Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Σχεδιασμός Μηχανισμών για Προβλήματα Χωροθέτησης σε Ευσταθή Στιγμιότυπα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΘΗΝΑ ΤΕΡΖΟΓΛΟΥ

Επιβλέπων : Δημήτριος Φωτάκης
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2022

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Σχεδιασμός Μηχανισμών για Προβλήματα Χωροθέτησης σε Ευσταθή Στιγμιότυπα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΘΗΝΑ ΤΕΡΖΟΓΛΟΥ

Επιβλέπων : Δημήτριος Φωτάκης
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Μαρτίου 2022.

....
Δημήτριος Φωτάκης
Αν. Καθηγητής Ε.Μ.Π.

....
Αριστείδης Παγουρτζής
Καθηγητής Ε.Μ.Π.

....
Ευάγγελος Μαρκάκης
Αν. Καθηγητής Ο.Π.Α

Αθήνα, Μάρτιος 2022

....

Αθηνά Τερζόγλου

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

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

Περίληψη

Σε αυτή τη διπλωματική, θα ασχοληθούμε με το σχεδιασμό μηχανισμών σε πρόβλημα χωροθέτησης υπηρεσιών, όπου n στρατηγικοί παίκτες έχουν διαφορετικές προτιμήσεις σε ένα μετρικό χώρο. Ένας μηχανισμός είναι μία συνάρτηση που έχει ως είσοδο τις τοποθεσίες των παικτών και επιστρέφει ως έξοδο τις θέσεις των υπηρεσιών. Κάθε παίκτης έχει στόχο να ελαχιστοποιήσει το κόστος του, δηλαδή την απόσταση του από τη κοντινότερη υπηρεσία. Αυτό το δίνει το κίνητρό να δηλώσει διαφορετική τοποθεσία από τη πραγματική του. Για αυτό το λόγο ενδιαφερόμαστε για φιλαλήθης μηχανισμούς που εξασφαλίζουν ότι κανένας παίκτης δεν θα ωφεληθεί δηλώνοντας ψευδή τοποθεσία. Για να ξεπεράσουμε τη αδυναμία κατασκευής ντετερμινιστικού φιλαλήθη μηχανισμού με φραγμένο λόγο προσέγγισης, εστιάζουμε τη προσοχή μας σε ένα υποσύνολο στιγμιοτύπων που είναι ευσταθή σε διαταραχές. Η έννοια της ευστάθειας σε διαταραχές ορίστηκε πρώτη φορά σε προβλήματα *Μέγιστης Τομής* και έπειτα σε προβλήματα Clustering. Με τα ευσταθή στιγμιότυπα μοντελοποιούμε τα στιγμιότυπα του “πραγματικού κόσμου” στα οποία η βέλτιστη λύση δεν επηρεάζεται από μικρές διαταραχές στα δεδομένα εισόδου. Αφού το πρόβλημα χωροθέτησης υπηρεσιών είναι πολύ στενά συνδεδεμένο με το clustering θέλουμε να δούμε πως μπορούμε να σχεδιάσουμε καλύτερους μηχανισμούς, αν υποθέσουμε ότι όλα τα πραγματικά στιγμιότυπα είναι ευσταθή. Δεδομένων των θετικών αποτελεσμάτων στη γραμμή, μας ενδιαφέρει να δούμε πως αυτά γενικεύονται και σε άλλους μετρικούς χώρους.

Λέξεις κλειδιά

Προβλήματα Χωροθέτησης, Σχεδιασμός μηχανισμών χωρίς χρήματα, Κοινωνική Επιλογή, Ευστάθεια σε διαταραχές

Abstract

In this thesis, we study k-Facility Location games, where n strategic agents have different ideal locations on a metric space. A mechanism maps the locations reported by the agents to k facilities. Each agent seeks to minimize her cost, namely the distance between her location and the nearest facility, which may incentivize her to misreport her location. Our goal is to design strategyproof (i.e, no agent can benefit by misreporting her location) mechanisms with a bounded approximation ratio to the optimal social cost. To overcome the impossibility results for deterministic strategyproof mechanisms, we restrict our attention in perturbation stable instances. The notion of perturbation stability was first introduced for the Max-Cut Problem and later was adapted for the Clustering Problems. This captures the "real world" instances in which the optimal solution is well-defined and unaffected by small perturbation on the input. Since k-Facility Location games and k-Clustering are closely related problems, we are interested to see how perturbation stability can help us design strategyproof mechanisms with stronger guarantees. Given the recent results in k-Facility location in stable instances on the line, we are interested in extending those results in more general metric spaces.

Key words

Facility Location Games, Mechanism Design without Money, Social Choice, Perturbation Stability

Ευχαριστίες

Κατ' αρχάς θα ήθελα να ευχαριστήσω τον επιβλέποντα μου κ. Φωτάκη, που με ενέπνευσε να ασχοληθώ με το τομέα της Θεωρητικής Πληροφορικής και που πίστεψε στις δυνατότητες μου από την αρχή. Επίσης θα ήθελα να ευχαριστήσω τον Παναγιώτη Πατσιλινάκο για την εξαιρετική συνεργασία και τη βοήθεια κατά την εκπόνηση της διπλωματικής μου εργασίας. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξαν όλα αυτά τα χρόνια αλλά και τους φίλους μου που ήταν πάντα δίπλα μου σε αυτό το όμορφο ταξίδι.

Αθηνά Τερζόγλου,

Αθήνα, 22η Μαρτίου 2022

Περιεχόμενα

Κατάλογος σχημάτων

Κεφάλαιο 1

Εκτεταμένη Ελληνική Περίληψη

1.1 Εισαγωγή

Στη παρούσα διπλωματική εργασία θα ασχοληθούμε με το πρόβλημα της χωροθέτησης υπηρεσιών. Υποθέτουμε ότι ένα σύνολο παικτών βρίσκονται σε ένα χώρο και εμείς επιθυμούμε να τοποθετήσουμε υπηρεσίες σε διαφορετικές θέσεις στο χώρο αυτό. Το κόστος κάθε παίκτη είναι η απόσταση του από τη πλησιέστερη υπηρεσία. Στόχος μας είναι να βρούμε τις βέλτιστες θέσεις ώστε το συνολικό κόστος των παικτών να είναι όσο το δυνατόν μικρότερο. Το συγκεκριμένο πρόβλημα έχει μελετηθεί εκτενώς τόσο στο πεδίο της επιστήμης των υπολογιστών όσο και στο πεδίο της επιχειρησιακής έρευνας. Ο λόγος είναι ότι μοντελοποιεί πληθώρα προβλημάτων, όπως τη επιλογή των θέσεων χτισίματος καινούριων βιβλιοθηκών από την κυβέρνηση βάση των προτιμήσεων των πολιτών.

Αν γνωρίζουμε τις πραγματικές τοποθεσίες των παικτών τότε μπορούμε να προσεγγίσουμε τη βέλτιστη λύση αρκετά καλά. Ωστόσο, υπάρχουν πολλές εφαρμογές που οι θέσεις των παικτών δεν είναι δημοσίως γνωστές και πρέπει να δηλώνονται στην κεντρική αρχή από στρατηγικούς παίκτες. Κάθε παίκτης έχει στόχο να ελαχιστοποιήσει το κόστος του, χωρίς να τον εδιαφέρει το συνολικό κόστος. Τώρα ο στόχος δεν είναι μόνο να βρούμε τις θέσεις των υπηρεσιών αλλά να τις βάλουμε με τέτοιο τρόπο που κανένας παίκτης δεν μπορεί να κερδίσει δηλώνοντας ψευδή τοποθεσία. Ενδιαφερόμαστε λοιπόν για το σχεδιασμό φιλαληθών μηχανισμών για για προβλήματα χωροθέτισης.

Σε πολλά προβλήματα σχεδιασμού μηχανισμών, όπως οι δημοπρασίες, εισάγουμε στο μοντέλο πληρωμές για να εγγυηθούμε ότι η ανάθεση των αγαθών γίνεται με φιλαλήθη τρόπο. Όμως, σε περιβάλλον κοινωνικής επιλογής όπως τα προβλήματα χωροθέτησης υπηρεσιών, μπορεί να είναι παράνομο ή ανήθικο να επιβάλουμε πληρωμές. Πάνω σε αυτή την ιδέα ξεκίνησε η εύρενα από τους rocaccia και Tennenholtz [44] πάνω στο σχεδιασμό προσεγγιστικών μηχανισμών χωρίς χρήματα.

1.2 Προβλήματα Χωροθέτησης

Πρώτα θα ορίσουμε το βασικό μοντέλο του προβλήματος. Έχουμε ένα μετρικό χώρο (X, d), όπου $d : X \times X \to \mathbb{R}_+$ είναι μια συνάρτηση απόστασης μεταξύ των σημείων του χώρου. Η συνάρτηση είναι μετρική που σημαίνει ότι $\forall x, y, z \in Z$: $d(x, x) = 0$ (ταύτιση), $d(x, y) = d(y, x)$ (συμμετρία) και $d(x, z) \leq d(x, y) + d(y, z)$ (τριγωνική ανισότητα). Οι παίκτες έχουν μια ιδανική τοποθεσία στον χώρο. Ένα στιγμιότυπο αποτελείται από τις θέσεις των παικτών στο χώρο $\vec{x} = (x_1, ..., x_n)$.

Κάθε παίκτης δηλώνει τη θέση του στο μηχανισμό M. Ένας ντετερμινιστικός μηχανισμός είναι μία συνάρτηση που απεικονίζει το σύνολο των δηλωθέντων θέσεων των παικτών $\vec{x}$ σε k υπηρεσίες $(c_1, ..., c_k)$. Αντίστοιχα ένας τυχαιοποιημένος μηχανισμός M απεικονίζει το σύνολο των δηλωθέντων θέσεων των παικτών $\vec{x}$ σε μια πιθανοτική κατανομή από k υπηρεσίες. $(c_1, ..., c_k)$.

Το κόστος κάθε παίκτη είναι η απόσταση του από τη πλησιέστερη υπηρεσία, $cost(x_i, M(\vec{x})) = min_{1 \leq j \leq k}\{d(x_i, c_j)\}$. Το κοινωνικό κόστος ενός μηχανισμού M είναι το άθροισμα των κοστών των παικτών, $cost(\vec{x}, M(\vec{x})) = \sum_{i=1}^{n} cost(x_i, \vec{C})$.

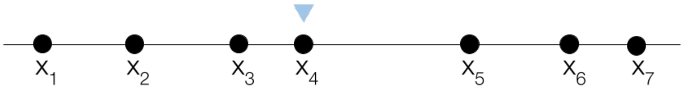
Κάθε παίκτης είναι στρατηγικός και έχει στόχο να ελαχιστοποιήσει το κόστος του. Ο μηχανισμός έχει στόχο να ελαχιστοποιήσει το κοινωνικό κόστος. Η διαφορά αυτή ωθεί τους παίκτες να δηλώσουν ψευδή τοποθεσία για να κερδίσουν. Για αυτό ενδιαφερόμαστε για φιλαλήθης μηχανισμούς που σημαίνει ότι κανένας παίκτης δεν μπορεί να κερδίσει λέγοντας ψέματα.

Στη συνέχεια θα δούμε τα βασικά αποτελέσματα για προβλήματα χωροθέτησης υπηρεσιών. Πρώτα θα αναλύσουμε τη περίπτωση που οι παίκτες είναι τοποθετημένοι στη γραμμή και στη συνέχεια τη περίπτωση που οι παίκτες είναι τοποθετημένοι σε δέντρα.

Μια υπηρεσία στη γραμμή

Όταν θέλουμε να τοποθετήσουμε μια υπηρεσία, η θέση του διάμεσου παίκτη είναι βέλτιστη και ταυτόχρονα είναι φιλαλήθης. Η βασική ιδέα είναι ότι ο μοναδικός τρόπος που ένας παίκτης μπορεί να αλλάξει τη θέση της υπηρεσίας είναι να δηλώσει μια θέση από τη "άλλη" μεριά του διάμεσου. Αυτό, όμως, θα έχει ως αποτέλεσμα η υπηρεσία να πάει πιο μακριά σε σχέση με τη πραγματική του θέση. Άρα δεν υπάρχει τρόπος κάποιος παίκτης να κερδίσει λέγοντας ψέματα.

Θεώρημα 1.2.1 ([40]). *Ο μηχανισμός που τοποθετεί την υπηρεσία στο διάμεσο παίκτη είναι βέλτιστος και φιλαλήθης.*



Σχήμα 1.1: Βέλτιστη λύση για μία υπηρεσία στη γραμμή

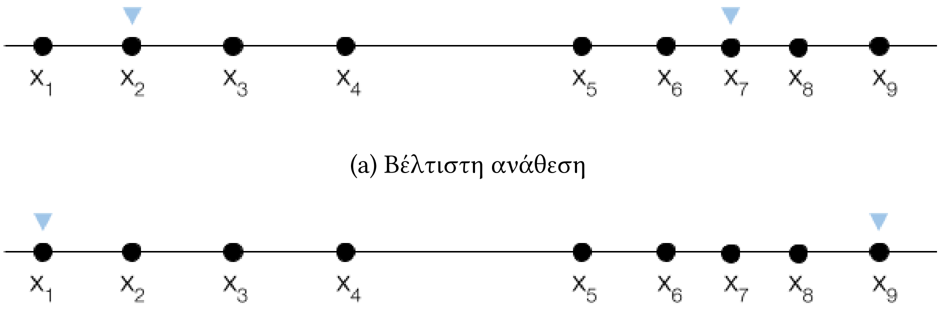
Δύο υπηρεσίες στη γραμμή

Όταν θέλουμε να τοποθετήσουμε δύο υπηρεσίες το πρόβλημα γίνεται πιο δύσκολο. Αρχικά, μπορούμε εύκολα να δούμε ότι η βέλτιστη λύση δεν είναι φιλαλήθης. Μπορούμε να χωρίσουμε τους παίκτες σε δύο σύνολα, το δεξί και το αριστερό, ανάλογα με την υπηρεσία που προτιμούν. Όμως ένας παίκτης μπορεί να δηλώσει μια θέση που να τον βάζει στο άλλο σύνολο από αυτό που ανήκει πραγματικά, έτσι ώστε να φέρει την υπηρεσία του εκείνου του συνόλου πιο κοντά στην πραγματική του θέση.

Ο μόνος τρόπος που μπορούμε να τοποθετήσουμε τις υπηρεσίες με φιλαλήθη τρόπο είναι να τις βάλουμε στο πιο δεξί και στον πιο αριστερό παίκτη. Παρατηρούμε ότι υποχρεωτικά ο πιο αριστερός παίκτης ανήκει στο αριστερό σύνολο και αντίστοιχα ο πιο δεξιά παίκτης ανήκει στο δεξί σύνολο. Με αυτό το τρόπο εξασφαλίζουμε φραγμένο λόγο προσέγγισης. Επιπλέον, τοποθετώντας τις υπηρεσίες στις ακριανές θέσεις παρατηρούμε ότι κανένας παίκτης δεν μπορεί να κερδίσει δηλώνοντας διαφορετική, αφού ο μόνος τρόπος να αλλάξει θέση μία υπηρεσία είναι να πάει πιο μακριά.

Θεώρημα 1.2.2 (Two Extremes [44]). *Ο μηχανισμός που τοποθετεί τις υπηρεσίες στους δύο ακριανούς παίκτες είναι φιλαλήθης και έχει λόγο προσέγγισης* $(n-2)$.

Ο Two Extremes είναι ο μοναδικός ντετερμινιστικός μηχανισμός με φραγμένο λόγο προσέγγισης [23]. Αν επιτρέψουμε στο μηχανισμό να χρησιμοποιήσει τυχαιότητα μπορούμε να πετύχουμε πολύ καλύτερο λόγο προσέγγισης. Ο proportional μηχανισμός είναι φιλαλήθης και έχει σταθερό λόγο προσέγγισης. Διαισθητικά τοποθετεί τη πρώτη υπηρεσία τυχαία σε μια από τις θέσεις των παικτών και τη δεύτερη τη τοποθετεί στη θέση κάποιου παίκτη με πιθανότητα ανάλογη της απόστασης του από τη πρώτη υπηρεσία.



(a) Βέλτιστη ανάθεση

(b) Ανάθεση στους δύο ακριανούς παίκτες

Ορισμός 1.2.1 (Propotional Μηχανισμός [34]). *Για ένα στιγμιότυπο $\vec{x} = (x_1, ...x_n)$, οι θέσεις των δύο υπηρεσιών επιλέγονται με τη παρακάτω τυχαία διαδικασία:*

- ***Βήμα 1:*** *Επιλέγουμε ένα $i \in N$ τυχαία. Τοποθετούμε τη πρώτη υπηρεσία c_1 στη θέση x_i*
- ***Βήμα 2:*** *Έστω $d_j = d(c_1, x_j)$ η απόσταση του παίκτη j από τη πρώτη τοποθεσία. Επιλέγουμε το j με πιθανότητα $\frac{d_j}{\sum_{k \in N} d_k}$. Τοποθετούμε τη δεύτερη υπηρεσία c_2 στη θέση x_j.*

Θεώρημα 1.2.3. *O Proportional μηχανισμός είναι φιλαλήθης και έχει λόγο προσέγγισης το πολύ 4.*

Περισσότερες από δύο υπηρεσίες στη γραμμή

Όπως είδαμε το πρόβλημα γίνεται αρκετά πιο δύσκολο ακόμα και αν θέλουμε να τοποθετήσουμε μόνο δύο υπηρεσίες στη γραμμή. Δεν μπορούμε λοιπόν να περιμένουμε καλύτερα αποτελέσματα όταν θέλουμε να τοποθετήσουμε περισσότερες από δύο υπηρεσίες. Στη πραγματικότητα μπορούμε να δείξουμε ότι δεν υπάρχει ντετερμινιστικός φιλαλήθης μηχανισμός με φραγμένο λόγο προσέγγισης για $k \geq 3$. Αυτό ισχύει ακόμα και για στιγμιότυπα με 4 παίκτες και 3 υπηρεσίες. Μπορούμε να δείξουμε ότι, όπως και πριν, οι δύο υπηρεσίες πρέπει να τοποθετηθούν στους δύο ακριανούς παίκτες για να είναι φιλαλήθης ο μηχανισμός. Η βασική ιδέα της απόδειξης είναι ότι δεν μπορούμε με φιλαλήθη τρόπο να επιλέξουμε σε ποιο παίκτη θα τοποθετηθεί η υπηρεσία.

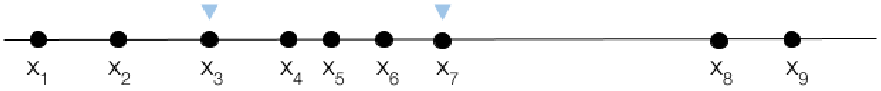
Θεώρημα 1.2.4 ([23]). *Δεν υπάρχει ντερμινιστικός φιλαλήθης μηχανισμός με φραγμένο λόγο προσέγγισης για $k \geq 3$.*

Υπάρχει μια πολύ σημαντική οικογένεια ντετερμινιστικών μηχανισμών για προβλήματα χωροθέτησης. Διαισθητικά η οικογένεια των percentile μηχανισμών "σπάει" το στιγμιότυπο σε τμήματα σύμφωνα με ένα διάνυσμα $\vec{p}$ και τοποθετεί μία υπηρεσία σε κάθε τμήμα.

Ορισμός 1.2.2 ($\vec{p}$-Percetntile Μηχανισμός [51]). *Ένας percentile μηχανισμός έχει ένα προκαθορισμένο διάνυσμα $\vec{p} = (p_1, ..., p_k)$ τέτοιο ώστε $0 \leq p_1 \leq ... \leq p_k \leq 1$. Ο μηχανισμός τοποθετεί τη j-οστή υπηρεσία στο p_j-οστό τμήμα του στιγμιότυπου.*

$$c_j = x_{i_j} \; : \; i_j = \lfloor (n-1) \cdot p_j \rfloor + 1$$

Παράδειγμα του $(0.25, 0.75)$-percentile μηχανισμού. Σε ένα στιγμιότυπο με 9 παίκτες ο μηχανισμός θα τοποθετεί τις υπηρεσίες στον 3ο παίκτη ($\lfloor 8 \cdot 0.25 \rfloor + 1 = 3$) και στον 7ο παίκτη ($\lfloor 8 \cdot 0.75 \rfloor + 1 = 7$), ανεξάρτητα από τις θέσεις των παικτών.



Αυτοί οι μηχανισμοί είναι φιλαλήθης για κάθε $k \geq 2$. Είναι πολύ σημαντικό να σημειώσουμε ότι το διάνυσμα είναι προκαθορισμένο και δεν εξαρτάται από το στιγμιότυπο. Αυτός είναι και ο λόγος που είναι φιλαλήθης οι μηχανισμοί. Οι υπηρεσίες τοποθετούνται σε συγκεκριμένες θέσεις στο στιγμιότυπο ανεξάρτητα από τις θέσεις των παικτών. Ο μόνος τρόπος κάποιος παίκτης να αλλάξει τη θέση μια υπηρεσίας είναι να δηλώσει τοποθεσία σε άλλο τμήμα. Όπως στη μία υπηρεσία και το διάμεσο παίκτη, αυτή η αλλαγή θα πάει την υπηρεσία πιο μακριά. Το μειονέκτημα αυτών των μηχανισμών είναι ότι δεν έχουν πεπερασμένο λόγο προσέγγισης για $k \geq 3$, όπως μας υποδηλώνει και το προηγούμενο θεώρημα. Ο μόνος percentile μηχανισμός με φραγμένο λόγο προσέγγισης είναι ο $(0, 1)$-percentile μηχανισμός που είναι ισοδύναμος με τον Two Extremes μηχανισμό. Σε κάθε άλλη περίπτωση μπορούμε να δημιουργήσουμε ένα στιγμιότυπο στο οποίο οι παίκτες που έχουν υπηρεσίες στις θέσεις τους να είναι κοντά μεταξύ τους ενώ οι παίκτες που δεν έχουν υπηρεσίες να είναι μακριά. Αφού οι μηχανισμοί δεν "κοιτούν" το στιγμιότυπο θα έχουν πολύ κακό λόγο προσέγγισης.

Είναι πολύ σημαντικό να παρατηρήσουμε ότι ο μόνος τρόπος για να έχουμε φραγμένο λόγο προσέγγισης είναι να τοποθετήσουμε μια υπηρεσία σε κάθε βέλτιστο cluster, δηλαδή σε κάθε ομάδα που εξυπηρετείται από την ίδια υπηρεσία στη βέλτιστη λύση. Όμως αυτό δεν μπορούμε να το πετύχουμε με φιλαλήθη τρόπο διότι η δομή των ομάδων αλλάζει όταν ένας παίκτης δηλώνει ψευδή τοποθεσία. Αυτό το εκμεταλλεύονται οι παίκτες για να κερδίσουν. Αν, από την άλλη, θέλουμε να έχουμε φιλαλήθης μηχανισμούς δεν πρέπει να "κοιτάμε" τη δομή των ομάδων χάνοντας όμως το πεπερασμένο λόγο προσέγγισης.

Από την άλλη πλευρά, μπορούμε να πετύχουμε καλύτερα αποτελέσματα αν επιτρέψουμε στους μηχανισμούς να χρησιμοποιήσουν τυχαιότητα. Πρώτα προτάθηκε ο Inversely Proportional Μηχανισμός [18], ο οποίος είναι $n/2$-προσεγγιστικός για k-υπηρεσίες με $n = k + 1$ παίκτες. Έπειτα, προτάθηκε ο Equal Cost [24] για k υπηρεσίες και αυθαίρετο αριθμό παικτών, ο οποίος πετυχαίνει λόγο προσέγγισης το πολύ n.

	$k = 1$	$k = 2$	$k \geq 3$
Ντερμινιστικός	1[40]	$n - 2$ [44]	∞ [25]
Τυχαιοποιημένος	1[40]	4 [34]	n [24]

Πίνακας 1.1: Συγκεντρωτικός πίνακας των καλύτερων λόγων προσέγγισης για προβλήματα χωροθέτησης k υπηρεσιών στη γραμμή.

Τοποθέτηση υπηρεσιών σε δέντρα

Τα δέντρα είναι ένας πιο γενικός μετρικός χώρος από τη γραμμή, με παρόμοιες ιδιότητες. Όταν θέλουμε να τοποθετήσουμε μια υπηρεσία, όπως και στη γραμμή, η βέλτιστη λύση είναι φιλαλήθης.

Θεώρημα 1.2.5 ([49]). *Ο μηχανισμός που τοποθετεί την υπηρεσία στο διάμεσο παίκτη είναι φιλαλήθης και βέλτιστος για το κοινωνικό κόστος.*

Η βασική ιδέα είναι ίδια με τη γραμμή. Ο μόνος τρόπος ένας παίκτης να αλλάξει τη θέση της υπηρεσίας είναι να δηλώσει ότι η θέση του είναι από την άλλη μεριά του διαμέσου. Έτσι όμως η υπηρεσία πηγαίνει πιο μακριά.

Αν όμως θέλουμε να τοποθετήσουμε δύο υπηρεσίες το πρόβλημα γίνεται πολύ πιο δύσκολο. Αρχικά να σημειώσουμε ότι η έννοια του πιο δεξιού και του πιο αριστερού παίκτη δεν μεταφέρεται στα δέντρα. Μπορούμε να δείξουμε ότι ακόμα και όταν δέντρο αποτελείται από κλαδιά (3 ημιευθείες $[0, \infty)$ με κοινή αρχή), δεν υπάρχει ντετερμινιστικός μηχανισμός για τη τοποθέτηση δύο υπηρεσιών [23].

1.3 Ευστάθεια σε Διαταραχές σε προβλήματα Συσταδοποίησης

Τους περισσότερους αλγορίθμους τους χαρακτηρίζουμε βάση της επίδοσης τους στο χειρότερο πιθανό στιγμιότυπο. Αν και είναι πολύ σημαντικό εργαλείο στην κατανόηση της απόδοσης ενός αλγορίθμου έχει και κάποια αρνητικά χαρακτηριστικά. Ένα από τα κυριότερα είναι ότι τα χειρότερα στιγμιότυπα, που μπορεί να επηρεάζουν σημαντικά την απόδοση του αλγορίθμου, δεν είναι πιθανό να εμφανιστούν στον πραγματικό κόσμο. Για αυτό, τα τελευταία χρόνια η έρευνα έχει στραφεί στην ανάλυση πέρα της χειρότερης περίπτωσης. Υπάρχουν διαφορετικές τεχνικές μέσα από τις οποίες μπορούμε μελετήσουμε τη μέση περίπτωση. Εμείς θα ασχοληθούμε κυρίως με την "ευστάθεια σε διαταραχές". Σε αυτό το μοντέλο θεωρούμε ότι η βέλτιστη λύση είναι καλά ορισμένη, που σημαίνει ότι δεν επηρεάζεται από μικρές αλλαγές στην είσοδο του προβλήματος. Με αυτό το τρόπο διαχωρίζουμε τα στιγμιότυπα που αξίζει να μελετήσουμε, αυτά δηλαδή που εμφανίζονται στη πράξη, από αυτά που δεν αξίζει.

Πιο συγκεκριμένα θα μελετήσουμε το πρόβλημα συσταδοποίησης σε ευσταθή στιγμιότυπα. Στο πρόβλημα συσταδοποίησης σκοπός είναι να χωρίσουμε ένα σύνολο σημείων του χώρου σε ομάδες, ώστε τα σημεία που ανήκουν στην ίδια ομάδα να είναι "όμοια" ενώ τα σημεία σε διαφορετικές ομάδες να είναι "διαφορετικά". Αυτό είναι ένα από τα πολύ γνωστά NP-hard προβλήματα. Παρόλα αυτά, σε ευσταθή στιγμιότυπα μπορούμε να βρούμε αποδοτικά τη βέλτιστη λύση.

Ορισμός 1.3.1 (k-Συσταδοποίηση (Clustering)). *Για ένα σύνολο σημείων X και μία μη-αρνητική συνάρτηση $d : X \times X \to [0, \infty)$ το clustering $\vec{C} = (C_1, C_2, ..., C_k)$ είναι ο διαχωρισμός των σημείων σε k μη-κενά σύνολα με κέντρα $c_1, ..., c_k$ τα οποία ελαχιστοποιούν το άθροισμα των αποστάσεων των σημείων από τα κέντρα τους (The k-median objective):*

$$\sum_{i=1}^{k} \left(\sum_{x \in C_i} d(c_i, x) \right)$$

Όπως περιγράψαμε και παραπάνω τα ευσταθή στιγμιότυπα του clustering είναι αυτά στα οποία η βέλτιστη λύση δεν επηρεάζεται από αλλαγές στην είσοδο. Ο μαθηματικός τρόπος να ορίσουμε αυτές τις "αλλαγές" είναι να δημιουργήσουμε κοντινά στιγμιότυπα στα οποία ένα υποσύνολο των αποστάσεων μεταξύ δύο σημείων έχει μειωθεί κατά ένα παράγοντα γ, με $\gamma \geq 1$. Άρα θα λέμε ένα στιγμιότυπο ευσταθές αν η βέλτιστη λύση παραμένει η ίδια σε κάθε πιθανό κοντινό στιγμιότυπο.

Ορισμός 1.3.2 (γ-διαταραχή). *Για $\gamma \geq 1$, μια γ-διαταραχή ενός μετρικού χώρου (X, d) είναι ένας άλλος χώρος (X, d') πάνω στα ίδια σημεία τέτοιος ώστε:*

$$d'(x, y) \in \left[\frac{1}{\gamma} d(x, y), d(x, y) \right]$$

Ορισμός 1.3.3 (γ-ευστάθεια). *Ένα στιγμιότυπο (X, d) είναι γ-ευσταθές για μία αντικειμενική συνάρτηση Φ αν υπάρχει ένα k-clustering $\vec{C} = C_1, ..., C_k$ τέτοιο ώστε, για κάθε γ-διαταραχή (X, d'), το $\vec{C}$ να παραμένει το μοναδικό βέλτιστο k-clustering.*

Βάση αυτού του ορισμού προκύπτουν 3 σημαντικές ιδιότητες για τις αποστάσεις μεταξύ σημείων από διαφορετικό cluster.

Ορισμός 1.3.4 (γ-Center Proximity). *Για κάποιο $\gamma \geq 1$, ένα γ-ευσταθές στιγμότυπο με μοναδικό βέλτιστο clustering $C_1, ..., C_k$ και βέλτιστα κέντρα $c_1, ..., c_k$ ικανοποιεί το γ-center proximity. Για δύο διαφορετικά cluster C_i και C_j και κάποιο σημείο $x_i \in C_i$:*

$$d(x_i, c_j) > \gamma d(x_i, c_i)$$

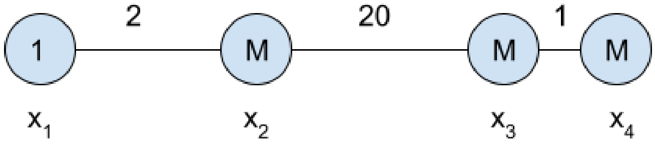
Λήμμα 1.3.1 (γ-Weak Center Proximity). *Για $\gamma \geq 2$, ένα γ-ευσταθές στιγμότυπο με μοναδικό βέλτιστο clustering $C_1, ..., C_k$ και βέλτιστα κέντρα $c_1, ..., c_k$ ικανοποιεί το $\gamma-$weak center proximity. Για κάποιο $x \in C_i$ και $y \notin C_i$:*

$$d(x, y) > (\gamma - 1)d(x, c_i)$$

Λήμμα 1.3.2 (Cluster Separation Property). *Έστω $C_1, ..., C_k$ το μοναδικό βέλτιστο clustering ενός γ-ευσταθούς στιγμιοτύπου με $\gamma \geq 2$. Για τα $x_i, x_i' \in C_i$ και το $x_j \in C_j$ $(i \neq j)$ ισχύει:*

$$d(x_i, x_j) > \frac{(\gamma - 1)^2}{2\gamma} d(x_i, x_i')$$

Για το clustering ένας από τους πιο συχνά χρησιμοποιούμενους αλγορίθμους είναι ο single-linkage. Ξεκινάει με n cluster (τα σημεία) και σε κάθε βήμα ενώνει τα δύο πιο "κοντινά" cluster, μέχρις ότου μείνουν k cluster. Για αυτό το παράδειγμα θεωρούμε ότι η απόσταση δύο clusters δίνεται από την ελάχιστη απόσταση ενός σημείου από το ένα cluster και ενός σημείου από το άλλο. Βλέπουμε ότι αυτός ο αλγόριθμος βασίζεται στις αποστάσεις μεταξύ των σημείων. Αν το στιγμιότυπο είναι ευσταθές μπορεί αυτός ο αλγόριθμος να βρει τη βέλτιστη λύση; Υπάρχει ένα απλό αντιπαράδειγμα που αυτός ο αλγόριθμος αποτυγχάνει. Ας υποθέσουμε ότι έχουμε ένα στιγμιότυπο με 4 διαφορετικές τοποθεσίες, έστω $\vec{x} = (x_1, x_2, x_3, x_4)$, ένας παίκτης είναι στη τοποθεσία x_1 και M παίκτες σε κάθε μία από τις υπόλοιπες. Στόχος μας είναι να τοποθετήσουμε 3 υπηρεσίες.

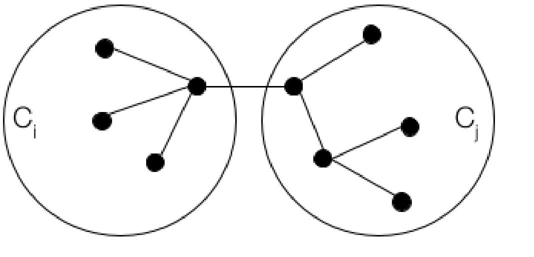


Σχήμα 1.3: Αντιπαράδειγμα του single-linkage

Είναι εύκολο να δούμε ότι η βέλτιστη λύση είναι να τοποθετήσουμε τα κέντρα στις θέσεις x_2,x_3 και x_4 και να εξυπηρετήσουμε τον x_1 από το κέντρο στο x_2. Το συνολικό κόστος της βέλτιστης λύσης είναι 2. Όμως ο single-linkage θα ενώσει σε ένα cluster τις θέσεις x_3 και x_4. Αυτή η λύση όμως έχει κόστος M.

Με μία μικρή τροποποίηση στον παραπάνω αλγόριθμο μπορούμε να παίρνουμε πάντα τη βέλτιστη λύση σε ευσταθή στιγμιότυπα. Αντί να σταματάμε στα k clusters συνεχίζουμε μέχρι να μείνει ένα cluster και στη συνέχεια με δυναμικό προγραμματισμό μπορούμε να βρούμε το βέλτιστο k-clustering.

Λήμμα 1.3.3. *O single-linkage αλγόριθμος βρίσκει πάντα τη βέλτιστη λύση αν, και μόνο αν, κάθε βέλτιστο cluster είναι υποδέντρο του ελάχιστου συνεκτικού δέντρου.*



Σχήμα 1.4: Τα βέλτιστα cluster είναι υποδέντρα του ΕΣΔ.

Το παραπάνω λήμμα ισχύει διότι ο single-linkage μπορεί να φτιάξει k-clusterings αφαιρώντας $k - 1$ ακμές από το δέντρο. Αν τα clusters δεν είναι υποδέντρα δεν υπάρχει τρόπος να τα φτιάξει.

Λήμμα 1.3.4 ([4]). *O single-linkage βρίσκει τη βέλτιστη λύση για κάθε 2-ευσταθές στιγμιότυπο.*

1.4 Προβλήματα Χωροθέτησης σε Ευσταθή Στιγμιότυπα

Όπως είδαμε και παραπάνω η βασική δυσκολία των προβλημάτων χωροθέτησης είναι:

(i) Αν θέλουμε φραγμένο λόγο προσέγγισης πρέπει να τοποθετήσουμε τις υπηρεσίες βάση των αποστάσεων μεταξύ των παικτών. Έτσι όμως οι μηχανισμοί δεν είναι φιλαλήθης.

(ii) Αν θέλουμε φιλαλήθης μηχανισμούς δεν πρέπει να "κοιτάμε" το στιγμιότυπο. Έτσι όμως οι μηχανισμοί δεν έχουν φραγμένο λόγο προσέγγισης.

Αν όμως υποθέσουμε ότι τα πραγματικά στιγμιότυπα είναι ευσταθή, μπορούμε να σχεδιάσουμε μηχανισμούς με καλύτερες εγγυήσεις; Τα βέλτιστα cluster στα ευσταθή στιγμιότυπα είναι εύκολα διαχωρίσιμα που σημαίνει ότι μπορούμε να τοποθετήσουμε μία υπηρεσία σε κάθε βέλτιστο cluster για να έχουμε φραγμένο λόγο προσέγγισης αλλά ταυτόχρονα οι παίκτες έχουν λιγότερη δύναμη στο να αλλάξουν τα βέλτιστα cluster. Αξίζει να σημειώσουμε ότι αυτό δεν είναι εφικτό χωρίς να υποθέσουμε ότι το στιγμιότυπο είναι ευσταθές, ακόμα και δύο υπηρεσίες στη γραμμή.

Η βασική διαφορά μεταξύ της προηγούμενης ενότητας και αυτής είναι ότι οι παίκτες είναι στρατηγικοί. Κάθε παίκτης έχει το δικαίωμα να δηλώσει οποιαδήποτε θέση επιθυμεί ανεξάρτητα από τη πραγματική του. Αντίθετα, οι διαταραχές που χρησιμοποιούμε για να δείξουμε ότι ένα στιγμιότυπο είναι ευσταθές παράγονται με συγκεκριμένο τρόπο. Από εδώ και πέρα, θα υποθέσουμε ότι τα πραγματικά στιγμιότυπα που θα έχει να χειριστεί ένας μηχανισμός είναι ευσταθή και θα μελετήσουμε τη δύναμη που έχει ένας παίκτη να αλλάξει την έξοδο του μηχανισμού.

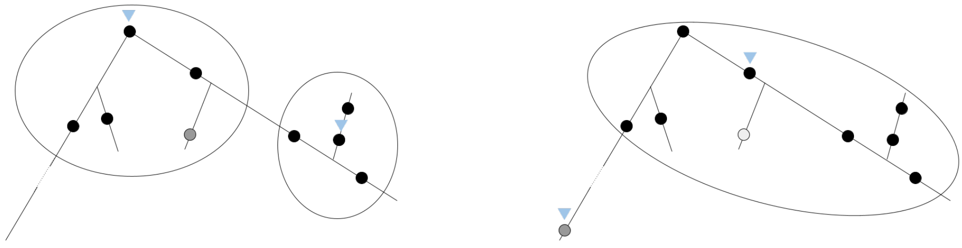
Για τα ευσταθή στιγμιότυπα έχουμε δύο πολύ σημαντικές ιδιότητες που θα μας φανούν πολύ χρήσιμες στη συνέχεια.

Λήμμα 1.4.1. *Έστω ένα γ-ευσταθές στιγμιότυπο $\vec{x}$, με $\gamma \geq 2$. Τότε κάθε βέλτιστο cluster είναι υποδέντρο του δέντρου.*

Λήμμα 1.4.2. *Έστω ένα γ-ευσταθές στιγμιότυπο, με $\gamma \geq 2 + \sqrt{3}$. Από το cluster separation property έχουμε ότι για κάθε τοποθεσία x_i κάθε εσωτερική απόσταση είναι μικρότερη από κάθε εξωτερική:*

$$d(x_i, x_j) \geq d(x_i, x_i') \; \forall x_i, x_i; \in C_i \;\&\; x_j \in C_j$$

Μια αρνητική παρατήρηση για τη βέλτιστη λύση είναι ότι δεν είναι φιλαλήθης αν περιέχει έστω και ένα cluster με ένα μόνο παίκτη (singleton cluster). Ας υποθέσουμε ότι το πραγματικό στιγμιότυπο είναι αυτή της εικόνας 1.5a. Αν ένας παίκτης δηλώσει μια θέση "πολύ μακριά" από τους υπόλοιπους παίκτες τότε θα έχει μια υπηρεσία στη θέση που δήλωσε ενώ το υπόλοιπο στιγμιότυπο θα εξυπηρετηθεί από $k-1$ υπηρεσίες. Όπως φαίνεται και στη εικόνα 1.5b τα δύο cluster θα ενωθούν και έτσι ο παίκτης που δήλωσε ψευδή θέση κατάφερε να φέρει μια υπηρεσία πιο κοντά στη πραγματική του θέση. Το πρόβλημα είναι ότι και τα δύο στιγμιότυπα είναι γ-ευσταθή και δεν μπορούμε με κάποιο τρόπο να διαχωρίσουμε αν το στιγμιότυπο είναι πραγματικό ή αν έχει προκύψει από κάποια αλλαγή. Παρόλα αυτά, στα πραγματικά στιγμιότυπα που μοντελοποιούν προβλήματα χωροθέτησης υπηρεσιών είναι φυσικό να υποθέσουμε ότι δεν υπάρχουν singleton clusters.



(a) Αρχικό στιγμιότυπο

(b) Το στιγμιότυπο μετά την αλλαγή.

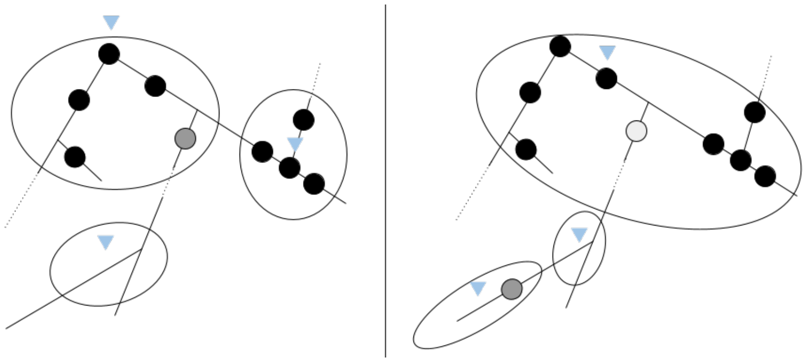
```
Algorithm 1: OPTIMAL
Result: Μια ανάθεση $k$-υπηρεσιών
Input: Ένα στιγμιότυπο $\vec{x}$ $k$-Facility Location.
Υπολογισμός του βέλτιστου clustering $(C_1, \dots, C_k)$. Έστω $c_i$ η διάμεση θέση του cluster $C_i$.
if $\big(\exists i \in [k]$ with $|C_i| = 1\big)$ or $\big(\exists\ x_i, x_i' \in Ci$ and $x_j, x_j' \in C_j$ with
  $\max\{d(x_i, x_i'), d(x_j, x_j')\} \geq d(x_i, x_j)\big)$ then
    Output: “Δεν τοποθετούνται υπηρεσίες”.
else
    Output: Οι $k$-υπηρεσίες τοποθετούνται στις θέσεις $(c_1, \dots, c_k)$
```

Το δεύτερο βήμα του αλγορίθμου λειτουργεί ως βήμα επαλήθευσης. Το cluster separation property είναι αναγκαία συνθήκη για την ευστάθεια, επομένως αν έχει παραβιαστεί είμαστε σίγουροι ότι κάποιος παίκτης έχει δηλώσει ψευδή τοποθεσία. Αν όμως ένα στιγμιότυπο “περάσει” τον έλεγχο δεν σημαίνει ότι είναι ευσταθές. Επίσης, να σημειώσουμε ότι μπορούμε αποδοτικά να ελέγξουμε όλες τις απαραίτητες αποστάσεις. Σύμφωνα με το λήμμα 1.4.1 κάθε βέλτιστο cluster είναι υποδέντρο του δέντρου, επομένως αρκεί να ελέγξουμε τις $k-1$ ακμές μεταξύ των διαφορετικών cluster.

Θεώρημα 1.4.1. *Ο* Optimal *μηχανισμός είναι φιλαλήθης και βέλτιστος για το πρόβλημα χωροθέτησης υπηρεσιών όταν το στιγμιότυπο είναι* $2 + \sqrt{3}$ *ευσταθές.*

Sketch. Έστω ένας παίκτης που ανήκει στο C_i δηλώνει μια άλλη τοποθεσία με σκοπό να φέρει μια υπηρεσία πιο κοντά στη πραγματική του τοποθεσία. Ο μόνος τρόπος που μπορεί να το πετύχει αυτό είναι να δηλώσει μια θέση x_i' τέτοια ώστε στη περιοχή που πήγε να χρειαστεί μια επιπλέον υπηρεσία για να τους εξυπηρετήσει (το τοπικό κόστος αυξήθηκε). Αυτό έχει ως αποτέλεσμα το C_i να ενωθεί με το γειτονικό του (το τοπικό κόστος μειώθηκε). Όπως φαίνεται και στην εικόνα από κάτω η υπηρεσία στο συνενωμένο cluster είναι πιο κοντά από την υπηρεσία που θα τον εξυπηρετούσε αν έλεγε την αλήθεια.

Όμως επειδή το στιγμιότυπο είναι ευσταθές είτε τα δύο γειτονικά cluster στη περιοχή που πήγε ο x_i θα παραβιάζουν την ελάχιστη εξωτερική απόσταση είτε αυτή η αλλαγή δεν θα είναι εφικτή. Μπορούμε να προσομοιώσουμε την αλλαγή στα κόστη με μια γ-διαταραχή. Αφού το αρχικό στιγμιότυπο είναι ευσταθές το βέλτιστο clustering δεν αλλάζει. Όμως, αυτό σημαίνει ότι ο μηχανισμός δεν θα έκανε αυτή την αλλαγή ούτε στο στιγμιότυπο που ο x_i έχει δηλώσει τη θέση x_i'.



Σχήμα 1.6: Πιθανή κερδοφόρα αλλαγή

□

1.5 Ανοιχτά προβλήματα

Στις παραπάνω ενότητες είδαμε τα βασικά αποτελέσματα για το πρόβλημα χωροθέτησης υπηρεσιών αλλά και από που πηγάζει η δυσκολία του προβλήματος. Επιπλέον, είδαμε πως αν εστιάσουμε στα "πραγματικά" στιγμιότυπα, από τη σκοπιά της ανάλυσης των αλγορίθμων πέρα της χειρότερης περίπτωσης, μπορούμε σχεδιάσουμε καλύτερους αλγορίθμους για προβλήματα που είναι πολύ δύσκολα στη γενική περίπτωση.Στον Optimal μηχανισμό είδαμε ότι πρέπει να αποκλείσουμε τα στιγμιότυπα που περιέχουν singleton clusters διότι ένας παίκτης μπορεί να δηλώσει μια θέση πολύ μακριά για να κερδίσει. Για να αντιμετωπίσουμε αυτό το πρόβλημα μια ενδιαφέρουσα ιδέα είναι να περιορίσουμε το εύρος των πιθανών τοποθεσιών που μπορεί να δηλώσει κάθε παίκτης σε σχέση με τη πραγματική του τοποθεσία. Σε αυτή τη περίπτωση πρέπει να δούμε πόσο ευσταθές πρέπει να είναι το στιγμιότυπο ώστε η βέλτιστη λύση να είναι φιλαλήθης. Μια άλλη κατεύθυνση είναι να "χαλαρώσουμε" τον ορισμό της ευστάθειας σε ένα που είναι πιο πιθανό να περιγράφει τα πραγματικά στιγμιότυπα. Έχει προταθεί στη βιβλιογραφία για το clustering τα (γ, ϵ)-ευσταθή στιγμιότυπα, στα οποία για κάθε γ-διαταραχή το πολύ ένα μικρό ποσοστό ϵ σημείων αλλάζουν cluster.

Chapter 2

Introduction

"How can a group of individuals choose a winning outcome from a given set of options?" This is a question that has been studied in various fields like sociology (social choice theory), economics (game theory), and most recently, computer science (algorithmic game theory and computation social choice). Throughout this thesis, we will focus on the algorithmic aspect of the previous question and, specifically, on how we can effectively achieve a desirable outcome. These types of problems are studied in the field of *Algorithmic Mechanism Design*, an intersection between *Economic Theory* and *Computer Science.* In classic mechanism design, the goal is to design a system for multi-participant environments. Depending on the setting, we are interested in different performance objectives like revenue maximization and social welfare maximization. Since the agents are strategic and rational, the system needs to ensure that all the agents will behave as the designer intended. Algorithmic mechanism design also considers computational constraints. Using tools from theoretical computer science while respecting game-theoretic constraints, the goal is to design efficient mechanisms.

2.1 Motivation

The *Facility Location Problem* is one of the most fundamental and well-studied problems in Theoretical Computer Science, Operation Research, and recently Algorithmic Mechanism Design. Such problems are motivated by natural scenarios in Social Choice, where the government plans to build a fixed number of public facilities in an area like schools, libraries, or hospitals(e.g., see [39]). Apart from locating actual facilities, the problem has a wide range of applications, including non-geographic problems such as selecting a committee to represent people with differing political views.

Let us look at it as an optimization problem first. The goal is to find the optimal placement of facilities to minimize transportation costs given a set of locations in a metric space. However, there are many applications where the locations are not publicly known and have to be reported by strategic agents. Now the goal is to design a strategyproof mechanism, i.e., does not incentivize agents to lie, which is also efficient with respect to the optimal solution. In many settings in Mechanism Design (e.g., auctions), payments guarantee that the optimal solution is strategyproof. On the other hand, payments could be illegal or unethical in Social Choice environments, such as Facility Location games[48]. Procaccia and Tennenholtz [44] showed that strategyproofness could also be achieved without payments by sacrificing the solution's optimality, thus initiating the research on *Approximate Mechanism design without Money.*

Since then, the problem has been studied in many different settings and generalizations. There are two survey papers to get a complete overview of the problem: one for mechanism design for Facility Location Problems [15] and one for approximation algorithms for Facility Location and Clustering Problems [3]. In this thesis, we are going to focus on the mechanism design aspect of the problem. This problem has been studied for different metric spaces. One of the most researched setting is the k-Facility Location on the line ([23, 24, 30, 35, 42]). It is also been researched for restricted metric spaces more general that the line(e.g., trees, circles, and plane [2, 17, 21, 29, 38]) and general metric spaces ([25, 34]). Another direction is to design mechanism for different objective functions, like *maximum cost* [44, 24] and *mini-sum-of-squares*[20]. For all those variations we assume that all the facilities are homogeneous. Recently, there is some work where the facilities serve different purposes ([33, 32, 50]). There are also considered not single-peaked preference profiles ([37, 16, 19])

In this thesis, we are going to focus on the classic k-Facility Location games, where k uncapacitated facilities are placed in a metric space based on the preferences of n strategic agents. Our goal is to minimize the social cost objective, namely the sum of the distances from each agent to the nearest facility. When the agents are located on the real line, we have a complete characterization of deterministic strategyproof mechanisms. For one facility, the mechanism that places the facility at the median location is strategyproof and optimal with respect to the social cost [44]. For two facilities, the only strategyproof mechanism that exist places the facility at the leftmost and the rightmost locations (Two Extremes mechanism) and has an approximation ratio at most $n - 2$ [44, 23]. However, for three or more facilities, there is no deterministic anonymous strategyproof mechanisms for k-Facility Location with a bounded approximation ratio [23]. On the positive side, randomized mechanism achieve better approximation ratios. For 2-Facility Location games Proportional Mechanism achieves a constant approximation ratio of 4 [34]. The Inversely Proportional Mechanism [18] has $n/2$-approximation ratio for k-facility location games with $n = k + 1$ agents. Fotakis and Tzamos [24] proposed the Equal Cost a randomized strategyproof mechanism with an approximation ration at most n for any number of agents.

	$k = 1$	$k = 2$	$k \geq 3$
Deterministic	1[40]	$n - 2$ [44]	∞ [25]
Randomized	1[40]	4 [34]	n [24]

Table 2.1: Best known results of approximation ratio for k-Facility Location on the line

When the agents are located on more general metric spaces than the line, the problem becomes much more complicated. For tree metrics, for one facility the mechanism that places the facility at the median location is strategyproof and optimal [49], but for two or more facilities there is no deterministic strategyproof mechanism with a bounded approximation. When the agents are located in a circle then any strategyproof mechanism is a dictatorship.

We can overcome the difficulty of the problem by giving "additional power" to the mechanisms. Nissim, Smorodinsky, and Tennenholtz first proposed the class of *imposing mechanisms* [42]. In that class, the mechanism can restrict the way agents are allowed to exploit the outcome. In the Facility Location setting, for example, agents are forced to connect to the facility closest to their reported location, even if there is a facility closer to their true location. This way, an agent that misreports has a greater connection cost since she cannot select the facility closest to her ideal location. Using the same techniques, Fotakis and Tzamos [25] showed that the Winner Imposing version of the Proportional mechanism for the k-Facility Location is strategyproof and has an approximation ratio of at most $4k$. The mechanism requires that any agent that has a facility at her reported location connect to it.

Another way to ensure strategyproofness and better approximation ratios is to consider *mechanisms with local verification.* This was first introduced by Green and Laffont [31]. In order to model partial verification, they restricted the set of each agent's allowable deviations to a so-called correspondence set. Later, Nisan [41] proposed a mechanism with verification for scheduling problems in unrelated machines, where each machine is controlled as a selfish agent. The mechanism has two stages: first, in the declaration phase, the agents report their types(namely the time of execution for each job) and the mechanism returns an allocation; then, in the execution phase, each machine executes the tasks it is allocated. The payments are given after the execution, so the mechanism can verify if the agents reported their true types and punish those who lied by reducing their payments. This notion of local verification has been adapted for different settings, see for example [6, 14, 13, 5, 27, 26].

Note that in the class of imposing mechanisms, the mechanism does not know whether or not an agent lied, but the way it restricts any agent's post-action options ensures that any lying agent will be penalized. On the other hand, a mechanism with verification can identify agents that lied and punish them.

2.2 Beyond Worst Case Analysis

The previous results are based on the worst-case analysis framework in which an algorithm is characterized by its performance on the worst possible input. A good worst-case guarantee shows that an algorithm works well without any assumptions. However, in many interesting problems, this is impossible. To get better insight about an algorithms performance we want to move beyond the worst-case [45].

Take the problem of linear programming, where the goal is to minimize (or maximize) a linear objective function subject to linear constraints. Two algorithms solve this problem: the simplex method (exponential time in the worst-case) and the ellipsoid method (polynomial in the worst-case). Yet, empirically, simplex performs way better than the ellipsoid method.

In this work we are going to focus on *Perturbation Stability*. The notion of perturbation stability was first introduced by Bilu and Linial [12] for the Max-Cut Problem. Intuitively stability implies that small perturbation on the input does not change the optimal solution. By restricting our attention to stable instances, that model "real world" instances, we can find the optimal solution. This was later adapted for clustering problems [9, 7, 8, 4, 1]. Clustering is the task of grouping a set of points in such a way that objects in the same group are more similar (in some sense) to each other than to those in other groups. Clustering, as an optimization problem, is NP-hard for most commonly used objective functions (e.g, k-median[36], k-center). However, simple clustering algorithms perform well in practice because the clusters are well defined. The notion of stability captures the structure that is usually found in practical instances. Therefore, by restricting our domain to a subset of instances, namely perturbation stable instances, we can design exact algorithms for an NP-hard problem.

We will call a clustering instance γ-*perturbation stable* if the optimal clustering remains the same even if we scale down any subset of the entries of the distance matrix by a factor of at most γ. By focusing on the geometry of the instance, we show that in stable instances, all the inter-cluster distances are bounded by some intra-cluster distance. We will show three properties that hold in any γ-stable instance: *Center-Proximity* (Definition 4.2.1), *Weak Center-Proximity* (Lemma 4.2.1) and *Cluster Separation Property* (Lemma 4.2.2).

2.3 Facility Location on Stable instances and Contribution

Since clustering and Facility Location games are closely related problems, an interesting question is whether we can achieve better results using the ideas and techniques from perturbation stability. The main difficulty in both problems is identifying the optimal clusters, namely the groups of points or agents that are served by the same center or facility. However, in Facility Location games, the agents are strategic, which means even if only one agent misreports her ideal location, the new instance can have a very different clustering structure than the original. Therefore, the question actually is how much power an agent has in manipulating the output of a mechanism assuming stability of the instance and, consequently, well-defined clusters.

Fotakis and Patsilinakos [22] initiate the study in that direction. The main idea is that the class of instances that appear in the real world are stable, and therefore, the mechanism should perform well only in those instances. Therefore, they consider the k-Facility Location games on the line restricted on perturbation stable instance. They showed that the optimal solution is strategyproof for $(2+\sqrt{3})$-stable instance, if the optimal clustering does not include any singleton clusters. Let us point out that without any assumptions about the input, the optimal solution is not strategyproof, even for 2-Facility location games on the line. To avoid the restriction on stable instances with out singleton clusters the also proposed a randomized strategyproof mechanism with a constant approximation ratio for 5-stable instances. Moreover, focusing on stable instances does not make the problem trivial. Extending the impossibility result for k-Facility Location games with $k \geq 3$, they also proved that there is no deterministic anonymous strategyproof mechanism for k-Facility Location, with $k \geq 3$, on $(2 - \delta)$-stable instances with bounded approximation ratio for any $\delta > 0$.

Our goal in this theses is to see how we can extend the previous results into more general metric spaces than the line. In the unrestricted domain the problem becomes non-trivial even when we want to place one in a circle or two facilities in a tree. We show that the optimal solution is strategyproof for k-Facility Location for $2 + \sqrt{3}$ in tree metrics.

Chapter 3

Mechanisms for Facility Location games

In this chapter, we will present the main results of the Facility Location Problem in more detail. To better understand the problem, we will start with some useful definitions from the relevant field of Social Choice. Then we will formally define the setting and the basic concepts of *Algorithmic Mechanism Design.*

3.1 Social Choice and Single Peaked Preferences

Social choice theory is the study of collective decision processes and procedures. How can a group of individuals choose a winning outcome (e.g., policy, electoral candidate) from a given set of options? In the most general setting there is a set N of n agents and a set $\mathcal{A}$ of alternatives. Each agent has a private order of the alternatives $\succ_i$ over the alternatives in $\mathcal{A}$. Let L be the set of all linear orders of $\mathcal{A}$. A *social choice function* $f : L^n \rightarrow \mathcal{A}$ maps the agents' preferences to a single alternative. A social choice function must satisfy the following properties:

Definition 3.1.1 (Unanimous). *A social choice function f is unanimous if, when all player prefer a certain outcome more than anything else, then that outcome must be the alternative chosen by the mechanism. That is, if $\exists a \in \mathcal{A}$ such that $\forall b \in \mathcal{A}$ and $i \in N$, $a \succ_i b$ then $f(\succ_1, ..., \succ_n) = a$.*

Definition 3.1.2 (Onto). *A social choice function f is onto if any alternative can be reached given the appropriate preference profiles That is, $\forall a \in \mathcal{A}$, $\exists \vec{x} \in L^n$ such that $f(\vec{x}) = a$.*

Definition 3.1.3 (Pareto Optimal). *A social choice function f is Pareto optimal if no other alternative is more preferred by every agent than the alternative chosen by the mechanism. That is, if $f(\vec{x}) = a$ for a $\vec{x} \in L^n$, then $\nexists b \in \mathcal{A}$ such that $b \succ_i$, $\forall i \in N$.*

Definition 3.1.4 (Strategyproof). *A social choice function f is strategyproof if no agent can change the outcome to a more preferable by misreporting her preferences. That is, for all preference profiles $\succ_1, ..., \succ_n$ and any agent i, and any alternative preference $\succ_i'$: $f(\succ_1, ..., \succ_i, ..., \succ_n) \succ_i f(\succ_1, ..., \succ_i'$ $, ..., \succ_n)$.*

Our goal is to find the best possible outcome given the agents' preferences. The first three properties ensure that the selected outcome is efficient, namely desirable for all the agents. The agents, on the other hand, are selfish and strategic, aiming to maximize their utility. This is why, if we want the social function to be fair, the fourth property is crucial.

An other important property of social choice function is *anonymity*, namely that the selection is based on the preference profiles only and not on the agent reporting it. This implies that all agents count the same in decision making.

Definition 3.1.5 (Anonymous). *A social choice f is anonymous if for all permutations π and all profiles $(\succ_1, ..., \succ_n) \in L^n$: $f(\succ_1, ..., \succ_n) = f(\succ_{\pi(1)}, ..., \succ_{\pi(n)})$*

There are social choice functions that do not satisfy anonymity. For example, a *dictatorial* social choice function that always selects the most preferred alternative for a particular agent, ignoring the preference profiles of the rest of the agents. We will call this agent a dictator. A dictatorial social choice function is strategyproof and Pareto optimal, but not fair because not all agents count the same.

Definition 3.1.6 (Dictator). *An agent i is a dictator in a social choice function f if for all $\succ_1, ..., \succ_n \in L$: $f(\succ_1, ..., \succ_n) = a$ where $a \succ_i b, \forall b \in \mathcal{A}$ with $b \neq a$.*

A simple and intuitive example of a social choice function is the majority vote. If there are only two alternatives then the majority is strategyproof. If there are 3 or more alternatives, however, the majority is not strategyproof. An agent may change her preference profile to prevent her least liked option from being selected. Unfortunately, there is an impossibility result that states we cannot do anything better than a dictatorship when there are more that 3 alternatives.

Theorem 3.1.1 (Gibbard [28] - Satterthwaite [47]). *Let f be an incentive compatible social choice function onto $\mathcal{A}$, where $\mathcal{A} \geq 3$, then f is a dictatorship.*

One way to overcome the impossibility result of the previous theorem is to introduce payments into the model. There are many strategyproof mechanisms with money. The idea is that if an agent misreports in an attempt to benefit, the extra payment from the mechanism will be more than the actual decrease in the cost. However, as we mentioned before, we are interested in applications where payments are not an option. So, in order to overcome the impossibility result without money, we are going to restrict the domain of possible preference profiles to single-peaked preferences. Preferences are said to be single-peaked if the alternatives can be represented as points on a line, and each agent has a unique most preferred point (“”peak” preference) and points that are further from her peak are preferred less. It is natural to assume that the agents have single-peaked preferences for many problems. For example, in the Facility Location setting, where the government wants to place public facilities, any agent would prefer the facility closest to her ideal location since any other facility will only increase the transportation cost.

Definition 3.1.7 (Single-Peaked Preferences). *Preferences are said to be single-peaked if the alternatives can be represented as points on a line, and there exists an alternative $a \in \mathcal{A}$ (the peak) such that if $x < y < a \implies y \succ x$ and if $a < x < y \implies x \succ y$.*

Moulin [40] showed that if all the agents have single-peaked preferences, there exists a strategyproof mechanism that only depends on the peak of each agent.

Theorem 3.1.2. *Assuming single peakedness, a rule f is strategyproof, onto and anonymous if and only if there exists $a_1, a_2, ..., a_n \in [0, 1]$ such that for all peaks $(x_1, ..., x_n) \in \mathbb{R}$.*

3.2 Preliminaries for Facility Location Games

By the definition of single-peaked preferences we can see that a profile is single-peaked if for any two points on the same side of the peak the agent always prefers the one which is closer to her peak. But there is no way to now by “how much” agent i prefers alternative x over y. From now on, we are going to assume that the agents rank the all the alternative locations based on the distance from their peak.

Definition 3.2.1 (Facility Location). *Let $N = \{1, .., n\}$ be a set of agents. The agents are located in a metric space (X, d), where $d : X \times X \to \mathbb{R}_{\geq 0}$ is the distance function. The function d is a metric on X satisfying $d(x, x) = 0$ for all $x \in X$, $d(x, y) = d(y, x)$ for all $x, y \in X$ (*symmetry*) and, $d(x, z) \leq d(x, y) + d(x, z)$ for all $x, y, z \in X$ (*triangle inequality*). Each agent $i \in N$ has a private location $x_i \in X$. We refer to the tuple $\vec{x} = (x_1, ...x_n)$ as the* location profile *or* instance.

For a location profile $\vec{x}$ and an agent i, let $\vec{x}_{-i}$ denote the tuple $\vec{x}$ without the coordinate x_i. Similarly, for a non-empty set S of indices, let $\vec{x}_S = (x_i)_{i \in S}$ be the locations of agents in S and $\vec{x}_{-S} = (x_i)_{i \notin S}$ the tuple $\vec{x}$ without the location of agents in S.

Each agent reports her ideal location to the mechanism M. A *deterministic mechanism* M for k-Facility Location maps an instance $\vec{x}$ to a k-tuple $\vec{c} = (c_1, ..., c_k) \in X^k$. We let $M(\vec{x})$ denote the outcome of M in instance $\vec{x}$. Similarly, a *randomized mechanism* M maps an instance $\vec{x}$ to a probability distribution over k-tuples $(c_1, ..., c_k) \in X^k$.

For a location profile $\vec{x}$ and a mechanism M, we define the *connection cost* of agent i as the minimum distance of her private location to the closest facility, $cost(x_i, M(\vec{x})) = min_{1 \leq j \leq k}\{d(x_i, c_j)\}$. The *social cost* of a mechanism M is the total distance of the agents' locations to the nearest facility, $cost(\vec{x}, M(\vec{x})) = \sum_{i=1}^{n} d(x_i, M(\vec{x}))$.

Since each agent is strategic, it is easy to see that the goal of each agent is different from the mechanism's. The mechanism's objective is to minimize the social cost, but each strategic agent seeks to minimize its connection cost. This divergence between the two goals motivates an agent to manipulate the mechanism by reporting a false location to achieve a better connection cost. This is why strategyproofness is a crucial property that any mechanism should satisfy.

Definition 3.2.2 (Strategyproof). *A mechanism is strategyproof if for all location profiles $\vec{x}$, any agent i, and all locations y:*

$$cost(x_i, M(x)) < cost(x_i, M(x_{-i}, y))$$

A mechanism can also be *group strategyproof* if for any coalition misreporting their location simultaneously, at least one does not benefit. Formally:

Definition 3.2.3 (Group Strategyproof). *A mechanism is strategyproof if for location profiles $\vec{x}$, all coalitions $S \subset N$ and all location profiles $\vec{x}\,' = (\vec{x}_{-S}, x'_S)$, there exists an agent $i \in S$ such that*

$$cost(x_i, M(\vec{x})) \leq cost(x_i, M(\vec{x}\,'))$$

Definition 3.2.4 (Image Set). *For any mechanism M, the* image set *of agent i with respect to a location profile $\vec{x}_{-i}$ is the set of all the possible facility locations the agent can obtain by varying her reported location. Formally:*

$$I_i(\vec{x}_{-i}) = \{a \in X : \exists\, y \in X \ with\ M(\vec{x}_{-i}, y)\}$$

We can see the image set as the power an agent has on the mechanism. Any strategyproof mechanism M always outputs some location in $I_i(\vec{x}_{-i})$ that is closest to the reported location as shown in the following lemma.

Lemma 3.2.1. *Let M be a strategyproof mechanism for the k-Facility game. For every location profile $\vec{x} \in X^n$ and any agent $i \in N$ we have:*

$$cost(y, M(\vec{x}_{-i}, y)) = \inf_{a \in I_i(\vec{x}_{-i})} \{d(a, y)\}$$

Proof. For the location profile $\vec{x}\,' = (\vec{x}_{-i}, y)$ let $a \in M(\vec{x}\,')$. Assume for contradiction that exists $a^* \in I_i(\vec{x}_{-i})$ such that $d(a^*, y) < d(a, y)$. By the definition of the image set there exists a y^* such that $a^* \in M(\vec{x}_{-i}, y^*)$. Then, if agent i is located at y she can benefit by misreporting to y^* lowering her connection cost from $cost(y, M(\vec{x}_{-i}, y))$ to $d(a^*, y)$. This contradicts the assumption that M is strategyproof □

We can extend the previous definition of image set from a single agent to a group of agents. For a given mechanism M we define the image set of agents in a subset S with respect to a location profile $\vec{x}_{-S}$ as the set of all possible facility locations they can obtain by varying their reported location:

$$I_S(\vec{x}_{-S}) = \{a \in X : \exists\, \vec{y} \in X^{|S|} \ with\ M(\vec{x}_{-S}, \vec{y})\}$$

We can also extend the previous lemma to hold for partial group strategyproof mechanisms, when all agents in the coalition report the same location.

Lemma 3.2.2. *Let M be a strategyproof mechanism for the k-Facility Location game. For every location profile $\vec{x} \in X^n$, any non-empty set of agents $S \subset N$, $\vec{y}$ and $\vec{y} = (y, ..., y)$ we have:*

$$cost(y, M(\vec{x}_{-S}, \vec{y})) = \inf_{a \in I_S(\vec{x}_{-S})} \{d(a, y)\}$$

3.3 Facility Location on the Line

In this section, we are going to focus on *Facility Location* on the real line. Let $N = \{1, ..., n\}$ be the set of agents. Each agent has a private ideal location $x_i \in \mathbb{R}$. We can assume without loss of generality that the locations in the location profile $\vec{x} = (x_1, ..., x_n)$ are ordered $(x_1 \leq x_2 \leq ... \leq x_n)$, since we focus on anonymous mechanisms. Given an instance $\vec{x}$ the most important locations are the leftmost and the rightmost because they delimit the instance. We denote the leftmost location as $lt(\vec{x}) = min_{i \in N}\{x_i\}$ and the rightmost location as $rt(\vec{x}) = max_{i \in N}\{x_i\}$. The distance function between any two locations is simply the length of the interval, $d(x, y) = |x - y|$.

3.3.1 Locating one facility

We start with the simplest setting. Given an instance $\vec{x}$ we want to place one facility that achieves the best possible social cost in a strategyproof way.

Theorem 3.3.1. *The mechanism that selects the median location of the reported instance is strategyproof and optimal for the social cost objective.*

Proof. We first show that the median location, $med(\vec{x})$, is the optimal solution. If n is odd the median location is $x_{(n+1)/2}$, if n is even any location in the interval $[x_{n/2}, x_{n/2+1}]$ is optimal without lost of generality we consider $x_{n/2}$ to be the selected location. Suppose the facility is placed at a location to the left of the median location, let that location be x. Then, at least half of the agents' connection cost has increased by $d(x, med(\vec{x}))$, and at most half of the agents' connection cost has decreased by $d(x, med(\vec{x}))$. This makes the social cost of x higher than the optimal. The same holds for any location selected to the right of the median.

We now show that the mechanism is strategyproof. The agent located at $med(\vec{x})$ has no initiative to misreport her location, since her connection cost is zero. Suppose an agent i located at x_i misreports to x'_i. Without loss of generality we can assume that $x_i < med(\vec{x})$. If $x'_i < med(\vec{x})$ then the output of the mechanism does not change. If $x'_i > med(\vec{x})$ the median location moves to the right leading to an increase in her cost. □

3.3.2 Locating two facilities

This section extends the previous setting from locating one facility on the real line to locating two. Now a mechanism returns a 2-tuple $\vec{c} = (c_1, c_2)$ with the location of the facilities; we can assume that $c_1 \leq c_2$. Each agent connects to the nearest facility with connection cost equal to $cost(x, \vec{c}) = min\{d(x, c_1), d(x, c_2)\}$.

Let us focus on the optimization problem of locating the two facilities without considering the strategic agents' motives. Let $\vec{x}$ a location profile, and c_1,c_2 the optimal facility locations. We can split the agents into two (non-empty) groups based on which facility they prefer. The agents that prefer c_1 belong to the "left" set $L(\vec{x})$, and the rest belong to the "right" set $R(\vec{x})$. Using the same argument as in the previous setting for one facility on the line, given the sets $L(\vec{x})$ and $R(\vec{x})$, the optimal locations c_1 and c_2 are the median of each set. So, for an arbitrary location profile $\vec{x}$, we can compute the optimal solution by selecting the locations that minimize the social cost over the (n-1) choices for $L(\vec{x})$ and $R(\vec{x})$.

Unfortunately, the mechanism that selects the optimal solution is not strategyproof. The reason is that the sets in the optimal solution are susceptible to minor changes, and the agents can benefit from that. In order to design a strategyproof mechanism we need to extract $L(\vec{x})$ and $R(\vec{x})$ in a strategyproof way. Since both sets are non-empty we are sure that the leftmost agent belongs to $L(\vec{x})$ and the rightmost agent to $R(\vec{x})$. So the mechanism that places the facilities to the leftmost and the rightmost agent is strategyproof and achieves $(n-2)$-approximation ratio. As we will see bellow this is the only deterministic anonymous strategyproof mechanism with bounded approximation for the 2-Facility Location game.

Theorem 3.3.2. *The Two Extremes mechanism that places the facilities at the leftmost $lt(\vec{x})$ and the rightmost $rt(\vec{x})$ is a $(n-2)$-strategyproof mechanism for social cost.*

Proof. To show the approximation ratio consider the location profile with n agents, where one agent is located at 0, $n-2$ agents are located at $\epsilon > 0$ (arbitrarily close to 0), and one agent at 1. The optimal solution places one facility at ϵ and the other at 1. The social cost of the optimal solution is $SC^* = \epsilon$. The social cost of the solution outputted by the mechanism is $(n-2)\epsilon = (n-2)SC^*$, since $(n-2)$ agents have connection cost equal to ϵ.

We now show that the mechanism is also strategyproof. Let $\vec{x}$ be a location profile. Any agent reporting a location in the interval $[lt(\vec{x}), rt(\vec{x})]$ cannot change the output of the mechanism. However, if an agent reports a location $x' \in (-\infty, lt(\vec{x})) \cup (rt(\vec{x}), \infty)$ will move the facility further away from her ideal location. So, no agent can benefit by misreporting her location □

In the paper [44] Procaccia and Tennenholtz proved that any deterministic strategy proof mechmanism for 2-Facility Location game has approximation ratio of at least 1.5. This result was later improved to 2 [35], and then to $(n-1)/2$ [34] . Fotakis and Tzamos [23] proved a tight lower bound of $n-2$. From the latest result we conclude that the Two Extremes mechanism is the only deterministic, anonymous, strategy-proof mechanism with a bounded approximation ratio.

The takeaway from the previous section is that we cannot improve the linear approximation ratio for any deterministic strategyproof mechanism. So, the next question is if we can achieve better results with randomized mechanisms. Fortunately, the answer is yes. The *Proportional Mechanism* [34] is strategyproof and achieves a constant approximation ratio. The idea for this mechanism is very simple and intuitive. Place the first facility uniformly at random among all the reported locations and the second with a probability proportional to its distance from the first. However, is not that simple to prove that the mechanism is indeed strategyproof with constant approximation ratio.

Definition 3.3.1 (Proportional Mechanism). *Given a location profile $\vec{x} = (x_1, ...x_n)$, the location of the two facilities are decided by the following random process:*

- ***Round 1**: Choose agent i uniformly at random from N. The first facility c_1 is placed at x_i*
- ***Round 2**: Let $d_j = d(c_1, x_j)$ be the distance from agent j to the first facility. Choose agent j with probability $\frac{d_j}{\sum_{k \in N} d_k}$. The second facility is placed at x_j.*

Theorem 3.3.3. *The proportional mechanism for the two-Facility Location game is strategyproof.*

Proof. Let $cost_k(x_i, M(\vec{x}))$ denote the expected cost of agent i when the mechanism places the first facility facility on x_k. The agent that has a facility at her location experiences after the first round zero cost, so $cost_k(x_k, M(\vec{x})) = 0$. Since the first facility is selected uniformly at random, we have that for any agent i her total cost is:

$$cost(x_i, M(\vec{x})) = \frac{1}{n} \sum_{k=1}^{n} cost_k(x_i, M(\vec{x})) = \frac{1}{n} \sum_{k \neq i} cost_k(x_i, M(\vec{x}))$$

Let $\vec{x}\,' = (\vec{x}_i, x'_i)$ be the location profile after the deviation of agent i. We need to show that she cannot decrease her cost by misreporting her location. We need to show that for all $k \neq i$:

$$cost(x_i, M(x)) < cost(x_i, M(\vec{x}\,'))$$

Fix the first facility on x_k. The expected cost of agent i to the second facility, conditional on the first facility is at x_k, is:

$$cost(c_2, x_i) = \sum_{j=1}^{n} Pr[c_2 = x_j] \cdot d(x_i, x_j) = \sum_{j=1}^{n} \frac{d_j}{\sum_{k=1}^{n} d_k} d(x_i, x_j) = \frac{\sum_{j=1}^{n} d_j \cdot d(x_i, x_j)}{\sum_{j=1}^{n} d_j}$$

Then we have that the cost of agent i is the minimum between the distance to x_k and the expected distance to the second facility:

$$\begin{aligned} cost_k(x_i, M(\vec{x})) &= \min\left\{ d_i, \frac{\sum_{j=1}^{n} d_j \cdot d(x_i, x_j)}{\sum_{j=1}^{n} d_j} \right\} \\ &= \frac{\sum_{j=1}^{n} d_j \min\{d_i, d(x_i, x_j)\}}{\sum_{j=1}^{n} d_j} \\ &= \frac{\sum_{j \neq i} d_j \min\{d_i, d(x_i, x_j)\}}{\sum_{j=1}^{n} d_j} \end{aligned}$$

Let $d'_i = d(c_1, x'_i)$. The cost of agent i if she misreports is:

$$cost_k(x_i, M(\vec{x}\,')) = \frac{\sum_{j \neq i} d_j \min\{d_i, d(x_i, x_j)\}}{\sum_{j=1}^{n} d_j + (d'_i - d_i)} + \frac{d'_i \min\{d_i, d(x_i, x'_i)\}}{\sum_{j=1}^{n} d_j + (d'_i - d_i)}$$

We can get the following relation between $cost_k(x_i, M(\vec{x}))$ and $cost_k(x_i, M(\vec{x}\,'))$:

$$cost_k(x_i, M(\vec{x}\,')) = \frac{cost_k(x_i, M(\vec{x})) \sum_{j=1}^{n} d_j}{\sum_{j=1}^{n} d_j + (d'_i - d_i)} + \frac{d'_i \min\{d_i, d(x_i, x'_i)\}}{\sum_{j=1}^{n} d_j + (d'_i - d_i)} \tag{3.1}$$

We need to distinguish two cases:

(i) $d'_i \leq d_i$: we have that $cost_k(x_i, M(\vec{x})) < cost_k(x_i, M(\vec{x}\,'))$ because $\frac{\sum_{j=1}^{n} d_j}{\sum_{j=1}^{n} d_j + (d'_i - d_i)} > 1$ and the second term in the previous equality is non-negative.

(ii) $d'_i > d_i$: This case is not that simple. If we subtract $cost_k(x_i, M(\vec{x}))$ from (3.1), we have:

$$cost_k(x_i, M(\vec{x}\,')) - cost_k(x_i, M(\vec{x})) = \frac{-(d'_i - d_i) cost_k(x_i, M(\vec{x}))}{\sum_{j=1}^{n} d_j + (d'_i - d_i)} + \frac{d'_i \min\{d_i, d(x_i, x'_i)\}}{\sum_{j=1}^{n} d_j + (d'_i - d_i)}$$

So it suffice to show that:

$$d'_i \min\{d_i, d(x_i, x'_i)\} - (d'_i - d_i) cost_k(x_i, M(\vec{x})) \geq 0 \tag{3.2}$$

We need to show that (3.2) holds for both cases:

(a) If $\min\{d_i, d(x_i, x'_i)\} = d_i$. We have that $d_i \geq cost(x_i, M(\vec{x}))$ because agent i always selects the facility closest to her location which cannot be further than the facility placed ad x_k. We also have $d'_i \geq d'_i - d_i$. By multiplying the previous inequalities we have that (3.2) holds.

(b) If $\min\{d_i, d(x_i, x_i')\} = d(x_i, x_i')$. We have that $d_i' = d_i \geq cost(x_i, M(\vec{x}))$. From triangle inequality we have $d(x_i, x_i') \geq d_i' - d_i$. Again by multiplying the previous inequalities we get that (3.2) holds.

□

It is very interesting that an agent cannot gain even if she knows where the first facility is placed after the first round.

Theorem 3.3.4. *The approximation ratio of the Proportional mechanism for the two Facility Location game is 4 for any metric space.*

The proportional mechanism is the best know randomized strategyproof mechanism for 2-Facility Location game. The previous know result [35] has an approximation ratio of $n/2$. A a constant upper bound is a great improvement to the previous liner bound, but there is still a big gap between the upper bound and the only know lower bound for randomized mechanisms, which is 1.045[35].

3.3.3 Locating more than two facilities

As we previously saw the problem becomes significantly more difficult when we want to locate two facilities instead of one. For the 2-Facility Location games we saw that Two Extremes is the only deterministic and it has a linear approximation ration. Therefore, we cannot expect any better results for k-facility location games, when $k \geq 3$. In fact we have a very negative result: there is no deterministic anonymous strategyproof mechanism with bounded approximation for $k \geq 3$ [23]. This holds even for $n = k + 1$ agents. We are going to focus on the intuition behind the proof and present a sketch-proof. We will refer to deterministic, anonymous, and strategyproof mechanisms with a bounded approximation ratio (in terms of n and k) as *nice mechanisms.*

The proof heavily relies at *well-separated instances*, namely instances with $k - 1$ isolated agents and two nearby agents. Let $\vec{x} = (x_1|x_2|...|x_k, x_{k+1})$ be a well separated instance. The optimal solution is to serve all the isolated agents by a different facility and the two nearby by from the remaining facility. First, we are going to show some properties of nice mechanisms.

Proposition 3.3.1. *Let M be a nice mechanism for k-Facility Location on the line. For any $(k + 1)$-location instance $\vec{x}$, $M_1(\vec{x}) \leq x_2$ and $M_k(\vec{x}) \geq x_k$*

Proposition 3.3.2. *Let M be a nice mechanism for k-Facility Location. For any $\vec{x} = (x_1|x_2|...|x_k, x_{k+1})$ well separated instance, $M_k(\vec{x}) \in [x_k, x_{k+1}]$*

The following propositions show that in well-separated instances if a mechanism places a facility on the left of the nearby agents and we "push" both agents to the right, while keeping the instance well-separated, the rightmost facility remains on the same agent. Similarly, if a mechanism places a facility on the right of the nearby agents and we "push" both agents to the left then the rightmost facility remains on the same agent.

Proposition 3.3.3. *Let M be a nice mechanism for k-Facility Location and let $\vec{x} = (x_1|x_2|...|x_k, x_{k+1})$ be a well separated instance such that $M_k(\vec{x}) = x_k$. Then for every $\vec{x}\,' = (\vec{x}_{-\{k,k+1\}}, x_k', x_{k+1}')$ well separated instance with $x_k \geq x_k'$ it holds $M_k(\vec{x}\,') = x_k'$*

Proposition 3.3.4. *Let M be a nice mechanism for k-Facility Location and let $\vec{x} = (x_1|x_2|...|x_k, x_{k+1})$ be a well separated instance such that $M_k(\vec{x}) = x_{k+1}$. Then for every $\vec{x}\,' = (\vec{x}_{-\{k,k+1\}}, x_k', x_{k+1}')$ well separated instance with $x_{k+1} \leq x_{k+1}'$ it holds $M_k(\vec{x}\,') = x_{k+1}'$*

Using the previous propositions we can show that any anonymous nice mechanism for k-facility location on $(k + 1)$-agent instances always allocates facilities to the leftmost and to the rightmost agent.

Lemma 3.3.1. *Let M be a nice mechanism for k-Facility Location with $k \geq 2$ and $n = k + 1$. Then for all instances $\vec{x} = (x_1, ..., x_{k+1})$ with $x_1 \leq ... \leq x_{k+1}$, $M_1(\vec{x}) = x_1$ and $M_k(\vec{x}) = x_{k+1}$.*

We can now show that there is no nice mechanism for k-Facility Location when $k \geq 3$.

Theorem 3.3.5. *For every $k \geq 3$, any deterministic strategyproof mechanism for k-Facility Location with $n \geq k + 1$ agents on the line has an unbounded approximation ratio.*

The previous propositions and the lemma describe how any nice mechanism should "behave". Suppose that there is a nice mechanism M with a bounded approximation for 3-Facility Location. Let $\vec{x} = (x_1 | x_2 | x_3, x_4)$ be a well-separated instance. By Lemma 3.3.1 x_1 and x_4 have a facility. And since the instance is well-separated agent x_3 is served by the facility at x_4.

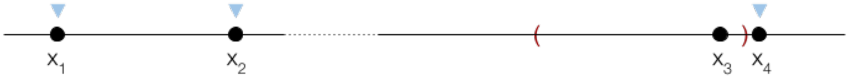


Figure 3.1: $\vec{x}$

Let us remind the definition of the image set (Definition 3.2.4). The image set of an agent i is the set of facilities the agent can obtain by varying her reported location. Any interval in the complement of an image set $I_i(\vec{x}_{-i})$ is called a hole. By Lemma 3.2.1 we have that the mechanism places a facility at the location in $I_i(\vec{x}_{-i})$ nearest to the location of agent i. Since agent x_3 is not allocated a facility in $\vec{x}$ there is a hole in the image set $I_3(\vec{x}_{-3})$ around x_3. Let l and r be the locations in $I_3(\vec{x}_{-3})$ nearest to x_3 on the left and on the right.(let the red lines represent the hole). Now consider the location profile $\vec{y} = (\vec{x}_{-3}, l + \epsilon)$. By Lemma 3.2.1 the mechanisms should place a facility at l, since l is the nearest location in the image set.

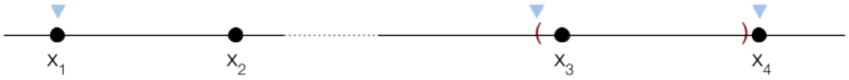


Figure 3.2: $\vec{y}$

Now consider the location profile $\vec{z} = (\vec{y}_{-4}, l) = (\vec{x}_{\{-3,4\}}, \{l, l + \epsilon\})$. The mechanism is anonymous, which means we can rearrange the indices so that $x_1 \leq x_2 \leq x_3 \leq x_4$. As a result, agents 3 and 4 switch indices in $\vec{y}$ and $\vec{z}$. We have that M is strategyproof. Since in $\vec{y}$ and in $\vec{z}$ there is an agent at $l + \epsilon$ and in $\vec{z}$ an agent is moving closer to l, M should keep a facility at l. By proposition 3.3.4 we have that x_4 has a facility because $\vec{x}$ and $\vec{z}$ are well-separated instances and the nearby agents move to the right. But this way in $\vec{z}$ the nearby agents are allocated two facilities which makes the cost arbitrarily larger than the optimal.

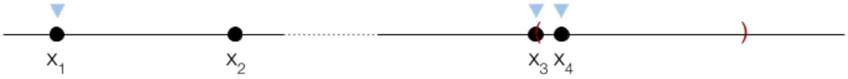


Figure 3.3: $\vec{z}$

Even if we cannot have deterministic strategyproof mechanisms with a bounded approximation ration for the k-Facility Location games with $k \geq 3$, there is a class of deterministic strategyproof mechanisms [51] for any $k \geq 2$. A percentile mechanism splits the instance into k parts, based on a predefined vector $\vec{p} \in (0,1)^k$, and allocates one facility at each part. One might think how can this class of mechanisms be practical if there is no guaranty for the outcome. However, in most "real world" applications the designer has some knowledge of the preferences of the participating agents. This allows for empirical optimization of the vector $\vec{p}$. Let us formally define the percentile mechanisms.

Definition 3.3.2 ($\vec{p}$-percentile mechanism). *The percentile mechanism is specified by a vector $\vec{p} = (p_1, ..., p_k)$ where $0 \leq p_1 \leq ... \leq p_k \leq 1$. The mechanism locates jth facility at the p_jth percentile of the reported locations.*

$$c_j = x_{i_j} \;:\; i_j = \lfloor (n-1) \cdot p_j \rfloor + 1$$

To better understand the mechanism let $\vec{x} = (x_1, ..., x_9)$ be a location profile with 9 agents. The $(0.25, 0.75)$-percentile mechanism will place the first facility at $c_1 = x_3$ since $\lfloor 8 \cdot 0.25 \rfloor + 1 = 3$ and the second facility at $c_2 = x_7$ since $\lfloor 8 \cdot 0.75 \rfloor + 1 = 7$.

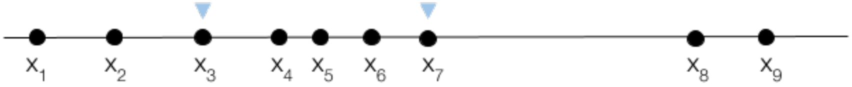


Figure 3.4: Example of $(0.25, 0.75)$-percentile mechanism for 9 agents

Lemma 3.3.2. *The percentile mechanism is group strategyproof for any $\vec{p}$.*

Proof. We are going to prove the lemma for $k = 2$ but the proof is similar for $k \geq 2$. Let $\vec{c} = (c_1, c_2)$ be the locations of the facilities when all the agents report their true preferences and $\vec{c'} = (c'_1, c'_2)$ be the locations after agents in a subset $S \subset N$ deviate from their true locations. Let $\Delta_1 = c_1 - c'_1$ and $\Delta_2 = c'_2 - c_2$. Now we need to show that at least one agent in S does not benefit from the deviation. There are four cases to consider:

(i) $\Delta_1 \geq 0$ and $\Delta_2 > 0$. That means that the first facility is moved to the left and the second to the right. In order for this to happen there is one agent that has an ideal location $x_i \in (c_1, c_2)$ who reported a location either to the left of c_1 or to the right of c_2. Otherwise the facilities could not move in such way. Now the cost of agent i is:

$$\begin{aligned} cost(x_i, M(\vec{x}\,')) &= min(d(x_i, c'_1), d(x_i, c'_2)) \\ &\geq min(d(x_i, c_1), d(x_i, c_2)) \\ &= cost(x_i, M(\vec{x})) \end{aligned}$$

(ii) $\Delta_1 \geq 0$ and $\Delta_2 < 0$. Now both facilities moved to the left. That means there is an agent i with $x_i > c_2$, that reported a location to the left of c_2. Her cost is:

$$cost(x_i, M(\vec{x}\,')) = x_i - c'_2 \geq x_i - c_2 = cost(x_i, M(\vec{x}))$$

(iii) $\Delta_1 < 0$ and $\Delta_2 \geq 0$. This case is completely symmetric to the previous case.

(iv) $\Delta_1 < 0$ and $\Delta_2 < 0$. The first facility is moved to the right and the second to the left. As in the second case there is an agent to the right of c_2 that reported a location to the left. She cannot benefit from this deviation.

Note that if $\Delta_1 = 0$ and $\Delta_2 = 0$ represents the case where neither facility moves and no agent benefits from the deviation. So without loss of generality we can assume that at least one of Δ_1 or Δ_2 is non-negative.

□

The main idea behind the proof is that the mechanism selects where to open a facility without “looking” at the instance. Like in the example above in any instance (with 9 agents) the $(0.25, 0.75)$-percentile mechanism will place the facilities at the 3nd and 7th agent respectively without looking at their actual locations. In order for an agent to change the outcome of the mechanism she would have to report a location x' in a different part than her true preference. Like in the setting with one facility (and the median location) this deviation would move the facility further away making it a non profitable deviation.

The Two Extremes mechanism mention above belongs in the class of percentile mechanism with a vector $\vec{p} = (0, 1)$. It is also the only mechanism within the family of percentile mechanisms that has a bounded approximation ratio. For any $\vec{p} \neq (0, 1)$ we can create an instance with arbitrarily small cost but since the mechanisms does not “look” at the instance the solution will have arbitrarily large cost. Consider a location profile $\vec{x} = (0, \epsilon, 1)$ where $\epsilon > 0$ but arbitrarily close to 0. The $(0, 0.6)$- percentile mechanism will allocate the facilities to the first two agents. It is easy to see that the cost of the optimal solution is ϵ but the cost of the solution of the mechanism is $1 - \epsilon$. Thus, the approximation ratio is unbounded.

Randomized mechanisms

Again we may get better results if we consider randomized mechanism for k-Facility Location on the line. Unfortunately, a simple extensions of the proportional mechanism is not strategyproof even when $k = 3$. The first two facilities are allocated like in the Proportional mechanism and the third is placed on a location of an agent with probability proportional to her minimal distance to the first two facilities.

Consider this counter-example: there exist n_0 agents at 0, n_1 agents at location 1, n_2 agents at location $1+x$ and 1 agent at location $1+x+y$. Here n_0 is sufficiently large such that we can assume the first facility to be always located at 0. In this configuration, let $y = 100$, $x = 10^{100}$, $n_1 = 50$ and $n_2 = 4$. An agent at location 1 may have the incentive to misreport to location 1+x.

If we take a closer look at the instance, we can see that the distance between locations 0 and 1 is 1, and between locations $1 + x$ and $1 + x + y$ is y. By selecting $x = 10^{100}$, the distance between locations 1 and $1 + x$ is huge compared to the other two distances. By construction, the mechanism places the first facility at 0. Since the second facility is placed with a probability proportional to the distance to the first facility, the agent’s deviation to $1 + x$ increases the probability the second facility is placed at $1 + x$. But that also means that the third facility has an increased probability of being placed at her true location, making that mechanism manipulable.

Due to the negative results in the general setting new approaches were proposed. Imposing mechanisms was proposed by Nissim, Smorodinsky, and Tennenholtz [42], namely mechanisms able to restrict how agents exploit their outcome. Fotakis and Tzamos [25] showed that the winner-imposing version of the Proportional mechanism is strategyproof for the k-Facility Location game, and achieves an approximation ratio of at most $4k$. In this setting any agent that “wins” a facility at her reported location must connect to it, even if there is a location closer to her ideal location. The mechanism runs in k rounds and allocates one facility at each round. For each $\ell = 1, ..., k$ let C_ℓ denote the facility set after the ℓth round.

Definition 3.3.3 (Winner Imposing Proportional Mechanism). *Given a location profile $\vec{x} = (x_1, ...x_n)$, the mechanism runs the following random process:*

- ***Round 1**: Choose agent i uniformly at random from N. The first facility is placed at x_i. Let $C_1 = \{x_i\}$*

- ***Round** ℓ, $\ell = 2, ..., k$: Let $d_\ell = d(x_\ell, C_{\ell-1})$ be the distance from agent ℓ to the closest facility. Choose agent ℓ with probability $\frac{d_\ell}{\sum_{k \in N} d_k}$. The ℓth facility is placed at x_ℓ, and agent x_ℓ connects to it. Let $C_\ell = C_{\ell-1} \cup \{x_\ell\}$*

3.4 Facility Location on general metrics

In the section, we are going to present the most interesting results of the Facility Location games in more general metric spaces than the line. As we saw in the previous section, the problem becomes much more difficult as the number of facilities increases. In general metric areas, however, even the placement of one facility is not trivial.

3.4.1 Tree Metrics

The first metric space that we are going to focus on is the tree, because it has similar properties to the line. For the problem of locating one facility on a tree, we can obtain the same result as in the line.

Theorem 3.4.1. *The mechanism that selects the median location of the reported instance is strategyproof and optimal for the social cost objective.*

Proof. Using the same arguments as in the line we can easily show that the median location is the optimal solution. The median on the tree is the location that when is viewed as the root all the subtrees have at most half of the nodes. Suppose the facility is placed at a location x that is on subtree T_1. Then, the connection cost for all agents on T_1 (at most half) has decreased by $d(x, med(\vec{x}))$, but the connection cost for the rest of the agents (at least half) has increased by $d(x, med(\vec{x}))$. This makes the social cost of x worst than the median location.

The median location is also strategyproof since the only way that an agent can change the median location is by reporting a location in a different subtree of the median. But this will only move the facility further away from her true location. □

It gets a lot more complicated even if we consider 2-Facility Location games on tree metric. For the next theorem we consider the star metric space. It consist of 3 half-lines $[0, \infty)$ with a common origin point O. We can see it as 3 long branches starting from O. So, we refer to this metric as S_3, and to the branches as b_1,b_2 and b_3. A location (x, b_l) in S_3 is determined by the distance $x \geq 0$ from the center and the corresponding branch. The distance between any two locations (x, b_l) and $(x', b_{l'})$ in $S3$ is $|x - x'|$, if $l = l'$ and $x + x'$ otherwise.

Theorem 3.4.2. *[23] Any deterministic strategy proof mechanism for 2-Facility Location game with $n \geq 3$ agents in S_3 has an unbounded approximation ratio.*

The proof again relies on well-separated instances with 3 agents. The idea is that even in those instances with an “obvious” optimal solution, there is no deterministic and strategyproof way to determine where to place the facility that serves the two nearby agents.

3.4.2 Circle

We know that the tree is an acyclic graph, so the next natural question is what guarantees can we get when we have a circle. Schummer and Vohra [49] prove that any strategyproof and onto mechanism for locating one facility on a circle is dictatorial.

The circle metric space (S^1, d), where $S^1 \subset \mathbb{R}^2$ is a circle in the two dimensional Euclidean space and the distance function $d(x, y)$ for $x, y \in S^1$ is the length of the minor arc spanned by x and y.

The approximation ratio of any dictatorial mechanism is $n - 1$. This can be verified by a simple example. Suppose $n - 1$ agents are located at x and one agent is located at y. The obvious optimal solution is to place the facility at x with social cost equal to $d(x, y)$. If the dictator is located at y then the social cost of that solution is equal to $(n - 1)d(x, y)$

Once again, we can use randomization to get a better approximation ratio. By selecting uniformly at random any location we can achieve constant approximation ratio.

Definition 3.4.1 (Random Dictator). *The mechanism that selects a location uniformly at random from the reported locations is strategyproof and has an approximation ratio of* $2 - \frac{2}{n}$.

Proof. The mechanism is strategyproof because an agent that deviates if selected loses since the location is further away from her ideal location and, if not selected, cannot change the outcome of the mechanism.

For the approximation ratio, let y be the optimal solution. Since the mechanism selects a location uniformly at random the expected social cost is:

$$\begin{aligned}
cost(\vec{x}, M(\vec{x})) &= \frac{1}{n} \sum_{i \in N} \sum_{j \neq i} d(x_i, x_j) \\
&= \frac{1}{n} \sum_{i \in N} \sum_{j \neq i} d(x_i, y) + d(y, x_j) \\
&= \frac{1}{n} \sum_{i \in N} \Big((n-1) d(x_i, y) + SC^* - d(x_i, y) \Big) \\
&= \frac{1}{n} \sum_{i \in N} \Big((n-2) d(x_i, y) + SC^* \Big) \\
&= SC^* + \frac{n-2}{n} SC^*
\end{aligned}$$

□

There is also a mechanism for 2-Facilities in a circle. Since the mechanism for one facility in the circle is dictatorial, we cannot expect that to change for 2 facilities. The mechanism works as follows: it places the first facility at the location of the dictator, then it cuts the circle in half and allocates the second facility based on the maximum distance in each semi-circle to the first facility.

Definition 3.4.2 (Circle mechanism for 2 Facilities). *Given a location profile* $\vec{x} = (x_1, ...x_n)$ *the first facility is allocated at* x_1*, the location of the first agent. Let* $\hat{x}_1$ *denote the antipodal of* x_1*. There are two semi-circles formed with* x_1 *and* $\hat{x}_1$ *as endpoints, the left circle* $\mathcal{L}$ *and the right circle* $\mathcal{R}$*. Let* $\mathcal{A}$ *and* $\mathcal{B}$ *be the set of agents on* $\mathcal{L}$ *and* $\mathcal{R}$ *respectively. We assume agents at location* x_1 *and* $\hat{x}_1$ *appear only in* $\mathcal{A}$*, and thus* $\mathcal{A} \cap \mathcal{B} = \emptyset$*. Define* $d_A = max_{i \in \mathcal{A}} d(x_1, x_i)$ *and* $d_B = max_{i \in \mathcal{B}} d(x_1, x_i)$*. If* $\mathcal{B}$ *is empty then* $d_B = 0$*. The second facility is allocated as follows:*

- *If* $d_A < d_B$ *facility* c_2 *is placed on* $\mathcal{R}$ *with distance* $min\{max\{d_B, 2d_A\}, 1/2\}$ *to* c_1
- *If* $d_A \geq d_B$ *facility* c_2 *is placed on* $\mathcal{L}$ *with distance* $min\{max\{d_A, 2d_B\}, 1/2\}$ *to* c_1

Theorem 3.4.3. *[34] The circle mechanism for the* 2*-Facility Location game is strategyproof and has an approximation ratio of at most* $n - 1$

There is a simple instance for which the circle mechanism has an approximation ration equal to $n - 1$. Consider the location profile $\vec{x} = (x_1, ..., x_n)$, where $d(x_1, x_2) = d(x_1, x_3) = 0.1$ and $x_3, ...x_n$. But x_2 and x_3 are in different sides of x_1. The optimal solution is to place one facility at x_1 (or x_2) and the second facility at x_3. The cost of the optimal solution is 0.1. But the circle mechanism will place one facility at x_1 and the second facility at the left semi-circle at a distance 0.2 to the first. The cost is $(n - 1)0.1$

3.4.3 Euclidean Space

For a set of points in a two dimensional Euclidean space the geometric median is the point that minimizes the sum of the distances to the data points. The distance function is the L_2-Norm. Formally given a set of points $x_1, ..., x_n \in R^m$ the geometric median is:

$$med = \underset{y \in \mathbb{R}^2}{\text{argmin}} \left\{ \sum_{i=1}^{n} d(x_i, y) \right\}$$

However, the geometric median can only be approximated.

Definition 3.4.3. *In the Euclidean metric space with $X = \mathbb{R}^m$, a mechanism f is called a generalized coordinate-wise median voting scheme with k constant points if there exists a coordinate system and points $a_1, ..., a_k \in (\mathbb{R} \cup \{-\infty, \infty\})^m$ so that for every profile $\vec{x} \in \mathbb{R}^m$ and every $j = 1, 2, ..., m$:*

$$M^j(\vec{x}) := \mathit{med}(x_1^j, x_2^j, ..., x_n^j, a_1^j, ..., a_k^j)$$

Where "med" denotes the median of the subsequent real numbers, and all coordinates are expressed with respect to the given coordinate system.

Lemma 3.4.1. *[43] In the Euclidean metric space with $X = \mathbb{R}^2$ and an odd number of agents, a mechanism M is Pareto optimal, anonymous and strategyproof if, and only if, it is a coordinate-wise median scheme with 0 constant points.*

The coordinate-wise median is strategyproof because it handles each coordinate separately. Using the same ideas as in the median on the line, the only way to change the median location is by reporting a location on the other side. But this will move the median further away.

Lemma 3.4.2. *[38] For $X = \mathbb{R}^m$ and the social cost objective, the coordinate-wise median mechanism has an approximation ratio of at most $\sqrt{m}$ for any number of agents n.*

3.5 Conclusion

In this chapter, we look at various settings for the k-Facility Location games. However, the goal was the same: the design of a mechanism with desirable proprieties such as strategyproofness, Pareto optimality, anonymity, and a bounded approximation ratio. Unfortunately, in most cases, this was infeasible. Since strategyproofness is a property we always want to have, we need to sacrifice one of the other two properties. We can have strategyproofness and a bounded approximation ratio on the circle metric space, but not anonymity since the mechanisms admit a dictator. On the other hand, in the case of the percentile mechanism, we have strategyproofness and anonymity but not a bounded approximation ratio. The trade-off between strategyproofness and approximability is the most interesting because it shows that those properties are incompatible. Intuitively, to achieve a bounded approximation ratio in the multi-facility setting, we need to place one facility in each optimal cluster; otherwise, we can construct an instance with an unbounded approximation ratio. But, as we saw for 3-Facility Location games with 4 agents, there is no deterministic strategyproof way to identify the optimal clusters and place the facilities even in well-separated instances.

Chapter 4

Perturbation Stable Clustering

Clustering and Facility Location games are two closely related problems. This is due to the fact that clustering, the primary optimization problem, and we know that is NP-Hard in general metric spaces. As we know very well, a problem is NP-Hard if no polynomial algorithm solves every input correctly. However, we are not interested in all instances but only those that appear in the real world. For clustering, it is cleverly stated in the paper by Bilu, Daniely, Linial, and Saks that "*clustering is either easy or pointless*"[11] and by Roughgarden, "*clustering is hard when it doesn't matter*"[46]. But what does this actually mean? Clustering is the task of partitioning a set of points into groups so that points in the same group are "similar" and those in different groups are "dissimilar". So in the "real world", we expect that the clusters are well-defined, meaning that "similar" points are relatively close to each other and far from any other point. This translates to communities or neighborhoods having "clear borders" in the Facility Location game, making it easy to identify them and locate the facilities. In the following section, we will refer to these instances as "stable instances" and explore all the interesting properties of stability.

4.1 Stable Clustering

Definition 4.1.1 (k-Clustering). *Given a set of points X and a non-negative distance function $d : X \times X \to [0, \infty)$ a clustering $\vec{C} = (C_1, C_2, ..., C_k)$ is the partitioning of the input points into k non-empty sets that minimizes an objective function:*

1. ***The k-median objective**: minimize the sum of distances from points to their centers*

$$\sum_{i=1}^{k} \left(\sum_{x \in C_i} d(c_i, x) \right)$$

2. ***The k-center objective**: minimize the maximum distance of a point to its corresponding center*

$$\max_{x_i \in C_i} d(x_i, c_i)$$

The k-median objective is equivalent to the social cost and the k-center to the maximum cost in the Facility Location game.

We are moving to the formal definition of "well-defined" or stable instances in clustering. Intuitively, stability means all points in the same cluster in the optimal solution are close together and that the clusters are well separated. It also means that the optimal solution is not affected by small changes in the input. We can perturb an instance to create nearby instances by scaling down the distance between any pair of points by a factor of at most γ, with $\gamma \geq 1$.

Definition 4.1.2 (γ-perturbation). *For $\gamma \geq 1$, a γ-perturbation of a metric space (X, d) is another space (X, d') on the same point set such that:*

$$d'(x, y) \in \left[\frac{1}{\gamma} d(x, y), d(x, y) \right]$$

Definition 4.1.3 (γ-perturbation stability). [1] *A clustering instance (X, d) is γ-perturbation stable to a given objective function Φ if there is a k-clustering $\vec{C} = C_1, ..., C_k$ such that, for every γ-perturbation (X, d'), $\vec{C}$ remains the unique optimal k-clustering.*

There is also a weaker notion of stability called γ-*metric perturbation stability*. In the definition of stability the perturbed space d' does not need to be metric (i.e. d' does not have to satisfy triangle inequality). In order for an instance to be γ-stable it needs to admit the same optimal solution in every γ-perturbation. For γ-metric stability we only require that the optimal solution remains the same for every γ-metric perturbation a subset of γ-perturbations. Thus, the class of γ-metric stable instances includes the class of γ-stable instances. The reason that we say γ-metric perturbation stability is a weaker notion of stability is because we relax the conditions for stability to more natural ones, therefore allowing more instances in that class.

Definition 4.1.4 (γ-metric perturbation stability). *A clustering instance (X, d) is γ-metric perturbation stable to a given objective function Φ if there is a k-clustering $\vec{C} = C_1, ..., C_k$ such that, for every γ-metric perturbation (X, d'), $\vec{C}$ remains the unique optimal k-clustering.*

4.2 Properties of Perturbation Stable Instances

One of the most important properties for γ-stable instances is the *center proximity* property which implies that in the optimal solution every point is closer to its own center than to any other center.

Definition 4.2.1 (γ-Center Proximity). *Let $\gamma \geq 1$, any γ-stable instance with unique optimal clustering $C_1, ..., C_k$ and optimal centers $c_1, ..., c_k$ satisfies γ-center proximity. For all distinct cluster pairs C_i and C_j and any point $x_i \in C_i$:*

$$d(x_i, c_j) > \gamma d(x_i, c_i)$$

Proof. Let C_i and C_j be any two clusters in the optimal solution and $x_i \in C_i$. We consider a γ-perturbation of the original instance in which all distances among points in C_i and points in C_j are scaled down by a factor γ. Since this is a valid perturbation, the optimal clustering remains the same. Furthermore, since the distances within the clusters C_i and C_j are the same, c_i remains the optimal cluster center for C_i and c_j for C_j. In the perturbed instance x_i still belongs in C_i so $d'(x_i, c_j) > d'(x_i, c_i)$. From the way the perturbed instance is created we have that $d'(x_i, c_j) = \frac{1}{\gamma} d(x_i, c_j)$ and that $d'(x_i, c_i) = d(x_i, c_i)$. Therefore, $d(x_i, c_j) > \gamma d(x_i, c_i)$ □

An immediate consequence of γ-center proximity is that any point is $\gamma - 1$ times closer to its own center than to any other point from a different cluster.

Lemma 4.2.1 (γ-Weak Center Proximity). *Let $\gamma \geq 2$, any γ-stable instance with unique optimal clustering $C_1, ..., C_k$ and optimal centers $c_1, ..., c_k$ satisfies $\gamma-$weak center proximity. For any point $x \in C_i$ and $y \notin C_i$:*

$$d(x, y) > (\gamma - 1) d(x, c_i)$$

Proof. Let x be any point in C_i and y any point in C_j. We need to distinguish two cases:

(i) $d(y, c_j) \geq d(x, c_i)$: By triangle inequality we have that $d(y, c_i) \leq d(y, x) + d(x, c_i)$. Since the instance is stable we have $d(y, c_i) > \gamma d(y, c_j) > \gamma d(x, c_i)$ from γ-center proximity property. Hence, we get that $d(x, y) \geq d(y, c_i) - d(x, c_i) > \gamma d(x, c_i) - d(x, c_i) = (\gamma - 1) d(x, c_i)$

(ii) $d(y, c_j) < d(x, c_i)$: Again by triangle inequality we have that $d(x, c_j) \leq d(x, y) + d(y, c_j)$. From γ-center proximity property we have $d(x, c_j) > \gamma d(x, c_i)$. Hence, we get that $d(x, y) \geq d(x, c_j) - d(y, c_j) > \gamma d(x, c_i) - d(x, c_i) = (\gamma - 1) d(x, c_i)$

[1] This definition does not state that the optimal cluster centers must remain the same.

□

We can use the previous properties and show that the distance between any two points from a different cluster cannot be arbitrarily small.

Lemma 4.2.2 (Cluster Separation Property). *Let $C_1, ..., C_k$ be the optimal clustering of a γ-stable instance with $\gamma \geq 2$. Let $x_i, x_i' \in C_i$ and $x_j \in C_j$ $(i \neq j)$ then:*

$$d(x_i, x_j) > \frac{(\gamma-1)^2}{2\gamma} d(x_i, x_i')$$

Proof. Let $\vec{C}$ be the optimal clustering of $\vec{x}$. Let $x_i, x_i' \in C_i$ have center c_i and $x_j \in C_j$ $(i \neq j)$ have center c_j. We have that:

$$\begin{aligned} \gamma d(x_i, c_i) < d(x_i, c_j) < d(x_i, c_i) + d(c_i, c_j) \implies \\ (\gamma - 1) d(x_i, c_i) < d(c_i, c_j) \end{aligned} \tag{4.1}$$

Where the first inequality follows from the definition of the γ-center proximity and the second from the triangle inequality. We also have that:

$$\begin{aligned} d(c_i, c_j) &< d(c_i, x_i) + d(x_i, x_j) + d(x_j, c_j) \\ &< \frac{2}{\gamma - 1} d(x_i, x_j) + d(x_i, x_j) \\ &< \frac{\gamma + 1}{\gamma - 1} d(x_i, x_j) \end{aligned} \tag{4.2}$$

Where the first inequality follows from the triangle inequality and the second from the definition of the γ-weak center proximity.

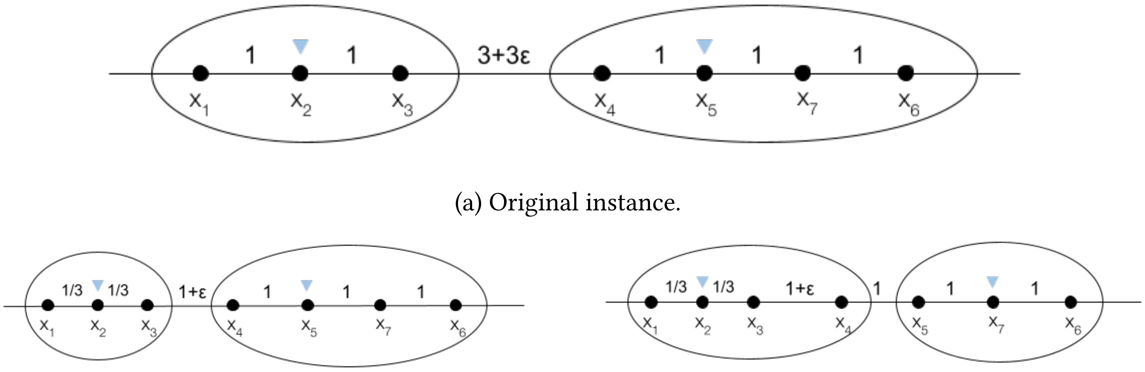
Finally we have that:

$$\begin{aligned} d(x_i, x_i') &\leq d(x_i, c_i) + d(c_i, x_i') \\ &\overset{(4.1)}{<} \frac{1}{\gamma - 1} d(x_i, x_j) + \frac{1}{\gamma - 1} d(c_i, c_j) \\ &\overset{(4.2)}{<} \frac{1}{\gamma - 1} d(x_i, x_j) + \frac{\gamma + 1}{(\gamma - 1)^2} d(x_i, x_j) \\ &= \frac{2\gamma}{(\gamma - 1)^2} d(x_i, x_j) \end{aligned}$$

□

It is very important to note that the three properties presented above are necessary for stability but not sufficient. We can easily construct a non-stable instance in which all the properties hold. In the instance in figure 4.1a we can easily verify that all the inequalities are satisfied for $\gamma = 3$ for any $\epsilon > 0$. However, if we perturb the instance by scaling down the distances between agents $x_1 - x_2$, $x_2 - x_3$ and $x_3 - x_4$ by a factor γ, the optimal clustering does not remain the same. On the left (figure 4.1b) we have the same clustering as in the original instance. The cost of the left clustering is $\frac{2}{3} + 4$. But if we place the centers as it is shown in the right (figure 4.1c) the cost is $4 + \epsilon$. If the original instance was stable the optimal solution should have remained the same for all perturbations. Thus, the instance is not 3-stable.

Never the less, the center-proximity is still a very important property. We can define the notion of γ-center stability. A instance is γ-center stable if in the optimal solution the center proximity holds for every point. This is a weak notion of stability since it allows more instances in the class. Figure 4.2 illustrates all the different classes of stability.



(a) Original instance.

(b) Perturbed instance with the same clustering.

(c) Perturbed instance with an alternative clustering.

Figure 4.1: Example of non stable instance that satisfies all the properties.

Definition 4.2.2 (γ-center stability). *A clustering instance (X, d), with optimal solution $\vec{C} = C_1, ..., C_k$, is γ-center stable to a given objective function Φ, if any $x_i \in C_i$ satisfies:*

$$d(x_i, c_j) > \gamma d(x_i, c_i)$$

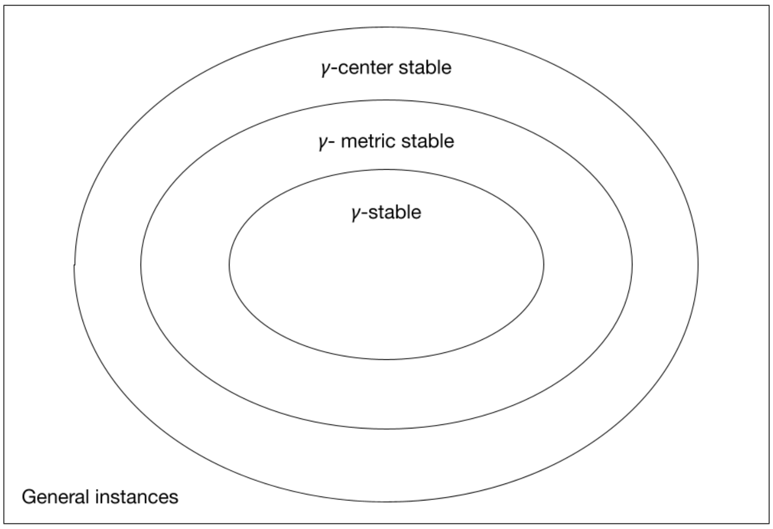


Figure 4.2: Classes of stable instances

4.3 Algorithms for Perturbation Stable Instances

The greater the factor of stability, the more structured the points are. For example, in every 2-stable instance from weak center proximity, we get that every point is closer to its optimal center than any other point from a different cluster. It is also very interesting that for a stability factor of $\gamma \geq 2 + \sqrt{3} \approx 3.7$ we obtain from the separation property that all points are closer to all other points in their own cluster than to points in any other cluster. The next question is how we can use the properties derived from stability in order to achieve better results. The first idea is to see how existing algorithms perform on stable instances. *Single-link* clustering is one of the most commonly used methods for clustering that uses the distances between the points. It starts with n clusters of size 1 (the data points), and at each step it merges the two clusters that have the minimum distance until there are k clusters. One may think that if the instance is stable enough this algorithm could return the optimal solution, since at each step it connects the two "closest" clusters. However, even in simple stable instances single linkage fails to find the optimal clustering. Consider an instance with 4 distinct locations $\vec{x} = (x_1, x_2, x_3, x_4)$. Our goal is to place 3 facilities. There is one agent located at x_1 and $M >> 1$ agents located at every other location.

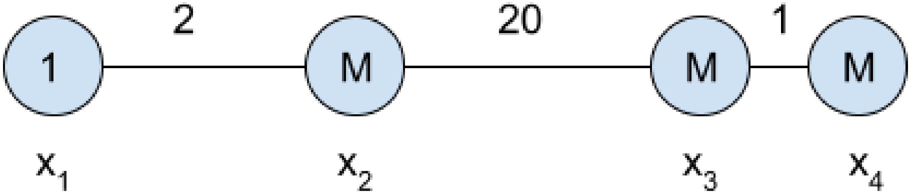


Figure 4.3: An example that single linkage fails to find the optimal solution

The optimal solution is to place the centers at x_2, x_3 and x_4 and connect x_1 to the center at x_2. The total cost of the optimal solution is 2. But the single-linkage clustering will merge the the clusters at x_3 and x_4 and places one center between them making total cost of the solution equal to M, which is arbitrarily larger than the optimal cost. Note that by construction the instance is γ-stable for arbitrarily large γ, for a properly selected value of M.

But a variation of the previous algorithm can find the optimal solution for a sufficient γ stability. It starts with n clusters of size 1. Instead of stopping at k clusters it continues until one cluster remains. Then it finds the best k clusters of the resulting tree T using dynamic programming.

Lemma 4.3.1. *Single linkage can find the optimal solution if and only if each cluster forms a subtree of the spanning tree.*

Proof. The single linkage algorithm outputs the best k-clustering among all the $\binom{n-1}{k-1}$ different choices. Every different clustering can be obtained by removing $k-1$ edges from the spanning tree. This way all the k clusters are different connected components of T, If some optimal cluster is not a connected subgraph of T the algorithm cannot find it. □

This can be illustrated with a simple example.

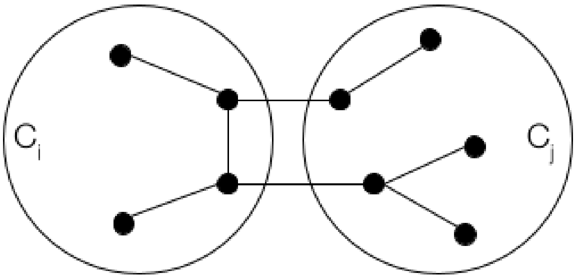


Figure 4.4: The optimal cluster C_j do not form a subtree of T

The function used to determine the distance between two clusters, is known as the linkage function. For different functions we get different stability factors for which we can find the optimal solution.

The first linkage function was proposed by Awasthi [7]. At each step it merges the two cluster C, C' minimizing $d_{min}(C, C')$. The minimum distance between any two subsets $A, B \subset X$ is defined as:

$$d_{min}(A, B) = min\{d(a, b)|a \in A, b \in B\}$$

They prove that the single-linkage algorithm can return the optimal solution for a stability factor 3 when centers must be data points and for stability factor $2 + \sqrt{3}$ for general metrics. With tighter analysis Angelidakis showed [4] that the instance only needs to be 2-metric stable, when the centers must be data points.

Lemma 4.3.2. *Single-linkage with d_{min} as the linkage function finds the optimal solution in every 2-metric stable instance.*

Proof. By Lemma 4.3.1 if every optimal cluster C_i is a connected subgraph on the spanning tree T then single-linkage returns the optimal solution. It is enough to show that the unique path that connects any point $a \in C_i$ to its optimal center c_i on T does not leave the cluster C_i. Suppose that b is the next point on the path. Consider the step at which the single-linkage adds the edge (a, b) to the spanning tree. At that step, a and c_i belonged in different clusters, otherwise the edge (a, b) would form a cycle. We have that $d(a, b) < d(a, c_i)$ or else the edge (a, c_i) would have been added instead of (a, b). By Lemma 4.2.1 the previous inequality implies that b belongs to C_i. By induction all the points on the path between a and c_i belong to C_i. □

The second linkage function was proposed by Balcan and Linang [9]. The single linkage algorithm merges the clusters with the minimum closure distance. The closure distance between two clusters C, C' is the radius of the minimum ball that covers all the points in $C \cup C'$ and has some margin from points outside of the ball. A ball around p with radius r is set of all points that have distance from p smaller than r: $\mathbb{B}(p, r) := \{q : d(p, q) < r\}$

Definition 4.3.1 (Closure Distance). *The closure distance $d_S(C, C')$ between two disjoint nonempty subsets $C, C' \subset X$ is the minimum $d > 0$ such that there is a point $c \in C \cup C'$ satisfying the following requirements:*

(i) Coverage: the ball $\mathbb{B}(c, d)$ covers C and C', that is, $C \cup C' \subseteq \mathbb{B}(c, d)$

(ii) margin: points inside $\mathbb{B}(c, d)$ are closer to the center c than to points outside, that is, $\forall p \in \mathbb{B}(c, d), q \notin \mathbb{B}(c, d)$, we have $d(c, p) < d(c, q)$

Lemma 4.3.3. *Single-linkage with d_S as the linkage function finds the optimal solution in every $1+\sqrt{2}$-center stable instance.*

4.4 Conclusion

Using the techniques from *Beyond Worst Case Analysis* helps us avoid the hardness results in the general case. For clustering, we are able to design exact algorithms for a problem that is NP-hard to even approximate. However, the notion of perturbation stability does not completely solves our problem. There is an NP-hardness lower bound for the stability factor. For any $\epsilon > 0$, finding the optimal k-center clustering for $(2 - \epsilon)$-perturbation stable instances is NP-hard, unless $NP = PR$[8]. We also have that is NP-hard to find the optimal k-median clustering for $(2 - \epsilon)$-center stable instances [10]. Since the single-linkage algorithm only relies on the center proximity, which means the previous algorithm is tight. However, the class of γ-stable instances is a subclass of γ-center stable instances, which means that the lower bound does not transfer immediately. To show a lower bound for the class of γ-stable instances or to find a better algorithm, we need a different approach, specifically to include valid perturbations.

One downside of this approach is that there is no algorithm that can find or even estimate the γ for which an instance is γ-stable. However, the notion of γ-perturbation stability is still very important because it captures all the instances that have a “clearly optimal” solution. That means we are clustering with the correct objective, which is not susceptible to minor perturbations.

Chapter 5

Facility Location on Perturbation Stable Instances

In the previous section we saw that when we focus on the average instances of a problem, namely the ”real-world” instances, we can avoid the hardness results and design better algorithms. Since clustering and facility location games are closely related problems, in this section we are going to see how perturbation stability in k-facility location games affects the design of mechanisms. Can we design strategyproof mechanisms with a bounded approximation ratio using the properties of stability? We have seen that in the simple setting where we want to locate 2 facilities in the line any deterministic mechanism has $(n-1)$-approximation ratio because the agents have the power to change the clustering by misreporting their preferred location. So naturally it is very interesting to investigate the power any agent has in changing the structure of the clustering when the instance is stable and the clusters are well defined.

5.1 Facility Location in Trees

In this section, the agents are located in a Tree. With out loss of generality, we can view the tree with the location of agent x_1 as the root. We first need to adapt the definition of perturbation stability for the Facility Location games.

Definition 5.1.1 (γ-Perturbation and γ-Stability). *Let N be a set of n agents located on a tree (T, d), where d is a metric distance function. Let $\vec{x} = (x_1, ..., x_n)$ be a location profile. Let $\vec{z} = (z_1, ..., z_m)$ denote the intersections on the tree, we will refer to them as phantom agents. A location profile $\vec{x}\,' = (x'_1, ..., x'_n) \in (T, d')$ is a γ-perturbation of $\vec{x}$, for some γ, if for every pair of consecutive agents x and y (agents x and y can be any of $x_i - z_j$,$x_i - x_j$ or $z_i - z_j$ pairs) it holds:*

$$d'(x, y) \in \left[\frac{1}{\gamma} d(x, y), d(x, y)\right]$$

A k-Facility Location instance $\vec{x}$, with optimal clustering $\vec{C} = (c_1, ..., c_k)$, is γ-(perturbation) stable, if for every γ-perturbation $\vec{x}\,'$ of $\vec{x}$ $\vec{C}$ remains the unique optimal k-clustering.

In order to properly define it we are going to assign phantom agents $\vec{z} = (z_1, ...z_m)$ on every intersection of the tree. A γ-perturbation $\vec{x}\,'$ of an instance $\vec{x}$ can be obtained by scaling down any subset of pairs of consecutive agents, including the phantom agents, by a factor of at most γ. We can see that every valid metric-perturbation (Definition 4.1.4) of the instance $\vec{x}$ (only agents' locations) can be performed by the process described above. Note that the distance between any pair of actual agents x_i and x_j is scaled down by a factor of at most γ because the length of every pair of consecutive agents in the unique path that connects them is scaled down by a factor of at most γ. Moreover, the perturbed space is metric because the distance between x_i and x_j is the length of the unique shortest path between x_i and x_j. Therefore, this class of γ-stable instances includes the class of γ-metric stable instances. This notion of γ-perturbation captures all the perturbations that can be depicted on the same tree.

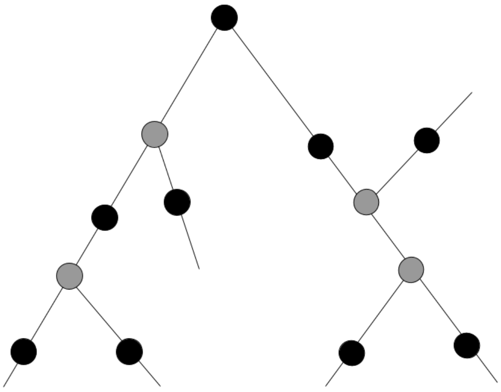


Figure 5.1: An example of a location profile $\vec{x}$ (black points) with phantom agents $\vec{z}$ on the intersections (gray points)

In the previous section, we showed that in general metric spaces in γ-stable instances, with $\gamma \geq 2$, every cluster in the optimal solution forms a subtree in the minimum spanning tree. Similarly, we can show that this property holds when the underline metric space is a tree T.

Observation 5.1.1. *Let $\vec{x}$ be a γ-stable, with $\gamma \geq 2$, then in the optimal clustering $\vec{C}$ every cluster C_i forms a subtree.*

Proof. For contradiction suppose there exist a cluster C_j that does not form a subtree, like in Figure 5.2. Since the instance is stable we have that $d(x_{j_2}, x_i) > (\gamma - 1)d(x_{j_2}, c_j)$. We also have that the distance between any two points is equal to the length of the unique path that connects those points, thus $d(x_{j_2}, c_j) = d(x_{j2}, x_i) + d(x_i, c_j)$. Using the previous inequality we get $d(x_{j_2}, c_j) > (\gamma - 1)d(x_{j2}, c_j) + d(x_i, c_j) \implies d(x_i, c_j) < (2 - \gamma)d(x_{j_2}, c_j)$. When $\gamma \geq 2$ the distance between $d(x_i, c_j)$ is negative but every distance function is non-negative. □

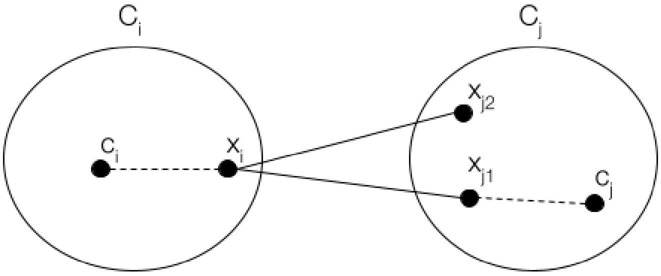


Figure 5.2: Example of a cluster that does not form a subtree.

As we mentioned before, perturbation stability implies that there is a structure in the original location profile. In other words, the optimal solution is resilient to minor changes. In the facility location setting, the agents are strategic, which means they are motivated to misreport in order to reduce their connection cost. The idea is to see how much power a strategic agent has to change the outcome of a mechanism with a single deviation now that the instance has a well-defined optimal solution. Let us point out that an agent's deviation is entirely different from a perturbation of an instance. An agent can misreport at any location in the metric space, which may result in a non-stable instance. However, after a perturbation, the distance between any two agents is at most γ times smaller than in the original instance.

5.1.1 The Optimal solution is strategyproof for $(2+\sqrt{3})$stable instances

```
Algorithm 2: OPTIMAL
Result: An allocation of $k$-facilities
Input: A $k$-Facility Location instance $\vec{x}$.
Compute the optimal clustering $(C_1, \ldots, C_k)$. Let $c_i$ be the median point of each cluster $C_i$.
if $\big(\exists i \in [k]$ with $|C_i| = 1\big)$ or $\big(\exists\, x_i, x_i' \in Ci$ and $x_j, x_j' \in C_j$ with
  $\max\{d(x_i, x_i'), d(x_j, x_j')\} \geq d(x_i, x_j)\big)$ then
    Output: "FACILITIES ARE NOT ALLOCATED".
else
    Output: The $k$-facility allocation $(c_1, \ldots, c_k)$
```

We next show that the the mechanism that returns the optimal solution is strategyproof for $(2+\sqrt{3})$-stable instances whose optimal clustering does not include singleton clusters. We need to exclude the stable instances with singleton clusters in their optimal solution from the mechanism because there is always an agent that can benefit by becoming a singleton cluster. Consider a γ-stable instance $\vec{x}$. Suppose an agent $x_i \in C_i$ reports a location x' far away from any location in $\vec{x}$ creating a different instance $\vec{x}\,'$. Since the mechanism allocates the facilities optimally x' becomes a singleton cluster. Now the mechanism has to allocate $k-1$ facilities to the remaining agents, which means two clusters from the original optimal solution merge. We can always create an instance in which cluster C_i merges with its neighbor cluster C_j which makes the new median closer to x_i. The main problem is that the new instance $\vec{x}\,'$ is also γ-stable as the original and there is no way to determine whether or not there is a deviation.

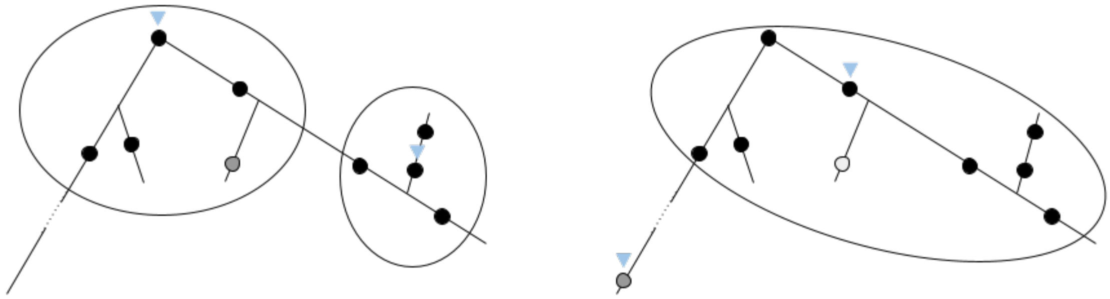


Figure 5.3: Singleton Deviation

It is also important to remember that we cannot efficiently estimate the stability factor of a given instance. This is why we rely on the properties that derive from stability, since the properties are necessary but not sufficient for stability. That also means that the mechanism serves instances that are non-stable but satisfy the cluster separation property. We can efficiently check if the cluster separation property is violated between any to clusters. Since every cluster in the optimal solution forms a subtree, we only have to check if any of the $k-1$ edges between 2 different clusters violates the cluster separation property.

Theorem 5.1.1. *The* OPTIMAL *Mechanism applied to $(2+\sqrt{3})$-stable instances of k-Facility Location without singleton cluster in their optimal clustering is strategyproof and minimizes the social cost.*

Proof. The mechanism returns the optimal solution, so we only need to prove that it is strategyproof. Let any $(2+\sqrt{3})$-stable instance with optimal clustering $\vec{C} = (C_1, ..., C_k)$. Suppose agent $x_i \in C_i$ reports a location y in order to decrease her cost. Let $\vec{y} = (\vec{x}_{-i}, y)$ be the instance after the deviation, and $\vec{Y}$ be its optimal clustering.There are three possible deviations:

1. If y is in the area of C_i, then the clustering structure would not change. But the median location for one facility on trees is strategyproof and thus she cannot benefit from such deviation.
2. If y is a singleton cluster in $\vec{Y}$, then the mechanism would not allocate any facilities, making her cost infinite times larger.
3. If y is cluster together with agents from different clusters in $\vec{C}$.

We will only focus on the third case since this is the only way she could benefit. In order for the mechanism to be strategyproof, we need to show that either the deviation is not profitable ($d(x_i, \vec{C}) < d(x_i, \vec{Y})$) or that the cluster separation property is violated and no facilities are allocated.

We have that y is clustered together with agents belonging in one or two clusters of $\vec{C}$, let that clusters be C_j and C_l. In $\vec{Y}$ the number of facilities serving agents from $C_j \cup C_l \cup \{y\}$ are at least the number of facilities serving agents from $C_j \cup C_l$ in $\vec{C}$, because both $\vec{C}$ and $\vec{Y}$ are optimal clusterings. Suppose one facility is placed to an agent in C_j. The original instance is $2+\sqrt{3}$-stable, so by the separation property we have that every agent is closer to every agent from her own cluster that to any other agent from a different cluster. This implies that no agent in C_j is served by a facility in $\vec{x} \setminus C_j$.

Case 1: *y is not allocated a facility in $\vec{Y}$*: This can happen in one of two ways:

Case 1a: y is clustered together with some agents from cluster C_j and no facility placed in C_j serves agents in $\vec{x} \setminus C_j$ in $\vec{Y}$.

Case 1b: y is clustered together with some agents from a cluster C_j and at least one of the facilities placed in C_j serve agents in $\vec{x} \setminus C_j$ in $\vec{Y}$.

Case 2: *y is allocated a facility in $\vec{Y}$*. This can happen in one of two ways:

Case 2a: y only serves agents that belong in C_j (by optimality, y must be the median location of the new cluster, which implies that either y is not in the area of C_j and only serves one agent from C_l or y is in the area of C_j and serves multiple agents).

Case 2b: In $\vec{Y}$, y serves agents that belong in both C_{j-1} and C_j.

We first consider the cases 1a and 2a, namely the cases where in $\vec{Y}$ there is a facility in C_j that only serves agents in C_j. If there is only one facility allocated in $C_j \cup \{y\}$ then both clusterings $\vec{C}$ and $\vec{Y}$ have the same structure, making x_i's deviation not profitable. So, in $\vec{Y}$ there must be two facilities in $C_j \cup \{y\}$. Suppose there is a location y such that the deviation is profitable, $d(x_i, \vec{C}) > d(x_i, \vec{Y})$. Since $\vec{Y}$ is the optimal clustering for $\vec{y}$ we have that:

$$\begin{aligned} cost(\vec{y}, \vec{C}) > cost(\vec{y}, \vec{Y}) \iff \\ cost(\vec{x}, \vec{C}) + d(y, \vec{C}) - d(x_i, \vec{C}) > cost(\vec{x}, \vec{Y}) + d(y, \vec{Y}) - d(x_i, \vec{Y}) \iff \\ d(y, \vec{C}) - d(y, \vec{Y}) > cost(\vec{x}, \vec{Y}) - cost(\vec{x}, \vec{C}) + d(x_i, \vec{C}) - d(x_i, \vec{Y}) \end{aligned}$$

Since i's deviation to y is profitable ($d(x_i, \vec{C}) - d(x_i, \vec{Y}) > 0$) we get:

$$\begin{aligned} d(y, \vec{C}) - d(y, \vec{Y}) &> cost(\vec{x}, \vec{Y}) - cost(\vec{x}, \vec{C}) \\ &= cost(C_j, \vec{Y}) - cost(C_j, \vec{C}) - cost(\vec{x} \setminus C_j, \vec{Y}) - cost(\vec{x} \setminus C_j, \vec{C}) \end{aligned} \tag{5.1}$$

We now consider a valid γ-perturbation $\vec{x}\,'$ of the original instance $\vec{x}$: We first remove from the instance all the agents from C_j and all the edges connected to them. This may break the instance in more than one connected components. By observation 5.1.1, we have that there is no cluster whose agents belong in more than one connected components. Then, we scale down by γ all the distances between consecutive agents that are in the same connected component. By stability, the clustering $\vec{C}$ remains the unique optimal clustering for $\vec{x}'$ therefore $cost(\vec{x}', \vec{C}) < cost(\vec{x}', \vec{Y})$. Since, in cases 1a and 2a, the facility allocated to an agent of $C_j \cup \{y\}$ does not serve agents in $\vec{x} \setminus C_j$ in $\vec{C}$ and $\vec{Y}$ we have:

$$\begin{aligned}
cost(\vec{x}', \vec{C}) &= cost(C_j, \vec{C}) + \frac{1}{\gamma} cost(\vec{x} \setminus C_j, \vec{C}) \\
cost(\vec{x}', \vec{Y}) &= cost(C_j, \vec{Y}) + \frac{1}{\gamma} cost(\vec{x} \setminus C_j, \vec{Y})
\end{aligned}$$

Using that $cost(\vec{x}', \vec{C}) < cost(\vec{x}', \vec{Y})$ and that for any $\gamma \geq 2$ it holds $\frac{1}{\gamma} \leq 1 - \frac{1}{\gamma}$ we get:

$$\begin{aligned}
cost(C_j, \vec{C}) - cost(C_j, \vec{Y}) &< \frac{1}{\gamma} \left(cost(\vec{x} \setminus C_j, \vec{Y}) - cost(\vec{x} \setminus C_j, \vec{C}) \right) \quad (5.2) \\
&\leq (1 - \frac{1}{\gamma}) \left(cost(\vec{x} \setminus C_j, \vec{Y}) - cost(\vec{x} \setminus C_j, \vec{C}) \right) \quad (5.3)
\end{aligned}$$

Rearranging (5.3) we get:

$$cost(\vec{x}, \vec{Y}) - cost(\vec{x}, \vec{C}) > \frac{1}{\gamma} \left(cost(\vec{x} \setminus C_j, \vec{Y}) - cost(\vec{x} \setminus C_j, \vec{C}) \right) \quad (5.4)$$

We know that there are at least two facilities serving $C_j \cup \{y\}$. Let Y_{j_1} be the cluster that contains y and some agents of C_j. In case 1a, let $x_j \in Y_{j_1}$ be the agent that has the facility. Then the decrease in the cost of y due to the additional facility in $\vec{Y}$ is at most the decrease in the cost of an agent from C_j in $\vec{Y}$. In case 2a, where agent y has a facility, by optimally of $\vec{Y}$, we have that y is the median of the new cluster. If we view c_j as the root of the tree then y has at least one agent from C_j as a child; otherwise it would not be the median location. Therefore, the decrease in the cost of y is at most the decrease in the cost of an agent from C_j in $\vec{Y}$. (The case where y is not in the area of C_j and only serves one agent is equivalent to placing the facility on the other agent and serving y from there). In both cases, we can bound from below the total decrease in the cost of C_j due to the additional facility.

$$\begin{aligned}
d(y, \vec{C}) - d(y, \vec{Y}) &\leq cost(C_j, \vec{C}) - cost(C_j, \vec{Y}) \xRightarrow{(5.2)} \\
&\leq \frac{1}{\gamma} \left(cost(\vec{x} \setminus C_j, \vec{Y}) - cost(\vec{x} \setminus C_j, \vec{C}) \right) \xRightarrow{(5.4)} \\
&< cost(\vec{x}, \vec{Y}) - cost(\vec{x}, \vec{C})
\end{aligned}$$

Which contradicts equation (5.1).

Now we consider cases 1b and 2b, namely the cases where some agents of C_j are clustered with agents of C_l. Let Y_{j_1} and Y_{j_2} denote the clusters of $\vec{Y}$ that include all the agents of C_j. Let Y_{j_1} be the cluster that has agents from C_j and C_l. Consider $x_1 \in Y_{j_1} \cap C_j$, $z \in Y_{j_1} \cap C_l$ and $x_2 \in Y_{j_2} \cap C_j$. By the cluster separation property $d(x_1, z) \geq D_{x_1}$, where D_{x_1} is the largest intra-cluster distance from x_1. Since x_1 and x_2 belong in the same cluster in $\vec{C}$ we have that $d(x_1, x_2) < D_{x_1}$. Therefore, $d(x_1, z) > d(x_1, x_2)$ which violates the cluster separation property, since an intra-cluster distance is greater than an inter-cluster distance. In this case, the mechanism will not allocate any facilities. □

5.2 Conclusion and Future Work

Remember that without making any assumptions about the instance, we cannot design strategyproof mechanisms with a bounded approximation because we cannot identify the optimal clusters in a strategyproof way. However, if we focus on perturbation stable instances, the structure helps us avoid that problem. Now we can efficiently find the optimal solution and place the facilities within each optimal cluster. We also showed that the strategic agents do not have the power to change the structure of the instance for a stability factor $\gamma \geq 2 + \sqrt{3}$ because all the inter-cluster distances are greater than the intra-cluster distances. This allows us to reduce the multi-facility location games on stable instances to a one-facility location game and use all the known results.

In the Optimal mechanism, we saw that we have to exclude the instances that contain singleton clusters because an agent can deviate to a location far away and bring a facility closer to her true location (singleton deviation). One way to overcome this problem is to limit the range of possible locations each player can declare based on their actual locations. In this case, we must also see how stable the instance must be in order for the optimal solution to be strategyproof.

Another direction is to assume a "weaker" notion of stability in the instances. In the literature, (γ, ϵ)-perturbation stability was also proposed, which states that at most ϵn total points can swap into or out of each cluster under any γ perturbation. These instances are more likely to describe "real world" instances. However, the non-stable points may be harder to handle.

Bibliography

[1] P. K. Agarwal, H. Chang, K. Munagala, E. Taylor, and E. Welzl, “Clustering under perturbation stability in near-linear time,” *CoRR*, vol. abs/2009.14358, 2020. [Online]. Available: https://arxiv.org/abs/2009.14358

[2] N. Alon, M. Feldman, A. D. Procaccia, and M. Tennenholtz, “Strategyproof approximation of the minimax on networks,” *Mathematics of Operations Research*, vol. 35, no. 3, pp. 513–526, 2010. [Online]. Available: http://www.jstor.org/stable/40800845

[3] H.-C. An and O. Svensson, *Recent Developments in Approximation Algorithms for Facility Location and Clustering Problems.* Singapore: Springer Singapore, 2017, pp. 1–19. [Online]. Available: https://doi.org/10.1007/978-981-10-6147-9_1

[4] H. Angelidakis, K. Makarychev, and Y. Makarychev, “Algorithms for stable and perturbation-resilient problems,” *Proceedings of the Annual ACM Symposium on Theory of Computing*, vol. Part F128415, pp. 438–451, jun 2017.

[5] A. Archer and R. Kleinberg, “Truthful germs are contagious: A local-to-global characterization of truthfulness,” *Games and Economic Behavior*, vol. 86, pp. 340–366, jul 2014.

[6] V. Auletta, R. De Prisco, P. Penna, and G. Persiano, “The power of verification for one-parameter agents,” *Journal of Computer and System Sciences*, vol. 75, no. 3, pp. 190–211, may 2009.

[7] P. Awasthi, A. Blum, and O. Sheffet, “Center-based clustering under perturbation stability,” *Information Processing Letters*, vol. 112, no. 1-2, pp. 49–54, jan 2012.

[8] M. F. Balcan, N. Haghtalab, and C. White, “k-center Clustering under Perturbation Resilience,” *ACM Transactions on Algorithms*, vol. 16, no. 2, may 2015. [Online]. Available: https://arxiv.org/abs/1505.03924v4

[9] M. F. Balcan and Y. Liang, “Clustering under Perturbation Resilience,” *SIAM Journal on Computing*, vol. 45, no. 1, pp. 102–155, dec 2011. [Online]. Available: https://arxiv.org/abs/1112.0826v5

[10] S. Ben-David and L. Reyzin, “Data stability in clustering: A closer look,” *Theoretical Computer Science*, vol. 558, no. C, pp. 51–61, nov 2014.

[11] Y. Bilu, A. Daniely, N. Linial, and M. Saks, “On the practically interesting instances of maxcut,” 2012.

[12] Y. Bilu and N. Linial, “Are stable instances easy?” *Combinatorics Probability and Computing*, vol. 21, no. 5, pp. 643–660, jun 2009. [Online]. Available: https://arxiv.org/abs/0906.3162v1

[13] I. Caragiannis, E. Elkind, M. Szegedy, and L. Yu, “Mechanism design: From partial to probabilistic verification,” *Proceedings of the ACM Conference on Electronic Commerce*, pp. 266–283, 2012.

[14] G. Carroll, “When are local incentive constraints sufficient?” *Econometrica*, vol. 80, no. 2, pp. 661–686, 2012. [Online]. Available: http://www.jstor.org/stable/41493830

[15] H. Chan, A. Filos-Ratsikas, B. Li, M. Li, and C. Wang, "Mechanism design for facility location problems: A survey," *CoRR*, vol. abs/2106.03457, 2021. [Online]. Available: https://arxiv.org/abs/2106.03457

[16] Z. Chen, K. C. K. Fong, M. Li, K. Wang, H. Yuan, and Y. Zhang, "Facility location games with optional preference," *Theoretical Computer Science*, vol. 847, pp. 185–197, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397520305661

[17] E. Dokow, M. Feldman, R. Meir, and I. Nehama, "Mechanism design on discrete lines and cycles," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, ser. EC '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 423–440. [Online]. Available: https://doi.org/10.1145/2229012.2229045

[18] B. Escoffier, L. Gourvès, K. T. Nguyen, F. Pascual, and O. Spanjaard, "Strategy-proof Mechanisms for Facility Location Games with Many Facilities," in *2nd International Conference on Algorithmic Decision Theory (ADT'11)*, ser. Lecture Notes in Artificial Intelligence, vol. 6992. Piscataway, NJ, United States: Springer, Oct. 2011, pp. 67–81. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01285708

[19] I. Feigenbaum, M. Li, J. Sethuraman, F. Wang, and S. Zou, "Strategic facility location problems with linear single-dipped and single-peaked preferences," *Autonomous Agents and Multi-Agent Systems*, vol. 34, no. 2, pp. 1–47, oct 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10458-020-09472-9

[20] M. Feldman and Y. Wilf, "Randomized strategyproof mechanisms for facility location and the mini-sum-of-squares objective," *CoRR*, vol. abs/1108.1762, 2011. [Online]. Available: http://arxiv.org/abs/1108.1762

[21] A. Filimonov and R. Meir, "Strategyproof facility location mechanisms on discrete trees," *CoRR*, vol. abs/2102.02610, 2021. [Online]. Available: https://arxiv.org/abs/2102.02610

[22] D. Fotakis and P. Patsilinakos, "Strategyproof facility location in perturbation stable instances," in *WINE*, 2021.

[23] D. Fotakis and C. Tzamos, "On the power of deterministic mechanisms for facility location games," *CoRR*, vol. abs/1207.0935, 2012. [Online]. Available: http://arxiv.org/abs/1207.0935

[24] ——, "Strategyproof facility location for concave cost functions," in *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, ser. EC '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 435–452. [Online]. Available: https://doi.org/10.1145/2492002.2482595

[25] ——, "Winner-imposing strategyproof mechanisms for multiple facility location games," *Theor. Comput. Sci.*, vol. 472, p. 90–103, feb 2013. [Online]. Available: https://doi.org/10.1016/j.tcs.2012.11.036

[26] D. Fotakis, C. Tzamos, and M. Zampetakis, "Mechanism design with selective verification," in *Proceedings of the 2016 ACM Conference on Economics and Computation*, ser. EC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 771–788. [Online]. Available: https://doi.org/10.1145/2940716.2940743

[27] D. Fotakis and E. Zampetakis, "Truthfulness flooded domains and the power of verification for mechanism design," *ACM Trans. Econ. Comput.*, vol. 3, no. 4, jul 2015. [Online]. Available: https://doi.org/10.1145/2790086

[28] A. Gibbard, "Manipulation of Voting Schemes: A General Result," *Econometrica*, vol. 41, no. 4, p. 587, jul 1973.

[29] S. Goel and W. Hann-Caruthers, “Optimality of the coordinate-wise median mechanism for strategyproof facility location in two dimensions,” *CoRR*, jul 2020. [Online]. Available: https://arxiv.org/abs/2007.00903v4

[30] I. Golomb and C. Tzamos, “Truthful facility location with additive errors,” *CoRR*, vol. abs/1701.00529, 2017. [Online]. Available: http://arxiv.org/abs/1701.00529

[31] J. R. Green and J.-J. Laffont, “Partially Verifiable Information and Mechanism Design,” *The Review of Economic Studies*, vol. 53, no. 3, pp. 447–456, mar 1986. [Online]. Available: http://www.jstor.org/stable/2297639

[32] M. Kyropoulou, C. Ventre, and X. Zhang, “Mechanism design for constrained heterogeneous facility location,” in *Algorithmic Game Theory: 12th International Symposium, SAGT 2019, Athens, Greece, September 30 – October 3, 2019, Proceedings.* Berlin, Heidelberg: Springer-Verlag, 2019, p. 63–76. [Online]. Available: https://doi.org/10.1007/978-3-030-30473-7_5

[33] M. Li, P. Lu, Y. Yao, and J. Zhang, “Strategyproof mechanism for two heterogeneous facilities with constant approximation ratio,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, ser. IJCAI’20, 2021.

[34] P. Lu, X. Sun, Y. Wang, and Z. A. Zhu, “Asymptotically optimal strategy-proof mechanisms for two-facility games,” *Proceedings of the ACM Conference on Electronic Commerce*, pp. 315–324, 2010.

[35] P. Lu, Y. Wang, and Y. Zhou, “Tighter bounds for facility games,” in *Internet and Network Economics - 5th International Workshop, WINE 2009, Proceedings*, ser. Lecture Notes in Computer Science, Dec. 2009, pp. 137–148.

[36] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, “The planar k-means problem is NP-hard,” *Theoretical Computer Science*, vol. 442, pp. 13–21, jul 2012.

[37] L. Mei, M. Li, D. Ye, and G. Zhang, “Facility location games with distinct desires,” *Discrete Applied Mathematics*, vol. 264, pp. 148–160, 2019. [Online]. Available: https://doi.org/10.1016/j.dam.2019.02.017

[38] R. Meir, “Strategyproof Facility Location for Three Agents on a Circle,” *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 4, pp. 2120–2122, feb 2019. [Online]. Available: https://arxiv.org/abs/1902.08070v2

[39] E. Miyagawa, “Locating libraries on a street,” *Social Choice and Welfare*, vol. 18, no. 3, pp. 527–541, 2001. [Online]. Available: http://www.jstor.org/stable/41060213

[40] H. Moulin, “On strategy-proofness and single peakedness,” *Public Choice 1980 35:4*, vol. 35, no. 4, pp. 437–455, jan 1980. [Online]. Available: https://link.springer.com/article/10.1007/BF00128122

[41] N. Nisan and A. Ronen, “Algorithmic Mechanism Design,” *Games and Economic Behavior*, vol. 35, no. 1-2, pp. 166–196, apr 2001.

[42] K. Nissim, R. Smorodinsky, and M. Tennenholtz, “Approximately Optimal Mechanism Design via Differential Privacy,” *ITCS 2012 - Innovations in Theoretical Computer Science Conference*, pp. 203–213, apr 2010. [Online]. Available: https://arxiv.org/abs/1004.2888v4

[43] H. Peters, H. van der Stel, and T. Storcken, “Pareto optimality, anonymity, and strategy-proofness in location problems,” *International Journal of Game Theory*, vol. 21, no. 3, pp. 221–235, sep 1992.

[44] A. D. Procaccia and M. Tennenholtz, “Approximate Mechanism Design without Money,” *ACM Transactions on Economics and Computation (TEAC)*, vol. 1, no. 4, pp. 1–26, dec 2013. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2542174.2542175

[45] T. Roughgarden, *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2020.

[46] ——, “Lecture 6: Perturbation-stable clustering,” *CS264: Beyond Worst-Case Analysis*, 2017. [Online]. Available: http://timroughgarden.org/w17/l/l6.pdf

[47] M. A. Satterthwaite, “Strategy-proofness and Arrow’s conditions: Existence and correspondence theorems for voting procedures and social welfare functions,” *Journal of Economic Theory*, vol. 10, no. 2, pp. 187–217, apr 1975.

[48] J. Schummer and R. Vohra, “Mechanism design without money,” *Algorithmic Game Theory*, vol. 10, pp. 243–299, 2007.

[49] J. Schummer and R. V. Vohra, “Strategy-proof location on a network,” *Journal of Economic Theory*, vol. 104, no. 2, pp. 405–428, jun 2002.

[50] P. Serafino and C. Ventre, “Heterogeneous facility location without money,” *Theor. Comput. Sci.*, vol. 636, no. C, p. 27–46, jul 2016. [Online]. Available: https://doi.org/10.1016/j.tcs.2016.04.033

[51] X. Sui, C. Boutilier, and T. Sandholm, “Analysis and optimization of multi-dimensional percentile mechanisms,” in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI ’13. AAAI Press, 2013, p. 367–374.