



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Βέλτιστη κατανομή πόρων για την  
ελαχιστοποίηση της απόκρισης και της  
ενέργειας σε σενάρια μεταφόρτωσης  
εφαρμογών

Διπλωματική Εργασία

του

ΓΙΑΝΝΟΠΟΥΛΟΥ ΦΩΤΙΟΥ

Επιβλέπων: Συμεών Παπαβασιλείου  
Καθηγητής

Αθήνα, Σεπτέμβριος 2024

---





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟ-  
ΦΟΡΙΚΗΣ

## Βέλτιστη κατανομή πόρων για την ελαχιστοποίηση της απόκρισης και της ενέργειας σε σενάρια μεταφόρτωσης εφαρμογών

Διπλωματική Εργασία  
του  
**ΓΙΑΝΝΟΠΟΥΛΟΥ ΦΩΤΙΟΥ**

**Επιβλέπων:** Συμεών Παπαβασιλείου  
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26η Σεπτεμβρίου 2024.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Συμεών Παπαβασιλείου  
Καθηγητής

.....  
Γεώργιος Ματσόπουλος  
Καθηγητής

.....  
Ιωάννα Ρουσσάκη  
Αναπληρώτρια Καθηγήτρια

Αθήνα, Σεπτέμβριος 2024





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟ-  
ΦΟΡΙΚΗΣ

Copyright © – All rights reserved. Με την επιφύλαξη παντός δικαιώματος.  
Γιαννόπουλος Φώτιος , 2024.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση ότι αναφέρεται η πηγή προέλευσης και ότι διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να θεωρηθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

(Υπογραφή)

.....

Γιαννόπουλος Φώτιος

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



## Περίληψη

---

Οι κινητές συσκευές στις μέρες μας χρησιμοποιούν σε καθημερινή βάση πολλαπλές εφαρμογές με σύνθετες απαιτήσεις. Επομένως, δημιουργείται η ανάγκη για τη βέλτιστη κατανομή των πόρων της κινητής συσκευής για την εκτέλεση των εργασιών της, μεταφορτώνοντας κάποιες από αυτές και σε άλλες υπολογιστικές μονάδες. Σε αυτή τη διπλωματική εργασία εξετάζουμε ένα σύστημα με μία κινητή συσκευή, η οποία μπορεί να εκτελεί τις εργασίες της τοπικά ή να τις μεταφορτώνει σε ορισμένα σημεία πρόσβασης, τα οποία εκτελούν τις εργασίες που λαμβάνουν. Σκοπός είναι να βρεθεί ο βέλτιστος συνδυασμός για την ανάθεση των εργασιών στα σημεία πρόσβασης, ώστε να ελαχιστοποιούνται η συνολική ενέργεια που δαπανείται για την εκτέλεση των εργασιών και ο συνολικός χρόνος εκτέλεσης των εργασιών.

Η μοντελοποίηση του συστήματος καταλήγει σε ένα quadratic πρόβλημα βελτιστοποίησης με quadratic περιορισμούς(QCQP), το οποίο είναι NP-hard και προτείνουμε μια τεχνική προσέγγισης της λύσης με τη χρήση του αλγορίθμου SDR(semi-definite relaxation), λόγω της ταχύτητας της εκτέλεσης του και της ακρίβειας της προσέγγισης του στην βέλτιστη λύση. Αφαιρώντας ορισμένους περιορισμούς, ο αλγόριθμος SDR μετατρέπει το πρόβλημα σε ένα SDP(semi-definite programming), το οποίο είναι ένα κυρτό(convex) πρόβλημα ελαχιστοποίησης και μπορεί να λυθεί σε πολυωνυμικό χρόνο. Οι προσομοιώσεις μας τονίζουν ότι η τεχνική SDR επιτυγχάνει σχεδόν βέλτιστη απόδοση. Επιπλέον, παρατηρούμε αύξηση της απόδοσης σχετικά με τον χρόνο εκτέλεσης και την εξοικονόμηση ενέργειας, με την αύξηση των σημείων πρόσβασης. Τέλος, σημειώνουμε τη σημασία της μεταφόρτωσης των εργασιών για την κινητή συσκευή.

## Λέξεις Κλειδιά

Σημεία πρόσβασης, Μεταφόρτωση εργασιών, Κυρτό πρόβλημα, Ανάθεση εργασιών, SDR, QCQP, SDP, NP-hard





## Abstract

---

Mobile devices nowadays use multiple applications with complex requirements on a daily basis. Therefore, there is a need for the optimal allocation of the mobile device's resources for performing its tasks, by offloading some of them to other computational units. In this thesis, we examine a system with a mobile device that can either perform its tasks locally or offload them to certain access points, which execute the tasks they receive. The aim is to find the optimal combination for assigning tasks to the access points in order to minimize the total energy expended for task execution and the overall execution time of the tasks.

The modeling of the system results in a quadratic optimization problem with quadratic constraints (QCQP), which is NP-hard, and we propose an approximation technique for the solution using the SDR (semi-definite relaxation) algorithm due to its execution speed and the accuracy of its approximation to the optimal solution. By removing certain constraints, the SDR algorithm converts the problem into an SDP (semi-definite programming), which is a convex minimization problem and can be solved in polynomial time. Our simulations highlight that the SDR technique achieves nearly optimal performance. Additionally, we observe an increase in performance in terms of execution time and energy savings with the increase in access points. Finally, we note the importance of task offloading for the mobile device.

## Keywords

Access points, Task offloading, Convex problem, Task allocation, SDR, QCQP, SDP, NP-hard



## Ευχαριστίες

---

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Συμεών Παπαβασιλείου για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων. Επίσης ευχαριστώ ιδιαίτερα τον Δρ. Δημήτρη Σπαθαράκη για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Αθήνα, Σεπτέμβριος 2024

*Γιαννόπουλος Φώτιος*



# Περιεχόμενα

---

<b>Περίληψη</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>Ευχαριστίες</b>	<b>5</b>
<b>1 Εισαγωγή</b>	<b>13</b>
1.1 Αντικείμενο Διπλωματικής Εργασίας	14
1.2 Οργάνωση Τόμου	15
<b>I Θεωρητικό Υπόβαθρο</b>	<b>17</b>
<b>2 Σχετική Βιβλιογραφία</b>	<b>19</b>
2.1 Βιβλιογραφία για την μεταφόρτωση εργασιών	19
2.1.1 Υπολογιστική δομή μεταφόρτωσης εργασιών	19
2.1.2 Το πρόβλημα της ανάθεσης εργασιών	19
2.1.3 Στρατηγικές για τη Μεταφόρτωση εργασιών	20
2.1.4 Υπολογιστικά Μοντέλα	20
2.1.5 Τεχνικές για τη λήψη αποφάσεων	21
2.1.6 Αλγόριθμοι για την ανάθεση εργασιών	21
2.2 Επίλυση με προσέγγιση SDR	22
2.2.1 SDR προσέγγιση σε μεταφόρτωση στο Cloud με υπολογιστικό ση- μείο πρόσβασης	22
2.2.2 SDR προσέγγιση σε μεταφόρτωση στο Cloud με Fog Computing	23
2.3 Κριτήρια Επίδοσης στη μεταφόρτωση εργασιών	23
2.3.1 Κριτήρια επίδοσης στη βιβλιογραφία	23
<b>3 Η τεχνική SDR</b>	<b>25</b>
3.1 Ο αλγόριθμος SDR αναλυτικά	25
3.1.1 Παραδείγματα quadratic προβλημάτων	25
3.1.2 Η ιδέα του SDR	26
3.1.3 Τεχνική Randomization	28
3.1.4 Ο SDR σε μη ομογενή QCQP	29

<b>II</b>	<b>Μοντελοποίηση, Αντιμετώπιση και Προσομοιώσεις</b>	<b>31</b>
<b>4</b>	<b>Ορισμός του Προβλήματος και προσέγγιση με SDR</b>	<b>33</b>
4.1	Μοντελοποίηση του συστήματος και Διατύπωση Προβλήματος . . . . .	33
4.1.1	Μοντελοποίηση του συστήματος . . . . .	33
4.1.2	Ορισμός του προβλήματος ελαχιστοποίησης . . . . .	35
4.2	Βελτιστοποίηση της ανάθεσης εργασιών με σταθερή συχνότητα CPU . .	37
4.3	Προσέγγιση με τον αλγόριθμο SDR . . . . .	38
<b>5</b>	<b>Πειράματα - Προσομοιώσεις</b>	<b>41</b>
5.1	Συγκρίσεις Αλγορίθμων . . . . .	41
5.1.1	Τιμές των παραμέτρων . . . . .	41
5.1.2	Αριθμητικά Αποτελέσματα . . . . .	41
5.2	Πολλαπλά σημεία πρόσβασης . . . . .	43
5.3	Κατανάλωση Ενέργειας και Χρονική Καθυστέρηση . . . . .	44
5.4	Συνολικό κόστος και Ποσοστό μεταφόρτωσης . . . . .	44
5.5	Συμπεράσματα από τις προσομοιώσεις . . . . .	46
<b>III</b>	<b>Επίλογος</b>	<b>49</b>
<b>6</b>	<b>Επίλογος</b>	<b>51</b>
6.1	Σύνοψη και Συμπεράσματα . . . . .	51
6.2	Μελλοντικές Επεκτάσεις . . . . .	52
	<b>Βιβλιογραφία</b>	<b>55</b>
	<b>Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια</b>	<b>57</b>
	<b>Απόδοση ξενόγλωσσων όρων</b>	<b>59</b>

## Κατάλογος σχημάτων

---

5.1	Κόστος αλγορίθμων για μικρά data rates . . . . .	42
5.2	Κόστος αλγορίθμων για μεσαία data rates . . . . .	42
5.3	Κόστος αλγορίθμων για μεγάλα data rates . . . . .	43
5.4	Ποσοστό προσέγγισης ανά L για διαφορετικά M . . . . .	43
5.5	Κατανάλωση ενέργειας και Χρόνος εκτέλεσης . . . . .	44
5.6	Συνολικό κόστος . . . . .	45
5.7	Ποσοστό μεταφόρτωσης . . . . .	45





## Κατάλογος πινάκων

---

4.1	Μεταβλητές που χρησιμοποιούνται σε αυτή την εργασία . . . . .	35
-----	---	----



## Εισαγωγή

---

**Ο**ι κινητές συσκευές έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητας μας. Πλέον, μία κινητή συσκευή εκτελεί έναν μεγάλο αριθμό εργασιών από κλήσεις και περιήγηση στο διαδίκτυο, μέχρι και αναγνώριση ομιλίας και χώρου [1]. Λαμβάνοντας υπόψη και τη λειτουργία τους με μπαταρία, γίνεται προφανής η σημασία για εξοικονόμηση δεδομένων, μέσω της χρήσης αποδοτικών μεθόδων για την επεξεργασία τους. Μία αντιμετώπιση για αυτό είναι η χρήση Mobile Cloud Computing (MCC), με τους επεξεργαστές που χρησιμοποιούνται από το cloud να έχουν μεγαλύτερη υπολογιστική ισχύ από αυτούς της κινητής συσκευής [2]. Με αυτόν τον τρόπο μπορούν να επεξεργάζονται τα δεδομένα που λαμβάνουν από την κινητή συσκευή, η οποία εξοικονομεί μπαταρία, υπολογιστική ενέργεια και χώρο αποθήκευσης.

Για την γρήγορη και ακριβή μεταφόρτωση των δεδομένων από την κινητή συσκευή στο cloud απαιτείται και η κατάλληλη δομή, όπως με τις εφαρμογές MAUI [3] και ThinkAir [4]. Με αυτές τις δομές, η κινητή συσκευή μπορεί να χωρίζει τα δεδομένα της σε εργασίες, τις οποίες στη συνέχεια επεξεργάζονται οι εξυπηρετητές του cloud. Οι εξυπηρετητές αυτοί βρίσκονται μακριά από την κινητή συσκευή, οπότε η χρονική καθυστέρηση της μεταφοράς των δεδομένων είναι μια σημαντική παράμετρος που πρέπει να λάβουμε υπόψη. Ένα δίκτυο small cell, το οποίο είναι ένα σύστημα ασύρματης επικοινωνίας που χρησιμοποιεί μικρούς σταθμούς για την βελτίωση της κάλυψης σε μεγάλες αποστάσεις, μπορεί να βοηθήσει σε αυτό [5]. Με τη χρήση μικρών σημείων πρόσβασης small cell κοντά στους χρήστες, τα οποία μπορούν να επεξεργαστούν τα δεδομένα της κινητής συσκευής, μειώνεται σημαντικά ο χρόνος μεταφόρτωσης των δεδομένων.

Η κινητή συσκευή μπορεί να στέλνει διαφορετικές εργασίες σε διαφορετικά σημεία πρόσβασης ή να εκτελεί κάποιες από αυτές τοπικά. Τα δύο σημαντικότερα κριτήρια που επηρεάζουν την απόφαση για την ανάθεση των εργασιών είναι η χρονική καθυστέρηση μεταφόρτωσης και εκτέλεσής τους και η κατανάλωση ενέργειας που απαιτείται. Ανάλογα με τη σημασία που έχει για τον χρήστη το κάθε ένα από τα δύο κριτήρια, μπορεί να παρθεί διαφορετική απόφαση για την ανάθεση των εργασιών. Για παράδειγμα, σε περίπτωση που ο χρήστης επιθυμεί να μην επιβαρύνει την συσκευή, οι περισσότερες εργασίες μεταφορτώνονται στα σημεία πρόσβασης, εξοικονομώντας ενέργεια.

Υπάρχουν πολλοί παράγοντες που επηρεάζουν τα δύο αυτά κριτήρια, όπως η υπο-

λογιστική ισχύς της κινητής συσκευής και των σημείων πρόσβασης, ο όγκος των δεδομένων, ο χρόνος μεταφόρτωσης των εργασιών, οι συχνότητες των επεξεργασιών κ.ά. Για αυτόν τον λόγο, η επιλογή της ανάθεσης των εργασιών είναι αρκετά πολύπλοκη διαδικασία, ειδικά για μεγάλο όγκο δεδομένων. Η εύρεση της πιο αποδοτικής λύσης είναι αρκετά χρονοβόρα στις περισσότερες περιπτώσεις, οπότε χρειαζόμαστε έναν άλλον τρόπο για να λύσουμε το πρόβλημα, καθώς η γρήγορη ανάθεση των εργασιών είναι απαραίτητη για να λειτουργεί σωστά το σύστημα αυτό. Προτιμάμε αλγορίθμους, οι οποίοι μπορούν να προσεγγίζουν την αποδοτική λύση σε χρόνο αρκετά μικρότερο από αλγορίθμους, οι οποίοι εξετάζουν ολόκληρο το χώρο επιλογών για να βρουν την βέλτιστη λύση. Αν οι προσεγγίσεις αυτές είναι σταθερά κοντά στην αποδοτική λύση, τότε έχουμε ένα καλό αποτέλεσμα.

## 1.1 Αντικείμενο Διπλωματικής Εργασίας

Στην εργασία αυτή, θα ασχοληθούμε με ένα σύστημα σαν αυτό που αναφέρθηκε παραπάνω. Θα μοντελοποιήσουμε το πρόβλημα της ανάθεσης εργασιών από την κινητή συσκευή στα σημεία πρόσβασης και θα το αναλύσουμε με σκοπό να μπορεί να αντιμετωπιστεί με τους κατάλληλους αλγορίθμους. Το σύστημα αυτό έχει σταθερή συχνότητα CPU και μπορεί να αναθέτει εργασίες σε διαφορετικά σημεία πρόσβασης. Θα ορίσουμε το πρόβλημα ως ένα μαθηματικό πρόβλημα ελαχιστοποίησης και θα καταλήξουμε σε μία μορφή που μπορεί να αντιμετωπιστεί από ορισμένους αλγορίθμους, τους οποίους θα εξετάσουμε ως προς την αποτελεσματικότητά τους. Το πρόβλημα αυτό είναι NP-hard, οπότε προτείνουμε προσεγγιστικούς αλγορίθμους για να το λύσουμε. Θα ερευνήσουμε τα πλεονεκτήματα της μεταφόρτωσης εργασιών στα σημεία πρόσβασης, όσον αφορά την χρονική καθυστέρηση και την κατανάλωση ενέργειας.

Θα ασχοληθούμε με τον αλγόριθμο SDR(semi-definite relaxation), θα παρουσιάσουμε την δομή του, τον τρόπο δράσης του και τις περιπτώσεις στην οποίες είναι χρήσιμος. Μετατρέποντας το πρόβλημα μας σε ένα quadratic πρόβλημα βελτιστοποίησης, μπορεί να χρησιμοποιηθεί ο αλγόριθμος SDR για την προσέγγιση της λύσης. Θα εφαρμόσουμε τον αλγόριθμο αυτόν στο σύστημα, υλοποιώντας τα βήματα του. Σκοπός είναι η κατασκευή μιας αποδοτικής λύσης για το πρόβλημα, χρησιμοποιώντας τον SDR.

Στη συνέχεια, θα παρουσιάσουμε προσομοιώσεις από το σύστημα. Συγκρίνουμε αλγορίθμους και παρατηρούμε την αποτελεσματικότητά τους στην επίλυση του προβλήματος για διαφορετικές τιμές των παραμέτρων. Επίσης, συγκρίνουμε για διαφορετικά μεγέθη εργασιών, σημείων πρόσβασης και ρυθμών μετάδοσης των δεδομένων. Συγκρίνονται διαφορετικές προσεγγίσεις σχετικά με τον συμβιβασμό ανάμεσα στην χρονική καθυστέρηση και την κατανάλωση ενέργειας και τη σχέση των δύο.

Όσον αφορά τον κώδικα, μέσω της python θα δημιουργήσουμε ένα πρόγραμμα το οποίο προσομοιώνει το σύστημα και θα ορίσουμε το πρόβλημα της ελαχιστοποίησης. Θα υλοποιήσουμε τον SDR και με την βοήθεια του θα υπολογίσουμε λύσεις για το πρόβλημα. Θα δοκιμάσουμε τον αλγόριθμο στις προσομοιώσεις, με σκοπό να κατανοήσουμε τη συμπεριφορά του σε διαφορετικές τιμές παραμέτρων. Βασικό κριτήριο είναι

το πόσο κοντά στην αποδοτική λύση μπορεί να πλησιάσει. Θα υλοποιήσουμε και αλγόριθμους για τοπική και τυχαία μεταφόρτωση και θα τους συγκρίνουμε.

Υποθέτουμε ότι οι εργασίες στέλνονται από μία κινητή συσκευή στα σημεία πρόσβασης και ότι τα σημεία πρόσβασης δεν λαμβάνουν δεδομένα από άλλες κινητές συσκευές. Σε αντίθετη περίπτωση θα παρατηρούνταν παραπάνω καθυστερήσεις και πιθανώς θα χάνονταν δεδομένα. Δεν θα ερευνήσουμε την περίπτωση των πολλαπλών κινητών συσκευών σε αυτή την εργασία.

## 1.2 Οργάνωση Τόμου

Η εργασία αυτή είναι οργανωμένη σε έξι κεφάλαια: Στο Κεφάλαιο 2 δίνεται η βιβλιογραφία, η οποία σχετίζεται με το αντικείμενο αυτής της εργασίας. Αναφέρονται πληροφορίες σχετικά με τη μεταφόρτωση εργασιών, το πρόβλημα της ανάθεσής τους και τον αλγόριθμο SDR. Στο Κεφάλαιο 3 περιγράφεται αναλυτικά η τεχνική SDR, οι εφαρμογές της και προβλήματα στα οποία μπορεί να βοηθήσει. Στο Κεφάλαιο 4 παρουσιάζεται η μοντελοποίηση του συστήματος, ο ορισμός του προβλήματος της ανάθεσης εργασιών και οι μαθηματικές σχέσεις που το εκφράζουν. Επιπλέον εφαρμόζεται ο αλγόριθμος SDR για την επίλυση του προβλήματος. Στο κεφάλαιο 5 παρουσιάζονται οι προσομοιώσεις του συστήματος και η απόδοση των προτεινόμενων αλγορίθμων για διάφορες τιμές των παραμέτρων του συστήματος, καθώς και πειράματα σχετικά με τη σχέση της κατανάλωσης ενέργειας και της χρονικής καθυστέρησης. Τέλος, στο Κεφάλαιο 6 αναφέρονται τα συμπεράσματα αυτής της διπλωματικής εργασίας, καθώς και πιθανές μελλοντικές επεκτάσεις.



Μέρος **I**

Θεωρητικό Υπόβαθρο

---





## Σχετική Βιβλιογραφία

---

Στο κεφάλαιο αυτό παρουσιάζεται η βιβλιογραφία σχετικά με την μεταφόρτωση εργασιών και τη χρήση της τεχνικής SDR.

### 2.1 Βιβλιογραφία για την μεταφόρτωση εργασιών

#### 2.1.1 Υπολογιστική δομή μεταφόρτωσης εργασιών

Με την ραγδαία άνοδο της χρήσης κινητών συσκευών τα τελευταία χρόνια, καλούμαστε να βρούμε τρόπους για τη βέλτιστη κατανομή των πόρων κατά τη χρήση εφαρμογών. Ειδικά οι εφαρμογές, οι οποίες χρησιμοποιούν μέσα όπως η κάμερα και το μικρόφωνο επιβαρύνουν σε μεγάλο βαθμό τη συσκευή. Για αυτό λόγο, προκύπτει η ανάγκη για μεταφόρτωση των δεδομένων αυτών σε άλλη δομή για την επεξεργασία τους [6]. Μέχρι σήμερα ο υπολογισμός μέσω cloud (Mobile Cloud Computing-MCC) θεωρούταν μια αποδοτική λύση σε αυτό το πρόβλημα, χάρη στην τεράστια υπολογιστική ισχύ και το χώρο αποθήκευσης που έχουν οι εξυπηρετητές cloud [7], οπότε η μεταφόρτωση εργασιών που απαιτούν έντονη επεξεργασία στο cloud μειώνει το φόρτο στην κινητή συσκευή και εξοικονομεί κρίσιμη ενέργεια. Το πρόβλημα που δημιουργείται στην περίπτωση αυτή είναι η καθυστέρηση που μπορούν να επιφέρουν οι εξυπηρετητές cloud λόγω της γεωγραφικής τους απόστασης από την κινητή συσκευή.

Για την αντιμετώπιση του προβλήματος αυτού έχει εμφανιστεί η τεχνική πολλαπλής πρόσβασης Edge Computing(MEC), με σκοπό να φέρει την υπολογιστική ισχύ του MCC στην άκρη του δικτύου, κοντά στην κινητή συσκευή [8]. Με αυτόν τον τρόπο, μπορεί τώρα η συσκευή να μεταφορτώνει τις εργασίες της σε έναν κοντινό εξυπηρετητή, μειώνοντας κατά πολύ τον χρόνο απάντησης και χρησιμοποιώντας τις ίδιες υπολογιστικές δυνατότητες. Οι εξυπηρετητές αυτοί βρίσκονται σε μικρούς σταθμούς - σημεία πρόσβασης στην άκρη του δικτύου της κινητής συσκευής [9].

#### 2.1.2 Το πρόβλημα της ανάθεσης εργασιών

Στη συνέχεια, προκύπτει το ζήτημα της βέλτιστης κατανομής των εργασιών για μεταφόρτωση, τόσο ανάμεσα στα σημεία πρόσβασης, αλλά και η επιλογή των εργασιών

που θα προωθηθούν από την κινητή συσκευή. Εκτός από τα χαρακτηριστικά των εξυπηρετητών - σημείων πρόσβασης, πρέπει να ληφθούν υπόψη και οι παράμετροι των εργασιών, όπως οι πόροι που χρειάζονται και οι προτεραιότητες που μπορεί να υπάρχουν για την εκτέλεση τους. Το πρόβλημα της επιλογής της κατάλληλης εργασίας για την μεταφόρτωση στο κατάλληλο σημείο πρόσβασης είναι ένα πρόβλημα βελτιστοποίησης με πολλαπλά κριτήρια, το οποίο γνωρίζουμε πως είναι NP-hard [10].

Η μεταφόρτωση εργασιών σε αυτό το πλαίσιο, μπορεί να οριστεί ως προώθηση μιας πλήρους δομής από μια εφαρμογή, η οποία περιλαμβάνει τα απαραίτητα δεδομένα για την εκτέλεση και άλλες απαραίτητες βιβλιοθήκες σε έναν απομακρυσμένο εξυπηρετητή και επιστροφή του αποτελέσματος της εκτέλεσης από τον εξυπηρετητή. Ένας βασικός στόχος είναι η βελτιστοποίηση της χρησιμοποίησης των πόρων και ταυτόχρονα η ελαχιστοποίηση του χρόνου εκτέλεσης των εργασιών [11]. Ο υπολογισμός αυτός μπορεί να γίνει είτε στη συσκευή είτε στα σημεία πρόσβασης. Για κάθε εργασία λαμβάνεται η απόφαση της μεταφόρτωσης της ή της εκτέλεσης της τοπικά, ανάλογα και με τις παραμέτρους του συστήματος.

### 2.1.3 Στρατηγικές για τη Μεταφόρτωση εργασιών

Όσον αφορά την μεταφόρτωση εργασιών μπορούμε να διακρίνουμε διαφορετικές περιπτώσεις- στρατηγικές για την αντιμετώπιση τους σύμφωνα με [12]. Από την πλευρά του υπολογιστικού μοντέλου, από την πλευρά της λήψης αποφάσεων και από την πλευρά των αλγορίθμων που χρησιμοποιούνται.

### 2.1.4 Υπολογιστικά Μοντέλα

Ένα υπολογιστικό μοντέλο που χρησιμοποιείται ευρέως είναι αυτό της συγκεντροποίησης (centralized). Το μοντέλο αυτό περιλαμβάνει έναν κεντρικό εξυπηρετητή με μεγάλη υπολογιστική δυνατότητα και σύγχρονες εφαρμογές. Ο κεντρικός εξυπηρετητής συνδέεται με υπολογιστές - πελάτες, οι οποίοι έχουν χαμηλή ισχύ επεξεργασίας και όταν χρειάζεται να εκτελεστεί μια εργασία, οι πελάτες την προωθούν στον κεντρικό εξυπηρετητή, ο οποίος εκτελεί το μεγαλύτερο μέρος της επεξεργασίας. Η σύνδεση ανάμεσα στον εξυπηρετητή και τους πελάτες μπορεί να γίνει είτε απευθείας είτε μέσω ενός δικτύου. Το σενάριο αυτό έχει χαμηλότερη αποδοτικότητα και μεγαλύτερο ρίσκο αποτυχίας, αφού όλες οι εργασίες περνούν από τον κεντρικό εξυπηρετητή. Προσφέρει όμως, μεγαλύτερη ασφάλεια και αξιοπιστία για τα δεδομένα του χρήστη, αφού όλα αποθηκεύονται σε ένα μέρος.

Το μοντέλο της αποκέντρωσης (Decentralized) από την άλλη, βασίζεται σε διαφορετικούς εξυπηρετητές, οι οποίοι αποφασίζουν για τις δικές τους επιλογές. Ένα τέτοιο σύστημα έχει μεγάλη διαθεσιμότητα, υψηλή αντοχή σε λάθη και δεν υπάρχει ένα συγκεκριμένο σημείο, το οποίο να επιφέρει αποτυχία στο σύστημα. Ο σχεδιασμός ενός τέτοιου συστήματος είναι αρκετά απαιτητικός, καθώς αντιμετωπίζει δυσκολίες σχετικά με τη διαχείριση αντιγράφων και την διατήρηση της συνέπειας ανάμεσα σε διαφορετικούς εξυπηρετητές.

Ένα τρίτο μοντέλο είναι το καταναμημένο (Distributed), κατά το οποίο πολλαπλοί εξυπηρετητές cloud επικοινωνούν και συνεργάζονται, ώστε να εμφανίζονται ως ένα κοινό σύστημα στους χρήστες. Ανάλογα με το κάθε σύστημα διαφέρει η υλοποίηση, αλλά γενικά σε αυτήν την περίπτωση ένας ελεγκτής κατανέμει τις εργασίες στους εξυπηρετητές με βάση διάφορα κριτήρια όπως το είδος της εφαρμογής, τον επιθυμητό χρόνο εκτέλεσης, το είδος των απαραίτητων πόρων, και άλλες παραμέτρους. Ένα καταναμημένο σύστημα έχει μεγάλη διαθεσιμότητα και μπορεί να αναπτυχθεί και οριζόντια και κάθετα. Παρέχει αρκετά πλεονεκτήματα, αλλά αντιμετωπίζει και προβλήματα σχετικά με την ετερογένεια, την διαχείριση της αποτυχίας και την ασφάλεια.

### 2.1.5 Τεχνικές για τη λήψη αποφάσεων

Όσον αφορά τον τρόπο επιλογής της ανάθεσης εργασιών, μια προσέγγιση είναι η επιλογή από το cloud. Σε αυτήν την περίπτωση, οι πληροφορίες των εργασιών στέλνονται στο cloud, όπου λαμβάνεται η απόφαση για την ανάθεση των εργασιών και στέλνεται στις κινητές συσκευές, οι οποίες μεταφορτώνουν αναλόγως. Σε μια άλλη προσέγγιση, οι κινητές συσκευές υπολογίζουν την ανάθεση των εργασιών και στη συνέχεια πραγματοποιούν την μεταφόρτωση. Σε αυτήν την στρατηγική, ο αλγόριθμος που χρησιμοποιούν είναι πολύ σημαντικός, καθώς πρέπει να είναι ακριβής χρησιμοποιώντας τους ελάχιστους πόρους της συσκευής. Οδηγεί συχνά σε μεγάλη κατανάλωση ενέργειας.

Μια άλλη περίπτωση είναι η χρήση ενός λογισμικού SDN (Software-defined networking) για την διαχείριση του δικτύου, το οποίο προσφέρει δυναμική, αποδοτική επεξεργασία για την βελτίωση της απόδοσης του δικτύου [13]. Στόχος είναι η κεντροποίηση της υπολογιστικής δύναμης του δικτύου. Η απόφαση της ανάθεσης των εργασιών μοντελοποιείται από το λογισμικό με βάση διαφορετικές πληροφορίες που συλλέγει από την κινητή συσκευή και από διαφορετικούς εξυπηρετητές του cloud. Η απόφαση μεταδίδεται από τους ελεγκτές που περιλαμβάνει το λογισμικό. Έχει πλεονεκτήματα, όπως η ελαστικότητα ανάλογα με το δίκτυο και η εύκολη λύση προβλημάτων, αλλά αντιμετωπίζει θέματα σχετικά με την ασφάλεια και την αξιοπιστία. Ένας άλλος τρόπος απόφασης είναι μέσω μιας συνεργασίας ανάμεσα στην κινητή συσκευή, τα σημεία πρόσβασης του cloud και το cloud. Υπάρχουν αρκετοί διαφορετικοί τρόποι να υλοποιηθεί αυτό το σύστημα και καθένας από αυτούς αντιμετωπίζει διαφορετικές προκλήσεις.

### 2.1.6 Αλγόριθμοι για την ανάθεση εργασιών

Μπορούμε να κατηγοριοποιήσουμε την ανάθεση εργασιών με βάση τους αλγόριθμους που χρησιμοποιούνται. Οι άπληστες ευριστικές είναι μια προσέγγιση, καθώς το πρόβλημα είναι NP-hard. Τείνουν να βρίσκουν μία σχεδόν βέλτιστη λύση, χωρίς να απαιτούν πολύ χρόνο υπολογισμού ή χώρο μνήμης, προσφέροντας σχετικά χαμηλή πολυπλοκότητα. Για την εφαρμογή τους χρειαζόμαστε γνώση από πριν σχετικά με τις παραμέτρους της βελτιστοποίησης. Μια άλλη λύση είναι ο ακέραιος προγραμματισμός (Integer Programming), με στόχο την ελαχιστοποίηση μιας συνάρτησης, η οποία σχετί-

ζεται με το κόστος της ανάθεσης εργασιών. Μπορεί να λύσει τέτοια προβλήματα, αλλά η μοντελοποίηση που απαιτείται είναι αρκετά δύσκολη. Η μηχανική μάθηση μπορεί επίσης να προσφέρει στην ανάθεση εργασιών, μέσω της δημιουργίας ενός μοντέλου και προπόνησης του. Έχει αρκετά πλεονεκτήματα, αλλά συχνά είναι δύσκολη η προπόνηση βαθιών νευρωνικών δικτύων.

Ο αλγόριθμος Branch and bound (BnB) χρησιμοποιείται για την αντιμετώπιση προβλημάτων βελτιστοποίησης. Χωρίζει το πρόβλημα σε υπό-προβλήματα και με την κατάλληλη μοντελοποίηση, μπορεί να αντιμετωπίσει το πρόβλημα της ανάθεσης εργασιών προσεγγίζοντας τη λύση σχετικά γρήγορα. Άλλη μια χρήσιμη τεχνική είναι ο δυναμικός προγραμματισμός (Dynamic Programming), ο οποίος μπορεί να μειώσει κατά πολύ την πολυπλοκότητα και κατά συνέπεια τον χρόνο εκτέλεσης των εργασιών. Όμως, απαιτεί αρκετό χώρο μνήμης για την εκτέλεσή του. Τέλος, υπάρχει και η κυρτή βελτιστοποίηση (Convex optimization), με την ελαχιστοποίηση ενός προβλήματος με ένα κυρτό (Convex) σύνολο. Με αυτήν την προσέγγιση, οι κατάλληλοι αλγόριθμοι μπορούν να λύσουν το πρόβλημα σε πολυωνυμικό χρόνο και να βρουν μια σχεδόν βέλτιστη λύση.

## 2.2 Επίλυση με προσέγγιση SDR

### 2.2.1 SDR προσέγγιση σε μεταφόρτωση στο Cloud με υπολογιστικό σημείο πρόσβασης

Οι συγγραφείς [14] ασχολούνται με ένα παρόμοιο σενάριο, με έναν χρήστη - κινητή συσκευή με πολλαπλές ανεξάρτητες εργασίες, ένα υπολογιστικό σημείο πρόσβασης και έναν απομακρυσμένο εξυπηρετητή cloud. Το σημείο πρόσβασης μπορεί είτε να επεξεργαστεί τις εργασίες είτε να τις προωθήσει στον εξυπηρετητή cloud. Στόχος είναι η βέλτιστη ανάθεση εργασιών ώστε να ελαχιστοποιείται το κόστος της ενέργειας, της εκτέλεσης του υπολογισμού και της χρονικής καθυστέρησης. Μοντελοποιούν το πρόβλημα ως ένα quadratic πρόβλημα βελτιστοποίησης με quadratic περιορισμούς, το οποίο είναι NP-hard. Στη συνέχεια, χρησιμοποιούν την προσέγγιση SDR και μια μέθοδο βασισμένη σε randomization αντιστοίχιση, παρόμοια με αυτήν που χρησιμοποιούμε και σε αυτήν την εργασία, για την επίλυση του προβλήματος.

Στις προσομοιώσεις τους, συγκρίνουν τα αποτελέσματα για μικρές και μεγάλες τιμές δεδομένων εισόδου και εξόδου. Επιπλέον, συγκρίνουν την προσέγγιση SDR με 1) την μέθοδο της τοπικής επεξεργασίας, κατά την οποία όλες οι εργασίες εκτελούνται στην κινητή συσκευή, 2) την μέθοδο της επεξεργασίας στο cloud, στην οποία έχουμε μεταφόρτωση όλων των εργασιών στο cloud, 3) μια μέθοδο κατά την οποία προωθούνται όλες οι εργασίες στο cloud, όταν δεν υπάρχουν ενδιάμεσα σημεία πρόσβασης και 4) την μέθοδο τυχαίας μεταφόρτωσης.

Τα αποτελέσματα καταλήγουν στο ότι η SDR προσέγγιση έχει σχεδόν βέλτιστη επίδοση σε σχέση με τις υπόλοιπες μεθόδους και η χρήση των σημείων πρόσβασης ανάμεσα στην κινητή συσκευή και στο cloud προσφέρει επιπλέον πλεονεκτήματα.

Όσον αφορά το σύστημα που μελετάμε σε αυτή την εργασία, χρησιμοποιούμε πολλαπλά σημεία πρόσβασης σε αντίθεση με την προσέγγιση του ενός σημείου πρόσβα-

σης. Επιπλέον, υπάρχουν διαφορές στη συνάρτηση αντιστοίχισης, η οποία επιστρέφει τα δυαδικά(binary) χαρακτηριστικά της λύσης του SDR, καθώς και στις παραμέτρους της κινητής συσκευής που χρησιμοποιείται στις προσομοιώσεις.

## 2.2.2 SDR προσέγγιση σε μεταφόρτωση στο Cloud με Fog Computing

Οι συγγραφείς [15] αντιμετωπίζουν το πρόβλημα της μεταφόρτωσης εργασιών σε δίκτυα fog. Χρησιμοποιούν μια παρόμοια δομή, αλλά αντί για τα σημεία πρόσβασης, έχουν πολλαπλούς fog κόμβους. Και πάλι η κινητή συσκευή έχει ορισμένες εργασίες, οι οποίες μπορούν μέσω των κόμβων αυτών να μεταφορτωθούν στο cloud για επεξεργασία. Σε αυτή την μελέτη λαμβάνονται υπόψη η μέση καθυστέρηση απάντησης και μεταφοράς των εργασιών. Βασικός στόχος είναι η ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης των εργασιών, αγνοώντας την κατανάλωση της ενέργειας. Η μοντελοποίηση του προβλήματος καταλήγει σε ένα quadratic πρόβλημα βελτιστοποίησης με quadratic περιορισμούς, το οποίο είναι NP-hard. Απορρίπτοντας τον περιορισμό για την τάξη του πίνακα ανάθεσης, μπορούν να χρησιμοποιήσουν τον SDR και να λύσουν το πρόβλημα σε πολυωνυμικό χρόνο.

Στις 10.000 προσομοιώσεις τους στο MATLAB συγκρίνουν τη συνολική χρονική καθυστέρηση σε σχέση με τον ρυθμό αφίξεως των δεδομένων στους κόμβους. Διερευνούν περιπτώσεις τοπικής επεξεργασίας, ολικής μεταφόρτωσης και μεταφόρτωσης με βάση τη λύση του SDR. Επιπλέον, συγκρίνουν για μεταφόρτωση από διαφορετικούς χρήστες στον ίδιο κόμβο ή σε διαφορετικούς. Υπάρχει και σύγκριση ανάμεσα σε ένα δίκτυο που αποτελείται από μόνο έναν κόμβο και ένα δίκτυο με πολλαπλούς κόμβους, όπου στις περισσότερες περιπτώσεις το μεγαλύτερο δίκτυο έχει γρηγορότερη συνολική εκτέλεση των εργασιών.

Τα αποτελέσματα των προσομοιώσεων ανέδειξαν την αποτελεσματικότητα της προτεινόμενης λύσης με τον SDR και το δίκτυο με τους πολλαπλούς κόμβους σε σχέση με εναλλακτικές λύσεις.

Στην μελέτη αυτή εξετάζεται το πρόβλημα της ανάθεσης εργασιών από την πλευρά της εξοικονόμησης χρόνου, σε αντίθεση με την προηγούμενη εργασία, η οποία είχε ως βασικότερα κριτήρια ελαχιστοποίησης την εξοικονόμηση χρόνου και ενέργειας.

## 2.3 Κριτήρια Επίδοσης στη μεταφόρτωση εργασιών

### 2.3.1 Κριτήρια επίδοσης στη βιβλιογραφία

Στη σχετική βιβλιογραφία, ο χρόνος εκτέλεσης και η κατανάλωση ενέργειας είναι οι δύο βασικές παράμετροι που κοιτάμε για την μεταφόρτωση εργασιών.

Όπως είδαμε και προηγουμένως, οι συγγραφείς [14] έχουν στόχο την ελαχιστοποίηση του κόστους ενέργειας και χρονικής καθυστέρησης. Ορίζουν ως συνολικό κόστος για τον χρήστη της κινητής συσκευής το άθροισμα με βάρη της συνολικής κατανάλωσης ενέργειας, του κόστους για τη μεταφόρτωση και την επεξεργασία όλων των εργασιών, λαμβάνοντας υπόψη την καθυστέρηση της μετάδοσης και της επεξεργασίας

στη χειρότερη περίπτωση. Σκοπός είναι η ελαχιστοποίηση του συνολικού κόστους της χειρότερης περίπτωσης επιλέγοντας όσο πιο αποδοτικά γίνεται την ανάθεση των εργασιών. Στη συνέχεια εξετάζουν το συνολικό κόστος αυτό, σε σχέση με διαφορετικές τιμές παραμέτρων όπως το μέγεθος των δεδομένων και τους ρυθμούς μετάδοσης.

Οι συγγραφείς [15] από την άλλη μοντελοποίησαν το πρόβλημα με βασικό κριτήριο την συνολική χρονική καθυστέρηση. Λαμβάνουν υπόψη την μέση καθυστέρηση απάντησης, η οποία περιλαμβάνει τον τοπικό χρόνο εκτέλεσης και την καθυστέρηση ουράς, και τον χρόνο μετάδοσης για τη μεταφόρτωση. Δεν ασχολούνται με την κατανάλωση ενέργειας και διατυπώνουν το πρόβλημα της ελαχιστοποίησης της συνολικής καθυστέρησης. Στις προσομοιώσεις εξετάζουν τη συνολική καθυστέρηση σε σχέση με τους ρυθμούς αφίξεως των εργασιών και τους ρυθμούς ανεβάσματος από τον χρήστη και από το fog στο cloud.

Στην μελέτη [16] οι συγγραφείς έχουν στόχο την ελαχιστοποίηση του κόστους ενέργειας και χρονικής καθυστέρησης. Εκτός από τα προηγούμενα, τρέχουν και μερικές επιπλέον προσομοιώσεις. Συγκρίνουν το ποσοστό των εργασιών που μεταφορτώνεται σε κάθε περίπτωση, για διαφορετικά μεγέθη δεδομένων και αλγόριθμους. Επιπλέον, εξετάζουν το ποσοστό προσέγγισης που έχει ο αλγόριθμος στην βέλτιστη λύση, για διαφορετικές τιμές των δειγμάτων του randomization. Τέλος, ερευνούν και τον συμβιβασμό ανάμεσα σε κατανάλωση ενέργειας και συνολική χρονική καθυστέρηση, δίνοντας διαφορετικές τιμές στα βάρη.

## Η τεχνική SDR

---

Στο κεφάλαιο αυτό γίνεται μια περιγραφή του αλγορίθμου SDR (semidefinite relaxation). Παρουσιάζεται η δομή του αλγορίθμου, τα βήματα εκτέλεσης του και οι εφαρμογές του.

### 3.1 Ο αλγόριθμος SDR αναλυτικά

#### 3.1.1 Παραδείγματα quadratic προβλημάτων

Ο SDR είναι μια πολύ ισχυρή, υπολογιστικά αποδοτική τεχνική προσέγγισης για δύσκολα προβλήματα βελτιστοποίησης [17]. Πιο συγκεκριμένα, μπορεί να εφαρμοστεί σε quadratic προβλήματα με quadratic περιορισμούς (QCQPs), όπως στο παρακάτω πρόβλημα:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T \mathbf{C} \mathbf{x} \quad (3.1\alpha')$$

$$\text{s.t. } \mathbf{x}^T \mathbf{F}_i \mathbf{x} \geq \mathbf{g}_i, \quad i = 1, \dots, p, \quad (3.1\beta')$$

$$\mathbf{x}^T \mathbf{H}_i \mathbf{x} = \mathbf{l}_i, \quad i = 1, \dots, q, \quad (3.1\gamma')$$

όπου οι δοσμένοι πίνακες  $\mathbf{C}$ ,  $\mathbf{F}_i$ ,  $\mathbf{H}_i$  υποθέτουμε ότι είναι πραγματικοί συμμετρικοί πίνακες. Η τάξη των QCQPs περιέχει πολλά προβλήματα σχετικά με σήματα και επικοινωνίες, όπως για παράδειγμα το Boolean quadratic πρόβλημα (BQP):

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T \mathbf{C} \mathbf{x} \quad (3.2\alpha')$$

$$\text{s.t. } x_i^2 = 1, \quad i = 1, \dots, n. \quad (3.2\beta')$$

Το πρόβλημα αυτό είναι γνωστό υπολογιστικά δύσκολο πρόβλημα και ανήκει στην τάξη των NP-hard προβλημάτων. Η δυνατότητα να λύνουμε τέτοιου είδους προβλήματα έχει μεγάλη σημασία σε πολλές καταστάσεις, για παράδειγμα η λύση του Boolean προβλήματος επηρεάζει σε μεγάλο βαθμό την ανίχνευση πολλαπλών εισόδων-εξόδων.

### 3.1.2 Η ιδέα του SDR

Έστω ένα ομογενές QCQP με πραγματικές τιμές:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \mathbf{x}^T \mathbf{C} \mathbf{x} \quad (3.3\alpha')$$

$$\text{s.t. } \mathbf{x}^T \mathbf{A}_i \mathbf{x} \geq_i \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.3\beta')$$

Το σύμβολο  $\geq_i$  μπορεί να είναι είτε " $\geq$ ", " $=$ " ή " $\leq$ " για κάθε  $i$ ,  $\mathbf{C}, \mathbf{A}_1, \dots, \mathbf{A}_m \in \mathbf{S}^n$ , όπου  $\mathbf{S}^n$  είναι το σύνολο όλων των πραγματικών συμμετρικών  $n \times n$  πινάκων,  $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbf{R}$ .

Παρατηρούμε πως

$$\mathbf{x}^T \mathbf{C} \mathbf{x} = \text{Tr}(\mathbf{x}^T \mathbf{C} \mathbf{x}) = \text{Tr}(\mathbf{C} \mathbf{x} \mathbf{x}^T) \quad (3.4)$$

$$\mathbf{x}^T \mathbf{A}_i \mathbf{x} = \text{Tr}(\mathbf{x}^T \mathbf{A}_i \mathbf{x}) = \text{Tr}(\mathbf{A}_i \mathbf{x} \mathbf{x}^T) \quad (3.5)$$

Και η συνάρτηση προς βελτιστοποίηση και οι περιορισμοί στο 3.3 είναι γραμμικοί ως προς τον πίνακα  $\mathbf{x} \mathbf{x}^T$ . Οπότε, με την νέα μεταβλητή  $\mathbf{X} = \mathbf{x} \mathbf{x}^T$ , σημειώνοντας ότι η  $\mathbf{X} = \mathbf{x} \mathbf{x}^T$  είναι ισοδύναμη με την  $\mathbf{X}$  να είναι ένας 1ης τάξης συμμετρικός θετικός semidefinite (PSD) πίνακας, έχουμε την ακόλουθη ισοδύναμη μορφή με το 3.3 :

$$\min_{\mathbf{X} \in \mathbf{S}^n} \text{Tr}(\mathbf{C} \mathbf{X}) \quad (3.6\alpha')$$

$$\text{s.t. } \text{Tr}(\mathbf{A}_i \mathbf{X}) \geq_i \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.6\beta')$$

$$\mathbf{X} \geq 0, \quad \text{rank}(\mathbf{X}) = 1. \quad (3.6\gamma')$$

Χρησιμοποιούμε  $\mathbf{X} \geq 0$  για να δείξουμε ότι η  $\mathbf{X}$  είναι PSD.

Η μοντελοποίηση του 3.6 μας βοηθάει να εντοπίσουμε την βασική δυσκολία στην επίλυση του 3.3. Στην πραγματικότητα, ο μόνος δύσκολος περιορισμός στο 3.6 είναι ο περιορισμός για την τάξη του πίνακα  $\text{rank}(\mathbf{X}) = 1$ , ο οποίος δεν είναι convex. Οπότε τον αγνοούμε και παίρνουμε την ακόλουθη προσέγγιση του 3.3:

$$\min_{\mathbf{X} \in \mathbf{S}^n} \text{Tr}(\mathbf{C} \mathbf{X}) \quad (3.7\alpha')$$

$$\text{s.t. } \text{Tr}(\mathbf{A}_i \mathbf{X}) \geq_i \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.7\beta')$$

$$\mathbf{X} \geq 0. \quad (3.7\gamma')$$

Το πρόβλημα 3.7 είναι γνωστό ως ένα SDR του 3.6, όπου το όνομα προκύπτει από το γεγονός ότι το 3.7 είναι μια μορφή semidefinite προγραμματισμού (SDP). Η μοντελοποίηση 3.7 μπορεί να λυθεί, με κάποια ακρίβεια, με έναν υπολογιστικά αξιόπιστο και αποδοτικό τρόπο. Τα προβλήματα SDR μπορούν να αντιμετωπιστούν αποτελεσματικά από διαθέσιμα πακέτα λογισμικού. Για παράδειγμα, έστω ότι " $\geq_i$ " είναι " $\geq$ " για  $i = 1, \dots, p$  και " $\geq_i$ " είναι " $=$ " για  $i = p + 1, \dots, m$ . Χρησιμοποιώντας το εργαλείο convex βελτι-



## Αλγόριθμος 3.1: CVX κώδικας για SDR

```

cvx_begin
    variable X(n,n) symmetric
    minimize(trace(C*X));
    subject to
        for i=1:p
            trace(A(:, :, i)*X) >= b(i);
        end
        for i=p+1:m
            trace(A(:, :, i)*X) == b(i);
        end
    X == semidefinite(n);
cvx_end

```

στοποίησης CVX μπορούμε να λύσουμε το 3.7 στο MATLAB με τον κώδικα που δίνεται παραπάνω ("CVX κώδικας για SDR") σύμφωνα με [17].

Τα περισσότερα εργαλεία για την αντιμετώπιση των προβλημάτων SDP χρησιμοποιούν έναν αλγόριθμο εσωτερικού σημείου (interior-point), ο οποίος επηρεάζει τον χρόνο εκτέλεσης του προγράμματος. Το SDR πρόβλημα 3.7 μπορεί να λυθεί με πολυπλοκότητα χειρότερης περίπτωσης ίση με  $\mathbf{O}(\max\{m, n\}^4 n^{1/2} \log(1/e))$ ,

για δοσμένη ακρίβεια λύσης  $e > 0$ .

Για συγκεκριμένα προβλήματα SDR μπορεί κάποιος να εκμεταλλευτεί την δομή του προβλήματος φτιάχνοντας γρήγορους αλγορίθμους εσωτερικού σημείου με συγκεκριμένα χαρακτηριστικά για την κάθε δομή. Γενικά, η πολυπλοκότητα του SDR αυξάνεται σιγά (λογαριθμικά) σε σχέση με το  $e$  και οι περισσότερες εφαρμογές δεν απαιτούν πολύ υψηλή ακρίβεια. Οπότε, μπορούμε να πούμε πως ο SDR είναι μια υπολογιστικά αποδοτική προσέγγιση για το QCQP, αφού έχει πολυωνυμική πολυπλοκότητα για πρόβλημα με μέγεθος  $n$  και  $m$  περιορισμούς.

Ένα δύσκολο κομμάτι στην τεχνική του SDR είναι η μετατροπή του NP-hard προβλήματος σε πρόβλημα που λύνεται σε πολυωνυμικό χρόνο. Ένα βασικό πρόβλημα είναι το πως θα μετατρέψουμε μια αποδοτική λύση  $\mathbf{X}^*$  για το 3.7 σε μία λύση  $\tilde{\mathbf{x}}$  για το 3.3. Αν η  $\mathbf{X}^*$  είναι πρώτης τάξης, τότε μπορούμε να γράψουμε  $\mathbf{X}^* = \mathbf{x}^* \mathbf{x}^{*T}$  και η  $\mathbf{x}^*$  θα είναι μια αποδοτική λύση για το 3.3. Από την άλλη, αν η τάξη της  $\mathbf{X}^*$  είναι μεγαλύτερη του ενός, τότε πρέπει να πάρουμε από αυτήν έναν πίνακα  $\tilde{\mathbf{x}}$ , ο οποίος να είναι μια λύση για το 3.3. Πρέπει να τονίσουμε ότι μια τέτοια λύση δεν είναι απαραίτητα μια βέλτιστη λύση για το πρόβλημα.

Ας δούμε την ιδέα της εφαρμογής μιας 1ης τάξης προσέγγισης στην  $\mathbf{X}^*$ . Έστω  $r = \text{rank}(\mathbf{X}^*)$  και

$$\mathbf{X}^* = \sum_{i=1}^r \lambda_i \mathbf{q}_i \mathbf{q}_i^T \quad (3.8)$$

συμβολίζει την ιδιοδιάσπαση της  $\mathbf{X}^*$ , όπου  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$  είναι οι ιδιοτι-

μές και  $q_1, \dots, q_r \in \mathbf{R}^n$  είναι τα αντίστοιχα ιδιοδιανύσματα. Αφού η καλύτερη 1ης τάξης προσέγγιση  $\mathbf{X}_1^*$  στην  $\mathbf{X}^*$  δίνεται ως  $\mathbf{X}_1^* = \lambda_1 q_1 q_1^T$ , μπορούμε να ορίσουμε  $\tilde{\mathbf{x}} = \sqrt{\lambda_1} q_1$  ως την υποψήφια λύση για το 3.3, αν είναι εφικτή. Αλλιώς, μπορούμε να προσπαθήσουμε να αντιστοιχίσουμε το  $\tilde{\mathbf{x}}$  σε μια "κοντινή" εφικτή λύση  $\hat{\mathbf{x}}$  του 3.3. Γενικά, μια τέτοια αντιστοίχιση εξαρτάται από το πρόβλημα. Για παράδειγμα, για το BQP (3.2) όπου  $x_i^2 = 1$  για κάθε  $i$ , μπορούμε να βρούμε μια εφικτή λύση από την  $\tilde{\mathbf{x}}$  μέσω  $\hat{\mathbf{x}} = \text{sgn}(\tilde{\mathbf{x}})$ , όπου  $\text{sgn}(\cdot)$  είναι η συνάρτηση προσήμου για το κάθε στοιχείο.

Πρέπει να τονίσουμε επίσης:

1) Σχετικά με την ποιότητα της λύσης  $\hat{\mathbf{x}}$  που βρήκαμε για το πρόβλημα. Υπάρχουν αρκετοί τρόποι να μετρηθεί η ακρίβεια της, αλλά εξαρτάται πάντα και από τη μέθοδο με την οποία καταλήγουμε στην  $\hat{\mathbf{x}}$ .

2) Εκτός από τον παραπάνω τρόπο με την χαλάρωση της τάξης του πίνακα που χρησιμοποιήσαμε, υπάρχει και μια άλλη τεχνική για τον SDR μέσω της Λαγκρανζιανής δυαδικότητας. Πιο συγκεκριμένα, μπορεί να αποδειχθεί ότι το SDR 3.7 είναι ένα Λαγκρανζιανό δυαδικό ενός δυαδικού (bidual) του αρχικού προβλήματος 3.3 σύμφωνα με [18].

### 3.1.3 Τεχνική Randomization

Εκτός από την παραπάνω μέθοδο προσέγγισης για τον SDR, υπάρχει και η μέθοδος randomization για την εξαγωγή μιας προσεγγιστικής λύσης QCQP από μια SDR λύση  $\mathbf{X}^*$ . Θεωρούμε ξανά το ομογενές QCQP με πραγματικές τιμές:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \quad \mathbf{x}^T \mathbf{C} \mathbf{x} \quad (3.9\alpha')$$

$$\text{s.t.} \quad \mathbf{x}^T \mathbf{A}_i \mathbf{x} \geq \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.9\beta')$$

Τώρα, έστω  $\mathbf{X} \in \mathbf{S}^n$  ένας συμμετρικός θετικός semidefinite πίνακας. Θεωρούμε έναν τυχαίο πίνακα  $\xi \in \mathbf{R}^n$  σχηματισμένο σύμφωνα με την γκαουσιανή κατανομή με μηδενική μέση τιμή και διασπορά  $\mathbf{X}$ , ή  $\xi^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{X})$ . Η ιδέα χρήσης του randomization προκύπτει αν θεωρήσουμε το ακόλουθο στοχαστικό QCQP:

$$\min_{\mathbf{x} \in \mathbf{S}^n, \mathbf{x} \geq \mathbf{0}} \quad E_{\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{X})} \{ \xi^T \mathbf{C} \xi \} \quad (3.10\alpha')$$

$$\text{s.t.} \quad E_{\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{X})} \{ \xi^T \mathbf{A}_i \xi \} \geq \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.10\beta')$$

όπου αλλάζουμε την διασπορά του πίνακα  $\xi$  έτσι ώστε η αναμενόμενη τιμή της quadratic συνάρτησης για ελαχιστοποίηση ελαχιστοποιείται και οι quadratic περιορισμοί ικανοποιούνται. Μέσα από τη σχέση  $\mathbf{X} = E_{\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{X})} \{ \xi \xi^T \}$ , μπορούμε να δούμε ότι το στοχαστικό QCQP 3.10 είναι ισοδύναμο με το SDR

---

 Αλγόριθμος 3.2: Γκαουσιανή randomization για το BQP
 

---

given an SDR solution  $\mathbf{X}^*$ , and a number of randomizations  $L$   
 for  $l=1, \dots, L$   
     generate  $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{X}^*)$  and construct a QCQP –  
     feasible point  
          $\tilde{\mathbf{x}}_l = \text{sgn}(\xi_l)$ .  
 end  
 determine  $l^* = \text{argmin}_{l=1, \dots, L} \tilde{\mathbf{x}}_l^T \mathbf{C} \tilde{\mathbf{x}}_l$ .  
 output  $\hat{\mathbf{x}} = \tilde{\mathbf{x}}_{l^*}$  as the approximate QCQP solution.

---

$$\min_{\mathbf{x} \in \mathbf{S}^n, \mathbf{x} \geq \mathbf{0}} \text{Tr}(\mathbf{C}\mathbf{x}) \quad (3.11\alpha')$$

$$\text{s.t. } \text{Tr}(\mathbf{A}_i \mathbf{x}) \geq_i \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.11\beta')$$

Οπότε, η στοχαστική QCQP μορφή του SDR 3.10 μας προσφέρει έναν εναλλακτικό τρόπο να δημιουργούμε προσεγγιστικές λύσεις για το QCQP. Πράγματι, αφού βρούμε μια βέλτιστη λύση  $\mathbf{X}^*$  για το SDP 3.11 μπορούμε να παράγουμε έναν τυχαίο πίνακα  $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{X}^*)$  και να τον χρησιμοποιήσουμε για να κατασκευάσουμε μια προσεγγιστική λύση για το QCQP. Σημειώνουμε ότι ο συγκεκριμένος σχεδιασμός της διαδικασίας randomization εξαρτάται από το εκάστοτε πρόβλημα.

Για παράδειγμα στο πρόβλημα BQP 3.2 η διαδικασία randomization είναι αυτή που περιγράφεται παραπάνω στο "Γκαουσιανή randomization για το BQP", σύμφωνα με [17].

### 3.1.4 0 SDR σε μη ομογενή QCQP

Ως τώρα έχουμε δει την τεχνική του SDR σε ομογενή QCQP. Σε περίπτωση που έχουμε ένα μη ομογενές πρόβλημα να λύσουμε, μετατρέπουμε το πρόβλημα σε ομογενές QCQP και στη συνέχεια εφαρμόζουμε τις τεχνικές που αναφέραμε παραπάνω.

Έστω το μη ομογενές QCQP

$$\min_{\mathbf{x} \in \mathbf{R}^n} \mathbf{x}^T \mathbf{C} \mathbf{x} + 2\mathbf{c}^T \mathbf{x} \quad (3.12\alpha')$$

$$\text{s.t. } \mathbf{x}^T \mathbf{A}_i \mathbf{x} + 2\mathbf{a}_i^T \mathbf{x} \geq_i \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.12\beta')$$

για κάποια κατάλληλα  $\mathbf{C}, \mathbf{c}, \mathbf{A}_i, \mathbf{a}_i, \mathbf{b}_i$ . Ομογενοποιούμε το σύστημα ως εξής:

$$\min_{\mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R}} \begin{bmatrix} \mathbf{x}^T & t \end{bmatrix} \begin{bmatrix} \mathbf{C} & \mathbf{c} \\ \mathbf{c}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \quad (3.13\alpha')$$

$$\text{s.t. } t^2 = 1, \quad (3.13\beta')$$

$$\begin{bmatrix} \mathbf{x}^T & t \end{bmatrix} \begin{bmatrix} \mathbf{A}_i & \mathbf{a}_i \\ \mathbf{a}_i^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \geq_i \mathbf{b}_i, \quad i = 1, \dots, m. \quad (3.13\gamma')$$

όπου το μέγεθος του προβλήματος και ο αριθμός των περιορισμών αυξάνονται κατά ένα. Οπότε, ο SDR μπορεί να εφαρμοστεί σε μη ομογενή QCQP στις ομογενοποιημένες μορφές τους.

Μέρος **II**

## Μοντελοποίηση, Αντιμετώπιση και Προ- σομοιώσεις

---



## Ορισμός του Προβλήματος και προσέγγιση με SDR

---

Στο κεφάλαιο αυτό παρουσιάζεται η μελέτη που έγινε για την υλοποίηση του συστήματος. Αρχικά περιγράφεται η μοντελοποίηση του συστήματος και ορίζεται το πρόβλημα της βελτιστοποίησης για την ανάθεση εργασιών. Στη συνέχεια, προτείνεται μια λύση με τη βοήθεια του SDR αλγορίθμου.

### 4.1 Μοντελοποίηση του συστήματος και Διατύπωση Προβλήματος

Στην ενότητα αυτή παρουσιάζεται η μοντελοποίηση του συστήματος και το πρόβλημα της βελτιστοποίησης που προκύπτει για την ανάθεση των εργασιών του συστήματος.

#### 4.1.1 Μοντελοποίηση του συστήματος

Θεωρούμε μια κινητή συσκευή με  $N$  ανεξάρτητες εργασίες. Κάθε μια από αυτές τις εργασίες μπορεί να μεταφορτωθεί σε μία CPU οποιουδήποτε από τα  $M$  ασύρματα σημεία πρόσβασης ή να την επεξεργαστεί τοπικά η CPU της κινητής συσκευής. Το κάθε σημείο πρόσβασης μπορεί να μοντελοποιηθεί με βάση έναν Fog-Radio Access Network (F-RAN) κόμβο σύμφωνα με [19], ώστε εκτός από τη δυνατότητα για επικοινωνία, να κατέχει και την δυνατότητα για εκτέλεση εργασιών. Ορίζουμε το σύνολο των εργασιών ως  $N=1, \dots, N$  και το σύνολο των CPU ως  $M=0, \dots, M$ , όπου για  $M=0$  αντιστοιχεί στη CPU της κινητής συσκευής. Στη συνέχεια, θεωρούμε σημεία πρόσβασης τα οποία χρησιμοποιούν παράλληλα μεταξύ τους κανάλια, ώστε σήματα από διαφορετικά σημεία πρόσβασης να μην παρεμβάλλονται. Η κινητή συσκευή πρέπει να αποφασίσει αν οι εργασίες θα εκτελεστούν τοπικά ή από απόσταση.

Θεωρούμε την τριάδα τιμών  $\{a_i, b_i, w_i\}$  για την αναπαράσταση της εργασίας  $i$ , για κάθε  $i \in N$ , όπου  $a_i$  είναι το μέγεθος των δεδομένων εισόδου σε bits,  $b_i$  είναι το μέγεθος των δεδομένων εξόδου σε bits και  $w_i$  είναι ο αριθμός των κύκλων της CPU που απαιτείται για την εκτέλεση της εργασίας. Στη συνέχεια θεωρούμε πως ο αριθμός των

δεδομένων για επεξεργασία είναι γνωστός εκ των προτέρων. Επιπλέον,  $N$  ανεξάρτητες εργασίες μοιράζονται σε  $M+1$  ανεξάρτητα σύνολα, με το κάθε σύνολο να θεωρείται ως μία παρτίδα. Κάθε σημείο πρόσβασης έχει σταθερό ρυθμό εξυπηρέτησης  $r_k$  (κύκλοι ανά δευτερόλεπτο) απέναντι στην κινητή συσκευή. Οι ρυθμοί ανεβάσματος (uplink) και κατεβάσματος (downlink) ανάμεσα στον χρήστη και τα σημεία πρόσβασης υπολογίζονται ως εξής:

$$C_k^{UL} = B_k^{UL} \log_2 \left( 1 + \frac{P^{Tx} h_{ik}^{UL}}{w_0} \right)$$

$$C_k^{DL} = B_k^{DL} \log_2 \left( 1 + \frac{P_{AP} h_{ik}^{DL}}{w_0} \right)$$

όπου  $B_k^{UL}$  και  $B_k^{DL}$  είναι το εύρος ζώνης των καναλιών που χρησιμοποιούνται ανάμεσα στο σημείο πρόσβασης  $k$  και στην κινητή συσκευή για uplink και downlink αντίστοιχα.  $P^{Tx}$  και  $P_{AP}$  είναι οι ισχύεις των μεταδόσεων της κινητής συσκευής και των σημείων πρόσβασης αντίστοιχα.  $w_0$  είναι το επίπεδο ισχύος του white noise. Όταν ο χρόνος συνοχής των καναλιών είναι συγκρίσιμος με την διάρκεια της μεταφόρτωσης, τα  $h_{ik}^{UL}$  και  $h_{ik}^{DL}$  μπορούν να θεωρηθούν σταθερά. Γενικά, θεωρούμε πως τα σημεία πρόσβασης έχουν πλήρη γνώση σχετικά με την κατάσταση των καναλιών. Η κινητή συσκευή μπορεί να αποκτήσει αυτές τις τιμές από τα σημεία πρόσβασης και με αυτόν τον τρόπο τα  $C_k^{UL}$  και  $C_k^{DL}$  είναι γνωστά στην κινητή συσκευή. Από την άλλη, αν η διάρκεια της μεταφόρτωσης είναι πολύ μεγαλύτερη από τον χρόνο συνοχής, οι τιμές των  $h_{ik}^{UL}$  και  $h_{ik}^{DL}$  ποικίλουν, με αποτέλεσμα τα  $C_k^{UL}$  και  $C_k^{DL}$  να αλλάζουν σχετικά με τη διάρκεια. Οπότε, υποθέτουμε ότι οι τιμές των  $C_k^{UL}$  και  $C_k^{DL}$  καθ' όλη τη διάρκεια προσεγγίζονται από τις μέσες τιμές τους. Για τοπική επεξεργασία, δεν υπάρχει καθυστέρηση ανεβάσματος και κατεβάσματος στη μετάδοση, καθώς  $C_0^{UL} = \infty$  και  $C_0^{DL} = \infty$ . Χρησιμοποιούμε έναν δείκτη  $x_{ik}$ ,  $\forall i \in N, \forall k \in M$ , για την αναπαράσταση της ανάθεσης των εργασιών:

$$x_{ik} = \begin{cases} 1, & \text{αν η εργασία } i \text{ αναθέτεται στην CPU } k \\ 0, & \text{αλλιώς} \end{cases} \quad (4.1)$$

Ορίζουμε ως  $\mathbf{X} = \{x_{ik}\} \in \{0, 1\}^{N \times (M+1)}$  τον πίνακα ανάθεσης εργασιών και  $\mathbf{x} = [\mathbf{x}_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_k^T, \dots, \mathbf{x}_M^T]^T$ , όπου  $\mathbf{x}_k = [x_{1k}, x_{2k}, \dots, x_{Nk}]^T$ , ο πίνακας στηλών αντίστοιχα με τον  $\mathbf{X}$ .

Ορίζουμε ως  $d_{ik}^{UL}$ ,  $d_{ik}^{DL}$  και  $d_{ik}^{Comp}$  τη χρονική καθυστέρηση για το ανέβασμα, το κατέβασμα και τον υπολογισμό, αντίστοιχα, όταν η εργασία  $i$  ανατίθεται στην CPU  $k$ , όπου  $d_{ik}^{UL} = \frac{a_i}{C_k^{UL}}$ ,  $d_{ik}^{DL} = \frac{b_i}{C_k^{DL}}$  και  $d_{ik}^{Comp} = \frac{w_i}{r_k}$ . Οπότε, η συνολική καθυστέρηση για το upload  $T_k^{UL}$ , το download  $T_k^{DL}$  και τον υπολογισμό  $T_k^{Comp}$  για την μεταφόρτωση στην CPU  $k$  υπολογίζονται ως  $T_k^{UL} = \frac{\sum_{i \in N} x_{ik} a_i}{C_k^{UL}}$ ,  $T_k^{DL} = \frac{\sum_{i \in N} x_{ik} b_i}{C_k^{DL}}$  και  $T_k^{Comp} = \frac{\sum_{i \in N} x_{ik} w_i}{r_k}$ , αντίστοιχα. Κάθε σημείο πρόσβασης  $k \in M$  δέχεται και επεξεργάζεται τις αντίστοιχες εργασίες και επιστρέφει το αποτέλεσμα στην κινητή συσκευή. Ένα σημείο πρόσβασης μπορεί να επεξεργαστεί τις εργασίες αφού τις λάβει όλες ή να επεξεργαστεί κάποιες από τις πρώτες, ενώ λαμβάνει και τις υπόλοιπες. Η δεύτερη περίπτωση δυσκολεύει αρκετά την ανάλυση, καθώς η σειρά άφιξης των εργασιών στα σημεία πρόσβασης αυξάνει κατά πολύ το πεδίο των πιθανών λύσεων. Για χάρη απλότητας, υποθέτουμε πως τα σημεία πρόσβασης επεξεργάζονται τις εργασίες αφού τις λάβουν όλες. Τότε η χρονική καθυστέρηση για τις εργασίες που μεταφορτώνονται στην CPU  $k$ ,  $\forall k \in M$  δίνεται ως:



$$T_k = T_k^{UL} + T_k^{DL} + T_k^{Comp} = \sum_{i \in \mathcal{N}} x_{ik} \left( \frac{a_i}{C_k^{UL}} + \frac{w_i}{r_k} + \frac{b_i}{C_k^{DL}} \right) = \sum_{i \in \mathcal{N}} x_{ik} D_{ik},$$

όπου  $D_{ik} = \frac{a_i}{C_k^{UL}} + \frac{w_i}{r_k} + \frac{b_i}{C_k^{DL}}$ . Η κατανάλωση ενέργειας από την κινητή συσκευή προέρχεται κυρίως από υπολογιστική επεξεργασία και ασύρματη μετάδοση.

**Πίνακας 4.1:** Μεταβλητές που χρησιμοποιούνται σε αυτή την εργασία

Μεταβλητή	Ορισμός
$i$	index εργασίας
$k$	index σημείου πρόσβασης ή CPU κινητής συσκευής.
$a_i$	μέγεθος δεδομένων input για εργασία $i$
$b_i$	μέγεθος δεδομένων output για εργασία $i$
$w_i$	απαιτούμενος αριθμός κύκλων CPU για την επεξεργασία της εργασίας $i$
$\lambda_t$	βάρος σχετικά με την χρονική καθυστέρηση της εκτέλεσης
$\lambda_e$	βάρος σχετικά με την κατανάλωση ενέργειας της κινητής συσκευής
$C_k^{UL}, C_k^{DL}$	ρυθμός δεδομένων uplink, downlink ανάμεσα στην κινητή συσκευή και το σημείο πρόσβασης $k$
$D_{ik}$	χρονική καθυστέρηση όταν η εργασία $i$ μεταφορτώνεται στην CPU $k$
$r_k$	ρυθμός εξυπηρέτησης της CPU του σημείου πρόσβασης $k$
$r_0$	συχνότητα CPU της κινητής συσκευής
$\rho$	σταθερά για την συχνότητα της CPU σχετικά με την ενέργεια της υπολογιστικής επεξεργασίας
$\rho^{Comp}$	υπολογιστική ισχύς της κινητής συσκευής
$\rho^{Tx}, \rho^{Rx}$	ισχύς μετάδοσης και υποδοχής της κινητής συσκευής
$E^{Comp}$	υπολογιστική κατανάλωση ενέργειας της κινητής συσκευής
$E^{TR}$	κατανάλωση ενέργειας της κινητής συσκευής για την μετάδοση
$x_{ik}$	μεταβλητή απόφασης για την ανάθεση της εργασίας $i$ στην CPU $k$
$\mathbf{X}$	πίνακας ανάθεσης εργασιών
$t(\mathbf{X}, r_0)$	χρονική καθυστέρηση των εργασιών
$e(\mathbf{X}, r_0)$	κατανάλωση ενέργειας της κινητής συσκευής
$r_{min}, r_{max}$	ελάχιστη και μέγιστη συχνότητα CPU της κινητής συσκευής

#### 4.1.2 Ορισμός του προβλήματος ελαχιστοποίησης

1) Υπολογιστική Κατανάλωση Ενέργειας: Η υπολογιστική κατανάλωση ενέργειας της κινητής συσκευής ορίζεται ως

$$E^{comp} = \rho r_0^\zeta \sum_{i \in \mathcal{N}} x_{i0} D_{i0},$$

όπου  $\rho r_0^\zeta$  είναι η υπολογιστική ισχύς της κινητής συσκευής.  $\rho$  είναι μία σταθερά η οποία εξαρτάται από την χωρητικότητα της συσκευής. Η  $\zeta$  είναι σταθερά ( $\zeta \geq 2$ ), συνήθως κοντά στο 3. Εδώ, θέτουμε  $\zeta = 3$  [20].

2) Κατανάλωση Ενέργειας Λόγω Ασύρματης Μετάδοσης: η κατανάλωση ενέργειας λόγω ασύρματης μετάδοσης της κινητής συσκευής ορίζεται ως:

$$E^{TR} = \rho^{Tx} \sum_{k \in \mathcal{M} \setminus \{0\}} \sum_{i \in \mathcal{N}} x_{ik} d_{ik}^{UL} + \rho^{Rx} \sum_{k \in \mathcal{M} \setminus \{0\}} \sum_{i \in \mathcal{N}} x_{ik} d_{ik}^{DL},$$

όπου  $P^{Tx}$  και  $P^{Rx}$  είναι οι ισχείς της μετάδοσης και της αποδοχής, αντίστοιχα, οι οποίες θεωρούνται σταθερές.

Οι δύο βασικοί στόχοι της κινητής συσκευής είναι να ελαχιστοποιήσει 1) την χρονική καθυστέρηση της εκτέλεσης των εργασιών και 2) την κατανάλωση ενέργειας.

1) Καθυστέρηση Εκτέλεσης: Αφού τα σημεία πρόσβασης επεξεργάζονται τις εργασίες παράλληλα, η καθυστέρηση εκτέλεσης δίνεται ως:

$$t(\mathbf{X}, r_0) = \max_{k \in \mathcal{M}} T_k.$$

2) Συνολική Κατανάλωση Ενέργειας: Η συνολική κατανάλωση ενέργειας δίνεται ως:

$$e(\mathbf{X}, r_0) = E^{comp} + E^{TR}.$$

Και οι δύο αυτοί στόχοι εξαρτώνται από το ζεύγος  $(\mathbf{X}, r_0)$ , οπότε δεν μπορούν να ελαχιστοποιηθούν ανεξάρτητα και ταυτόχρονα. Στη συνέχεια, διερευνούμε τον συμβιβασμό ανάμεσα στους δύο στόχους και ορίζουμε την κοινή συνάρτηση κόστους ως εξής:

$$\psi(\mathbf{X}, r_0) = \lambda_t t(\mathbf{X}, r_0) + \lambda_e e(\mathbf{X}, r_0) \quad (4.2)$$

όπου τα  $\lambda_t, \lambda_e \in [0, 1]$  είναι βάρη. Η εξίσωση 4.2 μπορεί να θεωρηθεί ως ένα άθροισμα με βάρη, το οποίο προσεγγίζει ένα γενικό πρόβλημα βελτιστοποίησης με πολλαπλά κριτήρια. Σύμφωνα με [21], με θετικά βάρη, η ελαχιστοποίηση του προβλήματος  $\mathcal{E}1$  οδηγεί σε μία αποδοτική λύση για το πρόβλημα ελαχιστοποίησης με πολλαπλά κριτήρια. Αν κάποιο από τα βάρη είναι 0, τότε οδηγεί σε μία αδύναμη αποδοτική λύση. Η προσέγγιση με το άθροισμα με τα βάρη χρησιμοποιείται ευρέως, επειδή είναι απλή στην κατανόηση και εύκολη στην εφαρμογή. Επιπλέον, τα βάρη δείχνουν την σχετική σημασία ανάμεσα στην ενέργεια και στην χρονική καθυστέρηση. Για παράδειγμα, όταν η μπαταρία της κινητής συσκευής είναι χαμηλή, η κινητή συσκευή μπορεί να θέσει  $\lambda_t = 0$  και  $\lambda_e = 1$ . Με αυτόν τον τρόπο, το πρόβλημα της βελτιστοποίησης διατυπώνεται ως εξής:

$$\mathcal{E}1 : \min_{\mathbf{X}, r_0} \psi(\mathbf{X}, r_0), \quad (4.3\alpha')$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{M}} x_{ik} = 1, \forall i \in \mathcal{N}, \quad (4.3\beta')$$

$$r_0 \in [r_{min}, r_{max}], \quad (4.3\gamma')$$

$$x_{ik} \in \{0, 1\} \quad (4.3\delta')$$

Συμβολίζουμε το σύνολο τιμών του προβλήματος  $\mathcal{E}1$  ως  $\mathcal{X}$ . Ο περιορισμός 4.3β' εγγυάται ότι μία εργασία μπορεί να ανατεθεί μόνο σε ένα σημείο πρόσβασης. Το πρόβλημα αυτό είναι NP-hard, όπως αναφέρεται στο Proposition 1 [16].

## 4.2 Βελτιστοποίηση της ανάθεσης εργασιών με σταθερή συχνότητα CPU

Στη συνέχεια, εφαρμόζουμε μια δομή βελτιστοποίησης για τον υπολογισμό μιας αποδοτικής ανάθεσης εργασιών, η οποία ελαχιστοποιεί το άθροισμα με βάρη της χρονικής καθυστέρησης και της κατανάλωσης ενέργειας στην κινητή συσκευή, όταν η συχνότητα της CPU είναι σταθερή. Με σταθερή συχνότητα, οι  $D_{i0}$  και  $P^{Comp} = \rho r_0^\zeta$  γίνονται σταθερές. Τότε, η συνάρτηση κόστους 4.2 απλοποιείται ως εξής:

1) Το χρονικό διάστημα  $t$  δίνεται ως:

$$t(\mathbf{X}) = \max_{k \in \mathcal{M}} (\sum_{i \in \mathcal{N}} x_{ik} D_{ik}).$$

2) Η συνολική κατανάλωση ενέργειας απλοποιείται ως:

$$e(\mathbf{X}) = P^{Comp} \sum_{i \in \mathcal{N}} x_{i0} D_{i0} + P^{Tx} \sum_{k \in \mathcal{M} \setminus \{0\}} \sum_{i \in \mathcal{N}} x_{ik} d_{ik}^{UL} + P^{Rx} \sum_{k \in \mathcal{M} \setminus \{0\}} \sum_{i \in \mathcal{N}} x_{ik} d_{ik}^{DL}.$$

Με αυτόν τον τρόπο, η δομή βελτιστοποίησης για σταθερή συχνότητα CPU μπορεί να γραφτεί ως:

$$\mathcal{F}1 : \min_{\mathbf{X}} \lambda_t t(\mathbf{X}) + \lambda_e e(\mathbf{X}), \quad (4.4\alpha')$$

$$\text{s.t.} \quad \sum_{k=0}^{\mathcal{M}} x_{ik} = 1, \forall i \in \mathcal{N}, \quad (4.4\beta')$$

$$x_{ik} \in \{0, 1\} \quad (4.4\gamma')$$

Σύμφωνα με την Proposition 1 ([16]) το πρόβλημα  $\mathcal{F}1$  είναι NP-hard.

Για τον υπολογισμό της πιο αποδοτικής λύσης του  $\mathcal{F}1$ , χρησιμοποιούμε μια νέα μεταβλητή  $t$  με επιπλέον περιορισμό

$t \geq \max_{k \in \mathcal{M}} (\sum_{i \in \mathcal{N}} x_{ik} D_{ik})$  και έχουμε το πρόβλημα  $\mathcal{F}2$ :

$$\mathcal{F}2 : \min_{\mathbf{X}, t} \lambda_t t + \lambda_e P^{Comp} \sum_{i \in \mathcal{N}} x_{i0} D_{i0} + \lambda_e P^{Tx} \sum_{k \in \mathcal{M} \setminus \{0\}} \sum_{i \in \mathcal{N}} x_{ik} d_{ik}^{UL} + \lambda_e P^{Rx} \sum_{k \in \mathcal{M} \setminus \{0\}} \sum_{i \in \mathcal{N}} x_{ik} d_{ik}^{DL}, \quad (4.5\alpha')$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{N}} x_{ik} D_{ik} \leq t, \forall k \in \mathcal{M}, \quad (4.5\beta')$$

$$\sum_{k \in \mathcal{M}} x_{ik} = 1, \forall i \in \mathcal{N} \quad (4.5\gamma')$$

$$x_{ik} \in \{0, 1\} \quad (4.5\delta')$$

Το  $\mathcal{F}2$  είναι mixed-integer linear programming (MILP) πρόβλημα. Η αποδοτική λύση του μπορεί να βρεθεί με τον αλγόριθμο Branch-and-Bound (BnB), αλλά ο αλγόριθμος αυτός έχει μεγάλη πολυπλοκότητα, ειδικά για μεγάλες διαστάσεις του  $\mathcal{X}$ . Για αυτό το λόγο, προτείνουμε την πιο αποδοτική προσέγγιση με τον αλγόριθμο SDR.

### 4.3 Προσέγγιση με τον αλγόριθμο SDR

Σε αυτό το σημείο προτείνουμε την λύση του  $\mathcal{F}2$  με τον αλγόριθμο SDR.

Με βάση το [22], το πρόβλημα  $\mathcal{F}2$  μετατρέπεται στο αντίστοιχο ομογενές quadratic constrained quadratic programming (QCQP). Αγνοώντας τον περιορισμό σχετικά με την τάξη του πίνακα, ένα ομογενές (QCQP) πρόβλημα γίνεται Semidefinite programming πρόβλημα (SDP), το οποίο μπορεί να λυθεί στην χειρότερη περίπτωση με πολυπλοκότητα  $O(\max\{m^4, n^4\}n^{1/2}\log(1/e))$ , όπου  $n$  είναι η διάσταση του συμμετρικού πίνακα  $\mathbf{Z}$ ,  $m$  είναι ο αριθμός των περιορισμών και  $e > 0$  είναι η ακρίβεια της λύσης. Οπότε, οι δυαδικοί περιορισμοί αντικαθιστούνται με quadratic περιορισμούς :

$$x_{ik}(1 - x_{ik}) = 0 \quad \forall i, k \Leftrightarrow x_{ik} \in \{0, 1\} \quad (4.6)$$

Ορίζουμε  $\mathbf{y} = [\mathbf{x}^T, t]^T$ ,  $\mathbf{D}_k = [D_{1k}, \dots, D_{Nk}]^T$ ,  $\forall k \in \mathcal{M}$  και  $\mathbf{d}_k^{UL} = [d_{1k}^{UL}, \dots, d_{Nk}^{UL}]^T$ ,  $\forall k \in \mathcal{M} \setminus \{0\}$ .

Τότε το πρόβλημα  $\mathcal{F}2$  μετατρέπεται στο :

$$\mathcal{F}3 : \min_{\mathbf{y}} \mathbf{b}_0^T \mathbf{y} \quad (4.7\alpha')$$

$$\text{s.t. } \mathbf{A}1\mathbf{y} \leq \mathbf{0}_{(M+1 \times 1)} \quad (4.7\beta')$$

$$\mathbf{A}2\mathbf{y} = \mathbf{1}_{(N \times 1)} \quad (4.7\gamma')$$

$$\mathbf{y}^T \text{diag}(\mathbf{u}_p)\mathbf{y} - \mathbf{u}_p^T \mathbf{y} = 0, \quad p = 1, \dots, NM + N, \quad (4.7\delta')$$

όπου  $\mathbf{u}_p$  είναι πίνακας διαστάσεων  $(NM + N + 1) \times 1$  με το στοιχείο στη θέση  $p$  να είναι 1. Ο  $\text{diag}(\mathbf{u}_p)$  είναι ένας διαγώνιος πίνακας, με τα διαγώνια στοιχεία του να ξεκινούν από την επάνω αριστερή γωνία και να είναι τα στοιχεία του  $\mathbf{u}_p$ . Οι τρεις περιορισμοί 4.7β', 4.7γ', 4.7δ' είναι ισοδύναμοι με τους αντίστοιχους 4.5β', 4.5γ', 4.5δ'. Ο 4.7δ' είναι ισοδύναμος με την σχέση 4.6.

Ορίζοντας  $\mathbf{Z} = \begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix}$  και  $Q = MN + N + 1$ , τότε το  $\mathcal{F}3$  ομογενοποιείται και μετατρέπεται στο:

$$\mathcal{F}3 - 1 : \min_{\mathbf{Z}} \text{Tr}(\mathbf{B}_0\mathbf{Z}) \quad (4.8\alpha')$$

$$\text{s.t. } \text{Tr}(\mathbf{H}_h\mathbf{Z}) \leq 0, \quad h = 1, \dots, M + 1, \quad (4.8\beta')$$

$$\text{Tr}(\mathbf{J}_j\mathbf{Z}) = 1, \quad j = 1, \dots, N, \quad (4.8\gamma')$$

$$\text{Tr}(\mathbf{G}_p\mathbf{Z}) = 0, \quad p = 1, \dots, NM + N, \quad (4.8\delta')$$

$$\mathbf{Z} \geq 0, \quad \text{rank}(\mathbf{Z}) = 1, \quad z_{Q+1, Q+1} = 1 \quad (4.8\epsilon')$$

όπου

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{0}_{Q \times Q} & \frac{1}{2} \mathbf{b}_0 \\ \frac{1}{2} \mathbf{b}_0^T & \mathbf{0} \end{bmatrix}, \quad \mathbf{H}_h = \begin{bmatrix} \mathbf{0}_{Q \times Q} & \frac{1}{2} \mathbf{A}_{1,h}^T \\ \frac{1}{2} \mathbf{A}_{1,h} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{J}_j = \begin{bmatrix} \mathbf{0}_{Q \times Q} & \frac{1}{2} \mathbf{A}_{2,j}^T \\ \frac{1}{2} \mathbf{A}_{2,j} & \mathbf{0} \end{bmatrix}, \quad \mathbf{G}_p = \begin{bmatrix} \text{diag}(\mathbf{u}_p) & -\frac{1}{2} \mathbf{u}_p \\ -\frac{1}{2} \mathbf{u}_p & \mathbf{0} \end{bmatrix},$$

$$\mathbf{b}_0 = \left[ \lambda_e \rho^{Comp} \mathbf{D}_0^T, \lambda_e \mathbf{b}'_0, \lambda_t \right]^T,$$

$$\mathbf{b}'_0 = \rho^{Tx} \left[ \mathbf{d}_1^{UL^T}, \mathbf{d}_2^{UL^T}, \dots, \mathbf{d}_M^{UL^T} \right]^T \\ + \rho^{Rx} \left[ \mathbf{d}_1^{DL^T}, \mathbf{d}_2^{DL^T}, \dots, \mathbf{d}_M^{DL^T} \right]^T,$$

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{D}_0^T & \mathbf{0}_{1 \times N} & \mathbf{0}_{1 \times N} & \dots & \mathbf{0}_{1 \times N} & -1 \\ \mathbf{0}_{1 \times N} & \mathbf{D}_1^T & \mathbf{0}_{1 \times N} & \dots & \mathbf{0}_{1 \times N} & -1 \\ \mathbf{0}_{1 \times N} & \mathbf{0}_{1 \times N} & \mathbf{D}_2^T & \dots & \mathbf{0}_{1 \times N} & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{1 \times N} & \mathbf{0}_{1 \times N} & \mathbf{0}_{1 \times N} & \dots & \mathbf{D}_M^T & -1 \end{bmatrix},$$

$$\mathbf{A}_2 = \left[ \mathbf{I}_{N,0}, \mathbf{I}_{N,1}, \mathbf{I}_{N,2}, \dots, \mathbf{I}_{N,k}, \dots, \mathbf{I}_{N,M}, \mathbf{0}_{N \times 1} \right],$$

όπου  $\mathbf{I}_{N,k}$  είναι ένας  $N$ -διάστατος μοναδιαίος πίνακας,  $\forall k \in M$ . Επιπλέον,  $z_{Q+1,Q+1} = 1$  είναι το στοιχείο στην γραμμή  $Q+1$  και στην στήλη  $Q+1$  του πίνακα  $\mathbf{Z}$ ,  $\mathbf{A}_{1,h}$  και  $\mathbf{A}_{2,j}$  είναι η γραμμή  $h$  του πίνακα  $\mathbf{A}_1$  και η γραμμή  $j$  του πίνακα  $\mathbf{A}_2$ , αντίστοιχα. Οι περιορισμοί 4.8β', 4.8γ' και 4.8δ' είναι ισοδύναμοι με τους περιορισμούς 4.7β', 4.7γ' και 4.7δ', αντίστοιχα.

Στο πρόβλημα  $\mathcal{F3} - 1$ , μόνο ο περιορισμός για την τάξη του πίνακα  $\text{rank}(\mathbf{Z}) = 1$  δεν είναι κυρτός, ενώ η συνάρτηση ελαχιστοποίησης και οι υπόλοιποι περιορισμοί είναι κυρτοί. Συμβολίζουμε με  $\mathbf{Z}^*$  την βέλτιστη λύση του  $\mathcal{F3} - 1$  χωρίς τον περιορισμό για την τάξη του πίνακα, η οποία υπολογίζεται από έναν convex programming solver. Αν η  $\mathbf{Z}^*$  είναι πρώτης τάξης πίνακας, μπορούμε να κατασκευάσουμε τη λύση του προβλήματος  $\mathcal{F3} - 1$  ως εξής:

$$\mathbf{Z}^* = \begin{bmatrix} \mathbf{y}^* \\ \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{y}^{*T} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^* \\ t^* \\ \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{*T} & t^* & \mathbf{1} \end{bmatrix}. \text{ Αφαιρούμε τον επάνω-αριστερά (MN+N)}$$

$\times$  (MN+N) υπο-πίνακα του  $\mathbf{Z}^*$ , ο οποίος είναι ο  $\mathbf{Z}'$ , όπου ο  $\mathbf{Z}'$  είναι θετικός semidefinite [23]. Έχουμε  $\mathbf{Z}' = \mathbf{x}^* \mathbf{x}^{*T}$ . Επειδή  $x_{ik} \in \{0, 1\}$ ,  $x_{ik} x_{ik} = x_{ik}$ . Οπότε προκύπτει ότι ο  $\mathbf{x}^*$  είναι η διαγώνιος του  $\mathbf{Z}'$ . Μπορούμε να κατασκευάσουμε τον πίνακα της βέλτιστης ανάθεσης εργασιών  $\mathbf{X}^*$  από τον  $\mathbf{x}^*$ .

Αν ο  $\mathbf{Z}^*$  δεν είναι πρώτης τάξης πίνακας, χρησιμοποιούμε έναν αλγόριθμο με βάση την Gaussian randomization για να βρούμε μια προσεγγιστική λύση στο πρόβλημα  $\mathcal{F3} - 1$ . Οι βασικές ιδέες του αλγορίθμου είναι οι εξής:

1) Δημιουργούμε τυχαία  $L$  πιθανές λύσεις του  $\mathcal{F3} - 1$  με βάση μια πολυμεταβλητή

Γκαουσιανή κατανομή με μηδενική μέση τιμή και με  $\mathbf{Z}'$  ως τον πίνακα της διασποράς.

2) Επιλέγουμε τη λύση που ελαχιστοποιεί την τιμή-στόχο για το  $\mathcal{F}1$ .

Έστω  $L$  το μέγεθος του δείγματος του randomization και  $l$  ο δείκτης ενός τυχαίου δείγματος. Ο προτεινόμενος αλγόριθμος μπορεί να θεωρηθεί ως μια παραλλαγή των προσεγγίσεων του randomization που αναφέρονται στα [24], [25]. Αφαιρώντας τον επάνω-αριστερά  $(MN+N) \times (MN+N)$  υπο-πίνακα  $\mathbf{Z}'$ , μπορούμε να παράγουμε  $L$  τυχαίους  $(MN+N) \times 1$  πίνακες, με τον  $l$ -οστό πίνακα να συμβολίζεται ως  $\xi^{(l)}$ , ο οποίος βασίζεται  $\xi^{(l)} \sim \mathcal{N}(\mathbf{0}_{(MN+N) \times 1}, \mathbf{Z}')$ , για  $l = 1, \dots, L$ . Η διάσταση του  $\xi^{(l)}$  είναι ίση με αυτήν του  $\mathbf{x}$ . Σε αυτό το σημείο πρέπει να επαναφέρουμε τα δυαδικά χαρακτηριστικά του  $\xi^{(l)}$  ώστε να ικανοποιείται το πρόβλημα  $\mathcal{F}1$ . Αντιστοιχίζουμε κάθε πίνακα  $\xi^{(l)} \in \mathcal{R}^{MN+N}$  σε έναν πίνακα  $\hat{\mathbf{x}}^{(l)} \in [0, 1]^{MN+N}$ , με μία σιγμοειδή συνάρτηση  $\text{sig}(x) \triangleq \frac{1}{1+\exp(-\mu x)}$ , όπου  $\mu \gg 1$ . Στη συνέχεια, μετατρέπουμε τον κάθε πίνακα  $\hat{\mathbf{x}}^{(l)} \in [0, 1]^{MN+N}$  στον αντίστοιχο πίνακα ανάθεσης  $\hat{\mathbf{X}}^{(l)} \in [0, 1]^{N \times (M+1)}$ . Για κάθε γραμμή του  $\hat{\mathbf{X}}^{(l)}$  θέτουμε το μεγαλύτερο στοιχείο ίσο με 1 και όλα τα υπόλοιπα ίσα με 0. Με αυτή την διαδικασία, για κάθε πίνακα  $\hat{\mathbf{X}}^{(l)}$ , παίρνουμε έναν πίνακα  $\bar{\mathbf{X}}^{(l)} \in \{0, 1\}^{N \times (M+1)}$ , ο οποίος ικανοποιεί τους περιορισμούς 4.3β'. Τέλος, ψάχνοντας το ελάχιστο από όλους τους  $L$  πίνακες  $\bar{\mathbf{X}}^{(l)}$  μπορούμε να βρούμε την λύση  $\{\mathbf{x}^{SDR}, r_0^{SDR}\}$  στο πρόβλημα  $\mathcal{F}1$ .

Παρατηρούμε ότι υπάρχει ένας συμβιβασμός ανάμεσα στον αριθμό των τυχαίων δειγμάτων  $L$  και την επίδοση του αλγορίθμου. Σύμφωνα με [22], μπορούμε να πλησιάσουμε σε μια αποδοτική λύση για μικρές τιμές του  $L$  σε σχέση με το συνολικό μέγεθος του χώρου των αποφάσεων. Η πολυπλοκότητα του αλγορίθμου είναι  $O(m^4 n^{1/2} \log(1/e) + LNM)$ , όπου  $n = N(M+1)$  και  $m = N(M+1) + M + N + 2$ .

## Πειράματα - Προσομοιώσεις

---

**Σ**το κεφάλαιο αυτό παρουσιάζονται οι προσομοιώσεις που τρέξαμε για την επίλυση του προβλήματος της ανάθεσης εργασιών με τη χρήση της τεχνικής του SDR. Ο κώδικας των προσομοιώσεων είναι γραμμένος στην Python και δίνεται στο [26].

### 5.1 Συγκρίσεις Αλγορίθμων

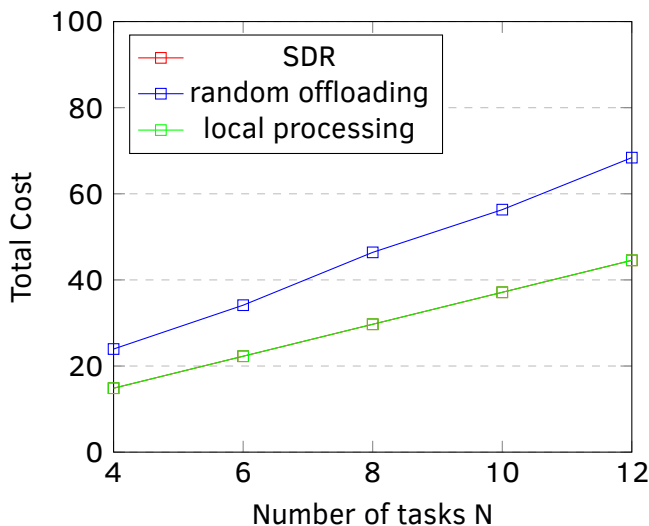
Στην ενότητα αυτή συγκρίνουμε την απόδοση του SDR σε σχέση με άλλους αλγορίθμους ανάθεσης εργασιών, με βάση τις παραμέτρους που αναφέρονται στα [27], [28]. Υπολογίζουμε το συνολικό κόστος της εκτέλεσης με τον κάθε αλγόριθμο για διαφορετικούς ρυθμούς μετάδοσης δεδομένων και αριθμούς εργασιών.

#### 5.1.1 Τιμές των παραμέτρων

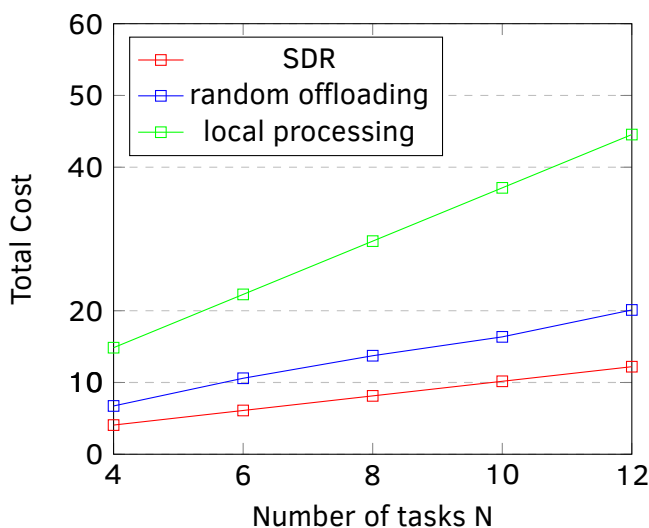
Θεωρούμε μια κινητή συσκευή με συχνότητα CPU  $400 \times 10^6$  κύκλους/sec και υπολογιστική ισχύ  $0.8 \text{ W}$ , οπότε το  $\rho$  υπολογίζεται ως  $\rho = \frac{0.8}{400 \times 10^6} = 1.25 \times 10^{-26} (J/)$ . Επιπλέον, οι τιμές των  $P^{Tx}$ ,  $P^{Rx}$  είναι  $1.258\text{W}$  και  $1.181\text{W}$  αντίστοιχα. Όσον αφορά τη σχέση των δεδομένων εισόδου και των κύκλων εκτέλεσης επιλέγουμε  $w_i = \kappa a_i$ . Σύμφωνα με [27]  $\kappa = 330$  κύκλοι/byte. Τα δεδομένα εξόδου  $b_i$  είναι ίσα με το 20% των δεδομένων εισόδου  $a_i$ . Οι ρυθμίσεις σχετικά με τα σημεία πρόσβασης αλλάζουν ανάλογα με το πείραμα. Το uplink κέρδος του κάθε καναλιού ανάμεσα σε ένα σημείο πρόσβασης και την κινητή συσκευή  $h_{ik}^{UL}$  θεωρείται ίσο με το downlink κέρδος του κάθε καναλιού  $h_{ik}^{DL}$ . Η παράμετρος  $\mu$  της σιγμοειδούς συνάρτησης είναι ίση με 10 και τα βάρη προκύπτουν από τη σχέση  $\lambda_t = 1 - \lambda_e$ . Έχουμε τρέξει τις προσομοιώσεις σε περιβάλλον Python 3.

#### 5.1.2 Αριθμητικά Αποτελέσματα

Στις γραφικές παραστάσεις 5.1, 5.2 και 5.3 συγκρίνουμε την απόδοση διαφορετικών αλγορίθμων σε σχέση με τον αριθμό  $N$  των εργασιών, για διαφορετικούς ρυθμούς μετάδοσης δεδομένων. Θέτουμε το μέγεθος των δεδομένων εισόδου  $a_i = 0.5\text{MB}$ ,  $\forall i \in N$ , τον αριθμό των γκαουσιανών δειγμάτων  $L = 100$  και  $\lambda_e = 0.5$ . Έχουμε 2 σημεία πρόσβασης με ρυθμούς εξυπηρέτησης  $2 \times 10^9$  και  $2.2 \times 10^9$  κύκλους/sec αντίστοιχα. Σε



Σχήμα 5.1: Κόστος αλγορίθμων για μικρά data rates

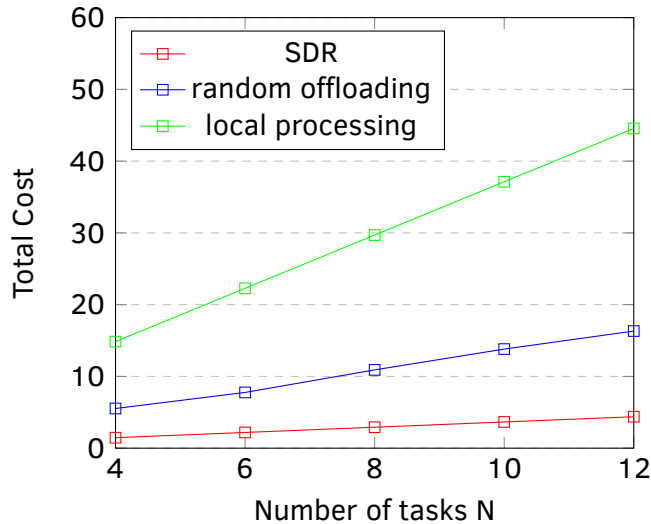


Σχήμα 5.2: Κόστος αλγορίθμων για μεσαία data rates

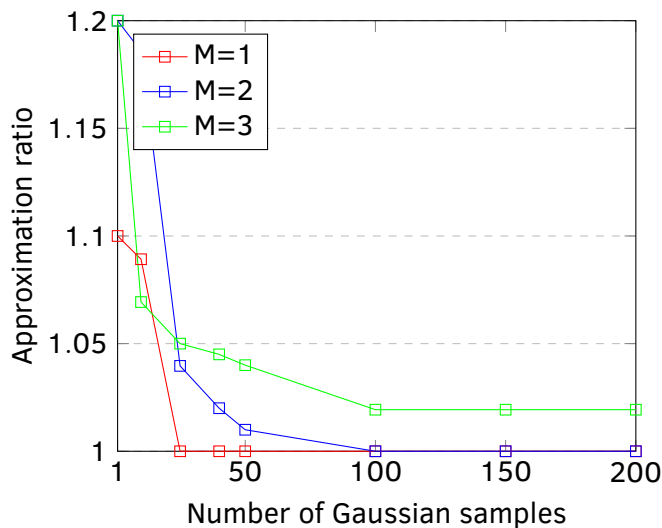
αυτές τις παραστάσεις οι ρυθμοί μετάδοσης των δεδομένων χωρίζονται σε 3 περιοχές, από 500kbps έως 1Mbps, από 2Mbps έως 10Mbps και από 20Mbps έως 50Mbps. Στις γραφικές παραστάσεις 5.1, 5.2 και 5.3 ο αλγόριθμος SDR συγκρίνεται με τις ακόλουθες μεθόδους : 1) Τοπική επεξεργασία (local processing), στην οποία όλες οι εργασίες εκτελούνται τοπικά και 2) Τυχαία Ανάθεση (Random assignment), στην οποία κάθε εργασία ανατίθεται τυχαία.

Παρατηρούμε ότι για μικρούς ρυθμούς μετάδοσης δεδομένων ο SDR αναθέτει όλες τις εργασίες τοπικά για κάθε τιμή του N (ταυτίζεται η γραφική παράστασή του με αυτήν της τοπικής επεξεργασίας). Για μέσους και μεγάλους ρυθμούς παρατηρούμε ότι ο SDR έχει αρκετά μικρότερο κόστος από τους άλλους δύο, το οποίο ισχύει για όλες τις τιμές του N. Όσο αυξάνονται οι ρυθμοί μετάδοσης, η τοπική επεξεργασία γίνεται λιγότερο αποδοτική σε σχέση με την τυχαία ανάθεση και τον SDR. Το γεγονός αυτό είναι λογικό, καθώς με μεγαλύτερη ταχύτητα μετάδοσης η μεταφόρτωση εργασιών μπορεί να γίνει





Σχήμα 5.3: Κόστος αλγορίθμων για μεγάλα data rates



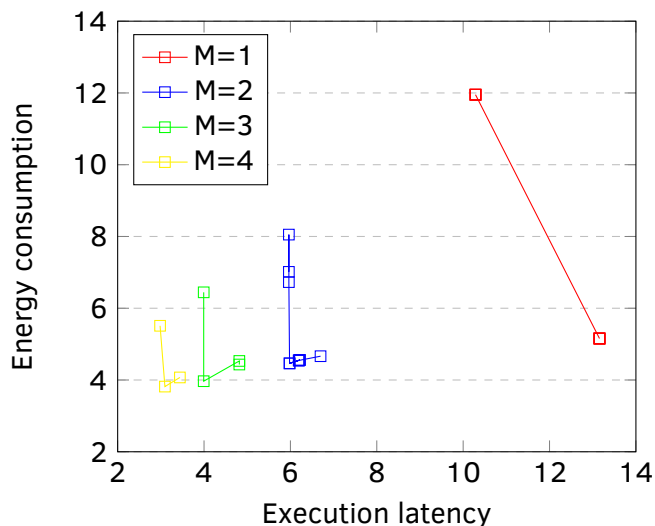
Σχήμα 5.4: Ποσοστό προσέγγισης ανά L για διαφορετικά M

γρηγορότερα, με αποτέλεσμα το συνολικό κόστος να μειώνεται.

## 5.2 Πολλαπλά σημεία πρόσβασης

Στη συνέχεια εξετάζουμε το ποσοστό προσέγγισης του SDR στη βέλτιστη λύση. Εξετάζουμε πως ο αριθμός των γκαουσιανών δειγμάτων επηρεάζει την προσέγγιση, για διαφορετικές τιμές  $M = 1, 2, 3$ . Θέτουμε  $N = 10$  και  $a_i = 0.1, 0.2, \dots, 1\text{MB}$  και  $\lambda_e = 0.5$ . Ο ρυθμός εξυπηρέτησης του κάθε σημείου πρόσβασης είναι  $r_k = 2 \times 10^9$  κύκλοι/sec.

Παρατηρούμε στην γραφική παράσταση 5.4 πως όσο αυξάνεται η τιμή του L τόσο πιο κοντά πλησιάζουμε στην βέλτιστη λύση, όμως μετά την τιμή  $L = 100$ , το ποσοστό προσέγγισης μεταβάλλεται αρκετά αργά. Επιπλέον, παρατηρούμε πως για το ίδιο L, όσο αυξάνεται ο αριθμός των σημείων πρόσβασης M, η ακρίβεια της προσέγγισης του αλγορίθμου μειώνεται.



Σχήμα 5.5: Κατανάλωση ενέργειας και Χρόνος εκτέλεσης

### 5.3 Κατανάλωση Ενέργειας και Χρονική Καθυστέρηση

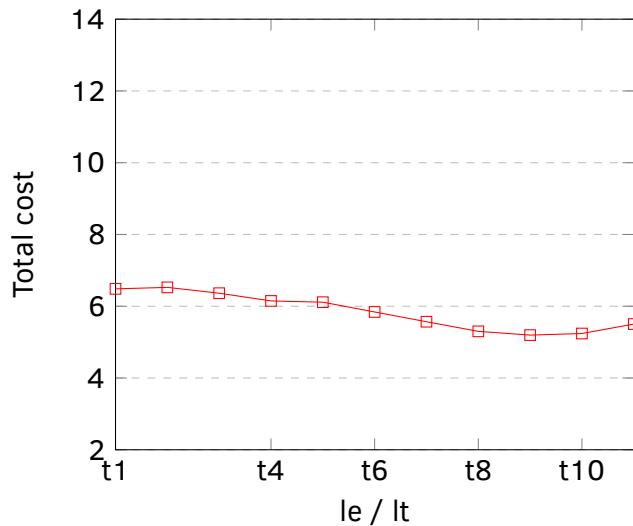
Εξετάζουμε τον συμβιβασμό ανάμεσα στην κατανάλωση ενέργειας και την χρονική καθυστέρηση της κινητής συσκευής. Θέτουμε  $N = 10$ ,  $a_i = 0.1, 0.2, \dots, 1\text{MB}$  και τον ρυθμό εξυπηρέτησης του κάθε σημείου πρόσβασης ίσο με  $r_k = 2 \times 10^9$  κύκλοι/sec. Η τιμή του  $\lambda_e$  επιλέγεται έτσι ώστε  $\frac{\lambda_e}{\lambda_t} = 10^q$ , όπου  $q \in [-2, 2]$ . Η τιμή του  $q$  αυξάνεται από -2 έως 2 με βήμα 0.4. Για κάθε  $\lambda_e$  υπολογίζουμε την κατανάλωση ενέργειας και την χρονική καθυστέρηση με τον SDR. Τα σημεία της γραφικής παράστασης 5.5 είναι για τις τιμές του  $q$  από -2 έως 2.

Βλέπουμε πως υπάρχουν μεγάλα κέρδη ως προς την ενέργεια και την χρονική καθυστέρηση, όσο αυξάνονται τα σημεία πρόσβασης. Αυτό είναι λογικό, καθώς η κινητή συσκευή έχει περισσότερες επιλογές για την ανάθεση των εργασιών, οι οποίες μπορούν να εκτελούνται παράλληλα. Το μεγαλύτερο κέρδος παρατηρείται για 2 σημεία πρόσβασης σε σχέση με την περίπτωση του ενός, ενώ για παραπάνω από 2 η διαφορά δεν είναι τόσο μεγάλη. Επίσης, παρατηρούμε πως με την αύξηση του  $q$ , η ενέργεια μειώνεται και η χρονική καθυστέρηση αυξάνεται. Αυτό συμβαίνει καθώς με την αλλαγή στις τιμές που έχουν τα βάρη, αλλάζει και η προτεραιότητα της συσκευής ανάμεσα σε ενέργεια και χρόνο.

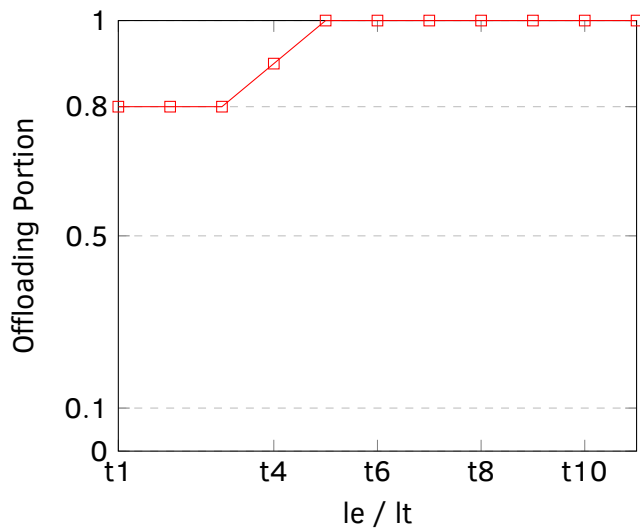
Συμπεραίνουμε πως είναι ωφέλιμο για την συσκευή να μεταφορτώνει τις εργασίες σε πάνω από ένα σημείο πρόσβασης. Το ποσοστό του κέρδους όμως, μειώνεται όσο αυξάνεται ο αριθμός των σημείων πρόσβασης.

### 5.4 Συνολικό κόστος και Ποσοστό μεταφόρτωσης

Στις γραφικές παραστάσεις 5.6 και 5.7 ερευνούμε την επιρροή που έχουν τα βάρη  $\lambda_e$ ,  $\lambda_t$  πάνω στο συνολικό κόστος και το ποσοστό μεταφόρτωσης, αντίστοιχα. Οι τιμές των παραμέτρων είναι παρόμοιες με αυτές του κεφαλαίου 5.3, αλλά με 2 σημεία πρό-



Σχήμα 5.6: Συνολικό κόστος



Σχήμα 5.7: Ποσοστό μεταφόρτωσης

σβασης. Στον οριζόντιο άξονα έχουμε τον λόγο των βαρών  $\frac{\lambda_e}{\lambda_t} = 10^q$ , με τις ενδείξεις  $t_i$  όπου  $q \in [-2, 2]$ .

Βλέπουμε στην γραφική παράσταση 5.6 ότι το συνολικό κόστος ταλαντεύεται γύρω από μια μέση τιμή (γύρω στο 5.5 με αυτές τις παραμέτρους). Παρατηρούμε τοπικά μέγιστα στις δύο άκρες της γραφικής, το οποίο είναι αναμενόμενο αφού σε εκείνα τα σημεία η συσκευή θέτει ως προτεραιότητα την χρονική καθυστέρηση έναντι της ενέργειας ή το αντίστροφο. Το συνολικό κόστος πλησιάζει την ελάχιστη τιμή του όσο πιο κοντά βρίσκεται στη μέση του διαγράμματος, όπου λαμβάνουμε υπόψη και τα δύο κριτήρια με τα κατάλληλα βάρη.

Στην γραφική παράσταση 5.7 παρατηρούμε πως εξ αρχής η κινητή συσκευή μεταφορτώνει το 80 τοις εκατό των εργασιών στα σημεία πρόσβασης. Για μικρές τιμές του λόγου  $\frac{\lambda_e}{\lambda_t}$  η συσκευή έχει ως προτεραιότητα την ελαχιστοποίηση του συνολικού χρόνου εκτέλεσης των εργασιών και για αυτό το λόγο παρατηρούμε το μικρότερο ποσοστό με-

ταφόρτωσης. Στην συνέχεια, όσο αυξάνεται ο λόγος  $\frac{\lambda_e}{\lambda_t}$ , αποκτά περισσότερη σημασία η εξοικονόμηση ενέργειας, οπότε περιμένουμε αύξηση του αριθμού των εργασιών που μεταφορτώνονται, για να μην επιβαρύνεται τόσο πολύ η κινητή συσκευή.

Παρατηρώντας την γραφική παράσταση 5.7, συνειδητοποιούμε την αξία της ιδέας της μεταφόρτωσης εργασιών. Ακόμα και στο σενάριο που έχουμε ως βασικό στόχο την ταχύτητα εκτέλεσης των εργασιών και αγνοούμε την κατανάλωση ενέργειας της συσκευής, το 80 τοις εκατό των εργασιών μεταφορτώνεται. Όσο αρχίζουμε να λαμβάνουμε υπόψη και την εξοικονόμηση ενέργειας το ποσοστό αυτό αυξάνεται και φτάνει αρκετά γρήγορα στο 100 τοις εκατό. Τα νούμερα αυτά εξαρτώνται και από τις παραμέτρους του συστήματος, αλλά παρατηρούμε την σημασία της δυνατότητας ανάθεσης των εργασιών και της γρήγορης απόφασης για τη μεταφόρτωση τους με τη βοήθεια του SDR.

## 5.5 Συμπεράσματα από τις προσομοιώσεις

Από τις προσομοιώσεις αυτού του κεφαλαίου μπορούμε να εξάγουμε ορισμένα συμπεράσματα:

- Ο αλγόριθμος SDR είναι αρκετά πιο αποδοτικός σε σχέση με τους δύο αλγορίθμους με τους οποίους τον συγκρίναμε. Μάλιστα, υπερτερεί των άλλων δύο όσο αυξάνεται ο αριθμός των εργασιών, αλλά και για μεγαλύτερους ρυθμούς μετάδοσης των δεδομένων. Παρατηρούμε πως η αποδοτικότητά του διατηρείται για μεγαλύτερα συστήματα.
- Όσον αφορά την ακρίβεια του αλγορίθμου, είδαμε πως πλησιάζει σε μεγάλο βαθμό τη βέλτιστη λύση και η ικανότητά του αυτή διατηρείται για διαφορετικό πλήθος από σημεία πρόσβασης. Μπορούμε να πλησιάσουμε όλο και περισσότερο την βέλτιστη λύση ρυθμίζοντας τον αριθμό των γκασουσιανών δειγμάτων, αυξάνοντας βέβαια την πολυπλοκότητα του αλγορίθμου. Παρατηρούμε πως για 100 γκασουσιανά δείγματα έχουμε την ιδανική απόδοση, με μεγάλη προσέγγιση στην βέλτιστη λύση και σχετικά μικρή πολυπλοκότητα.
- Σημαντικό πλεονέκτημα του αλγορίθμου είναι η ταχύτητα εκτέλεσης του. Είναι απαραίτητο να ληφθεί άμεσα η απόφαση για την ανάθεση των εργασιών, ώστε να εξοικονομηθεί χρόνος από τη συνολική επεξεργασία τους.
- Ο αριθμός των σημείων πρόσβασης επηρεάζει την απόδοση του συστήματος. Όσο αυξάνουμε τα σημεία πρόσβασης, έχουμε μείωση της χρονικής καθυστέρησης και την κατανάλωσης της ενέργειας, μοιράζοντας τις εργασίες σε περισσότερα σημεία. Ανάλογα με τις ανάγκες του συστήματος μπορούμε να επιλέξουμε τον ιδανικό αριθμό των σημείων πρόσβασης, αλλά όσο αυξάνονται κατά ένα το κέρδος αυξάνεται με φθίνοντα ρυθμό.
- Σημαντική είναι και η δυνατότητα που μας προσφέρει ο αλγόριθμος SDR, να μπορούμε να θέτουμε προτεραιότητα ανάμεσα στον χρόνο εκτέλεσης και την εξοικονόμηση ενέργειας.

κονόμηση ενέργειας. Αυτό μπορεί να επιτευχθεί με την ρύθμιση στις τιμές που έχουν τα βάρη.

- Σε όλες τις προσομοιώσεις το ποσοστό της μεταφόρτωσης των εργασιών δεν έπεσε χαμηλότερα του 0.8. Σταθερά ο αλγόριθμος επιλέγει να μεταφορτώνει τουλάχιστον το 80% των εργασιών, επιβεβαιώνοντας την σημασία για τη μεταφόρτωση των εργασιών σε ένα σύστημα.



Μέρος **III**

Επίλογος

---





## Επίλογος

---

### 6.1 Σύνοψη και Συμπεράσματα

Σε αυτή τη διπλωματική εργασία ασχοληθήκαμε με το πρόβλημα της ανάθεσης εργασιών από μια κινητή συσκευή σε ορισμένα σημεία πρόσβασης. Λόγω του μεγάλου φόρτου των σύγχρονων εφαρμογών στις κινητές συσκευές, αναζητούμε έναν τρόπο για να μειώσουμε την επιβάρυνση της συσκευής κατά την εκτέλεση των εφαρμογών. Οπότε, μια αντιμετώπιση του προβλήματος είναι η μεταφόρτωση των εργασιών προς εκτέλεση σε άλλες υπολογιστικές μονάδες για την επεξεργασία τους. Καταλήγουμε στο πρόβλημα της επιλογής για την ανάθεση των εργασιών στα σημεία πρόσβασης. Στόχος είναι η ελαχιστοποίηση του χρόνου εκτέλεσης των εργασιών και της κατανάλωσης ενέργειας της κινητής συσκευής.

Θεωρήσαμε το σύστημα της κινητής συσκευής και των σημείων πρόσβασης, στα οποία μπορεί να μεταφορτώνει τις εργασίες της για την εκτέλεσή τους. Αναφέραμε παρόμοιες περιπτώσεις από τη σχετική βιβλιογραφία και εξηγήσαμε την λειτουργία της τεχνικής SDR, η οποία είναι εξαιρετικά χρήσιμη σε προβλήματα αυτού του είδους.

Μοντελοποιήσαμε το σύστημα και διατυπώσαμε το πρόβλημα ως ένα quadratic πρόβλημα με quadratic περιορισμούς, για την ελαχιστοποίηση του χρόνου εκτέλεσης των εργασιών και της κατανάλωσης ενέργειας της κινητής συσκευής, το οποίο είναι NP-hard. Στη συνέχεια, προτείνουμε τη λύση με τον αλγόριθμο SDR για την προσέγγιση της βέλτιστης λύσης του προβλήματος, καθώς δεν μπορούμε να λύσουμε άμεσα το NP-hard πρόβλημα. Η ταχύτητα εκτέλεσης του αλγορίθμου SDR, καθώς και η ακρίβεια της προσέγγισής του, τον καθιστούν μια αποδοτική αντιμετώπιση του προβλήματος.

Μέσω των προσομοιώσεων παρατηρείται η αποδοτικότητα του SDR σε σχέση με άλλους αλγορίθμους για την επίλυση του προβλήματος, ακόμα και για μεγάλο αριθμό εργασιών. Ανάλογα με τους διαθέσιμους πόρους, μπορεί να προσεγγίσει την βέλτιστη λύση σε μεγάλο βαθμό, διατηρώντας επαρκή πολυπλοκότητα. Ο αριθμός των σημείων πρόσβασης είναι ένας σημαντικός παράγοντας για την απόδοση του συστήματος. Με την αύξηση τους, μπορεί να μειωθεί το συνολικό κόστος. Χρησιμοποιώντας και τα κατάλληλα βάρη στο πρόβλημα, προσφέρεται και η δυνατότητα στη συσκευή να θέτει προτεραιότητα ανάμεσα στον χρόνο εκτέλεσης των εργασιών και την κατανάλωση της ενέργειας της, εξοικονομώντας τους αντίστοιχους πόρους. Επιπλέον, τονίζεται η ση-

μασία της μεταφόρτωσης των εργασιών στα σημεία πρόσβασης, καθώς ο αλγόριθμος καταλήγει να μεταφορτώνει τουλάχιστον 8 από τις 10 εργασίες στα σημεία πρόσβασης, για κάθε σενάριο.

## 6.2 Μελλοντικές Επεκτάσεις

Όσον αφορά το περιεχόμενο αυτής της εργασίας, θα μπορούσε να επεκταθεί ως προς την αντιμετώπιση του προβλήματος της ανάθεσης εργασιών με ελαστική συχνότητα CPU. Η ρύθμιση της συχνότητας CPU της κινητής συσκευής ανάλογα με τις εργασίες που έχει, μπορεί να προσφέρει έναν επιπλέον τρόπο εξοικονόμησης ενέργειας, αυξάνοντας την αποδοτικότητα του συστήματος. Σε περιπτώσεις που ο βασικός στόχος είναι η γρήγορη εκτέλεση των εργασιών, μπορεί να αυξάνεται η συχνότητα CPU της συσκευής, ώστε να μπορεί να εκτελέσει περισσότερες εργασίες στο ίδιο χρονικό διάστημα. Αντίθετα, με μείωση της συχνότητας CPU μπορεί να εξοικονομήσει ενέργεια. Με αυτόν τον τρόπο προσφέρει επιπλέον επιλογές στην κινητή συσκευή για την τοπική εκτέλεση των εργασιών. Το πρόβλημα της ανάθεσης των εργασιών παραμένει ένα QCCP ελαχιστοποίησης, με μία παραπάνω παράμετρο σχετικά με την τιμή της συχνότητας CPU της συσκευής και έναν παραπάνω περιορισμό σχετικά με την διακύμανση της. Μπορεί να διερευνηθεί η επίλυση του προβλήματος με τον SDR, με την διαδικασία που αναφέρεται στο [16]. Με τη σύγκριση των προσομοιώσεων αυτής της εργασίας και των αντίστοιχων προσομοιώσεων για την περίπτωση της ελαστικής συχνότητας στην Python, μπορούμε να συμπεράνουμε για την σημασία της συχνότητας CPU της συσκευής.

Μια δεύτερη πιθανή επέκταση της παρούσας εργασίας θα μπορούσε να είναι η περίπτωση που έχουμε παραπάνω από μία κινητή συσκευή. Σε αυτήν την περίπτωση, η ανάθεση των εργασιών γίνεται αρκετά πολύπλοκη και αντιμετωπίζονται δυσκολίες σχετικά με τη μετάδοση των δεδομένων και τον φόρτο των σημείων πρόσβασης.

## Βιβλιογραφία

---

- [1] K. Kumar, J. Liu, Y. H. Lu και B. Bhargava. *A survey of computation offloading for mobile systems*. *Mobile Netw. Appl.*, 18(1):129–140, 2013.
- [2] Z. Sanaei, S. Abolfazli, A. Gani και R. Buyya. *Heterogeneity in mobile cloud computing: Taxonomy and open challenges*. *IEEE Commun. Surveys Tuts*, 16(1):369–392, 2014.
- [3] E. Cuervo et al. *Maui: Making smartphones last longer with code offload*. *Proc. ACM MobiSys*, pp. 49–62, 2010.
- [4] S. Kosta, A. Aucinas, P. Hui, R. Mortier και X. Zhang. *Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading*. *Proc. IEEE INFOCOM*, pp. 945–953, 2012.
- [5] T. Q. Quek, G. D. L. Roche, I. Güvenç και M. Kountouris. *Small Cell Networks Deployment Phy Techniques and Resource Management*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [6] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila και T. Taleb. *Survey on multi-access edge computing for internet of things realization*. *IEEE Commun. Surv. Tutor*, 20(4):2961–2991, 2018.
- [7] B. Liu, X. Xu, L. Qi, Q. Ni και W. Dou. *Task scheduling with precedence and placement constraints for resource utilization improvement in multi-user MEC environment*. *J. Syst. Archit*, 114(101970), 2021.
- [8] X. Deng, J. Li, E. Liu και H. Zhang. *Task allocation algorithm and optimization model on edge collaboration*. *J. Syst. Archit*, 110(101778), 2020.
- [9] J. W. Ryu, Q. V. Pham, H.N. Luan, W. J. Hwang, J. D. Kim και J. T. Lee. *Multi-access edge computing empowered heterogeneous networks: A novel architecture and potential works*. *Symmetry*, 11(7), 2019.
- [10] Q. V. Pham, L.B. Le, S. H. Chung και W. J. Hwang. *Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation*. *IEEE Access* 7, pp. 16444–16459, 2019.
- [11] P. Salot. *A survey of various scheduling algorithm in cloud computing environment*. *Int. J. Res. Eng. Technol.*, 2(2):131–135, 2013.

- [12] Akhirul Islam, Arindam Debnath, Manojit Ghose και Suchetana Chakraborty. *A Survey on Task Offloading in Multi-access Edge Computing*. *Journal of Systems Architecture*, pp. 1–16, 2021.
- [13] K. Benzekki, A. El Fergougui και A. Elbelrhiti Elalaoui. *Software-defined networking (SDN): a survey*. *Secur. Commun. Netw.*, 9(18):5803–5833, 2016.
- [14] Meng Hsi Chen, Ben Liang και Min Dong. *A Semidefinite Relaxation Approach to Mobile Cloud Offloading with Computing Access Point*. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, pp. 186–190, 2015.
- [15] Mithun Mukherjee, Suman Kumar, Constandinos X. Mavromoustakis, George Mastorakis, Rakesh Matam, Vikas Kumar και Qi Zhang. *Latency-Driven Parallel Task Data Offloading in Fog Computing Networks for Industrial Applications*. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, pp. 6050–6058, 2020.
- [16] Thinh Quang Dinh, Jianhua Tang, Quang Duy La και Tony Q. S. Quek. *Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling*. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 65(8):3571–3584, 2017.
- [17] Zhi Quan Luo, Wing Kin Ma, Anthony Man Cho So, Yinyu Ye και Shuzhong Zhang. *Semidefinite Relaxation of Quadratic Optimization Problems*. *IEEE SIGNAL PROCESSING MAGAZINE*, pp. 20–34, 2010.
- [18] L. Vandenberghe και S. Boyd. *Semidefinite programming*. *SIAM Rev*, pp. 49–95, 1996.
- [19] Y. Y. Shih, W. H. Chung, A. C. Pang, T. C. Chiu και H. Y. Wei. *Enabling low-latency applications in fog-radio access networks*. *IEEE Netw.*, 31(1):52–58, 2017.
- [20] M. E. T. Gerards, J. L. Hurink και J. Kuper. *On the interplay between global DVFS and scheduling tasks with precedence constraints*. *IEEE Trans. Comput.*, 64(6):1742–1754, 2015.
- [21] M. Ehrgott. *Multicriteria Optimization*. Springer, New York, NY, USA, 2η έκδοση, 2006.
- [22] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye και S. Zhang. *Semidefinite relaxation of quadratic optimization problems*. *IEEE Signal Process. Mag.*, 27(3):20–34, 2010.
- [23] F. Zhang. *SMatrix Theory: Basic Results and Techniques*. Springer-Verlag, Berlin, Germany, 1999.
- [24] D. Bertsimas και Y. Ye. *Semidefinite relaxations, multivariate normal distributions, and order statistics*. Kluwer, Boston, MA, USA, 1998.
- [25] Z. Q. Luo, N. D. Sidiropoulos, P. Tseng και S. Zhang. *Approximation Bounds for Quadratic Optimization with Homogeneous Quadratic Constraints*. *SIAM Journal on Optimization*, 18(1):1–28, 2007.

- [26] Fotis Giannopoulos. *SDR programming for QCQP with fixed CPU frequency in Python*. [https://github.com/fotisgiann0/sdr\\_prog/blob/main/fixed\\_freq.py](https://github.com/fotisgiann0/sdr_prog/blob/main/fixed_freq.py).
- [27] A. P. Miettinen και J. K. Nurminen. *Energy efficiency of mobile clients in cloud computing*. in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput., Boston, MA, USA*, pp. 4–11, 2010.
- [28] Yu Xiao, Petri Savolainen, Arto Karppanen, Matti Siekkinen και Antti Ylä-Jääski. *Practical power modeling of data transmission over 802.11g for wireless applications*. in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw., Passau, Germany*, pp. 75–84, 2010.



## Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

---

SDR	Semi Definite Relaxation
QCQP	Quadratically Constrained Quadratic Program
BQP	Boolean Quadratic Problem
F-RAN	Fog-Radio Access Network
BnB	Branch and Bound
MILP	Mixed Integer Linear Programming
SDP	Semi Definite Programming
MCC	Mobile Cloud Computing
MEC	Mobile Edge Computing
κ.ά	και άλλα
SDN	Software Defined Networking
PSD	Symmetric Positive Semidefinite





## Απόδοση ξενόγλωσσων όρων

---

### Απόδοση

μεταφόρτωση  
σημείο πρόσβασης  
προσέγγιση  
συγκεντροποίηση  
αποκέντρωση  
εξυπηρετητής  
επεξεργασία  
κατανεμημένο  
προγραμματισμός  
χρονική καθυστέρηση  
κριτήριο  
διαθεσιμότητα  
ελεγκτής  
ελαστικότητα  
σταθερός  
κυρτό  
εργασία  
ανάθεση  
εργασία  
μεταφορά  
ασύρματη  
συχνότητα  
σταθμός  
απάντηση  
τοπικά  
αποτυχία  
ασφάλεια  
ετερογένεια  
άπληστος  
ευριστική  
πολυπλοκότητα  
προπόνηση  
βαθύ  
νευρωνικό

### Ξενόγλωσσος όρος

offloading  
access point  
approximation  
centralization  
decentralization  
server  
processing  
distributed  
programming  
latency  
objective  
availability  
controller  
flexibility  
fixed  
convex  
task  
allocation  
task  
transfer  
wireless  
frequency  
station  
response  
locally  
failure  
safety  
heterogeneity  
greedy  
heuristic  
complexity  
training  
deep  
neural

δυναμικός	dynamic
απομακρυσμένος	remote
ελαχιστοποίηση	minimization
βελτιστοποίηση	optimization
δυναδικός	binary
πολυωνυμικός	polynomial
άφιξη	arrival
ουρά	queue
ανέβασμα	upload
λογαριθμικό	logarithmic
ιδιοδιάσπαση	eigen decomposition
ιδιοτιμή	eigen value
ιδιοδιάνυσμα	eigen vector
αντιστοίχιση	mapping
χαλάρωση	relaxation
εύρος ζώνης	bandwidth
χρόνος συνοχής	coherence time
κατέβασμα	download
χωρητικότητα	capacitance
μοναδιαίος πίνακας	identity matrix
δείκτης	index