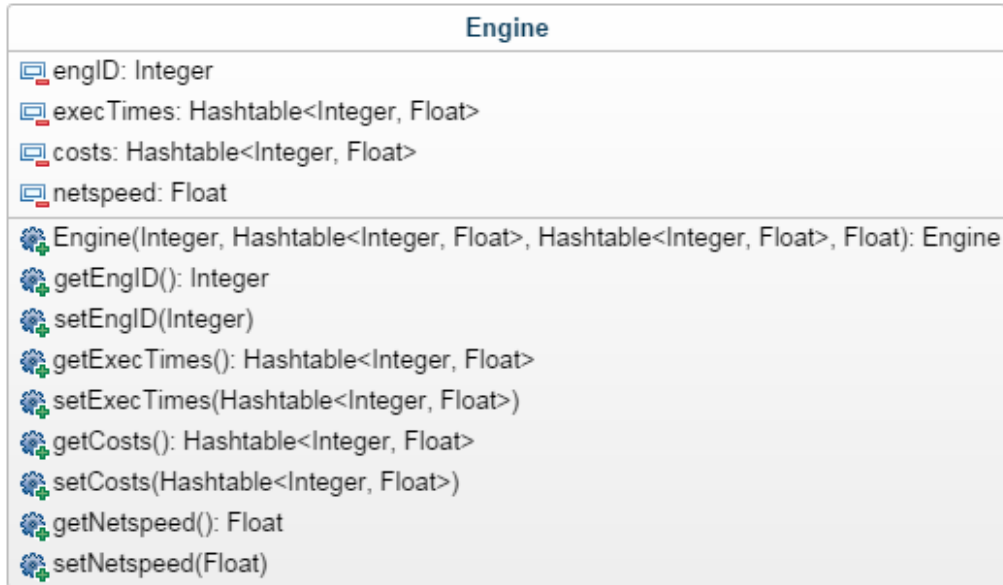
3.3.2. Η ΚΛΑΣΗ GRAPHNODE



Εικόνα 3-5: Η κλάση *GraphNode*

Η κλάση `GraphNode` αντιπροσωπεύει έναν κόμβο της ροής εργασιών. Περιέχει όλες τις απαραίτητες πληροφορίες για να περιγράψει τον κόμβο, δηλαδή τις πληροφορίες για την εργασία που αντιστοιχεί στον κόμβο αυτό, ποιοι είναι οι πατρικοί κόμβοι και ποιοι τα παιδιά του. Επίσης, υπάρχουν και άλλα δύο γνωρίσματα, τα οποία κατέχουν σημαντικό ρόλο στη σωστή λειτουργία του αλγορίθμου, όπως θα περιγραφεί στη συνέχεια. Το πρώτο (`layer`) είναι ένας ακέραιος μεγαλύτερος ή ίσος του μηδέν. Όταν δύο κόμβοι έχουν την ίδια τιμή σε αυτή τη μεταβλητή μπορούν να εκτελεστούν ταυτόχρονα σε ίδιες ή διαφορετικές μηχανές. Επίσης, το δεύτερο (`furthestChild`) δηλώνει ποιο είναι το πιο μακρινό παιδί του κόμβου στον γράφο. Η χρησιμότητα του πεδίου αυτού θα φανεί στην επεξήγηση του ακριβούς αλγορίθμου σε επόμενη υποπαράγραφο. Τέλος, υπάρχουν και πάλι μέθοδοι για την αναθέση τιμών στα πεδία αυτά, όπως και μέθοδοι που επιστρέφουν την τιμή των τιμών αυτών.

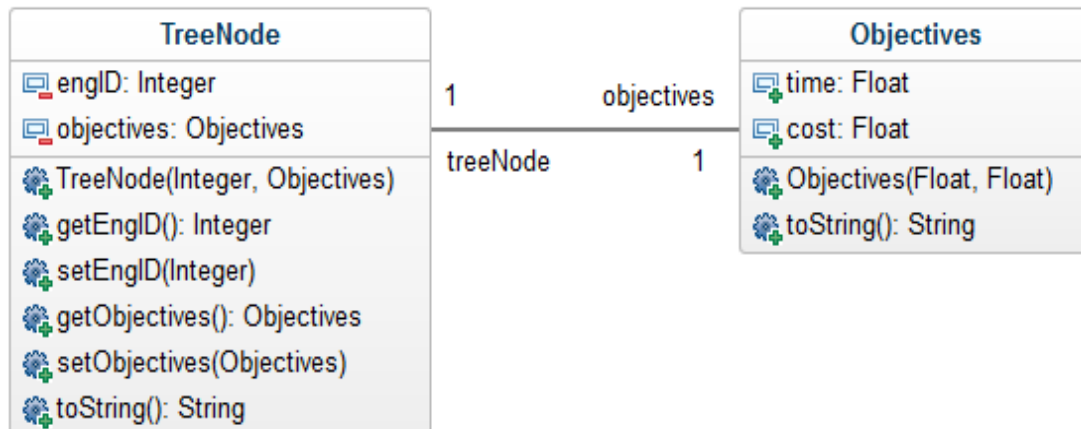
3.3.3. Η ΚΛΑΣΗ ENGINE



Εικόνα 3-6: Η κλάση *Engine*

Η κλάση *Engine* χρησιμοποιείται για την περιγραφή μίας οποιασδήποτε μηχανής εκτέλεσης που είναι διαθέσιμη. Αρχικά, περιέχει κάποιον ακέραιο ως αναγνωριστικό για να μπορεί να διακρίνεται από τις υπόλοιπες. Επίσης, περιέχει τον χρόνο και το κόστος για την εκτέλεση κάθε μίας από τις εργασίες που μπορούν να εκτελεστούν στη δεδομένη μηχανή. Τέλος, περιέχει την πληροφορία για την ταχύτητα του δικτύου, στο οποίο βρίσκεται η μηχανή εκτέλεσης. Στην Εικόνα 3-6 δίνεται το διάγραμμα της κλάσης αυτής.

3.3.4. ΟΙ ΚΛΑΣΕΙΣ OBJECTIVES ΚΑΙ TREENODE

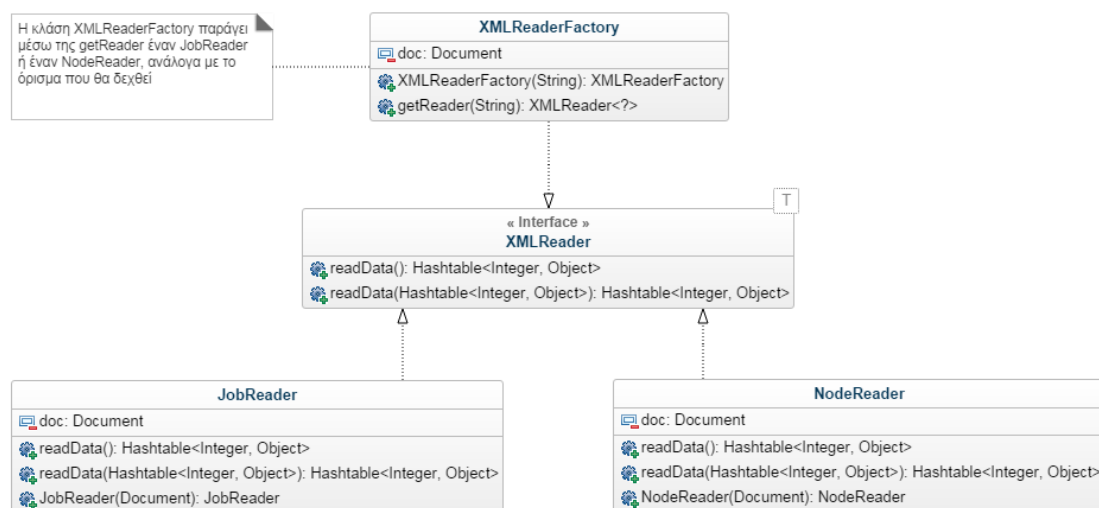


Εικόνα 3-7: Διάγραμμα κλάσεων για τις *TreeNode*, *Objectives*

Πρόκειται για δύο από τις πιο βασικές κλάσεις στην εκτέλεση, τόσο του ακριβούς όσο και του προσεγγιστικού αλγορίθμου. Η κλάση Objectives περιγράφει για μια δεδομένη αντιστοίχιση κόμβου σε μηχανή τις τιμές των κριτηρίων του χρόνου και του κόστους, όπως έχουν υπολογιστεί με βάση τις αναδρομικές σχέσεις (2.1) και (2.3) αντίστοιχα. Κάθε αντικείμενο της κλάσης TreeNode αντιστοιχεί σε μία τέτοια αντιστοίχιση και για αυτό οι πληροφορίες που περιέχει είναι το αναγνωριστικό της μηχανής που θα τοποθετηθεί μια εργασία και τα αντίστοιχα κριτήρια (objectives). Και οι δύο κλάσεις διαθέτουν κατάλληλους κατασκευαστές (constructors) και μεθόδους τόσο για την ανάθεση τιμών στα πεδία όσο και για την ανάγνωση των αντίστοιχων τιμών. Ο τρόπος λειτουργίας τους θα φάνει στην περιγραφή των τεχνικών εύρεσης των βέλτιστων λύσεων.

3.4. ΑΝΑΓΝΩΣΗ ΑΠΟ XML ΑΡΧΕΙΟ

Όταν αρχίσει η εκτέλεση του συστήματος, το πρώτο συμβάν που λαμβάνει χώρα είναι η ανάγνωση από το αρχείο xml που περιγράφει την ροή εργασιών. Με την ολοκλήρωση της ανάγνωσης έχουν παραχθεί τόσα στιγμιότυπα της κλάσης GraphNode όσοι είναι και οι κόμβοι του γράφου που περιγράφει την ροή εργασιών.



Εικόνα 3-8: Διάγραμμα των κλάσεων που συμμετέχουν στην ανάγνωση της ροής εργασιών από το xml αρχείο.

Στην Εικόνα 3-8 δίνεται το διάγραμμα των κλάσεων που συμμετέχουν στην διαδικασία της ανάγνωσης. Το σχεδιαστικό μοτίβο που ακολουθήθηκε για την υλοποίηση αυτού του τμήματος του συστήματος είναι το Factory. Σύμφωνα με αυτό το μοτίβο, έχουμε μία κλάση – εργοστάσιο (XMLReaderFactory), η οποία αναλόγως με την περίπτωση θα

επιστρέφει τον κατάλληλο αναγνώστη για το xml αρχείο. Για τη σωστή λειτουργία της διαδικασίας, αρχικά κατασκευάζεται ένας στιγμιότυπο της κλάσης JobReader από το εργοστάσιο, το οποίο χρησιμοποιείται για να κατασκευαστούν αντικείμενα της κλάσης GraphNode, τα οποία θα περιέχουν όλες τις πληροφορίες για τις εργασίες που εκτελεί το καθένα. Στη συνέχεια, το εργοστάσιο χρησιμοποιείται για να κατασκευαστεί ένα στιγμιότυπο της κλάσης NodeReader. Αυτό αξιοποιείται ώστε κάθε ένα αντικείμενο που κατασκευάστηκε προηγουμένως να γνωρίζει τους πατέρες και τα παιδιά του στο γράφο. Η αλγοριθμική περιγραφή της διαδικασίας φαίνεται στην επόμενη εικόνα.

- Κατασκευή νέου εργοστασίου παραγωγής αναγνωστών XML.
- Παραγωγή JobReader από το εργοστάσιο.
- Χρήση JobReader για ανάγνωση των εργασιών από το αρχείο μαζί με τα αρχεία εισόδου και εξόδου της καθεμίας.
- Παραγωγή NodeReader από το εργοστάσιο.
- Χρήση NodeReader για ανάγνωση των σχέσεων μεταξύ των κόμβων.

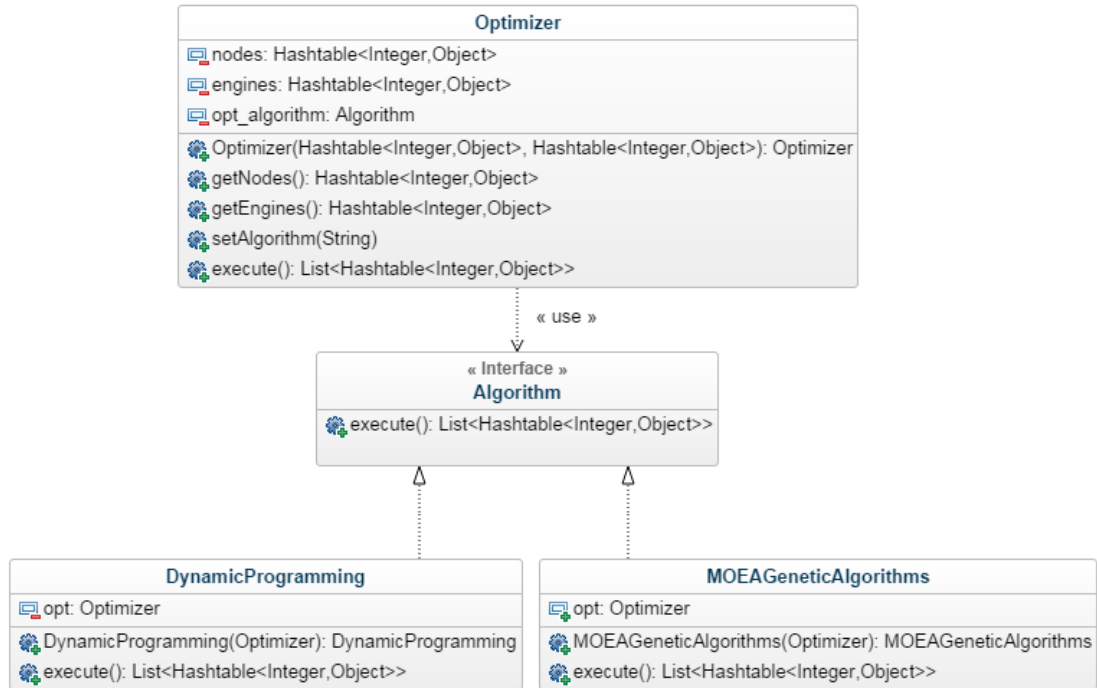
Εικόνα 3-9: Αλγοριθμική περιγραφή της διαδικασίας ανάγνωσης από αρχείο.

Αφού ολοκληρωθεί η παραπάνω διαδικασία είναι διαθέσιμοι όλοι οι κόμβοι του γράφου εργασιών με όλα τα απαραίτητα στοιχεία για να ξεκινήσει η διαδικασία για την αναζήτηση των επιθυμητών λύσεων. Οι κόμβοι έχουν αποθηκευτεί σε δομή τύπου Hashtable, ώστε να είναι προσπελάσιμοι όποτε χρειάζονται σε σταθερό χρόνο $O(1)$.

3.5. ΑΝΑΖΗΤΗΣΗ ΒΕΛΤΙΣΤΩΝ ΛΥΣΕΩΝ ΚΑΤΑ PARETO

Αφού έχει ολοκληρωθεί η ανάγνωση του XML αρχείου που περιγράφει τον γράφο, στην συνέχεια το σύστημα επιχειρεί την αναζήτηση της βέλτιστης λύσης με δύο τρόπους. Πρώτα πραγματοποιείται η αναζήτηση της με έναν ακριβή αλγόριθμο. Στη συνέχεια, ξεκινάει η αναζήτηση μιας προσεγγιστικής λύσης με τον γενετικό αλγόριθμο NSGA-II. Για κάθε μία από αυτές τις μεθόδους, το σύστημα κράταει το χρόνο εκτέλεσης της και στο τέλος υπολογίζει το ποσοστό ταύτισης της λύσης που έδωσε η προσεγγιστική μέθοδος σε σχέση με την πραγματική.

Στην Εικόνα 3-10 δίνεται το διάγραμμα των κλάσεων που συμμετέχουν σε όλη αυτή τη διαδικασία.



Εικόνα 3-10: Διάγραμμα των κλάσεων που συμμετέχουν στη διαδικασία εύρεσης της ακριβούς και της προσεγγιστικής λύσης.

Η γενική αυτή διαδικασία ακολουθεί το σχεδιαστικό μοτίβο Strategy. Επομένως, δημιουργείται ένα αντικείμενο της κλάσης Optimizer, το οποίο ανάλογως με τον αλγόριθμο (στρατηγική) που θα του ανατεθεί, αναλαμβάνει να βρει τη λύση στο πρόβλημα βελτιστοποίησης. Άρα χρησιμοποιώντας το ίδιο αντικείμενο λαμβάνονται δύο διαφορετικές λύσεις, η καθεμία σύμφωνα με τη στρατηγική που έχει ανατεθεί στο βελτιστοποιητή. Στην Εικόνα 3-11 περιγράφεται η συνολική αυτή διαδικασία.

- Κατασκευή στιγμιότυπου anOptimizer της κλάσης Optimizer
- anOptimizer.opt_algorithm <- DynamicProgramming
- Έναρξη χρονομέτρου
- βέλτιστηΛύση <- anOptimizer.execute()
- Λήξη χρονομέτρου και αποθήκευση χρόνου εκτέλεσης σε χρόνοςΕκτιΒέλτ
- anOptimizer.opt_algorithm <- MOEAGeneticAlgorithms
- Έναρξη χρονομέτρου
- προσεγγιστικήΛύση <- anOptimizer.execute()
- Λήξη χρονομέτρου και αποθήκευση χρόνου εκτέλεσης σε χρόνοςΕκτιΠρος
- Υπολογισμός ποσοστού ταύτισης ανάμεσα σε βέλτιστηΛύση και προσεγγιστικήΛύση

Εικόνα 3-11: Αλγοριθμική περιγραφή γενικής λειτουργίας του συστήματος στην πορεία εύρεσης λύσης.

Στις δύο επόμενες υποπαραγράφους γίνεται αναλυτική περιγραφή τόσο του αλγόριθμου που βρίσκει την βέλτιστη λύση όσο και αυτού που βρίσκει την προσεγγιστική λύση.

3.5.1. ΠΕΡΙΓΡΑΦΗ ΑΛΓΟΡΙΘΜΟΥ ΕΥΡΕΣΗΣ ΒΕΛΤΙΣΤΗΣ ΛΥΣΗΣ ΚΑΤΑ PARETO

Η λειτουργία του αλγορίθμου αυτού στηρίζεται στη γενική αρχή του δυναμικού προγραμματισμού. Συγκεκριμένα, για κάθε μία αντιστοίχιση κόμβου – μηχανής, βρίσκουμε ποιος είναι χρόνος εκτέλεσης και ποιο το κόστος μέχρι εκείνο το σημείο με βάση όλα τα προηγούμενα διαθέσιμα ζεύγη. Όταν η εκτέλεση του αλγορίθμου φτάσει σε κάποιο κόμβο ο οποίος δεν μπορεί να εκτελεστεί ταυτόχρονα με κάποιον άλλον και επίσης δεν υπάρχει ροή δεδομένων από κάποιον πρόγονο του σε κάποιον απόγονο του, τότε γίνεται «κλάδεμα» των μέχρι τώρα υπολογισμένων δεδομένων. Συγκεκριμένα, στον κόμβο αυτό, για κάθε μία μηχανή, αφού υπολογιστούν όλες οι πιθανές τιμές των κριτηρίων από όλους τους προγόνους, τότε υπολογίζεται το βέλτιστο σύνολο κατά Pareto, το οποίο είναι αυτό που διατηρείται για τη συνέχιση των υπολογισμών στους επόμενους κόμβους.

Ο αλγόριθμος περιγράφεται αναλυτικά στην Εικόνα 3-10 με τη μορφή ψευδοκώδικα. Όπως θα γίνει φανερό και από την πειραματική του αξιολόγηση, ο αλγόριθμος αυτός είναι αρκετά καλός σε απλούς γράφους, όπως γραμμικούς. Ωστόσο, σε περίπλοκους γράφους ο χρόνος εκτέλεσης του είναι αρκετά μεγάλος. Συγκεκριμένα, στην χειρότερη περίπτωση όπου δεν είναι πουθενά δυνατό το κλάδεμα των λύσεων, ο αλγόριθμος εκφύλιζεται σε εξαντλητική αναζήτηση με χρόνο εκτέλεσης της τάξης $O(m^n)$, όπου m το πλήθος των μηχανών και n το πλήθος των κόμβων του γράφου.

Στον ψευδοκώδικα που δίνεται στην Εικόνα 3-10, αναφέρονται κάποιες συναρτήσεις που χρησιμοποιούνται για την εκτέλεση του. Οι συναρτήσεις *υπολογισμόςΧρόνου* και *υπολογισμόςΚόστους* λειτουργούν με βάση τα όσα περιγράφηκαν στο κεφάλαιο 2 για τον υπολογισμό του χρόνου και του κόστους ενός δεδομένου ζεύγους σε ένα πλάνο εκτέλεσης και οι ψευδοκώδικες τους δίνονται στις εικόνες 3-11 και 3-12 αντίστοιχα. Η συνάρτηση PARETO λαμβάνει ζεύγη κριτηρίων και επιστρέφει τις βέλτιστες λύσεις κατά Pareto και ο ψευδοκώδικας της δίνεται στην εικόνα 3-13.

```

κλάδεμαΣτονΚόμβο ← -1
προηγούμενοςΚόμβος ← NULL
Για κάθε κόμβο n:
    νέοςΚόμβος ← {}
    Εάν προηγούμενοςΚόμβος = NULL:
        Για κάθε μηχανή e:
             $t \leftarrow t_n^e$ 
             $c \leftarrow c_n^e$ 
            νέοςΚόμβος ← νέοςΚόμβος  $\cup$  {(n, e, t, c)}
    Αλλιώς:
        Για κάθε μηχανή e:
            temp ← {}
            Για κάθε prev  $\in$  προηγούμενοςΚόμβος:
                 $t \leftarrow$  υπολογισμόςΧρόνου(n, e, prev)
                 $c \leftarrow$  υπολογισμόςΚόστους(n, e, prev)
                temp ← temp  $\cup$  {(n, e, t, c)}
            Εάν n.id = κλάδεμαΣτονΚόμβο:
                temp ← PARETO(temp)
            νέοςΚόμβος ← νέοςΚόμβος  $\cup$  temp
    προηγούμενοςΚόμβος ← νέοςΚόμβος
    Εάν n.furthestChild > κλάδεμαΣτονΚόμβο
        κλάδεμαΣτονΚόμβο ← n.furthestChild

```

Εικόνα 3-12: Ψευδοκώδικας περιγραφής αλγορίθμου εύρεσης βέλτιστων λύσεων κατά Pareto

```

υπολογισμόςΧρόνου(n, e, prev):
    χρόνος ← 0
    Εάν n.Parents <> NULL:
        Για κάθε p  $\in$  n.Parents:
            χρόνοςΓονιού ← prev. $t_p^w$ 
            Εάν e <> w:
                Για κάθε αρχείο f που μεταφέρεται:
                    χρόνοςΓονιού ← χρόνοςΓονιού +
                        f.size/ελάχιστηΤαχύτητα(e, w)
            Εάν χρόνοςΓονιού > χρόνος:
                χρόνος ← χρόνοςΓονιού
    χρόνος ← χρόνος +  $t_n^e$ 
    Επέστρεψε χρόνος

```

Εικόνα 3-13: Ψευδοκώδικας της συνάρτησης υπολογισμόςΧρόνου

```
υπολογισμόςΚόστους(n, e, prev):  
    κόστος ← prev. $c_{n-1}^w$  +  $c_n^e$   
    Επέστρεψε κόστος
```

Εικόνα 3-14: Ψευδοκώδικας της συνάρτησης υπολογισμόςΚόστους

```
PARETO(list(n, e, t, c)):  
    ταξινομημένηΚατάΚόστοςΛίστα ← sortc(list(n, e, t, c))  
    μέγιστοςΧρόνος ← ∞  
    paretoList ← {}  
    Για κάθε (n, e, t, c) ∈ ταξινομημένηΚατάΚόστοςΛίστα:  
        Εάν t < μέγιστοςΧρόνος:  
            paretoList ← paretoList ∪ {(n, e, t, c)}  
            μέγιστοςΧρόνος ← t  
    Επέστρεψε paretoList
```

Εικόνα 3-15: Ψευδοκώδικας της συνάρτησης PARETO

3.5.2. ΠΕΡΙΓΡΑΦΗ ΑΛΓΟΡΙΘΜΟΥ ΕΥΡΕΣΗΣ ΠΡΟΣΕΓΓΙΣΤΙΚΗΣ ΛΥΣΗΣ

Για την εύρεση της προσεγγιστικής λύσης στο πρόβλημα χρησιμοποιείται ο γενετικός αλγόριθμος NSGA-II. Ο αλγόριθμος αυτός παρέχεται έτοιμος μέσα από το MOEA framework και για αυτό στην υποπαράγραφο αυτή περιγράφεται ο τρόπος που ορίζεται το πρόβλημα στο framework και τι αποτελέσματα λαμβάνονται από αυτό.

Αρχικά, για να γίνει ο ορισμός του προβλήματος πρέπει να οριστούν οι μεταβλητές αυτού. Στο συγκεκριμένο πρόβλημα, ως μεταβλητές ορίζονται τόσες ακέραιου τύπου μεταβλητές, όσοι και οι κόμβοι του γράφου. Κάθε μία από αυτές μπορεί να λάβει ως δύνατες τιμές κάποιο από τα αναγνωριστικά των μηχανών που είναι διαθέσιμα. Έπειτα, ορίζονται ως κριτήρια προς βελτιστοποίηση ο χρόνος εκτέλεσης και το κόστος ενός πλάνου εκτέλεσης. Περιγράφοντας την διαδικασία υπολογισμού των κριτηρίων για δεδομένο πλάνο εκτέλεσης, το framework επιστρέφει ένα σύνολο από πλάνα εκτέλεσης, τα οποία προσεγγίζουν σε κάποιο βαθμό την βέλτιστη λύση.

4

ΠΕΙΡΑΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ – ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ

Στο κεφάλαιο αυτό χρησιμοποιούνται διάφορες ροές εργασιών για την αξιολόγηση των αλγορίθμων που υλοποιούνται στο σύστημα. Όλοι οι γράφοι δίνονται σε μορφή xml αρχείων, όπως αυτοί που παράγονται από το Pegasus. Συγκεκριμένα, δίνονται οι λύσεις που προκύπτουν ως πλάνα εκτέλεσης από τους δύο αλγορίθμους και οι χρόνοι εκτέλεσης τους ώστε να γίνει σύγκριση και αξιολόγηση αυτών.

Όλα τα πειράματα έχουν εκτελεστεί σε μηχανήματα του Εργαστηρίου Υπολογιστικών Συστημάτων και συγκεκριμένα στον scirouter σε μηχανήματα της ουράς termis. Τα μηχανήματα αυτά επιλέχθηκαν, καθώς έχουν μεγάλη RAM διαθέσιμη, της τάξης των 90GB. Για τα πειράματα χρησιμοποιήθηκε 1 πυρήνας και δόθηκε στο JVM όλη η διαθέσιμη RAM. Σε κάθε υποπαράγραφο στη συνέχεια παρουσιάζονται αποτελέσματα για διάφορες κατηγορίες πειραμάτων.

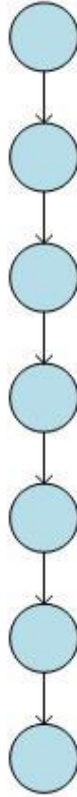
Χωρίς βλάβη της γενικότητας, έχει γίνει η υπόθεση ότι κάθε κόμβος της ροής μπορεί να εκτελέσει σε οποιαδήποτε από τη διαθέσιμες μηχανές. Επίπλέον, έχει θεωρηθεί ότι κάθε μηχανή έχει απεριόριστους πόρους διαθέσιμους, οπότε σε αυτήν μπορούν να εκτελούνται ταυτόχρονα άπειρες εργασίες.

4.1. ΈΛΕΓΧΟΣ ΟΡΘΟΤΗΤΑΣ ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΜΙΚΡΟΥΣ ΓΡΑΦΟΥΣ

Για την επιβεβαίωση της σωστής λειτουργίας τόσο του αλγορίθμου που βρίσκει τη βέλτιστη λύση τόσο αυτού που βρίσκει τη προσεγγιστική εκτελέστηκαν στην αρχή κάποια πειράματα σε μικρούς γράφους. Εκτελέστηκαν πειράματα με 3, 5, 7 και 9 διαθέσιμες μηχανές, ώστε να γίνει αντιληπτό ο τρόπος που η αύξηση τους επηρεάζει τον κάθε αλγόριθμο. Σε όλα τα πειράματα ο γενετικός αλγόριθμος NSGA-II έτρεξε για 10000 γενιές. Στη συνέχεια, παρουσιάζονται τα αντίστοιχα αποτελέσματα για κάθε είδος γράφου που χρησιμοποιήθηκε.

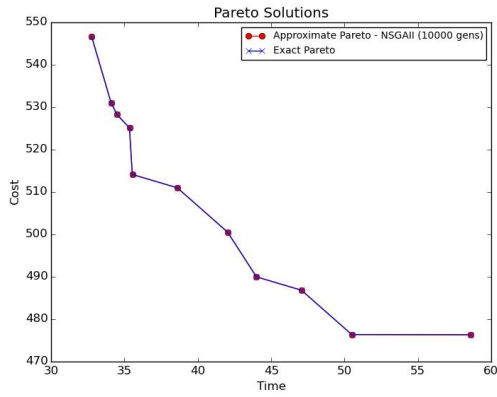
4.1.1. ΓΡΑΜΜΙΚΗ ΡΟΗ ΕΡΓΑΣΙΩΝ ΜΕ 7 ΚΟΜΒΟΥΣ

Αρχικά χρησιμοποιήθηκε η πιο απλή ροή εργασιών, δηλαδή η γραμμική. Ο γράφος που την περιγράφει φαίνεται στην Εικόνα 4-1.

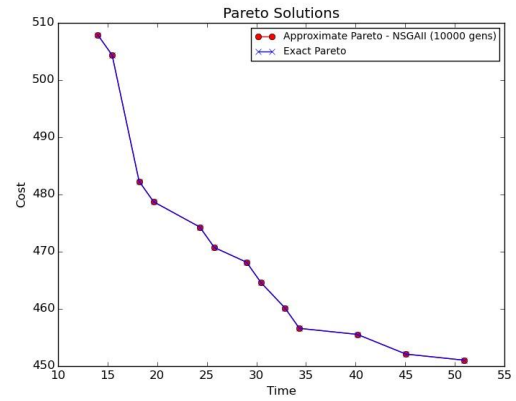


Εικόνα 4-1: Η γραμμική ροή εργασιών που χρησιμοποιείται στη σειρά των πειράματων

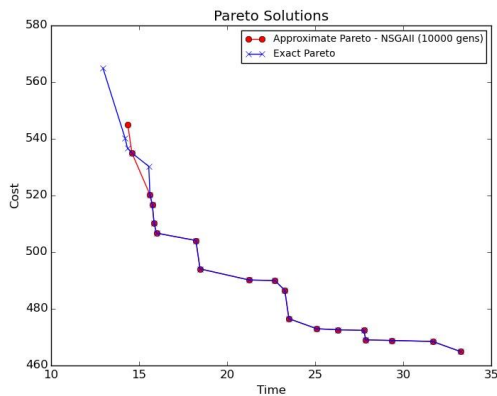
Στις επόμενες εικόνες παρουσιάζονται οι καμπύλες λύσεων που προέκυψαν για όλα τα διαθέσιμα πλήθη μηχανών. Όπως βλέπουμε σε όλες τις περιπτώσεις σχεδόν υπάρχει ταύτιση στις λύσεις των δύο αλγορίθμων. Ενδεικτικά, δίνονται και οι χρόνοι εκτέλεσης των δυο αλγορίθμων στην Εικόνα 4-3. Αναλυτικά θα σχολιαστούν όλα τα αποτελέσματα των απλών γράφων στο τέλος της παραγράφου 4.1.



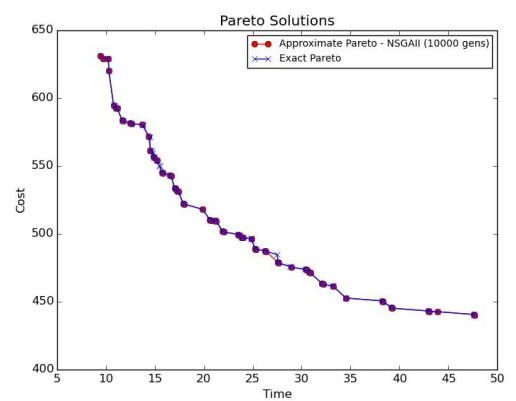
(α)



(γ)

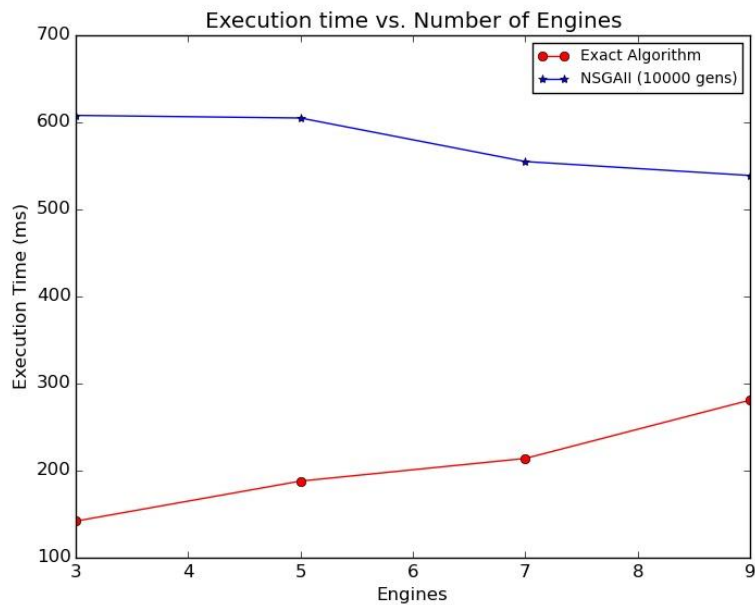


(β)



(δ)

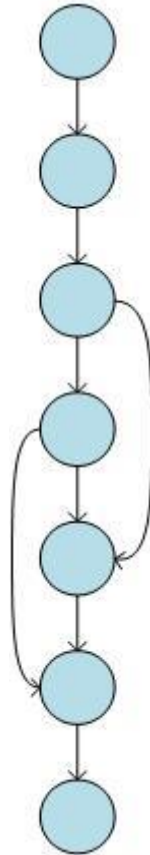
Εικόνα 4-2: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-1.



Εικόνα 4-3: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-1.

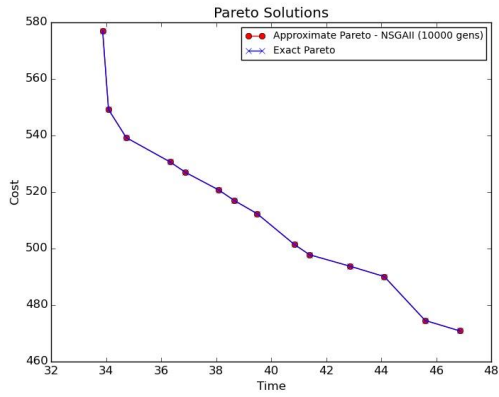
4.1.2. ΓΡΑΜΜΙΚΗ ΡΟΗ ΕΡΓΑΣΙΩΝ ΜΕ ΚΟΜΒΟΥΣ ΠΟΥ ΕΧΟΥΝ ΕΞΑΡΤΗΣΗ ΑΠΟ ΠΕΡΙΣΣΟΤΕΡΟΥΣ ΠΡΟΓΟΝΟΥΣ

Το ιδιαίτερο χαρακτηριστικό της είναι ότι υπάρχουν κόμβοι που δεν περιμένουν δεδομένα μόνο από τον αμέσως προηγούμενο κόμβο που εκτελέστηκε αλλά και από προγονικούς του. Αυτή η ροή εργασιών φαίνεται στην Εικόνα 4-4.

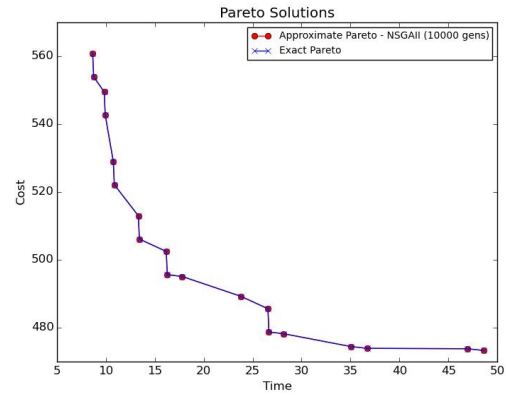


Εικόνα 4-4: Γραμμική Ροή εργασιών με κόμβους με παραπάνω εξαρτήσεις που χρησιμοποιείται στη σειρά των πειράματων

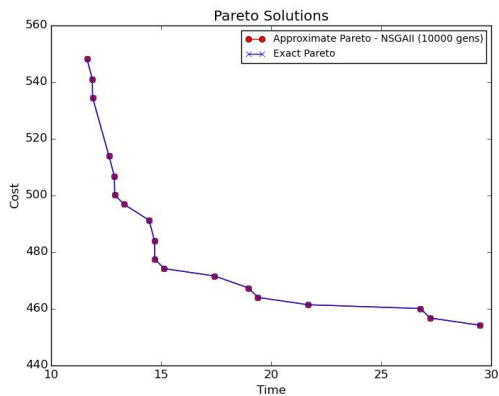
Στη συνέχεια, παρουσιάζονται και πάλι πρώτα τα σύνολα των λύσεων από τους δύο αλγορίθμους (Εικόνα 4-5) και έπειτα το διάγραμμα των χρόνων εκτέλεσης (Εικόνα 4-6).



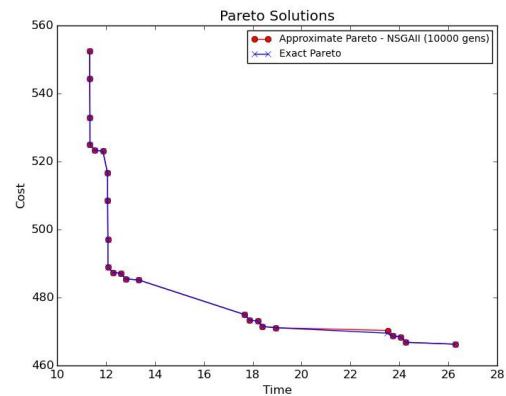
(α)



(γ)

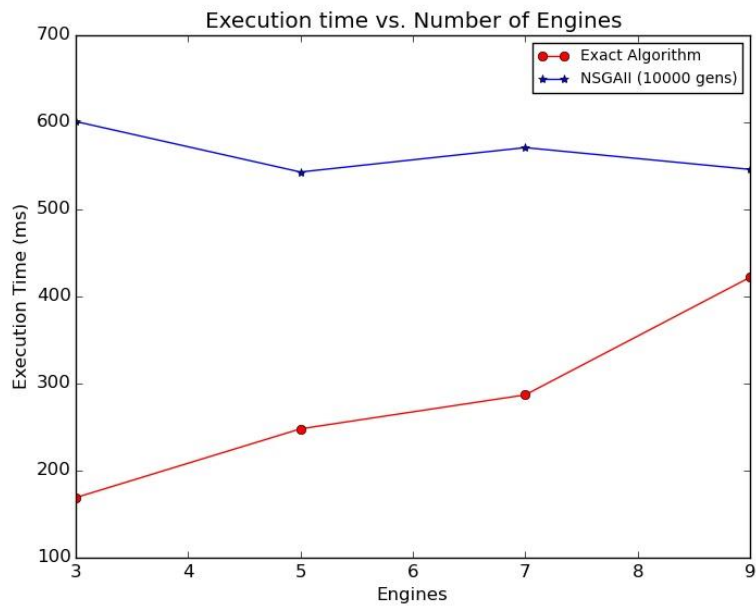


(β)



(δ)

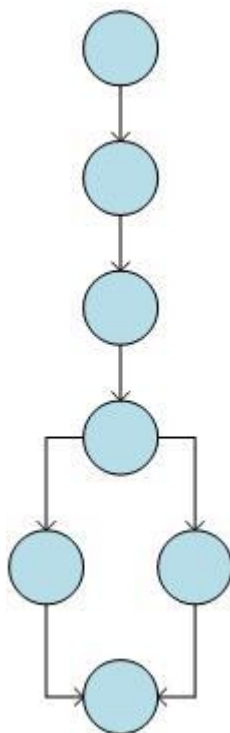
Εικόνα 4-5: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-4.



Εικόνα 4-6: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-4.

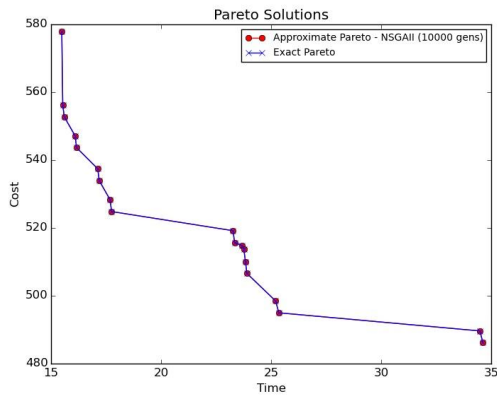
4.1.3. ΡΟΗ ΕΡΓΑΣΙΩΝ 7 ΚΟΜΒΩΝ ΠΟΥ ΥΠΟΣΤΗΡΙΖΕΙ ΠΑΡΑΛΛΗΛΗ ΕΚΤΕΛΕΣΗ ΔΥΟ ΚΟΜΒΩΝ

Αυτή η ροή εργασιών είναι γραμμική, αλλά κάποιος κόμβος διακλαδίζεται σε δύο κόμβους – παιδιά, οι οποίοι έχουν τη δυνατότητα να εκτελεστούν παράλληλα. Στη συνέχεια, η εκτέλεση συνεχίζεται αφού ολοκληρωθούν και οι αφού έχουν κοινό παιδί τον επόμενο κόμβο. Μια απεικόνιση της ροής φαίνεται στην Εικόνα 4-7.

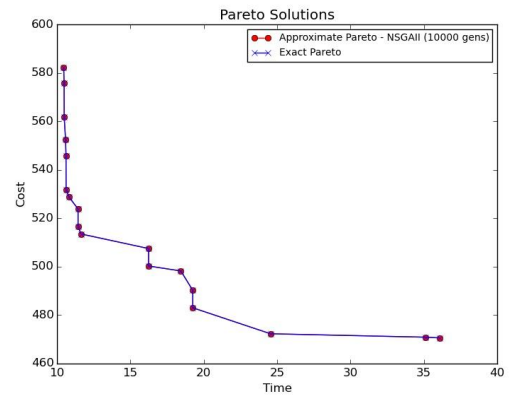


Εικόνα 4-7: Γραμμική Ροή εργασιών με κόμβους με δυνατότητα ταυτόχρονης εκτέλεσης που χρησιμοποιείται στη σειρά των πειράματων

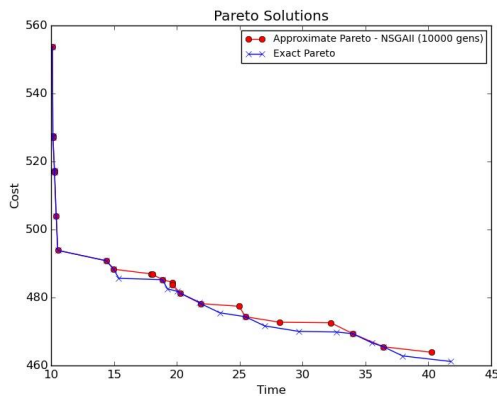
Ακολούθως, στην Εικόνα 4-8 φαίνονται οι λύσεις που προέκυψαν από την εκτέλεση του συστήματος, ενώ στην Εικόνα 4-9 φαίνεται το διάγραμμα των χρόνων εκτέλεσης.



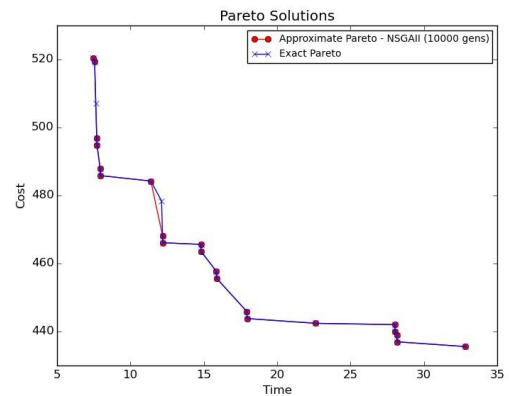
(α)



(γ)

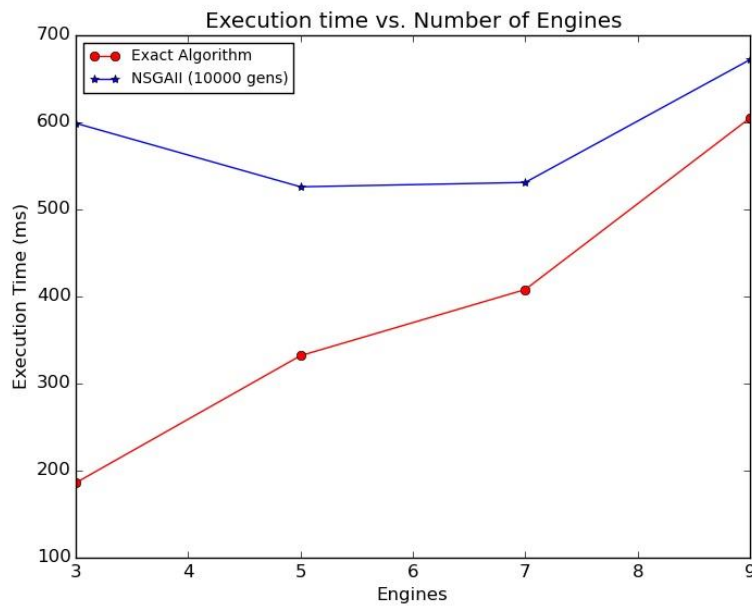


(β)



(δ)

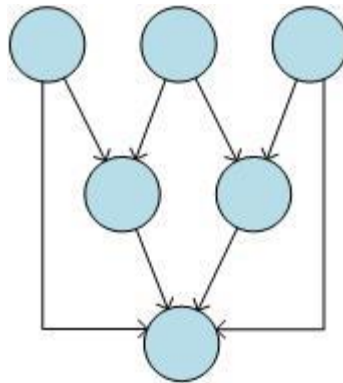
Εικόνα 4-8: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-7.



Εικόνα 4-9: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-7.

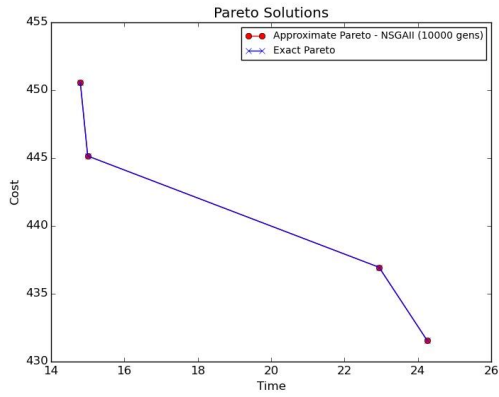
4.1.4. ΡΟΗ ΕΡΓΑΣΙΩΝ 6 ΚΟΜΒΩΝ ΜΕ ΠΟΛΛΕΣ ΕΞΑΡΤΗΣΕΙΣ ΑΝΑΜΕΣΑ ΤΟΥΣ

Αυτή η ροή είναι η πιο πολύπλοκη από τις προηγούμενες, που χρησιμοποιήθηκαν. Υπάρχει ροή δεδομένων ανάμεσα σε πολλούς κόμβους, με τον τελευταίο κόμβο συγκεκριμένα να περιμένει δεδομένα σχεδόν από όλους τους προηγούμενους. Η ροή παρουσιάζεται στην Εικόνα 4-10.

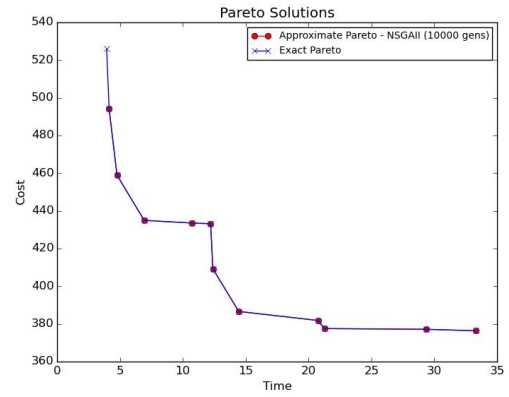


Εικόνα 4-10: Πολύπλοκη ροή εργασιών από θέμα ροής δεδομένων που χρησιμοποιήθηκε στα πειράματα

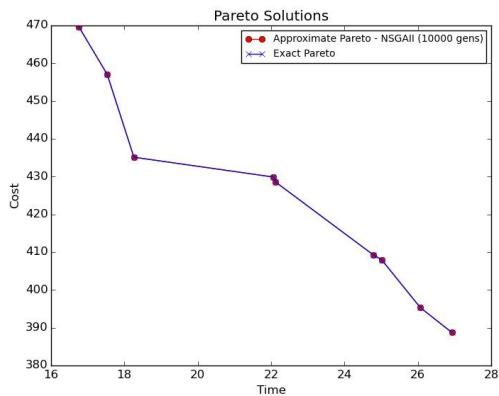
Στην Εικόνα 4-11 παρουσιάζονται οι καμπύλες των λύσεων που προέκυψαν από τον ακριβή και τον προσεγγιστικό αλγόριθμο, ενώ στην 4-12 φαίνεται η εξέλιξη του χρόνου εκτέλεσης των δύο αλγορίθμων σε συνάρτηση με το πλήθος των διαθέσιμων διαφορετικών μηχανών. Σε αυτό το πείραμα αρχίζει να γίνεται εμφανές, παρατηρώντας τους χρόνους εκτέλεσης συγκριτικά με τα προηγούμενα πειράματα και τα αποτελέσματα των αλγορίθμων, ότι όσο αυξάνεται η πολυπλοκότητα του προβλήματος (δηλαδή περισσότερες ροές δεδομένων και περισσότερες μηχανές), αξίζει η χρήση του ευριστικού αλγορίθμου.



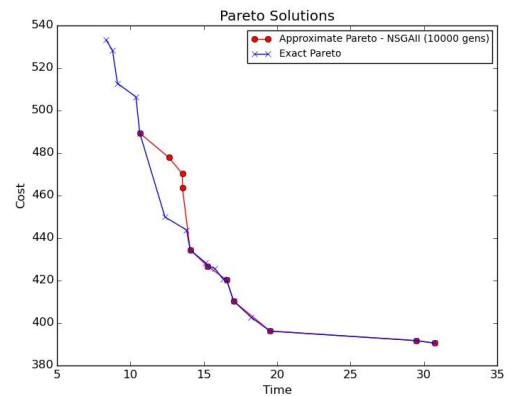
(α)



(γ)

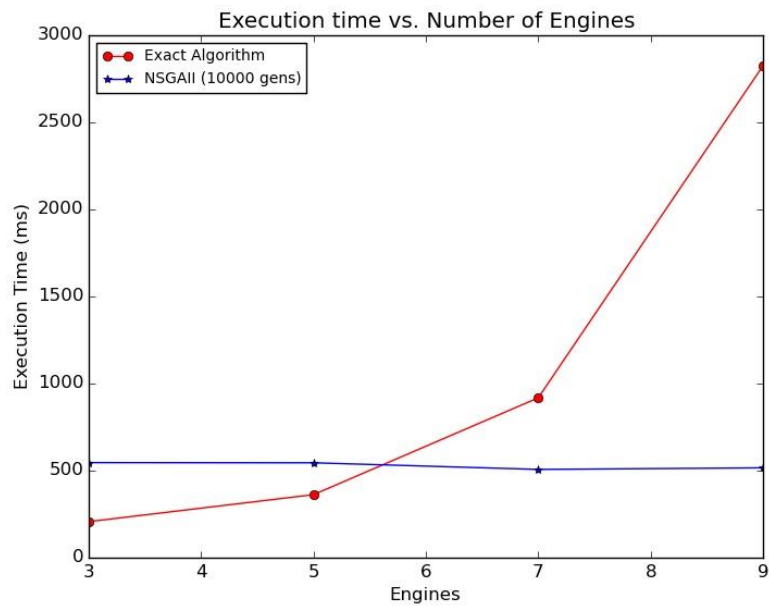


(β)



(δ)

Εικόνα 4-11: Οι καμπύλες λύσεων των δύο αλγορίθμων για (α) 3, (β) 5, (γ) 7 και (δ) 9 μηχανές για τη ροή εργασιών της Εικόνας 4-10.



Εικόνα 4-12: Διάγραμμα χρόνων εκτέλεσης των δύο αλγορίθμων σε σχέση με το πλήθος των μηχανών για το της Εικόνας 4-10.

4.1.5. ΣΧΟΛΙΑΣΜΟΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Αρχικά, κρίνεται σκόπιμο να σχολιαστεί η απόδοση του ευριστικού αλγορίθμου σε σχέση με αυτήν του αλγορίθμου που υπολογίζει τη βέλτιστη λύση. Γίνεται φανερό ότι σε περιπτώσεις που έχουμε λίγες διαθέσιμες μηχανές, ο προσεγγιστικός αλγόριθμος βρίσκει την ίδια λύση με τον βέλτιστο για μικρές ροές εργασιών. Όσο αυξάνεται το πλήθος των μηχανών, παρατηρείται ότι αρχίζει να φαίνεται μια μικρή απόκλιση από το πραγματικό βέλτιστο σύνολο κατά Pareto. Επομένως, σε τέτοιες περιπτώσεις γίνεται φανερό ότι και με την προσεγγιστική διαδικασία λαμβάνονται ικανοποιητικά αποτελέσματα ως λύση στο πρόβλημα βελτιστοποίησης των ροών εργασιών.

Επιπλέον, μία βασική παράμετρος ως προς την αξιολόγηση τους είναι ο χρόνος εκτέλεσης. Παρατηρώντας όλα τα διαγράμματα των χρόνων εκτέλεσης, γίνεται φανερό ότι στην περίπτωση του βέλτιστου αλγορίθμου σε κάθε περίπτωση αυξάνεται ο χρόνος εκτέλεσης όσο αυξάνεται ο αριθμός των μηχανών που πρόκειται να εξεταστούν. Αντιθέτως, σε όλες σχεδόν τις περιπτώσεις ο χρόνος εκτέλεσης του ευριστικού αλγορίθμου επηρεάζεται ελάχιστα από τον αριθμό των μηχανών, αφού σε όλα τα πειράματα κυμαίνεται στο εύρος 550 – 650ms. Επίσης, ο χρόνος του εμφανίζει μικρή μεταβολή σε διαφορετικές μορφές ροών εργασιών κάτι που δεν συμβαίνει στον βέλτιστο αλγόριθμο, αφού, για παράδειγμα, στη σύνθετη ροή εργασιών έχουμε πολύ μεγαλύτερους χρόνους εκτέλεσης σε σχέση με τη γραμμική.

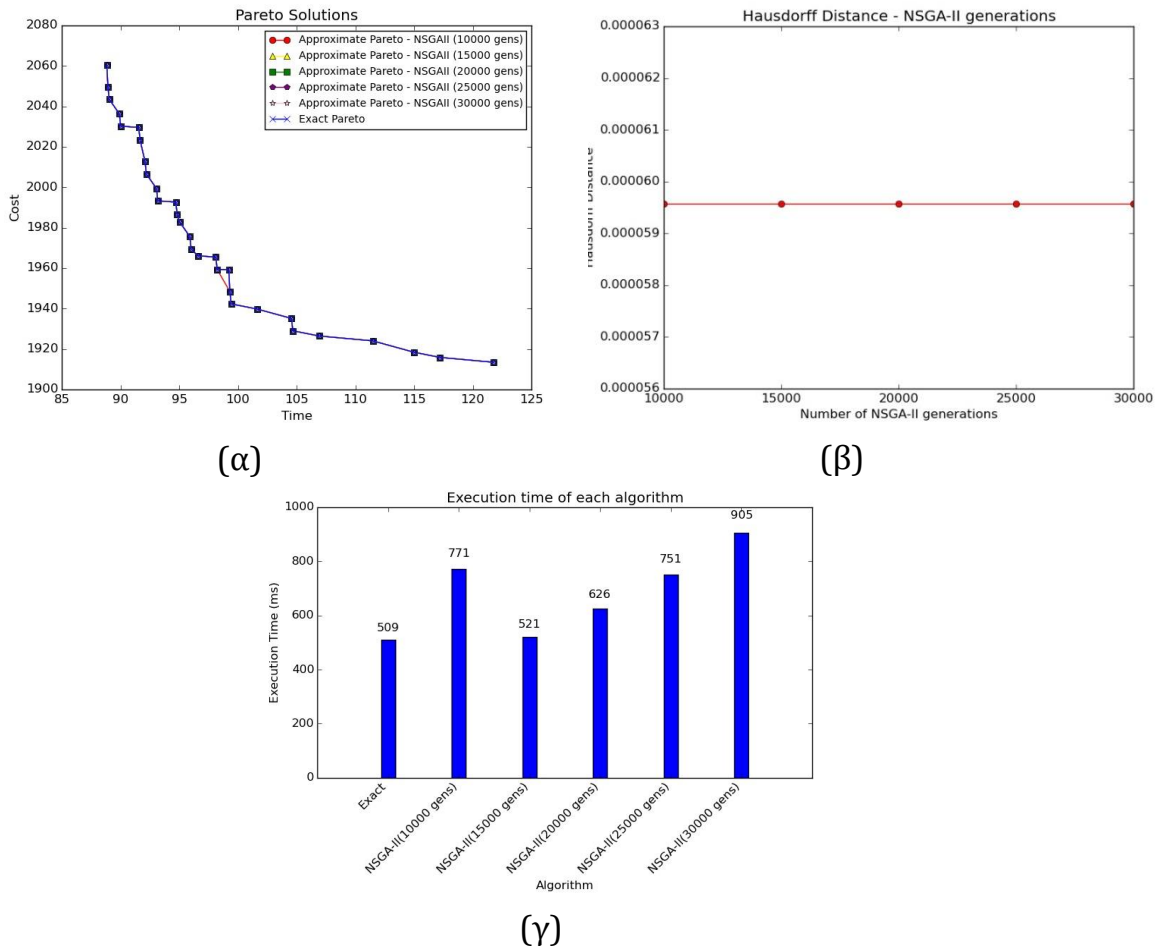
4.2. ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ ΣΕ ΓΡΑΜΜΙΚΕΣ ΡΟΕΣ ΕΡΓΑΣΙΩΝ ΔΙΑΦΟΡΩΝ ΜΕΓΕΘΩΝ

Σε αυτήν την ενότητα γίνεται σύγκριση των δύο αλγορίθμων πάνω σε γραμμικές ροές διαφόρων μεγεθών. Συγκεκριμένα, χρησιμοποιήθηκαν γράφοι 25, 50, 100 και 500 κόμβων, ώστε να παρατηρηθεί πως επηρεάζει η αύξηση του μεγέθους του γράφου τα αποτελέσματα. Για αυτήν την σειρά πειραμάτων θεωρήθηκε ότι υπάρχουν 2, 3 και έπειτα 5 διαφορετικές μηχανές για την κάθε περίπτωση. Επίσης, αξιολογείται και το πως επηρεάζει το πλήθος των γενέων που θα τρέξει ο NSGA-II τα αποτελέσματα και έγινε μελέτη από 10 έως 30 χιλιάδες γενεές αυξάνοντας κάθε φορά κατά 5000. Στην ενότητα αυτή, η απόδοση του ευριστικού αλγορίθμου θα αξιολογηθεί τόσο οπτικά από τις λύσεις που θα προκύψουν όσο και με τη μετρική Hausdorff.

4.2.1. ΡΟΗ ΜΕ 25 ΕΡΓΑΣΙΕΣ

Αυτή είναι η μικρότερη από τις ροές που εξετάζονται στο πείραμα. Στη συνέχεια, παρουσιάζονται τα αποτελέσματα για κάθε ένα από τα πειράματα ομαδοποιημένα με βάση το πλήθος των διαθέσιμων μηχανών και ο σχετικός σχολιασμός τους.

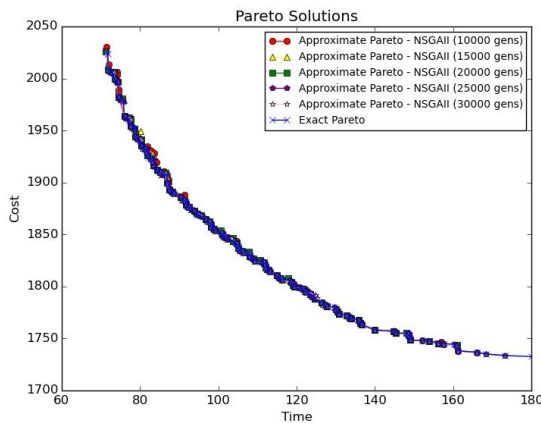
- 2 μηχανές



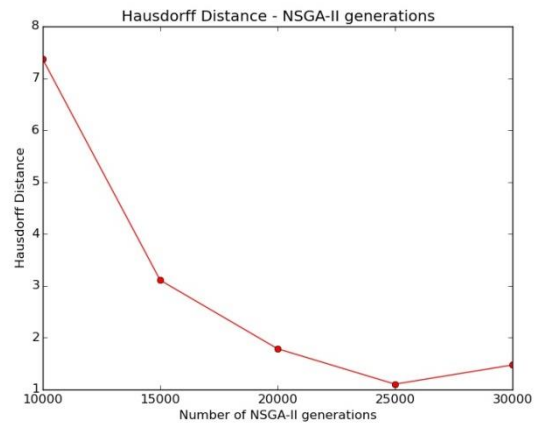
Εικόνα 4-13: Αποτελέσματα για γράφο 25 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Παρατηρώντας τα αποτελέσματα, γίνεται φανερό ότι σε όλες τις περιπτώσεις ο προσεγγιστικός αλγόριθμος δίνει το ίδιο ή σχεδόν το ίδιο σύνολο λύσεων με το βέλτιστο αλγόριθμο. Επειδή και ο χρόνος εκτέλεσης του βέλτιστου αλγορίθμου είναι εξίσου καλός με του προσεγγιστικού σε τέτοια περίπτωση θα ήταν προτιμότερη η χρήση του πρώτου, ώστε να διασφαλιστεί ότι θα παραχθούν οι βέλτιστες λύσεις.

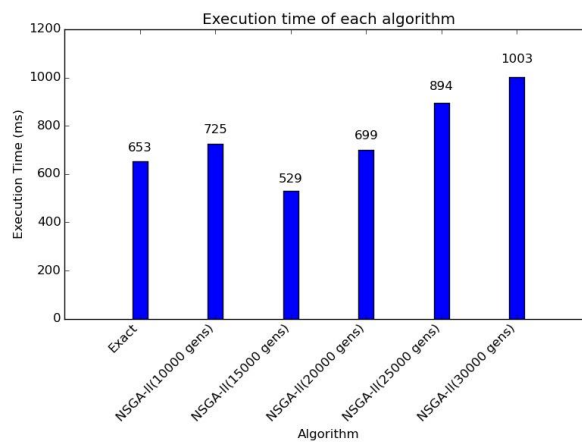
- 3 μηχανές



(α)



(β)

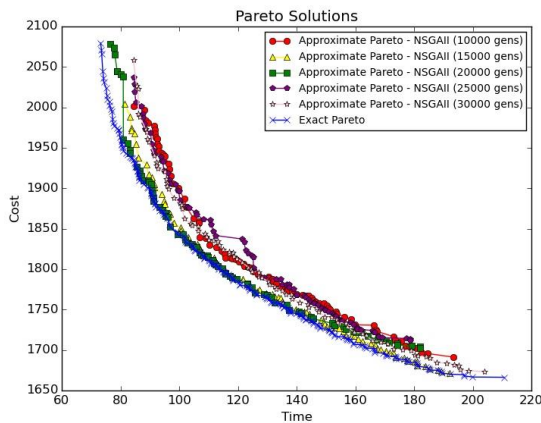


(γ)

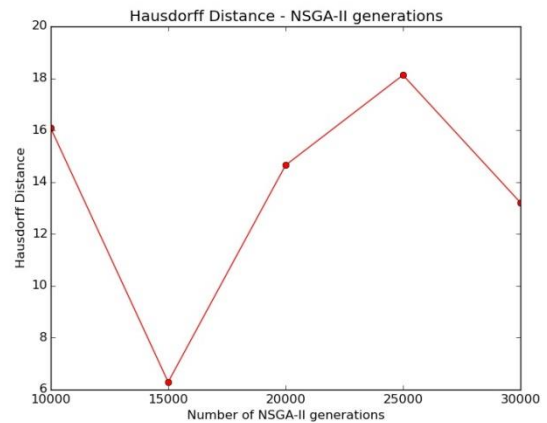
Εικόνα 4-14: Αποτελέσματα για γράφο 25 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Από θέμα απόδοσης παρατηρείται και πάλι σύγκλιση του ευριστικού στον βέλτιστο αλγόριθμο. Συγκρίνοντας τις επιδόσεις, φαίνεται ότι μέσω του ευριστικού αλγορίθμου μπορεί να ληφθεί λύση σε μικρότερο χρονικό διάστημα. Ωστόσο, επειδή η χρονική διαφορά είναι μικρή, αν χρειάζεται να διασφαλιστεί ότι θα προκύψουν οι βέλτιστες λύσεις θα προτιμηθεί ο βέλτιστος αλγόριθμος.

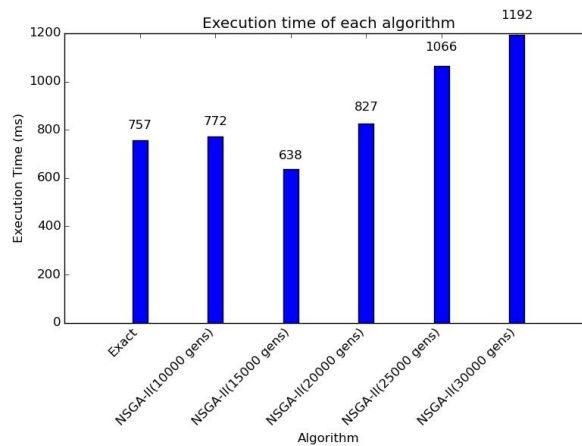
- 5 μηχανές



(α)



(β)



(γ)

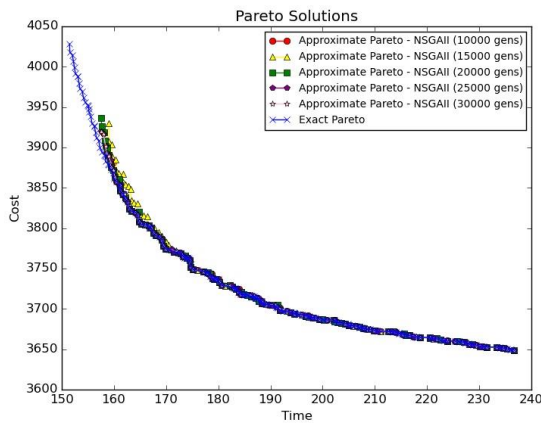
Εικόνα 4-15: Αποτελέσματα για γράφο 25 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) απόσταση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Το παραπάνω είναι το πρώτο από τα πειράματα στο οποίο έχουμε εμφανή διαφορά ανάμεσα στα αποτελέσματα των δύο αλγορίθμων. Όπως φαίνεται τόσο από το διάγραμμα της μετρικής όσο και από το σύνολο των λύσεων, όσο αυξάνουμε το πλήθος των γενεών που θα τρέξει ο προσεγγιστικός αλγόριθμος γίνεται πιο σίγουρο ότι θα πλησιάσει η λύση του τη βέλτιστη. Ωστόσο, το παραπάνω δεν εξασφαλίζεται, διότι το αποτέλεσμα του αλγορίθμου εξαρτάται άμεσα από τις πιθανές λύσεις τις οποίες θα συμπεριλάβει στην πρώτη γενιά. Αξίζει επίσης να σημειωθεί πως, όπως φαίνεται συγκρίνοντας το διάγραμμα των λύσεων με αυτό της απόστασης Hausdorff, σύμφωνα με την απόσταση καλύτερο αποτέλεσμα λαμβάνουμε στις 30000 γενεές, αλλά παρατηρώντας το διάγραμμα των λύσεων φαίνεται ότι στις 15000 ή στις 20000 γενεές το σύνολο λύσεων πλησιάζει περισσότερο το πραγματικό σύνολο. Λαμβάνοντας υπόψιν και τους χρόνους εκτέλεσης, εάν επιλεγόταν ο προσεγγιστικός αλγόριθμος θα

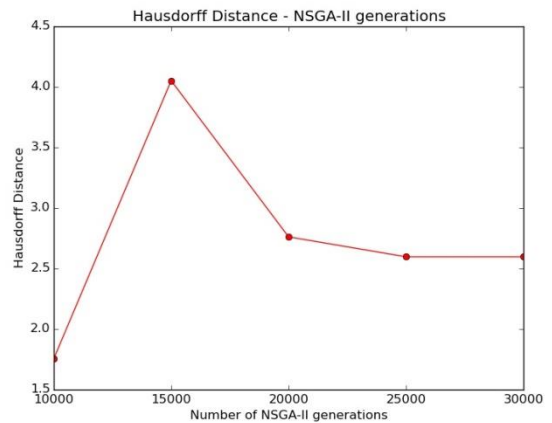
είχε νόημα να εκτελεστεί για 15000 γενεές, καθώς και θα προέκυπτε ικανοποιητικό αποτέλεσμα και σε λιγότερο χρόνο από το βέλτιστο αλγόριθμο.

4.2.2. ΡΟΗ ΜΕ 50 ΕΡΓΑΣΙΕΣ

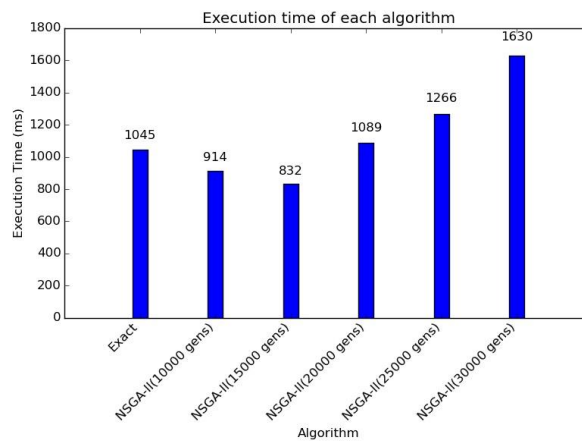
- 2 μηχανές



(α)



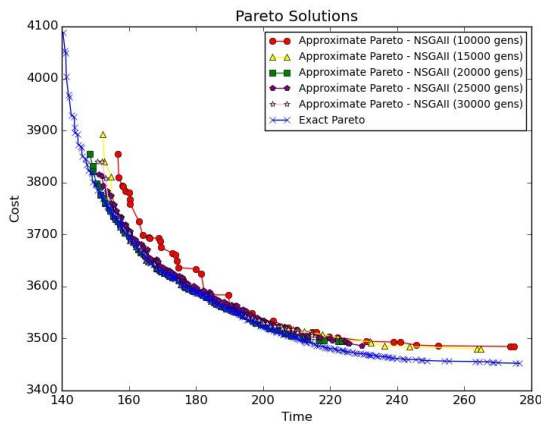
(β)



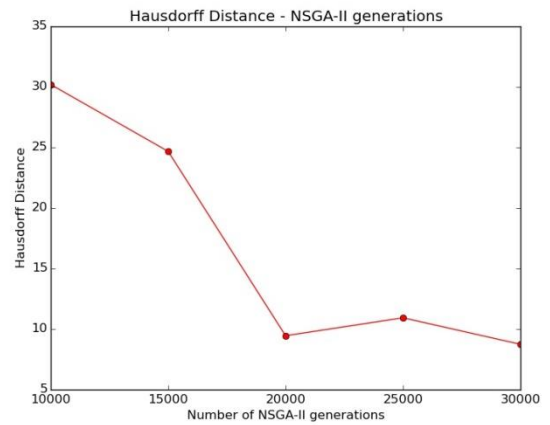
(γ)

Εικόνα 4-16: Αποτελέσματα για γράφο 50 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

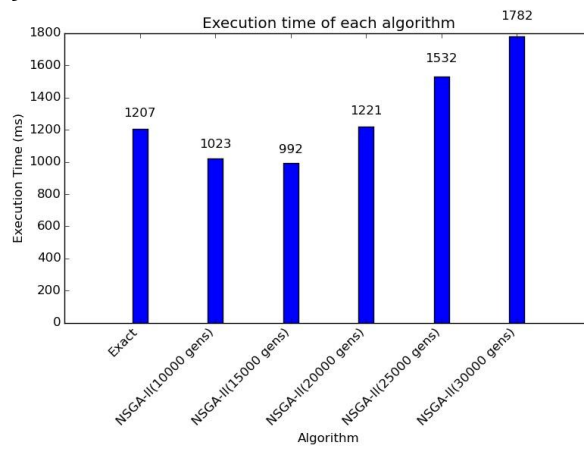
- 3 μηχανές



(α)



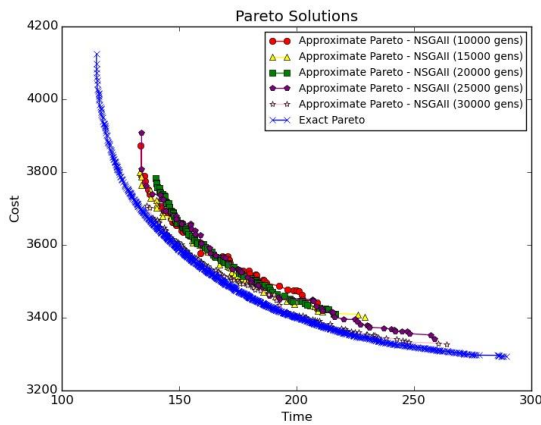
(β)



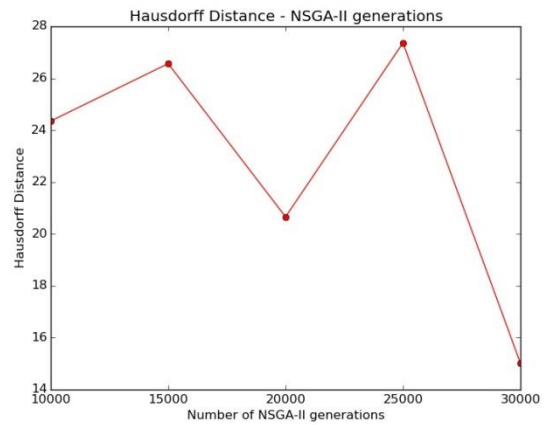
(γ)

Εικόνα 4-17: Αποτελέσματα για γράφο 50 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

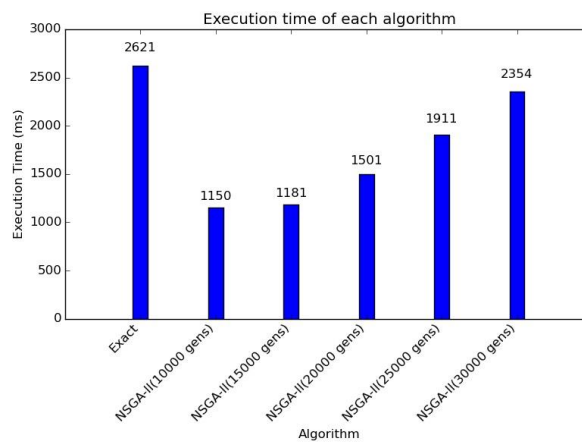
- 5 μηχανές



(α)



(β)



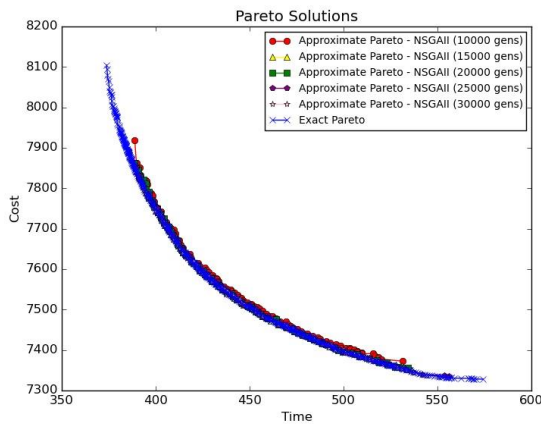
(γ)

Εικόνα 4-18: Αποτελέσματα για γράφο 50 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

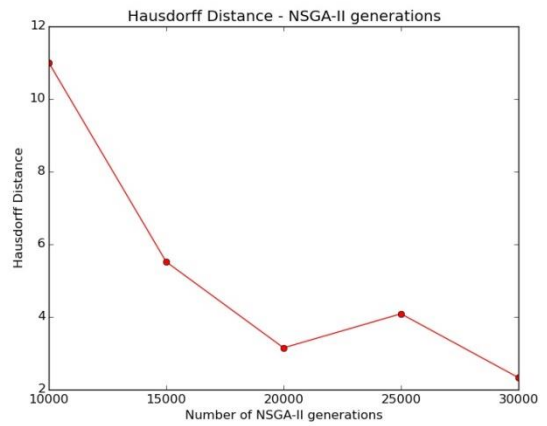
Παρατηρώντας όλα τα παραπάνω, φαίνεται ότι με την αύξηση του αριθμού των κόμβων ο χρόνος εκτέλεσης του βέλτιστου αλγορίθμου αυξάνεται. Κρίνοντας τόσο από τους χρόνους εκτέλεσης του προσεγγιστικού αλγορίθμου όσο και από τα αποτελέσματα απόδοσης θα μπορούσε να χρησιμοποιηθεί με ικανοποιητική προσέγγιση και καλύτερο χρόνο ο ευριστικός αλγόριθμος με 15000 γενεές επανάληψης. Ωστόσο, ακόμα οι χρόνοι του βέλτιστου αλγορίθμου είναι σε επιτρεπτά πλαίσια, οπότε η χρήση προσέγγισης δεν είναι ακόμα απαραίτητη.

4.2.3. ΠΟΗ ΜΕ 100 ΕΡΓΑΣΙΕΣ

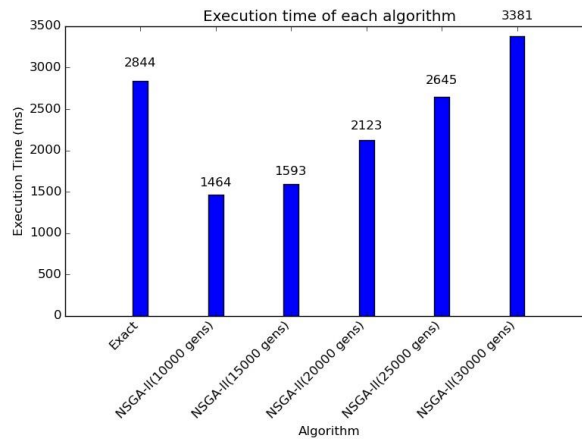
- 2 μηχανές



(α)



(β)

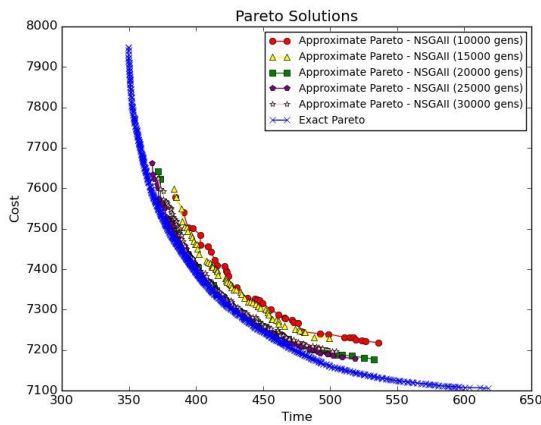


(γ)

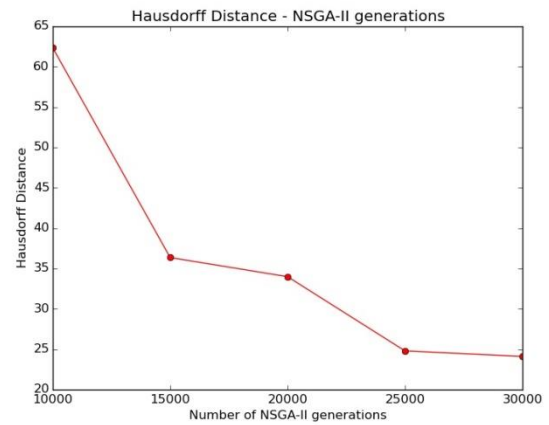
Εικόνα 4-19: Αποτελέσματα για γράφο 100 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Με χρήση όλων των δεδομένων στην περίπτωση αυτή, θα ήταν ικανοποιητική η προσέγγιση του ευριστικού αλγορίθμου με 15000 γενεές. Φυσικά, ο βέλτιστος αλγόριθμος εκτελείται ακόμα σε χρόνους που επιτρέπουν τη χρήση του.

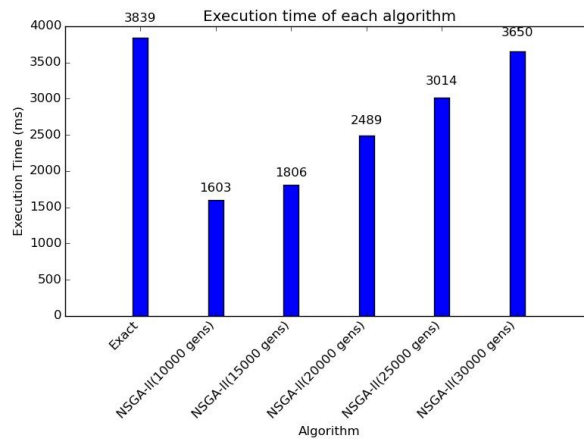
- 3 μηχανές



(α)



(β)

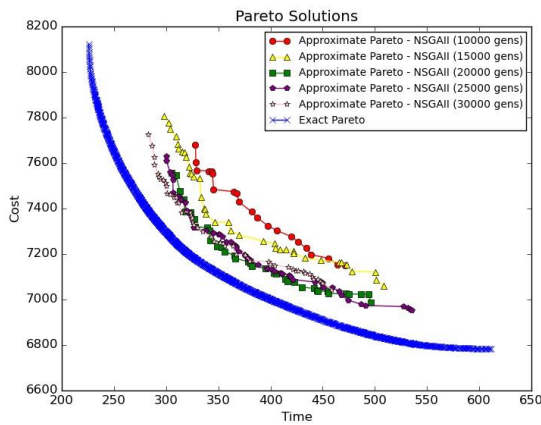


(γ)

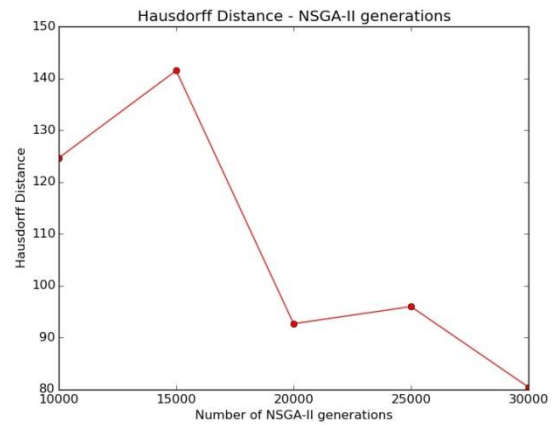
Εικόνα 4-20: Αποτελέσματα για γράφο 100 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Στην περίπτωση αυτή ικανοποιητική προσέγγιση λαμβάνεται από τις 20000 γενιές και πάνω. Ανάλογα με το αν είναι επιθυμητή αισθητή χρονική βελτίωση σε σχέση με το βέλτιστο αλγόριθμο ή όχι, μία καλή επιλογή θα ήταν ο NSGA-II να τρέξει για 20 ή 25 χιλιάδες γενιές, αντίστοιχα.

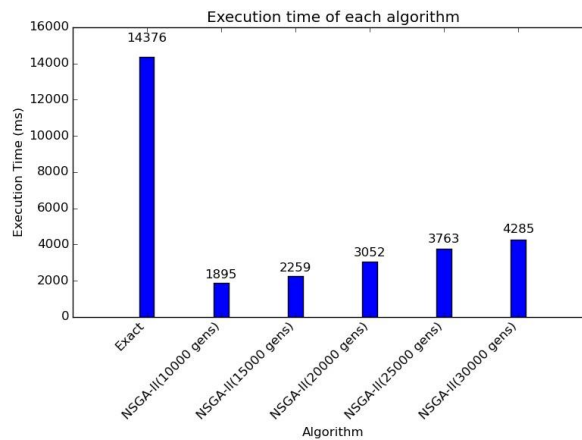
- 5 μηχανές



(α)



(β)



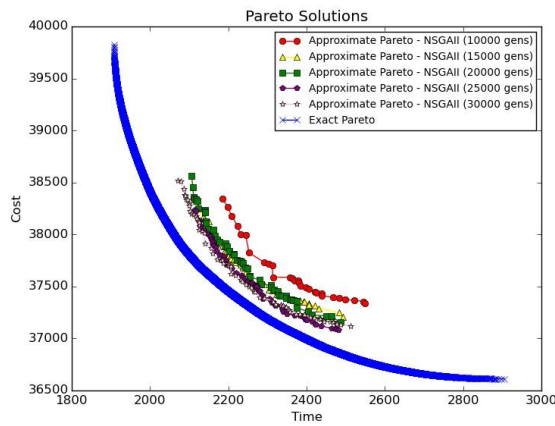
(γ)

Εικόνα 4-21: Αποτελέσματα για γράφο 100 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) αποστάση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

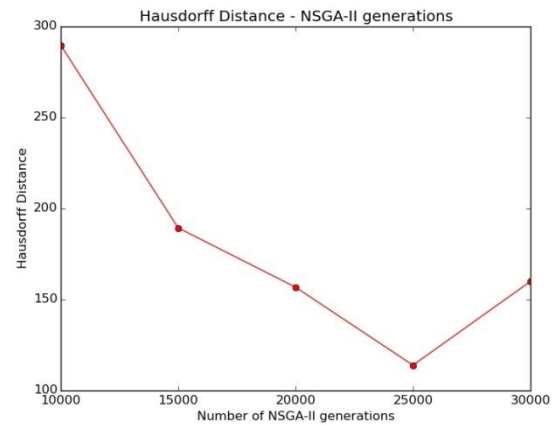
Η ταυτόχρονη αύξηση και του αριθμού των κόμβων και των διαθέσιμων μηχανών οδήγησε σε έντονη αύξηση του χρόνου εκτέλεσης του βέλτιστου αλγορίθμου. Εδώ γίνεται πλέον φανερή η ανάγκη χρήσης του προσεγγιστικού αλγορίθμου. Μια αρκετά ικανοποιητική προσέγγιση λαμβάνεται στις 20000 επαναλήψεις του αλγορίθμου, ενώ με βάση τη μετρική παρατηρείται αισθητή μείωση της απόστασης αν ο αλγόριθμος τρέξει για 30000 επαναλήψεις.

4.2.4. ΡΟΗ ΜΕ 500 ΚΟΜΒΟΥΣ

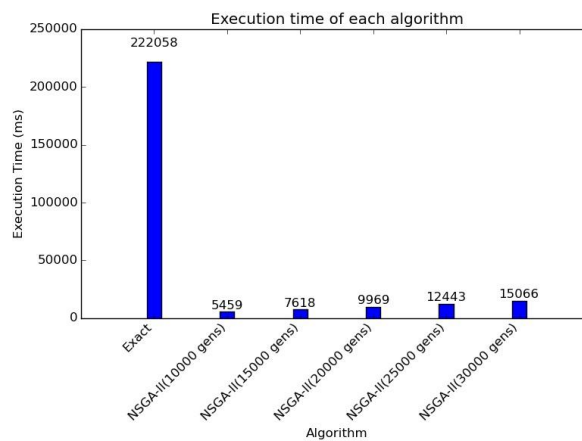
- 2 μηχανές



(α)



(β)

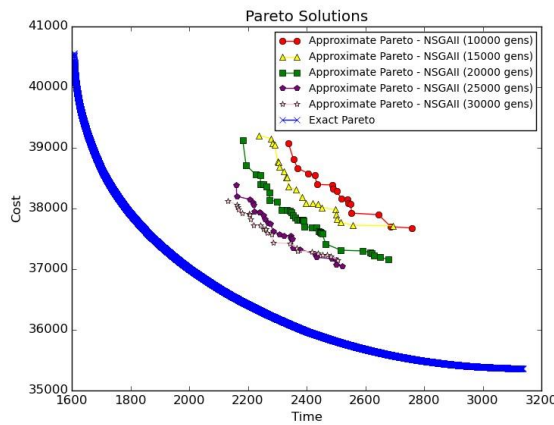


(γ)

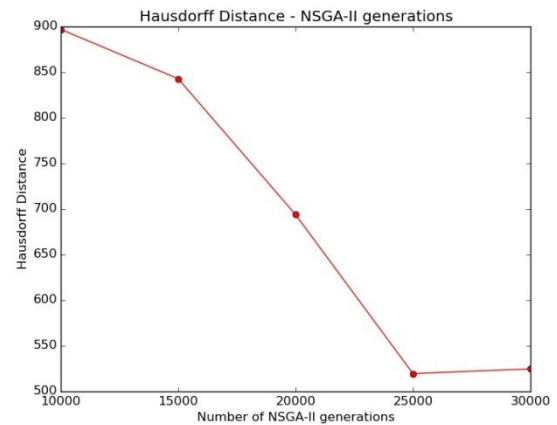
Εικόνα 4-22: Αποτελέσματα για γράφο 500 κόμβων με 2 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) απόσταση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Σε γράφους τέτοιου μεγέθους γίνεται φανερό ότι ο βέλτιστος αλγόριθμος δεν είναι αποδοτικός. Επομένως, είναι αναγκαστικό να χρησιμοποιείται ο προσεγγιστικός αλγόριθμος. Λαμβάνοντας υπόψιν την απόσταση από τη βέλτιστη λύση, μία ικανοποιητική προσέγγιση λαμβάνεται όταν ο αλγόριθμος εκτελεστεί για 25000 επαναλήψεις και σε σημαντικά καλύτερο χρόνο σε σχέση με τον βέλτιστο αλγόριθμο.

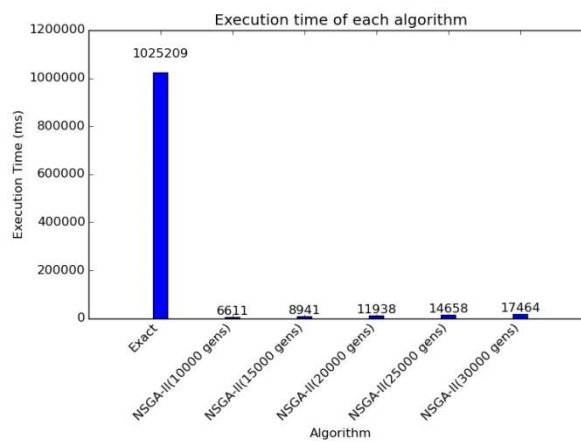
- 3 μηχανές



(α)



(β)

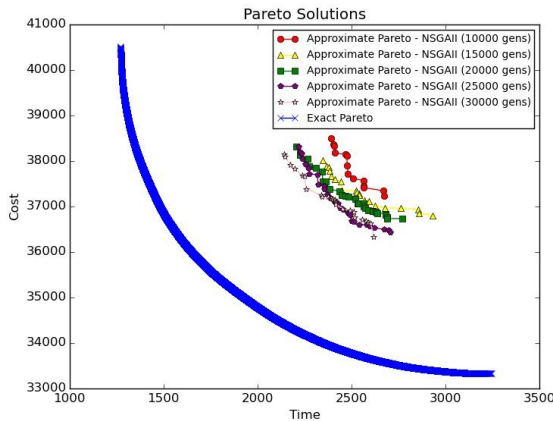


(γ)

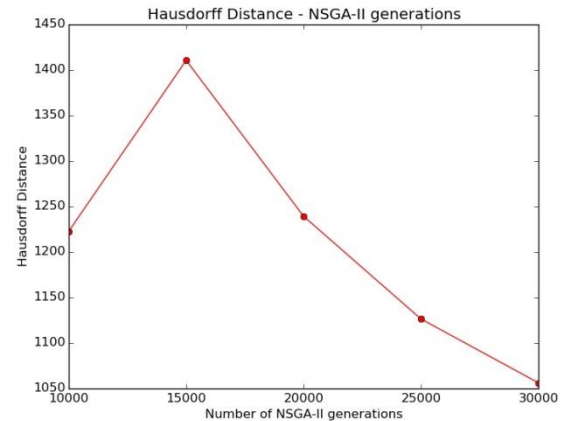
Εικόνα 4-23: Αποτελέσματα για γράφο 500 κόμβων με 3 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) απόσταση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Όπως ήταν αναμενόμενο, ο βέλτιστος αλγόριθμος δεν είναι αποδοτικός σε αυτήν την περίπτωση. Αναφορικά με τις γενιές του προσεγγιστικού αλγορίθμου, από το διάγραμμα της απόστασης Hausdorff, παρατηρείται ότι για 25000 και 30000 γενιές η απόσταση από τη βέλτιστη λύση είναι παραπλήσια. Επόμενως, επιλέγονται οι 25000 γενιές λόγω του συγκριτικά καλύτερου χρόνου εκτέλεσης.

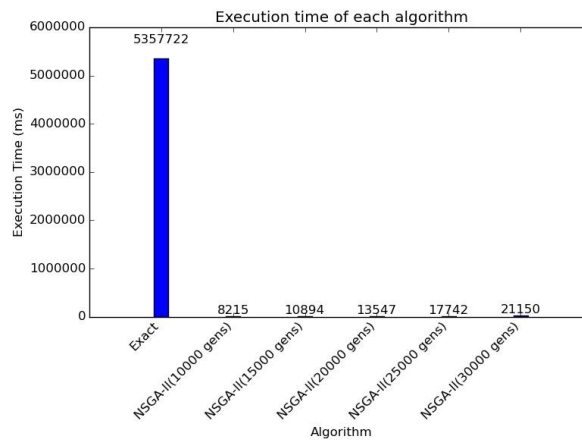
- 5 μηχανές



(α)



(β)



(γ)

Εικόνα 4-24: Αποτελέσματα για γράφο 500 κόμβων με 5 διαθέσιμες μηχανές: (α) καμπύλες λύσεων (β) απόσταση Hausdorff προσεγγιστικού συνόλου λύσεων από ακριβές και (γ) χρόνοι εκτέλεσης αλγορίθμων

Σε αυτήν την περίπτωση θα χρησιμοποιηθεί ο προσεγγιστικός αλγόριθμος με τουλάχιστον 30000 επαναλήψεις. Η απόσταση από τη πραγματική καμπύλη στην περίπτωση αυτή είναι αρκετά μεγάλη γεγονός που υποδεικνύει ότι χρειάζονται παραπάνω επαναλήψεις για πιο ακριβές αποτέλεσμα.

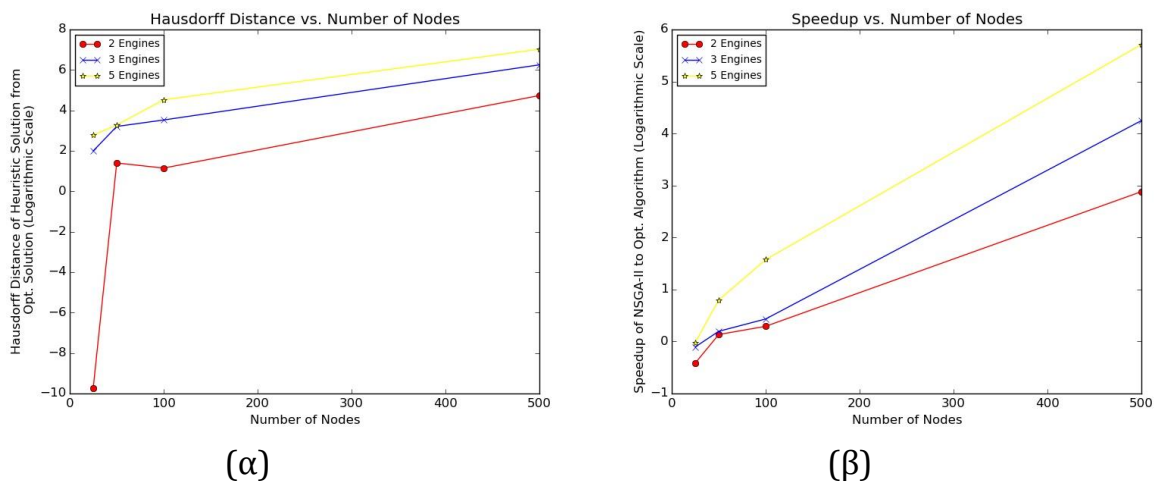
4.2.5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως αναφέρθηκε στη προηγούμενη σειρά πειράματων ο ευριστικός αλγόριθμος δεν επηρεάζεται από τη μορφή των γράφων για την ποιότητα των αποτελεσμάτων που δίνει, αλλά εξαρτάται από το πλήθος των γενεών που θα εκτελεστεί. Επαγωγικά μπορούμε να εξάγουμε κάποια συμπεράσματα, όπως το πόσες γενιές εκτέλεσης του ευριστικού αλγορίθμου χρειάζονται ανάλογα με την τάξη μεγέθους της ροής εργασιών για να προκύψει ένα ικανοποιητικό αποτέλεσμα. Με βάση τα όσα αναλύθηκαν παραπάνω, στον Πίνακα 4-1 υπάρχει μια αντίστοιχη πλήθους κόμβων ροής εργασιών με πλήθος επαναλήψεων του ευριστικού αλγορίθμου.

Πλήθος Κόμβων	Γενιές NSGA-II
25	10000
50	15000
100	20000
500	25000

Πίνακας 4-1: Αριθμός επαναλήψεων NSGA-II για διάφορα πλήθη κόμβων ροής εργασιών ώστε να ληφθεί ένα ικανοποιητικό αποτέλεσμα.

Για το πλήθος των γενεών που υπάρχουν στον Πίνακα 4-1, δίνονται στη συνέχεια το διάγραμμα επιτάχυνσης του προσεγγιστικού αλγορίθμου ως προς το βέλτιστο, καθώς και το διάγραμμα της απόστασης Hausdorff ως συνάρτηση των κόμβων του γράφου για όλα τα διαφορετικά πλήθη μηχανών που εξετάστηκαν.



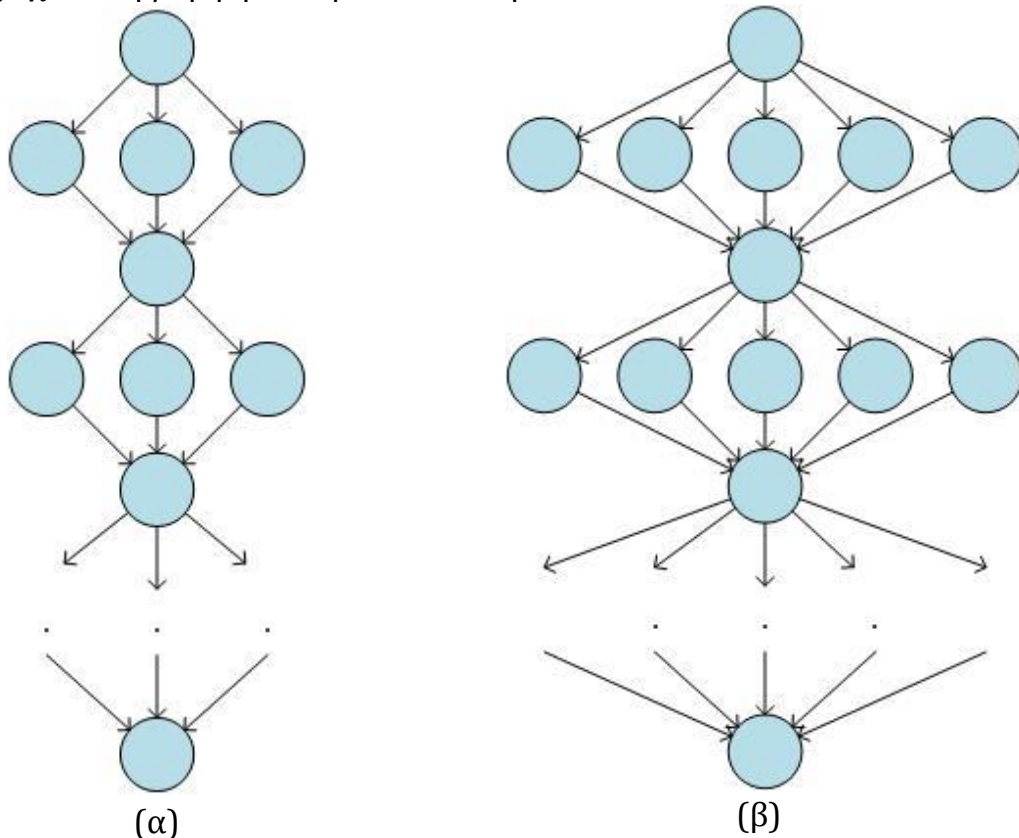
Εικόνα 4-25: Διάγραμμα Επιτάχυνσης του NSGA-II ως προς τον βέλτιστο και η απόσταση της λύσης του από τη βέλτιστη ως συνάρτηση του πλήθους των κόμβων του γραμμικού γράφου

Όπως φαίνεται από την Εικόνα 4-25, η αύξηση του αριθμού των κόμβων του γράφου οδηγεί σε αύξηση της απόστασης των δύο λύσεων.

Ωστόσο, σε μεγάλους γράφους παρατηρούμε ότι η επιτάχυνση του NSGA-II σε σχέση με το βέλτιστο αλγόριθμο είναι τόσο μεγάλη που αξίζει να μειωθεί η ακρίβεια του αποτελέσματος. Αντίθετα σε μικρούς γράφους που παρουσιάζεται μικρή επιτάχυνση, είναι προτιμότερη η χρήση του βέλτιστου αλγορίθμου. Στην περίπτωση που αυξάνεται το πλήθος των διαθέσιμων μηχανών, παρατηρείται μικρή διαφορά στην απόδοση του ευριστικού αλγορίθμου, αλλά σημαντική βελτιώση στην επιτάχυνση. Αυτό οφείλεται στο γεγονός ότι ο χρόνος εκτέλεσης του βέλτιστου αλγορίθμου επηρεάζεται σημαντικά από την αύξηση των μηχανών, ενώ του προσεγγιστικού σε πολύ μικρότερο βαθμό.

4.3. ΜΕΛΕΤΗ ΣΕ ΓΡΑΦΟΥΣ ΜΕ ΒΑΘΜΟΣ ΠΑΡΑΛΛΗΛΙΣΜΟΥ ΜΕΓΑΛΥΤΕΡΟ ΤΟΥ 1

Σε αυτήν την ενότητα γίνεται μελέτη σε λίγο πιο περίπλοκες ροές εργασιών, όπου έχουν κόμβους οι οποίοι μπορούν να εκτελεστούν παράλληλα. Μελετώνται ροές εργασιών που έχουν βαθμό παραλληλισμού 3 και 5. Επίσης, έχουν θεωρηθεί δεδομένα τα συμπεράσματα της προηγούμενης ενότητας και για αυτό ο ευριστικός αλγόριθμος τρέχει τόσες γενιές όσες αναφέρονται στον Πίνακα 4-1, ανάλογα με το μέγεθος της ροής. Επιπλέον, στα πειράματα αυτά έχει θεωρηθεί ότι υπάρχουν δύο διαθέσιμες διαφορετικές μηχανές. Ανάλογα με τον βαθμό παραλληλισμού οι ροές έχουν τη μορφή που φαίνεται στην Εικόνα 4-26.



Εικόνα 4-26: Γράφος με βαθμό παραλληλίας (α) 3 και (β) 5

Τα αποτελέσματα αυτής της ενότητας θα αφορούν το πως επηρεάζει το μέγεθος του γράφου τους χρόνους εκτέλεσης των αλγορίθμων και την απόσταση της προσεγγιστικής από τη βέλτιστη λύση. Στη συνέχεια, θα διαπιστωθεί και ο τρόπος με τον οποίο επηρεάζει τα παραπάνω η αύξηση του βαθμού παραλληλισμού.

4.3.1. ΜΕΛΕΤΗ ΕΠΙΔΡΑΣΗΣ ΑΡΙΘΜΟΥ ΚΟΜΒΩΝ

Από τα πειράματα που εκτελέστηκαν, παρουσιάζονται στον Πίνακα 4-2 (βαθμός παραλληλισμού 3) και στον Πίνακα 4-3 (βαθμός παραλληλισμού 5) οι χρόνοι εκτέλεσης των δύο αλγορίθμων καθώς και επίσης η απόσταση Hausdorff ανάμεσα στα δύο σύνολα λύσεων.

Πλήθος Κόμβων	Χρόνος Εκτέλεσης Βέλτιστου Αλγόριθμου (ms)	Χρόνος Εκτέλεσης NSGA-II (ms)	Απόσταση Hausdorff
25	598	726	$4 \cdot 10^{-5}$
49	1063	1274	10^{-4}
101	3303	2848	2,77
501	496078	14406	153,04

Πίνακας 4-2: Στοιχεία από την εκτέλεση των αλγορίθμων σε γράφους με βαθμό παραλληλισμού 3

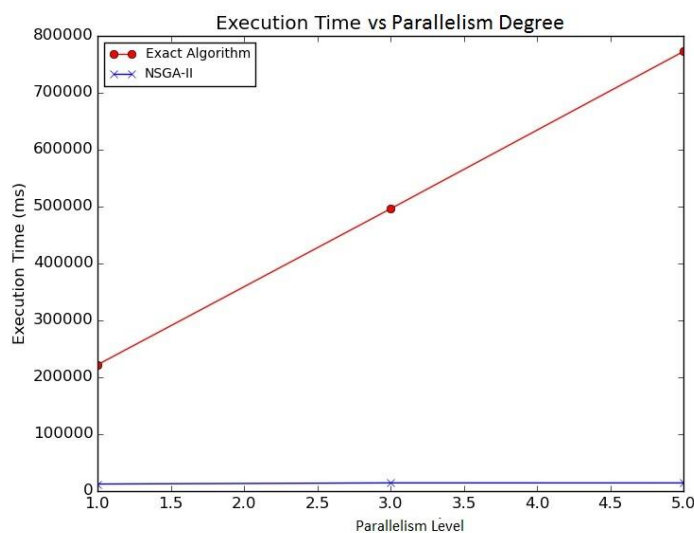
Πλήθος Κόμβων	Χρόνος Εκτέλεσης Βέλτιστου Αλγόριθμου (ms)	Χρόνος Εκτέλεσης NSGA-II (ms)	Απόσταση Hausdorff
25	636	800	$4,68 \cdot 10^{-5}$
49	1169	1265	$8,6 \cdot 10^{-5}$
97	3162	2771	1,68
499	772528	14382	79,94

Πίνακας 4-3: Στοιχεία από την εκτέλεση των αλγορίθμων σε γράφους με βαθμό παραλληλισμού 5

Από τις πληροφορίες που υπάρχουν στους παραπάνω πίνακες, προκύπτει ότι η αύξηση του αριθμού των κόμβων επηρεάζει τόσο τον βέλτιστο αλγόριθμο όσο και τον NSGA-II. Ωστόσο, ο τελευταίος επηρεάζεται εμφανώς σε μικρότερο βαθμό σε σχέση με τον πρώτο. Επίσης, όπως ήταν αναμενόμενο η αύξηση του πλήθους των κόμβων της ροής εργασιών οδηγεί σε αύξηση της απόστασης της προσεγγιστικής από τη βέλτιστη λύση. Σε μεγάλους γράφους, όμως, παρά το γεγονός ότι έχει μεγαλώσει η απόσταση τους, θα προτιμηθεί ο προσεγγιστικός αλγόριθμος λόγω του αρκετά καλύτερου χρόνου εκτέλεσης.

4.3.2. ΜΕΛΕΤΗ ΕΠΙΔΡΑΣΗΣ ΒΑΘΜΟΥ ΠΑΡΑΛΛΗΛΙΣΜΟΥ

Για να μελετηθεί ο τρόπος με τον οποίο επιδράει η αύξηση του βαθμού παραλληλισμού στην επίδοση των αλγορίθμων θα χρησιμοποιηθούν τα δεδομένα που έχουν προκύψει από την εκτέλεση των μεγάλων γράφων (500 κόμβοι) με δύο διαφορετικές μηχανές. Ουσιαστικά έχουμε τον γραμμικό γραφο (βαθμός παραλληλισμού 1) και τους γράφους με βαθμό παραλληλισμού 3 και 5.



Εικόνα 4-27: Διάγραμμα χρόνου εκτέλεσης των αλγορίθμων ως συνάρτηση του βαθμού παραλληλισμού του γράφου

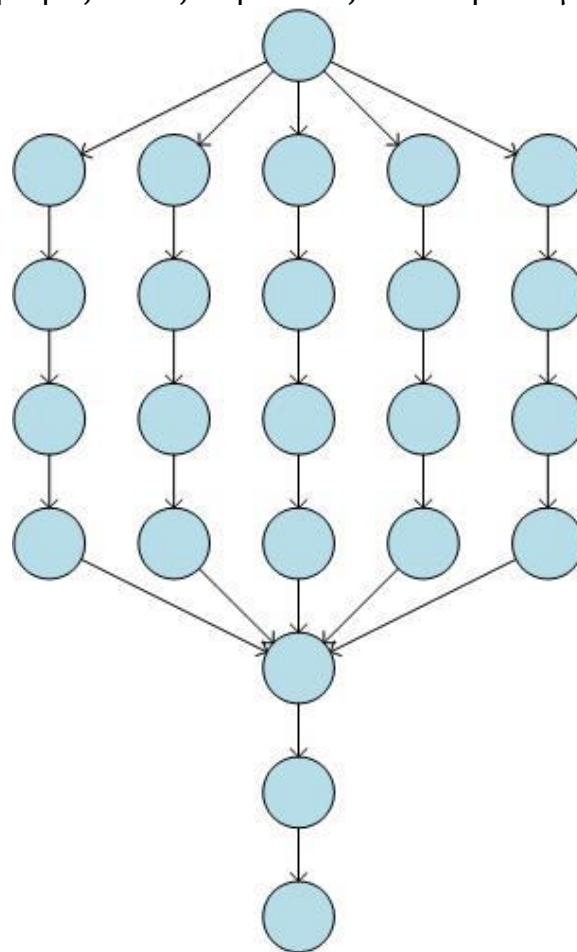
Από την Εικόνα 4-27 γίνεται φανερό ότι η αύξηση του βαθμού παραλληλισμού επηρεάζει γραμμικά το χρόνο εκτέλεσης του βέλτιστου αλγορίθμου, αλλά δεν μεταβάλλει το χρόνο εκτέλεσης του προσεγγιστικού αλγορίθμου.

4.4. ΜΕΛΕΤΗ ΣΕ ΓΡΑΦΟΥΣ ΤΟΥ PEGASUS

Σε αυτή τη σειρά πειραμάτων γίνεται εφαρμογή των αλγορίθμων του συστήματος σε επιστημονικούς γράφους ροών εργασιών που έχουν παραχθεί από τη γεννήτρια Pegasus. Οι γράφοι αυτοί είναι αρκετά πιο περίπλοκοι στη μορφή τους σε σχέση με αυτούς που χρησιμοποιήθηκαν στα προηγούμενα πειράματα. Από τους τύπους των γράφων που παράγονται εξετάστηκαν οι Epigenomics και Montage. Σε αυτή τη σειρά πειραμάτων θεωρήθηκε ότι υπάρχουν δύο διαθέσιμες διαφορετικές μηχανές. Σημειώνεται επίσης ότι η γεννήτρια παράγει ως μέγιστο αριθμό κόμβων στη ροή 1000 αντί για 500, επομένως οι ροές που θα εξεταστούν θα έχουν πλήθος κόμβων της τάξης των 25, 50, 100 και 1000. Στην περίπτωση των 1000 κόμβων (παρατηρώντας τον Πίνακα 4-1) θα χρησιμοποιήσουμε 30000 γενιές για την εκτέλεση του ευριστικού αλγορίθμου.

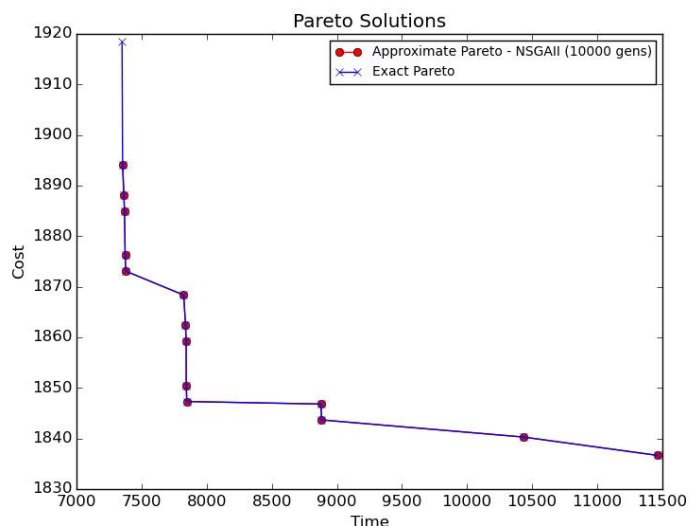
4.4.1. EPIGENOMICS

Αρχικά, παρουσιάζονται τα αποτελέσματα που αφορούν τον γράφο Epigenomics. Ο γράφος αυτός παρουσιάζεται στην επόμενη εικόνα.



Εικόνα 4-28: Γράφος της οικογένειας Epigenomics 24 κόμβων

Το πρώτο πείραμα που εκτελέστηκε ήταν να βρεθούν οι λύσεις για το γράφο της Εικόνας 4-28 όταν υπάρχουν διαθέσιμες δύο μηχανές. Στην Εικόνα 4-29 φαίνονται τα σύνολα των λύσεων που προέκυψαν από την εκτέλεση των δύο αλγορίθμων και στον Πίνακα φαίνονται οι χρόνοι εκτέλεσης των δύο αλγορίθμων και η απόσταση της προσεγγιστικής από τη βέλτιστη λύση.



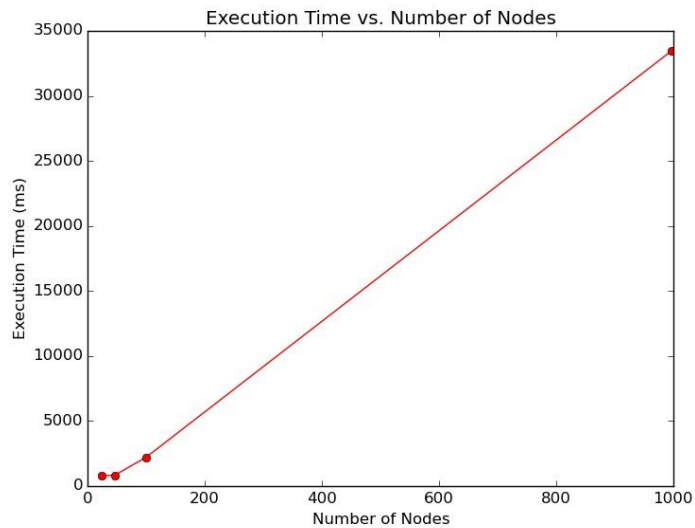
Εικόνα 4-29: Σύνολα λύσεων για ροή εργασιών τύπου *Eripenomics 24* κόμβων με δύο διαθέσιμες μηχανές από τον βέλτιστο και τον προσεγγιστικό αλγόριθμο.

Χρόνος εκτέλεσης βέλτιστου αλγορίθμου	Χρόνος εκτέλεσης NSGA-II (10000 γενιές)	Επιτάχυνση	Απόσταση προσεγ.από βέλτιστη λύση
107517 ms	705 ms	152.506	$4,86 \cdot 10^{-4}$

Πίνακας 4-4: Χρόνοι εκτέλεσης των δύο αλγορίθμων και η απόσταση των λύσεων τους για τη ροή εργασιών *Eripenomics 24* κόμβων με δύο διαθέσιμες μηχανές

Γίνεται φανερό ότι σε προβλήματα όπως αυτό που αυξάνεται η πολυπλοκότητα της μορφής του γράφου ο χρόνος εκτέλεσης του βέλτιστου αλγορίθμου γίνεται απαγορευτικός ακόμα και για γράφους με λίγους κόμβους. Ωστόσο, με τα συμπεράσματα που προέκυψαν από τη μελέτη στους γραμμικούς γράφους, παρατηρείται ότι το πλήθος των γενεών που επιλέχθηκε να τρέξει ο NSGA-II δίνει λύση με σχεδόν μηδενική απόσταση από την πραγματική, με επιτάχυνση 152 φορές πιο γρήγορα. Λόγω των παραπάνω για μεγαλύτερους γράφους της μορφής αυτής επιλέχθηκε να τρέξει μόνο ο προσεγγιστικός αλγόριθμος με πλήθος γενεών ανάλογα με το μέγεθος του γράφου, όπως παρουσιάζεται στον

Πίνακα 4-1. Ο βέλτιστος αλγόριθμος δεν εκτελείται διότι η αύξηση των κόμβων παρατηρήθηκε ότι οδηγεί σε σημαντική αύξηση του χρόνου εκτέλεσης του και επόμενως η εκτέλεση θα διαρκέσει για υπερβολικά μεγάλο χρονικό διάστημα. Στην Εικόνα 4-30 δίνεται το διάγραμμα των χρόνων εκτέλεσης του προσεγγιστικού αλγορίθμου ανάλογα με το πλήθος των κόμβων της ροής εργασιών στην οποία επιχειρείται η εύρεση λύσης.

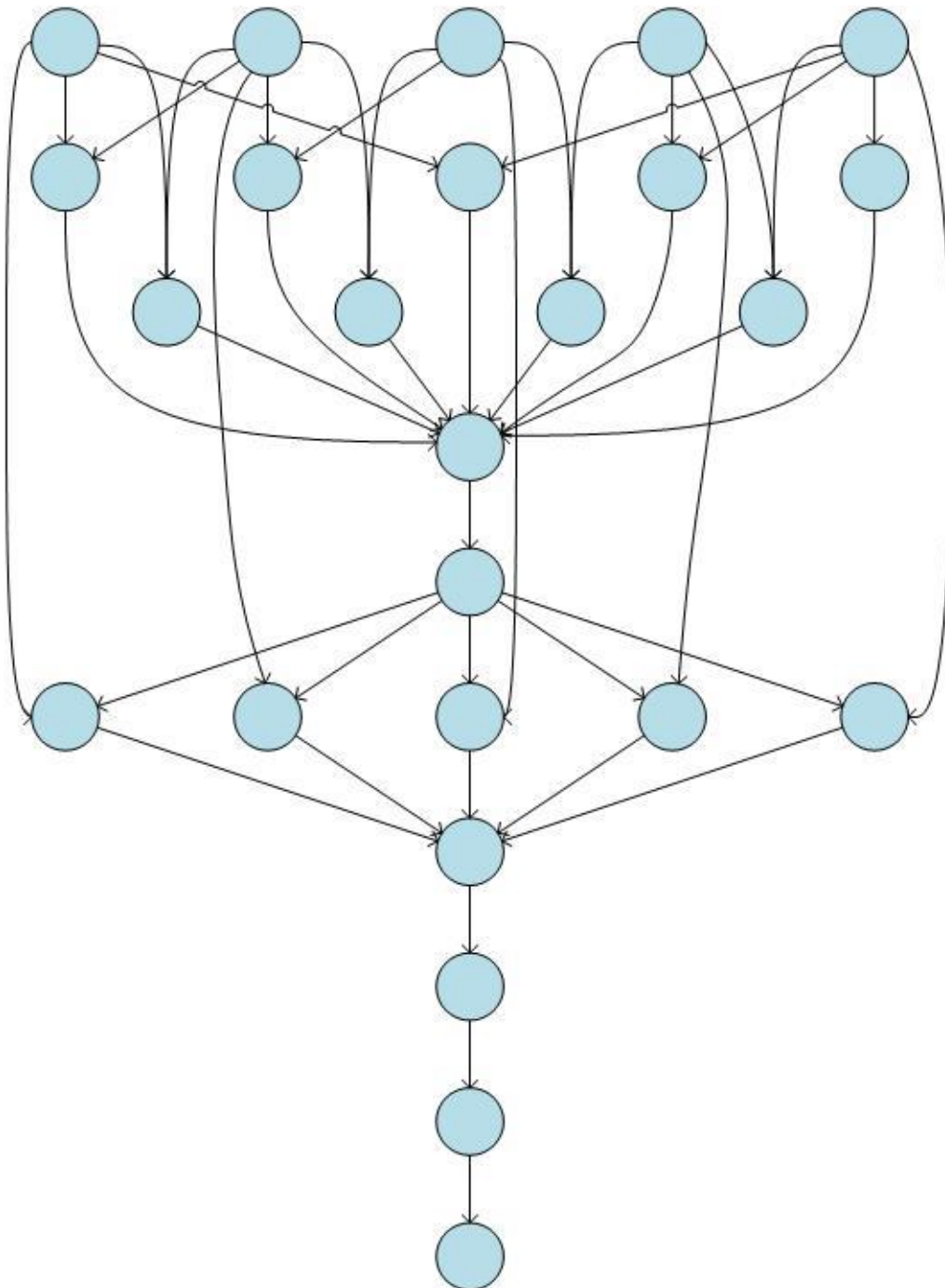


Εικόνα 4-30: Χρόνος εκτέλεσης του *NSGA-II* ανάλογα με το πλήθος των κόμβων του γράφου *Eriogenomics*

Από το διάγραμμα της Εικόνας 4-30 γίνεται φανερό ότι στον ευριστικό αλγόριθμο αυξάνεται σχεδόν γραμμικά ο χρόνος εκτέλεσης συναρτήσει των κόμβων της ροής εργασιών της οικογένειας *Eriogenomics*.

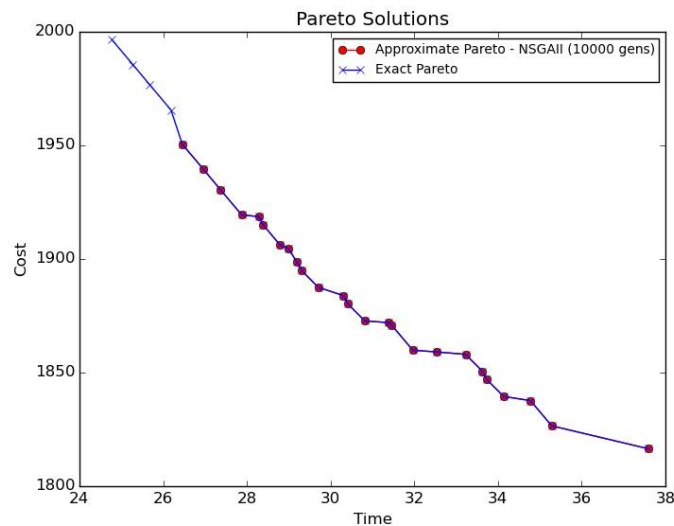
4.4.2. MONTAGE

Στο σημείο αυτό παρουσιάζονται τα αποτελέσματα που αφορούν ροές εργασιών της οικογένειας Montage. Στην Εικόνα 4-31 φαίνεται ένας γράφος της οικογένειας με 25 κόμβους, ο οποίος χρησιμοποιήθηκε πρώτος στη σειρά των πειραμάτων.



Εικόνα 4-31: Γράφος της οικογένειας Montage 25 κόμβων

Αρχικά, για τη ροή εργασιών της Εικόνας 4-31, εκτελέστηκαν και οι δύο αλγόριθμοι για δύο διαθέσιμες μηχανές. Τα αποτελέσματα των δύο αλγορίθμων φαίνονται στην Εικόνα 4-32, ενώ στον Πίνακα 4-5 παρουσιάζονται οι χρόνοι εκτέλεσης των δύο αλγορίθμων και η απόσταση της προσεγγιστικής λύσης από τη βέλτιστη.

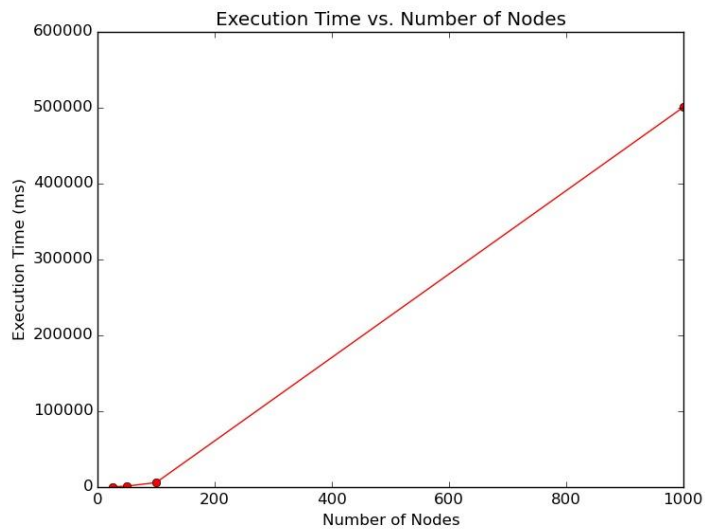


Εικόνα 4-32: Σύνολα λύσεων για ροή εργασιών τύπου *Montage 25* κόμβων με δύο διαθέσιμες μηχανές από τον βέλτιστο και τον προσεγγιστικό αλγόριθμο.

Χρόνος εκτέλεσης βέλτιστου αλγορίθμου	Χρόνος εκτέλεσης NSGA-II (10000 γενιές)	Επιτάχυνση	Απόσταση προσεγ. από βέλτιστη λύση
96665 ms	751 ms	128,72	$5 \cdot 10^{-5}$

Πίνακας 4-5: Χρόνοι εκτέλεσης των δύο αλγορίθμων και η απόσταση των λύσεων τους για τη ροή εργασιών *Montage 25* κόμβων με δύο διαθέσιμες μηχανές

Και σε αυτή την περίπτωση εξάγουμε αντίστοιχα συμπεράσματα με αυτά της οικογένειας *Erigonomics*. Λόγω της πολυπλοκότητας της μορφής του γράφου δεν συνίσταται η χρήση του βέλτιστου αλγορίθμου, αλλά του προσεγγιστικού, ο οποίος φαίνεται ότι παρουσιάζει αμελητέο σφάλμα και εκτελείται 128,72 φορές γρηγορότερα. Για το λόγο αυτό, για μεγαλύτερους γράφους εκτελούμε και πάλι μόνο τον προσεγγιστικό αλγόριθμο NSGA-II με τον κατάλληλο αριθμό γενεών ανάλογα με το μέγεθος του γράφου. Στην Εικόνα 4-33 φαίνεται το διάγραμμα των χρόνων εκτέλεσης του αλγορίθμου.

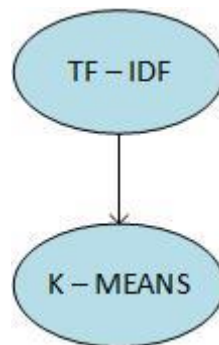


Εικόνα 4-33: Χρόνος εκτέλεσης του *NSGA-II* ανάλογα με το πλήθος των κόμβων του γράφου *Montage*

Στην περίπτωση αυτή, φαίνεται ότι η αυξημένη πολυπλοκότητα της μορφής του γράφου, επηρεάζει αρκετά το χρόνο εκτέλεσης του προσεγγιστικού αλγορίθμου όταν αυξηθεί σημαντικά το πλήθος των κόμβων. Ωστόσο, δεν πρόκειται για απαγορευτικούς χρόνους αναλογικά με το μέγεθος του γράφου.

4.5. ΕΥΡΕΣΗ ΛΥΣΗΣ ΣΕ ΡΟΗ ΕΡΓΑΣΙΩΝ ΑΠΟ ΠΡΑΓΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ

Σε αυτό το τμήμα της άσκησης χρησιμοποιείται ο βέλτιστος αλγόριθμος για να βρεθούν τα βέλτιστα πλάνα εκτέλεσης για μία πραγματική ροή εργασιών. Η ροή αυτή είναι γραμμική και αποτελείται από δύο βασικούς τελεστές. Ο πρώτος είναι υπολογίζει το TF – IDF πάνω σε ένα σύνολο κειμένων, ενώ ο δεύτερος ομαδοποιεί τις τιμές που προέκυψαν με χρήση του αλγόριθμου k-means. Αυτή η ροή εργασιών φαίνεται στην Εικόνα 4-34.



Εικόνα 4-34: Πραγματική ροή εργασιών η οποία θα βελτιστοποιηθεί.

Για κάθε έναν από τους δύο τελεστές της ροής υπάρχουν δύο διαφορετικές υλοποιήσεις, μία κεντρική σε scikit και μία κατανεμημένη σε Spark. Επειδή οι υλοποιήσεις αυτές είναι στην ίδια μηχανή εκτέλεσης, δεν υπάρχει μεταφορά δεδομένων ανάμεσα στους κόμβους. Επίσης, ως κόστος χρησιμοποιείται το γινόμενο χρόνος εκτέλεσης * μνήμη. Στον Πίνακα 4-6 για διάφορα μεγέθη αρχείων φαίνονται οι χρόνοι εκτέλεσης και η διαθέσιμη μνήμη που είχε ο κάθε τελεστής.

input data	TF - IDF				K - means			
	Spark		Scikit		Spark		Scikit	
	Χρόνος Εκτέλεσης	Μνήμη	Χρόνος Εκτέλεσης	Μνήμη	Χρόνος Εκτέλεσης	Μνήμη	Χρόνος Εκτέλεσης	Μνήμη
3103	32.6326	4096	11.1512	1024	27.2022	5120	17.4487	1024
6306	28.0087	4096	11.0638	1024	27.3955	5120	26.1718	1024
12503	29.3004	6144	12.055	1024	27.2846	5120	48.4506	1024
158012	32.7928	6144	32.3527	1024	27.5931	5120	-	-
3159319	41.5308	6144	-	-	30.0084	5120	-	-

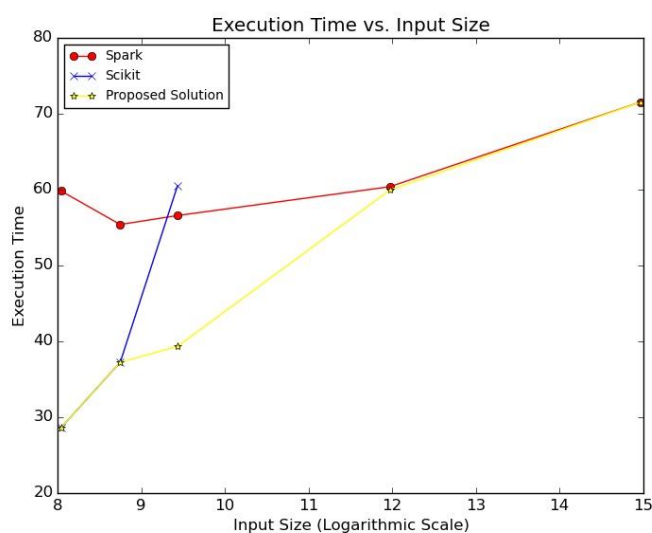
Πίνακας 4-6: Πειραματικά δεδομένα από τη ροή εργασιών της Εικόνας 4-33.

Εκτελώντας τον αλγόριθμο προεκύψαν οι εξής λύσεις:

- Για είσοδο μεγεθους 3103
TF - IDF → Scikit
K - means → Scikit
Χρόνος Εκτέλεσης: 28.5999
Κόστος: 29286,297
- Για είσοδο μεγεθους 6306
TF - IDF → Scikit
K - means → Scikit
Χρόνος Εκτέλεσης: 37.2356
Κόστος: 38129.254
- Για είσοδο μεγεθους 12503
TF - IDF → Scikit
K - means → Scikit
Χρόνος Εκτέλεσης: 60.5056
Κόστος: 61957.734
TF - IDF → Scikit
K - means → Spark
Χρόνος Εκτέλεσης: 39.3396
Κόστος: 152041.47
- Για είσοδο μεγεθους 158012
TF - IDF → Scikit
K - means → Spark
Χρόνος Εκτέλεσης: 59.9458
Κόστος: 174405.84
- Για είσοδο μεγεθους 3159319
TF - IDF → Spark
K - means → Spark
Χρόνος Εκτέλεσης: 71.5392
Κόστος: 408808.2432

Παρατηρώντας τα δεδομένα του Πίνακα 4-6 και με τον τρόπο που ορίστηκε το κόστος, γίνεται φανερό ότι τα παραπάνω αποτελέσματα είναι σωστά ως βέλτιστες λύσεις, καθώς τόσο ο χρόνος όσο και το κόστος είναι εμφανώς μικρότερα όταν εκτελούνται στο Scikit σε σχέση με το Spark. Μοναδική εξαίρεση είναι η περίπτωση του μεγέθους εισόδου 12503, όπου φαίνεται ότι υπάρχει και δεύτερη λύση βέλτιστη κατά Pareto, καθώς η μία είναι καλή ως προς το χρόνο και η άλλη ως προς το κόστος.

Στο επόμενο διάγραμμα φαίνεται ο χρόνος εκτέλεσης της ροής εργασιών αν είχε εκτελεστεί μόνο σε Spark, μόνο σε Scikit ή σύμφωνα με τη λύση που προτάθηκε από το σύστημα, για κάθε μέγεθος εισόδου, συμπεριλαμβάνοντας το overhead του συστήματος. Στην περίπτωση που σε κάποια είσοδο έχουν προταθεί περισσότερα από ένα πλάνα εκτέλεσης λαμβάνεται αυτό με το μικρότερο χρόνο εκτέλεσης.



Εικόνα 4-35: Ο χρόνος εκτέλεσης του πλάνου εκτέλεσης αν εκτελεστεί όλος ο γράφος σε Spark, Scikit ή σύμφωνα με το πλάνο που πρότεινε ο σχεδιαστής.

Γίνεται φανερό ότι η λύση που προτείνεται από το σύστημα πάντα θα έχει χρόνο είτε καλύτερο είτε ίδιο με μία από τις καθολικές λύσεις. Επόμενως, μπορεί να εξασφαλιστεί η καλύτερη απόδοση για την εκτέλεση της ροής εργασιών με τη χρήση του μετα-σχεδιαστή.

5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ

Στην παρούσα διπλωματική εργασία στόχος ήταν η εύρεση των βέλτιστων πλάνων εκτέλεσης ροών εργασιών σε περιβάλλον με πολλαπλές μηχανές. Αυτό πραγματοποιήθηκε τόσο με αλγόριθμο που εγγυάται εύρεση της βέλτιστης λύσης όσο και με προσεγγιστικό αλγόριθμο. Παρατηρήθηκε ότι για μικρές ροές εργασιών αξίζει να χρησιμοποιηθεί ο βέλτιστος αλγόριθμος, όπως και επίσης σε ροές εργασιών με απλή μορφή (γραμμικές ή με μικρό βαθμό παραλληλισμού). Σε ροές με μεγάλο πλήθος κόμβων ή περίπλοκη μορφή, ο χρόνος εκτέλεσης του βέλτιστου αλγόριθμου δεν ευνοεί τη χρήση του και επομένως κρίνεται σκόπιμο να χρησιμοποιηθεί ο ευριστικός αλγόριθμος, ο οποίος με κατάλληλη επιλογή στον αριθμό των γενέων που θα υπολογίσει μπορεί να δώσει ένα ικανοποιητικό αποτέλεσμα σε επιθυμητό χρόνο. Επιπλέον, διαπιστώθηκε ότι με χρήση του μέτα-σχεδιαστή είναι πιθανόν να προκύψουν πλάνα εκτέλεσης με χρόνο ή κόστος εκτέλεσης καλύτερο από το να είχε ανατεθεί όλη η ροή εργασιών να εκτελεστεί στην ίδια μηχανή.

Τέλος, παρατίθενται κάποιες σκέψεις για μελλοντικές επεκτάσεις του συστήματος:

- Τροποποίηση του βέλτιστου αλγόριθμου ώστε να μπορεί να έχει καλύτερη χρονική επίδοση σε μεγαλύτερους και πολυπλοκότερους γράφους.
- Εύρεση προσεγγιστικού αλγορίθμου που να προσεγγίζει ακόμα περισσότερο την πραγματική λύση και ενδεχομένως με σταθερό αριθμό επαναλήψεων που δεν εξαρτάται από τις παραμέτρους του προβλήματος (π.χ. πλήθος κόμβων στη ροή εργασιών, βαθμός παραλληλισμού, κλπ).

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] D. Tsoumakos και C. Mantas, «The Case for Multi-Engine Data Analytics,» *Euro-Par*, 2013.
- [2] «Cloudera Distribution CDH 5.2.0,» [Ηλεκτρονικό]. Available: <http://www.cloudera.com/content/cloudera/en/downloads/cdh/cdh-5-2-0.html>.
- [3] «Hortonworks Sandbox 2.1,» [Ηλεκτρονικό]. Available: <http://hortonworks.com/products/hortonworks-sandbox>.
- [4] «heroku add-ons,» [Ηλεκτρονικό]. Available: <https://addons.heroku.com>.
- [5] «Running Databases on AWS,» [Ηλεκτρονικό]. Available: http://aws.amazon.com/running_databases.
- [6] Wikipedia. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Cloud_computing.
- [7] [Ηλεκτρονικό]. Available: <http://searchcio.techtarget.com/definition/workflow>.
- [8] H. Kllapi, E. Sitaridi, M. Tsangaris και Y. Ioannidis, «Schedule Optimization for Data Processing Flows On the Cloud,» *SIGMOD*, 2011.
- [9] Wikipedia. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Optimization_problem.
- [10] D. Savic, «Single-Objective vs. Multiobjective Optimisation for Integrated Decision Support,» 2002.
- [11] Wikipedia. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Multi-objective_optimization.
- [12] H. Choi, «A Brief Introduction to Skyline Problem (Pareto-optimal Tuples),» [Ηλεκτρονικό]. Available: <http://dbserver.korea.ac.kr/blog/2009/09/06/a-brief-introduction-to-skyline-problem-pareto-optimal-tuples-1/>.
- [13] T. Cormen, C. Leiserson, R. Rivest και C. Stein, «Δυναμικός Προγραμματισμός,» σε *Εισαγωγή στους Αλγορίθμους*, 2006.
- [14] Wikipedia. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Evolutionary_algorithm.
- [15] K. Deb, A. Pratap, S. Agarwal και T. Meyarivan, «A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,» *IEEE Transactions on Evolutionary Computation*, τόμ. 6, 2002.
- [16] Wikipedia. [Ηλεκτρονικό]. Available:

https://en.wikipedia.org/wiki/Hausdorff_distance.

[17] Wikipedia. [Ηλεκτρονικό]. Available:
[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)).

[18] D. Hadka, σε *MOEA Framework, A Free and Open Source Java Framework for Multiobjective Optimization, User Manual*, 2014.

[19] Wikipedia. [Ηλεκτρονικό]. Available:
https://en.wikipedia.org/wiki/MOEA_Framework.