Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων


# Αλγόριθμοι για Προβλήματα Χωροθέτησης σε Χρονικά Μεταβαλλόμενους Χώρους

## Διπλωματικη Εργασια

του


**ΓΙΩΡΓΟΥ Π. ΜΟΥΣΑ**


**Επιβλέπων:** Δημήτρης Φωτάκης
Επίκουρος Καθηγητής Ε.Μ.Π.


Εργαστηριο Λογικης και Επιστημης Υπολογισμων

Αθήνα, Οκτώβριος 2016

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

# Αλγόριθμοι για Προβλήματα Χωροθέτησης σε Χρονικά Μεταβαλλόμενους Χώρους

## Διπλωματική Εργασία

του

**ΓΙΩΡΓΟΥ Π. ΜΟΥΣΑ**

**Επιβλέπων:** Δημήτρης Φωτάκης
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31η Οκτωβρίου 2016.

| | | |
|---|---|---|
| *(Υπογραφή)* | *(Υπογραφή)* | *(Υπογραφή)* |
| .............................. | .................................. | .................................... |
| Δημήτρης Φωτάκης<br>Επίκουρος<br>Καθηγητής Ε.Μ.Π. | Νικόλαος Παπασπύρου<br>Αναπληρωτής<br>Καθηγητής Ε.Μ.Π. | Αριστείδης Παγουρτζής<br>Αναπληρωτής<br>Καθηγητής Ε.Μ.Π. |

Αθήνα, Οκτώβριος 2016

*(Υπογραφή)*

................................

**Γιωργος Μουσα**

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

# Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον επιβλέποντα καθηγητή αυτής της εργασίας, κ. Δημήτρη Φωτάκη, για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα πολύ ενδιαφέρον θέμα στο εργαστήριο λογικής και επιστήμης υπολογισμών, καθώς και για το χρόνο που αφιέρωσε παρέχοντάς μου σημαντικές ιδέες και πολύτιμη καθοδήγηση.

Επιπλέον, δεδομένου ότι η ολοκλήρωση αυτής της εργασίας σημάνει τη λήξη των σπουδών μου στο Εθνικό Μετσόβιο Πολυτεχνείο, θέλω επίσης να ευχαριστήσω το σύνολο εκ των καθηγητών, ανεξαρτήτως τομέα, που αποτέλεσαν πρότυπα επιστημονικού αλλά και ανθρώπινου πνεύματος.

Τέλος, ευχαριστώ τους συμφοιτητές μου χάρη στους οποίους η εκπόνηση των σπουδών μου έγινε πολύ πιο ευχάριστη και παραγωγική.

# Περίληψη

Το αντικείμενο της παρούσας διπλωματικής εργασίας αφορά το πρόβλημα Χωροθέτησης σε Χρονικά Μεταβαλλόμενους Χώρους, το οποίο χρησιμοποιείται για την μοντελοποίηση της δυναμικής που περιγράφει εξελισσόμενα δίκτυα και μπορεί να νοηθεί ως μία γενίκευση του εκτενώς διερευνημένου προβλήματος Χωροθέτησης χωρίς χωρητικότητες. Η επιπλέον απαίτηση του προβλήματος, που το καθιστά δυσκολότερο από το πρόβλημα του οποίου αποτελεί γενίκευση, εντοπίζεται στην προτίμηση διαχρονικά σταθερών λύσεων, όπου οι αλλαγές ανάθεσης μεταξύ δύο οποιονδήποτε διαδοχικών χρονικών στιγμών είναι λίγες. Στα πλαίσια της εργασίας παρουσιάζονται και υλοποιούνται προϋπάρχοντες προσεγγιστικοί αλγόριθμοι καθώς και δύο νέοι αλγόριθμοι που προτείνουμε για το πρόβλημα. Τέλος, αξιολογείται η επίδοση των αλγόριθμων που περιγράφηκαν σε πληθώρα στιγμιοτύπων.

## Λέξεις Κλειδιά

Προσεγγιστικοί Αλγόριθμοι, Προβλήματα Χωροθέτησης, Χωροθέτηση Εξυπηρετητών, Χωροθέτηση χωρίς χωρητικότητες, Χρονικώς Μεταβαλλόμενοι Χώροι, Τεχνικές ακεραιοποίησης, Επιχειρησιακή έρευνα

# Abstract

This diploma thesis addresses the Facility Location Problem in Temporally Evolving Metrics, which is used to model the dynamics of evolving social or infrastructure networks and which can be viewed as a generalization of the widely researched Uncapacitated Facility Location Problem. The reason that sets the former problem harder than its static counterpart, relies on the requirement that its generated solutions are described by a temporal stability. In this context, there is a preference for a small number of switches of the assignments of clients to facilitites in any given set of consecutive time steps. We present and implement pre-existing approximation algorithms as well as two new algorithms that we propose. Lastly, we evaluate the performance and efficiency of the above mentioned algorithms in a variety of instances.

## Keywords

Approximation Algorithms, Facility Location, Uncapacitated Facility Location, Evolving Metrics, Dynamic Facility Location, Rounding Techniques, Operations Research

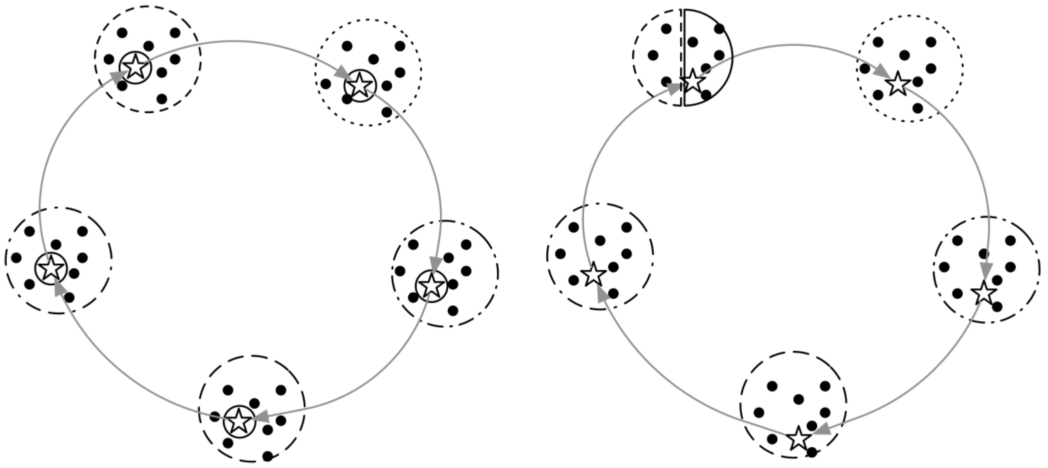
# Περιεχόμενα

# Κεφάλαιο 1

# Εισαγωγή

Η ανάγκη μελέτης του προβλήματος που αποτελεί το αντικείμενο της παρούσας εργασίας, δηλαδή του προβλήματος Χωροθέτησης σε Χρονικά Μεταβαλλόμενους Χώρους, αναδύθηκε κατά την αναζήτηση ενός καλύτερου τρόπου περιγραφής και μοντελοποίησης χρονικώς εξελισσόμενων δικτύων, είτε αυτά είναι κοινωνικά δίκτυα είτε είναι δίκτυα υποδομών.
Το πρόβλημα αυτό ανήκει σε μια ευρύτερη οικογένεια προβλημάτων, τα οποία στη διεθνή βιβλιογραφία ονομάζονται Facility Location προβλήματα. Σε αυτό το πλαίσιο, το πρόβλημα ορίζεται σαν μία γενίκευση στο πεδίο του χρόνου του προβλήματος Facility Location χωρίς χωρητικότητες. Αυτό που καθιστά το πρόβλημα ενδιαφέρον και σημαντικά δυσκολότερο του προβλήματος του οποίου αποτελεί γενίκευση, είναι η εισαγωγή της προτίμησης διαχρονικά σταθερών λύσεων, δηλαδή λύσεων με μικρό αριθμό αλλαγών ανάθεσης. Το πόσο ισχυρή είναι αυτή η προτίμηση, καθορίζεται από μία νέα μεταβλητή εισόδου την οποία ονομάζουμε κόστος αλλαγής ανάθεσης.

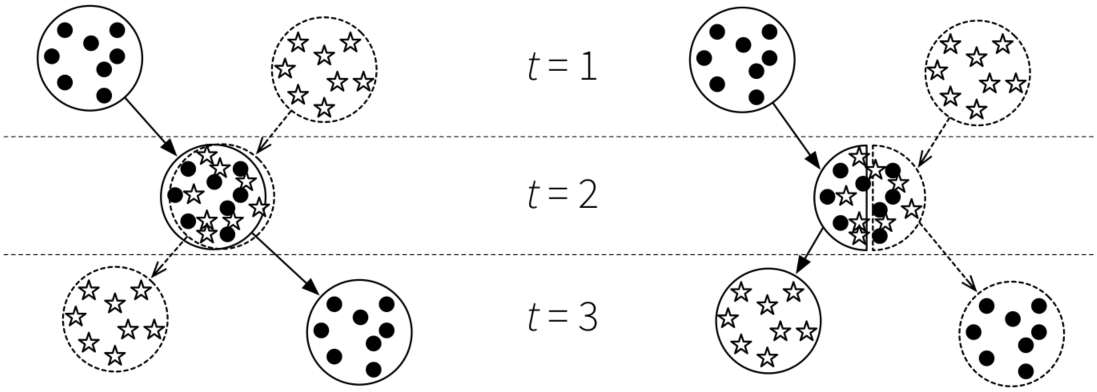
Πριν προβούμε στο κύριο σώμα της εργασίας, αξίζει να παρουσιάσουμε κάποιες καταστάσεις από την ανθρώπινη εμπειρία που θα μπορούσαν, μέσω μιας αφαιρετικής διαδικασίας, να αποτυπωθούν ως ένα πρόβλημα Χωροθέτησης σε Χρονικά Μεταβαλλόμενους Χώρους.
Αρχικώς, πρέπει να αποσαφηνίσουμε ότι τα Facility Location προβλήματα μπορούν να μας παρέχουν μία ομαδοποίηση (clustering) των clients επιτρέποντάς μας να αναγνωρίζουμε κοινά στοιχεία μεταξύ αυτών. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό, είναι να θεωρήσουμε υποψήφια facilities τις θέσεις που αντιστοιχούν σε κάθε client, πράγμα που συνεπάγεται ότι η επιλογή ενός facility συνιστά την επιλογή ενός αντιπρόσωπου για κάθε ομάδα από clients που παρουσιάζει σημαντικώς διαφορετική συμπεριφορά. Αυτή την ιδιότητα θέλουμε να γενικεύσουμε για δυναμικώς εξελισσόμενα προβλήματα όπου, για παράδειγμα, οι clients κινούνται εντός κάποιου μετρικού χώρου. Μερικά αντιπροσωπευτικά παραδείγματα όπου η αναγνώριση της δυναμικής που περιγράφει μια κατάσταση οδηγεί σε πολύ καλύτερη ομαδοποίηση από την στατική αντιμετώπιση είναι τα παρακάτω:



6 ομάδες δυναμικά 6 ομάδες στατικά

Σχήμα 1.1. Δάσκαλος που επισκέπτεται 5 τάξεις μαθητών.

Στο Σχήμα 1.1, παριστάνονται 5 τάξεις από μαθητές και ένας δάσκαλος που τις επισκέπτεται με κυκλικό τρόπο. Σε αυτό το σενάριο η επιθυμητή διαμέριση σε 6 ομάδες είναι μία ομάδα για κάθε τάξη και μία ομάδα για τον δάσκαλο. Ωστόσο, αν δούμε το πρόβλημα στατικά, τότε ο δάσκαλος ομαδοποιείται μαζί με τους μαθητές διαφορετικής κάθε φορά τάξης. Από την άλλη, όπως φαίνεται από το σχήμα, η διαμέριση που λαμβάνει χώρα χρησιμοποιώντας μια δυναμική προσέγγιση είναι αυτή που επιθυμούμε αφού γίνεται αναγνώριση της ιδιαίτερης φύσης της συμπεριφοράς του δασκάλου.




2 ομάδες δυναμικά 2 ομάδες στατικά

Σχήμα 1.2. Δύο κινούμενες ομάδες ανθρώπων που διασταυρώνονται.

Το παράδειγμα του Σχήματος 1.2 αποτυπώνει τη διασταύρωση δύο ομάδων από ανθρώπους. Μια στατική προσέγγιση του προβλήματος θα αναγνώριζε αρχικά τις δύο ομάδες, έπειτα θα τις συγχώνευε σε μία ομάδα, και τέλος θα τις ξαναχώριζε. Αντιθέτως, μία δυναμική προσέγγιση θα διατηρούσε σταθερή την ομαδοποίηση σε δύο ξεχωριστές ομάδες με τους ίδιους αντιπρόσωπους.

Αυτό που συμπεραίνεται από τα παραπάνω παραδείγματα είναι ότι η δυναμική προσέγγιση σε προβλήματα τύπου Facility Location αποδίδει καλύτερα στον εντοπισμό ομάδων με παρόμοια συμπεριφορά σε μεταβαλλόμενα δίκτυα. Ως τέτοια δίκτυα μπορούν να θεωρηθούν τα online κοινωνικά δίκτυα, από τα οποία μπορεί να αντληθεί τεράστιος όγκος πληροφορίας, αλλά και άλλα δίκτυα που αφορούν τις πραγματικές αλληλεπιδράσεις ανθρώπων, πχ σε νοσοκομεία, σχολεία, επιχειρήσεις. Η κατανόηση του τρόπου με τον οποίο εξελίσσονται αυτά τα δίκτυα με την πάροδο του χρόνου είναι πολύ σημαντική για την διαχείριση και την μετακίνηση πόρων με αποτελεσματικότητα, για την ανάλυση μετακίνησης πληθυσμών και κατ΄ επέκταση για κλάδους όπως είναι η επιδημιολογία.

Στις επόμενες υποενότητες του κεφαλαίου παρουσιάζεται το πεδίο με το οποίο καταπιάνεται η διπλωματική και δίνεται περιληπτικά μια προεπισκόπηση του περιεχομένου της εργασίας.

## 1.1 Αντικείμενο της διπλωματικής

Αντικείμενο της διπλωματικής αποτελεί η μελέτη του προβλήματος Facility Location σε Χρονικά Μεταβαλλόμενους Χώρους, όπως αυτό εισήχθη στην εργασία των Eisenstat et al [7]. Στόχοι της εργασίας αποτελούν η υλοποίηση και πειραματική αξιολόγηση αλγόριθμων προυπάρχοντων εργασιών καθώς και η προσφορά κάποιων νέων αλγόριθμων που θα περιγραφούν στα πλαίσια της εργασίας.

## 1.2 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε 7 κεφάλαια, το περιεχόμενο των οποίων (εξαιρούμενου του παρόντος) παρουσιάζεται περιληπτικά παρακάτω:

Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο που σχετίζεται με το υπό ανάλυση πρόβλημα. Συγκεκριμένα, περιλαμβάνεται μια επισκόπηση του προβλήματος Facility Location χωρίς χωρητικότητες, για το οποίο παρουσιάζονται οι βασικές παραλλαγές του καθώς και σημαντικές τεχνικές επίλυσης που έχουν επιστρατευτεί για την αντιμετώπισή του.

Στο Κεφάλαιο 3 ορίζεται το πρόβλημα που αποτελεί το κύριο αντικείμενο της παρούσας εργασίας, δηλαδή του Facility Location σε Χρονικά Μεταβαλλόμενους Χώρους. Επιπλέον γίνεται μία λεπτομερής παρουσίαση των δύο βασικότερων εργασιών που προσφέρουν τις πρώτες μεθόδους προσέγγισης της βέλτιστης λύσης του προβλήματος.

Στο Κεφάλαιο 4 παραθέτονται αποτελέσματα και επιλεγμένα μέρη κάποιων εργασιών τα οποία θα αποτελέσουν τη θεωρητική βάση ή/και την πηγή έμπνευσης για τους καινούριους αλγόριθμους που προτείνονται στη συνέχεια της εργασίας.

Στο Κεφάλαιο 5 λαμβάνει χώρα ο σχεδιασμός και η πλήρης παρουσίαση δύο νέων αλγόριθμων που προτείνονται για το δυναμικό Facility Location πρόβλημα. Οι αλγόριθμοι αυτοί αναλύονται για τον τρόπο λειτουργίας τους και εντοπίζονται πιθανές αδυναμίες τους που προβλέπονται θεωρητικά.

Στο Κεφάλαιο 6 γίνεται η πειραματική αξιολόγηση και σύγκριση, σε πληθώρα σεναρίων και για διάφορα κριτήρια, όλων των αλγόριθμων που έχουν παρουσιαστεί και αφορούν το πρόβλημα Facility Location σε Χρονικά Μεταβαλλόμενους Χώρους.

Στο Κεφάλαιο 7 δίνονται τα συμπεράσματα και η συνεισφορά αυτής της διπλωματικής εργασίας, καθώς και προτάσεις για συνέχιση της έρευνας.

Τέλος, δίνονται όλες οι σχετικές βιβλιογραφικές αναφορές.

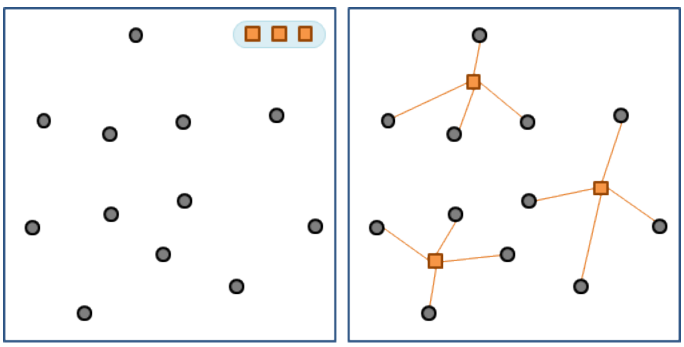
# Κεφάλαιο 2

# Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό γίνεται μία εκτενής παρουσίαση του θεωρητικού υποβάθρου που κρίνεται χρήσιμο για την κατανόηση και την ιστορική πλαισίωση του κύριου σώματος της εργασίας. Αρχικά παρουσιάζεται το πρόβλημα Facility Location χωρίς χωρητικότητες και εν συνεχεία οι διάφορες παραλλαγές αυτού. Τέλος, δίνονται οι κυριότερες τεχνικές και προσεγγίσεις επίλυσής του.

## 2.1 Facility Location

Ο όρος Facility Location στην πραγματικότητα αναφέρεται σε μία οικογένεια προβλημάτων. Σε αυτά τα προβλήματα έχουμε δύο είδη οντοτήτων που ονομάζονται clients και facilities. Το κοινό σημείο των προβλημάτων της οικογένειας αυτής είναι το γεγονός ότι αναζητούν μια ανάθεση από τους clients στα facilities που να έχει ελάχιστο κόστος ως προς κάποιο κριτήριο. Στο Σχήμα 2.1 φαίνεται μια αφαιρετική απόδοση ενός αντιπροσωπευτικού προβλήματος τέτοιας μορφής και μια λύση του που αναθέτει τους clients στα facilities με κάποιο κριτήριο ελαχιστοποίησης των αποστάσεων.



Σχήμα 2.1. Στιγμιότυπο Facility Location προβλήματος και λύση.
Γκρι: clients, Πορτοκαλί: facilities

Στην υποενότητα που ακολουθεί δίνεται ο ορισμός του κυριότερου και πιο ευρέως μελετημένου προβλήματος της οικογένειας προβλημάτων Facility Location.

### 2.1.1 Ορισμός του προβλήματος

Σε αυτό το πρόβλημα μας δίνονται δύο σύνολα $F$ και $C$, όπου το σύνολο $F$ περιέχει τα facilities και το σύνολο $C$ περιέχει τους clients. Επιπλέον, μας δίνεται μία μη αρνητική τιμή $f_i$ για κάθε facility $i$ που αντιπροσωπεύει το κόστος λειτουργίας του, καθώς και μία συνάρτηση $d : F \times C \to \mathbb{R}^+$ η οποία αντιστοιχεί κάθε ζεύγος facility και client σε μία τιμή κόστους σύνδεσης των δύο οντοτήτων.
Αναζητούμε ένα σύνολο $S \subseteq F$ από facilities (τα οποία ονομάζουμε ανοιχτά) και μία αντιστοίχιση $\sigma : C \to S$ από τους clients στα facilities τέτοια ώστε να ελαχιστοποιείται η παράσταση:

$$\sum_{i \in S} f_i + \sum_{j \in C} d(\sigma(j), j)$$

Για διευκόλυνση στην περιγραφή και μοντελοποίηση του προβλήματος μπορούμε να εισάγουμε τις μεταβλητές - δείκτες $y_i$ και $x_{ij}$ που παίρνουν τιμές 0 ή 1. Η $y_i$ λαμβάνει την τιμή 1 αν το facility με δείκτη $i$ είναι ανοιχτό ενώ το $x_{ij}$ λαμβάνει την τιμή 1 αν ο client $j$ είναι συνδεδεμένος με το facility $i$. Σε κάθε άλλη περίπτωση οι μεταβλητές αυτές λαμβάνουν την τιμή 0. Δεδομένων των παραπάνω, πρέπει να αντιστοιχίσουμε κάθε client σε ένα ανοιχτό facility με στόχο να ελαχιστοποιήσουμε την κάτωθι παράσταση:

$$\sum_{i \in F} f_i y_i + \sum_{j \in C} \sum_{i \in F} x_{ij} d(i, j)$$

Βάσει των άνωθι συμβολισμών, το πρόβλημα μπορεί επίσης να μοντελοποιηθεί ως ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού με τους παρακάτω περιορισμούς:

$$\begin{cases} (\forall ij)\ x_{ij} \leq y_i \\ (\forall j)\ \sum_{i \in F} x_{ij} = 1 \\ y_i, x_{ij} \in \{0, 1\} \end{cases}$$

Το πρόβλημα αυτό στη διεθνή βιβλιογραφία ονομάζεται Uncapacitated Facility Location Problem, δηλαδή Facility Location χωρίς χωρητικότητες, και έχει μελετηθεί εκτενώς στα πλαίσια του κλάδου της επιχειρησιακής έρευνας (Operations research). Για την μετρική εκδοχή του προβλήματος (βλ. Εν. 2.1.2), ο καλύτερος γνωστός πολυωνυμικός αλγόριθμος έχει σταθερό παράγοντα προσέγγισης 1.488 [16], μία βελτίωση από το προηγούμενο 1.5 [5], ενώ έχει αποδειχθεί ότι δεν μπορεί να υπάρξει πολυωνυμικός αλγόριθμος με παράγοντα προσέγγισης μικρότερο από 1.463 υποθέτοντας ότι $NP \notin DTIME[n^{O(\log \log n)}]$ [11].

**NP-hardness**

Είναι εύκολο να δει κανείς, ότι το παραπάνω Facility Location πρόβλημα αποτελεί γενίκευση του γνωστού Set Cover προβλήματος. Για λόγους πληρότητας, παρατίθεται ο ορισμός αυτού του NP-complete προβλήματος.

- **Set Cover**

Δοθέντος ενός ζεύγους $(U, S)$, όπου το $U$ είναι ένα πεπερασμένο σύνολο και $S$ είναι μία οικογένεια υποσυνόλων του $U$ τέτοια ώστε $\bigcup_{A \in S} A = U$, και δοθέντων επίσης των βαρών $c : S \to \mathbb{R}^+$, αναζητούμε ένα σύνολο $R \subseteq S$ με $\bigcup_{A \in R} A = U$ τέτοιο ώστε το συνολικό βάρος $\sum_{A \in R} c(A)$ να είναι ελάχιστο.

Θα δείξουμε, μέσω της αναγωγής που ακολουθεί, ότι το Facility Location χωρίς Χωρητικότητες ανήκει στην κλάση των NP-hard προβλημάτων.

Έστω ένα στιγμιότυπο $(U, S, c)$ όπως παραπάνω. Ορίζουμε $C := U$, $F := S$ και $f_A = c(A)$ για $A \in S$. Έστω επίσης ότι το κόστος $d_{Aj}$ είναι μηδέν αν $j \in A \in S$ και άπειρο αν $j \in U \setminus A$. $\square$

Διαισθητικά, η παραπάνω αναγωγή θέτει τα στοιχεία του συνόλου $U$ να είναι οι clients και τα υποσύνολα που ορίζονται στο $S$ να είναι τα facilities. Σε κάθε τέτοιο facility αποδίδουμε το κόστος που προβλεπόταν στο αντίστοιχο σύνολο από τη συνάρτηση κόστους $c$. Τέλος, απαιτούμε κάθε client να συνδεθεί με ένα facility που τον "εμπεριέχει" (αλλιώς το κόστος θα ήταν άπειρο) φροντίζοντας έτσι, αφού εξυπηρετηθούν όλοι οι clients, όλα τα στοιχεία του $U$ να αντιπροσωπεύονται στην τελική επιλογή των ανοιχτών facilities-συνόλων.

Η σχέση των δύο αυτών προβλημάτων συνεπάγεται τη μεταφορά των αρνητικών αποτελεσμάτων που έχουν αποδειχθεί για το Set Cover πρόβλημα στο Facility Location χωρίς χωρητικότητες. Κατά συνέπεια, το καλύτερο που μπορούμε να επιτύχουμε για τη γενική εκδοχή του προβλήματος είναι μία λογαριθμική ως προς το μέγεθος της εισόδου προσέγγιση [18] [8]. Βέβαια, όπως θα δούμε στη συνέχεια, κάποιες παραλλαγές ή επιπλέον υποθέσεις καθιστούν το πρόβλημα ευκολότερο και προσεγγίσιμο κατά σταθερό παράγοντα.

### 2.1.2 Παραλλαγές

Πέρα από το πρόβλημα Facility Location χωρίς χωρητικότητες που περιγράφηκε, έχει υπάρξει σημαντική ποσότητα έρευνας στις βασικές παραλλαγές του προβλήματος αυτού. Σε αυτήν την υποενότητα περιγράφονται οι κυριότερες παραλλαγές συνοδευόμενες από σχετικά αποτελέσματα.

#### k-median πρόβλημα

Το πρόβλημα αυτό αφορά σενάρια όπου ο προϋπολογισμός για άνοιγμα facilities είναι περιορισμένος. Στην προκειμένη περίπτωση η είσοδος περιέχει και ένα αριθμό $k \leq |F|$ ο οποίος αποτελεί το όριο του πλήθους των facilities που έχουμε τη δυνατότητα να θέσουμε σε λειτουργία. Για αυτόν τον τύπο προβλήματος υπάρχει ευριστική μέθοδος που εγγυάται μία $3 + 2/p$ προσέγγιση της βέλτιστης λύσης [3].

#### Με ή χωρίς Χωρητικότητες

Άλλες παραλλαγές Facility Location προβλημάτων θέτουν περιορισμούς στον αριθμό των clients που μπορεί να εξυπηρετήσει ένα facility, δηλαδή ορίζουν μία χωρητικότητα (capacity) για τα facilities. Στο Soft-Capacitated Facility Location πρόβλημα μας δίνεται η δυνατότητα να ανοίξουμε αντίγραφα των facilities, πληρώνοντας ωστόσο το ανάλογο κόστος για κάθε αντίγραφο, κάτι το οποίο δεν αποτελεί επιλογή στο Hard-Capacitated Facility Location. Για την Soft Capacities μετρική εκδοχή υπάρχει αλγόριθμος με σταθερό παράγοντα προσέγγισης ίσο με 2 [19].

#### Μετρική και Μη-Μετρική εκδοχή

Συχνά, στην προσπάθεια δημιουργίας καλών προσεγγιστικών αλγόριθμων, εισάγονται στο πρόβλημα κάποιες επιπλέον υποθέσεις που το καθιστούν ευκολότερο. Μία τέτοια παραδοχή είναι ότι οι αποστάσεις μεταξύ των facilities και των clients είναι συμμετρικές και ικανοποιούν την τριγωνική ανισότητα,

$$d(i, j) + d(j, k) \geq d(i, k)$$

Αυτό σημαίνει ότι οι clients και τα facilities ανήκουν σε ένα μετρικό χώρο, γεγονός που δικαιολογεί τον προσδιορισμό μετρική. Μάλιστα, όλοι οι αλγόριθμοι σταθερού παράγοντα προσέγγισης για το Facility Location χωρίς χωρητικότητες βασίζονται σε αυτήν την παραδοχή. Σε περιπτώσεις που δεν ισχύουν αναγκαστικά οι παραπάνω συνθήκες μιλάμε για τη μη-μετρική εκδοχή του προβλήματος.

#### Offline, Online και Incremental

Ο χαρακτηρισμός **Offline** αναφέρεται στο γεγονός ότι κατέχουμε εξ αρχής όλη τη σχετική με το πρόβλημα πληροφορία. Συνεπώς, στο πρόβλημά μας ο αλγόριθμος γνωρίζει όλα τα αιτήματα σύνδεσης εκ των προτέρων και μπορεί να λάβει αποφάσεις εκμεταλλευόμενος το γεγονός αυτό. Κατά αυτή την έννοια το Facility Location χωρίς Χωρητικότητες αποτελεί ένα Offline πρόβλημα.

Αντιθέτως, σε ένα **Online** Facility Location πρόβλημα οι clients εμφανίζονται δυναμικά και ο αλγόριθμος οφείλει να συνδέσει κάθε client σε ένα facility με το που αυτός εμφανιστεί. Επίσης, ο αλγόριθμος στερείται της δυνατότητας αλλαγής των συνδέσεων για τους clients που έχει ήδη επεξεργαστεί. Σε αυτόν τον τύπο προβλήματος, η αξιολόγηση της λύσης γίνεται συνήθως με χρήση ανταγωνιστικής (competitive) ανάλυσης, κατά την οποία η επίδοση του αλγόριθμου συγκρίνεται με την επίδοση ενός αντίστοιχου βέλτιστου Offline αλγόριθμου. Επιπλέον, έχουν θεωρηθεί παραλλαγές του προβλήματος για τις οποίες οι clients δειγματοληπτούνται από κάποια κατανομή.

Στην τελευταία κατηγορία, δηλαδή σε ένα **Incremental** πρόβλημα, όπως και σε ένα Online, τα αιτήματα σύνδεσης των clients έρχονται δυναμικά. Η διαφορά έγκειται στη δυνατότητα κλεισίματος ενός ανοιχτού facility και προώθησης όσων clients συνδέονταν σε αυτό σε ένα νέα ανοιγμένο facility. Δηλαδή, διατηρώντας ή αυξάνοντας τον αριθμό των facilities, δίνεται μία δυνατότητα επαναξιολόγησης και αλλαγής των συνδέσεων που έχουν καθοριστεί μέχρι εκείνο το στάδιο της εκτέλεσης του αλγόριθμου, επιτρέποντας έτσι μεγαλύτερο επίπεδο προσαρμοστικότητας.

### 2.1.3 Τεχνικές επίλυσης

Για τα διάφορα Facility Location προβλήματα έχει χρησιμοποιηθεί επιτυχώς πληθώρα αλγοριθμικών τεχνικών καθώς και συνδυασμοί από διαφορετικές μεθόδους.

Μερικές από τις τεχνικές αυτές είναι:

**• Linear Programming Rounding**
Πολλά Facility Location προβλήματα, όπως άλλωστε είδαμε για το Uncapacitated Facility Location πρόβλημα, μπορούν να μοντελοποιηθούν ως προβλήματα ακέραιου γραμμικού προγραμματισμού, τα οποία είναι γνωστά NP-hard προβλήματα. Ωστόσο, για την κατασκευή ενός προσεγγιστικού αλγόριθμου χρειαζόμαστε έναν αποδεδειγμένα πολυωνυμικό αλγόριθμο. Για αυτό το σκοπό, σε αυτή την τεχνική εκμεταλλευόμαστε το γεγονός ότι τα (μη ακέραια) προβλήματα γραμμικού προγραμματισμού επιλύονται σε πολυωνυμικό χρόνο. Αρχικά, πραγματοποιούμε μία χαλάρωση των συνθηκών που πρέπει να ικανοποιούνται επιτρέποντας στις συμμετέχουσες μεταβλητές να λαμβάνουν τιμές σε ένα διάστημα πραγματικών αριθμών. Αφού επιλυθεί αυτό το νέο πρόβλημα, θα λάβουμε μία κλασματική λύση την οποία θα χρησιμοποιήσουμε για να εξάγουμε μία ακέραια-έγκυρη λύση του προβλήματος που μας αφορά. Σε αυτό το στάδιο, υπεισέρχεται η τεχνική Rounding, η οποία αναλαμβάνει την ακεραιοποίηση των κλασματικών τιμών που παράγονται από την επίλυση του αντίστοιχου προβλήματος γραμμικού προγραμματισμού ώστε να λάβουμε μία έγκυρη λύση. Ο τρόπος με τον οποίο θα γίνει αυτή η ακεραιοποίηση είναι σημαντικός επειδή η διαφορά του κόστους της βέλτιστης λύσης από τη λύση που θα αποκτήσουμε εφαρμόζοντας την μέθοδο, είναι το πολύ ίση με την αύξηση του κόστους που προκύπτει κατά τη διαδικασία της ακεραιοποίησης. Οπότε, οι όποιες

εγγυήσεις για την ποιότητα των λύσεων εξαρτώνται από αυτό το στάδιο.
Η τεχνική αυτή προσφέρει προσεγγίσεις κατά σταθερό παράγοντα για πολλά Facility Location προβλήματα. Για παράδειγμα, για το μετρικό Uncapacitated Facility Location υπάρχει 1.582-προσεγγιστικός LP Rounding αλγόριθμος από τον Sviridenko [22]. Επίσης, οι δύο καλύτεροι γνωστοί αλγόριθμοι, οι οποίοι παρουσιάζουν συντελεστές προσέγγισης 1.488 [16] και 1.5 [5], βασίζονται στην τεχνική σε συνδυασμό με greedy βελτιώσεις. Αξίζει να παρατηρήσουμε πόσο κοντά βρισκόμαστε στο κάτω φράγμα 1.463.

- **Primal-Dual αλγόριθμοι**

Αυτοί οι αλγόριθμοι υπολογίζουν εφικτές λύσεις για το πρωταρχικό (primal) και δυϊκό (dual) LP πρόβλημα ταυτόχρονα. Στην ενότητα 2.1.1 παρουσιάστηκε το πρωταρχικό LP πρόβλημα. Το δυϊκό του ορίζεται ως εξής:

Μεγιστοποίησε το άθροισμα $\sum_{j\in C} u_j$ έτσι ώστε να ικανοποιούνται οι συνθήκες:

$$\begin{cases} u_j - w_{ij} \leq d_{ij} \quad (i \in F, j \in C) \\ \sum_{i\in F} w_{ij} \leq f_i \quad (j \in C) \\ w_{ij} \geq 0 \quad (i \in F, j \in C) \end{cases}$$

Για να κατανοηθεί αυτή η ισοδύναμη μοντελοποίηση του προβλήματος, μπορούμε να θεωρήσουμε ότι οι ποσότητες $u_j$ αντιστοιχούν στο κόστος που πληρώνει ο client $j$ ώστε να συνδεθεί σε κάποιο facility. Οι ποσότητες $w_{ij}$ αναφέρονται στη συνεισφορά του client $j$ για το άνοιγμα του facility $i$.
Κατά την ταυτόχρονη επίλυση των δύο μορφών, συνήθως αρχικοποιούμε τις μεταβλητές του δυϊκού στην τιμή μηδέν. Έπειτα ξεκινάμε να τις αυξάνουμε σταδιακά λαμβάνοντας αποφάσεις σύνδεσης των clients ή ανοίγματος των facilities χωρίς να παραβιάζουμε κάποια συνθήκη, φροντίζοντας έτσι η λύση του πρωταρχικού να είναι ακέραια. Η εγγύηση για την ποιότητα της προσέγγισης είναι απόρροια των συμπληρωματικών συνθηκών χαλαρότητας. Συγκεκριμένα, αρκεί να δείξουμε ότι η λύση του πρωταρχικού που παρήχθη έχει κόστος το οποίο είναι πολλαπλάσιο κατά ένα παράγοντα της λύσης του δυϊκού προβλήματος. Τέλος, επειδή το δυϊκό είναι ένα πρόβλημα μεγιστοποίησης, η λύση που παρήγαγε ο αλγόριθμος για αυτό έχει κόστος το πολύ ίσο με αυτό της βέλτιστης λύσης του προβλήματος.
Υπάρχουν τέτοιοι αλγόριθμοι οι οποίοι είναι πολύ ταχύτεροι από τους LP rounding αλγορίθμους, επειδή δεν χρησιμοποιούν έναν αλγόριθμο γραμμικού προγραμματισμού ως υπορουτίνα. Τέτοιες προσεγγίσεις για το μετρικό Uncapacitated Facility Location πρόβλημα είναι αυτές των Jain και Vazirani με συντελεστή προσέγγισης 3 και χρόνο εκτέλεσης $O(n^2 \log n)$ [24] καθώς και των Charikar και Guha, οι οποίοι χρησιμοποιώντας επιπλέον greedy βελτιστοποίηση, κατάφεραν μία 1.853-προσέγγιση σε χρόνο $O(n^3)$ [6].

- **Τοπική Αναζήτηση**

Άλλοι προσεγγιστικοί αλγόριθμοι για Facility Location προβλήματα βασίζονται στην τεχνική Local Search. Γενικά, η τεχνική αυτή ακολουθεί την εξής λογική: ο αλγόριθμος ξεκινά από μία αυθαίρετη εφικτή λύση και επαναληπτικά βελτιώνει τη λύση κάνοντας ένα βήμα τοπικής αναζήτησης. Έπειτα από ένα πολυωνυμικό αριθμό βημάτων η διαδικασία σταματά. Αυτοί οι αλγόριθμοι χαρακτηρίζονται από μεγάλη ευελιξία και έχουν συνεισφέρει προσεγγίσεις με σταθερό παράγοντα προσέγγισης για πληθώρα Facility Location προβλημάτων που μάλιστα είναι συγκρίσιμες με αυτές από άλλες τεχνικές. Η αξία αυτών των τεχνικών συναντάται στην συχνά εύκολη υλοποίησή τους και διαδεδομένη χρήση τους καθώς και στο γεγονός ότι για κάποιες παραλλαγές Facility Location προβλημάτων είναι οι μόνες που προσφέρουν προσεγγίσεις σταθερού παράγοντα. Συγκεκριμένα, για το μετρικό Uncapacitated Facility Location πρόβλημα υπάρχει ένας $5+\epsilon$ προσεγγιστικός αλγόριθμος τοπικής αναζήτησης με χρόνο εκτέλεσης $O(n^6 \log(n/\epsilon))$ από τους Korupolu et al. [15].

# Κεφάλαιο 3

# Περιγραφή Θέματος

## 3.1 Facility Location σε Χρονικά Μεταβαλλόμενους Χώρους

Εδώ θα επικεντρωθούμε στο δυναμικό πρόβλημα Facility Location όπως εισήχθη στην εργασία Facility Location in Evolving Metrics των Eisenstat, Mathieu και Schabanel [7]. Ουσιαστικά, πρόκειται για μία γενίκευση του αντίστοιχου στατικού προβλήματος που ορίσθηκε στο προηγούμενο κεφάλαιο, αφού πλέον θεωρούμε ότι οι clients κινούνται στο χώρο και επομένως η απόστασή τους από τα facilities μεταβάλλεται χρονικά. Επιπροσθέτως, απαιτούμε μια λύση η οποία να χαρακτηρίζεται από μία έννοια σταθερότητας ή διαχρονικής δυσμεταβλητότητας. Η τελευταία απαίτηση μοντελοποιείται με την εισαγωγή μιας τιμής κόστους $g$ για κάθε αντιστοίχιση κάποιου client σε διαφορετικό facility από αυτό στο οποίο ήταν αντιστοιχισμένος την προηγούμενη χρονική στιγμή.

### 3.1.1 Ορισμός του προβλήματος

Μας δίνεται ένα σύνολο $F$ από $m$ facilities και ένα σύνολο $C$ από $n$ clients μαζί με μία πεπερασμένη ακολουθία συναρτήσεων $d_t : F \times C \to \mathbb{R}^+$ και δύο μη αρνητικές τιμές $f$ και $g$. Η τιμή $f$ αντιπροσωπεύει το κόστος λειτουργίας ενός facility ενώ η τιμή $g$ αντιπροσωπεύει το κόστος αλλαγής ανάθεσης κάποιου client από μία χρονική στιγμή στην επόμενη. Ο στόχος είναι να παράγουμε ως έξοδο μία ακολουθία υποσυνόλων $A_t \subseteq F$ από facilities καθώς και, για κάθε χρονική στιγμή $t \in [T]$ μία αντιστοίχηση, $\phi_t : C \to A_t$ από τους clients στα facilities, τέτοια ώστε να ελαχιστοποιείται η παράσταση:

$$f \cdot \sum_{1 \le t \le T} |A_t| + \sum_{1 \le t \le T} \sum_{j \in C} d_t(\phi_t(j), j) + g \cdot \sum_{1 \le t < T} \sum_{j \in C} 1\{\phi_t(j) \neq \phi_{t+1}(j)\}$$

Το πρόβλημα μπορεί να γενικευθεί για διαφορετικές τιμές κόστους λειτουργίας για κάθε facility $i$ σε κάθε χρόνο $t$, αν εισάγουμε τις μεταβλητές εισόδου $f_i^t$.

### 3.1.2 Μία αναγωγή για επαρκώς μεγάλες τιμές κόστους αλλαγής ανάθεσης g

Θα δείξουμε ότι για επαρκώς μεγάλες τιμές του κόστους αλλαγής ανάθεσης $g$, το πρόβλημα Facility Location σε Χρονικά Μεταβαλλόμενους Χώρους ανάγεται στο πρόβλημα Facility Location χωρίς χωρητικότητες.

Αν για κάθε client $j \in C$ και κάθε facility $i \in F$ ισχύει ότι

$$f(T-1) + \sum_{2 \leq t \leq T} d_t(i,j) < g$$

τότε το πρόβλημα είναι ισοδύναμο με το ακόλουθο στιγμιότυπο του Uncapacitated Facility Location προβλήματος:

Διατηρούμε τα ίδια σύνολα $F$ και $C$, θέτουμε $f_i = fT$ για κάθε facility $i \in F$ και ορίζουμε τη συνάρτηση $d$ ως εξής:

$$d(i,j) = \sum_{1 \leq t \leq T} d_t(i,j)$$

**Απόδειξη**

Αρκεί να δείξουμε ότι στη βέλτιστη λύση δεν πραγματοποιείται καμία αλλαγή ανάθεσης. Αν $m = 1$ ή $T = 1$ το ζητούμενο είναι προφανές, αλλιώς:
Έστω μία βέλτιστη λύση με κόστος $OPT$ και έστω επίσης ότι σε κάποια χρονική στιγμή $t' < T$ και για κάποιον client $j$ ισχύει ότι $\phi_{t'}(j) \neq \phi_{t'+1}(j)$.
Από την υπόθεση, ισχύει ότι

$$f \cdot (T - t') + \sum_{t'+1 \leq t \leq T} d_t(i,j) \leq f(T-1) + \sum_{2 \leq t \leq T} d_t(i,j) < g$$

Συνεπώς, αν ο client $j$ διατηρούσε σταθερή ανάθεση από τη στιγμή $t'$ έως τέλους θα παραγόταν μία λύση με κόστος

$$cost \leq OPT - g + f(T - t') + \sum_{t'+1 \leq t \leq T} d_t(i,j) < OPT$$

.

Άτοπο, αφού έχουμε υποθέσει ότι η λύση είναι βέλτιστη. Συνεπώς η βέλτιστη λύση δεν περιλαμβάνει καμία αλλαγή. □

### 3.1.3 Μία αναγωγή για επαρκώς μεγάλες τιμές κόστους λειτουργίας f των facilities

Με παρόμοια λογική με αυτήν που ακολουθήθηκε για την τιμή $g$, θα δείξουμε ότι για επαρκώς μεγάλες τιμές κόστους λειτουργίας f των facilities, το πρόβλημα Facility Location σε Χρονικά Μεταβαλλόμενους Χώρους ανάγεται στο πρόβλημα Facility Location χωρίς χωρητικότητες.

Αν για κάθε χρονική στιγμή $t$ και ισχύει ότι

$$\sum_{j \in C}(2g + \max_{i \in F} d_t(i,j) - \min_{i \in F} d_t(i,j)) < f$$

τότε το πρόβλημα είναι ισοδύναμο με το ακόλουθο στιγμιότυπο του Uncapacitated Facility Location προβλήματος:

Διατηρούμε το ίδιο σύνολο $C$ και κατασκευάζουμε ένα νέο σύνολο $F'$ από $m^T$ facilities όπου κάθε ένα αντιπροσωπεύει ένα συνδυασμό από επιλογές για ένα ανοιχτό facility του συνόλου $F$ ανά χρονική στιγμή. Συμβολίζουμε το νέο facility ως μια πλειάδα από $T$ facilities του συνόλου $F$, ως εξής:

$$i = (i_1, ..., i_T), \quad i \in F', \ i_k \in F, \ k \in \{1, ..., T\}$$

Θέτουμε $f_i = fT + gn \sum_{1 \leq t < T} 1\{i_t \neq i_{t+1}\}$ για κάθε facility $i \in F'$. Τέλος, ορίζουμε τη συνάρτηση $d$ ως εξής:

$$d(i,j) = \sum_{1 \leq t \leq T} d_t(i_t, j)$$

**Απόδειξη**

Αρκεί να δείξουμε ότι στη βέλτιστη λύση δεν είναι ποτέ ανοιχτά δύο facilities ταυτόχρονα. Αν $m = 1$ ή $n = 1$ το ζητούμενο είναι προφανές, αλλιώς:
Έστω μία βέλτιστη λύση με κόστος $OPT$ και έστω επίσης ότι σε κάποια χρονική στιγμή $t$ υπάρχουν δύο μη κενά και ξένα σύνολα από clients, $C_1$ και $C_2$, των οποίων τα μέλη είναι συνδεδεμένα στα facilities $i$ και $i'$ αντίστοιχα ($i \neq i'$).

Όμως, αν αναθέταμε όλους τους clients του συνόλου $C_2$ στο facility $i$ θα παραγόταν μία λύση με κόστος

$$\begin{aligned} cost &\leq OPT - f + \sum_{j \in C_2} (2g + d_t(i,j) - d_t(i',j)) \\ &\leq OPT - f + \sum_{j \in C_2} (2g + \max_{i \in F} d_t(i,j) - \min_{i \in F} d_t(i,j)) \\ &\leq OPT - f + \sum_{j \in C}(2g + \max_{i \in F} d_t(i,j) - \min_{i \in F} d_t(i,j)) < OPT. \end{aligned}$$

Άτοπο, αφού έχουμε υποθέσει ότι η λύση είναι βέλτιστη. Συνεπώς στη βέλτιστη λύση δεν είναι ποτέ ανοιχτά δύο facilities για κάποια στιγμή $t$. □

Μάλιστα, στην προκείμενη περίπτωση η λύση είναι τετριμμένη αφού θα χρησιμοποιηθεί μόνο ένα εκ των facilities του συνόλου $F'$.

### 3.1.4 Ο Αλγόριθμος των Eisenstat, Mathieu, Schabanel

Ο αλγόριθμος αυτός βασίζεται στην τεχνική LP rounding. Επομένως, το πρώτο βήμα του συνίσταται στην επίλυση ενός προβλήματος γραμμικού προγραμματισμού. Για την μοντελοποίηση του προβλήματος σε όρους γραμμικού προγραμματισμού, εισάγουμε τις μεταβλητές - δείκτες $y_i^t$, $x_{ij}^t$, και $z_{ij}^t$ για $i \in F$, $j \in C$, και $t \in [T]$: $y_i^t = 1$ ανν το facility $i$ είναι ανοιχτό τη χρονική στιγμή $t$, $x_{ij}^t = 1$ ανν ο client $j$ είναι συνδεδεμένος με το facility $i$ τη χρονική στιγμή $t$, και $z_{ij}^t = 1$ ανν ο client $j$ είναι συνδεδεμένος με το facility $i$ τη χρονική στιγμή $t$ αλλά παύει να είναι τη χρονική στιγμή $t+1$. Τότε το δυναμικό Facility Location πρόβλημα είναι ισοδύναμο με το να βρούμε μια ακέραια λύση για το παρακάτω πρόβλημα γραμμικού προγραμματισμού.

Ελαχιστοποίησε την παράσταση

$$f \sum_{1 \le t \le T} \sum_{i \in F} y_i^t + \sum_{j \in C} \sum_{1 \le t \le T} \sum_{i \in F} x_{ij}^t d_t(i,j) + g \sum_{j \in C} \sum_{1 \le t < T} \sum_{i \in F} z_{ij}^t$$

έτσι ώστε

$$\begin{cases} (\forall ijt)\ x_{ij}^t \le y_i^t \\ (\forall jt)\ \sum_{i \in F} x_{ij}^t = 1 \\ (\forall ij, \forall t < T)\ z_{ij}^t \ge x_{ij}^t - x_{ij}^{t+1} \\ y_i^t, x_{ij}^t, z_{ij}^t \ge 0 \end{cases}$$

Αφού επιλύσουμε το LP θα λάβουμε μία έξοδο, έστω $(x, y, z)$, της οποίας οι μεταβλητές λαμβάνουν τιμές στο διάστημα $[0, 1]$. Το επόμενο βήμα είναι ο μετασχηματισμός αυτής της εξόδου ώστε όλες οι μεταβλητές - δείκτες να ανήκουν στο επιθυμό σύνολο $\{0, 1\}$. Αυτό επιθυμούμε να γίνει με όσο το δυνατόν μικρότερη αύξηση κόστους αφού αυτή η αύξηση θα καθορίσει την ποιότητα των λύσεων του συνολικού αλγόριθμου. Χρησιμοποιώντας την έξοδο $(x, y, z)$ καθώς και τον πιθανοτικό αλγόριθμο που ακολουθεί παράγουμε μία λύση του προβλήματος.

**Προσεγγιστικός Αλγόριθμος**

*Έστω $(x, y, z)$ η λύση που λάβαμε επιλύοντας το πρόβλημα γραμμικού προγραμματισμού LP.*

*Για κάθε facility $i$, επίλεξε ένα τυχαίο κατώφλι $\rho_i$ από μία εκθετική κατανομή με αναμενόμενη τιμή $1/(2\log(2nT))$: δηλ. $Pr\{\rho_i > a\} = e^{-2a\log(2nT)}$, $\forall a > 0$. Άνοιξε το facility $i$ σε κάθε χρόνο $t$ όπου ισχύει ότι $y_i^t > \rho_i$ . Για κάθε χρόνο $t$, έστω $A_t$ το σύνολο από ανοιχτά facilities.*

***Για κάθε** client $j$ **επανάλαβε***

*Καθόρισε πότε θα πρέπει να αλλάξει από ένα facility σε ένα άλλο χρησιμοποιώντας τις μεταβλητές $z$, και σύνδεσέ τον στο φθηνότερο επιλεχθέν facility ανάμεσα στις αλλαγές:*

*(α) Διαμέρισε το χρόνο σε $l_j$ διαστήματα $[t_k^j, t_{k+1}^j)$ τ.ω $t_1^j = 1$ και όπου το $t_{k+1}^j$ ορίζεται αναδρομικά ως το μέγιστο $t \in (t_k^j, T+1]$ με $\sum_{i \in F}(min_{t_k^j \leq u < t}\ x_{ij}^u) \geq 1/2$, και $t_{l_j+1}^j = T+1$.*

*(β) Για κάθε χρονικό διάστημα $I = [t_k^j, t_{k+1}^j)$ και facility $i$, έστω $x_{ij}^I = min_{t \in I}\ x_{ij}^t$. Σύνδεσε τον client $j$ στο facility $i \in A_t$ το οποίο ελαχιστοποιεί το λόγο $\rho_i / x_{ij}^I$.*

***Τέλος επανάληψης***

Αυτός ο αλγόριθμος δεν απαιτεί οι αποστάσεις να ικανοποιούν την τριγωνική ανισότητα. Συγκεκριμένα, αποδεικνύεται ότι παράγει μία λύση η οποία είναι μία $O(\log(nT))$-προσέγγιση της βέλτιστης με σταθερή θετική πιθανότητα. Επιπλέον, μπορεί να αποδειχθεί ότι το κόστος της λύσης ικανοποιεί την ανισότητα $Pr\{cost \leq 4\log(2nT)LP\} \geq 1/4$. Αναλυτικά, το αναμενόμενο κόστος για τα facilities είναι μικρότερο από $4\log(2nT)$ φορές ο αντίστοιχος όρος στο LP, για τις αλλαγές ανάθεσης μικρότερο από 2 φορές ενώ έδειξαν επίσης ότι το αναμενόμενο κόστος σύνδεσης είναι το πολύ $4\log(2nT)$ φορές ο αντίστοιχος όρος στο LP.

### Απόδειξη νέου άνω φράγματος σταθερού συντελεστή για το κόστος σύνδεσης

Σε αυτήν την ενότητα, θα αποδείξουμε ότι ο αλγόριθμος παράγει μία λύση με κόστος σύνδεσης που φράσσεται από το $6 \cdot OPT$. Υπενθυμίζουμε ότι οι Eisenstat et al έδειξαν ότι το αναμενόμενο κόστος σύνδεσης είναι το πολύ $4\log(2nT)$ φορές ο αντίστοιχος όρος στο LP. Για τον σκοπό αυτό θα χρησιμοποιήσουμε το δυϊκό LP του προβλήματος, το οποίο είναι το παρακάτω:

Μεγιστοποίησε την παράσταση

$$\sum_{1\leq t\leq T}\sum_{j\in C}\alpha_j^t$$

έτσι ώστε

$$\begin{cases} (\forall ij, t>1) \quad d_{ij}^t \geq \alpha_j^t - \beta_{ij}^t - \gamma_{ij}^t + \gamma_{ij}^{t-1} \\ (\forall ij, t=1) \quad d_{ij}^t \geq \alpha_j^t - \beta_{ij}^t - \gamma_{ij}^t \\ (\forall it) \quad f_i^t \geq \sum_{j\in C}\beta_{ij}^t \\ (\forall ij, t<T) \quad g \geq \gamma_{ij}^t, \quad (\forall ij, t=T) \quad \gamma_{ij}^t = 0 \\ (\forall ijt) \quad \alpha_j^t, \beta_{ij}^t, \gamma_{ij}^t \geq 0 \end{cases}$$

Πρώτα, παρατηρούμε πως με πιθανότητα μεγαλύτερη ή ίση του 1/2, για όλα τα $l_j$ διαστήματα $I = [t_k^j, t_{k+1}^j)$ που υπολογίζει ο αλγόριθμος για κάθε client $j$, το facility $i$ που θα επιλεχθεί θα έχει $x_{ij}^I > 0$, γεγονός που συνεπάγεται ότι όλες οι τιμές $x_{ij}^t$ στο διάστημα $I$ θα είναι μεγαλύτερες του μηδενός. Αυτό αποκαλούμε συμβάν Β.

Από τις συνθήκες συμπληρωματικής χαλαρότητας, αν $x_{ij}^t > 0$ τότε
αν $t > 1$, $\quad d_{ij}^t = \alpha_j^t - \beta_{ij}^t - \gamma_{ij}^t + \gamma_{ij}^{t-1} \leq \alpha_j^t - \gamma_{ij}^t + \gamma_{ij}^{t-1}$
αν $t = 1$, $\quad d_{ij}^t = \alpha_j^t - \beta_{ij}^t - \gamma_{ij}^t \leq \alpha_j^t - \gamma_{ij}^t$

Αθροίζοντας για όλες τις χρονικές στιγμές $t$ για κάποιον client $j$ λαμβάνουμε

$$\sum_{1\leq t\leq T} d_{\phi_t(j)j}^t \leq \sum_{1\leq t\leq T}\alpha_j^t + \sum_{1\leq k\leq l_j}\gamma_{\phi_{t_{k+1}^j}(j)j}^{t_{k+1}^j-1}$$

Στο τελευταίο άθροισμα έχουμε $l_j - 1$ μη μηδενικούς $\gamma$ όρους, ίσους στο πλήθος με τον αριθμό αλλαγών ανάθεσης που είναι το πολύ 2 φορές ο αντίστοιχος όρος στο LP. Κατά συνέπεια ισχύει ότι

$$\sum_{1\leq t\leq T} d_{\phi_t(j)j}^t \leq \sum_{1\leq t\leq T}\alpha_j^t + 2g\sum_{1\leq t\leq T}\sum_{i\in F} z_{ij}^t$$

Αθροίζοντας για όλους τους clients καταλήγουμε ότι

$$\sum_{1 \le t \le T} \sum_{j \in C} d^t_{\phi_t(j)j} \le \sum_{1 \le t \le T} \sum_{j \in C} \alpha^t_j + 2g \sum_{1 \le t \le T} \sum_{i \in F} \sum_{j \in C} z^t_{ij}$$

$$\sum_{1 \le t \le T} \sum_{j \in C} d^t_{\phi_t(j)j} \le OPT + 2 \cdot OPT = 3 \cdot OPT$$

Τέλος, άροντας τη συνθήκη στο συμβάν B με χρήση της ανισότητας Markov λαμβάνουμε το τελικό άνω φράγμα $6 \cdot OPT$.
Ως φυσικό επακόλουθο, στα πλαίσια αυτής της εργασίας, τέθηκε το εξής ερώτημα: Υπάρχει αλγόριθμος ο οποίος να είναι μία $O(1)$-προσέγγιση της βέλτιστης λύσης; Όπως θα δούμε στην επόμενη ενότητα, στη μετρική περίπτωση, η απάντηση είναι καταφατική.

### 3.1.5 Ο Αλγόριθμος των An, Norouzi-Fard, Svensson

Ο αλγόριθμος αυτός, όπως δημοσιεύτηκε στην εργασία Dynamic Facility Location via Exponential Clocks των An, Norouzi-Fard και Svensson [2], εισάγει μία νέα μέθοδο στρογγυλοποίησης της μη ακέραιας λύσης που λαμβάνουμε από το LP και αποτελεί τον πρώτο προσεγγιστικό αλγόριθμο με σταθερό λόγο προσέγγισης για το δυναμικό πρόβλημα Facility Location. Ο αλγόριθμός τους τοποθετεί ανταγωνιζόμενα εκθετικά ρολόγια στους clients και τα facilities, και συνδέει κάθε client με μία διαδρομή η οποία ακολουθεί συστηματικά το μικρότερο ρολόι στη γειτονιά.

Η πρώτη προεπεξεργασία προέρχεται από τον $O(\log(nT))$-προσεγγιστικό αλγόριθμο των Eisenstat et al.

**Λήμμα 2.1.** Δοθείσας μίας λύσης από το LP, μπορούμε να αποκτήσουμε σε πολυωνυμικό χρόνο, αυξάνοντας το κόστος το πολύ κατά έναν παράγοντα 2, μία εφικτή λύση $(x, y, z)$ η οποία να ικανοποιεί το εξής:
• Έστω $Z^t = \{j \in C | x_{ij}^t \neq x_{ij}^{t+1}$ για κάποιο $i \in F\}$ το σύνολο των clients που άλλαξαν την κλασματική σύνδεσή τους μεταξύ της χρονικής στιγμής $t$ και $t+1$. Τότε $\sum_{t=1}^{T} |Z^t| \leq \sum_{t=1}^{T} \sum_{i \in F, j \in C} z_{ij}^t$.

Το δεύτερο βήμα της προεπεξεργασίας προέρχεται από τη μέθοδο δημιουργίας αντιγράφων των facilities.

**Παρατήρηση 2.2.** Χωρίς βλάβη της γενικότητας, μπορούμε να υποθέσουμε ότι η $(x, y, z)$ ικανοποιεί τα ακόλουθα:
1. Για κάθε facility $i \in F$, client $j \in C$, και χρονικό βήμα $t \in [T]$, $x_{ij}^t \in \{0, y_i^t\}$.
2. Για κάθε facility $i \in F$, υπάρχει ένα $o_i \in [0, 1]$ τέτοιο ώστε $y_i^t \in \{0, o_i\}$ για κάθε χρονικό βήμα $t \in [T]$.

#### Περιγραφή του Αλγόριθμου

Δοθείσας μία προεπεξεργασμένης λύσης $(x, y, z)$ του προβλήματος γραμμικού προγραμματισμού LP που ικανοποιεί το Λήμμα 2.1 και την Παρατήρηση 2.2, ο αλγόριθμος ξεκινά κάνοντας μία τυχαία επιλογή, και έπειτα ανοίγει facilities και συνδέει clients σε κάθε χρονικό βήμα.
**Τυχαία επιλογή:** Για κάθε facility $i \in F$ δειγματοληπτούμε ανεξάρτητα ένα εκθετικό ρολόι $Q_i \sim Exp(o_i)$, και για κάθε client $j \in C$ δειγματοληπτούμε ανεξάρτητα ένα εκθετικό ρολόι $R_j \sim Exp(1)$.
**Άνοιγμα και σύνδεση:** Σε κάθε χρονική στιγμή $t \in [T]$, ανοίγουμε facilities και συνδέουμε clients ως εξής. Θεώρησε τους clients με βάση την αύξουσα σειρά που υποδεικνύουν τα δειγματοληφθέντα ρολόγια τους (τα $R_j$). Όταν φτάσεις στον client $j \in C$, βρες το facility $i = arg\ min_{i: x_{ij}^t > 0} Q_i$ με το μικρότερο ρολόι ανάμεσα στα facilities με τα οποία ο $j$ είναι

συνδεδεμένος στο $x^t$. Ομοίως, έστω $j' = arg\ min_{j':x^t_{ij'}>0} R_{j'}$ ο client με το μικρότερο ρολόι στη γειτονιά του $i$.
Δεδομένων των παραπάνω, η σύνδεση του $j$ τη χρονική στιγμή $t$ καθορίζεται ως εξής:
Αν $j = j'$ τότε άνοιξε το $i$ και σύνδεσε τον $j$ στο $i$, αλλιώς, σύνδεσε τον $j$ στο ίδιο facility με τον $j'$.

Ο παραπάνω πιθανοτικός αλγόριθμος αποτελεί στη μετρική περίπτωση, δηλαδή όταν οι αποστάσεις ικανοποιούν την τριγωνική ανισότητα, μία 14-προσέγγιση του προβλήματος λαμβάνοντας υπόψιν και την απώλεια ενός παράγοντα 2 στην προεπεξεργασία. Συγκεκριμένα, για κάθε χρονική στιγμή $t$ ισχύουν οι εξής σχέσεις:

$$\mathrm{E}[\sum_{i\in F} f_i^t Y_i^t] \leq 2\sum_{i\in F} f_i^t y_i^t$$

$$\mathrm{E}[\sum_{i\in F}\sum_{j\in C} d_t(i,j) X_{ij}^t] \leq 12\sum_{i\in F}\sum_{j\in C} d_t(i,j) x_{ij}^t$$

$$\mathrm{E}[g\sum_{j\in C}\sum_{i\in F} Z_{ij}^t] \leq 14g\sum_{j\in C}\sum_{i\in F} z_{ij}^t$$

όπου οι μεταβλητές - δείκτες $(X, Y, Z)$ αντιστοιχούν στην τελική ακέραια λύση ενώ ως $(x, y, z)$ συμβολίζεται η μη προεπεξεργασμένη έξοδος της επίλυσης του γραμμικού προβλήματος.

# Κεφάλαιο 4

# Σχετικές Εργασίες

Στο προηγούμενο κεφάλαιο γνωρίσαμε μεταξύ άλλων τις δύο κυριότερες εργασίες που αφορούν το δυναμικό Facility Location πρόβλημα, οι οποίες ήταν συνοδευόμενες από δύο προσεγγιστικούς αλγόριθμους επίλυσης του προβλήματος αυτού. Χτίζοντας πάνω σε αυτήν την προσπάθεια, προτείνουμε δύο αλγόριθμους οι οποίοι, αν και νέοι στη σύλληψη, βασίζονται ισχυρά σε προϋπάρχουσες τεχνικές αντιμετώπισης παρεμφερών προβλημάτων. Το παρόν κεφάλαιο αφιερώνεται στην παρουσίαση των σχετικών εργασιών από τις οποίες έγινε άντληση ιδεών ενώ στο επόμενο κεφάλαιο γίνεται η προσαρμογή των ιδεών αυτών στις ανάγκες του προβλήματος που μας αφορά.

## 4.1 Μια γενική προσέγγιση σε online προβλήματα βελτιστοποίησης δικτύων

Σε αυτή την εργασία των Alon, Awerbuch, Azar, Naor και Buchbinder [1] μελετάται ένα εύρος από online προβλήματα σε γραφήματα καθώς και προβλήματα βελτιστοποίησης δικτύων. Μεταξύ άλλων, παρουσιάζουν έναν γενικό ντετερμινιστικό $O(\log m)$-αλγόριθμο για την παραγωγή μίας κλασματικής λύσης που ικανοποιεί την online συνδεσμολογία του εκάστοτε γραφήματος, όπου με m συμβολίζουμε τον αριθμό των ακμών του γραφήματος.

Ο αλγόριθμος της τεχνικής αυτής παρουσιάζεται παρακάτω:
Αρχικά, τα κλασματικά βάρη όλων των ακμών αρχικοποιούνται στην τιμή $1/2m^3$. Έστω ότι ο αλγόριθμος δέχεται ένα νέο αίτημα $(S, T)$. Τότε πράττουμε ως εξής:

1. Αν η μέγιστη ροή από το S στο T είναι τουλάχιστον ίση με 1, δεν κάνουμε τίποτα.

2. Αλλιώς, όσο η ροή μεταξύ του S και του T είναι μικρότερη του 1, ανανεώνουμε τα βάρη ως εξής:
- Υπολογίζουμε μία ελάχιστη τομή C μεταξύ των S και T.
- Για κάθε ακμή $e \in C$, $w_e \leftarrow w_e(1 + 1/c_e)$.

Η παραπάνω προσέγγιση μπορεί να χρησιμοποιηθεί για την επίλυση των Multicast και Non-metric Facility Location προβλημάτων. Το Multicast πρόβλημα αποτελεί γενίκευση του Non-metric Facility Location και του Set Cover προβλήματος. Ο ορισμός του ακολουθεί:

**Πρόβλημα Multicast**

Έστω $X = \{1, 2, ..., n\}$ ένα σύνολο γειωμένων τερματικών, και έστω $T = \{T_1, T_2, ..., T_m\}$ μια οικογένεια ριζωμένων δένδρων με συνάρτηση κόστους $c : E \to R^+$ που ορίζεται στις ακμές. Κάθε φύλλο του δένδρου σχετίζεται με ένα υποσύνολο των τερματικών, όπου κάθε τερματικό $i$ ανήκει στο πολύ ένα φύλλο σε κάθε ένα από τα δένδρα. Αναζητούμε ένα σύνολο ριζωμένων υποδένδρων $T' = \{T'_1, T'_2, ..., T'_m\}$, όπου $T'_i \subseteq T_i$, $(1 \leq i \leq m)$, των οποίων η ένωση να καλύπτει το σύνολο $X$ και το κόστος των οποίων να είναι ελάχιστο. Το κόστος ορίζεται ως το άθροισμα του κόστους των ακμών στα επιλεχθέντα υποδένδρα $T'$.

Οι Alon et Al. περιγράφουν έναν πιθανοτικό αλγόριθμο για το Multicast πρόβλημα. Έπειτα από κάθε αίτημα, αρχικά υπολογίζουν μία $O(\log m)$ ανταγωνιστική κλασματική λύση. Στη συνέχεια, μέσω μιας τεχνικής rounding λαμβάνουν μία λύση του προβλήματος η οποία αποτελεί μία $O(\log n' \log m)$-προσέγγιση της βέλτιστης, όπου με $n'$ συμβολίζεται ο αριθμός των τερματικών για τα οποία υπήρξαν αιτήματα. Ως συνέπεια αυτού και δεδομένου ότι το Non-metric Facility Location πρόβλημα αποτελεί ειδική περίπτωση του Multicast προβλήματος αποδεικνύεται το παρακάτω

**Θεώρημα.** Υπάρχει ένας $O(\log n \log m)$ ανταγωνιστικός αλγόριθμος για το Non-metric Facility Location πρόβλημα, όπου $m$ είναι ο αριθμός των facilities και $n$ είναι ο αριθμός των clients.

Το αποτέλεσμα αυτό είναι αρκετά σημαντικό καθώς υποδεικνύει μία πολύ ευέλικτη μέθοδο κατά την οποία η λήψη της κλασματικής λύσης του προβλήματος δεν γίνεται διαμέσου της επίλυσης ενός προβλήματος γραμμικού προγραμματισμού. Αυτό μπορεί να αποδειχθεί ιδιαίτερα χρήσιμο καθώς οι κλασματικές και κατ' επέκταση οι τελικές λύσεις μπορούν να παραχθούν πολύ ταχύτερα συγκριτικά με τις τεχνικές LP Rounding. Επίσης αξίζει να σημειωθεί ότι, παρά το γεγονός ότι η κλασματική λύση που παράγεται υστερεί σαφώς από την βέλτιστη που μπορεί να παραχθεί από έναν αλγόριθμο επίλυσης προβλημάτων γραμμικού προγραμματισμού, διατηρούνται ισχυρές εγγυήσεις για την ποιότητα των λύσεων.

## 4.2 Ευριστικές και Άλλες Μέθοδοι

Κάποιες εναλλακτικές προσεγγίσεις για το Facility Location πρόβλημα κάνουν χρήση ευριστικών μεθόδων, νευρωνικών δικτύων και γενετικών αλγόριθμων. Για παράδειγμα, οι Alves και Almeida το 1992 ανάπτυξαν κάποιους αλγόριθμους που βασίζονταν στην τεχνική της προσομοιωμένης ανόπτησης με ιδιαίτερη επιτυχία όσον αφορά την ποιότητα των λύσεων. Ωστόσο, η ταχύτητα εκτέλεσης ήταν σημαντικά αργότερη σε σχέση με άλλες μεθόδους.
Αργότερα, το 1996, αναπτύχθηκε το νευρωνικό δίκτυο TANN (Tabu Neural Network) από τον Vaithyanathan [23] ο οποίος δημιούργησε ένα μαζικώς παράλληλο δίκτυο που του επέτρεπε να λύνει στιγμιότυπα του Facility Location χωρίς χωρητικότητες σχετικά μικρών διαστάσεων (10x10, 20x20 και 30x15). Οι λύσεις που παρήγαγε είχαν σφάλμα μεγαλύτερο του 20% από τις βέλτιστες.
Το 1999 οι Charikar και Guha [6] ανέπτυξαν ένα greedy αλγόριθμο τοπικής αναζήτησης με με συντελεστή προσέγγισης $2.414 + \epsilon$ για το k-median πρόβλημα και το Facility Location χωρίς χωρητικότητες. Ο αλγόριθμος αυτός απαιτεί $\tilde{O}(n^2/\epsilon)$ χρόνο.
Μια άλλη μέθοδος από τους Kratica et al. (2001), εφαρμόζει γενετικούς αλγόριθμους σε Facility Location προβλήματα χωρίς χωρητικότητες με ικανότητα να λύνει μεγέθη εισόδων της τάξης 1000x1000. Όπως έδειξαν, η μέθοδός τους είναι αποδοτικότερη για μεγάλα προβλήματα (>100x1000) από τον αλγόριθμο DUALOC του Erlenkotter.

# Κεφάλαιο 5

# Σχεδίαση και Υλοποίηση

Σε αυτή την ενότητα παρουσιάζονται δύο νέοι αλγόριθμοι για την προσέγγιση της βέλτιστης λύσης στο Facility Location πρόβλημα σε χρονικά μεταβαλλόμενους χώρους. Οι αλγόριθμοι αυτοί αντλούν ιδέες από τις τεχνικές που παρουσιάστηκαν στο Κεφάλαιο 4. Ο στόχος των υλοποιήσεων αυτών συνίσταται στην προσφορά εναλλακτικών τρόπων προσέγγισης του προβλήματος καθώς και στην παραγωγή, σε κάποιες περιπτώσεις, καλύτερων αποτελεσμάτων από τις προϋπάρχουσες μεθόδους. Σε κάθε περίπτωση παρατίθεται μία εκτενής περιγραφή του αλγόριθμου καθώς και μετέπειτα ανάλυση και επεξήγηση των λεπτών σημείων.

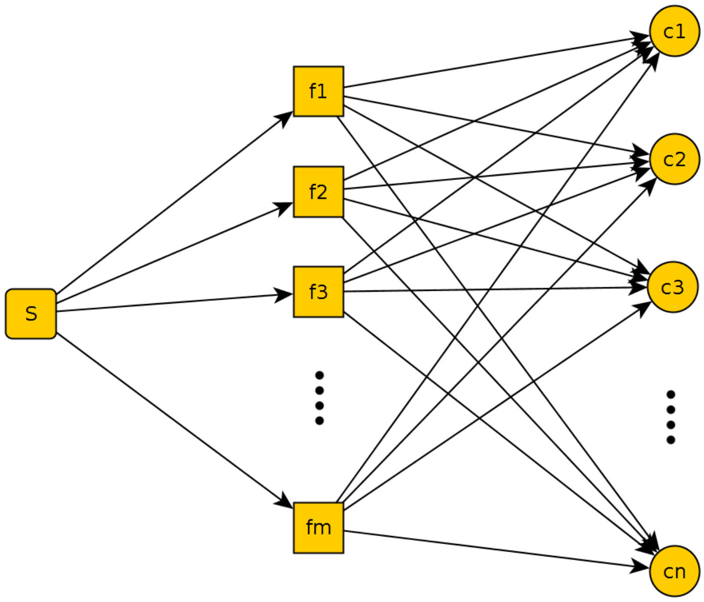
## 5.1 Αλγόριθμος βασιζόμενος σε ροή δικτύου

### 5.1.1 Περιγραφή

Όπως φάνηκε στην εργασία των Alon et al. στο προηγούμενο κεφάλαιο, μπορούμε να αντιμετωπίσουμε τα Non-Metric Facility Location προβλήματα μοντελοποιώντας τα ως προβλήματα ροής δικτύου. Αυτήν την ιδέα μπορούμε να προσαρμόσουμε στις απαιτήσεις του δυναμικού Facility Location προβλήματος, λύνοντας αρχικά ένα πρόβλημα ροής δικτύου για την απόκτηση μιας κλασματικής λύσης, και εφαρμόζοντας έπειτα μία κατάλληλη τεχνική rounding για την λήψη της τελικής λύσης. Αυτή η προσέγγιση αποσκοπεί στην ραγδαία επιτάχυνση της διαδικασίας παραγωγής της κλασματικής λύσης. Ο τρόπος με τον οποίο θα επιτευχθεί αυτό εκθέτεται παρακάτω.
Δεδομένων του ορισμού και των συμβολισμών που χρησιμοποιήθηκαν στο κεφάλαιο 3 για το δυναμικό πρόβλημα Facility Location, ακολουθεί η περιγραφή των δικτύων-γραφημάτων που θα χρησιμοποιήσουμε.

Αρχικά αξίζει να αποσαφηνίσουμε ότι θα κατασκευάσουμε $T$ το πλήθος γραφήματα, έστω $G_t$ με $t \in 1, 2, ..., T$, καθένα εκ των οποίων θα αναφέρεται σε κάθε ένα εκ των διαφορετικών χρονικών στιγμιοτύπων του προβλήματος. Για κάθε γράφημα $G_t$ θεωρούμε μία αρχική κορυφή $s$ και τα σύνολα κόμβων $F = \{f_1, f_2, ..., f_m\}$ και $C = \{c_1, c_2, ..., c_n\}$, όπου κατά σύμβαση θεωρούμε ότι το σύνολο $F$ αναφέρεται στα $m$ facilities και ότι το σύνολο $C$ αναφέρεται στους $n$ clients. Έχουμε δηλαδή $V = \{s\} \cup F \cup C$. Επίσης θεωρούμε το σύνολο κατευθυνόμενων ακμών $E = \{\ e = (x, y) \mid (x = s \wedge y \in F)\ \vee\ (x \in F \wedge y \in C)\ \}$ με το οποίο ολοκληρώνεται η δομική περιγραφή του γραφήματος $G_t = (V, E)$.

Για διευκόλυνση του αναγνώστη, παρατίθεται μία απόδοση του δικτύου που περιγράφηκε στο Σχήμα 5.1.




Σχήμα 5.1. Γράφημα $G_t$.

Πέρα από τη δομική περιγραφή, αποδίδουμε ένα αρχικό βάρος $w_e = 1/2(m + mn)^3$ σε κάθε ακμή καθώς και ένα κόστος $c_e$, για $e = (x, y) \in G_t$ για κάποιο $t$, ως εξής:

$$c_e = \begin{cases} f, & x = s \\ d_1(x,y) + f, & x \neq s, T = 1 \\ d_t(x,y) + f + g \cdot (1 + \max(-1, \frac{w_{xy}^t - w_{xy}^{t+1}}{w_{xy}^t + w_{xy}^{t+1}})), & x \neq s, t = 1,\ T > 1 \\ d_t(x,y) + f + g \cdot (1 + \max(-1, \frac{w_{xy}^t - w_{xy}^{t-1}}{w_{xy}^t + w_{xy}^{t-1}})), & x \neq s, t = T,\ T > 1 \\ d_t(x,y) + f + g \cdot (2 + \max(-1, \frac{w_{xy}^t - w_{xy}^{t+1}}{w_{xy}^t + w_{xy}^{t+1}}) + \max(-1, \frac{w_{xy}^t - w_{xy}^{t-1}}{w_{xy}^t + w_{xy}^{t-1}})), & x \neq s, 1 < t < T \end{cases}$$

Όπως διαφαίνεται, τα κόστη δεν είναι όλα σταθερά καθώς μερικά από αυτά υπολογίζονται συναρτήσει των βαρών κάποιων ακμών του δικτύου που υπόκεινται σε αλλαγές κατά την εκτέλεση του αλγόριθμου. Η επεξήγηση της επιλογής για τα κόστη γίνεται στην επόμενη υποενότητα.

Σε αυτό το δίκτυο θα απαιτήσουμε η ροή $(s, x)$ για κάθε $s \in G_t$ και $x \in C \in G_t$ για κάποιο $t$ να είναι τουλάχιστον ίση με 1. Μάλιστα, για την βελτίωση της κλασματικής αυτής λύσης ο αλγόριθμος θα επεξεργάζεται τα αιτήματα για όλους τους clients σε όλες τις χρονικές στιγμές με Round Robin τρόπο. Αυτό πρακτικά σημαίνει πως κάθε φορά που εξετάζουμε έναν client $x$ για τον οποίο η ροή $(s, x)$ είναι μικρότερη από 1, αυξάνουμε τη ροή στις ακμές της

ελάχιστης τομής με την multiplicative weights update τεχνική που υποδηλώνει η μέθοδος και συνεχίζουμε θεωρώντας τον επόμενο client. Άπαξ και εξετάσουμε όλους τους clients για μία χρονική στιγμή πηγαίνουμε στην επόμενη, και άπαξ και εξετάσουμε όλους τους clients για όλες τις χρονικές στιγμές, επαναλαμβάνουμε τη διαδικασία έως ότου να ικανοποιηθούν όλα τα αιτήματα ροής. Στην πραγματικότητα, κάτι τέτοιο μπορεί να γίνει με ένα τρόπο που να μην χαρακτηρίζεται από την κανονικότητα που περιγράφηκε αλλά μέσω της χρήσης τυχαίων αντιμεταθέσεων των υπολειπόμενων αιτημάτων σε κάθε κύκλο του αλγόριθμου.
Στο τέλος αυτής της διαδικασίας θα έχουμε ικανοποιήσει όλα τα αιτήματα σύνδεσης, αφού σε όλους τους clients για κάθε χρονικό στιγμιότυπο θα καταλήγει τουλάχιστον 1 μονάδα ροής. Το στάδιο που απομένει για την παραγωγή της τελικής λύσης και την ολοκλήρωση του αλγόριθμου είναι η τεχνική ακεραιοποίησης των κλασματικών τιμών που αντιστοιχούν στα βάρη των ακμών του γραφήματος. Για το σκοπό αυτό, κανονικοποιούμε τα βάρη των ακμών που καταλήγουν σε κάθε client σε κάθε στιγμή στην τιμή 1, και αναθέτουμε τις κάτωθι τιμές στις μεταβλητές - δείκτες του προβλήματος:

$$\begin{cases} x_{ij}^t = w_{ij}^t, \quad (\forall ijt) : i \in F \in G_t, j \in C \in G_t \\ y_i^t = \max_j w_{ij}, \quad (\forall it) : i \in F \in G_t, j \in C \in G_t \\ z_{ij}^t = x_{ij}^t - x_{ij}^{t+1}, \quad (\forall ij, t < T) : i \in F \in G_t, j \in C \in G_t \end{cases}$$

Με αυτή τη μέθοδο επιτυγχάνεται η ανεύρεση μιας κλασματικής λύσης η οποία ικανοποιεί τις συνθήκες των LP rounding αλγόριθμων που είδαμε στην Ενότητα 3.1:

$$\begin{cases} (\forall ijt)\ x_{ij}^t \leq y_i^t \\ (\forall jt)\ \sum_{i \in F} x_{ij}^t = 1 \\ (\forall ij, \forall t < T)\ z_{ij}^t \geq x_{ij}^t - x_{ij}^{t+1} \\ y_i^t, x_{ij}^t, z_{ij}^t \geq 0 \end{cases}$$

Συνεπώς, μπορεί να εφαρμοσθεί ένας εκ των δύο αυτών αλγόριθμων για την απόκτηση της τελικής λύσης. Μια σημαντική παρατήρηση εδώ είναι ότι το φράγμα $6 \cdot OPT$ που αποδείξαμε για το κόστος σύνδεσης στον αλγόριθμο των Eisentat et al. δεν ισχύει εδώ καθώς η λύση του LP δεν είναι βέλτιστη και κατ' επέκταση δεν ισχύουν οι συνθήκες συμπληρωματικής χαλαρότητας. Ωστόσο, ισχύει η εγγύηση για λογαριθμικό αναμενόμενο κόστος σύνδεσης που έχουν υπολογίσει οι ίδιοι στην εργασία τους.
Ο αλγόριθμος γενικεύεται πολύ εύκολα για διαφορετικές τιμές κόστους λειτουργίας για κάθε facility $i$ σε κάθε χρόνο $t$, με τις μεταβλητές εισόδου $f_i^t$.

### 5.1.2 Ποιοτική Ανάλυση

Αρχικά θα αιτιολογήσουμε τα ορισθέντα κόστη για τις ακμές. Οι ακμές που συνδέουν την εκάστοτε αρχική κορυφή $s$ με κάποιο facility έχουν το αναμενόμενο κόστος $f$ που αντιστοιχεί στο κόστος λειτουργίας του facility. Ο πρώτος όρος στις ακμές που συνδέουν κάποιο facility με κάποιον client είναι, επίσης αναμενόμενα, το κόστος σύνδεσης των δύο οντοτήτων που ορίζει η συνάρτηση $d_t$.

Ωστόσο, σε αυτές τις ακμές προσθέτουμε επίσης το κόστος $f$ για τον κάτωθι λόγο:
Δεν επιθυμούμε το βάρος της ακμής που αντιπροσωπεύει την σύνδεση ενός facility και ενός client να αυξηθεί αυθαίρετα σε σχέση με το κόστος λειτουργίας του facility. Κατά τη διάρκεια εκτέλεσης του αλγόριθμου το βάρος αυτής της ακμής πιθανώς να υπερβεί το βάρος της ακμής που αντιπροσωπεύει το ποσοστό κατά το οποίο θεωρούμε ανοιχτό το αντίστοιχο facility. Κατά συνέπεια, αν δεν περιλαμβάναμε το κόστος $f$, στο τελικό στάδιο κατά το οποίο θέτουμε το ποσοστό κατά το οποίο είναι ανοιχτό ένα facility ίσο με το μέγιστο εκ των βαρών των ακμών που το συνδέουν με τους clients, θα είχαμε αυθαίρετα μεγάλη αύξηση κόστους. Αντιθέτως, με αυτήν την προσθήκη, η αύξηση κόστους που προκύπτει για κάθε facility στην τελική ανάθεση είναι της τάξης $1/f$.
Τέλος, η σημαντικότερη ίσως εισαγωγή στο κόστος των ακμών που συνδέουν facilities και clients είναι ο τρίτος όρος (όπου εφαρμόζεται) και αφορά το κόστος αλλαγής ανάθεσης $g$. Αυτός ο όρος αποσκοπεί στην μοντελοποίηση και τον περιορισμό του κόστους που προκύπτει από την αλλαγή της ανάθεσης ενός client από κάποια χρονική στιγμή στην επόμενη ή προηγούμενη (όπου αυτές υπάρχουν). Παρατηρείται ότι αυτός ο όρος επιβραδύνει ή επιταχύνει τις αυξήσεις στα βάρη των ακμών όπου το κόστος αλλαγής ανάθεσης που θα προκύψει είναι μεγαλύτερο ή μικρότερο αντίστοιχα. Αυτή η επιβράδυνση ή επιτάχυνση είναι ανάλογη της διαφοράς των βαρών των ακμών που συνδέουν τον ίδιο client με το ίδιο facility αλλά για διαφορετικές χρονικές στιγμές. Επιπροσθέτως, οι διαφορές αυτές κανονικοποιούνται διαρούμενες με το άθροισμα των αντίστοιχων βαρών ώστε να υπάρχει εξ αρχής μία προβολή του τελικού κόστους που θα προκύψει από κάθε αύξηση. Αν δεν γινόταν αυτό, οι αρχικές επαναλήψεις θα αγνοούσαν σε μεγάλο βαθμό τα κόστη αλλαγής ανάθεσης αφού τα αντίστοιχα βάρη των εμπλεκόμενων ακμών θα ήταν πολύ μικρά.

Σε αυτήν την παράγραφο παραθέτονται κάποιες θεωρητικώς αναμενόμενες αδυναμίες.
1. Σε περίπτωση που κάποιος τύπος κόστους (πχ. σύνδεσης, ανάθεσης) είναι σημαντικά μεγαλύτερος από κάποιον άλλον, η διακριτική ικανότητα της διαδικασίας μειώνεται όσον αφορά τον τύπο κόστους με μικρότερες τιμές. Φερ΄ ειπείν, ας υποθέσουμε ένα στιγμιότυπο του προβλήματος όπου η βελτιστη λύση δεν απαιτεί καμία αλλαγή ανάθεσης. Αυξάνοντας το κόστος ανάθεσης προς το άπειρο τα κόστη σύνδεσης παύουν να επιδρούν στον ρυθμό αύξησης των βαρών των ακμών και έτσι επιλέγεται μία αυθαίρετη και χρονικώς σταθερή σύνδεση για κάθε client. Βέβαια, σε κάποιες τέτοιες περιπτώσεις μπορούν να χρησιμοποιηθούν οι αναγωγές που αποδείχθηκαν στο Κεφάλαιο 3.

Η ταχύτητα εκτέλεσης του αλγόριθμου, λόγω της φύσης της multiplicative weights update τεχνικής, επηρεάζεται από τα διάφορα κόστη που ορίζονται στο πρόβλημα. Για παράδειγμα, αν θεωρήσουμε δύο προβλήματα, με μόνη διαφορά μεταξύ τους ότι τα κόστη του δεύτερου είναι ίσα με τα κόστη του πρώτου πολλαπλασιασμένα κατά ένα παράγοντα $c > 1$ ο αλγόριθμος θα καθυστερήσει περισσότερο στο δεύτερο πρόβλημα. Παρ΄ όλα αυτά, η βέλτιστη λύση και των δύο είναι η ίδια. Το πρόβλημα αυτό μπορεί να αποφευχθεί αν εκμεταλλευτούμε την ελαστικότητα της μεθόδου και θέσουμε ένα σταθερό παράγοντα ρυθμού “μάθησης” (learning ratio)

$\eta$, παραλλάσοντας τη μέθοδο ως εξής:

$$w_e \leftarrow w_e(1 + \eta/c_e)$$

Ορίζοντας αυτόν τον ρυθμό κατάλληλα ως τον σταθμισμένο μέσο όρο των κοστών που συμμετέχουν στο πρόβλημα διασφαλίζεται η ταχεία εκτέλεση του αλγόριθμου και διατηρείται το επίπεδο της ποιότητας των λύσεων.

### 5.1.3 Παραλλαγή για την Online εκδοχή του προβλήματος

Η μεθοδολογία που χρησιμοποιήθηκε παραπάνω, για την Offline έκδοση, αρχικώς αναπτύχθηκε για την αντιμετώπιση Online προβλημάτων. Εμείς καταφέραμε να την εφαρμόσουμε σε ένα Offline πρόβλημα χρησιμοποιώντας μία Round Robin αντιμετώπιση των αιτημάτων. Αυτό είχε ως συνέπεια την εκμετάλλευση όλης της πληροφορίας του προβλήματος.
Ωστόσο, αξίζει να δούμε κατά ποιον τρόπο μπορούμε να παραλλάξουμε τον αλγόριθμο ώστε να παρέχει λύσεις για εκδοχές του προβλήματος όπου τα αιτήματα καταφθάνουν με έναν δυναμικό τρόπο. Η επόμενη παράγραφος αφιερώνεται στην περιγραφή του αλγόριθμου για την Online εκδοχή του προβλήματος.

#### Περιγραφή

Σε αυτή την εκδοχή τα αιτήματα σύνδεσης των clients έρχονται δυναμικά και κάθε απόφαση σύνδεσης είναι μη αναστρέψιμη. Ο αλγόριθμος που θα περιγραφεί είναι ιδιαίτερα ευέλικτος και μπορεί να εφαρμοσθεί σε περιπτώσεις στις οποίες ο τρόπος με τον οποίο καταφθάνουν τα αιτήματα διαφέρει σημαντικά. Συγκεκριμένα μπορεί να διαχειριστεί αιτήματα που αναφέρονται σε σύνολα clients διαφορετικών μεγεθών. Παραδείγματος χάριν, μπορεί σε ένα σενάριο για κάθε χρονική στιγμή όλοι οι clients να καταφθάνουν μαζί ή μπορεί να έρχονται σειριακά (ο ένας μετά τον άλλο). Επίσης, θα μπορούσε το μέγεθος του συνόλου των αιτημάτων να διαφέρει κατά τη διάρκεια του προβλήματος. Σε κάθε περίπτωση πράττουμε ως εξής:
Έστω ένα νέο αίτημα $T$ που αναφέρεται σε ένα σύνολο από clients. Ανανεώνουμε το δίκτυο έως ότου ικανοποιηθεί το νέο αίτημα με τον τρόπο που περιγράφηκε στην Offline εκδοχή. Υπολογίζουμε μία εφικτή λύση βάσει των βαρών του δικτύου, τα οποία και μετατρέπουμε έπειτα στις τιμές 1 ή $1/2|E|^3$ αναλόγως με το αν χρησιμοποιούνται ή όχι στη λύση. Συνεχίζουμε με το επόμενο αίτημα διατηρώντας τα βάρη στις ακμές που έχουν καθοριστεί έως το τρέχον σημείο εκτέλεσης.

## 5.2 Προσομοιωμένη Ανόπτηση

Η προσομοιωμένη ανόπτηση είναι μία ευρέως χρησιμοποιούμενη πιθανοκρατική μέθοδος για την προσέγγιση του ολικού ακρότατου μίας συνάρτησης.
Η έννοια της ανόπτησης (annealing) προέρχεται από τη φυσική διαδικασία κατά τη οποία ένα υλικό θερμαίνεται αρχικά σε υψηλή θερμοκρασία και στη συνέχεια ψύχεται σταδιακά, επιτρέποντας έτσι την στερεοποίησή του σε μια κρυσταλλική κατάσταση χαμηλής ενέργειας.

Στην ενότητα αυτή παρουσιάζεται συνοπτικά η αλγοριθμική μέθοδος καθώς ο τρόπος με τον οποίο η προσομοιωμένη ανόπτηση μπορεί να εφαρμοστεί συγκεκριμένα στο δυναμικό Facility Location πρόβλημα.

### 5.2.1 Περιγραφή

Η μέθοδος της προσομοιωμένης ανόπτησης είναι μια απάντηση στο δίλημμα μεταξύ της εξερεύνησης και της εκμετάλλευσης σε προβλήματα αναζήτησης. Ο αλγόριθμος έχει ως εξής:

Σε κάθε επανάληψη επιλέγεται μια τυχαία κίνηση η οποία, αν οδηγεί σε μία κατάσταση καλύτερη της τρέχουσας γίνεται αποδεκτή, διαφορετικά γίνεται αποδεκτή με κάποια πιθανότητα η οποία μειώνεται εκθετικά συναρτήσει της ακαταλληλότητάς της και μιας παραμέτρου θερμοκρασίας $T$. Συγκεκριμένα, η πιθανότητα για να γίνει η κίνηση σε αυτήν την περίπτωση είναι $e^{\Delta E/T}$, όπου $\Delta E$ είναι η διαφορά κόστους της υποψήφιας κατάστασης από την τρέχουσα. Η παράμετρος $T$ αρχικοποιείται σε μία σχετικώς υψηλή τιμή και μειώνεται σταδιακά. Όσο μικρότερος είναι ο ρυθμός μείωσής της τόσο καλύτερα προσεγγίζεται το ολικό ακρότατο της συνάρτησης. Για μεγάλες τιμές της θερμοκρασίας $T$, η μέθοδος συμπεριφέρεται σαν τυχαίος περίπατος ενώ για χαμηλές τιμές επιλέγει μόνο βελτιωτικές κινήσεις.

Η προσομοιωμένη ανόπτηση ενδείκνυται για την αντιμετώπιση διάφορων δύσκολων προβλημάτων και το δυναμικό πρόβλημα Facility Location δεν αποτελεί εξαίρεση. Στη συνέχεια παρατίθεται ο ψευδοκώδικας του αλγόριθμου που υλοποιήθηκε για τις ανάγκες του προβλήματος. Για αποφυγή πιθανής σύγχυσης, διευκρινίζεται ότι στον παρακάτω κώδικα η μεταβλητή $T$ αναφέρεται στα χρονικά στιγμιότυπα του προβλήματος και όχι στην θερμοκρασία.

**Προσομοιωμένη Ανόπτηση στο Δυναμικό πρόβλημα Facility Location**

*Αρχικοποίησε την μεταβλητή θερμοκρασίας temp* $\leftarrow 1000$.

*Όρισε μία έγκυρη αρχική κατάσταση αναθέτοντας κάθε client για κάθε χρόνο σε κάποιο τυχαίο facility.*

*Υπολόγισε το κόστος της λύσης αυτής και αποθήκευσέ τη στη μεταβλητή cost.*

***Όσο*** $(temp > 0)$ ***επανάλαβε***

*Επίλεξε μία τυχαία τριάδα* $(t, i, j)$, $t \in \{1, 2, ..., T\}$, $i \in \{1, 2, ..., m\}$ *και* $j \in \{1, 2, ..., n\}$.

*Υπολόγισε το κόστος της λύσης που θα προέκυπτε αν στον χρόνο t ο client j συνδεόταν στο facility i και αποθήκευσέ τη στη μεταβλητή tentative_cost.*

*Θέσε* $\Delta E \leftarrow cost - tentative\_cost$.

***Αν*** $(\Delta E > 0$ *ή* $e^{\Delta E/temp} > rand(0, 1))$ ***τότε***

- *Ανάθεσε στον χρόνο t τον client j στο facility i.*
- $cost \leftarrow tentative\_cost$

***Τέλος αν***

$temp \leftarrow temp - 1/(Tmn)$

***Τέλος επανάληψης***

### 5.2.2 Παρατηρήσεις

Όπως είναι εμφανές από τον κώδικα που προηγήθηκε, τα βήματα της επαναληπτικής διαδικασίας που ορίσαμε μπορούν να υπολογιστούν εύκολα. Συγκεκριμένα, το πλήθος τους είναι $1000Tmn$, γεγονός που πρακτικά σημαίνει ότι στη διάρκεια εκτέλεσης του αλγόριθμου θα εξεταστεί κάθε δυνατή ανάθεση κατά μέσο όρο 1000 φορές.
Η αρχική πιθανότητα για μία κίνηση που αυξάνει το κόστος κατά 1000 μονάδες είναι $e^{-1} = 0,367879$ ενώ στην τελευταία επανάληψη η πιθανότητα να επιλεχθεί μία κίνηση που αυξάνει το κόστος κατά μία μονάδα είναι $e^{-Tmn} \leq e^{-1}$ σε κάθε περίπτωση.

Σαφώς, μία σημαντική αδυναμία του αλγόριθμου είναι η έλλειψη της ικανότητας αποφυγής βαθειών τοπικών ελαχίστων. Μάλιστα, ο αλγόριθμος μπορεί να υποπέσει εύκολα σε τέτοια κατάσταση αν το κόστος αλλαγής ανάθεσης $g$ ή το κόστος λειτουργίας των facilities $f$ είναι μεγάλο σε σύγκριση με τα υπόλοιπα. Μια ιδέα για την αντιμετώπιση τέτοιων καταστάσεων είναι η σχετική μείωση όλων των κοστών που συμμετέχουν στο πρόβλημα ώστε το μέγιστο εξ αυτών να μην ξεπερνά την τιμή 1000. Επίσης, σε κάποιες από αυτές τις περιπτώσεις μπορούν να εφαρμοσθούν οι αναγωγές που αποδείχθηκαν στο Κεφάλαιο 3.

# Κεφάλαιο 6

# Πειραματική Αξιολόγηση

Στα προηγούμενα κεφάλαια γνωρίσαμε τέσσερις αλγόριθμους για τους οποίους θα χρησιμοποιήσουμε τις παρακάτω συντομογραφίες:

1. EMS (ο αλγόριθμος των Eisenstat, Mathieu και Schabanel, εν. 2.2.2)
2. EXPCLOCKS (ο αλγόριθμος των An, Norouzi-Fard και Svensson, εν. 2.2.3)
3. NETFLOW (ο αλγόριθμος που βασίζεται σε ανάλυση ροής δικτύου, εν. 4.1)
4. SIMAN (ο αλγόριθμος που βασίζεται στην προσομοιωμένη ανόπτηση, εν. 4.2)

Σκοπός αυτού του κεφαλαίου είναι η αξιολόγηση και σύγκριση των αλγόριθμων αυτών ως προς διάφορα χαρακτηριστικά, όπως είναι η ποιότητα των λύσεων που αποδίδουν και η ταχύτητα εκτέλεσής τους.
Για αυτό το σκοπό έγινε προσπάθεια παραγωγής μεγάλου εύρους και ποικιλίας από προβλήματα για την δοκιμή των αλγόριθμων και τον εντοπισμό των ασθενών σημείων τους.

## 6.1 Τεχνικές λεπτομέρειες

Στον αλγόριθμο NETFLOW, έπειτα από την ανεύρεση της κλασματικής λύσης, χρησιμοποιήσαμε την τεχνική Rounding του αλγόριθμου EXPCLOCKS. Για την επίλυση των προβλημάτων γραμμικού προγραμματισμού χρησιμοποιήθηκε ο αλγόριθμος simplex από το πακέτο GNU Linear Programming Kit (GLPK 4.60, April 1, 2016) [13].

## 6.2 Κριτήρια αξιολόγησης

### 6.2.1 Ποιότητα λύσεων

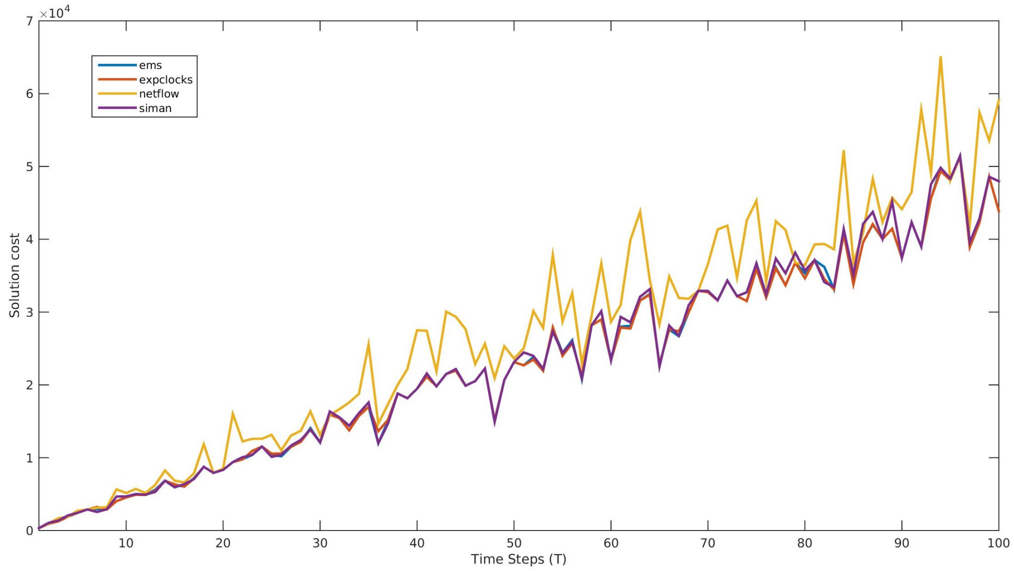
Ως πρώτο κριτήριο αξιολόγησης, οι αλγόριθμοι εξετάζονται ως προς το κόστος των λύσεων που παράγουν σε πληθώρα από τυχαίες και μη εισόδους. Οι είσοδοι αυτές χαρακτηρίζονται από κάποιες παραμέτρους. Αρχικά καθορίζεται το μέγεθος της εισόδου που εξαρτάται από τρεις μεταβλητές, το πλήθος των χρονικών στιγμιοτύπων $T$, τον αριθμό των facilities $m$ και το πλήθος των clients $n$. Έπειτα επιλέγονται για κάθε είσοδο οι τιμές του κόστους λειτουργίας του κάθε facility $f$ και του κόστος αλλαγής σύνδεσης $g$. Τέλος, καθορίζονται οι τιμές των αποστάσεων μεταξύ των facilities και των clients, δηλαδή η συνάρτηση $d$.

#### Metric

Αρχικά θα εξετάσουμε στιγμιότυπα των οποίων οι αποστάσεις ικανοποιούν την τριγωνική ανισότητα. Υπενθυμίζουμε ότι ο αλγόριθμος EXPCLOCKS δεν είναι σχεδιασμένος για την μη μετρική εκδοχή του προβλήματος.

#### Μεταβολές στο μέγεθος εισόδου

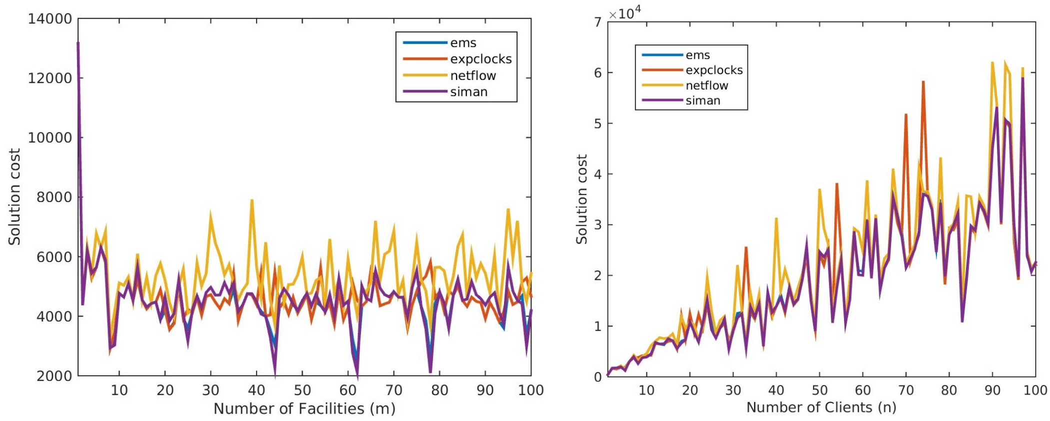
$(T = 1\ to\ 100, m = 10, n = 10)\ \ f \sim U(1, 100),\ g \sim U(1, 100),\ d \sim U(1, 100)$




Στο παραπάνω διάγραμμα αποτυπώνονται τα αποτελέσματα της εκτέλεσης 100 εισόδων με αριθμό χρονικό βημάτων από 1 έως 100.
Όπως φαίνεται, η σχέση του παραγόμενου κόστους των λύσεων και των χρονικών βημάτων ενός προβλήματος με κόστη που λαμβάνονται ομοιόφορφα από ένα διάστημα είναι γραμμική. Κανείς από τους αλγόριθμους δεν φαίνεται να αποκλίνει από αυτήν την σχετικώς ευσταθή γραμμική πορεία, γεγονός επιθυμητό.

$$(T = 10) \quad f \sim U(1, 100), \ g \sim U(1, 100), \ d \sim U(1, 100)$$

$(m = 1 \ to \ 100, n = 10)$ $(m = 10, n = 1 \ to \ 100)$

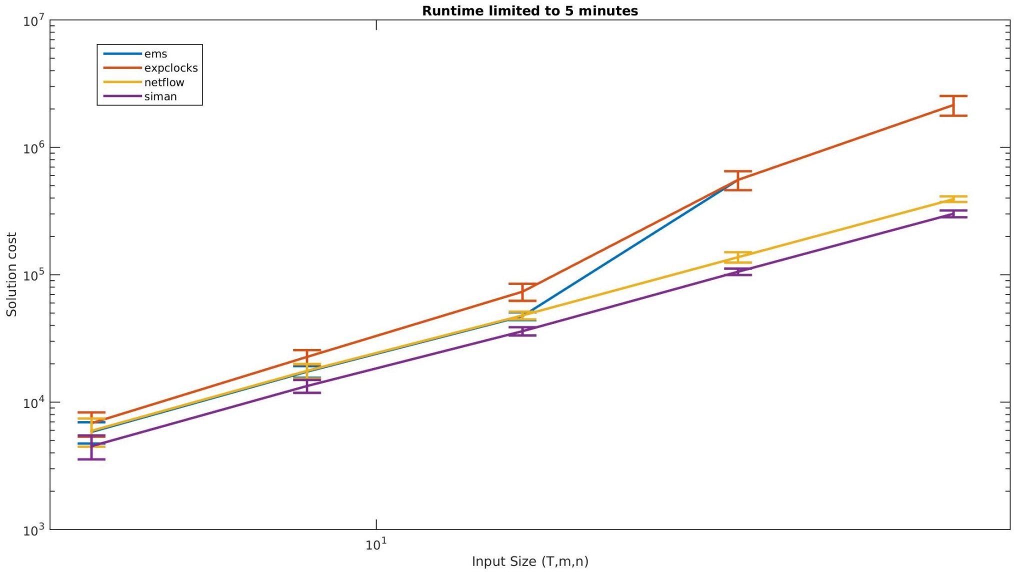



Τα παραπάνω δύο διαγράμματα παρουσιάζουν τον τρόπο με τον οποίο μεταβάλλεται το κόστος της λύσης σε σχέση με τον αριθμό των facilities (αριστερά) και σε σχέση με τον αριθμό των clients (δεξιά).

Καθώς τα facilities αυξάνονται, ενώ είναι ακόμα λίγα, παρατηρείται πτώση του συνολικού κόστους της λύσης. Το φαινόμενο αυτό είναι λογικό αν αναλογιστούμε ότι κάθε επιπλέον facility προσφέρει νέες επιλογές για την σύνδεση των clients. Γρήγορα όμως αυτή η αύξηση παύει να βοηθάει ουσιαστικά καθώς στατιστικώς πλέον κάθε νέο facility είναι λιγότερο πιθανό να είναι εντός των λίγων καλύτερων που θα χρησιμοποιηθούν.

Η σχέση του κόστους της λύσης και του αριθμού των clients μοιάζει να είναι επίσης γραμμική, αν και λιγότερο ευσταθής σε αντιδιαστολή με αυτήν που προκύπτει ως προς τον αριθμό των χρονικών βημάτων.

$$(T = m = n = 2^2 \ to \ 2^6) \quad f \sim U(1, 10), \ g \sim U(1, 10), \ d \sim U(1, 1000)$$

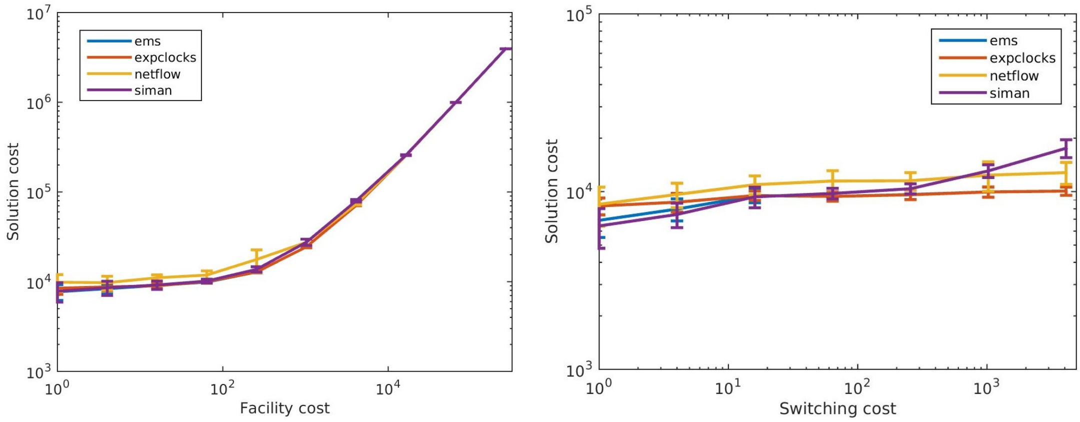



Το παραπάνω σχήμα παρουσιάζει ένα αναμενόμενα καλό αποτέλεσμα για τους αλγόριθμους NETFLOW και SIMAN. Περιορίζοντας τον χρόνο εκτέλεσης για όλους τους αλγόριθμους στα 5 λεπτά, διαπιστώνουμε ότι, ενώ για μικρές εισόδους έχουμε παραπλήσια συμπεριφορά, για μεγάλες εισόδους ($\geq$ 32x32x32) η επίδοση των αλγόριθμων EXPCLOCKS και EMS χειροτερεύουν σημαντικά καθώς ο αλγόριθμος επίλυσης προβλημάτων γραμμικού προγραμματισμού δεν έχει καταφέρει να συγκλίνει σε μία καλή λύση εντός του χρονικού περιορισμού.

### Μεταβολές στις παραμέτρους εισόδου

$(T = 15, m = 15, n = 15) \;\; d \sim U(1, 100)$

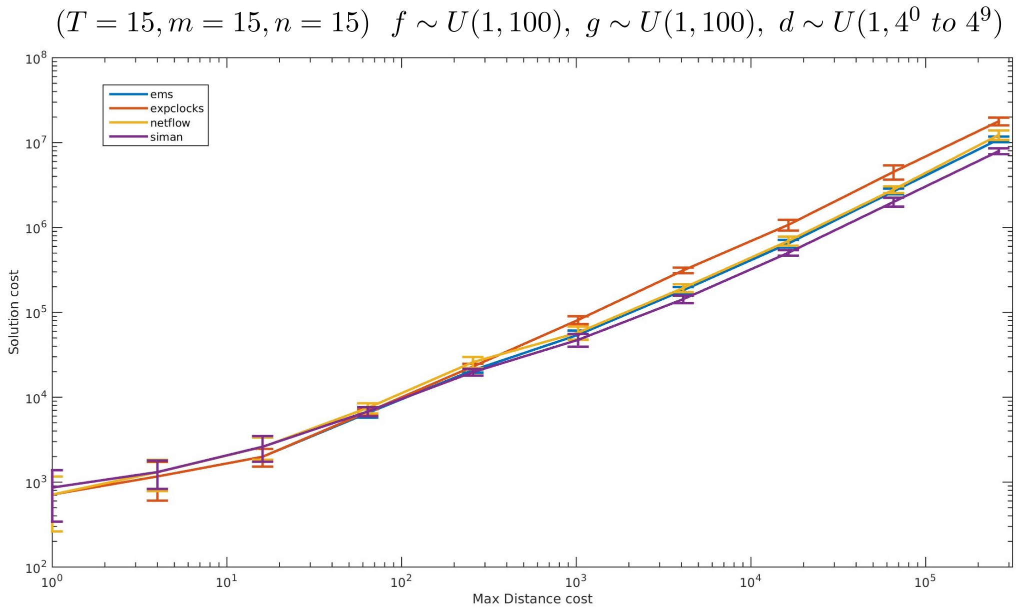
$(f = 4^0 \; to \; 4^9, \;\; g \sim U(1, 100))$ $\qquad$ $(f \sim U(1, 100), \;\; g = 4^0 \; to \; 4^6)$




Για μεγάλο κόστος των facilities παρατηρούμε την χειρότερη επίδοση στον αλγόριθμο SIMAN, ένα θεωρητικώς αναμενόμενο αποτέλεσμα αφού η αύξηση αυτού του κόστους οδηγεί στην ύπαρξη πολλών βαθειών τοπικών ελάχιστων στη συνάρτηση προς ελαχιστοποίηση. Δηλαδή, ο αλγόριθμος χάνει την ικανότητα εξερεύνησης καθώς κάθε αλλαγή που ανοίγει κάποιο νέο facility είναι σχεδόν απαγορευτική.

Αυξάνοντας το κόστος αλλαγής ανάθεσης παρατηρούμε τα εξής:

Οι λύσεις των αλγόριθμων EMS και EXPCLOCKS τείνουν να συμπίπτουν καθώς η επίλυση του αντίστοιχου προβλήματος γραμμικού προγραμματισμού παράγει συχνότερα ακέραιες τιμές 0 ή 1.



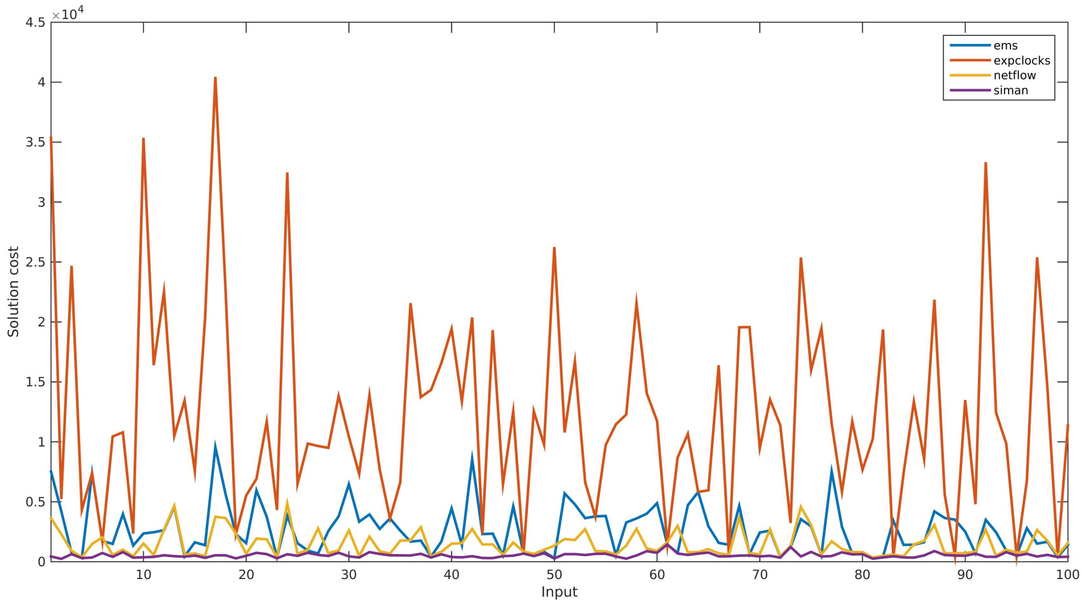

Για μεγαλύτερο κόστος αποστάσεων παρατηρούμε ότι η σχετική επίδοση του αλγόριθμου EXPCLOCKS χειροτερεύει. Σε άλλα πειράματα επιχειρήθηκε η περαιτέρω αύξηση του κόστους των αποστάσεων, χωρίς επίπτωση στην σχετική επίδοση του αλγόριθμου. Το γεγονός που κεντρίζει το ενδιαφέρον εδώ είναι η καλύτερη επίδοση της τεχνικής Rounding του αλγόριθμου EXPCLOCKS όταν αυτή εφαρμοσθεί στην κλασματική λύση που παράγει ο NETFLOW. Μία άλλη σημαντική παρατήρηση είναι η καλή επίδοση του αλγόριθμου SIMAN, κάτι που αναμένεται για συγκριτικά μικρά κόστη λειτουργίας των facilities και μικρά κόστη αλλαγής ανάθεσης.

Γενικά, παρατηρούμε ότι οι αλγόριθμοι EMS, EXPCLOCKS, NETFLOW έχουν παραπλήσια συμπεριφορά, με τον αλγόριθμο NETFLOW να έχει συνήθως την χειρότερη επίδοση, κάτι που όπως ειπώθηκε είναι αναμενόμενο αφού οι αλγόριθμοι EMS και EXPCLOCKS τροφοδοτούνται με την βέλτιστη κλασματική λύση. Ο αλγόριθμος SIMAN έχει συχνά την καλύτερη επίδοση σε περιπτώσεις όπου τα κόστη λειτουργίας των facilities και των αλλαγών ανάθεσης δεν είναι πολύ μεγάλα.

### Non-Metric

Σε αυτήν την ενότητα εξετάζουμε στιγμιότυπα των οποίων οι αποστάσεις δεν ικανοποιούν αναγκαστικά την τριγωνική ανισότητα. Οι αλγόριθμοι που προβλέπονται σε αυτά τα σενάρια είναι όλοι πλην του EXPCLOCKS. Λόγω αυτού, στον αλγόριθμο NETFLOW χρησιμοποιήθηκε η μέθοδος Rounding του αλγόριθμου EMS.

$$(T = 11, m = 11, n = 11) \quad f \sim U(1, 10), \; g \sim U(1, 10), \; d = 1 \; or \; 1000$$

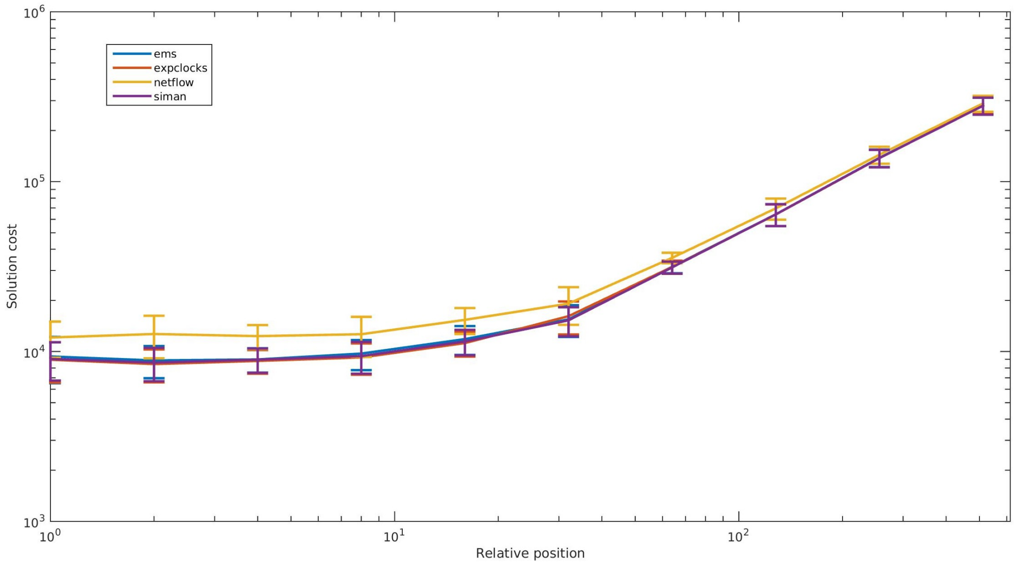



Σε αυτήν την είσοδο έχουμε μικρές τιμές $f$ και $g$ και αποστάσεις που έχουν τιμή 1 ή 1000. Άρα ο αλγόριθμος οφείλει να αποφύγει όσες μπορεί από τις αναθέσεις με κόστος 1000. Δεδομένων αυτών, τα αποτελέσματα είναι αναμενόμενα. Οι αποστάσεις, λόγω των ακραίων τιμών τους, παραβιάζουν σημαντικά την τριγωνική ανισότητα και επομένως δεν ισχύουν οι στατιστικές εγγυήσεις του αλγόριθμου EXPCLOCKS, γεγονός που δικαιολογεί την πολύ κακή συμπεριφορά του. Από την άλλη, ο αλγόριθμος SIMAN παράγει πολύ καλές λύσεις καθώς οι μικρές τιμές $f$ και $g$ τον καθιστούν ιδιαίτερα ευέλικτο κατά τη διαδικασία της αναζήτησης.

### Μοντελοποίηση της κίνησης στο χώρο

Για την μοντελοποίηση της κίνησης στο χώρο, δημιουργήσαμε ένα σύνολο από εισόδους στις οποίες για την πρώτη χρονική στιγμή ($t = 1$) τοποθετήσαμε τους clients και τα facilities σε ένα πλέγμα 100x100. Έπειτα, για να λάβουμε τις θέσεις των clients για κάθε επόμενο χρονικό βήμα, επιτρέπουμε σε κάθε ςλιεντ να κινηθεί τυχαία σε μία ακτίνα γύρω από την προηγούμενή του θέση. Δηλαδή, επιτρέπουμε μία μέγιστη σχετική μετατόπιση για κάθε client μεταξύ κάθε δύο συνεχόμενων χρονικών στιγμών, ώστε να εισάγουμε μία έννοια χωροχρονικής συνέχειας.

$$(T = 17, m = 17, n = 17) \;\; f \sim U(1, 100), \;\; g \sim U(1, 100)$$

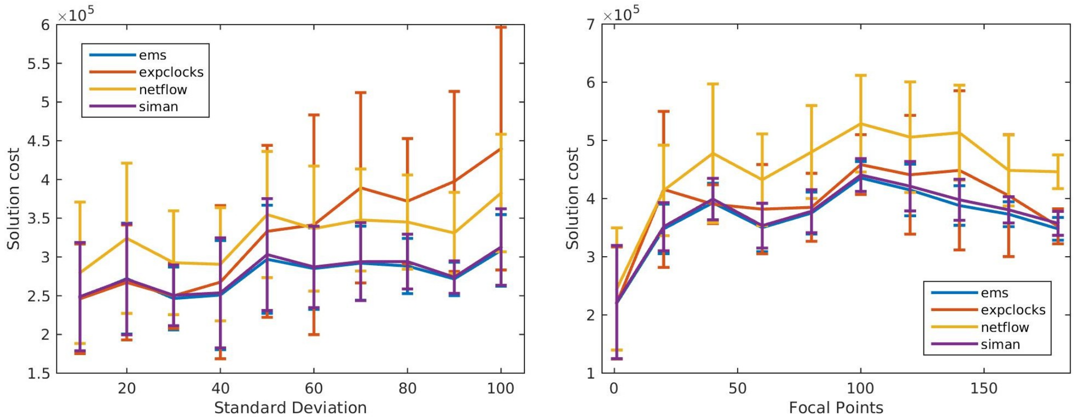



Οι τιμές της επιτρεπτής σχετικής μετατόπισης (Relative Position) λαμβάνουν τιμές από 0 έως 511 σε γεωμετρική πρόοδο. Παρατηρούμε ότι πέραν του αλγόριθμου NETFLOW, οι υπόλοιποι 3 παρουσιάζουν παρόμοια συμεριφορά. Όσον αφορά τον αλγόριθμο NETFLOW, αυτός παρουσιάζει σταθερά ελαφρώς χειρότερη επίδοση, γεγονός που δικαιολογείται από τη μη βέλτιστη κλασματική λύση με την οποία τροφοδοτεί τη διαδικασία Rounding. Για μεγαλύτερες τιμές της σχετικής μετατόπισης ο αλγόριθμος έχει επίδοση που συγκλίνει σε αυτή των υπόλοιπων καθώς πλέον οι clients συχνά ξεφεύγουν εκτός του πλέγματος στο οποίο είχαν αρχικά τοποθετηθεί.

### Προβλήματα όπου οι clients χωρίζονται σε clusters κανονικής κατανομής

Σε αυτά τα προβλήματα επιλέγουμε μερικά υποψήφια facilities σε ένα πλέγμα 1000x1000, καθώς επίσης και κάποια σημεία γύρω από τα οποία τοποθετούμε ένα συγκεκριμένο υποσύνολο των clients. Οι clients κινούνται τυχαία με κανονική κατανομή γύρω από το σημείο που αντιστοιχεί στον καθένα. Έτσι, δημιουργούμε κάποια clusters και οι αλγόριθμοί μας οφείλουν να εντοπίσουν την διαφορετική συμπεριφορά των διάφορων ομάδων από clients παρά τις πιθανές επικαλύψεις των περιοχών εντός των οποίων κινούνται.

$$(T = 10, m = 10, n = 180)$$



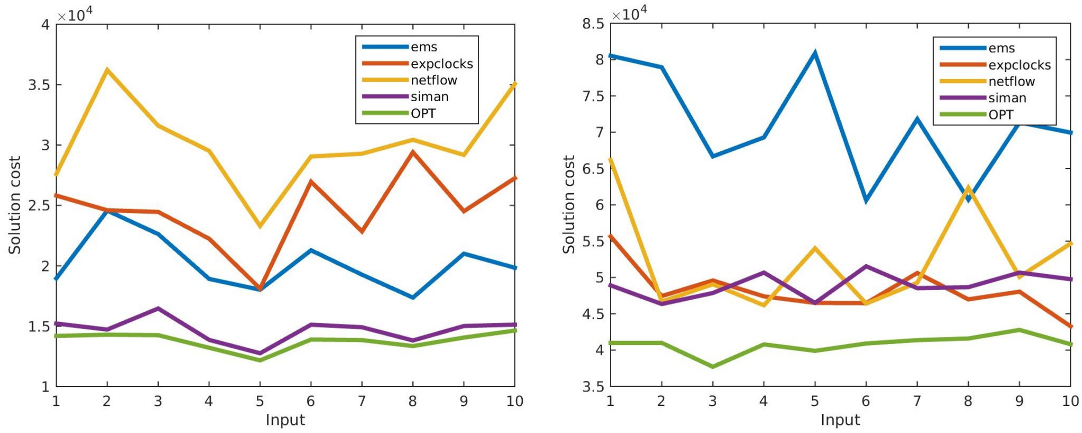

Στο αριστερό διάγραμμα μεταβάλλουμε την τυπική απόκλιση της κατανομής που ακολουθούν οι clients, διατηρώντας ωστόσο σταθερό των αριθμό των σημείων γύρω από τα οποία κινούνται. Απεναντίας, στο διάγραμμα δεξιά βλέπουμε πως επιδρά η μεταβολή στο πλήθος των σημείων γύρω από τα οποία κινούνται οι clients.

### Προβλήματα από τη βιβλιοθήκη UfLib

Μία βιβλιοθήκη με διάφορα testcases για το Facility Location χωρίς χωρητικότητες προέρχεται από το Max-Planck-Institut für Informatik [9]. Η βιβλιοθήκη περιέχει στιγμιότυπα διαφόρων μεγεθών και τύπων, όπως πχ Metric και Non-Metric. Οι αλγόριθμοι που αναπτύξαμε μπορούν να εφαρμοσθούν και εδώ, δεδομένου ότι το Uncapacitated Facility Location πρόβλημα είναι ισοδύναμο με το Facility Location πρόβλημα σε Χρονικά Μεταβαλλόμενους Χώρους για μία χρονική στιγμή ($T = 1$). Επιπλέον, κάτι σημαντικό που παρέχει η βιβλιοθήκη είναι η βέλτιστη λύση σε μεγάλο ποσοστό από τα στιγμιότυπά της.

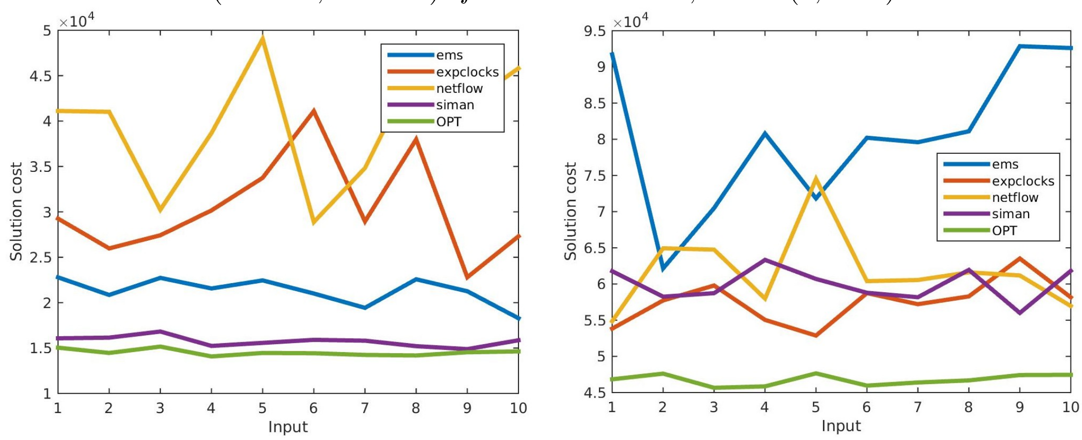
Πακέτο: BildeKrarup - Dq1 & Dq10

$(m = 30, n = 80)$ $f = 1000$ & $10000$, $d \sim U(0, 1000)$




Πακέτο: BildeKrarup - Eq1 & Eq10

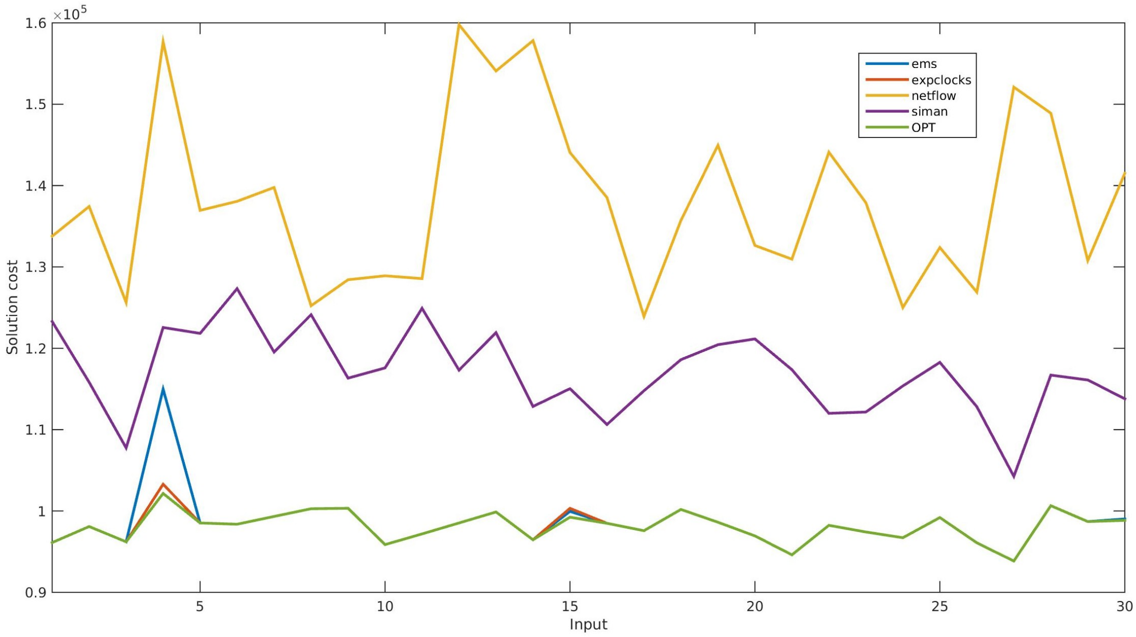
$(m = 50, n = 100)$ $f = 1000$ & $10000$, $d \sim U(0, 1000)$




Από τα παραπάνω παρατηρούμε ότι για μικρότερες τιμές του κόστους των φασιλιτιες ο αλγόριθμος SIMAN παρουσιάζει μία επίδοση πολύ κοντά στη βέλτιστη, με δεύτερο καλύτερο τον αλγόριθμο EMS. Αυξάνοντας το κόστος των facilities, οι αλγόριθμοι EXPCLOCKS, NETFLOW και SIMAN αποκτούν παραπλήσια συμπεριφορά, με πτώση της επίδοσης του SIMAN. Η πτώση αυτή δικαιολογείται από τη μείωση της εξερευνητικής ικανότητας του SIMAN, καθώς κινήσεις που ανοίγουν νέα facilities είναι δυσκολότερο να επιλεχθούν. Επιπλέον, ο αλγόριθμος EMS αποκτά τη χειρότερη επίδοση ανοίγοντας παραπάνω facilities από ότι θα ήταν βέλτιστο.

Πακέτο: Euclid

$(m = 100, n = 100)\quad f = 3000$

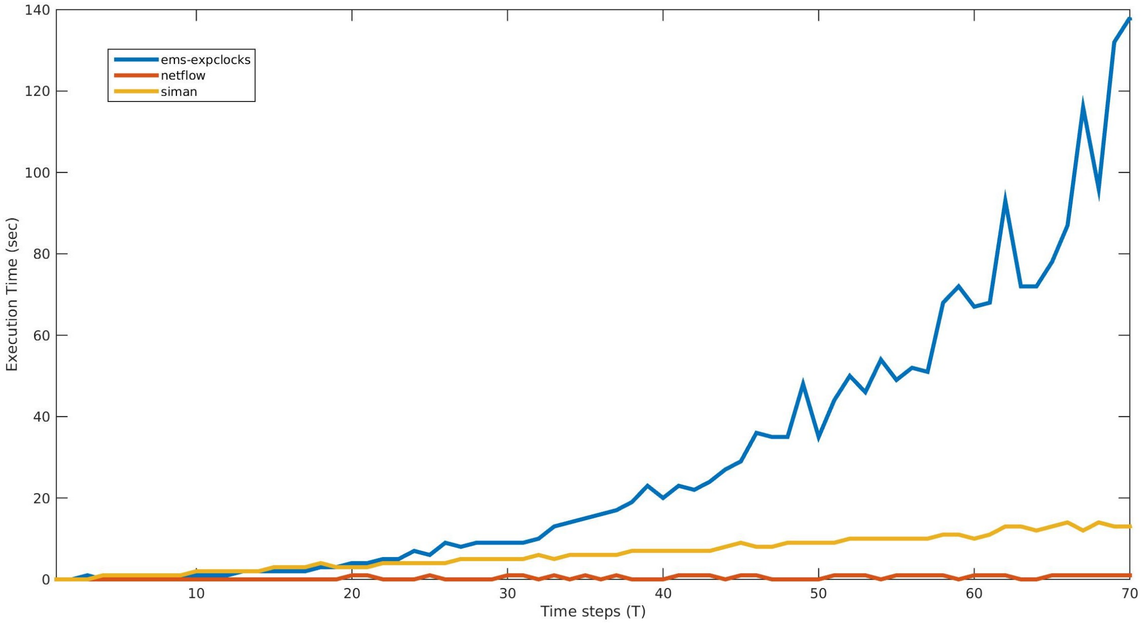



Αυτά τα μετρικά στιγμιότυπα παράχθηκαν από τους Kochetov και Ivanenko [14]. Σε ένα τετράγωνο πλευράς 7000 επιλέγονται $n$ σημεία. Κάθε επιλεχθέν σημείο αντιστοιχεί σε ένα facility και έναν client ($n = m$). Το κόστος λειτουργίας κάθε facility είναι 3000 και τα κόστη σύνδεσης αντιστοιχούν στην απόσταση στο Ευκλείδιο επίπεδο. Οι τιμές στρογγυλοποιούνται σε ακέραιες και ικανοποιούν την τριγωνική ανισότητα. Σε αυτό το σύνολο από στιγμιότυπα επιλέχθηκε $n = 100$. Οι αλγόριθμοι EMS και EXPCLOCKS παρουσιάζουν ταυτόσημη και βέλτιστη συμπεριφορά σε όλες σχεδόν τις εισόδους. Αυτό που παρατηρήθηκε κατά την εκτέλεση είναι ότι, στις προαναφερθείσες περιπτώσεις, η επίλυση του αντίστοιχου προβλήματος γραμμικού προγραμματισμού παρήγαγε τις βέλτιστες ακέραιες λύσεις. Όπου οι δύο αυτοί αλγόριθμοι διαφωνούν, η λύσεις ήταν κλασματικές αλλά με σχεδόν συντριπτική εμφάνιση ακέραιων τιμών 0 ή 1.

### 6.2.2 Ταχύτητα εκτέλεσης

Ένα σημαντικό κριτήριο για την αξιολόγηση ενός αλγόριθμου είναι η ταχύτητα εκτέλεσής του για διαφορετικά μεγέθη εισόδου. Σε αυτήν την υποενότητα θα εξετάσουμε πως μεταβάλλονται οι χρόνοι εκτέλεσης ως συνάρτηση των μεγεθών $T$, $m$ και $n$.

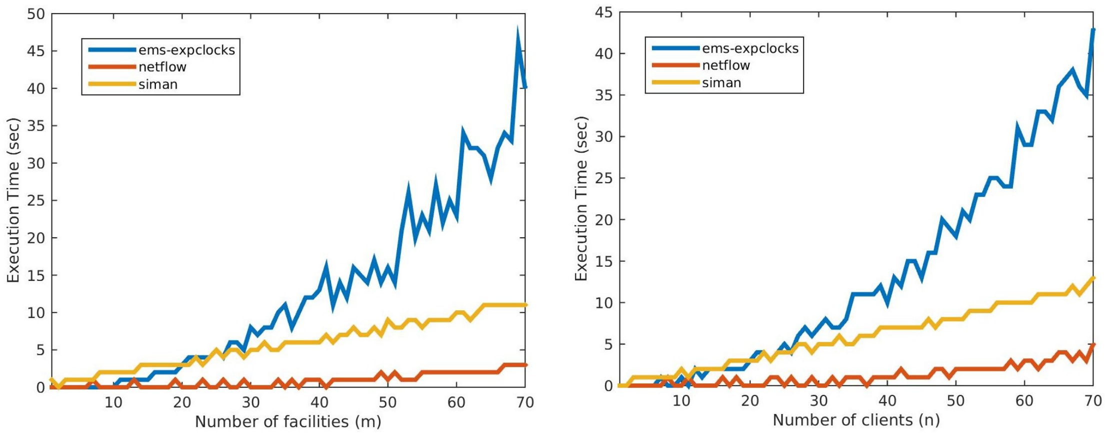
$(T = 1 \ to \ 70, m = 10, n = 10) \ \ f = 50, \ g = 50, \ d \sim U(1, 100)$




Στο παραπάνω διάγραμμα φαίνεται ξεκάθαρα πως οι αλγόριθμοι EMS και EXPCLOCKS είναι σημαντικά βραδύτεροι από τους άλλους δύο, γεγονός αναμενόμενο αφού καλούν μία υπορουτίνα επίλυσης ενός προβλήματος γραμμικού προγραμματισμού η οποία είναι πολύ χρονοβόρα για μεγάλα στιγμιότυπα.

$(T = 10) \ \ f = 50, \ g = 50, \ d \sim U(1, 100)$

$(m = 1 \ to \ 70, n = 10)$ $(m = 10, n = 1 \ to \ 70)$




Παρόμοιο μοτίβο παρατηρείται και για μεταβολές στο πλήθος των facilities ή των clients, ωστόσο η σχετική βραδύτητα μοιάζει να είναι μειωμένη κατά ένα σταθερό παράγοντα για τους αλγόριθμους EXPCLOCKS και EMS. Ο αλγόριθμος NETFLOW επηρεάζεται περισσότερο από μεταβολές σε αυτές τις δύο μεταβλητές παρά στον αριθμό των χρονικών βημάτων. Αυτό συμβαίνει διότι οι προαναφερθείσες μεταβλητές επηρεάζουν το μέγεθος του δικτύου στο οποίο γίνονται οι υπολογισμοί της ελάχιστης τομής, κάτι που ο χρόνος αφήνει ανεπηρέαστο.

Τα αποτελέσματα αυτά κρίνονται ιδιαίτερα σημαντικά καθώς αναδεικνύουν την αξία των αλγόριθμων NETFLOW και SIMAN. Σε μεγάλα στιγμιότυπα είναι οι μόνοι που μπορούν να μας παρέχουν καλές λύσεις σε εύλογο χρονικό διάστημα.

# Κεφάλαιο 7

# Επίλογος

Κλείνοντας την εργασία, κάνουμε μία ανασκόπηση των σημαντικότερων αποτελεσμάτων και συμπερασμάτων. Επίσης, προτείνουμε κάποιες κατευθυντήριες γραμμές στις οποίες μπορεί να κινηθεί η μελλοντική έρευνα.

## 7.1 Συμπεράσματα και Συνεισφορά

Η συνεισφορά της εργασίας έγκειται στην προσφορά ενός νέου αλγόριθμου για το πρόβλημα Χωροθέτησης σε Χρονικά Μεταβαλλόμενους Χώρους, ο οποίος παράγει την σχετική κλασματική λύση σε χρόνο που υπερτερεί σημαντικά συγκριτικά με τους LP Rounding αλγόριθμους από προϋπάρχουσες εργασίες. Αυτό το αποτέλεσμα είναι ιδιαίτερα χρήσιμο, καθώς μας δίνει τη δυνατότητα επίλυσης πολύ μεγαλύτερων στιγμιοτύπων τα οποία δεν μπορούσαμε να προσεγγίσουμε πριν. Επίσης, δείξαμε ότι για μη ακραίες τιμές του κόστους των facilities και του κόστους αλλαγής ανάθεσης, μία σχετικώς απλή υλοποίηση βασισμένη στη μέθοδο της προσομοιωμένης ανόπτησης προσφέρει πολύ καλά αποτελέσματα τα οποία είναι συχνά ανώτερα αυτών που παράγονται από κάθε άλλη γνωστή μέθοδο. Πέραν αυτών, ένα ενδιαφέρον θεωρητικό αποτέλεσμα ήταν η απόδειξη ενός νέου φράγματος για το κόστος σύνδεσης στον αλγόριθμο των Eisenstat et al. Οι ιδέες και οι τεχνικές που χρησιμοποιήθηκαν για αυτήν την απόδειξη πιθανώς να βοηθήσουν για την παραγωγή καλύτερων προσεγγιστικών αλγόριθμων.

## 7.2 Μελλοντικές κατευθύνσεις

Όπως συνέβη με το UFLP, θα ήταν πολύ σημαντικό αποτέλεσμα η εύρεση ενός αλγόριθμου με σταθερό παράγοντα προσέγγισης ο οποίος να μην καλεί μία υπορουτίνα επίλυσης ενός προβλήματος γραμμικού προγραμματισμού, καθώς αυτή καθιστά τον αλγόριθμο πολύ αργό σε μεγάλα στιγμιότυπα. Μία τέτοια ιδέα είναι η υλοποίηση ενός Primal Dual αλγόριθμου, ένα εγχείρημα που κρίνεται ωστόσο αρκετά δυσκολότερο για το υπό μελέτη πρόβλημα. Τέλος, μία μέθοδος για την βελτίωση των θεωρητικών αποτελεσμάτων και την κατασκευή νέων αλγόριθμων εντοπίζεται στην εκμετάλλευση των συνθηκών συμπληρωματικής χαλαρότητας που χαρακτηρίζουν το πρόβλημα, όπως φάνηκε στα πλαίσια της εργασίας.

# Βιβλιογραφία


[1] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder και Joseph Naor. A general approach to online network optimization problems. Στο *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, σελίδες 577–586. SIAM, 2004.

[2] Hyung-Chan An, Ashkan Norouzi-Fard και Ola Svensson. Dynamic facility location via exponential clocks. *CoRR*, abs/1411.4476, 2014.

[3] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala και Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[4] ME Aydin, V Yigit και TC Fogarty. Two approaches to simulated annealing for uncapacitated facility location problems. *submitted to European Journal of Operational Research*, 2002.

[5] Jaroslaw Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *CoRR*, abs/cs/0703010, 2007.

[6] Moses Charikar και Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. Στο *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, σελίδες 378–388. IEEE Computer Society, 1999.

[7] David Eisenstat, Claire Mathieu και Nicolas Schabanel. Facility location in evolving metrics. Στο *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, τόμος 8573 στο *Lecture Notes in Computer Science*, σελίδες 459–470. Springer, 2014.

[8] Uriel Feige. A threshold of ln $n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[9] Benchmark Instancesfor the Uncapacitated Facility Location Problem (U-FLP). Max-planck-institut für informatik. http://resources.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/, [Online; accessed 25/08/2016], 2016.

[10] Dimitris Fotakis. Online and incremental algorithms for facility location. *SIGACT News*, 42(1):97–131, 2011.

[11] Sudipto Guha και Samir Khuller. Greedy strikes back: Improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999.

[12] Kamal Jain και Vijay V. Vazirani. Approximation algorithms for metric facility location and $k$-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[13] GNU Linear Programming Kit. GLPK. https://www.gnu.org/software/glpk/ [Online; accessed 25/08/2016], 2016.

[14] Yu. Kochetov και D. Ivanenko. Computationally difficult instances for the uncapacitated facility location problem. Proceedings of the 5th Metaheuristic Conference (MIC 2003), Kyoto, 2003.

[15] Madhukar R. Korupolu, C. Greg Plaxton και Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1):146–188, 2000.

[16] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. Στο *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, τόμος 6756 στο *Lecture Notes in Computer Science*, σελίδες 77–88. Springer, 2011.

[17] Wolfram Alpha LLC. Linear Programming. [Online; accessed 25/08/2016] http://mathworld.wolfram.com/LinearProgramming.html, 2016.

[18] Carsten Lund και Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[19] Mohammad Mahdian, Yinyu Ye και Jiawei Zhang. Approximation algorithms for metric facility location problems. *SIAM J. Comput.*, 36(2):411–432, 2006.

[20] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie και Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. *CoRR*, αβς/1603.04954, 2016.

[21] Vinayaka Pandit. Local search heuristics for facility location problems. http://www.cse.iitd.ernet.in/˜pandit/thesis.pdf, [Online; accessed 25/08/2016], 2004.

[22] Maxim Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. Στο *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, May 27-29, 2002, Proceedings*, τόμος 2337 στο *Lecture Notes in Computer Science*, σελίδες 240–257. Springer, 2002.

[23] Shivakumar Vaithyanathan, Laura I. Burke και Michael A. Magent. Massively parallel analog tabu search using neural networks applied to simple plant location problems. *European Journal of Operational Research*, 93(2):317–330, 1996.

[24] Jens Vygen. Approximation algorithms for facility location problems (lecture notes). http://www.or.uni-bonn.de/~vygen/files/fl.pdf, [Online; accessed 25/08/2016], 2005.