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

Multiple Facility Location Mechanisms with Predictions

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΜΜΑΝΟΥΗΛ ΠΑΔΟΥΒΑΣ

Επιβλέπων : Δημήτριος Φωτάκης
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2023

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

Multiple Facility Location Mechanisms with Predictions

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΜΜΑΝΟΥΗΛ ΠΑΔΟΥΒΑΣ

Επιβλέπων : Δημήτριος Φωτάκης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 01η Νοεμβρίου 2023.

....	....	....
Δημήτριος Φωτάκης Καθηγητής Ε.Μ.Π.	Αριστείδης Παγουρτζής Καθηγητής Ε.Μ.Π.	Ευάγγελος Μαρκάκης Αν. Καθηγητής Ο.Π.Α.

Αθήνα, Νοέμβριος 2023

....

Εμμανουήλ Παδουβάς

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

Περίληψη

Σε αυτή τη διπλωματική εργασία, μελετούμε τα παίγνια χωροθέτησης πολλαπλών υπηρεσιών, όπου n στρατηγικοί παίκτες αναφέρουν τις θέσεις τους στη γραμμή, και ένας μηχανισμός τους αντιστοιχίζει σε k $\geq$ 2 υπηρεσίες. Κάθε παίκτης επιδιώκει να ελαχιστοποιήσει την απόστασή του από την πλησιέστερη υπηρεσία. Ενδιαφερόμαστε για μηχανισμούς που είναι ανθεκτικοί στη στρατηγική συμπεριφορά των παικτών (strategyproof) χωρίς πληρωμές, οι οποίοι παρέχουν μία λογική προσέγγιση στο κοινωνικό βέλτιστο κόστος των παικτών. Για να αντιμετωπίσουμε το θέωρημα ανυπαρξίας φιλαληθών ντετερμινιστικών μηχανισμών με φραγμένο λόγο προσέγγισης των παιγνίων k-Facility Location για k ≥ 3, περιορίζουμε την προσοχή μας σε στιγμιότυπα που επιδεικνύουν ευστάθεια σε διαταραχές (perturbation stable instances). Η ευστάθεια σε διαταραχές εισήχθη για το πρόβλημα MAX CUT από τους Bilu και Linial και αργότερα εφαρμόστηκε στο πρόβλημα της συσταδοποίησης (clustering). Τα παραδείγματα με ευστάθεια σε διαταραχές έχουν μια καλά καθορισμένη βέλτιστη λύση, η οποία δεν επηρεάζεται από μικρές διαταραχές στα δεδομένα. Παρομοίως, ένα παράδειγμα του προβλήματος k-Facility Location στη γραμμή είναι γ-ευσταθές, για κάποιο $\gamma \geq 1$, αν η βέλτιστη λύση δεν επηρεάζεται από την αλλαγή στην απόσταση μεταξυ των θέσεων διαφορετικών παικτών, η οποία εξαρτάται από ενα παράγοντα γ. Θα επωφεληθούμε επίσης από την πρόσφατη έρευνα στον τομέα "Σχεδιασμού Μηχανισμών ενισχυμένο από Μάθηση". Αυτή η προσέγγιση συμπληρώνει την παραδοσιακή προσέγγιση στην επιστήμη των υπολογιστών, που αναλύει την απόδοση αλγορίθμων βασισμένων στην χειρότερη περίπτωση και επικεντρώνεται στο σχεδιασμό και ανάλυση μηχανισμών που ενισχύονται με προβλέψεις, οι οποίες έχουν αποκτηθεί από μηχανική μάθηση σχετικά με τη βέλτιστη λύση. Χρησιμοποιώντας αυτές τις προβλέψεις ως καθοδηγητικά στοιχεία, στόχος μας είναι να επιτύχουμε πολύ καλές εγγυήσεις όταν οι προβλέψεις μας είναι ακριβείς (συνέπεια), παραμένοντας παράλληλα κοντά στην βέλτιστη δυνατή προσέγγιση για την χειρότερη περίπτωση, ακόμα και όταν οι προβλέψεις είναι λανθασμένες (ανθεκτικότητα). Στόχος μας είναι να συνδυάσουμε τα παραπάνω στοιχεία στον σχεδιασμό μηχανισμών ενισχυμένων από μάθηση για τα παιγνιά χωροθέτησης πολλαπλών υπηρεσιών σε στιγμιότυπα με ευστάθεια σε διαταραραχές και να κάνουμε παρατήρησεις πάνω στους περιορισμούς τους.

Λέξεις κλειδιά

Προβλήματα Χωροθέτησης, Σχεδιασμός μηχανισμών χωρίς χρήματα, Ευστάθεια σε διαταραχές, Σχεδιασμός μηχανισμών ενισχυμένων με Μάθηση.

Abstract

In this diploma thesis, we study k-Facility Location games, where n strategic agents report their locations on the real line, and a mechanism maps them to $k \geq 2$ facilities. Each agent seeks to minimize his distance to the nearest facility. We are interested in strategyproof mechanisms without payments that achievve a reasonable approximation ratio to the optimal social cost of the agents. To circumvent the unbounded approximability of k-Facility Location by deterministic strategyproof mechanisms for $k \geq 3$, we restrict our attention to perturbation stable instances. Perturbation Stability was introduced for the MAX CUT problem from Bilu and Linial and was later applied to the clustering problem. Perturbation Stable instances have a well-defined optimal clustering, which is unaffected by small perturbations of the input. Similarly, an instance of k-Facility Location on the line is γ-perturbation stable (or simply, γ-stable), for some $\gamma \geq 1$, if the optimal agent clustering is not affected by moving any subset of consecutive agent locations closer to each other by a factor at most γ. We will also benefit from the recent surge of work in "learning-augmented mechanism design". This approach complements the traditional approach in computer science, which analyzes the performance of algorithms based on worst-case instances, and focuses on the design and analysis of mechanisms that are enhanced with machine-learned predictions regarding the optimal solution. Using the predictions as guides, our aim is to achieve much better guarantees when the predictions are accurate (consistency), while maintaining near-optimal worst-case guarantee, even when the predictions are wrong (robustness). Our goal is to combine the above elements in designing learning-augmented mechanisms for the K-facility Location games problem on perturbation stable instances and make observations on their limitations.

Key words

Facility Location Games, Mechanism Design without Money, Perturbation Stability, Learning-Augmented Mechanism Design.

Ευχαριστίες

Ξεκινώντας, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητη για την διπλωματική μου - κύριο Φωτάκη. Από την αρχή της διπλωματικής μου ήταν δίπλα μου και με την βοήθεια και τις συμβούλες του ήταν μαζί μου σε κάθε βήμα. Επίσης θα ήθελα να ευχαριστήσω και τον Παναγιώτη Πατσιλινάκο, με τον οποίο είχα άριστη συνεργασία. Εκτιμώ πραγματικά τον χρόνο που μου αφιέρωσαν και οι δύο, καθώς έμαθα πολλά για τον τροπο που πρέπει να εργάζεται κάποιος που νοιάζεται για αυτό που κάνει, αλλά και γιατί με ενέπνευσαν με το χαρακτηρα τους και τη προσωπικότητα τους.

Έπειτα θα ήθελα να ευχαριστήσω τους φίλους μου και συμφοιτητές μου Βαγγελή, Πάνο και Τάσο, καθώς και τους φίλους μου Νίκο, Νικήτα και την κοπέλα μου, Φαιναρέτη, οι οποίοι με την συντροφιά τους με στήριξαν όλα αυτά τα χρόνια.

Θα ήθελα επίσης να ευχαριστήσω την οικογένεια μου, την αδελφή μου, τον αδελφο μου και τους γονείς που χωρίς την υποστήριξη τους και την αγάπη τους δεν θα μπορούσα να ολοκληρώσω αυτά που θέλω να πέτυχω.

Τέλος θέλω να ευχαριστήσω την κυρία Αναστασία Πατρικίου που με τις συμβουλές και την φροντίδα της, με βοήθησε να αντιμετωπίσω ότι δυσκολίες προέκυπταν στο δρόμο μου.

Εμμανουήλ Παδουβάς,

Αθήνα, 01η Νοεμβρίου 2023

Περιεχόμενα

Κεφάλαιο 1

Εκτενής Ελληνική Περίληψη

Ένα από τα πιο ενδιαφέροντα χαρακτηριστικά των μαθηματικών είναι πώς μπορούμε να ερμηνεύσουμε τις πολλές εφαρμογές τους σε φαινομενικά μη-ομοιόμορφους ακαδημαϊκούς τομείς. Ένας αναδυόμενος τομέας είναι ο Αλγοριθμικός Σχεδιασμός Μηχανισμών, ένας συνδυασμός της Θεωρίας Κοινωνικής Επιλογής, της Θεωρίας Παιγνίων, της Σχεδιασμού Μηχανισμων και της Επιστήμης των Υπολογιστών. Η Θεωρία Κοινωνικής Επιλογής εξετάζει τις διαδικασίες συλλογικών αποφάσεων και μηχανισμών. Δεν αποτελεί μια μόνο θεωρία, αλλά ένα σύμπλεγμα μοντέλων και αποτελεσμάτων που αφορούν την συγκέντρωση ατομικών δεδομένων (π.χ. ψήφοι, προτιμήσεις, κρίσεις, ευημερία) σε συλλογικά αποτελέσματα (π.χ. συλλογικές αποφάσεις, προτιμήσεις, κρίσεις, ευημερία). Ο Σχεδιασμός Μηχανισμών ανήκει στον τομέα της οικονομικής θεωρία, χαρακτηριστικά του οποίου αναλύει με σκοπια μηχανικού. Στοχεύει στο σχεδιασμό οικονομικών μηχανισμών, όπως οι επιστήμονες υπολογιστών ενδιαφέρονται να σχεδιάσουν αλγόριθμους, πρωτόκολλα ή συστήματα. Ο Αλγοριθμικός Σχεδιασμός Μηχανισμών μελετά προβλήματα βελτιστοποίησης όπου τα αρχικά δεδομένα - όπως η αξία ενός αγαθού ή το κόστος εκτέλεσης μιας εργασίας - δεν είναι γνωστά στον σχεδιαστή του αλγορίθμου και πρέπει να αποκτηθούν είτε έμμεσα είτε ρητά από συμμετέχοντες, που ενεργούν στρατηγικά. Στον σχεδιασμό αλγορίθμων, συχνά δεν ανησυχούμε για την εγκυρότητα της εισόδου μας. Σε αυτήν την περίπτωση, θεωρούμε τους συμμετέχοντες ως στρατηγικούς στις ενέργειές τους, καθώς και ότι πάντοτε αναζητούμε το μέγιστο κέρδος τους, το οποίο εξαρτάται από το ”παιχνίδι” που παίζουν, δηλαδή μπορούν να ψεύδονται. Ο σχεδιαστής επιδιώκει να εκμεταλλευτεί αυτό το γεγονός και να δημιουργήσει ένα σύνολο κανόνων που θα καθοδηγήσουν τους παίκτες να ενεργούν με τρόπο που παράγει το βέλτιστο αποτέλεσμα, με βάση έναν στόχο που σχετίζεται με το παιχνίδι (π.χ. μέγιστα έσοδα, μεγιστοποίηση κοινωνικής ευημερίας κ.λπ.). Συνεχίζοντας τη σύνδεση μεταξύ της διαδικασίας κατασκευής μηχανισμών και αλγορίθμων, στοχεύουμε στη δημιουργία υπολογιστικά αποδοτικών μηχανισμών που δεν επηρεάζονται από τα ψέματα των συμμετεχόντων, διατηρώντας την πολύτιμή ιδιότητα της φιλαλήθειας (strategyproofness).

Όπως αναφέρθηκε παραπάνω, οι μηχανισμοί μας πρέπει να μην επηρεάζονται από το ψέμα ενός συμμετέχοντα ή, ακόμη καλύτερα, να πείθουν όλους τους συμμετέχοντες ότι το να λένε την αλήθεια είναι στο δικό τους συμφέρον. Ορισμένοι μηχανισμοί χρησιμοποιούν χρήματα και πληρωμές για να επιβάλλουν τέτοιες συνθήκες, αλλά σε άλλες περιπτώσεις, οι πληρωμές θα μπορούσαν να είναι παράνομες ή ανήθικες. Σε αυτή τη διατριβή, εξετάζουμε ένα από τα βασικά προβλήματα στην τελευταία κατηγορία, τα Παίγνια Χωροθέτησης Υπηρεσιών (Facility Location Games). Εξετάζουμε τα k-Facility Location games, όπου τουλάχιστον $k \geq 1$ υπηρεσίες τοποθετούνται στην πραγματική γραμμή βάσει των προτιμήσεων των n στρατηγικών παικτών. Αυτά τα προβλήματα προκύπτουν από σενάρια της Κοινωνικής Επιλογής, όπου μια τοπική αρχή σχεδιάζει να κατασκευάσει έναν σταθερό αριθμό δημόσιων υπηρεσιών σε μια περιοχή [39]. Η επιλογή των τοποθεσιών βασίζεται στις προτιμήσεις των τοπικών κατοίκων ή παικτών. Κάθε παίκτης αναφέρει την ιδανική του τοποθεσία και η τοπική αρχή εφαρμόζει έναν (ντετερμινιστικό ή τυχαιοποιμένο) μηχανισμό που αντιστοιχεί τις προτιμήσεις των παικτών σε k τοποθεσίες υπηρεσιών. Κάθε παίκτης προσπαθεί να μειώσει το κόστος σύνδεσής του - την απόστασή του από την πλησιέστερη υπηρεσία, και ο σχεδιαστής του μηχανισμού προσπαθεί να βελτιστοποιήσει ένα συγκεκριμένο στόχο (Κοινωνικό Κόστος, Μέγιστο Κόστος, κ.λπ.). Από τότε που οι Procaccia και Tennenholtz [43] ξεκίνησαν την έρευνα περί σχεδίασης μηχανισμών χωρίς πληρωμές, το k-Facility Location χρησιμοποιήθηκε ως πρόβλημα αναφοράς στον τομέα και εξετάστηκε εκτενώς σε σχεδόν όλες τις δυνατές παραλλαγές. Για παράδειγμα, η προηγούμενη έρευνα εξέτασε πολλαπλές υπηρεσίες στη γραμμή (βλ. π.χ., ([21],[24],[27],[33],[42]) και γενικούς μετρικούς χώρους ([20],[32]), διάφορετικούς στόχους (π.χ., κοινωνικό κόστος, μέγιστο κόστος, η L_2 νόρμα των συνδέσεων των παικτών ([19],[24],[43]), περιορισμένους μετρικούς χώρους πιο γενικούς από τη γραμμή (κύκλος, επίπεδο, δέντρα, βλ. π.χ., ([2],[16],[16],[25],[16]), υπηρεσίες που εξυπηρετούν διάφορους στόχους (βλ. π.χ.,[30],[31]), και διάφορετικές έννοιες των ιδιωτικών πληροφοριών σχετικά με τις προτιμήσεις των παικτών που πρέπει να ανακοινώνονται στον μηχανισμό (βλ. π.χ., [15],[18],[36]). Λόγω του σημαντικού ενδιαφέροντος για το θέμα, η πιο βασική ερώτηση σχετικά με την προσέγγιση του βέλτιστου κοινωνικού κόστους από μηχανισμούς που διατηρούν τη φιλαληθή ιδιότητα για το k-Facility Location στη γραμμή έχει κατανοηθεί σχετικά καλά. Για μία υπηρεσία, ο μηχανισμός που τοποθετεί την υπηρεσία στη θέση του διάμεσου παίκτη είναι και βέλτιστος και φιλαληθής. Για δύο υπηρεσίες, η τοποθέτηση στα δύο άκρα τηου στιγμιότυπου θα διατηρήσει τη φιλαληθή ιδιότητα, παράγοντας τη καλύτερη δυνατή προσέγγιση (n - 2). Δυστυχώς, για k $\geq$ 3, μας δίνεται το αρνητικό αποτέλεσμα της [21], όπου αποδεικνύεται ότι δεν υπάρχει ανώνυμος, ντετερμινιστικός, φιλαληθής μηχανισμός με φραγμένη προσέγγιση, για το K-Facility Location, με k $\geq$ 3. Αυτό το αποτέλεσμα μας οδήγησε στο να αναγνωρίσουμε τα όρια της γενικού πλαίσου των παίγνιων χωροθέτησης υπηρεσιών και να στραφούμε προς περιπτώσεις που είναι πιο κοντά στον πραγματικό κόσμο και διατηρούν ιδιότητες που μπορούν να εκμεταλλευτούν από τον μηχανισμό μας. Σε αυτή τη διατριβή, θα επικεντρωθούμε σε αυτόν τον νέο τύπο περίπτωσης, όπου δεν θα έχουμε μόνο μία εμφανή βέλτιστη ομαδοποίηση, αλλά θα υποστηριζόμαστε και από έναν εξωτερικό σύστημα στην τοποθέτηση των υπηρεσιών μας.

Οι ομοιότητες μεταξύ της ομαδοποίησης των δεδομένων, που προσομοιώνεται από ένα μετρικό χώρο (X, d), και των παίγνιων χωροθέτησης υπηρεσιών είναι τόσες πολλές ώστε δεν μπορούν να αγνοηθούν. Και τα δύο αυτά προβλήματα αναζητούν τρόπους για την βέλτιστη ομαδοποίηση. Στην ομαδοποίηση, ενδέχεται να μην πρέπει να ασχοληθούμε με έναν ”ψεύτικο” μετρικό χώρο, αλλά αφού πρόκειται για έναν εξαιρετικά ερευνημένο τομέα, έχουμε βρει πολλούς τρόπους για να χαρακτηρίζουμε τα δεδομένα εισόδου και την ανάπτυξη αλγορίθμων που εκμεταλλεύονται αυτά τα χαρακτηριστικά. Ένα από αυτά τα χαρακτηριστικά είναι τα ”στιγμιότυπα ευσταθή σε διαταραχές” (perturbation stable instances), οι οποίες είναι περιπτώσεις που μοιάζουν με τα δεδομένα του πραγματικού κόσμου. Σε αυτές τις περιπτώσεις, υποθέτουμε ότι υπάρχει δομή στα δεδομένα και ακόμα και μικρές διαταραχές δεν μπορούν να αλλάξουν τη δομή της εισόδου. Οι διαταραχές αυτές εισήχθησαν από τους Bilu και Linial στο [13] και από τους Awasthi, Blum και Sheffet στο [6] και έχουν ενθαρρύνει ένα σημαντικό όγκο επιπλέον εργασίας μετά από αυτό, (βλ. π.χ. [4],[8],[10],[45] και τις αναφορές εκείνων) στην προσπάθεια να αποκτήσουν θεωρητική κατανόηση της ανώτερης πρακτικής απόδοσης απλών αλγορίθμων ομαδοποίησης για γνωστά NP-δύσκολα προβλήματα ομαδοποίησης (όπως το k-Facility Location σε γενικούς μετρικούς χώρους). Στην ουσία, η βέλτιστη ομαδοποίηση ενός στιγμιοτύπου που είναι γ-ευσταθές είναι εμφανής και, συνεπώς, απλοί αλγόριθμοι ομαδοποίησης, όπως ο αλγόριθμος single-clustering, μπορούν να επιλεχθούν στην προσπάθειάς μας για την παραγωγή βέλτιστης ομαδοποίησης σε πολυωνυμικό χρόνο, για μια κατάλληλη τιμή του γ. Η απλότητα αυτών των αλγορίθμων ενισχύεται από τις ιδιότητες ευστάθειας που έχουν αποδειχτεί, όπως η γ-center proximity, weak γ-center proximity, και η Cluster-Separation Property. Αυτές οι ιδιότητες καθορίζουν τα όρια μεταξύ των αποστάσεων μεταξύ των ομάδων, αλλά και των σημείων εντός των ομάδων, καθιστώντας πιο εύκολο τον διαχωρισμό των ομάδων σε μια περίπτωση. Μια φυσική επέκταση θα ήταν να εφαρμόσουμε αυτές τις ιδιότητες στον τομέα των παίγνιών χωροθέτησης υπηρεσιών και να εξετάσουμε την βέλτιστη λύση και την φιλαλήθεια του μηχανισμού μας σε στιγμιότυπα γ-ευσταθή. Όλα τα αποτελέσματα στο προηγούμενο τμήμα βασίζονται στο σενάριο χειρότερης περίπτωσης, το οποίο περιορίζεται από το [21]. Ωστόσο, αν χρησιμοποιήσουμε την έννοια της ευστάθειας στις περιπτώσεις μας, μπορούμε να καταλήξουμε σε ορισμένα ενδιαφέροντα αποτελέσματα. Στο [23], η μελέτη των αποτελεσματικών (όσον αφορά τον λόγο προσέγγισης για το κοινωνικό κόστος) φιλαληθών μηχανισμών ξεκίνησε για τη μεγάλη και φυσική κλάση των γ-ευσταθών στιγμιοτύπων του k-Facility Location στη γραμμή. Παρουσιάστηκαν ντετερμινιστικοί και τυχαιοποιημένοι φιλαληθείς μηχανισμοί με φραγμένο λόγο προσέγγισης για 5-ευσταθή στιγμιότυπα και οποιοδήποτε αριθμό υπηρεσιών $k \geq 2$. Επιπλέον, έδειξαν ότι η βέλτιστη ομαδοποίηση είναι φιλαληθής για $(2 + \sqrt{3})$-ευσταθή στιγμιότυπα αν η βέλτιστη ομαδοποίηση δεν περιλαμβάνει καμία μονομελή ομάδα (κάτι που είναι πιθανό να συμβαίνει σχεδόν σε όλες τις πρακτικές εφαρμογές). Επιπλέον, η αδυναμία του αποτελέσματος του [21] ενισχύθηκε, έτσι ώστε να ισχύει για γ-ευσταθή στιγμιότυπα, με $\gamma < (\sqrt{2} - \delta)$. Συγκεκριμένα, δείχθηκε ότι για οποιονδήποτε $k \geq 3$ και οποιοδήποτε $\delta > 0$, δεν υπάρχουν ντετερμινιστικοί ανώνυμοι φιλαληθείς μηχανισμοί για το k-Facility Location σε $(\sqrt{2} - \delta)$-σταθείς περιπτώσεις με φραγμένο (όσον αφορά το n και το k) λόγο προσέγγισης. Απομένει να δούμε εάν το $(\sqrt{2} - \delta)$ όριο είναι βέλτιστο και αν υπάρχει ντετερμινιστικός, φιλαληθής μηχανισμός για γ-ευσταθή στιγμιότυπα, με $\gamma < 5$, ο οποίος μπορεί επίσης να αντιμετωπίσει μονομελείς ομάδες. Αυτή η διατριβή θα επικεντρωθεί στον εντοπισμό της βέλτιστης ομαδοποίησης σε γ-ευσταθή στιγμιότυπα, λαμβάνοντας κάποιες ”πρόσθετες” πληροφορίες από μια εξωτερική πηγή.

Η έννοια των στιγμιοτύπων με ευσταθεια σε διαταραχές αποτελεί μόνο μία από τις πολλές διάφορες προσεγγίσεις που μπορούμε να ακολουθήσουμε όταν κοιτάμε πέρα από το πλαίσιο ανάλυσης της χειρότερης περίπτωσης. Τα τελευταία χρόνια, έχει γίνει αξιόλογη έρευνα στους αλγορίθμους που ενισχύονται από τη μηχανική μάθηση, δημιουργώντας έτσι τον νέο τομέα των ”αλγορίθμων με προβλέψεις”, προσδίδοντας έτσι πραγματικά κίνητρα για να καθοριστεί εάν η συμβολή των προβλέψεων στους γνωστούς μηχανισμούς των παίγνιών χωροθέτησης υπηρεσιών ή η ανάπτυξη νέων μηχανισμών με ενσωματωμένη μάθηση μπορεί να οδηγήσει σε ενδιαφέροντα αποτελέσματα. Νέες μετρικές - Συνέπεια και Ανθεκτικότητα - εισήχθησαν στο [38] ως οι βασικές μέτρικές στους αλγόριθμους με προβλέψεις. Αυτές επεκτείνονται πέρα από την έννοια του λόγου προσέγγισης: Η Συνέπεια είναι ο λόγος προσέγγισης του μηχανισμού, όταν οι προβλέψεις συμπίπτουν με τη βέλτιστη λύση και η Ανθεκτικότητα είναι ο λόγος προσέγγισης του μηχανισμού όταν οι προβλέψεις είναι τυχαία εσφαλμένες. Στο [1], υπήρξε μια αρχική προσπάθεια να εξερευνηθούν αυτοί οι νέοι μηχανισμοί. Παρουσιασαν έναν μηχανισμό για μία υπηρεσία στη γραμμή, επιτυγχάνοντας 1-συνέπεια και 1-ανθεκτικότητα για τον κοινωνικό κόστος και 1-συνέπεια και $(1 + \sqrt{2})$-ανθεκτικότητα για τον μέγιστο κόστος, ενώ αποδείχθηκε επίσης ότι η 1 -συνέπεια και $(1 + \sqrt{2})$-ανθεκτικότητας είναι η βέλτιστη ανταλλαγή συνέπειας-ανθεκτικότητας. Ένα ανοιχτό πρόβλημα είναι να βρεθεί τη βέλτιστη συνέπεια και ανθεκτικότητα στη γενική περίπτωση των k υπηρεσιών για γ-ευσταθή στιγμιότυπα, που είναι επίσης ο κύριος στόχος αυτής της διατριβής. Θα παρουσιάσουμε μια γενικευμένη έκδοση του ενισχυμένου με μάθηση μηχανισμού για μία υπηρεσία, πάνω στην περίπτωση παίγνιων χωροθέτησης υπηρεσιών για τουλάχιστον 5-ευσταθή στιγμιότυπα, που επιτυγχάνει 1-συνέπεια για και τον ΜΕΓΙΣΤΟ ΚΟΣΤΟΣ και τον ΚΟΙΝΩΝΙΚΟ ΣΤΟΧΟ, 2-ανθεκτικότητα για τον ΜΕΓΙΣΤΟ ΚΟΣΤΟΣ, (n-1)-ανθεκτικότητα για τον ΚΟΙΝΩΝΙΚΟ ΣΤΟΧΟ, ενώ παρατηρούμε επίσης τα όρια που συναντά αυτού του είδους η γενίκευση.

1.1 Παίγνια Χωροθέτησης Υπηρεσιών

Το πρόβλημα μας αποτελείται από n στρατηγικούς παίκτες και k υπηρεσίες. Οι παίκτες τοποθετούνται στον μετρικό χώρο (X, d), όπου $d: X \times X \implies R{\geq}0$ είναι η συνάρτηση απόστασης. Η συνάρτηση d είναι μια μετρική στο X που ικανοποιεί τις εξής ιδιότητες: $d(x, x) = 0$ για όλα τα $x \in X$, $d(x, y) = d(y,$

x) για όλα τα x, y ∈ X (συμμετρία) και d(x, z) ≥ d(x, y) + d(x, z) για όλα τα x, y, z ∈ X (ανισότητα τριγώνου). Κάθε παίκτης i ∈ N έχει μια τοποθεσία x_i, η οποία είναι η ιδιωτική του πληροφορία, και στον ίδιο μετρικό χώρο πρέπει να τοποθετήσουμε τις υπηρεσίες. Αναφερόμαστε στο σύνολο $\vec{x}$ = (x_1, ..., x_n) ως προφίλ στιγμιοτύπου. Το κόστος σύνδεσης του παίκτη i, που σημειώνεται ως cost($x_i, \vec{y}$), είναι η ελάχιστη απόσταση ανάμεσα στη θέση του παίκτη και την πλησιέστερη θέση της υπηρεσίας. Ο στόχος μας είναι να τοποθετήσουμε k υπηρεσίες στον μετρικό χώρο, προσπαθώντας να ελαχιστοποιήσουμε μια συνάρτηση κόστους, η οποία εξαρτάται από τα κόστη σύνδεσης των παικτών. Κάθε παίκτης προσπαθεί να ελαχιστοποιήσει το κόστος της σύνδεσής του.

Ένας ντετερμινιστικός Μηχανισμός M αντιστοιχίζει το $\vec{x}$ σε ένα διάνυσμα (y_1, ..., y_k) ∈ X^k τοποθεσίων για υπηρεσίες. Το M($\vec{x}$) συμβολίζει το αποτέλεσμα του Μηχανισμού M. Ένας τυχαιοποιημένος Μηχανισμός M αντιστοιχίζει το $\vec{x}$ σε μια κατανομή πάνω σε διανύσματα (y_1, ..., y_k) ∈ X^k τοποθεσιών για υπηρεσίες. Δύο από τις πιο βασικές συναρτήσεις κόστους είναι η συνάρτηση Κοινωνικού Κόστους, το άθροισμα όλων των κόστων σύνδεσης των παικτών, και η συνάρτηση Μέγιστου Κόστους, το μέγιστο ανάμεσα στα κόστα σύνδεσης όλων των παικτών. Το Κοινωνικό κόστος ενός προφίλ στιγμιοτύπου (y_1, ..., y_k) ∈ X^k είναι $SC(\vec{x}, \vec{y}) = \sum_{i=1}^{n} cost(x_i, \vec{y})$. Το Μέγιστο κόστος ενός προφίλ στιγμιοτύπου $(y_1, ..., y_k) \in X^k$ is $MC(\vec{x}, \vec{y}) = max_{i \in N}[cost(x_i, \vec{y})]$.

Ένας μηχανισμός M για τα Παίγνια Χωροθέτησης Υπηρεσιών επιτυγχάνει ένα λόγο προσέγγισης ρ ≥ 1 για τον σκοπό του Κοινωνικού κόστους (αντίστοιχα για το Μέγιστο κόστος), για όλα τα προφίλ $\vec{x}, SC(\vec{x}, M(\vec{x})) \leq \rho SC^*(\vec{x})$ (αντίστοιχα για το μέγιστο κόστος $MC(\vec{x}, M(\vec{x})) \leq \rho MC^*(\vec{x})$). Ένας μηχανισμός M είναι φιλαληθής αν κανένας παίκτης δεν μπορεί να επωφεληθεί από το να παραποιήσει την τοποθεσία του. Συγκεκριμένα, για όλα τα προφίλ $\vec{x}$, κάθε παίκτης i και όλες τις τοποθεσίες $y \in X$, $cost(x_i, M(\vec{x})) \leq cost(x_i, M(\vec{x}_{-i}, y))$.

1.1.1 Μηχανισμός για μία υπηρεσία στη γραμμή

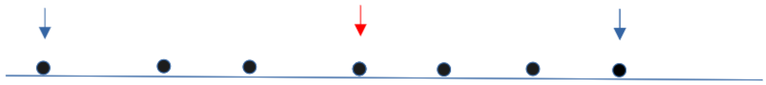
Ο πρώτος μας στόχος είναι να βρούμε ένα φιλαληθή Μηχανισμό που ελαχιστοποιεί το κοινωνικό κόστος για το παιχνίδι χωροθέτησης υπηρεσίων με n παίκτες και μια υπηρεσία. Η λύση είναι αρκετά απλή. Μπορούμε να επιλέξουμε τη τοποθεσία του διάμεσου παίκτη στο $\vec{x}$ - $med(\vec{x})$. Εάν επιλέξουμε οποιονδήποτε παίκτη αριστερά από το $med(\vec{x})$, τότε το κοινωνικό κόστος αυξάνεται, αφού είναι πιο μακριά από τουλάχιστον k + 1 παίκτες και πιο κοντά σε το πολύ k παίκτες. Το ίδιο ισχύει για οποιονδήποτε παίκτη δεξιά από τη μέση. Υποθέτουμε ότι το n είναι ζυγό, n = 2k, τότε οποιοδήποτε σημείο στο $[x_k, x_{k+1}]$ παράγει το βέλτιστο κοινωνικό κόστος, για τον ίδιο λόγο με την περίπτωση του n = 2k + 1. Έτσι, το $med(\vec{x})$ είναι βέλτιστο. Είναι επίσης φιλαληθής, καθώς ο παίκτης μπορεί μόνο να μετακινήσει την υπηρεσία πιο μακριά με το ψέμα του.

Θεώρημα 1.1. $M(\vec{x}) = med(\vec{x})$ *είναι φιλαληθής βελτιστος μηχανισμός για το κοινωνικό κόστος.*

Η δεύτερος μας στόχος είναι να βρούμε ένα βέλτιστο και φιλαληθή μηχανισμό για την συναρτηση του μέγιστου κόστους στην περίπτωση μιας υπηρεσίας. Η τοποθέτηση της υπήρεσιας που ελαχιστοποιεί τον στόχο μας είναι η θέση $cen(\vec{x})$. Δυστυχώς, αυτή η τοποθέτηση δεν είναι φιλαληθή, διότι κάθε παίκτης μπορεί να πει ψεματα και να αλλάξει το μήκος του στιγμιοτύπου, μέχρι ο $cen(\vec{x'})$ να καταλήξει στη θέση αυτού του παίκτη. Οι Procaccia και Tennenholtz [43] πρότειναν τον παρακάτω φιλαληθή μηχανισμό με λόγο προσέγγισης 2 για τον μέγιστο κόστος, $M(\vec{x}) = lt(\vec{x})$.

Θεώρημα 1.2. *Ο* $M(\vec{x}) = lt(\vec{x})$ *είναι ένας φιλαληθής μηχανισμός προσέγγισης 2 για τον μέγιστο κόστος.*

Κόκκινη Υπηρεσία είναι η βέλτιστη επιλογή



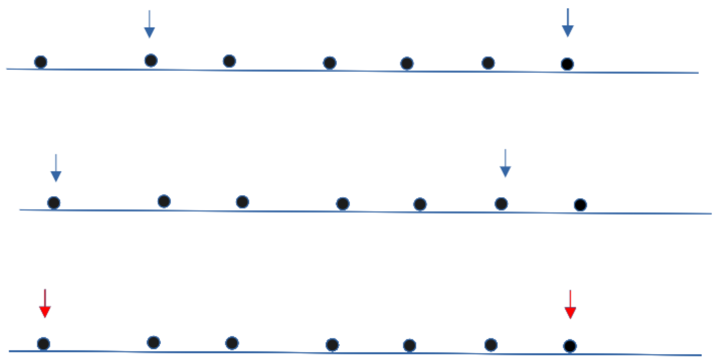
Ένα φυσικό ερώτημα που προκύπτει είναι αν αυτός είναι ο καλύτερος μηχανισμός που μπορούμε να χρησιμοποιήσουμε για τον στόχο του μέγιστου κόστους. Οι Procaccia και Tennenholtz [43] πράγματι απέδειξαν ότι ο $lt(\vec{x})$ είναι ο καλύτερος δυνατός μηχανισμός για την περίπτωση μιας υπηρεσίας, ως προς την συνάρτηση του μέγιστου κόστους. Και πράγματι, η επιλογή του αριστερότερου παίκτη είναι η καλύτερη που μπορούμε να κάνουμε αν θέλουμε έναν ντετερμινιστικό φιλαληθή μηχανισμό.

Θεώρημα 1.3. *Για $N = \{1, ..., n\}$, $n \geq 2$.. Οποιοσδήποτε ντετερμινιστικός φιλαληθής μηχανισμός M: $\mathbb{R}^n \rightarrow R$ έχει ένα λόγο προσέγγισης τουλάχιστον 2 για το μέγιστο κόστος.*

1.1.2 Μηχανισμός για δύο υπηρεσίες στη γραμμή

Τώρα μπορούμε να ασχοληθούμε με την επέκταση της προηγούμενης ρύθμισης, την εύρεση δύο υπηρέσιων αντί για μία. Πρώτα, θα εξετάσουμε τον στόχο του μέγιστου κόστους. Δεδομένου του $\vec{x}$, ας ορίσουμε την τοποθεσία του αριστερού ορίου ως $lb(\vec{x}) = max\{x_i : i \in N, x_i \leq cen(\vec{x})\}$ και την τοποθεσία του δεξιού ορίου ως $rb(\vec{x}) = min\{x_i : i \in N, x_i \geq cen(\vec{x}\}$. Σημειώνουμε $dist(\vec{x}) = max\{lb(\vec{x}) - lt(\vec{x}), rb(\vec{x}) - rt(\vec{x})\}$. Θα εξετάσουμε τώρα την ελαχιστοποιησή του κοινωνικού κόστους με φιλαλήθη τρόπο. Εάν εξετάσουμε το αλγοριθμικό πρόβλημα τοποθέτησης δύο υπηρέσιων με τρόπο που ελαχιστοποιεί το κοινωνικό κόστος, αγνοώντας τα κινήτρα των παικτών. Δεδομένου ενός προφίλ $\vec{x} \in R^n$, οι βέλτιστες τοποθεσίες των υπηρέσιων είναι $y_1, y_2 \in R, y_1 \leq y_2$. Σε γενικές γραμμές, μπορούμε να συσχετίσουμε με το y_1 ένα πολλαπλό σύνολο τοποθεσιών $L(\vec{x}) \subsetneq (x_1, ..., x_n)$ (για την "αριστερή" υπηρεσία) των οποίων το κόστος υπολογίζεται ως προς το y_1, και αντίστοιχα συσχετίζουμε με το y_2 ένα πολλαπλό σύνολο τοποθεσιών $R(\vec{x}) \subsetneq (x_1, ..., x_n)$ (για την "δεξιά" υπηρεσία) των οποίων το κόστος υπολογίζεται ως προς το y_2, με τρόπο τέτοιο ώστε για κάθε $x_i \in L(\vec{x}), x_j \in R(\vec{x}), x_i \leq xj$. Τώρα, το y_1 είναι η μέση τιμή του $L(\vec{x})$ και το y_2 είναι η μέση τιμή του $R(\vec{x})$. Επομένως, είναι αρκετό να βελτιστοποιήσουμε για τις n - 1 δυνατές επιλογές των $L(\vec{x})$ και $R(\vec{x})$. Μπορεί να επιβεβαιωθεί ότι ένας ομαδικός φιλαληθής με (n - 1) προσέγγιση μηχανισμός δίνεται από την επιλογή των $lt(\vec{x})$ και $rt(\vec{x})$ με βάση το προφίλ $\vec{x} \in R^n$. Συνοπτικά, ο λόγος είναι ότι το $lt(\vec{x}) \in L(\vec{x})$ και το $rt(\vec{x}) \in R(\vec{x})$ [43].

Είναι αυτός ο καλύτερος τρόπος που μπορούμε να χρησιμοποιήσουμε για τον στόχο του κοινωνικού κόστους όταν εφαρμόζουμε καλούς μηχανισμούς στη ρύθμιση της τοποθεσίας της υπηρεσίας; Η απάντηση είναι ναι.



1.1.3 Φιλαληθείς Ανώνυμοι Μηχανισμοί για τα Παίγνια Χωροθέτησης k-Υπηρεσίων

Στο [21] αποδείχθηκε ότι ο λόγος προσέγγισης του μηχανισμού με τις δύο ακραίες υπηρεσιών είναι βέλτιστος. Ωστόσο, προέκυψε ένα αρνητικό αποτέλεσμα: δεν υπάρχει ντετερμινιστικός ανώνυμος φιλαληθής μηχανισμός για την τοποθεσία k-υπηρεσιών, με $k \geq 3$ και $n \geq k + 1$ παίκτες.

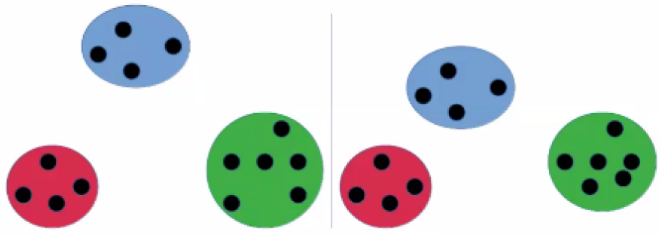
Θεώρημα 1.4. *Για κάθε $k \geq 3$, κάθε ντετερμινιστικός ανώνυμος φιλαληθής μηχανισμός για την τοποθεσία των k-υπηρεσιών με $n \geq k + 1$ παίκτες στην πραγματική γραμμή έχει έναν μη φραγμένο λόγο προσέγγισης.*

Παρόλο που το παραπάνω αποτέλεσμα φαίνεται αποθαρρυντικό, μπορεί να μας ενθαρρύνει να εφαρμόσουμε τα ευρήματα του Παίγνιου Χωροθέτησης k-Υπηρεσίων που έχουν αναπτυχθεί προηγουμένως σε έναν τομέα που δεν επηρεάζεται από το παραπάνω θεώρημα: Η οικογένεια των στιγμιοτυπων με ευστάθεια σε διαταραχές.

1.2 Ευστάθεια σε διαταραχές στην Ομαδοποίηση Δεδομένων

Η πιο κοινή προσέγγιση στον σχεδιασμό και την ανάλυση υπολογιστικών προβλημάτων είναι η εξέταση της χειρότερης περίπτωσης. Ο σχεδιασμός μηχανισμού δεν μπορεί να αποτελεί εξαίρεση σε αυτόν τον κανόνα. Αν και αυτή η μέθοδος παρέχει την πλέον ολοκληρωμένη μέτρηση της δυσκολίας ενός προβλήματος, μας περιορίζει να χρησιμοποιούμε τον ίδιο αλγόριθμο, ακόμα κι αν ενδιαφερόμαστε μόνο για ”ειδικές” περιπτώσεις του προβλήματος που μπορούν να επιλυθούν με πιο βέλτιστο τρόπο. Δυστυχώς, η πλειονότητα των προβλημάτων λήψης αποφάσεων και βελτιστοποίησης με κάποια μορφή πρακτικής χρήσης, συνήθως εμπίπτει στην κατηγορία των προβλημάτων NP-Hard. Ωστόσο, το γεγονός ότι αυτά τα προβλήματα έχουν πρακτική χρήση μπορεί να μας βοηθήσει στη συσχέτισή τους με τις ”πραγματικές” περιπτώσεις, πράγμα που σημαίνει ότι μπορούμε να εκμεταλλευτούμε τις ιδιότητες αυτών των περιπτώσεων.

Οι Bilu και Linial [13] ήταν οι πρώτοι που πρότειναν μια προσέγγιση με στόχο να εκμεταλλευτούν αυτήν την δομή. Συγκεκριμένα, εισήγαγαν τον όρο της ευστάθειας και υποστήριξαν ότι οι περιπτώσεις που έχουν πρακτική εφαρμογή θα έπρεπε να είναι ευσταθείς ως προς μικρές διαταραχές στο μετρικό



χώρο. Ένα πρόβλημα που έχει "πραγματικές" περιπτώσεις με πολύ ενδιαφέρουσες ιδιότητες είναι το πρόβλημα της ομαδοποίησης δεδομένων.

Ορισμός 1.1. *(Πρόβλημα Ομαδοποίσης Δεδομένων) Μια περίπτωση ενός προβλήματος ομαδοποίησης δεδομένων αποτελείται από ένα tuple ((X, d), H, k) ενός μετρικού χώρου (X, d), μίας συνάρτησης στόχου H και ενός ακέραιου αριθμού k > 1. Η συνάρτηση H, δεδομένης μιας διαίρεσης του X σε k σύνολα* $C_1, ..., C_k$ *και μιας μετρικής d στο X, επιστρέφει έναν μη αρνητικό πραγματικό αριθμό, που ονομάζουμε κόστος της διαίρεσης.*

Στόχος μας είναι να ελαχιστοποιήσουμε μια συνάρτηση κόστους που εξαρτάται από το κόστος κάθε σημείου. Οι πιο ενδιαφέροντες στόχοι ομαδοποίησης δεδομένων είναι οι *k-means*, *k-median*, και *k-center*. Αυτοί οι στόχοι καθορίζονται ως εξής. Δεδομένου μίας ομαδοποίησης $C_1, ..., C_k$, ο στόχος είναι ίσος με το ελάχιστο ανάμεσα σε όλες τις επιλογές κέντρων $c_1 \in C_1, ..., c_k \in C_k$ των ακόλουθων συναρτήσεων:

$$H_{means}(C_1, ..., C_k; d) = \sum_{i=1}^{k} \sum_{u \in C_i} = d(u, c_i)^2$$

$$H_{median}(C_1, ..., C_k; d) = \sum_{i=1}^{k} \sum_{u \in C_i} = d(u, c_i)$$

$$H_{center}(C_1, ..., C_k; d) = max_{i \in 1,...,k}\{max_{u \in C_i}\{d(u, c_i)\}\}$$

Ένας τρόπος για να περιγράψουμε την ιδιότητα της ευστάθειας σε ένα στιγμιότυπο είναι να ορίσουμε μια ποσότητα γ, η οποία καθορίζει πόσο μπορούν να αποκλίνουν κοντινά δεδομένα σημεία δεδομένων, διατηρώντας την ίδια βέλτιστη ομαδοποίηση.

Ορισμός 1.2. *(γ-διαταραχή). Δεδομένου ενός μετρικού χώρου (S, d) και* $\gamma \geq 1$*, λέμε ότι μια συνάρτηση* $d' : S \times S \to \mathbb{R}_{>0}$ *είναι γ-διαταραχή της d, αν για οποιαδήποτεx,y ∈ S, ισχύει:*

$$d(x, y)/\gamma \leq d'(x, y) \leq d(x, y)$$

Ορισμός 1.3. *(γ-ευστάθεια). Υποθέτουμε ότι έχουμε μια περίπτωση ομαδοποίησης που αποτελείται από n σημεία δεδομένων που βρίσκονται σε ένα μετρικό χώρο (S, d) και μια συνάρτηση στόχου Φ που επιθυμούμε να βελτιστοποιήσουμε. Λέμε ότι η ομαδοποίηση δεδομένων είναι γ-ευσταθής σε διαταραχές για τον στόχο Φ, αν για κάθε d', που είναι γ-διαταραχή της d, η (μόνη) βέλτιστη ομαδοποίηση δεδομένων του (S, d') υπό τον στόχο Φ είναι ταυτόσημη, ως διαίρεση των σημείων σε υποσύνολα, με τη βέλτιστη ομαδοποίηση δεδομένωνη του (S, d) υπό τον στόχο Φ.*

Αναπτύχθηκαν τρεις βασικες ιδιότητες από τους παραπάνω ορισμούς, τα οποία χρησιμοποιήθηκαν εκτενώς από τους επόμενους αλγόριθμους:

Ιδιότητα 1.1. *(γ-center proximity) Έστω* $p \in S$ *ένα τυχαίο σημείο, έστω* c_i^* *το κέντρο στο οποίο ανατίθεται το p στη βέλτιστη ομαδοποίηση και έστω* $c_j \neq c_i$ *να είναι οποιοσδήποτε άλλο κέντρο στη βέλτιστη* *ομαδοποίηση. Λέμε ότι μια περίπτωση ομαδοποίησης ικανοποιεί την ιδιότητα γ-center proximity αν για κάθε p ισχύει:*

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

Ισχύει ότι αν ενα στιγμιότυπο ομαδοποίησης δεδομένων ικανοποιεί την ιδιότητα γ-ευστάθειας, τότε ικανοποιεί και την ιδιότητα γ-center proximity

Ιδιότητα 1.2. *(weak γ-center proximity) Έστω p ∈ S ένα τυχαίο σημείο, έστω c_i^* το κέντρο στο οποίο ανατίθεται το p στη βέλτιστη ομαδοποίηση και έστω $c_j \neq c_i$ να είναι οποιοσδήποτε άλλο κέντρο στη βέλτιστη ομαδοποίηση . Λέμε ότι μια περίπτωση ομαδοποίησης ικανοποιεί την ιδιότητα αδύναμης γ-κέντρο κοντινότητας αν για κάθε p ισχύει:*

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

Ισχύει ότι αν ενα στιγμιότυπο ομαδοποίησης δεδομένων ικανοποιεί την ιδιότητα γ-ευστάθειας, τότε ικανοποιεί και την ιδιότητα weak γ-center proximity

Ιδιότητα 1.3. *(Cluster-Separation Property) Έστω $(C_1, ..., C_k)$ η βέλτιστη ομαδοποίηση ενός γ-ευσταθούς στιγμιότυπου με γ ≥ 2. Έστω $x_i, x_i' \in C_k$ και $x_j \in C_{k'}$, με i ≠ j, τότε:*

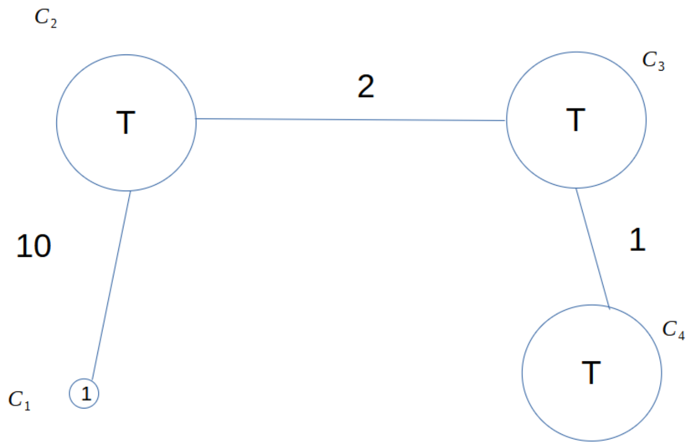
$$d(x_i, x_j) > \frac{(\gamma - 1)^2}{2\gamma} d(x_i, x_{i'})$$

Το Single-link++ είναι ένας αλγόριθμος ομαδοποίησης δεδομένων που εφαρμόζεται σε περιπτώσεις γ-ευσταθών στιγμιότυπων για τον στόχο H_{median}, ο οποίος αναπτύχθηκε από έναν απλούστερο αλγόριθμο που ονομάζεται Single-link ομαδοποίηση και ανακτά τη βέλτιστη ομαδοποίηση σε πολυωνυμικό χρόνο. Ο αλγόριθμος της Single-link ομαδοποίησης είναι γνωστός αλγόριθμος ομαδοποίησης δεδομένων. Η ιδέα είναι να σκεφτούμε τον μετρικό χώρο εισόδου (X, d) ως έναν πλήρη γράφο, με κορυφές X και βάρη ακμών που δίνονται από το d. Ο αλγόριθμος εκτελεί τον αλγόριθμο ελάχιστου συνδεδεμένου δέντρου Kruskal, μέχρι να σχηματιστούν k συνδεδεμένα σύνολα, όπου k είναι ο επιθυμητός αριθμός ομάδων, πράγμα που σημαίνει ότι παραλείπουμε τις τελευταίες k-1 επαναλήψεις του Kruskal. Ωστόσο, υπάρχει ένα προφανές αντί-παράδειγμα. Για να βελτιωθεί αυτό, το Single-link++ δημιουργεί έναν πλήρη γράφο με κορυφές που δίνονται από το X και βάρος ακμής που δίνεται από το d και στη συνέχεια εκτελεί τον αλγόριθμο του Kruskal μέχρι την ολοκλήρωση για να υπολογίσει το ελάχιστο συνδεδεμένο δέντρο T του πλήρους γράφου που προκαλείται από (X, d). Το τελικό βήμα είναι να υπολογίσει ανάμεσα σε όλα τα $\binom{n-1}{k-1}$ υποσύνολα των k-1 ακμών του T και τις παραγώμενες k-ομάδες που προκαλούν, αυτή με την ελάχιστη τιμή της συνάρτησης αντικειμένου H_{median}. Στη συνέχεια αποδεικνύεται το ακόλουθο λήμμα:

Λήμμα 1.1. *Το Single-link++ ανακαλύπτει τη βέλτιστη λύση ενός προβλήματος k-median (X, d) αν και μόνον αν κάθε βέλτιστη ομάδα C_i^* προκαλεί ένα συνδεδεμένο υπογράφημα του ελάχιστου συνδεδεμένου δέντρου.*

Αυτό είναι προφανές αν απεικονίσουμε κάθε συνδεδεμένο στοιχείο που απομένει από τις πρώτες k-1 επαναλήψεις του Kruskal ως μια ανεξάρτητη ομάδα. Αυτό οδηγεί στο ακόλουθο αποτέλεσμα:

Θεώρημα 1.5. *Σε κάθε k-median στιγμίοτυπο με 2-ευστάθεια, ο αλγόριθμος Single-link++ ανακαλύπτει τη βέλτιστη λύση (σε πολυωνυμικό χρόνο).*



Αντιπαράδειγμα του Single Clustering, k = 3

1.3 Ευστάθεια σε διαταραχές στα Παίγνια Χωροθέτησης Υπηρεσιών

Είναι εύκολο να δούμε ότι αν στρέψουμε την προσοχή μας σε πιο πρακτικά παραδείγματα, ακόμα και το πρόβλημα ομαδοποίησης δεδομένων, το οποίο είναι NP-hard μπορεί να αντιμετωπιστεί με ένα απλό και αποτελεσματικό αλγόριθμο. Αυτό μπορεί να μας οδηγησει στην διερεύνηση των συμπεριφορών αυτών των στιγμιοτύπων στο πρόβλημα χωροθέτησης υπηρεσιών, το οποίο σχετίζεται στενά με το ομαδοποίησης δεδομένων. Η επιπλέον πολυπλοκότητα αυτού του προβλήματος είναι ότι δεν μπορούμε πλέον να εμπιστευτούμε πλήρως την είσοδό μας, καθώς οι παίκτες είναι στρατηγικοί και πρέπει να βρούμε τρόπους για να τους αποτρέψουμε από το να δηλώσουν ψευδή τοποθεσία.

Μπορούμε εύκολα να ορίσουμε το γ-διαταραχή και τη γ-ευστάθεια για αυτό το πρόβλημα:

Ορισμός 1.4. *(Γραμμική γ-διαταραχή) Έστω $\vec{x} = (x_1, .., x_n)$ ένα προφίλ στιγμιοτύπου. Ένα προφίλ στιγμιοτύπου$\vec{x} = (x_1, .., x_n)$ είναι γ-διαταραχή του $\vec{x}$, για κάποιο $g \geq 1$, αν $x'_1 = x_1$ και για κάθε $i \in [n-1]$, ισχύει ότι*

$$d(x_i, x_{i+1})/\gamma \leq d(x'_i, x'_{i+1}) \leq d(x_i, x_{i+1})$$

Ορισμός 1.5. *(Γραμμική γ-ευστάθεια). Ένα πρόβλημα χωροθέτησης κ-υπηρεσιων είναι γ-ευσταθές, αν το $\vec{x}$ έχει μια μοναδική βέλτιστη ομαδοποίηση $(C_1, ..., C_k)$ και κάθε γ-διαταραχή$\vec{x}'$ του $\vec{x}$ έχει την ίδια μοναδική βέλτιστη ομαδοποίηση $(C_1, ..., C_k)$.*

Και το ίδιο συμβαίνει με τις τρεις βασικές ιδιότητες της ευστάθειας: γραμμική γ-center proximity, γραμμική weak γ-center proximity και Cluster-Separation Property.

Στην μελέτη [23], παρουσιάστηκε ένας ντετερμινιστικός, φιλαληθής μηχανισμός που επιτυγχάνει τη βέλτιστη ομαδοποίηση για περιπτώσεις που είναι 2 + √3-ευσταθείς. Δυστυχώς, αυτός μπορεί να εφαρμοστεί μόνο σε περιπτώσεις στις οποίες η βέλτιστη ομαδοποίηση δεν περιλαμβάνει μια ομάδα αποτελούμενη από ένα παίκτη, διαφορετικά η φιλαλήθεια αποτυγχάνει.

Mechanism 1 OPTIMAL : Deterministic mechanism on $2+\sqrt{3}$-stable instances without Singleton Deviations.

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ Compute the optimal clustering $(C_1, ..., C_k)$. Let c_i be the left median point of each cluster C_i.
if ($\exists i \in [k]$ with $|C_i| = 1$) or ($\exists i \in [k-1]$ with $max\{D(C_i), D(C_{i+1})\} \leq d(C_i, C_{i+1})\}$) **then**
  Output: ”FACILITIES ARE NOT ALLOCATED”.
else
  Output: The k-facility allocation $(c_1, ..., c_k)$
end if

Στην ίδια μελέτη, παρουσιάστηκε ένας ντετερμινιστικος, φιλαληθής μηχανισμός που αποκτά τη βέλτιστη ομαδοποίηση για 5-ευσταθή στιγμιότυπα. Αυτός ο μηχανισμός αυξάνει τη ευστάθεια των στιγμιοτύπων στα οποία εφαρμόζεται, έτσι ώστε να μπορεί να αντιμετωπίσει τις ομάδες μοναδικών παικτών στη βέλτιστη ομαδοποίηση του στιγμιότυπου.

Mechanism 2 ALMOSTRIGHTMOST : Deterministic Mechanism Resistant to Singleton Deviations 5-stable instances.

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ Find the optimal clustering $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$.
if there are two consecutive clusters C_i and C_{i+1} with $max\{D(C_i), D(C_{i+1})\} \geq d(C_i, C_{i+1})\}$) **then**
  Output: ”FACILITIES ARE NOT ALLOCATED”.
  for i $\in$ 1, . . . ,k **do**
    if $|C_i| > 1$ **then**
      Allocate a facility to the location of the second rightmost agent of C_i, i.e., $c_i \leftarrow x_{i,r-1}$.
    else
      Allocate a facility to the single agent location of C_i : $c_i \leftarrow x_{i,l}$
    end if
  end for
end if
Output: The k-facility allocation $\vec{c} = (c_1, ..., c_k)$

1.4 Ενισχυμένοι απο Μάθηση Μηχανισμοι πάνω στα Παίγνια Χωροθέτησης Υπηρεσίων

Η έννοια της ευστάθειας δεν είναι η μοναδική προσέγγιση στον σχεδιασμό μηχανισμού, την οποία μπορούμε να χρησιμοποιήσουμε για να αποφύγουμε τη μέθοδο της ανάλυσης χειρότερης περίπτωσης. Αν και η χρήση της ανάλυσης χειρότερης περίπτωσης παρέχει μια συγκεκριμένη ανθεκτικότητα στο αποτέλεσμα του αλγορίθμου μας, μας αφαιρεί την ευελιξία να μελετήσουμε παραδείγματα που είναι σε θέση να παράγουν ένα πιο κοντινό στον ”πραγματικό κόσμο” μοντέλο για τα προβλήματά μας. Αυτού του είδους τα προβλήματα διαθέτουν συγκεκριμένες ιδιότητες που οι αλγόριθμοι μηχανικής μάθησης μπορούν να εκμεταλλευτούν για να παράγουν χρήσιμες ”προβλέψεις”. Στο [1] αυτή η γραμμή σκέψης χρησιμοποιήθηκε για την ανάπτυξη των Ενισχυμένων από Μάθηση Μηχανισμών για το πρόβλημα των Παίγνιων Χωροθέτησης Υπηρεσιών.

Για αυτόν τον νέο τύπο μηχανισμού, πρέπει να εισάγουμε κάποιες νέες μετρικές: Τη Συνέπεια και την Ανθεκτικότητα. Εάν η πρόβλεψη είναι ακριβής, ορίζουμε τον λόγο προσέγγισης ως τη συνέπεια του μηχανισμού. Εάν η πρόβλεψη είναι αυθαίρετη, ορίζουμε τον λόγο προσέγγισης ως ανθεκτικότητα.

Στο πλαίσιο του σχεδιασμού μηχανισμού ενισχυμένου με μάθηση, πριν ζητήσει το σύνολο των προτιμώμενων τοποθεσιών P από τους παίκτες, ο σχεδιαστής λαμβάνει μια πρόβλεψη $\hat{o}$ σχετικά με την βέλτιστη τοποθεσία υπηρεσίας o(P). Ο σχεδιαστής μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες για να επιλέξει τους κανόνες του μηχανισμού, αλλά, όπως και στο κλασσικό πρόβλημα χωροθέτησης υπηρεσιών, ο μηχανισμός που συμβολίζεται ως $M(P, \hat{o})$, πρέπει να είναι φιλαλήθης. Ουσιαστικά, αν υπάρχουν πολλοί φιλαλήθεις μηχανισμοί από τους οποίους μπορεί να επιλέξει ο σχεδιαστής, η πρόβλεψη μπορεί να καθοδηγήσει την επιλογή τους, με στόχο την επίτευξη βελτιωμένων εγγυήσεων εάν η πρόβλεψη είναι ακριβής (συνέπεια), διατηρώντας παράλληλα βέλτιστες εγγυήσεις σε περίπτωση χειρότερης περίπτωσης (ανθεκτικότητα).

Ορισμός 1.6. *(Συνέπεια α) Δεδομένη κάποια συναρτηση κοινωνικού κόστους C (δηλαδή MC(·),SC(·)), ένας μηχανισμός είναι α-συνεπής αν επιτυγχάνει μία α-προσέγγιση όταν η πρόβλεψη είναι σωστή ($\hat{o} = o(P)$), δηλαδή*

$$max_P[\frac{C(M(P, o(P)), P))}{C(o(P), P)}] \leq \alpha$$

.

Ορισμός 1.7. *(Ανθεκτικότητα β) Ο μηχανισμός μας είναι β-ανθεκτικός αν επιτυγχάνει έναν β-προσέγγιση ακόμα και όταν η πρόβλεψη είναι αυθαίρετα λανθασμένη, δηλαδή*

$$max_{P,\hat{o}}[\frac{C(M(P, \hat{o}), P)}{C(o(P), P)}] \leq \beta$$

.

Θα ασχοληθούμε με την περίπτωση μίας διάστασης, με την εισαγωγή του μηχανισμού MinMaxP. Αυτός ο μηχανισμός χρησιμοποιεί την πρόβλεψη $\hat{o}$ως την προεπιλεγμένη επιλογή τοποθεσίας υπηρεσία, εκτός αν η πρόβλεψη βρίσκεται ”αριστερά” όλων των σημείων στο P ή ”δεξιά” όλων των σημείων στο P. Στην πρώτη περίπτωση, η υπηρεσία τοποθετείται στο αριστερότερο σημείο του P, ενώ στη δεύτερη περίπτωση, τοποθετείται στο δεξιότερο σημείο του P.

Αυτός ο μηχανισμός επιτυγχάνει 1-συνέπεια και 2-ανθεκτικότητα, που αποτελεί την βέλτιστη ισορροπία ανάμεσα σε αυτά τα δύο χαρακτηριστικά.

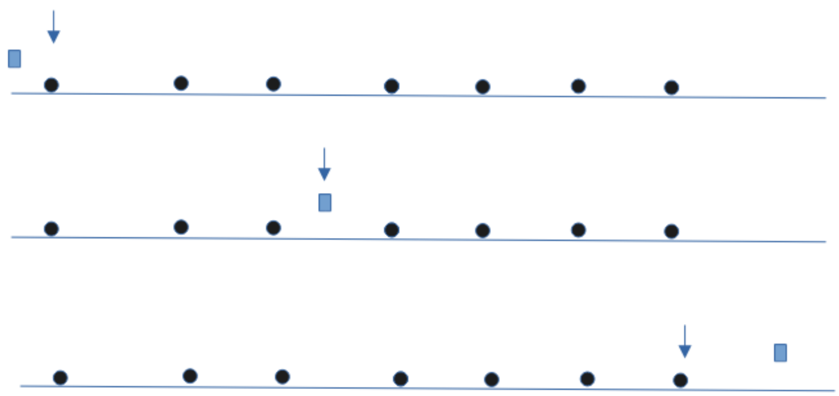
1.5 Σχεδιασμός Μηχανισμού με Προβλέψεις για Σταθερά Στιγμιότυπα των παίγνιων Χωροθέτησης Υπηρεσιών

Στόχος μας είναι να ενσωματώσουμε όλα τα προηγούμενα συστατικά που παρουσιάστηκαν (Παίγνια χωροθέτησης υπηρεσιών, γ-ευσταθείς στιγμιότυπα και σχεδιασμό μηχανισμού ενισχυμένου με μάθηση) και να προσπαθήσουμε να δημιουργήσουμε έναν κομψό μηχανισμό που να τα συνδυάζει.

Η εργασία μας είναι να εντοπίσουμε k υπηρεσίες στη γραμμή. Θα δημιουργήσουμε τον μηχανισμό $M(\vec{x}, \hat{o})$, ο οποίος λαμβάνει το ζεύγος $(\vec{x}, \hat{o})$, ως είσοδο. Το στιγμιότυπο $\vec{x}$ είναι ένα διάνυσμα που αποτελείται από τη θέση κάθε παίκτη στη γραμμή και το $\hat{o}$ είναι ένα διάνυσμα με τις προβλεπόμενες τοποθεσίες των υπηρεσιών, που παράγονται από ένα εξωτερικό σύστημα. Η εργασία μας μπορεί να φαίνεται πιο εύκολη, αφού ήδη έχουμε μηχανισμούς που εφαρμόζονται με επιτυχία σε γ-ευσταθείς στιγμιότυπα και με την προσθήκη εξωτερικών προβλέψεων, φαίνεται λογικό να ελπίζουμε για ακόμα καλύτερο αποτέλεσμα. Ωστόσο, προκύπτει πρόβλημα από το γεγονός ότι οι παίκτες μπορούν τώρα να εκμεταλλευτούν τον μηχανισμό μέσω της τοποθεσίας των προβλέψεων, προσθέτοντας ένα ακόμα επίπεδο πολυπλοκότητας.

Ο μηχανισμός που προτείνουμε είναι ένας γενικευμένος MINMAXP σε k υπηρεσίες. Ωστόσο, στο MINMAXP, έπρεπε να αντιμετωπίσουμε μόνο μια ομάδα και μόνο μια πρόβλεψη, ενώ στη γενική περίπτωση, έπρεπε να αντιστοιχίσουμε k προβλέψεις σε k όμαδα με φιλαληθή τρόπο. Αυτή είναι η νέα μορφή ψέματος που ένας παίκτης μπορεί να χρησιμοποιήσει για να κερδίσει. Επιλέγουμε να αντιστοιχίσουμε την i-στη πρόβλεψη στην i-στη ομάδα. Για να αποφύγουμε οποιοδήποτε πρόβλημα με τη φιλαλήθεια του μηχανισμού, χρειαζόμαστε τουλάχιστον 5 ευστάθεια και τον αποκλεισμό περιπτώσεων ομάδων ενός μοναδικού παίκτη στη βέλτιστη ομαδοποίηση τους.

Ο μηχανισμός δέχεται την είσοδο, εκτελεί έλεγχο για την cluster-separation property και ελέγχει αν υπάρχει μια ομάδα ενός μοναδικού παίκτη. Εάν το παράδειγμά περνά και τους δύο αυτούς ελέγχους, εξάγουμε k υπηρεσίες. Για να λειτουργήσει ο μηχανισμός μας, χρειαζόμαστε το στιγμιότυπο μας να έχει τουλάχιστον 5 ευστάθεια και η βέλτιστη ομαδοποίηση του να μην περιλαμβάνει ομάδες μοναδικών παικτών. Εάν δεν παραβιάζεται cluster-separation property και δεν υπάρχει ομάδα μοναδικού παίκτη στη βέλτιστη ομαδοποίηση μας, μπορούμε να αντιστοιχίσουμε την i-στη πρόβλεψη $\hat{o}_i$ στην i-στη ομάδα C_i. Δεδομένου ότι το παράδειγμά μας είναι γ-ευσταθές, μπορούμε να θεωρήσουμε κάθε ομάδα ως ένα μοναδικό στιγμιότυπο το οποίο είναι πλήρως διαχωρισμένο από τις υπόλοιπες ομάδες, και να εφαρμόσουμε τον Μηχανισμό MinMaxP [1] σε κάθε ομάδα. Δυστυχώς, πρέπει να συμπεριλάβουμε τους περιορισμούς της απουσίας ομάδων μοναδικού παίκτη στη βέλτιστη ομαδοποίηση, διότι εάν επιτρέψουμε σε έναν παίκτη να αποκλίνει και να δημιουργήσει μια ομάδα μοναδικού παίκτη, χωρίς να διαταράξει τη ευστάθεια του στιγμιότυπου, τότε μπορεί να απομονώσει μια απομακρυσμένη



Οι παίκτες αναπαρίστανται από μαύρους κύκλους, η πρόβλεψη αναπαρίσταται από μπλε ορθογώνια, η τοποθεσία της υπηρεσίας αναπαρίσταται από ένα βέλος. Αυτές είναι οι 3 διαφορετικοί τρόποι με τους οποίους μπορούμε να αναθέσουμε μια υπηρεσία στο παράδειγμα. Ο πρώτος δείχνει τι συμβαίνει όταν $\hat{o} < min_i p_i$, ο δεύτερος όταν $\hat{o} \in [min_i p_i, max_i p_i]$ και ο τρίτος όταν $\hat{o} > max_i p_i$

.

Mechanism 3 Generalized MinMaxP $M(\vec{x}, \hat{o})$:Deterministic Mechanism for 5-stable instances with no singleton clusters

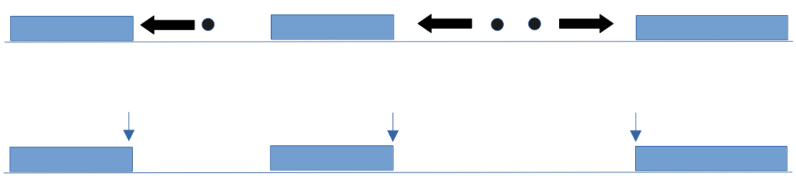
Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ and k-vector of predictions on facilities locations $\hat{o}$ Find the optimal clustering $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$.
for i $\in$ 1, . . . ,k **do**
  Match $\hat{o}_i$ to i-th cluster, C_i.
  if $\hat{o}_i \in [x_{i,l}, x_{i,r}]$ **then**
    Allocate a facility to $\hat{o}_i$.
  end if
  if $\hat{o}_i < x_{i,l}$ **then**
    Allocate a facility to $x_{i,l}$
  end if
  if $\hat{o}_i > x_{i,r}$ **then**
    Allocate a facility to $x_{i,r}$
  end if
end for
Output: The k-facility allocation that was previously defined.

πρόβλεψη και να αλλάξει την αντιστοίχηση των προβλέψεων-ομάδων προς όφελός του.

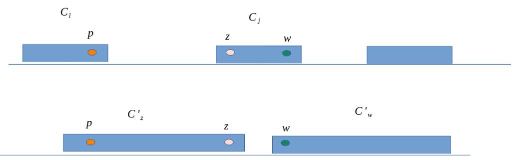
Ο μηχανισμός είναι 1-συνεπής και 2-ανθεκτικός για τον στόχο του Μέγιστου Κόστους, 1-συνεπής και (n-1)-ανθεκτικός για τον στόχο του Κοινωνικού Κόστους. Η φιλαλήθεια του διασφαλίζεται από το γεγονός ότι απαριθμούμε παρακάτω τις αποκλίσεις που μπορεί να κάνει ένας παίκτης:

1. Διαίρεση \ Συγχώνευση που διατηρεί την αρχική αντιστοίχιση των προβλέψεων στις ομάδες
2. Διαίρεση \ Συγχώνευση που δεν διατηρεί την αντιστοίχιση των προβλέψεων στις ομάδες.
3. Απόκλιση του παίκτη που αλλάζει μόνο το μήκος της δικής του ομάδας.

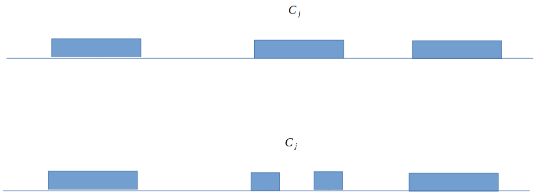
Οποιαδήποτε διαίρεση είτε παραβιάζει την Cluster-Separation Property είτε δεν μπορεί να παράγει μια καλύτερη ομαδοποίηση από την αρχική. Ο μόνος τρόπος για έναν παίκτη να πραγματοποιήσει κερδοφόρες συγχωνεύσεις είναι να παραβιάσει την αρίθμηση των ομάδων της συγκεκριμένης περίπτωσης. Όλες οι συγχωνεύσεις δημιουργούν περιπτώσεις που "απαγορεύονται". Η πρώτη περίπτωση είναι μια περίπτωση με ομάδα μοναδικού παίκτη, η οποία αποτυγχάνει το τεστ της μοναδικής ομάδας του μηχανισμού. Οι δεύτερη και τρίτη περιπτώσεις περιλαμβάνουν διαιρέσεις που είτε δεν είναι



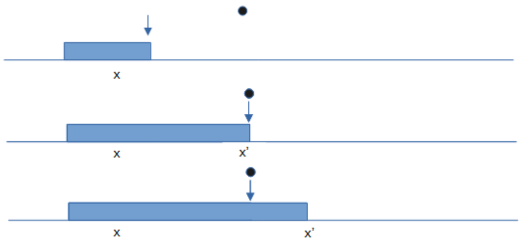
Τα μπλε ορθογώνια είναι ομάδες, οι μαύροι κύκλοι είναι προβλέψεις, τα μπλε βέλη είναι οι τοποθετημένες υπηρεσίες, και κάθε i-στη πρόβλεψη αντιστοιχεί στην i-στη ομάδα.



Στην 1η περίπτωση διαίρεσης, μπορούμε να δούμε ότι η Cluster-Separation Property παραβιάζεται.



Στην 2η περίπτωση διαίρεσης, βλέπουμε ότι είναι υποβέλτιστο να ”σπαταλήσουμε” δύο υπηρεσίες σε παίκτες C_j καθώς για τη βέλτιστη ομαδοποίηση έχει αποδειχθεί ότι χρειαζόμαστε μόνο μία.

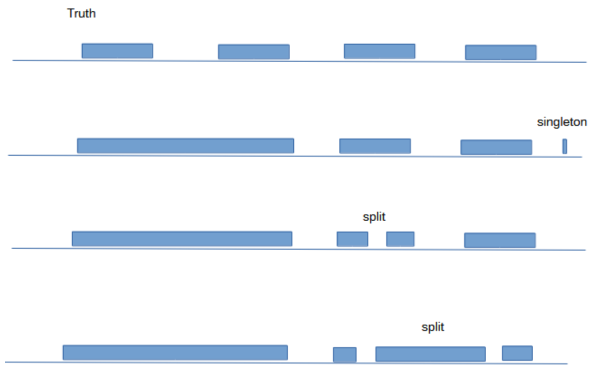


εφικτές είτε αποτυγχάνουν στην ιδιότητα του διαχωρισμού των ομάδων. Αυτό σημαίνει ότι δεν υπάρχει δυνατότητα κερδοφόρας διαίρεσης. Τέλος, αν η απόκλιση του παίκτη δεν επηρεάζει καμία άλλη ομάδα, τότε προσπαθεί να συμπεριλάβει ή να αποκλείσει μέσα στη δική του ομάδα την πρόβλεψη που αντιστοιχεί στη δική του ομάδα. Χρησιμοποιώντας τον ίδιο σχεδιασμό απόδειξης με τον MINMAXP, καταλήγουμε στο συμπέρασμα ότι δεν μπορεί να κερδίσει λέγοντας ψέματα.

1.6 Συμπεράσματα και Μελλοντικές Προσεγγίσεις

Το αποτέλεσμα της μελέτης [21] οδήγησε τους ερευνητές να επικεντρωθούν σε διάφορους τρόπους προσέγγισης των παίγνιων χωροθέτησης k υπηρεσιών, προσπαθώντας συνεχώς να συνδυάσουν διαφορετικά πεδία στην προσπάθειά τους να κατανοήσουν πλήρως όλους τους περιορισμούς που μπορεί να υπάρχουν. Αν και η ευσταθεια και οι αλγόριθμοι ενίσχυσης πρόβλεψης δεν είναι εντελώς νέες έννοιες, υπάρχει ακόμα δουλειά που πρέπει να γίνει όταν εφαρμόζονται στα Παίγνια χωροθέτησης k υπηρεσιών. Στη διπλωματική μου εργασία, παρουσίασα ένα μηχανισμό που επιτυγχάνει 1-συνέπεια και n-ανθεκτικότητα, αλλά έχει δύο αδυναμίες. Ο μηχανισμός απαιτεί τουλάχιστον 5 ευστάθεια, όπως και ο μηχανισμός ALMOSTRIGHTMOST, και δεν εξετάζει περιπτώσεις που περιέχουν ομάδες με ένα μοναδικό παίκτη, όπως ο μηχανισμός OPTIMAL.

Ένας φυσικός κατεύθυνση θα ήταν να εξετάσουμε εάν μπορούμε να επιτύχουμε χαμηλότερη ευστάθεια ή εάν μπορούμε να βρούμε έναν τρόπο να αντιμετωπίσουμε το πρόβλημα που προκύπτει από την



Ο αριστερότερος παίκτης της τρίτης ομάδας επιθυμεί να ανήκει στη δεύτερη ομάδα ενός τροποποιημένης στιγμιοτύπου. Αυτό μπορεί να το κάνει με τρεις διαφορετικές συγχωνεύσεις. Μπορεί να συγχωνεύσει την πρώτη και τη δεύτερη ομάδα, δημιουργώντας μία ομάδα με ένα παίκτη μόνο στα δεξιά του. Μπορεί να συγχωνεύσει την πρώτη και τη δεύτερη ομάδα, διαιρώντας τη ομάδα του. Μπορεί να συγχωνεύσει την πρώτη και τη δεύτερη ομάδα, διαιρώντας την τέταρτη ομάδα και τη δική του.

ύπαρξη ομάδων με ένα μοναδικό παίκτη. Η ευστάθεια περιορίστηκε σε 5, καθώς βασιστήκαμε σε μεγάλο βαθμό στον μηχανισμό ALMOSTRIGHTMOST. Ωστόσο, ο ALMOSTRIGHTMOST δεν περιορίζεται από μοναδικές ομάδες όπως ο μηχανισμός μας. Επιπλέον, η εισαγωγή προβλέψεων έγινε λόγω της απόστασης της ευστάθειας που υπάρχει μεταξύ του ALMOSTRIGHTMOST και του OPTIMAL, καθώς παραμένει να εξεταστεί εάν μπορούμε να βρούμε έναν μηχανισμό με ευστάθεια < 5, ο οποίος αντιμετωπίζει επίσης ομάδες με ένα μοναδικό παίκτη στο στιγμιότυπο. Τέλος, στη μελέτη [23] το αποτέλεσμα της αδυνατότητας για τα Παίγνια χωροθέτησης υπηρεσιών k με $k \geq 3$ διευρύνθηκε, αποδεικνύοντας ότι δεν υπάρχει κανένας προσδιορισμένος ανώνυμος στρατηγικά ανθεκτικός μηχανισμός για τοποθεσία υπηρεσιών k, με $k \geq 3$, σε περιπτώσεις με ευστάθεια $(2 - \delta)$, με φραγμένο λόγο προσέγγισης για οποιοδήποτε $\delta > 0$. Απομένει να δούμε αν αυτό το φράγμα είναι ακριβές.

Chapter 2

Introduction

One of the most interesting aspects of mathematics is how we can interpret its numerous applications in seemingly non-uniform academic fields. Such an upcoming field is the *Algorithmic Mechanism design*, a combination of Social choice theory, Game theory, Mechanism Design, and Computer Science. Social choice theory is the study of collective decision procedures and mechanisms. It is not a single theory, but a cluster of models and results concerning the aggregation of individual inputs (e.g., votes, preferences, judgments, welfare) into collective outputs (e.g., collective decisions, preferences, judgments, welfare). Mechanism Design is a sub-field of economic theory that is unique within economics in having an engineering perspective. It is interested in designing economic mechanisms, just like computer scientists are interested in designing algorithms, protocols, or systems. *Algorithmic Mechanism design* studies optimization problems where the underlying data — such as the value of a good or the cost of performing a task — is initially unknown to the algorithm designer, and must be implicitly or explicitly elicited from self-interested participants. In algorithm design, most of the time we were not concerned with the validity of our input. In this setting we consider the participants to be rational in their actions and to always look to maximize their gain, which depends on the "game" that they are being played, meaning that they opt to lie. The designer is looking to exploit this and create a set of rules that will lead the players to act in a way that produces the optimal outcome, based on a game-related objective (e.g. maximum revenue, social welfare maximization, etc.). Continuing the correlation between the process of constructing mechanisms and algorithms, we aim to create computationally efficient mechanisms that are not susceptible to a player's deviation from the truth, preserving the valuable property of *strategyproofness*.

As mentioned above, our mechanisms need to be unaffected by a participant's lie or even better to convince all the participants that telling the truth is in their best interest. Some mechanisms use money and payments to enforce such conditions, while in others, payments could be illegal or unethical. In this thesis, we are interested in one of the fundamental problems in the latter category, Facility Location Games. We consider k-Facility Location games, where $k \geq 2$ facilities are placed on the real line based on the preferences of n strategic agents-participants. Such problems are motivated by natural scenarios in Social Choice, where a local authority plans to build a fixed number of public facilities in an area [39]. The choice of the locations is based on the preferences of local people, or agents. Each agent reports his ideal location and the local authority applies a (deterministic or randomized) mechanism that maps the agents' preferences to k facility locations. Each agent strives to reduce its connection cost - his distance from the closest facility and the mechanism designer strives to optimize a certain objective (Social Cost, Maximum Cost, etc.), while also. Since Procaccia and Tennenholtz [43] initiated the research agenda of approximate mechanism design without money, k-Facility Location has served as the benchmark problem in the area and its approximability by deterministic or randomized strategyproof mechanisms has been studied extensively in virtually all possible variants and generalizations. For instance, previous work has considered multiple facilities on the line (see e.g., ([21],[24],[27],[33],[42]) and in general metric spaces ([20],[32]), different objectives (e.g., social cost, maximum cost, the L_2 norm of agent connection costs ([19],[24],[43]), restricted metric spaces more general than the line (cycle, plane, trees, see e.g.,([2],[16],[16],[25],[16]), facilities that serve different purposes (see e.g., [30],[31]), and different notions of private information about the agent preferences that should be declared to the mechanism (see e.g., [15],[18],[36]) and the references therein). Due to the significant research interest in the topic, the fundamental and most basic question of approximating the optimal social cost by strategyproof mechanisms for k-Facility Location on the line has been relatively well understood. For one facility, the mechanism that places the location at the median agent of the instance is both optimal and strategyproof. For two facilities, the placement at the two extreme points of the instance will preserve strategyproofness, while producing the best possible approximation (n - 2). Unfortunately for $k \geq 3$, we are given the negative result of [21], where it stated that there is no anonymous, deterministic, strategyproof mechanism with bounded approximation, for K-Facility Location, with $K \geq 3$. This result led us to acknowledge the limitations of the normal setting of facility location games and turn our interest to instances that are closer to the real world and thus maintain properties that can be exploited by our mechanism. In this thesis, we will focus on this new kind of instance, where not only we will have a distinguishable optimal clustering, but also we will be aided by an external system in our location placing.

The similarities between the *Clustering* of data, simulated by a metric space (X,d), and Facility Location Games are too many to ignore. Both of those problems look to find ways of optimal grouping. In clustering, we may not have to deal with a "lying' metric space, but since it is a heavily researched field, we have found many ways to characterize our input instance and develop algorithms that exploit this characterization. One of these attributes is the so-called *perturbation stable instances*, which are instances resembling real-world data, in the sense that we are implicitly assuming that interesting structure exists in the data and even small perturbations can not alter the structure of the input. Perturbation stability was introduced by Bilu and Linial in [13] and Awasthi, Blum and Sheffet in [6] (and has motivated a significant volume of follow-up work since then, see e.g., ([4],[8],[10],[45] and the references therein) in an attempt to obtain a theoretical understanding of the superior practical performance of relatively simple clustering algorithms for well known NP-hard clustering problems (such as k-Facility Location in general metric spaces). Intuitively, the optimal clusters of a γ-stable instance are somehow well-separated, and thus simple clustering algorithms, such as single-clustering, become viable choices in our attempt to produce optimal clustering in polynomial time, for an appropriate value of γ. The simplicity of these algorithms is enhanced by the stability properties that have been proven, such as γ-center proximity, weak γ-center proximity, and the Cluster-Separation Property. These properties define bounds between inter-cluster and intra-cluster distances between data points, making it easier to distinguish clusters on an instance. A natural extension would be to apply these properties on the field of Facility Location Games and investigate the optimality and strategyproofness of our mechanism on γ-stable instances. All the results in the previous section are based on the worst-case scenario, which is bounded by [21]. However, if we utilize the notion of stability in our instances, we can come up with some interesting results. In [23], the study of efficient (wrt. their approximation ratio for the social cost) strategyproof mechanisms was initiated for the large and natural class of γ-stable instances of k-Facility Location on the line. The existence of deterministic (resp. randomized) strategyproof mechanisms with a bounded (resp. constant) approximation ratio for 5-stable instances and any number of facilities k $\geq$ 2 was exhibited. Moreover, it was shown that the optimal solution is strategyproof for (2 + $\sqrt{3}$)-stable instances if the optimal clustering does not include any singleton clusters (which is likely to be the case in virtually all practical applications). Furthermore, the impossibility result of Fotakis and Tzamos [21] was strengthened, so that it applies to γ-stable instances, with $\gamma < \sqrt{2}$. Specifically, it was shown that for any k $\geq$ 3 and any $\delta > 0$, there do not exist any deterministic anonymous strategyproof mechanisms for k-Facility Location on ($\sqrt{2}$

$-\ \delta$)-stable instances with bounded (in terms of n and k) approximation ratio. It remains to be seen, if the ($\sqrt{2} - \delta$) bound for γ is tight and if there is a deterministic, strategyproof mechanism for γ-stable instances, with $\gamma < 5$, who can also deal with singleton clusters in the instance. This thesis will focus on finding optimal clustering on γ-stable instances while receiving some "extra' information from an external source.

The notion of perturbation stability on instances is only one of the many different approaches we can take when we look further from the worst-case analysis framework. In recent years, there has been a significant amount of research applied to algorithms enhanced by machine learning creating the new field of "algorithms with predictions", thus giving a real motivation to determine whether the contribution of predictions to known Facility Location Games mechanism or the development of new learning-augmented Facility Location Games mechanisms can produce interesting results. New metrics - *Consistency* and *Robustness* were introduced in [38] as the standard measures in algorithms with predictions. They expand upon the notion of approximation ratio: Consistency is the approximation ratio of the mechanism, when the predictions coincide with the optimal solution and Robustness is the approximation ratio of the mechanism when the predictions are arbitrarily wrong. In [1], there was an initial attempt to explore these new mechanisms. They developed a mechanism for one facility located on the line, achieving 1-consistency and 1-robustness for the social cost objective and 1-consistency and (1 + $\sqrt{2}$)-robustness for the maximum cost objective, while also proving that the 1-consistency and (1 + $\sqrt{2}$)-robustness trade-off is optimal. An open problem is to find out the optimal trade-off between consistency and robustness in the general case of k facilities for γ-stable instances, which is also the focus of this thesis. We will present a generalized version of the one-facility learning-augmented mechanism, for the k-facility variation of at least 5-perturbation stable instance, that achieves 1-consistency for both MAXIMUM COST and SOCIAL COST OBJECTIVE, 2-robustness for MAXIMUM COST, (n-1)-robustness for SOCIAL COST OBJECTIVE, while also observing the limitations that this kind of generalization meets.

2.1 Facility Location Games

Our game consists of n strategic agents and k facilities. The agents are placed in the metric space metric space (X, d), where d: X × X → 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) $\geq$ d(x, y) + d(x, z) for all x, y, z $\in$ X (triangle inequality). Each agent $i \in N$ has a location x_i, which is his *private information*, and on the same metric space, we intend to place the facilities. We refer to the collection $\vec{x} = (x_1, ..., x_n)$ as the *location profile* or as the *instance*.The *connection cost* of agent i, denoted as $cost(x_i, \vec{y})$, is $min_{1 \leq j \leq k} d(x_i, y_j)$, which is the distance between the agent's location and the closest facility location. Our task is to place k facilities on the metric space while trying to minimize a *cost function*, which depends on the agents' connection costs. Each agent tries to minimize his connection cost.

A deterministic Mechanism M maps $\vec{x}$ to a k-tuple $(y_1, ..., y_k) \in X^k$ of facility locations. We let $M(\vec{x})$ denote the outcome of Mechanism M. A randomized Mechanism M maps $\vec{x}$ to a probability distribution over a k-tuples $(y_1, ..., y_k) \in X^k$ of facility locations. Two of the most basic cost functions are the Social Cost function, the sum of all the agents' connection costs, and the Maximum Cost function, the maximum over all the agents' connection costs. The *Social cost* of a facility locations profile $(y_1, ..., y_k) \in X^k$ is $SC(\vec{x}, \vec{y}) = \sum_{i=1}^{n} cost(x_i, \vec{y})$. The *Maximum cost* of a facility locations profile $(y_1, ..., y_k) \in X^k$ is $MC(\vec{x}, \vec{y}) = max_{i \in N}[cost(x_i, \vec{y})]$

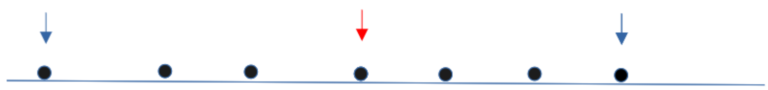
A mechanism M for k-Facility Location achieves an approximation ratio of $\rho \geq 1$ for the objective of Social cost (respectively for Maximum cost), for all instances $\vec{x}$, $SC(\vec{x}, M(\vec{x})) \leq \rho SC^*(\vec{x})$, (respectively for maximum cost $MC(\vec{x}, M(\vec{x})) \leq \rho MC^*(\vec{x})$). A mechanism M is strategyproof if no agent can benefit from misreporting his location. Formally, for all instances $\vec{x}$, every agent i, and all locations $y \in X$, $cost(x_i, M(\vec{x})) \leq cost(x_i, M(\vec{x}_{-i}, y))$

2.1.1 Mechanism for one facility on the line

Our first task is to find a strategyproof Mechanism that minimizes the social cost for the Facility Location Game of n agents and one facility. The solution is pretty straightforward. We can choose the median location in $\vec{x}$ - $med(\vec{x})$. If we choose any agent on the left of $med(\vec{x})$, then the Social Cost increases, since it is further away from at least k + 1 agents and closer to at most k agents. The same holds for any agent to the right of the median. Assume that n is even, n = 2k, then any point in $[x_k, x_{k+1}]$ produces the optimal social cost, for the same reason as in the case of n = 2k + 1. So, $med(\vec{x})$ is optimal. It is also strategyproof since the agent can only move the facility further away with his lie.

Theorem 2.1. *$M(\vec{x}) = med(\vec{x})$ is a strategyproof optimal mechanism for social cost*

Red facility is the optimal selection



Our second task is to find an optimal and strategyproof mechanism for the maximum cost objective in the one facility setting. The facility placement that minimizes our objective, is the location $cen(\vec{x})$. Unfortunately, this placement is not strategyproof, since the any agent can lie and change the instance's length, until $cen(\vec{x'})$ lands on this agent's location. Procaccia and Tennenholtz [43] proposed the following group strategyproof 2-approximation mechanism for the maximum cost, $M(\vec{x}) = lt(\vec{x})$.

Theorem 2.2. *$M(\vec{x}) = lt(\vec{x})$ is a strategyproof 2-approximation mechanism for the maximum cost*

A natural question that arises is, if this the best we can do for the maximum cost objective. Procaccia and Tennenholtz [43], indeed proved that $lt(\vec{x})$ is the best possible mechanism for the one facility setting, wrt the maximum cost function. And indeed the leftmost agent choice is the best that we can do if we want a deterministic strategyproof mechanism

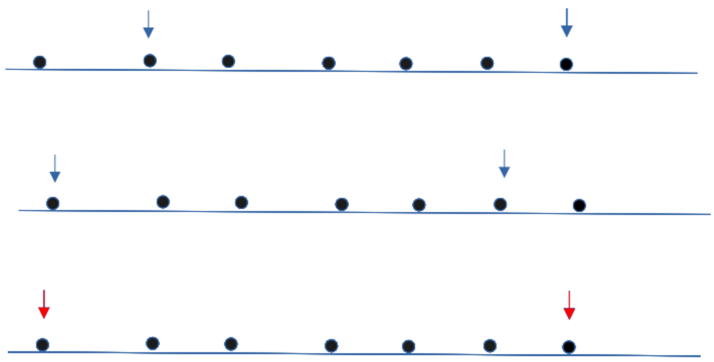
Theorem 2.3. *Let $N = \{1, ..., n\}$, $n \geq 2$. Any deterministic strategyproof mechanism M: $\mathbb{R}^n \rightarrow R$ has an approximation ratio of at least 2 for the maximum cost.*

2.1.2 Mechanisms for two facilities on the line

We can now deal with the extension of the previous setting, locating two facilities, instead of one. First, we will examine the maximum cost objective. Given $\vec{x}$, let the *left boundary location* be $lb(\vec{x}) = max\{x_i : i \in N, x_i \leq cen(\vec{x})\}$ and the *right boundary location* be $rb(\vec{x}) = min\{x_i : i \in N, x_i \geq cen(\vec{x}\}$. We denote $dist(\vec{x}) = max\{lb(\vec{x}) - lt(\vec{x}), rb(\vec{x}) - rt(\vec{x})\}$. We shall, now, look into minimizing the social cost in a strategyproof way. If we consider the algorithmic problem of locating two facilities in a way that minimizes the social cost, disregarding incentives. Given a location profile $\vec{x} \in R^n$, let the optimal facility locations be $y_1, y_2 \in R, y_1 \leq y_2$. Informally, we can associate with y_1 a multiset of locations $L(\vec{x}) \subsetneq (x_1, ..., x_n)$ (for "left") whose cost is computed with respect to y_1, and similarly associate with y_2 a multiset of locations $R(\vec{x}) \subsetneq (x_1, ..., x_n)$ (for "right") whose cost is computed with respect to y_2, such that for all $x_i \in L(\vec{x}), x_j \in R(\vec{x}), x_i \leq xj$. Now, y_1 is the median of $L(\vec{x})$ and y_2 is the median of $R(\vec{x})$. Hence, it is sufficient to optimize over the n - 1 possible choices of $L(\vec{x})$ and $R(\vec{x})$.

It can be verified that a group strategy-proof (n - 1)-approximation mechanism is given by choosing $lt(\vec{x})$ and $rt(\vec{x})$ given the location profile $\vec{x} \in R^n$. In brief, the reason is that $lt(\vec{x}) \in L(\vec{x})$ and $rt(\vec{x}) \in R(\vec{x})$ [43]

Is this the best we can do for the social cost objective when we implement nice mechanisms in the facility location setting? Surprisingly, the answer is yes.



2.1.3 Anonymous Strategyproof Mechanisms for k-Facility Location

In [21], it was proved that the approximation ratio of two-extremes mechanism is tight. However, a negative result was also established: there is no deterministic anonymous strategyproof mechanism for K-Facility Location, with k $\geq$ 3 and $n \geq K + 1$ agents.

Theorem 2.4. *For every $K \geq 3$, any deterministic anonymous strategyproof mechanism for K-Facility Location with $n \geq K + 1$ agents on the real line has an unbounded approximation ratio.*

Although the above result may seem discouraging, it can motivate us to apply the concepts of the Facility Locations Game that were previously developed to a field unaffected by the impossibility theorem: The perturbation stable family of instances.

2.2 Perturbation Stability on Clustering

The most common approach to the design and analysis of computational problems is the worst-case analysis. Mechanism Design could not be an exception to that rule. Although this method provides the most complete measurement of a problem's difficulty, it bounds us to use the same algorithm, even if we only care for "special" cases of the problem that can be solved more optimally. Unfortunately, the majority of decision and optimization problems with some sort of practical use, usually fall into the class of NP-hard problems. However, the fact that these problems have practical use can assist us in correlating them to "real-world" instances, meaning that we can take advantage of these instances' properties.

Bilu and Linial [13] were the first to suggest an approach aimed at taking advantage of this underlying structure. In particular, they introduced the notion of *stability* and they argued that instances in practice should be stable to small perturbations in the metric space. One problem that has "real-world" instances with very interesting properties, is the problem of *clustering*.

Definition 2.5 (Clustering Problem)**.** *An instance of a clustering problem is a tuple ((X,d), H,k) of a metric space (X,d), objective function H, and integer number k > 1. The objective H is a function that, given a partition of X into k sets $C_1, ..., C_k$ and a metric d on X returns a nonnegative real number, which we call the cost of the partition.*

Our goal is to minimize a cost function, depending on each data point's cost. The most well-studied and, perhaps, most interesting clustering objectives are *k-means*, *k-median*, and *k-center*. These objectives are defined as follows. Given a clustering $C_1, ..., C_k$, the objective is equal to the minimum over all choices of centers $c_1 \in C_1, ..., c_k \in C_k$ of the following functions:

$$H_{means}(C_1, ..., C_k; d) = \sum_{i=1}^{k} \sum_{u \in C_i} = d(u, c_i)^2$$

$$H_{median}(C_1, ..., C_k; d) = \sum_{i=1}^{k} \sum_{u \in C_i} = d(u, c_i)$$

$$H_{center}(C_1, ..., C_k; d) = max_{i \in 1,...,k}\{max_{u \in C_i}\{d(u, c_i)\}\}$$

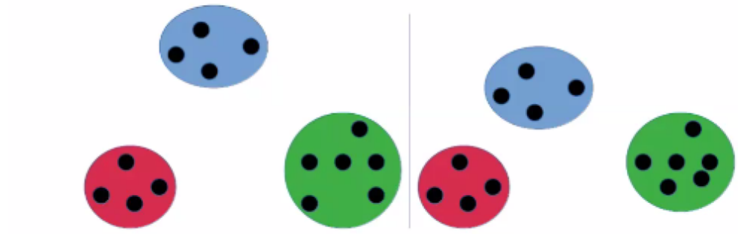
A way to describe the stability property on an instance is to define a quantity γ, which defines how close two data points can deviate while maintaining the same optimal clustering.

Definition 2.6 (γ-perturbation)**.** *Given a metric (S,d) and $\gamma \geq 1$, we say a function $d' : S \times S \to \mathbb{R}_{>0}$ is a γ perturbation of d, if for any x,y $\in$ S, it holds that*

$$d(x, y)/\gamma \leq d'(x, y) \leq d(x, y)$$

Definition 2.7 (γ-stability)**.** *Suppose we have a clustering instance composed of n points residing in a metric (S, d) and an objective function Φ we wish to optimize. We call the clustering instance γ-perturbation stable for Φ if for any d' which is an γ-perturbation of d, the (only) optimal clustering of (S, d') under Φ is identical, as a partition of points into subsets, to the optimal clustering of (S, d) under Φ.*

We developed three basic properties from the above definitions, which were used extensively by the subsequent algorithms:



Property 2.8 (γ-center proximity). *Let $p \in S$ be an arbitrary point, let c_i^* be the center p is assigned to in the optimal clustering, and let $c_j \neq c_i$ be any other center in the optimal clustering. We say a clustering instance satisfies the γ-center proximity property if for any p it holds that :*

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

. It holds that if a clustering instance satisfies the γ-pertubation stability property, then it satisfies the γ-center proximity property.

Property 2.9 (weak γ-center proximity). *Let $p \in S$ be an arbitrary point, let c_i^* be the center p is assigned to in the optimal clustering, and let $c_j \neq c_i$ be any other center in the optimal clustering. We say a clustering instance satisfies the γ-center proximity property if for any p it holds that :*

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

It holds that if a clustering instance satisfies the γ-pertubation stability property, then it satisfies the weak γ-center proximity property.

Property 2.10 (Cluster-Separation Property). *Let $(C_1, ..., C_k)$ be the optimal clustering of γ-stable instance with $\gamma \geq 2$. Let $x_i, x_i' \in C_k$ and $x_j \in C_{k'}$, with i $\neq$ j then :*

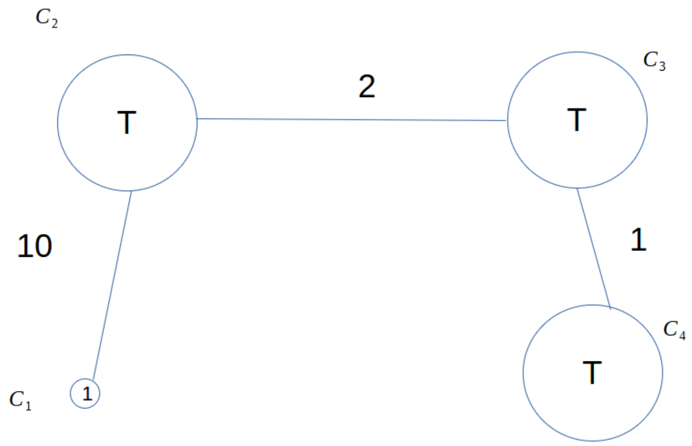
$$d(x_i, x_j) > \frac{(\gamma - 1)^2}{2\gamma} d(x_i, x_{i'})$$

Single-link++ is a clustering algorithm applied to γ-stable instances for the H_{median} objective. It recovers the optimal clustering in polynomial time. It was developed from a simpler algorithm called *single-link clustering*. *Single-link clustering* is a widely known clustering algorithm. The idea is to think of the input metric space (X,d) as a complete graph, with vertices X and edge weights given by d. The algorithm runs Kruskal's minimum spanning tree algorithm, except it stops when there are k connected components, where k is the desired number of clusters, meaning that we skip the last k-1 iterations of Kruskal. However, there is one obvious counter-example. To improve upon that *single-link++* creates a complete graph with vertices given by X and edge weight given by d and then runs Kruskal's algorithm until completion to compute the minimum spanning tree T of the complete graph induced by (X, d). The final step is to compute among all $\binom{n-1}{k-1}$ subsets of k−1 edges of T and the induced k-clusterings (with one cluster per connected component), the one with the minimum H_{median} objective function value. The following lemma then is proved :

Lemma 2.11. *Single-link++ recovers the optimal solution of instance (X, d) if and only if every optimal cluster C_i^* induces a connected subgraph of the minimum spanning tree*

The above is evident if we visualize each connected component that is left from the first k-1 iterations of Kruskal as an individual cluster. This leads to the following result:

Theorem 2.12. *In every 2-perturbation-stable k-median instance, the single-link++ algorithm recovers the optimal solution (in polynomial time).*



Single Clustering Counter Example, k = 3

2.3 Perturbation Stability on Facility Location Games

Anyone can admit that by simply taking a step back and turning our focus to more practical instances, even the NP-hard clustering problem can be dealt with with a simple and efficient algorithm. This has motivated us to investigate how these instances will behave in the Facility Location Games problem, which is heavily correlated with the clustering problem. The additional complexity of this problem is that we can no longer fully trust our input since the agents are strategic and we must find ways to discourage them from trying to declare a false location.

We can easily define γ-perturbation and γ-perturbation stability for this problem:

Definition 2.13 (Linear γ-perturbation)**.** *Let $\vec{x} = (x_1, .., x_n)$ be a locations profile. A locations profile $\vec{x}' = (x_1, .., x_n)$ is a γ-perturbation of $\vec{x}$, for some $g \geq 1$, if $x'_1 = x_1$ and for every $i \in [n-1]$, it holds that*

$$d(x_i, x_{i+1})/\gamma \leq d(x'_i, x'_{i+1}) \leq d(x_i, x_{i+1})$$

Definition 2.14 (Linear γ-stability)**.** *A k-Facility Location instance $\vec{x}$ is γ-pertubation stable (or simply, γ-stable), if $\vec{x}$ has a unique optimal clustering $(C_1, ..., C_k)$ and every γ-perturbation $\vec{x}'$ of $\vec{x}$ has the same unique optimal clustering $(C_1, ..., C_k)$*

And the same happens with the three basic properties of stability: linear γ-center proximity, linear weak γ-center proximity, and Cluster Separation Property.

In [23], a deterministic, strategyproof mechanism that obtains the optimal clustering for $2+\sqrt{3}$-stable instances was introduced. Unfortunately, it can only be applied on instances, in which their optimal clustering does not contain a singleton cluster, otherwise, the strategyproofness breaks down.

Mechanism 4 OPTIMAL : Deterministic mechanism on $2 + \sqrt{3}$-stable instances without Singleton Deviations.

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ Compute the optimal clustering $(C_1, ..., C_k)$. Let c_i be the left median point of each cluster C_i.
if ($\exists i \in [k]$ with $|C_i| = 1$) or ($\exists i \in [k-1]$ with $max\{D(C_i), D(C_{i+1})\} \leq d(C_i, C_{i+1})\}$) **then**
  Output: "FACILITIES ARE NOT ALLOCATED".
else
  Output: The k-facility allocation $(c_1, ..., c_k)$
end if

In the same paper, a deterministic, strategyproof mechanism that obtains the optimal clustering for 5-stable instances was introduced. This mechanism increases the stability of the instances that he is applied to so that it can deal with the singleton clusters in the optimal clustering of the instance.

Mechanism 5 ALMOSTRIGHTMOST : Deterministic Mechanism Resistant to Singleton Deviations 5-stable instances.

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ Find the optimal clustering $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$.
if there are two consecutive clusters C_i and C_{i+1} with $max\{D(C_i), D(C_{i+1})\} \geq d(C_i, C_{i+1})\}$) **then**
  Output: "FACILITIES ARE NOT ALLOCATED".
  for i $\in$ 1, . . . ,k **do**
    if $|C_i| > 1$ **then**
      Allocate a facility to the location of the second rightmost agent of C_i, i.e., $c_i \leftarrow x_{i,r-1}$.
    else
      Allocate a facility to the single agent location of C_i : $c_i \leftarrow x_{i,l}$
    end if
  end for
end if
Output: The k-facility allocation $\vec{c} = (c_1, ..., c_k)$

2.4 Learning Augmented Mechanisms on Facility Location Games

The notion of perturbation stability is not the only approach to mechanism design, which we can use to avoid the method of worst-case analysis. Although the use of worst-case analysis provides a certain *robustness* to the outcome of our algorithm, it deprives us of the flexibility of studying instances that are able to produce a closer-to-the-"real world" model for our problems. These kinds of problems have certain properties, that machine learning algorithms can exploit to produce useful "predictions". In [1] this train of thought was used to develop *Learning-Augmented Mechanisms* for the Facility Location Games problem.

For this new kind of mechanism, we need to introduce some new measures: *Consistency* and *Robustness*. If the prediction is accurate we define the guarantee-approximation ratio of our mechanism design as *consistency*. If the prediction is completely off, we define the guarantee-approximation ratio of our mechanism design as *robustness*

In the learning-augmented mechanism design framework, before requesting the set of preferred locations P from the agents, the designer is provided with a prediction $\hat{o}$ regarding the optimal facility location o(P). The designer can use this information to choose the rules of the mechanism but, as in the standard facility location problem, the mechanism denoted $M(P, \hat{o})$, needs to be strategyproof. In essence, if there are multiple strategyproof mechanisms the designer can choose from, the prediction can guide their choice, aiming to achieve improved guarantees if the prediction is accurate (consistency), but retaining some worst-case guarantees (robustness).

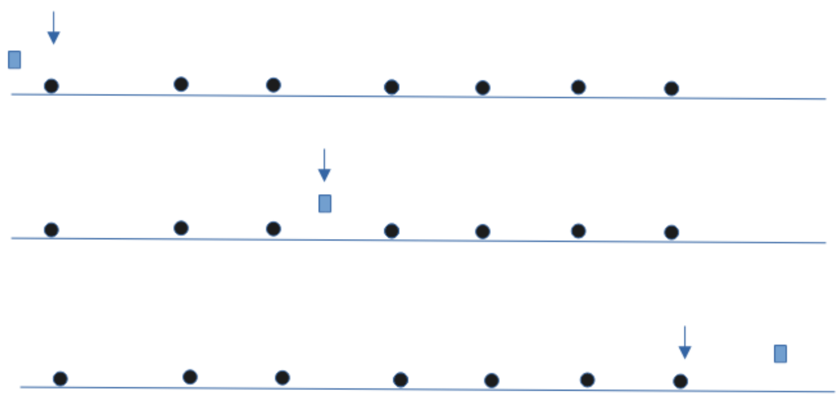
Definition 2.15 ($\alpha-consistency$)**.** *Given some social cost function C (i.e MC(·),SC(·)), a mechanism is $\alpha - consistent$ if it achieves an α-approximation ratio when the prediction is correct ($\hat{o} = o(P)$), i.e.*

$$max_P[\frac{C(M(P, o(P)), P)}{C(o(P), P)}] \leq \alpha$$

Definition 2.16 ($\beta - robustness$)**.** *Our mechanism is $\beta - robust$ if it achieves a β-approximation ratio even when the prediction is arbitrarily wrong, i.e.*

$$max_{P,\hat{o}}[\frac{C(M(P, \hat{o}), P)}{C(o(P), P)}] \leq \beta$$

We will deal with the 1-dimensional case, with the introduction of the MinMaxP mechanism. This mechanism uses the prediction $\hat{o}$ as the default facility location choice unless the prediction lies "on the left" of all the points in P or "on the right" of all the points in P. In the former case, the facility is placed at the leftmost point in P instead, and in the latter, it is placed at the rightmost point in P.



Agents are represented by black circles, prediction is represented by blue rectangular, facility location is represented by an arrow. These are the 3 different ways we can assign a facility to the instance. The first one shows what happens when $\hat{o} < min_i p_i$, the second one when $\hat{o} \in [min_i p_i, max_i p_i]$ and the third one when $\hat{o} > max_i p_i$

Mechanism 6 MinMaxP mechanism for maximum cost in one dimension.

Input: points $(p_1, ..., p_n) \in \mathbb{R}^n$, prediction $\hat{o} \in \mathbb{R}$
if $\hat{o} \in [min_i p_i, max_i p_i]$ **then**
    return $\hat{o}$
else if $\hat{o} < min_i p_i$ **then**
    return $min_i p_i$
else
    return $max_i p_i$
end if

This mechanism achieves 1-consistency and 2-robustness, which is the optimal trade-off.

2.5 Learning-Augmented Mechanism Design with Predictions for stable instances of Facility Location Games

Our goal is to integrate all the previous ingredients that were introduced (facility location games, γ-stable instances, and learning-augmented mechanism design) and try to come up with an elegant mechanism that incorporates them. Our task is to locate k facilities on the line. We will create mechanism $M(\vec{x}, \hat{o})$, which receives the tuple $(\vec{x}, \hat{o})$ as input. Our instance $\vec{x}$ is a vector consisting of each agent's location on the line and $\hat{o}$ is a vector with the predicted locations of the facilities, produced by an external system. Our work may seem easier since we already have mechanisms that are applied successfully in $gamma$-stable instances and with the addition of external predictions, it seems right to hope for an even better outcome. However, a problem arises from the fact that the agents can now exploit the mechanism through the predictions' location, adding another layer of complexity.

The mechanism that we propose is a generalized MINMAXP on k facilities. However, in MINMAXP, we had to deal with only one cluster and only one prediction, while in the general case, we had to assign k predictions to k clusters in a strategyproof way. This is the new kind of deviation that an agent can use to profit. We choose to assign the i-th prediction to the i-cluster. To avoid any issue with strategyproofness, we need at least 5 stability and the exclusion of instances with singleton clusters in their optimal clustering.

The Mechanism receives the input, runs a check on the cluster-separation property, and checks if a singleton cluster exists. If the instance passes both of these tests, we output k facilities. For our mechanism to work, we need our instance to have at least 5 stability and its optimal clustering to not include singleton clusters. If the Cluster Separation Property is not violated and there is no singleton cluster in our optimal clustering, we can match the i-th prediction $\hat{o}_i$ to the i-th cluster C_i. Since our instance is γ-stable, we can treat each cluster as a single instance, which is completely divided by the rest of the clusters, and apply the MinMaxP Mechanism [1] on each cluster. Unfortunately, we need to include the restrictions of no singleton cluster in the optimal clustering, since if we allow an agent to deviate and create a singleton cluster, without disturbing the stability of the instance, then he can isolate a distant prediction and change the enumeration of the predictions to his gain.

The mechanism is 1-consistent, and 2-robust for the Maximum Cost objective, 1-consistent and (n-

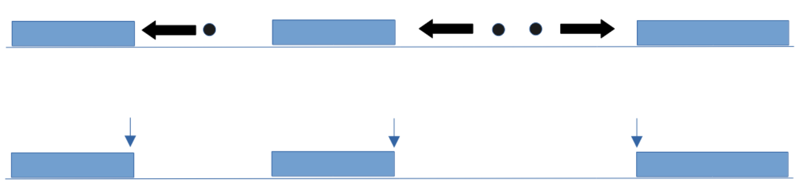
Mechanism 7 Generalized MinMaxP $M(\vec{x}, \hat{o})$:Deterministic Mechanism for 5-stable instances with no singleton clusters

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ and k-vector of predictions on facilities locations $\hat{o}$ Find the optimal clustering $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$.
for i $\in$ 1, . . . ,k **do**
  Match $\hat{o}_i$ to i-th cluster, C_i.
  if $\hat{o}_i \in [x_{i,l}, x_{i,r}]$ **then**
    Allocate a facility to $\hat{o}_i$.
  end if
  if $\hat{o}_i < x_{i,l}$ **then**
    Allocate a facility to $x_{i,l}$
  end if
  if $\hat{o}_i > x_{i,r}$ **then**
    Allocate a facility to $x_{i,r}$
  end if
end for
Output: The k-facility allocation that was previously defined.

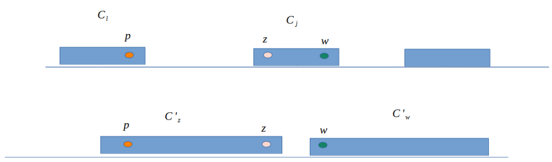
1)-robust for the Social Cost objective. Its strategyproofness is ensured by the fact that we enumerate the deviations that an agent can make:

1. Case 1 - Splits \ Merges that maintain the initial assignment of predictions to clusters
2. Case 2 - Splits \ Merges that do not maintain the assignment of predictions to clusters.
3. Case 3 - Agent deviation that changes only its own cluster's length.

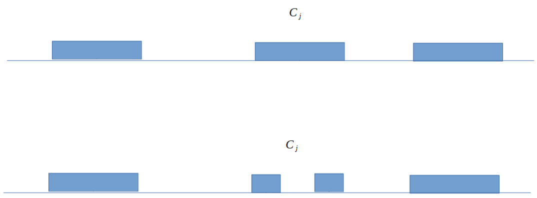
Any kind of split either violates the Cluster-Separation Property or is unable to produce a better clustering than the original. The only way for an agent to make profitable merges is to manipulate his cluster's enumeration in the instance. All of the merges produce instances that are "banned". The first one is an instance with a single-ton, which fails the mechanism singleton test. The second and third instances include splits that are either not feasible or fail the cluster-separation property. This means that there exists no profitable merge deviation. Finally, if the agent's deviation does not modify any other cluster, then he tries to include or exclude inside his cluster, the prediction that is matched to his cluster. Using the same proof plan as MINMAXP, we conclude that he can not gain by lying.



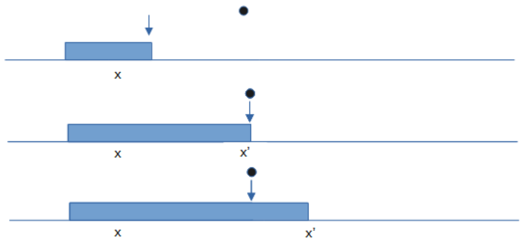
Blue rectangles are clusters, black cycles are predictions, blue arrows are allocated facilities, and i-th prediction to the i-th cluster.



In the split case 1, we can see that the CSP is violated.



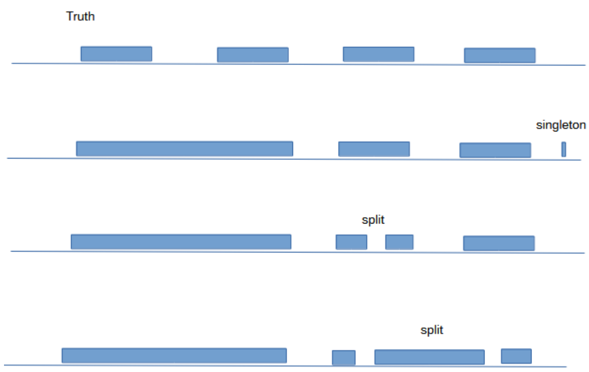
In split case 2, we can see that it is suboptimal to ”waste” two facilities on C_j agents since the optimal clustering has proven that we only need one.



2.6 Conclusion and Future Work

The result of [21] has led researchers to focus on different ways to approach k-Facility Location Games, constantly trying to combine seemingly different fields in their effort to fully understand all the limitations that may exist. Although stability and prediction-enhanced algorithms are not entirely new concepts, there is still work to be done when they are applied to k-Facility Location Games. In my thesis, I have presented a mechanism that achieves 1-consistency and n-robustness but has two shortcomings. The mechanism needs at least 5 stability, like the ALMOSTRIGHTMOST mechanism and it does not examine instances that have singleton clusters, like the OPTIMAL mechanism.

One natural direction would be to examine if we can achieve lower stability or if we can find a way to deal with the strategyproof issue that is created by the existence of singleton clusters. The stability was bounded in 5 since we based much of our proof on ALMOSTRIGHTMOST. However, ALMOSTRIGHTMOST is not restricted by singleton clusters like our mechanism. Moreover, the addition of predictions was introduced, due to the gap of stability that exists between ALMOSTRIGHTMOST and OPTIMAL, since it remains to be seen, if we can find a mechanism with $\gamma < 5$ stability, which also deals with singleton clusters in the instance. Finally, in [23] the impossibility result for k-Facility Location games with $k \geq 3$ was extended, proving 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 $\delta > 0$. It remains to be seen if this bound is tight.



The leftmost agent of the third cluster wants to belong in the second cluster of a modified instance. He can do that with 3 different merges. He can merge the first and second clusters, while he creates a singleton on his right. He can merge the first and second clusters, while he splits his cluster. He can merge the first and second clusters, while he splits the fourth and his cluster.

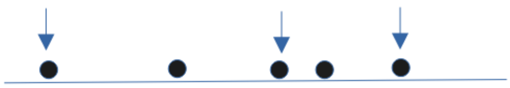
Chapter 3

Facility Location Games

Facility Location Games are among the central problems in the research agenda of approximate mechanism design without money. The simplicity of its setting yields a smooth presentation of the core concepts of Algorithmic Mechanism Design, such as *strategyproofness*, while the wide variety of its applications in real-world problems provides us with a strong motivation to analyze and improve the algorithms we implement on this problem. In this chapter, we will define the basic concepts of *Algorithm Mechanism Design*, analyze the algorithms that are involved in the most important variations of Facility Location Games, and finally we will investigate the impossibility of the existence of deterministic Mechanisms with bounded approximation, for Location Games with $k \geq 3$ facilities

3.1 Basic Setting, Definitions and Preliminaries

Our game consists of n strategic agents and k facilities. Each agent $i \in N$ has a location x_i on a metric space (X, d), which is his *private information*, and on the same metric space, we intend to place the facilities. We refer to the collection $\vec{x} = (x_1, ..., x_n)$ as the *location profile* or as the *instance*.



Definition 3.1 (Connection Cost)**.** *Given agent i, whose location is* x_i *and a collection of the facilities* $\vec{y} = (y_1, ...y_k)$*, which are all located in a metric space (X, d), where d :* $X \times X \rightarrow R{\geq}0$ *is the distance function, the connection cost of agent i, denoted as* $cost(x_i, \vec{y})$*, is the* $cost(x_i, \vec{y}) = min_{1 \leq j \leq k} d(x_i, y_j)$*, which is the distance between the agent's location and the closest facility location.*

Definition 3.2 (k-Facility Location Game)**.** *Let* $N = \{1, ..., n\}$ *be a set of strategic agents and let* $\vec{x} = (x_1, ..., x_n)$ *be the instance of the agents' locations, which is located in a metric space (X, d), where d:* $X \times X \rightarrow R{\geq}0$ *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)* $\geq$ *d(x, y) + d(x, z) for all x, y, z* $\in$ *X (triangle inequality) Our task is to place k facilities on the metric space while trying to minimize a cost function, which depends on the agents' connection costs. Each agent tries to minimize his connection cost.*

For an instance $\vec{x} = (x_1, ..., x_n) \in X^n$, we let $\vec{x}_{-i}$ denote the tuple $\vec{x}$ without coordinate x_i. For a non-empty set S of indices, we let $\vec{x}_S = (x_i)_{i \in S}$ and $\vec{x}_{-S} = (x_i)_{i \notin S}$.

We need a set of rules that will define the placement of facilities. These rules will take into account, the location that each agent presents as his "real" location. This location can be misguiding, since the agents are strategic and their goal is to manipulate the rules, as far as they can, if that places a facility closer to them. We define these rules as a *Mechanism*.

Definition 3.3 (Mechanism)**.** *Consider a $\vec{x} = (x_1, ..., x_n) \in X^n$. A deterministic Mechanism M maps $\vec{x}$ to a k-tuple $(y_1, ..., y_k) \in X^k$ of facility locations. We let $M(\vec{x})$ denote the outcome of Mechanism M, for instance, $\vec{x}$ and $M_l(\vec{x})$ denote the l-th smallest coordinate of M. A randomized Mechanism M maps $\vec{x}$ to a probability distribution over a k-tuples $(y_1, ..., y_k) \in X^k$ of facility locations.*

Given a deterministic mechanism M and instance $\vec{x}$, we denote the *connection cost* of agent *i* with respect to the outcome of $M(\vec{x})$ as $cost(x_i, M(\vec{x}))$. Given a randomized mechanism M and instance $\vec{x}$, we denote the *expected connection cost* of agent *i* with respect to the outcome of $M(\vec{x})$ as $E_{\vec{y} \sim M(\vec{x})}[cost(x_i, \vec{y})]$.

Our goal is to place the facilities in a way that minimizes a *cost function*, which depends on the agents' connection cost. Two of the most basic cost functions are the Social Cost function, the sum of all the agents' connection costs, and the Maximum Cost function, the maximum over all the agents' connection costs.

Definition 3.4 (Social Cost)**.** *The social cost of a facility locations profile $(y_1, ..., y_k) \in X^k$ is*

$$SC(\vec{x}, \vec{y}) = \sum_{i=1}^{n} cost(x_i, \vec{y})$$

Respectively, the social cost of a deterministic Mechanism M on an instance $\vec{x}$ is

$$SC(\vec{x}, M(\vec{x})) = \sum_{i=1}^{n} cost(x_i, M(\vec{x}))$$

and the social cost of a randomized Mechanism M on an instance $\vec{x}$ is

$$SC(\vec{x}, M(\vec{x})) = \sum_{i=1}^{n} E_{\vec{y} \sim M(\vec{x})}[cost(x_i, \vec{y})]$$

The optimal social cost, denoted $SC^*(\vec{x})$, is

$$SC^*(\vec{x}) = min_{\vec{y} \in X^k} \sum_{i=1}^{n} cost(x_i, \vec{y})$$

Definition 3.5 (Maximum Cost)**.** *The Maximum cost of a facility locations profile $(y_1, ..., y_k) \in X^k$ is*

$$MC(\vec{x}, \vec{y}) = max_{i \in N}[cost(x_i, \vec{y})]$$

Respectively, the Maximum cost of a deterministic Mechanism M on instance $\vec{x}$ is

$$MC(\vec{x}, M(\vec{x})) = max_{i \in \mathbb{N}}[cost(x_i, M(\vec{x}))]$$

and the Maximum cost of a randomized Mechanism M on instance $\vec{x}$ is

$$MC(\vec{x}, M(\vec{x})) = E_{\vec{y} \sim M(\vec{x})} max_{i \in N}[cost(x_i, \vec{y})]$$

The optimal maximum cost, denoted $MC^*(\vec{x})$, is

$$MC^*(\vec{x}) = min_{\vec{y} \in X^k} max_{i \in N}[cost(x_i, \vec{y})]$$

We also need to define a metric, that determines if the facilities produced by our Mechanism, are close to the optimal facility placement, with regard to the instance $\vec{x}$.

Definition 3.6 (Approximation Ratio)**.** *A mechanism M for k-Facility Location achieves an approximation ratio of $\rho \geq 1$ for the objective of Social cost (respectively for Maximum cost), for all instances $\vec{x}$,*

$$SC(\vec{x}, M(\vec{x})) \leq \rho SC^*(\vec{x})$$

(respectively for maximum cost $MC(\vec{x}, M(\vec{x})) \leq \rho MC^*(\vec{x})$)

We have mentioned above, that in the facility location setting, our goal is to minimize a cost function, which depends on the agents' connection costs and the goal of each agent is to minimize his connection cost. It becomes evident that these two goals can be conflicting, since the location that each agent presents as his "real" location can be misguiding, since the agents are strategic and their goal is to manipulate the rules, as far as they can, if that places a facility closer to them. We need our mechanisms not only to be unaffected by these kinds of motives but also to provide the agents an incentive to share their true location. To define such Mechanisms, we introduce the property of *Strategyproofness*.

Definition 3.7 (Strategyproofness, Group Strategyproofness and Partial Group strategyproof)**.** *A mechanism M is strategyproof if no agent can benefit from misreporting his location. Formally, for all instances $\vec{x}$, every agent i, and all locations $y \in X$,*

$$cost(x_i, M(\vec{x})) \leq cost(x_i, M(\vec{x}_{-i}, y))$$

A Mechanism M is group strategyproof, if any coalition of agents misreports their locations, at least one agent does not benefit. Formally, for all instances $\vec{x}$, every coalition of agents S, and all sub-instances y_S, there exists an agent $i \in S$ such that

$$cost(x_i, M(\vec{x})) \leq cost(x_i, M(\vec{x}_{-S}, y_S))$$

Mechanism M is a partial group strategyproof, if for any coalition of agents that occupy the same location, none of them can benefit if they misreport their location simultaneously. Formally, for all instances $\vec{x}$, every coalition of agents S, all occupying the same location x in $\vec{x}$, and all sub-instances y_S,

$$cost(x, M(\vec{x})) \leq cost(x, M(\vec{x}_{-S}, y_S))$$

By definition, any group strategyproof mechanism is a partial group strategyproof, and any partial group strategyproof mechanism is strategyproof. In Lu et al. [2010, Lemma 2.1] [32], it is shown that any strategyproof mechanism for K-Facility Location is also a partial group strategyproof .

3.2 Facility Location Mechanisms on the line

Although we have introduced the basic concepts of Facility Location Games on the general metric (X,d), we will focus on the most important variations of this problem on the line, meaning that we assume (X,d) = ($\mathbb{R}, |\cdot|$), where $|\cdot|$ is the euclidean distance. We also denote $lt(\vec{x})$, as the location of the leftmost agent of the instance, $rt(\vec{x})$, as the location of the rightmost agent of the instance, and $cen(\vec{x}) = \frac{lt(\vec{x})+rt(\vec{x})}{2}$, as the center of the interval $[lt(\vec{x}), rt(\vec{x})]$.

3.2.1 Mechanism for one facility on the line

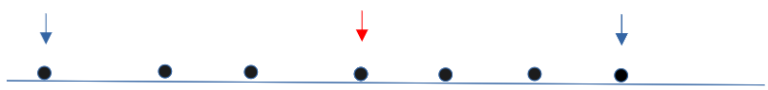
Our first task is to find a strategyproof Mechanism that minimizes the social cost for the Facility Location Game of n agents and one facility. The solution is pretty straightforward. We can choose the median location in $\vec{x}$ - $med(\vec{x})$.

Theorem 3.8. *$M(\vec{x}) = med(\vec{x})$ is a group-strategyproof optimal mechanism for social cost*

Proof. **Is $med(\vec{x})$ optimal?** Assume that n is odd, n = 2k + 1. If we choose any agent on the left of $med(\vec{x})$, then the Social Cost increases, since it is further away from at least k + 1 agents and closer to at most k agents. The same holds for any agent to the right of the median. Assume that n is even, n = 2k, then any point in $[x_k, x_{k+1}]$ produces the optimal social cost, for the same reason as in the case of n = 2k + 1. So, $med(\vec{x})$ is optimal.

Is $med(\vec{x})$ strategyproof? The structure of the preferences of our agents is known in the social choice literature as single peaked: the peak, or bliss point, of agent i is at x_i, and the closer a location is to x_i, the more preferred it is. It has long been known that, when agents have single-peaked preferences, the selection of the k-th order statistic for some $k \in 1, ..., n$ is group strategyproof [40]. So $med(\vec{x})$ is group strategyproof.

It is evident that $med(\vec{x})$ is a group-strategyproof optimal mechanism for social cost □



Our second task is to find an optimal and strategyproof mechanism for the maximum cost objective in the one facility setting. The facility placement that minimizes our objective, is the location $cen(\vec{x})$. Unfortunately, this placement is not strategyproof, since any agent can lie and change the instance's length, until $cen(\vec{x'})$ lands on this agent's location. Procaccia and Tennenholtz [43] proposed the following group strategyproof 2-approximation mechanism for the maximum cost, $M(\vec{x}) = lt(\vec{x})$.

Theorem 3.9. *$M(\vec{x}) = lt(\vec{x})$ is a group strategyproof 2-approximation mechanism for the maximum cost*

Proof. We have already showed that the selection of any k-th order statistic for some $k \in 1, ..., n$ is group strategyproof [40], so $lt(\vec{x})$ is group strategyproof. The optimal maximum cost is $\frac{rt(\vec{x})-lt(\vec{x})}{2}$ and the maximum cost of $lt(\vec{x})$ is $rt(\vec{x}) - lt(\vec{x})$. So we end up with 2-approximation. □

A natural question that arises is, if this the best we can do for the maximum cost objective. Procaccia and Tennenholtz [43], indeed proved that $lt(\vec{x})$ is the best possible mechanism for the one facility setting, wrt the maximum cost function.

Theorem 3.10. *Let $N = 1, ..., n$, $n \geq 2$. Any deterministic strategyproof mechanism M : $\mathbb{R}^n \to R$ has an approximation ratio of at least 2 for the maximum cost.*

Proof. In order to prove that no deterministic strategyproof mechanism can have a better approximation ratio than 2, in the one facily setting for the maximum cost, we will create an instance, that will cause a strategyproof inefficiency to any deterministic strategyproof mechanism with approximation ratio lower than 2. Assume that we have to deal with 2 agents, N = 1,2 and that M, a deterministic strategyproof mechanism with approximation ratio lower than 2, exists. Suppose that $x_1 = 0$ and $x_2 = 1$, $\vec{x} = (x_1, x_2)$, which means that without loss of generality $M(\vec{x}) \neq \frac{1}{2}$, since this is not a strateygproof facility allocation, as we have mentioned above. So $M(\vec{x}) = \frac{1}{2} + \epsilon$, $\epsilon \geq 0$. We, now, create another instance $\vec{x'} = 1, \frac{1}{2} + \epsilon$. This instance has an optimal maximum cost of $\frac{1}{4} + \frac{\epsilon}{2}$. Since, M achieves an approximation lower than 2, the mechanism places the facility in the interval $(1, \frac{1}{2} + \epsilon)$. The agent that is located at $\frac{1}{2} + \epsilon)$ has an incentive to lie and declare that his location is at 1. This creates the instance $\vec{x}$ and since $M(\vec{x}) = \frac{1}{2} + \epsilon$, agent 2 manages to land the facility on him, making mechanism M not strategyproof, which contradicts our initial assumption that such a mechanism exists. To generalize this family of instances to arbitrary n, we just need to locate all agents $N \setminus \{1, 2\}$ at $\frac{1}{2}$ in each one of the profiles, discussed above. □

3.2.2 Mechanisms for two facilities on the line

We can now deal with the extension of the previous setting, locating two facilities, instead of one. First, we will examine the maximum cost objective. Given $\vec{x}$, let the *left boundary location* be $lb(\vec{x}) = max\{x_i : i \in N, x_i \leq cen(\vec{x})\}$ and the *right boundary location* be $rb(\vec{x}) = min\{x_i : i \in N, x_i \geq cen(\vec{x}\}$. We denote $dist(\vec{x}) = max\{lb(\vec{x}) - lt(\vec{x}), rb(\vec{x}) - rt(\vec{x})\}$.

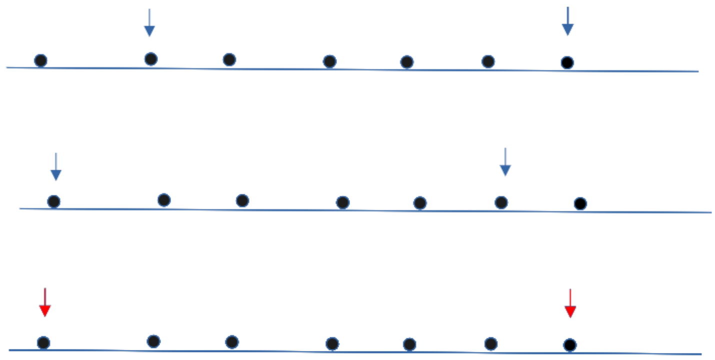
From [43], we get the following result :

Lemma 3.11. *Given $\vec{x}$, the optimal placement of two facilities has a maximum cost of* $\frac{dist(\vec{x})}{2}$

We shall, now, look into minimizing the social cost in a strategyproof way. If we consider the algorithmic problem of locating two facilities in a way that minimizes the social cost, disregarding incentives. Given a location profile $\vec{x} \in R_n$, let the optimal facility locations be $y_1, y_2 \in R, y_1 \leq y_2$. Informally, we can associate with y_1 a multiset of locations $L(\vec{x}) \subsetneq (x_1, ..., x_n)$ (for "left") whose cost is computed with respect to y_1, and similarly associate with y_2 a multiset of locations $R(\vec{x}) \subsetneq (x_1, ..., x_n)$ (for "right") whose cost is computed with respect to y_2, such that for all $x_i \in L(\vec{x}), x_j \in R(\vec{x}), x_i \leq xj$. Now, $y_1$is the median of $L(\vec{x})$ and y_2 is the median of $R(\vec{x})$. Hence, it is sufficient to optimize over the n - 1 possible choices of $L(\vec{x})$ and $R(\vec{x})$.

It can be verified that a group strategy-proof (n - 1)-approximation mechanism is given by choosing $lt(\vec{x})$ and $rt(\vec{x})$ given the location profile $\vec{x} \in R_n$. In brief, the reason is that $lt(\vec{x}) \in L(\vec{x})$ and $rt(\vec{x}) \in R(\vec{x})$ [43]

Is this the best we can do for the social cost objective when we implement nice mechanisms to the facility location setting? Surprisingly, the answer is yes.



3.3 Anonymous Strategyproof Mechanisms for k-Facility Location

In this section, our goal is to examine, whether it is possible to find a mechanism with a better approximation than n-1 and whether we can find a nice mechanism for the setting of k-Facility Location with $k \geq 3$. We will need to define some extra notations, while also presenting some important lemmas.

For an instance $\vec{x} = (x_1, ..., x_n)$, we say that the agents are arranged on the line according to a permutation π if π arranges them in increasing order of their locations in $\vec{x}$, that is, $x_{\pi(1)} \leq x_{\pi(2)} \leq ... \leq x_{\pi(n)}$. We consider *3-agent instances*, where n = 3, and 3-location instances, where there are three different locations x_1, x_2, x_3 and a partition of N into three coalitions N_1, N_2, N_3 such that all agents in coalition N_i occupy location $x_i, i \in 1, 2, 3$. We denote such an instance as $(x1 : N1, x2 : N2, x3 : N3)$. For a set N of agents, we let I(N) denote the set of all instances, and let $I_3(N)$ denote the set of all 3-location instances

A mechanism M is *anonymous*, if for all $\vec{x}$ and all agents permutations π, the outcome of M depends only on the location of the agents and not their identities, that is π, $M(\vec{x}) = M(x_{\pi(1)}, ..., x_{\pi(n)})$

Definition 3.12 (Nice Mechanisms). *Let M be a Mechanism that is deterministic, strategyproof and has a bounded approximation ratio. We declare M to be a nice Mechanism for the social cost objective.*

Any nice mechanism F for K-Facility Location is unanimous since otherwise, M would not have a bounded approximation ratio. We can prove this easily , since for any instance $\vec{x}$, where the agents occupy K different locations $x_1, ..., x_K$, we have $SC^*(\vec{x}) = 0$, with optimal facility allocation = $(x_1, ..., x_K)$. Any other facility allocation $(y_1, ..., y_K) \neq (x_1, ..., x_K)$ produces a social cost $SC(\vec{x}, \vec{y}) > 0$, which means that our approximation ratio is unbounded.

Definition 3.13 (Well-Separated instances). *Given a nice mechanism M for K-Facility Location with approximation ratio ρ, a (K + 1)-agent instance $\vec{x}$ is called $(i_1|...|i_{K-1}|i_K, i_{K+1})-well-separated$ if $x_{i_1} < ... < x_{i_{K+1}}$ and $\rho(x_{i_{K+1}} - x_{i_K}) < min_{2\leq\ell\leq K}\{x_{i_{\ell+1}} - x_{i_\ell}\}$*

Hence, given a ρ-approximate mechanism for K-Facility Location, a well-separated instance includes a pair of nearby agents at a distance to each other less than $\frac{1}{\rho}$ times the distance between any other pair of consecutive agents. Therefore, any ρ-approximate mechanism serves the two nearby agents by the same facility and serves each of the remaining "isolated" agents by a different facility. The intuition is that by employing well-separated instances, we are able to reduce K-Facility Location on well-separated instances to a single-facility two-agent location game parameterized by the identities and the locations of the K−1 isolated agents.

Definition 3.14 (Image Sets). *Given a mechanism M, the image (or option) set $I_i(\vec{x}_{-i})$ of an agent i with respect to an instance $\vec{x_i}$ is the set of facility locations the agent i can obtain by varying her reported location. Formally, $I_i(\vec{x}_{-i}) = \{a \in \mathbb{R} : \exists y \in \mathbb{R}\, with\, M(\vec{x}_{-i}, y) = a\}$*

One can show that if M is strategyproof, any image set $I_i(\vec{x}_{-i})$ is a collection of closed intervals and M places a facility at the location in $I_i(\vec{x}_{-i})$ nearest to the location of agent i.

Lemma 3.15. *Let M be a strategyproof mechanism for the k-Facility game. For every location profile $\vec{x}$, any agent $i \in N$ and any location $y \in \mathbb{R}$ we have*

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

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, he can benefit by misreporting to y^* lowering his connection cost from $d(y, M((\vec{x}_{-i}\ , y^*)))$ to $d(a^*, y)$. This contradicts the assumption that M is strategyproof. □

We can extend the previous definition of the 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:

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

Also, in [32] the previous lemma is extended to hold for partial group strategyproof mechanisms when all agents in the coalition report the same location.

Lemma 3.16. *Let M be a strategyproof mechanism for the k-Facility game. For every location profile $\vec{x}$, any non-empty set of agents S, any location $y \in \mathbb{R}$ and vector of locations $\vec{y} = (y, ...y)$ we have :*

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

Definition 3.17 (Holes). *Any (open) interval in the complement of an image set $I \equiv I_i(\vec{x}_{-i})$ is called a hole of I. Given a location $y \in I$, we let $l_y = sup_{a \in I}\{a < y\}$ and $r_y = inf_{a \in I}\{a > y\}$ be the locations in I nearest to y on the left and on the right, respectively. Since I is a collection of closed intervals, l_y and r_y are well defined and satisfy $l_y < y < r_y$. Given a $y \in I$, we refer to the interval (l_y, r_y) as y-hole in I.*

Any hole in an image set $I_i(\vec{x}_{-i})$ of F is a bounded interval. Otherwise, that is, if there was an image set $I_i(\vec{x}_{-i})$ with a hole that extends either to $-\infty$ or to $+\infty$, we could move agent i sufficiently far away from the remaining agents, and obtain an instance for which F would have approximation ratio larger than ρ. Therefore, if F is a nice mechanism, for any instance $\vec{x}$) and any agent i, there is a sufficiently small (respectively, large) a such that if i moves to a, F allocates a facility to a that is, $a \in F(\vec{x}_{-i}, a)$.

A mechanism M is *unanimous*, if for all $M(\vec{x})$, where the agents occupy K different locations $(x_1, ..., x_k)$, the outcome of M is $M(\vec{x}) = (x_1, ..., x_k)$

First, we will present some useful properties of nice mechanisms for k-Facility Location applied to instances with k+1 agents.

Proposition 3.18. *Let M be a nice mechanism for K-Facility Location on the line. For any (K + 1)-location instance $\vec{x}$ with $x_{i_1} < \cdots < x_{i_{K+1}}$, we have that $M_1(\vec{x}) \leq x_{i_2}$ and $M_K(\vec{x}) \geq x_{i_K}$.*

Proof. Let us assume that $x_{i_2} < M_1(\vec{x})$ (the other case is symmetric). Then, the agents at x_{i_1} have an incentive to report x_{i_2} and decrease their connection cost, since $x_{i_2} \in M(\vec{x_{-i}}, x_{i_2})$, due to the bounded approximation ratio of M. This contradicts M's (partial group) strategyproofness. □

Proposition 3.18 points out the relative power of the two extreme agents for (K+1)-location instances. Those two agents force any nice mechanism to place the facilities at the two "edges" of the instance, otherwise, any agent can exploit the mechanism for his own gain. It is also evident that this proposition applies to instances with more than k+1 agents, as long as the agents are located to k+1 points on the line.

Proposition 3.19. *Let M be a nice mechanism for K-Facility Location on the line. For any $(i_1|...|i_{K-1}|i_K, i_{K+1})$-well-separated instance $\vec{x}$, $M_K(\vec{x}) \in [x_{i_K}, x_{i_{K+1}}]$.*

Proof. Since M has a bounded approximation ratio, the two nearby agents i_K and i_{K+1} are both served by the same facility at $M_K(\vec{x})$. By Proposition 3.18, $M_K(\vec{x}) \geq x_{i_K}$. Moreover, $M_K(\vec{x}) \leq x_{i_{K+1}}$. Otherwise, the agent i_K could report $x_{i_{K+1}}$ and decrease his cost, since $x_{i_{K+1}} \in M(\vec{x}_{-i_K}, x_{i_{K+1}})$, due to the bounded approximation ration of M □

Proposition 3.19 suggests that in a well-separated instance, the two nearby agents have a facility allocated between them, while every other agents is served by a facility that is placed on top of them.

Proposition 3.20. *Let M be a nice mechanism for K-Facility Location on the line and let $\vec{x}$ any $(i_1|...|i_{K-1}|i_K, i_{K+1})$-well-separated instance, such that $M_K(\vec{x}) = x_{i_K}$. Then, for every $(i_1|...|i_{K-1}|i_K, i_{K+1})$-well-separated instance $\vec{x'} = (\vec{x}_{-\{i_K, i_{K+1}\}}, x'_{i_K}, x'_{i_{K+1}})$ with $x_{i_K} \leq x'_{i_K}$, it holds that $M_K(\vec{x'}) = x'_{i_K}$*

Proposition 3.21. *Let M be a nice mechanism for K-Facility Location on the line and let $\vec{x}$ any $(i_1|...|i_{K-1}|i_K, i_{K+1})$-well-separated instance, such that $M_K(\vec{x}) = x_{i_{K+1}}$. Then, for every $(i_1|...|i_{K-1}|i_K, i_{K+1})$-well-separated instance $\vec{x'} = (\vec{x}_{-\{i_K, i_{K+1}\}}, x'_{i_K}, x'_{i_{K+1}})$ with $x'_{i_{K+1}} \leq x_{i_{K+1}}$, it holds that $M_K(\vec{x'}) = x'_{i_{K+1}}$*

Propositions 3.20, 3.21, show that if there exists a$(i_1|...|i_{K-1}|i_K, i_{K+1})$-well-separated instance $\vec{x}$ with $M_K(\vec{x'}) = x'_{i_K}$ (respectively, $M_K(\vec{x'}) = x'_{i_{K+1}}$), then as long as we "push" the locations of agents i_K and i_{K+1} to the right (respectively, left), while keeping the instance well-separated, the rightmost facility of F stays with the location of agent i_K (respectively, i_{K+1}). We sometimes refer to this property as the *consistent allocation property* of well-separated instances.

Propositions 3.18, 3.19, 3.20, 3.21 combine a set of characteristics, which anonymous, nice mechanisms are required to exhibit. Taking that into consideration, we can establish that the image sets of nice mechanisms do not include holes in certain intervals. If such a hole (l, r) exists, we are able to construct a well-separated instance (or a pair of well-separated instances), utilizing the (l, r) hole and show that such an instance contradicts either the *consistent allocation property* of well-separated instances or the bounded approximation ratio of the nice mechanism. This contradiction leads us to certain interesting properties about anonymous, nice mechanisms that are applied on K-Facility Location, with $K \geq 3$.

Using the *consistent allocation property* of well-separated instances, we can prove that any anonymous nice mechanism for K-Facility Location on instances with K+1 agents has to place the facilities in the two extreme locations. First, we show that the rightmost facility of an anonymous, nice mechanism is always allocated to the rightmost agent.

Lemma 3.22. *Let M be any anonymous nice mechanism for K-Facility Location with $K \geq 2$ and n = K + 1 agents. Then, for all instances $\vec{x} = (x_1, ..., x_K, x_{K+1})$, with $x_1 \leq \cdots \leq x_K \leq x_{K+1}$, $M_K(\vec{x}) = x_{K+1}$.*

Proof. We will prove this lemma for K = 3 facilities. We can generalize this result for any $K \geq 2$. We consider instance $\vec{x} = (x_1, x_2, x_3, x_4)$, where $x_1 \leq x_2 \leq x_3 \leq x_4$. Our proof plan is to examine the image set of agent 4, $I_4(\vec{x}_{-4})$ and show that $I_4(\vec{x}_{-4})$ includes the entire interval $[x_3, \infty]$. Since the image set $I_4(\vec{x}_{-4})$ does not have any holes on the right of x_3, we can conclude that for any $y \geq x_3$, $M_4(\vec{x}) = y$, while taking into account that M is strategyproof. So, agent 4 receives a facility on him, as long as he is the rightmost agent. We will assume that there is a hole (l,r) in $I_4(\vec{x}_{-4})$ on the right of x_3, which will allow us to obtain two well-separated instances, that will contradict the properties of nice mechanism M. Our first instance $\vec{x'} = (x_1, x_2, l, l + \epsilon)$ will move agent 3 to l and agent 4 to $l + \epsilon$ and our second instance $\vec{x''} = (x_1, x_2, r, r - \epsilon)$ will move agent 3 to r and agent 4 to r - ϵ. By strategyproofness, $l \in M(\vec{x})$ and $M_3(\vec{x}) = r$. By applying 3.21, when we try to go from $\vec{x}''$ to $\vec{x}'$, we have that $M(\vec{x}')$ needs to allocate facilities to l and $l + \epsilon$, which is the contradiction we are looking for.

Formally, we assume that there is a hole (l, r) in $I_4(\vec{x}_{-4})$, with $l \geq x_3$. We consider an intermediate instance $\vec{x}^0 = (x_1, x_2, x_3, l + \epsilon)$ and the instance $\vec{x}' = (x_1, x_2, l, l + \epsilon)$, where $\epsilon > 0$ is chosen small enough that the instance $\vec{x}'$ is (1||2|3,4)-well-separated and that $l + \epsilon$ is closer to l than to r. Since M is strategyproof and l is the closest point of $I_4(\vec{x}_{-4})$ to $l + \epsilon$, we have that $l \in M(\vec{x}^0)$. Moreover, due to M's strategyproofness, $l \in M(\vec{x}')$. If $l \notin M(\vec{x}')$, then the agent that resided on location l can declare his location as x_3 and change the instance from $\vec{x'}$ to $\vec{x}^0$, which leads to the dissolution of strategyproofness, since $l \in M(\vec{x}^0)$.

We, now, create two more instances, the intermediate instance $\vec{x}^1 = (x_1, x_2, x_3, r-\epsilon)$ and the instance $\vec{x}'' = (x_1, x_2, r, r - \epsilon)$, where $\epsilon > 0$, $\vec{x}''$ is (1||2|3,4)-well-separated and $r - \epsilon$ is closer to r than to l. Since M is strategyproof and r is the closest point of $I_4(\vec{x}_{-4})$ to $r - \epsilon$, we have that $r \in M(\vec{x}^1)$. Moreover, due to M's strategyproofness, $r \in M(\vec{x}'')$. If $r \notin M(\vec{x}'')$, then the agent that resided on location r can declare his location as x_3 and change the instance from $\vec{x}''$ to $\vec{x}^1$, which leads to the dissolution of strategyproofness, since $r \in M(\vec{x}^1)$. From 3.19, we know that $M_3(\vec{x}'') \in [r - \epsilon, r]$. Since, $r \in M(\vec{x}'')$ and $M_3(\vec{x}'') \in [r - \epsilon, r]$, we have that $M_3(\vec{x}'') = r$. A key note is that since M is anonymous, we only care for the agents' locations. We highlight that agents 3 and 4 implicitly switch indices in $\vec{x}^1$ and $\vec{x}''$. More specifically, since we require that the agents are arranged on the line in increasing order of their indices, the location of agent 3 is x_3 in $\vec{x^1}$ and $r-\epsilon$ in $\vec{x}''$ and the location of agent 4 is $r-\epsilon$ in $\vec{x}^1$ and r in $\vec{x''}$. Therefore, to argue about the outcome of M($\vec{x}''$) based on the outcome of M($\vec{x}^1$), we resort to the anonymity of M.

Our last step is to observe that $\vec{x}''$ is (1||2|3,4)-well-separated such that $M_3\vec{x}'') = r = x''_4$ From 3.21, for every (1||2|3,4)-well-separated instance $\vec{x}_{new} = (\vec{x}''_{-\{3,4\}}, x_{new_3}, x_{new_4})$, with $x_{new_4} \leq x''_4$, it holds that $M_3(\vec{x''}) = x_{new_4}$. We know that $\vec{x}'$ is a (1||2|3,4)-well-separated instance and that $x'_4 = l + \epsilon \leq r = x''_4$, which means that $M_3(\vec{x}'') = l + \epsilon$. Since both x'_3 and x'_4 are served by a facility, either agents 1 and 2 are served by one facility or agent 2 is served by the facility of x'_3. In both cases, we obtain a contradiction to the bounded approximation ratio of M, which means that there is no hole (l, r) in $I_4(\vec{x}_{-4})$ and $M_3(\vec{x}) = x_4$ □

Using a symmetric argument, we can show that the leftmost facility of an anonymous nice mechanism is always allocated to the leftmost agent.

Lemma 3.23. *Let M be any anonymous nice mechanism for K-Facility Location with $K \geq 2$ and n = K + 1 agents. Then, for all instances $\vec{x} = (x_1, ..., x_K, x_{K+1})$, with $x_1 \leq \cdots \leq x_K \leq x_{K+1}$, $M_1(\vec{x}) = x_1$*

Combining 3.22, 3.23, we can easily conclude that any anonymous nice mechanism for 2-Facility Location on instances with three agents has to place the facilities in the two extreme locations.

Lemma 3.24. *Let M be any deterministic anonymous strategyproof mechanism with a bounded approximation ratio for 2-Facility Location with n = 3 agents. Then, for all instances $\vec{x}$, with $x_1 \leq x_2 \leq x_3$, $M(\vec{x}) = (x_1, x_3)$.*

We can extend Lemma 3.3 to 3-location instances by restating the proofs of lemmas 3.22, 3.23 (and those of propositions 3.20 and 3.21 with three coalitions of agents instead of three agents. Using the fact that any strategyproof mechanism is also partial group strategyproof [32], we obtain that:

Corollary 3.25. *Let F be any deterministic anonymous strategyproof mechanism with a bounded approximation ratio for 2-Facility Location applied to 3-location instances with n $\geq$ 3 agents. Then, for all instances $\vec{x} \in I_3(N)$, $M(\vec{x}) = (min\vec{x}, max\vec{x})$.*

Finally, we can use induction on the number of different locations and extend 3.25 to instances with any number of agents on any number of locations.

Theorem 3.26. *Let M be any deterministic anonymous strategyproof mechanism with a bounded approximation ratio for 2-Facility Location. Then, for all instances $\vec{x} \in I(N)$, $M(\vec{x}) = (min\vec{x}, max\vec{x})$.*

Proof. On a conceptual level, we will try to create an instance $\vec{x}$ for which $M(\vec{x}) \neq (min\vec{x}, max\vec{x})$, which can be converted to another instance with less distinct locations for the agents, while the new instance has a non-extreme facility location, thus contradicting 3.25.

Formally, let $\vec{x} = (x_1, ..., x_n) \in I(N)$ be an instance for which $M(\vec{x}) \neq (min\vec{x}, max\vec{x})$. We let j be an agent located at $min\vec{x}$ and k be an agent located at $max\vec{x}$. Since, $M(\vec{x}) \neq (x_j, x_k)$, then there is a location a , $a \neq x_j$ and $a \neq x_k$, such that $a \in M(\vec{x})$

First, we examine the case, where $x_j < a < x_k$. We iteratively move every agent $i \in N\ \{j, k\}$ to the location a, creating instance $\vec{x'}$. The new instance needs to allocate a facility on a, since if it allocated elsewhere, any agent $i \in N\ \{j, k\}$, can lie and create instance $\vec{x}$, for which M allocates a facility on a. So, we now have the 3-location instance $\vec{x}' = (x_j : j, a : N\ \{j, k\}, x_k : k)$, where $a \in M(\vec{x})$. From 3.25, $M(\vec{x}') = (min\vec{x}, max\vec{x}) = (x_j, x_k)$. This leads to a contradiction, since $a \neq x_j$ and $a \neq x_k$. Therefore, for all instances $\vec{x} \in I(N)$, $M(\vec{x})$ does not allocate a facility inside the two extremes.

Now, we examine the case, where $x_j < a$ or $a < x_k$. We will focus on the case of $x_j < a$ (case of $x_k > a$ is identical). We assume that $\vec{x}$ has the minimum number of distinct locations among all instances for which M assigns a facility outside of $[min\vec{x}, max\vec{x}]$. Since $x_j < a$, either $x_j \notin M(\vec{x})$ or $x_k \notin M(\vec{x})$. Without loss of generality, we assume that $x_j \notin M(\vec{x})$. We declare $S_j =$ agents located in x_j and $b =$ minx$\vec{x}_{-S_j}$, which is the second location from left of $\vec{x}$. Since $x_j \notin M(\vec{x})$, there exists a x_j hole (l, r) in the image set $I_{S_j}(\vec{x}_{-S_j})$. If we create the instance $\vec{x}'' = (\vec{x}_{-S_j}), (b, b, .., b))$, where every agent of S_j went to location b, then we have an instance with less distinct locations than $\vec{x}$. Since, $\vec{x}$ has the minimum number of distinct locations among all instances for which M assigns a facility outside of $[min\vec{x}, max\vec{x}]$, M needs to assign a facility inside the interval $[min\vec{x}'', max\vec{x}''] = [b, x_k]$. However, we know that from the previous case, we can not locate a facility at $(min\vec{x}'', max\vec{x}'') = (b, x_k)$. So we conclude that $M(\vec{x}'') = (b, x_k)$, $b \in I_j(\vec{x}_{-j})$ and $r \leq b$. We can now create the instance $\vec{x}''' = (\vec{x}_{-S_j}), (r - \epsilon, ..., r - \epsilon))$. Since M is strategyproof and r is the closest location to $r - \epsilon$ in $I_{S_j}(\vec{x}_{-S_j})$, $M(\vec{x}''')$ allocates a facility to $r > r - \epsilon$ [32]. Therefore, $M(\vec{x}''')$ allocates a facility inside the two extremes of $\vec{x}'''$, which contradicts our assumption that M does not allocate a facility inside the two extremes. □

A consequence of 3.26 is that the TWO-EXTREMES mechanism of Procaccia and Tennenholtz [43] is the only anonymous nice mechanism for 2-Facility Location.

Corollary 3.27. *A deterministic anonymous mechanism M for 2-Facility Location is strategyproof and has a bounded approximation ratio if and only if for all instances $\vec{x}$, $M(\vec{x}) = (min\vec{x}, max\vec{x})$.*

We have now produced all the tools needed to prove the impossibility result for anonymous nice K-Facility Location mechanisms, for all $K \geq 3$.

Theorem 3.28. *For every $K \geq 3$, any deterministic anonymous strategyproof mechanism for K-Facility Location with $n \geq K+1$ agents on the real line has an unbounded approximation ratio.*

Proof. We will consider the case where $K = 3$ and $n = 4$. From there, it is straightforward to verify that this proof applies for $K \geq 3$ and $n \geq K+1$. We assume that there is an anonymous and nice mechanism M, which has a bounded approximation ratio ρ. For some large enough $\lambda > \rho$, we build the instance $\vec{x} = (x_1, x_2, x_3, x_4) = (0, \lambda, 3\lambda^2 + \lambda, 3\lambda^2 + \lambda + 1)$. From 3.22 and 3.23, we allocate facilities at 0 and at $3\lambda^2 + \lambda + 1$, $M(\vec{x}) = (0, 3\lambda^2 + \lambda + 1)$. Since $3\lambda^2 + \lambda \notin M(\vec{x})$, that means $3\lambda^2 + \lambda \notin I_3(\vec{x}_{-3})$, which implies that there is a hole (l, r) in $I_3(\vec{x}_{-3})$.

We will now try to place the hole (l, r) on the real line. If agent 3 moves to location $3\lambda^2 + \lambda + 1$, he will be assigned a facility, since M is unanimous. That fact leads us to an upper bound of r, while also having a lower bound at $3\lambda^2 + \lambda$, which provides us with the relative location of r, $3\lambda^2 + \lambda < r \leq 3\lambda^2 + \lambda + 1$. Suppose that $l < \lambda^2 + \lambda - 1$ and that y is the location of the third agent, if he moves on the left and creates the instance $\vec{x}' = (0, \lambda, y, 3\lambda^2 + \lambda + 1)$. Moreover, suppose that $y = 2\lambda^2 + \lambda$, then the closest point of $I_3(\vec{x}_{-3})$ will be r, where $r > 3\lambda^2 + \lambda$. Since M is strategyproof , a facility is going to be allocated on r and $SC(\vec{x}', M(\vec{x}')) > 2\lambda^2 +$. Since, the optimal social cost for $\vec{x}'$ is λ, we have that M approximation ratio would be $\lambda > \rho$, which is a contradiction. So we have that $l \geq \lambda^2 + \lambda - 1$

Finally, we will follow the same proof steps as in 3.18, 3.18. We create instance $\vec{x}^0 = (0, \lambda, l + \epsilon, 3\lambda^2 + \lambda + 1)$. From 3.22, 3.23, we have that $M_1(\vec{x}^0) = 0$ and $M_1(\vec{x}^0) = 3\lambda^2 + \lambda + 1$. Location l is the closest point of $I_3(\vec{x}_{-3})$ to $l + \epsilon$, so from M strategyproofness, we must allocate a facility on l. We create instance $\vec{x}'' = (x''_1, x''_2, x''_3, x''_4) = (0, \lambda, l, l + \epsilon)$. We must allocate a facility on l, else the agent that is located at l can lie and create the instance $\vec{x}$, for which $M_2(\vec{x}) = l$. Moreover, since $\vec{x}$, $\vec{x}''$ are (1|2|3,4)-well-separated, $M_3(\vec{x}) = 3\lambda^2 + \lambda + 1$ and $l + \epsilon \leq 3\lambda^2 + \lambda + 1$, from 3.21, $M_3(\vec{x}) = l + \epsilon$. Since both $x''_3, x''_4 \in M(\vec{x})$, either agents 1 and 2 are served by the same facility of $M(\vec{x})''$ or agent 2 is served by the facility at l. In both cases, $SC[M(\vec{x})''] \geq \lambda$, which is a contradiction. ☐

3.4 Conclusion

In this chapter, we introduced the most important definitions and notations of Algorithmic Design, applied to K-Facility Locations Game, on of the most studied problem of this field. Although there have been many different and interesting results for the cases, where k < 3 , the impossibility result of [21] for $k \geq 3$ bounds our research on this topic. However, there are still ways to tackle this problem and produce interesting results, if we ignore the ”worst cases” and focus our interest in instances that are closer to the ”real-world”. We can choose to study only instances that are meaningful to be studied [46] or even allow ourselves to be aided by some external system, which can even guide our mechanism to choose a better clustering [1]

Chapter 4

Stability on Clustering and Facility Location Games

The most common approach to the design and analysis of computational problems is the worst-case analysis. Mechanism Design could not be an exception to that rule. Although this method provides the most complete measurement of a problem's difficulty, it bounds us to use the same algorithm, even if we only care for "special" cases of the problem that can be solved more optimally. Unfortunately, the majority of decision and optimization problems with some sort of practical use, usually fall into the class of NP-hard problems. However, the fact that these problems have practical use can assist us in correlating them to "real-world" instances, meaning that we can take advantage of these instances' properties. One problem that has "real-world" instances with very interesting properties, is the problem of *clustering*. Given a set of data points, our goal is to divide them into groups that make "sense", using a similarity function that makes this distinction. The main property, which we can use in the "real-world" application of clustering, is, that, probably the data points have an underlying structure, which we can exploit. The groups or clusters of these instances are well defined, giving us the option to not care for cases that deal with entangled clusters, as the worst-case analysis method would do. This can be summarized in the following phrase by Tim Roughgarden *"Clustering is hard, only when it doesn't matter"* [46]. We can apply this train of thought to the design of our algorithms, not only to their analysis, meaning that we can solve a modified version of the clustering problem (or any kind of optimization problem, with "real-world" significance), where we only care for inputs that satisfy the assumption of this underlying structure. For every other "not interesting" input we may produce a solution with "bad" approximation or not even produce a solution at all.

Bilu and Linial [13] were the first to suggest an approach aimed at taking advantage of this underlying structure. In particular, they introduced the notion of *stability*, which we will define later, they argued that instances in practice should be stable to small perturbations in the metric space and gave an efficient algorithm for clustering instances of the MAX-CUT problem that are stable to perturbations of size $O(n^{\frac{1}{2}})$. Awasthi, Blum and Sheffet [6] proved that any instance that is stable to as little as O(1) perturbations should be solvable in polynomial time for any center-based clustering objective (such as k-median, k-means and k-center). More specifically, they proposed an algorithm that finds the optimal clustering assuming only 3-stability for metric spaces and $2 + \sqrt{3}$ for general metrics. This algorithm is the popular Single-Linkage algorithm combined with dynamic programming. Later, Balcan and Liang [10] improved upon the above result to an algorithm that finds the optimal clustering in polynomial time, assuming $1 + \sqrt{2}$-stability. Finally, Angelidakis, Makarychev K., and Makarychev Y. [4] gave an exact algorithm for 2-perturbation stable instances of clustering problems with center-based objectives. This result was tight, since Balcan, Haghtlab, and White [8] have proven that no polynomial-time algorithm can solve (2 - ϵ)- perturbation stable instances of k-center unless NP = RP. There are also the results of [9], which deals with another kind of stability, called approximation stability. Finally, there has been success in the design of polynomial time exact algorithms for perturbation stable clustering instances in [35], [14], [37]

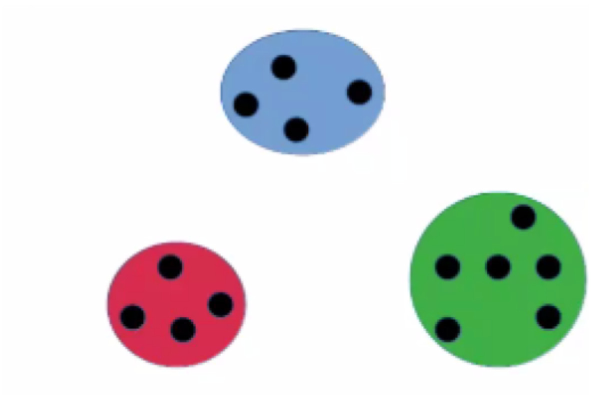
Our goal is to analyze the clustering problem inside the realm of algorithmic game theory, in our effort to circumvent the strong impossibility theorem [21], which means that the notions of *strategyproofness, connection cost, approximation ratio, social cost, maximum cost, etc* that were introduced in 3 still apply to our setting. However, there are still some preliminaries, unique to the clustering problem, that need to be added, such as the notions γ-perturbation and γ-stability, while also proving some important properties. After introducing these definitions, we will analyze two mechanisms that are designed to solve the Facility Location Game on γ-stable instances from [23]. The OPTIMAL mechanism is applied on instances with stability $\gamma \geq 2 + \sqrt{3}$, whose optimal clustering does not contain singleton clusters and ALMOSTRIGHTMOST is applied on instances with stability $\gamma \geq 5$, whose optimal clustering may contain singleton clusters.

4.1 Clustering and Stability

4.1.1 Definitions and Preliminaries

First of all, we will define the components of the classic clustering problem.

Definition 4.1 (Clustering Problem)**.** *An instance of a clustering problem is a tuple ((X,d),H,k) of a metric space (X,d), objective function H and integer number k > 1. The objective H is a function that, given a partition of X into k sets $C_1, ..., C_k$ and a metric d on X, returns a nonnegative real number, which we call the cost of the partition.*



Given an instance of a clustering problem ((X,d),H,k), our goal is to partition X into disjoint (nonempty) sets $C_1, ..., C_k$, so as to minimize $H(C_1, ..., C_k, d)$. Awasthi et al. [6] gave the following definition for *center-based* and *separable center-based objectives*.

Definition 4.2 (Center-based and Separable Objectives)**.** *A clustering objective is center-based if the optimal solution can be defined by k points $c_1, ..., c_k$ in the metric space, called centers, such that every data point is assigned to its nearest center. Such a clustering objective is separable if it further satisfies the following two conditions:*

- *The objective function value of a given clustering is either a (weighted) sum or the maximum of the individual cluster scores.*
- *Given a proposed single cluster, its score can be computed in polynomial time.*

The most well-studied and, perhaps, most interesting clustering objectives are *k-means*, *k-median*, and *k-center*. These objectives are defined as follows. Given a clustering $C_1, ..., C_k$, the objective is equal to the minimum over all choices of centers $c_1 \in C_1, ..., c_k \in C_k$ of the following functions:

$$H_{means}(C_1, ..., C_k; d) = \sum_{i=1}^{k} \sum_{u \in C_i} = d(u, c_i)^2$$

$$H_{median}(C_1, ..., C_k; d) = \sum_{i=1}^{k} \sum_{u \in C_i} = d(u, c_i)$$

$$H_{center}(C_1, ..., C_k; d) = max_{i \in 1,...,k}\{max_{u \in C_i}\{d(u, c_i)\}\}$$

It is evident that H_{means} objective is equivalent to the social cost and the H_{center} to the maximum cost in the Facility Location game.

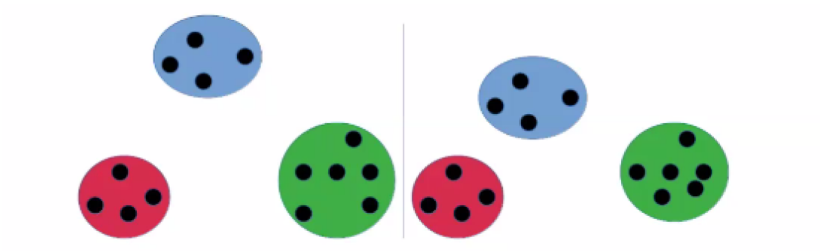
There have been a number of investigations of different notions of stability for the problem of clustering. Balcan, Blum and Gupta [9] introduced *approximation stability*. They considered a clustering instance to be approximation stable if good approximations to the given objective are guaranteed to be close, to a desired ground-truth partitioning. Formally, the target clustering of a k-median instance is (c, ϵ)-approximation stable if every c-approximate k-clustering is ϵ-accurate, meaning that it agrees with the target clustering on at least a $1 - \epsilon$ fraction of the data points. As mentioned above, we will use another notion of stability called *pertubation stability* introduced by Bilu and Linial and later used in [13] [6], [10], [4]. The intuition, behind this definition, lies in the fact, that in practice, distances between data points are typically just the result of some heuristic, like the Euclidean distance. Thus, unless, the optimal solution on the given distances is correct by luck, it likely will be correct on small perturbations of the given distances, as well.

Definition 4.3 (γ-perturbation). *Given a metric (S,d) and $\gamma \geq 1$, we say a function $d' : S \times S \to \mathbb{R}_{>0}$ is a γ perturbation of d, if for any x,y $\in S$, it holds that*

$$d(x, y)/\gamma \leq d'(x, y) \leq d(x, y)$$

Definition 4.4 (γ-stability). *Suppose we have a clustering instance composed of n points residing in a metric (S, d) and an objective function Φ we wish to optimize. We call the clustering instance γ-perturbation stable for Φ if for any d' which is an γ-perturbation of d, the (only) optimal clustering of (S, d') under Φ is identical, as a partition of points into subsets, to the optimal clustering of (S, d) under Φ.*

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 that we relax the conditions for stability to more natural ones, therefore allowing more instances in that class.



Definition 4.5 (Metric γ-perturbation and Metric γ-stability)**.** *Consider a metric space (X,d). We say that a metric d' is an* (a_1, a_2)*-metric perturbation of (X,d), for* $a_1, a_2 \geq 1$*, if* $a^{-1}d(u,v) \leq d'(u,v) \leq a_2 d(u;v)$ *for every* $u, v \in X$*. An instance ((X,d),H,k) is* (a_1, a_2)*-metric perturbation-stable if for every* (a_1, a_2)*-metric perturbation d' of d, the unique optimal clustering for ((X,d'),H,k) is the same as for ((X,d),H,k). We say that an instance ((X,d),H,k) is a-metric perturbation-stable if it is (a,1)-metric perturbation-stable.*

All γ-stable instances are essentially distinct clusters, whose intra-cluster distance can be quantified by the following property, which states that any point that belongs to a cluster assigned to it by the optimal clustering is at least γ times closer to its center than any other center of the optimal clustering.

Definition 4.6 (γ-center proximity)**.** *Let p* $\in$ *S be an arbitrary point, let* c_i* *be the center p is assigned to in the optimal clustering, and let* $c_j \neq c_i$ *be any other center in the optimal clustering. We say a clustering instance satisfies the* γ*-center proximity property if for any p it holds that :*

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

4.1.2 Properties Of Perturbation Stable Instances

The definitions of γ-stability and γ-pertubation produce some very interesting properties that each γ-stable instance must satisfy. We can easily prove that γ-center proximity property is implied on a γ-stable instance.

Proposition 4.7. *If a clustering instance satisfies* γ*-pertubation stability property, then it satisfies the* γ*-center proximity property*

Proof. Let $\gamma \geq 1$, C_i^* and C_j^* be any two clusters in the optimal clustering and pick any p $\in C_i^*$. Assume we blow up all the pairwise distances within cluster C_i^* by a factor of γ. As this is a legitimate perturbation of the metric, it still holds that the optimal clustering under this perturbation is the same as the original optimum. Hence, p is still assigned to the same cluster. Furthermore, since the distances within γ were all changed by the same constant factor, c_i^* will still remain an optimal center of cluster i. The same holds for cluster C_j^*. It follows that even in this perturbed metric, p prefers c_i^* to c_j^*. Hence

$$d(p, c_j^*) = d'(p, c_i^*) < d'(p, c_j^*) = d(p, c_j^*)$$

□

An immediate consequence of the above proposition is that γ-stable instances, with $\gamma \geq 2$, satisfy the weak γ-center proximity : For all clusters C_i and C_j, with i $\neq$ j, and all locations $x \in C_i$ and $y \in C_j$, $d(x, y) > (\gamma - 1)d(x, c_i)$

Proposition 4.8. *Let $\gamma \geq 2$ and let $\vec{x}$ be any γ-stable instance, with unique optimal clustering $(C_1, ..., C_k)$ and optimal centers $(c_1, ..., c_k)$. Then, for all clusters C_i and C_j, with $i \neq j$, and all locations $x \in C_i$ and $y \in C_j$, $d(x, y) > (\gamma - 1)d(x, c_i)$*

Proof. Denote by c_j the center of the cluster that y belongs to. Now, consider two cases

- Case (a): $d(y, c_j) \geq d(x, c_i)$. In this case, by triangle inequality, we get that $d(x, y) \geq d(y, c_i) - d(x, c_i)$. Since instance is stable to γ-perturbations, we have that $d(y, c_i) > \gamma d(y, c_j)$. Hence we get that

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

.

- Case (b): $d(y, c_j) < d(x, c_i)$. Again by triangle inequality, we get that

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

□

Finally, we state the cornerstone of our stability properties, the cluster-separation property. For any γ-stable instance, the distance between points that belong to different clusters has a lower bound.

Lemma 4.9 (Cluster-Separation Property)**.** *Let $(C_1, ..., C_k)$ be the optimal clustering of γ-stable instance with $\gamma \geq 2$. Let $x_i, x_i' \in C_k$ and $x_j \in C_{k'}$, with $i \neq j$ then :*

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

.

Proof. Let c_k be the center of C_k and $c_{k'}$ be the center of $C_{k'}$. Since γ-stability implies γ-center proximity we have that :

$$d(x_i, c_{k'}) > \gamma d(x_i, c_k)$$

From triangle inequality, we get the following statement:

$$\begin{aligned} d(x_i, c_k) + d(c_k, c_{k'}) > \gamma d(x_i, c_k) \implies \\ d(c_k, c_{k'}) > (\gamma - 1)d(x_i, c_k) \end{aligned}$$

We also get from triangle inequality, the following:

$$d(c_k, c_{k'}) < d(c_k, x_i) + d(x_i, x_j) + d(x_j, c_{k'})$$

And since γ-stability implies weak γ-center proximity, we get that :

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

and

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

So we conclude that:

$$d(c_k, c_{k'}) < \frac{\gamma + 1}{\gamma - 1} d(x_i, x_j)$$

Finally, again from triangle inequality:

$$\begin{aligned}
&d(x_i, x'_i) < d(x_i, c_k) + d(c_k, x'i) \implies \\
&d(x_i, x'_i) < \frac{1}{\gamma - 1} d(x_i, x_j) + \frac{1}{\gamma - 1} d(c_k, c_{k'}) \implies \\
&d(x_i, x'_i) < \frac{1}{\gamma - 1} d(x_i, x_j) + \frac{\gamma + 1}{(\gamma - 1)^2} d(x_i, x_j)) \implies \\
&d(x_i, x_j) > \frac{(\gamma - 1)^2}{2\gamma} d(x_i, x'_i)
\end{aligned}$$

□

Those three properties, 4.7, 4.8, 4.9, provide us with useful inequalities that define inter and intra-cluster distances of our instance. We can then create algorithms, that exploit these bounds and transform our instance to more manageable forms, such as trees. This is the core idea behind the algorithm of *single-link++*.

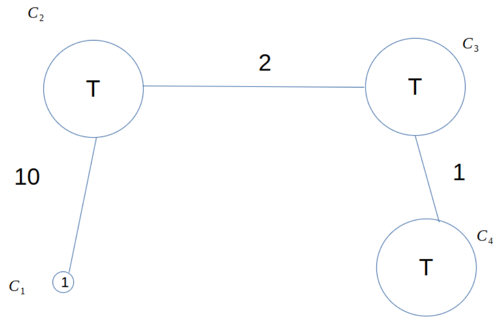
4.1.3 Single-link++

Single-link++ is a clustering algorithm applied to γ-stable instances for the H_{means} objective. It recovers the optimal clustering in polynomial time. To fully appreciate its brilliance, we must first analyze another simpler algorithm called *single-link clustering*.

Single-link clustering is a widely known clustering algorithm. The idea is to think of the input metric space (X,d) as a complete graph, with vertices X and edge weights given by d. The algorithm runs Kruskal's minimum spanning tree algorithm, except it stops when there are k connected components, where k is the desired number of clusters, meaning that we skip the last k-1 iterations of Kruskal. Although single-link clustering is very intuitive in its analysis, it has shortcomings, since the objective of H_{means} plays no part in the algorithm's cluster-creation process, making it possible to output a sub-optimal clustering of an instance. For example, suppose that after some iterations of Kruskal's algorithm, we found ourselves with four connected components, one with 1 data point in it (C_1) and three with T » 1 data points in them (C_2, C_3, C_4). Suppose that the T data points in C_2, C_3, C_4 are located in the corresponding centers c_2, c_3, c_4. If our goal is to output four clusters, our job is done and we have indeed found the optimal 4-clustering. If our goal is to output three clusters, we need to make another iteration of the Kruskal algorithm and since C_3 and C_4 are the closest clusters, they merge and we output a sub-optimal clustering.

The issue of single-clustering was that it paid no mind to the objective cost function. *Single-link++* is a more sophisticated version of single-clustering.

Single Clustering Counter Example



Mechanism 8 Single-link++

Input: metric space (X,d)
Create a complete graph with vertices given by X and edge weight given by d
Run Kruskal's algorithm until completion to compute the minimum spanning tree T of the complete graph induced by (X, d).
Among all $\binom{n-1}{k-1}$ subsets of k−1 edges of T and the induced k-clusterings (with one cluster per connected component), compute the one with the minimum k-median objective function value.
Output the clustering

First, we need a way to verify that our algorithm not only has a way to validate the existence of an optimal clustering but can also produce it as an output.

Lemma 4.10. *Single-link++ recovers the optimal solution of a k-median instance (X, d) if and only if every optimal cluster C_i^* induces a connected subgraph of the minimum spanning tree.*

Proof. The single-link++ algorithm can only output a k-clustering obtained by removing k − 1 edges from the MST T. Such an output necessarily produces clusters that are connected subgraphs of T. Thus if some optimal cluster C_i^* is not a connected subgraph of T, the single-link++ algorithm has no chance of finding it. Conversely, every partition of X into k (non-empty) connected subgraphs of T can be obtained by deleting k−1 edges from T (namely, every edge with an endpoint in two different optimal clusters). Since the single-link++ algorithm explicitly optimizes over k-clusterings of this form, if the optimal algorithm has this form, then the algorithm will recover it. □

We now have a method to distinguish optimal clusterings in our induced MST. We only have to apply our core stability properties on the induced instance to receive the following result:

Theorem 4.11. *In every 2-perturbation-stable k-median instance, the single-link++ algorithm recovers the optimal solution (in polynomial time).*

Proof. It is enough to show that the correctness condition in 4.10 holds—that is, in every 2-perturbation-stable H_{median} instance, every optimal cluster C_i^* induces a connected subgraph of T. We proceed by contradiction. If not, there is a point x ∈ C_i^* such that the (unique) c_i-x path in T concludes with the edge (y, x) with y ∉ C_i^*. At the time (y, x) was added by Kruskal's algorithm, x and c_i were in different connected components (otherwise the addition of (y, x) would have created a cycle). Thus, Kruskal's algorithm also had the option of including the edge (x, c_i) instead. Since the algorithm chose (y, x) over (x, c_i), d(x, y) $\geq$ d(x, c_i). But then x is as close to y $\notin C_i^*$ as its own center, contradicting the weak 2-center proximity property. □

To the extent that we believe that "real-world" clustering instances with "meaningful solutions" are 2-perturbation-stable, 4.11 gives a formal sense in which clustering is hard only when it does not matter. It is a largely open research direction to prove robust versions of Theorem 4.11, where perturbations can cause a small number of points to switch clusters, while still preserving the optimal clustering of the instance, a property called *approximation stability*.

4.2 Stability and Facility Location Games

It is evident that our research on Facility Location Games can benefit from the many algorithmic approaches to the clustering problem. We can envision the Facility Location Games as a clustering problem, whose input data is not to be completely trusted, adding an extra layer of difficulty. However, it seems that the notion of stability can be applied here in the same context as in the clustering problem and act as a deterrent to any agent's desire to gain from their lie. We will expand the definitions of γ-pertubation and γ-stability on the Facility Location Games setting, as well as modify the most basic properties of stability, to suit the mechanisms that will solve the Facility Location Games on γ-stable instances.

4.2.1 Definitions and Preliminaries

Facility Location Game is essentially a clustering problem on the line with data points that can not be trusted. Since our goal is to implement clustering algorithms, utilizing the properties of stability, on the problem of Facility Location, we need to define clusterings on the line.

Definition 4.12 (Clusterings in Facility Location Games)**.** *A clustering (or k-clustering, if k is not clear from the context) of an instance $\vec{x}$ is any partitioning $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$ into k sets of consecutive agent locations. We index clusters from left to right. I.e., $C_1 = \{x_1, ..., x_{|C_1|}\}, C_1 = \{x_1, ..., x_{|C_1|}\}$, and so on. We refer to a cluster C_i that includes only one agent (i.e., with $|Ci| = 1$) as a singleton cluster. We sometimes use $(\vec{x}, \vec{C})$ to highlight that we consider $\vec{C}$ as a clustering of instance $\vec{x}$.*

Two clusters C and C' are identical, denoted C = C', if they include the exact same locations. Two clusterings $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$ and $\vec{Y} = (Y_1, ..., Y_k)$ of an instance $\vec{x}$ are the same, if $C_i = Y_i$, for all i $\in$ [k]. Abusing the notation, we say that a clustering $\vec{C}$ of an instance $\vec{x}$ is identical to a clustering $\vec{Y}$ of a γ -perturbation $\vec{x}'$ of $\vec{x}$, if $|C_i| = |Y_i|$, for all $i \in [k]$.

We let $x_{i,l}$ and $x_{i,r}$ denote the leftmost and the rightmost agent of each cluster C_i. In this setting, we will be more interested in the leftmost and rightmost agents of each cluster $C_i(x_{i,l}, x_{i,r})$, instead of the leftmost and rightmost agents of the whole instance - $\vec{x}$. Under this notation, $x_{i-1,r} < x_{i,l} \leq xi, r <$

$x_{i+1,l}$, for all $i \in \{2, ...k-1\}$. Exploiting the linearity of instances, we extend this notation to refer to other agents by their relative location in each cluster. Namely, $x_{i,l+1}$ (resp. $x_{i,r-1}$) is the second agent from the left (resp. right) of cluster C_i.

The diameter of a cluster C_i is $D(C_i) = d(x_{i,l}, x_{i,r})$. The distance of clusters C_i and C_j is $d(C_i, C_j) = min_{x \in C_i, y \in C_j}\{d(x, y)\}$, i.e., the minimum distance between a location $x \in C_i$ and a location $y \in C_j$. A k-facility locations (or k-centers) profile $\vec{c} = (c_1, ..., c_k)$ induces a clustering $\vec{C} = (C_1, ..., C_k)$ of an instance $\vec{x}$ by assigning each agent / location x_j to the cluster C_i with facility c_i closest to x_j. Formally, for each $i \in [k]$, $C_i = \{x_j \in \vec{x} : d(x_j, c_i) = d(x_j, c_i)\}$. The optimal clustering of an instance $\vec{x}$ is the clustering of $\vec{x}$ induced by the facility locations profile with minimum social cost. The social cost of a clustering $\vec{C}$ induced by a k-facility locations profile $\vec{c}$ on an instance $\vec{x}$ is simply $SC(\vec{x}, \vec{c})$ i.e., the social cost of $\vec{c}$ for $\vec{x}$. We sometimes refer to the social cost $SC(\vec{x}, \vec{C})$ of a clustering $\vec{C}$ for an instance $\vec{x}$, without any explicit reference to the corresponding facility locations profile. Then, we refer to the social cost $SC(\vec{x}, \vec{c})$, where each facility c_i is located at the median location of C_i (the left median location of C_i, if $|Ci|$ is even).

We often consider certain structural changes in a clustering due to agent deviations. Let $\vec{C}$ be a clustering of an instance $\vec{x}$, which due to an agent deviation, changes to a different clustering $\vec{C}'$. We say that cluster C_i is *split* when $\vec{C}$ changes to $\vec{C}'$, if not all agents in C_i are served by the same facility in $\vec{C}'$, We say thatC_i is merged in $\vec{C}'$, if all agents in C_i are served by the same facility, but this facility also serves in $\vec{C}'$ some agents not in C_i.

We have defined γ-perturbation and γ-stability on the many available choices of metric, d. Since the Facility Location Game on the line is using the Euclidean Distance as a metric - $|\cdot|$, we can change the definitions to be more suited to our problem. Namely, a γ-perturbation $\vec{x}'$ of an instance $\vec{x}$ is obtained by moving a subset of pairs of consecutive locations closer by a factor at most $\gamma \geq 1$. A k-Facility Location instance $\vec{x}$ is γ-stable, if $\vec{x}$ and any γ-perturbation $\vec{x}'$ of $\vec{x}$ admit the same unique optimal clustering.

Definition 4.13 (Linear γ-perturbation)**.** *Let $\vec{x} = (x_1, .., x_n)$ be a locations profile. A locations profile $\vec{x}' = (x_1, .., x_n)$ is a γ-perturbation of $\vec{x}$, for some $g \geq 1$, if $x'_1 = x_1$ and for every $i \in [n-1]$, it holds that*

$$d(x_i, x_{i+1})/\gamma \leq d(x'_i, x'_{i+1}) \leq d(x_i, x_{i+1})$$

Definition 4.14 (Linear γ-stability)**.** *A k-Facility Location instance $\vec{x}$ is γ-pertubation stable (or simply, γ-stable), if $\vec{x}$ has a unique optimal clustering $(C_1, ..., C_k)$ and every γ-perturbation $\vec{x}'$ of $\vec{x}$ has the same unique optimal clustering $(C_1, ..., C_k)$*

Essentially, the above notion of linear perturbation stability is a natural adaptation of the *metric stability* [4], to the line.

The definition of γ-center proximity can be adapted to the following:

Definition 4.15 (linear γ-center proximity)**.** *Let $\vec{x}$ be our instance and $p \in \vec{x}$ be an arbitrary agent location. Also, let c_i* be the center, p is assigned to in the optimal clustering, and let $c_j \neq c_i$ be any other center in the optimal clustering. We say a clustering instance satisfies the γ-center proximity property if for any p it holds that :*

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

4.2.2 Properties Of Perturbation Stable Instances on the Facility Location Games

We can now adapt the three basic properties of stability 4.7, 4.8, 4.9, to our own linear stability. Linear γ-center stability implies linear γ-center proximity and subsequently linear weak γ-center proximity

Proposition 4.16 (linear weak γ-center proximity)**.** *Let $\gamma \geq 2$ and let $\vec{x}$ be any γ-stable instance, with unique optimal clustering $(C_1, ..., C_k)$ and optimal centers $(c_1, ..., c_k)$. Then, for all clusters C_i and C_j, with i ≠ j, and all locations $x \in C_i$ and $y \in C_j$, $d(x, y) > (\gamma - 1)d(x, c_i)$*

Proof. Denote by c_j the center of the cluster that y belongs to. Now, consider two cases

- Case (a): $d(y, c_j) \geq d(x, c_i)$. In this case, by triangle inequality, we get that $d(x, y) \geq d(y, c_i) - d(x, c_i)$. Since instance is stable to γ-perturbations, we have that $d(y, c_i) > \gamma d(y, c_j)$. Hence we get that

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

.

- Case (b): $d(y, c_j) < d(x, c_i)$. Again by triangle inequality, we get that

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

□

Next, we state the cornerstone of our stability properties, the cluster-separation property. For any γ-stable instance there is a certain distance that each cluster needs to have from its neighbors, which is larger than their diameters

Lemma 4.17 (Cluster-Separation Property)**.** *For any γ-stable instance on the line with optimal clustering $(C_1, ..., C_k)$ and all clusters C_i and C_j, with i ≠ j, $d(C_i, C_j) > \frac{(\gamma-1)^2}{2\gamma} max D(C_i), D(C_j)$.*

Proof. Obviously,it suffices to prove the lemma for two consecutive clusters C_i , C_{i+1}. We recall that $d(C_i, C_{i+1}) = d(x_{i,r}, x_{i+1,l})$ and we assume that $D(C_i) \geq D(C_{i+1})$. We will divide the proof into three cases.

- Case 1 : C_i singleton. $D(C_i) = 0$ and the lemma holds trivially.
- Case 2 : $|C_i| = 2$

  Since $C_i = x_{i,l}, x_{i,r}$, wlog we consider $x_{i,l} = c_i$ to be the center of C_i. We then have:

$$D(C_i) = d(x_{i,l}, x_{i,r} = d(c_i, x_{i,r}) \implies (4.15)$$
$$D(C_i) < \frac{1}{(\gamma - 1)} d(x_{i,r}, x_{i+1,l}) \implies$$
$$D(C_i) < \frac{1}{(\gamma - 1)} d(C_i, C_{i+1}) \implies$$
$$d(C_i, C_{i+1}) > (\gamma - 1)D(C_i)$$

For any $\gamma \geq 1$,

$$\gamma - 1 \geq \frac{\gamma^2+1}{2\gamma} - 1 \implies d(C_i, C_{i+1}) > \frac{(\gamma-1)^2+1}{2\gamma} D(C_i)$$

- Case 3 : $|C_i| \geq 3$

  We have that $x_{i,l} \leq c_i \leq x_{i,r}$. We will use $\beta \in (0,1]$, which quantifies how close c_i is to C_i's extreme points and to the closest point of C_{i+1}. So we denote $d(c_i, x_{i,l}) = \beta D(C_i)$ and $d(c_i, x_{i,r}) = (1-\beta) D(C_i)$. We have that:

$$\begin{aligned}
d(x_{i,l}, x_{i+1,l}) &\geq d(x_{i,l}, c_{i+1}) - d(x_{i+1,l}, c_{i+1}) \implies (4.15) \\
d(x_{i,l}, x_{i+1,l}) &> \gamma d(x_{i,l}, c_i) - \frac{d(x_{i+1,l}, c_i)}{\gamma} \implies \\
d(x_{i,l}, x_{i+1,l}) &> \gamma d(x_{i,l}, c_i) - \frac{d(x_{i+1,l}, x_{i,l}) - d(c_i, x_{i,l})}{\gamma} \implies \\
d(x_{i,l}, x_{i+1,l}) &> \frac{\gamma^2+1}{\gamma+1} d(c_i, x_{i,l}) \implies \\
D(C_i) + d(C_i, C_{i+1}) &> \frac{\gamma^2+1}{\gamma+1} \beta D(C_i) \implies \\
d(C_i, C_{i+1}) &> (\frac{\beta\gamma^2+1}{\gamma+1} - 1) D(C_i)
\end{aligned}$$

  From 4.15 we have that $d(x_{i,r}, x_{i+1,l}) > (\gamma-1) d(x_{i,r}, c_i)$ and $d(x_{i,r}, c_i) = (1-\beta) D(C_i)$, so $d(C_i, C_{i+1}) > (\gamma-1)(1-\beta) D(C_i)$. Finally, we have that :

$$d(C_i, C_{i+1}) > max\{(\gamma-1)(1-\beta), (\frac{\beta(\gamma^2+1)}{\gamma+1} - 1)\} D(C_i)$$

  We now observe that for any fixed $\gamma > 1$, the term $(\frac{\beta(\gamma^2+1)}{\gamma+1} - 1)$ is increasing for all $\beta > 0$, while the term $(\gamma-1)(1-\beta)$, is decreasing for all $\beta \in (0,1]$. Hence,for any fixed $\gamma > 1$, the minimum value of the max in $max\{(\gamma-1)(1-\beta), (\frac{\beta(\gamma^2+1)}{\gamma+1} - 1)\} D(C_i)$ is achieved when β satisfies:

$$(\gamma-1)(1-\beta) = \frac{\beta(\gamma^2+1)}{\gamma+1} - 1$$

  Solving for β, we get that: $\beta = \frac{1}{2} + \frac{1}{2\gamma}$, with $\beta \in (1/2, 1]$, when $\gamma \geq 1$. So if we substitute $\beta = \frac{1}{2} + \frac{1}{2\gamma}$ in $d(C_i, C_{i+1}) > max\{(\gamma-1)(1-\beta), (\frac{\beta(\gamma^2+1)}{\gamma+1} - 1)\} D(C_i)$, we get our result.

□

We can now expand our existing properties and prove more lemmas and propositions, which play an integral part in the development of mechanisms OPTIMAL and ALMOSTRIGHTMOST, that solve the k-Facility Location Games for γ-stable instances. A natural consequence of 4.14 is that we can treat stability factors multiplicatively:

Proposition 4.18. *Every α-perturbation followed by a β-perturbation of a location's profile can be implemented by an $(\alpha\beta)$-perturbation and vice versa. Hence, a γ-stable instance remains (γ / γ')-stable after a γ'-perturbation, with $\gamma' < \gamma$, is applied to it.*

If we set $\gamma \geq 2 + \sqrt{3}$, we get the following corollary from 4.17:

Corollary 4.19. *Let $\gamma \geq 2 + \sqrt{3}$ and let $\vec{x}$ be any γ-stable instance with unique optimal clustering $(C_1, ..., C_k)$. Then, for all clusters C_i and C_j, with $i \neq j, d(C_i, C_j) > maxD(C_i), D(C_j)$.*

We can receive the following proposition, as a direct consequence of the Cluster Separation Property.

Proposition 4.20. *Let $\vec{x}$ a k-Facility Location with a clustering $\vec{C} = (C_1, ..., C_k)$ such that for any two clusters C_i and C_j, $maxD(C_i), D(C_j) < d(C_i, C_j)$. Then, if in the optimal clustering of $\vec{x}$, there is a facility at the location of some $x \in C_i$, no agent in C_i is served by a facility at $x_j \notin C_i$.*

Lastly, from [23] we will state the so-called *no direct improvement from singleton deviation properties*, for $\gamma \geq 3$. Namely, we show that in any 3-stable instance, no agent deviating to a singleton cluster in the optimal clustering of the resulting instance can improve his connection cost through the facility of that singleton cluster.

Lemma 4.21 (no direct improvement from singleton deviation property, $\gamma \geq 3$)**.** *Let $\vec{x}$ be a γ-stable instance with $\gamma \geq 3$ and optimal clustering $(\vec{C} = (C_1, ..., C_k)$ and cluster centers $(c_1, ..., c_k)$, and let an agent $x_i \in C_i \;\; \{c_i\}$ and a location x' such that x' is a singleton cluster in the optimal clustering of the resulting instance $(\vec{x}_{-i}, x')$. Then, $d(x_i, x') > d(x_i, c_i)$*

The following shows that for 5-stable instances $\vec{x}$, an agent cannot form a singleton cluster, unless he deviates by a distance larger than the diameter of his cluster in $\vec{x}$'s optimal clustering.

Lemma 4.22 (no direct improvement from singleton deviation property, $\gamma \geq 5$)**.** *Let $\vec{x}$ be a γ-stable instance with $\gamma \geq 5$ and optimal clustering $(\vec{C} = (C_1, ..., C_k)$. Let $x_i \in C_i \;\; \{c_i\}$ be any agent and x' any location such that x' is a singleton cluster in the optimal clustering of the resulting instance $(\vec{x}_{-i}, x')$. Then, $d(x_i, x') > D(C_i)$*

4.3 Optimal Solution for a special case of $2 + \sqrt{3}$-stable instances

---

Mechanism 9 OPTIMAL : Deterministic mechanism on $2 + \sqrt{3}$-stable instances without Singleton Deviations.

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ Compute the optimal clustering $(C_1, ..., C_k)$. Let c_i be the left median point of each cluster C_i.
if ($\exists i \in [k]$ with $|C_i| = 1$) or ($\exists i \in [k-1]$ with $max\{D(C_i), D(C_{i+1})\} \leq d(C_i, C_{i+1})\}$) **then**
  Output: "FACILITIES ARE NOT ALLOCATED".
else
  Output: The k-facility allocation $(c_1, ..., c_k)$
end if

We will introduce the OPTIMAL mechanism, whose goal is to minimize the social cost of $2 + \sqrt{3}$-stable instances$\vec{x}$, $SC(\vec{x})$, on k-Facility Location Game, while maintaining the strategyproofness property. One key note is that the optimal clustering of $\vec{x}$ does not include any singleton clusters. This mechanism computes the optimal clustering $\vec{C} = (C_1, ..., C_k)$ and checks for two transgressions: The existence of a singleton cluster in the optimal clustering $\vec{C}$ and the violation of the cluster separation property. If any of these two transgressions exist, we will output no facilities. In any other case, our output will be the center $(c_1, ..., c_k)$ of each cluster of $\vec{C}$.

4.3.1 Approximation Ratio

The computation of the approximation ratio is straightforward since the mechanism only outputs the optimal clustering $\vec{C} = (C_1, ..., C_k)$.

4.3.2 Strategyproofness

Our proof plan will be to show that if an agent i deviates to any location $y \neq x_i$, the new clustering that will be produced (which will not contain a singleton cluster) will either violate the Cluster Separation Property or the new clustering will not provide a closer facility to the true location of agent i. Formally, for any agent i and any location y, let $\vec{Y}$ be the optimal clustering of the instance $\vec{y} = (\vec{x_{-i}}, y)$ resulting from the deviation of agent i from x_i to y. Then, if y does not form a singleton cluster in $(\vec{y}, \vec{Y})$ either $d(x_i, \vec{Y}) > d(x_i, \vec{C})$, or there is an $i \in [k-1]$ for which max$D(Y_i), D(Y_{i+1}) \geq d(Y_i, Y_{i+1})$

So, we let $x_i \in Ci$ deviate to a location y, resulting in $\vec{y} = (\vec{x_{-i}}, y)$ with optimal clustering $\vec{Y}$. Since y is not a singleton cluster, it is clustered with agents belonging in one or two clusters of $\vec{C}$, say either in cluster C_j or in clusters C_{j-1} and C_j. By optimality of $\vec{C}$ and $\vec{Y}$, the number of facilities serving $C_{j-1} \cup C_j \cup$ y in $(\vec{y}, \vec{Y})$ is no less than the number of facilities serving $C_{j-1} \cup C_j$ in $(\vec{x}, \vec{C})$. Hence, there is at least one facility in either C_{j-1} or C_j in $\vec{Y}$.

Wlog, suppose that a facility is allocated to an agent in C_j in $(\vec{y}, \vec{Y})$. By 4.19 and 4.20, no agent in C_j is served by a facility in $\vec{x}$ C_j in $\vec{Y}$. Thus we get the following cases about what happens with the optimal clustering $\vec{Y}$ of instance $\vec{y} = (\vec{x_{-i}}, y)$:

- **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 < $x_{i,l}$ and y only serves $x_{i,l}$ or $x_{j,l} \leq y \leq x_{j,r}$).
  - **Case 2b**: In $\vec{Y}$, y serves agents that belong in both C_{j-1} and C_j.

First, we will deal with the cases that violate the Cluster Separation Property - **Case1a** and **Case2a**, i.e. the cases where some agents of C_j are clustered with agents of $\vec{x}\backslash C_j$ in $\vec{Y}$. By hypothesis, there are agents z,w $\in C_j$, that in $\vec{Y}$ are clustered with agents p $\in C_l, l \neq j$. Suppose that in $\vec{Y}$, agent z is clustered in C'_j and agents z,p are clustered in C'_{j+1}, which is in agreement with our initial hypothesis. We will try to produce a contradiction, using the known properties on intra-cluster and inter-cluster distances. Since our mechanism produces a clustering $\vec{Y}$, that clustering must have the Cluster Separation Property. From 4.19 in $\vec{Y}$ we know that :

$$d(C'_j, C'_{j+1}) > D(C'_{j+1}) \implies$$

$$d(w, z) > d(w, p)$$

From 4.19 in $\vec{C}$ we know that :

$$d(C_j, C_l) > D(C_j) \implies$$

$$d(w, p) > d(w, z)$$

Hence, we have our contradiction.

Finally, we will deal with the cases in the new clustering $\vec{Y}$, does not improve the connection cost of agent i - **Case1b** and **Case2b**, i.e. the cases where $\vec{Y}$ allocates facilities to agents of C_j (between $x_{j,l}$ and $x_{j,r}$). We show that the cost of the original clustering $\vec{C}$ is less than the cost of clustering $\vec{Y}$ in $\vec{y}$. Hence, mechanism Optimal would also select clustering $\vec{C}$ for $\vec{y}$, which would make x_i's deviation to y non-profitable. In particular, it suffices to show that:

$$\begin{aligned} SC(\vec{y}, \vec{C}) < SC(\vec{y}, \vec{Y}) \implies \\ SC(\vec{x}, \vec{C}) + d(y, \vec{C}) - d(x_i, \vec{C}) < SC(\vec{x}, \vec{Y}) + d(y, \vec{Y}) - d(x_i, \vec{Y}) \implies \\ d(y, \vec{C}) - d(y, \vec{Y}) < SC(\vec{x}, \vec{Y}) - SC(\vec{x}, \vec{C}) + d(x_i, \vec{C}) - d(x_i, \vec{Y}) \end{aligned}$$

Since x_i's deviation to y is profitable, $d(x_i, \vec{C}) - d(x_i, \vec{Y}) > 0$. Hence, it suffices to show that:

$$\begin{aligned} d(y, \vec{C}) - d(y, \vec{Y}) \leq SC(\vec{x}, \vec{Y}) - SC(\vec{x}, \vec{C}) \implies \\ d(y, \vec{C}) - d(y, \vec{Y}) \leq SC(C_j, \vec{Y}) - SC(C_j, \vec{C}) + SC(\vec{x}\backslash C_j, \vec{Y}) - SC(\vec{x}\backslash C_j, \vec{C}) \end{aligned} \quad (4.1)$$

Note that in case 2a, y can also be located outside of C_j and serve only $x_{j,l}$. We can treat this case as Case 1a since it is equivalent to placing the facility on $x_{i,l}$ and serving y from there. In Case 1a and Case 2a, we note that (3.1) holds if the clustering $\vec{Y}$ allocates a single facility to agents in $C_j \cup y$, because the facility is allocated to the median of $C_j \cup y$, hence $d(y, \vec{C}) - d(y, \vec{Y}) = SC(C_j, \vec{Y}) - SC(C_j, \vec{C})$, while $SC(\vec{x}\backslash C_j, \vec{Y}) - SC(\vec{x}\backslash C_j, \vec{C}) \geq 0$, since $\vec{C}$ is optimal for $\vec{x}$. So, we focus on the most interesting case where the agents in$C_j \cup$ y are allocated at least two facilities.

We will, now, introduce a perturbation to the instance for reasons, that will be explained later. Consider the valid γ-perturbation of the original instance $\vec{x}$ where all distances between consecutive agent pairs to the left of C_j (i.e. agents $x_1, x_2, ..., x_{j-1,r}$ and between consecutive agent pairs to the right of C_j

(i.e. agents $x_{j+1,l}, ..., x_{k,r}$ are scaled down by γ. By stability, the clustering $\vec{C}$ remains the unique optimal clustering for the perturbed instance $\vec{x}'$. Moreover, since agents in $\vec{x} \setminus C_j$ are not served by a facility in C_j in $\vec{C}$ and $\vec{Y}$, and since all distances outside C_j are scaled down by γ, while all distances within C_j remain the same, the cost of the clusterings $\vec{C}$ and $\vec{Y}$ for the perturbed instance $\vec{x}$' is SC(C_j, $\vec{C}$) + SC($\vec{x} \setminus C_j$, $\vec{C}$) / γ and SC(C_j, $\vec{Y}$) + SC($\vec{x} \setminus C_j$, $\vec{Y}$) / γ, respectively. Using SC($\vec{x}$', $\vec{C}$) < SC($\vec{x}'$, $\vec{Y}$) and $\gamma \geq 2$, we obtain:

$$SC(C_j, \vec{Y}) - SC(C_j, \vec{C}) < \frac{1}{\gamma}(SC(\vec{x}\backslash C_j, \vec{Y}) - SC(\vec{x}\backslash C_j, \vec{C})) \tag{4.2}$$

$$SC(C_j, \vec{Y}) - SC(C_j, \vec{C}) \leq (1 - \frac{1}{\gamma})(SC(\vec{x}\backslash C_j, \vec{Y}) - SC(\vec{x}\backslash C_j, \vec{C})) \tag{4.3}$$

Moreover, if $C_j\cup$ y is served by at least two facilities in $\vec{Y}$, the facility serving y (and some agents of C_j) is placed at the median location of $\vec{Y}$'s cluster that contains y. Wlog, we assume that y lies on the left of the median of C_j. Then, the decrease in the cost of y due to the additional facility in $\vec{Y}$ is equal to the decrease in the cost of $x_{i,l}$ in $\vec{Y}$, which bounds from below the total decrease in the cost of C_j due to the additional facility in $\vec{Y}$. Hence,

$$d(y, \vec{C}) - d(y, \vec{Y}) \leq SC(C_j, \vec{C}) - SC(C_j, \vec{Y}) \tag{4.4}$$

We can now construct the following inequality, one from (3.2) and (3.4)

$$d(y, \vec{C}) - d(y, \vec{Y}) \leq \frac{1}{\gamma}(SC(\vec{x}\backslash C_j, \vec{Y}) - SC(\vec{x}\backslash C_j, \vec{C})) \tag{4.5}$$

If we combine (3.3) and (3.5), we get (3.1) and our proof is over.

4.4 A Deterministic Mechanism Resistant to Singleton Deviations 5-stable instances

Mechanism 10 ALMOSTRIGHTMOST: Deterministic Mechanism Resistant to Singleton Deviations 5-stable instances.

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ Find the optimal clustering $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$.
if there are two consecutive clusters C_i and C_{i+1} with $max\{D(C_i), D(C_{i+1})\} \geq d(C_i, C_{i+1})\}$)
then
    Output: ”FACILITIES ARE NOT ALLOCATED”.
    for i $\in$ 1, . . . ,k **do**
        if $|C_i|$ > 1 **then**
            Allocate a facility to the location of the second rightmost agent of C_i, i.e., $c_i \leftarrow x_{i,r-1}$.
        else
            Allocate a facility to the single agent location of C_i : $c_i \leftarrow x_{i,l}$
        end if
    end for
end if
Output: The k-facility allocation $\vec{c} = (c_1, ..., c_k)$

Now, we present the ALMOSTRIGHTMOST mechanism, whose goal is to minimize the social cost $SC(\vec{x})$ of 5-stable instance $\vec{x}$, whose optimal clustering may include singleton clusters, while maintaining the strategyproofness property. This mechanism computes the optimal clustering $\vec{C} = (C_1, ..., C_k)$ and checks for only one transgressions : The violation of the cluster separation property. If that transgression exists, we will output no facilities. In any other case, we allocate facilities near the edge of each optimal cluster, so that cluster merging will be discouraged by the facility allocation rule. We will end up with a significantly larger approximation and a requirement for larger stability, in order to achieve strategyproofness.

4.4.1 Approximatio Ratio

The approximation ratio of (n − 2)/2 follows directly from the fact that the mechanism allocates the facility to the second rightmost agent of each non-singleton optimal cluster.

4.4.2 Strategyproofness

Our proof plan will be to show that if an agent i deviates to any location $y \neq x_i$, the new clustering that will be produced (which will not contain a singleton cluster) will either violate the Cluster Separation Property or the new clustering will not provide a closer facility to the true location of agent i.

Let $\vec{x}$ denote the true instance and $\vec{C} = (C_1, ..., C_k)$ its optimal clustering. We consider an agent $x_i \in C_j$ deviating to location y, resulting in an instance $\vec{y} = (\vec{x}_{-i}, y)$ with optimal clustering $\vec{Y}$. Agent x_i's cost is at most D(C_j). Thus we get the following cases about what happens with the optimal clustering $\vec{Y}$ of instance $\vec{y} = (\vec{x_{-i}}, y)$:

- **Case 1:** The agents in C_j are clustered together in $\vec{Y}$ and y is allocated a facility with d(y, x_i) < d(x_i, $x_{i,r-1}$) $\leq$ D(C_j) ($x_{i,r-1}$is the location of x_i's facility, when he is truthful).
  - **Case 1a:** y is a singleton cluster and d(y, x_i) < D(C_j). For 5-stable instances, 4.22 implies that $x_i \in C_j$ has to move by at least D(C_j) to become a singleton cluster, a contradiction.
  - **Case 1b:** y is the second rightmost agent of a cluster C_j in $\vec{y}, \vec{Y}$. Then, the agent x_i can gain only if d(y, x_i) < D(C_j). In Case 1, the agents in Cj are clustered together in Y .If y < xi, y must be the second rightmost agent of a cluster on the left of $x_{i,l}$ and by Cluster-Separation Property, d(x_i, y) $\geq$ d($x_{i,l}$, $x_{j-1,r}$) > D(C_j). Hence, such a deviation cannot be profitable for x_i (note how this case crucially uses the facility allocation to the second rightmost agent of a cluster). If y > x_i, x_i can only gain if y is the second rightmost agent of a cluster including C_j) ∪ y, $x_{j,l+1}$ and possibly some agents on the left of C_j), which is treated below.
- **Case 2:** The agents in C_j are clustered together in $\vec{Y}$ and C_j is merged with some agents from C_{j+1} and possibly some other agents to the left of $x_{i,l}$ (note that merging C_j only with agents to the left of $x_{i,l}$ does not change the facility of x_i). Then, we only need to consider the case where the deviating agent x_i is $x_{i,r}$, since any other agent to the left of $x_{j-1,r}$ cannot gain, because

cluster merging can only move their serving facility further to the right. As for $x_{j,r}$, we note that by optimality and the hypothesis that agents in C_j belong in the same cluster of $\vec{Y}$, $x_{i,r}$ cannot cause the clusters C_j and C_{j+1} to merge in $\vec{Y}$ by deviating in the range $[x_{j,r}, x_{j+1,l}]$. The reason is that the set of agents $(C_j \setminus x_{j,r}) \cup \{y\} \cup C_{j+1}$ cannot be served optimally by a single facility, when the set of agents $C_j \cup C_{j+1}$ requires two facilities in the optimal clustering $\vec{C}$. Hence, unless C_{j+1} is split in $\vec{Y}$ (which is treated similarly to Case 3a), $x_{j,r}$ can only move her facility to C_{j+1}, which is not profitable for her, due to Cluster-Separation Property.

- **Case 3:** C_j is split into two clusters in $\vec{Y}$. Hence, the leftmost agents, originally in C_j, are served by a different facility than the rest of the agents originally in C_j. We next show that in any profitable deviation of x_i where C_j is split, either the deviation is not feasible or the cluster separation property is violated.
  - **Case 3a:** By hypothesis, there are agents z,w $\in C_j$, that in $\vec{Y}$ are clustered with agents p $\in C_l, l \neq j$. Suppose that in $\vec{Y}$, agent z is clustered in C'_j and agents z,p are clustered in C'_{j+1}, which is in agreement with our initial hypothesis. We will try to produce a contradiction, using the known properties of intra-cluster and inter-cluster distances. Since our mechanism produces a clustering $\vec{Y}$, that clustering must have the Cluster Separation Property. From 4.19 in $\vec{Y}$ we know that :

$$d(C'_j, C'_{j+1}) > D(C'_{j+1}) \implies$$
$$d(w, z) > d(w, p)$$

    From 4.19 in $\vec{C}$ we know that :

$$d(C_j, C_l) > D(C_j) \implies$$
$$d(w, p) > d(w, z)$$

    Hence, we have our contradiction.
  - **Case 3b:** Agents in C_j are split and are not clustered together with any agents of $\vec{x} \setminus C_j$ in $\vec{Y}$. Hence, y is not clustered with any agents in $\vec{x} \setminus C_j$ in $\vec{Y}$. Otherwise, i.e., if y is not clustered with agents of C_j in $\vec{Y}$, it would be suboptimal for clustering $\vec{Y}$ to allocate more than one facility to agents of $C_j \setminus x_i$ and at most k − 2 facilities to ($\vec{x} \cup$ y) $\setminus C_j$, while the optimal clustering $\vec{C}$ allocates a single facility to C_j and k − 1 facilities to $\vec{x} \setminus C_j$. But again if y is only clustered with agents of C_j, it is suboptimal for clustering $\vec{Y}$ to allocate more than one facility to agents of $(C_j \square \{y\}) \setminus x_i$ and at most k − 2 facilities to $\vec{x} \setminus C_j$, while the optimal clustering $\vec{C}$ allocates a single facility to C_j and k − 1 facilities to $\vec{x} \setminus C_j$.

4.5 Conclusion

Our beyond worst-case analysis approach circumvents a great number of issues that arise with the result of 3.28 and allows us to study problems closer to the real world. However, there are still some results that restrict this approach. 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]. Moreover, the impossibility theorem holds for $(\sqrt{2} - \epsilon)$-stable instances, [23]. Also, the OPTIMAL mechanism is applied to instances with no singleton in their optimal clustering and the ALMOSTRIGHTMOST mechanism does not produce an optimal approximation ratio. A natural extension to these issues would be to find a mechanism that can deal with instances with lower stability than 5, while also dealing with singletons in their optimal clustering.

Chapter 5

Learning-Augmented Mechanisms on Facility Location Games

The notion of perturbation stability is not the only approach to mechanism design, which we can use to avoid the method of worst-case analysis. Although the use of worst-case analysis provides a certain *robustness* to the outcome of our algorithm, it deprives us of the flexibility of studying instances that are able to produce a closer-to-the-"real world" model for our problems. These kinds of problems have certain properties, that machine learning algorithms can exploit to produce useful "predictions". Motivated by this tension between worst-case analysis and machine-learning algorithms, a surge of recent work is aiming for the best of both worlds by designing robust algorithms that are guided by machine-learned predictions. The goal of this exciting new literature on "algorithms with predictions" is to combine the *robustness* of worst-case guarantees with *consistency* guarantees, which prove stronger bounds on the performance of an algorithm whenever the prediction that it is provided with is accurate. Although, the introduction of these *learning-augmented algorithms* has taken place mostly in the field of online algorithms (as we can observe in this survey [38]), there is an application in the field of Mechanism design, presented here [1]. These kinds of Mechanisms can be called from now on *Learning-Augmented Mechanisms*. There is a long list of classic algorithmic problems that have been studied in that framework, including online paging [34], scheduling [44], and secretary problems [17], [5], optimization problems with covering [11] and knapsack constraints [28], as well as Nash social welfare maximization [12] and several graph [7] problems. We note that this line of work has also studied facility location problems [22], [29]. However, the crucial difference is that these papers are restricted to non-strategic settings, and the predictions are used to overcome information limitations regarding the future, rather than limitations regarding privately held information. [41] use bid predictions in auctions to learn reserve prices and yield revenue guarantees as a function of the prediction error but provide no bounded robustness guarantees. We can implement this thought process to the Facility Location Games, hoping that the addition of predictions to our established algorithms can improve their time complexity or even introduce new algorithms. We have already introduced the notions of *Maximum Cost*, *Social Cost*, *Approximation Ratio*, and *Strategyproofness*. We will introduce two new notations that measure our dependence on the predictions, while also parameterizing the worst-case performance guarantee of our mechanism. These notations are *consistency* and *robustness*.

5.1 Basic Setting, Definitions and Preliminaries

In the setting of Facility Locations Games, our mechanism is provided with a prediction $\hat{o}$, along with the instance $\vec{x}$. The prediction is, essentially, a set k of facility locations, that a machine learning algorithm proposes, as the optimal solution to our problem. Depending on our goal - *Maximum Cost* or *Social Cost* - there is an optimal placement for the k facility locations, $o(\vec{x}) = o$

Now we will re-introduce the same setting for facility location as in chapter 2. We deal with the single facility location problem in the two-dimensional Euclidean space. Our goal is to choose a location $f \in R^2$ for a facility, aiming to serve a group of n agents. Each agent I have a preferred location $p_i \in R^2$ and is assigned to a price that he must pay, denoted $cost(f, p_i) = d(f, p_i)$, where $d(f, p_i)$ corresponds to the Euclidean distance between his preferred location p_i and the chosen location f. We refer to the set of preferred locations $P = (p_1, ..., p_n)$ for the agents, as the problem's *instance*. The mechanism receives as input the instance P and the prediction of the optimal facility location $\hat{o}$. We are trying to minimize two standard cost functions. The Social Cost function $SC(f, P) = \sum_{i=1}^{n} cost(f, p_i)$ (i.e the aggregated cost of all agents) and the Maximum Cost Function $MC(f, P) = max_{i \in N}[cost(f, p_i)]$ (i.e. the maximum cost over all agents. Depending on the cost function that we are trying to minimize, there exists an optimal facility location $o(P) = (x_o(P), y_o(P)$. In the strategic version of the facility location problem, the preferred location p_i of each agent i is private information. Similarly, to a mechanism that deals with the normal facility location setting a *learning-augmented Mechanism M* maps the set P to a location $f = M(P, \hat{o})$. However, the goal of each agent is to minimize their own cost, so they can choose to misreport their preferred location if that can reduce their cost.

The ideal mechanism would produce the optimal solution when he received a prediction with η = 0 (consistency = 1) and the worst-case approximation ratio when η is arbitrarily large (robustness = approximation ratio of worst-case scenario). This is, essentially, a mechanism that utilizes the prediction only when it is close to the optimal solution, disregarding any other case. This is an unreachable goal in most cases, since if we have consistency = 1, our mechanism is more than likely to trust the prediction in most cases, which leads to unbounded robustness. Our goal is to find the best possible trade-off between robustness and consistency.

In the learning-augmented mechanism design framework, before requesting the set of preferred locations P from the agents, the designer is provided with a prediction $\hat{o}$ regarding the optimal facility location o(P). The designer can use this information to choose the rules of the mechanism but, as in the standard strategic facility location problem, the mechanism denoted $M(P, \hat{o})$, needs to be strategyproof. In essence, if there are multiple strategyproof mechanisms the designer can choose from, the prediction can guide their choice, aiming to achieve improved guarantees if the prediction is accurate (consistency), but retaining some worst-case guarantees (robustness). Consistency and robustness are the standard measures in algorithms with predictions.

Definition 5.1 ($\alpha - consistency$). *Given some social cost function C (i.e MC(·),SC(·)), a mechanism is α − consistent if it achieves an α-approximation ratio when the prediction is correct ($\hat{o} = o(P)$), i.e.*

$$max_P[\frac{C(M(P, o(P)), P))}{C(o(P), P)}] \leq \alpha$$

Definition 5.2 ($\beta - robustness$). *Our mechanism is β − robust if it achieves a β-approximation ratio even when the prediction is arbitrarily wrong, i.e.*

$$max_{P,\hat{o}}[\frac{C(M(P, \hat{o}), P)}{C(o(P), P)}] \leq \beta$$

Note that any known strategyproof mechanism that guarantees a γ -approximation without predictions, directly implies bounds on the achievable robustness and consistency. The designer could just disregard the prediction and use this mechanism to achieve γ-robustness. However, this mechanism would also be no better than γ-consistent, since it ignores the prediction. The main challenge is to achieve improved consistency guarantees, without sacrificing too much in terms of robustness. For an even more refined understanding of the performance of a learning-augmented mechanism, one can also prove worst-case approximation ratios as a function of the prediction error $\eta > 0$. In facility location, we let the error $\eta(\hat{o}, P) = cost(\hat{o}, o(P)\ C(o(P), P)$ be the distance between the predicted optimal location $\hat{o}$ and the true optimal location o(P), normalized by the optimal cost. Given a bound η on the prediction error, a mechanism achieves $\gamma(\eta)$-approximation if

$$max_{P,\hat{o}:\eta(\hat{o},P)\leq\eta}[\frac{C(M(P,\hat{o}),P)}{C(o(P),P)}] \leq \gamma(\eta)$$

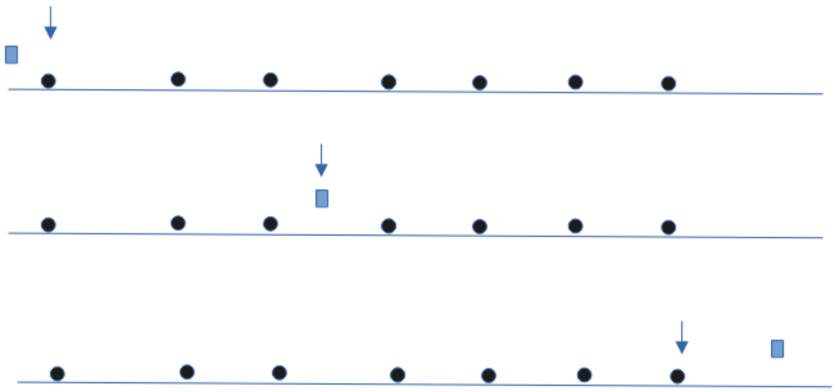
A strategyproof mechanism that plays a central role in the strategic facility location problem is the *Coordinatewise Median (CM) mechanism*. This mechanism takes as input the locations $P = \{(x_i, y_i)\}_{i\in[n]}$ of the n agents and determines the facility location by considering each of the two coordinates separately. The x-coordinate of the facility is chosen to be the median of $\{(x_i\}_{i\in[n]}$, i.e., the median of the x-coordinates of the agents' locations, and its y-coordinate is the median of $\{(y_i\}_{i\in[n]}$ (if n is even, we assume the smaller of the two medians is returned). The more general class of *Generalized Coordinatewise Median (GCM) mechanisms* take as input the locations P of the n agents, as well as a multiset P' of points that are constant and independent of the locations reported by the agents, and outputs $CM(P \cup P')$. In other words, a GCM mechanism is the coordinatewise median mechanism over the locations of the agents and the additional constant points P' chosen by the designer (often called phantom points). Apart from being deterministic and strategyproof, any GCM mechanism is also anonymous: its outcome does not depend on the identity of the agents, i.e., it is invariant under a permutation of the agents.

5.2 Minimizing the Maximum Cost Objective

Our focus will be on the maximum cost objective. It is known that a deterministic and strategyproof mechanism can not achieve anything better than a 2-approximation, even for the one-dimensional case [Procaccia and Tennenholtz, 2013] [43]. Our goal is to provide a deterministic, strategyproof, and anonymous mechanism that is 1-consistent, while also achieving the best possible trade-off between robustness and consistency, which is what most learning-augmented mechanisms aim to accomplish.

5.2.1 Facility location on the line

Initially, we will deal with the 1-dimensional case, with the introduction of the MinMaxP mechanism. This mechanism uses the prediction $\hat{o}$ as the default facility location choice unless the prediction lies "on the left" of all the points in P or "on the right" of all the points in P. In the former case, the facility is placed at the leftmost point in P instead, and in the latter, it is placed at the rightmost point in P.



Agents are represented by black circles, prediction is represented by blue rectangular, facility location is represented by an arrow. These are the 3 different ways we can assign a facility to the instance. The first one shows what happens when $\hat{o} < min_i p_i$, the second one when $\hat{o} \in [min_i p_i, max_i p_i]$ and the third one when $\hat{o} > max_i p_i$

Mechanism 11 MinMaxP mechanism for maximum cost in one dimension.

Input: points $(p_1, ..., p_n) \in \mathbb{R}^n$, prediction $\hat{o} \in \mathbb{R}$
if $\hat{o} \in [min_i p_i, max_i p_i]$ **then**
  return $\hat{o}$
else if $\hat{o} < min_i p_i$ **then**
  return $min_i p_i$
else
  return $max_i p_i$
end if

We will now prove some of the properties of *MinMaxP Mechanism*. The mechanism is strategyproof, 1-consistent, which means that we choose the optimal facility location when we are provided with the correct prediction, while also being 2-robust, when the prediction is arbitrarily wrong. MinMaxP is able to achieve the best of both worlds: when the prediction is correct, it yields an optimal outcome, and when the prediction is incorrect, the approximation factor never exceeds 2, which is the best-possible worst-case approximation.

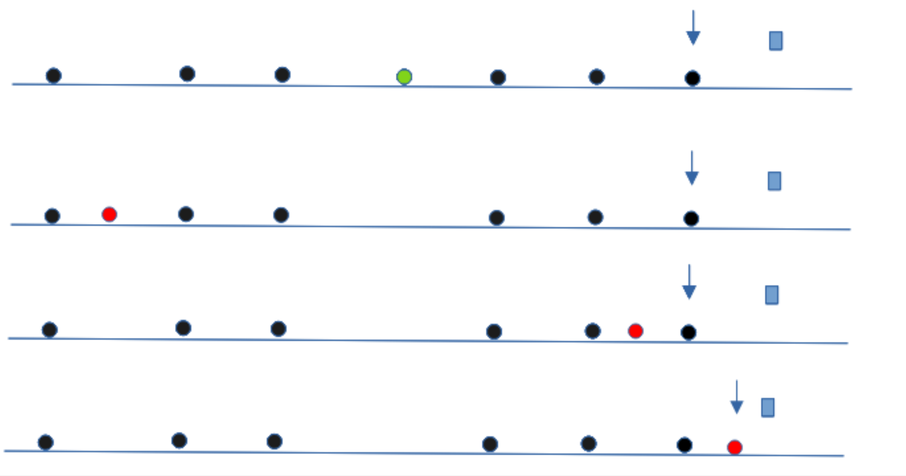
Theorem 5.3. *The MinMaxP mechanism is deterministic, strategyproof, and anonymous. It is 1-consistent and 2-robust for the maximum cost objective.*

Proof. First, we will tackle the issue of the algorithm's strategyproofness. Consider an agent *i* and without loss of generality, assume that $p_i \leq \hat{o}$, i.e. the agent's true location is weakly on the left of the prediction's location. An observation regarding the facility allocation rule is that the mechanism allocates a facility to the point of the instance which is the closest to our prediction. So, if the prediction is out of the instance's bounds, we project the prediction on the instance. We will consider two cases, depending on the position of agent *i* in relation to the position of the other agents.

Suppose that agent *i* is the rightmost agent, meaning that $\forall j \neq i \; p_j \leq p_i$. In this case, agent *i* does not have an incentive to lie, since a facility will be allocated to him if he just reveals his true location.

The prediction will either be on him ($\hat{o} = p_i$) meaning that the mechanism will return $\hat{o}(= p_i)$, or the prediction is strictly to the right of the agent($\hat{o} > p_i$), meaning that the mechanism will return $max_j p_j$, which is the location of the rightmost agent.

Now, let's suppose that agent *i* is not the rightmost agent, which means that $\exists j$ s.t. $p_j > p_i$. We declare as *f*, the facility location that the mechanism returns. We can conclude that $f > p_i$, since if $f = p_i$, agent *i* does not have an incentive to lie. We also declare as p_i', the false location that agent *i* reports, with which he hopes to manipulate the location of the facility. It is clear that the agent's lie must change the bounds of the instance, otherwise, facility f will stay in the same location, so $p_i' \notin [min_i p_i, max_i p_i]$. If $p_i' < p_i$, does not affect f, since f is either a projection of $\hat{o}$ to the rightmost agent of the instance or is already a point inside the instance. If $p_i' > p_i$, the only change that can happen is p_i' becomes the location of the new rightmost agent, essentially pushing facility location f, further away from agent *i*. We can now conclude that MinMaxP is strategyproof.



The true location of the agent is the green circle and the different deviations he can try are the red circles. It is evident from the above schema, that no deviation is profitable

Another approach to prove the strategyproofness of the mechanism is to view MinMaxP as a *Generalized Coordinatewise Median (GCM) mechanism*. As mentioned above, a *GCM* receives as input the locations P of the n agents, as well as a multiset P' of points that are constant and independent of the locations reported by the agents and outputs $CM(P \cup P')$. We can observe that if P' contains n-1 copies of the prediction $\hat{o}$, then $GCM(P, P')$ produces the same results, as MinMaxP.

- If $\hat{o} \in [min_i p_i, max_i p_i]$, then the median of $P \cup P'$ is $\hat{o}$
- If $\hat{o} < min_i p_i$, then the median of $P \cup P'$ is $min_i p_i$
- If $\hat{o} > max_i p_i$, then the median of $P \cup P'$ is $max_i p_i$

Since $GCM(P, P')$ is strategyproof, the same applies to MinMaxP.

To verify the consistency guarantee, to verify the consistency guarantee, consider any instance where the prediction $\hat{o}$ is accurate. Since the truly optimal location for the maximum cost objective is the middle of the leftmost and rightmost point in P, then we know that whenever $\hat{o}$ is accurate, it must be that $\hat{o} \in [min_i p_i, max_i p_i]$. As a result, for any such instance, the mechanism will place the facility at the optimal location, $\hat{o}$, leading to a consistency of 1.

Finally, to compute the robustness guarantee, we only need to observe that the facility f is a projection of $\hat{o}$ on the instance P, meaning that $f \in [min_i p_i, max_i p_i]$. The worst case scenario is to allocate a facility on the leftmost or rightmost agent, making the maximum cost of the mechanism equal to $max_i p_i] - min_i p_i$. The optimal maximum cost is $(max_i p_i] - min_i p_i)/2$, since the optimal facility location is in the middle of $[min_i p_i, max_i p_i]$. This leads us to our 2-robustness guarantee. □

5.2.2 Facility location on 2 dimensions

We then move on to the two-dimensional version of the problem, for which prior work has produced an optimal deterministic strategyproof mechanism achieving a 2 approximation n [Alon et al., 2010 - [3], Goel and Hann-Caruthers, 2021 - [26]]. For the 2D Case, i.e. $p_i \in \mathbb{R}$, we introduce the Minimum Bounding Box mechanism. We extend the MinMaxP mechanism to this setting by running it separately for each of the two dimensions. An alternative, more geometric, description of this mechanism is that it first computes the minimum axis-parallel bounding box of the set P of agent locations and then places the facility at the location within that box that is closest to the predicted optimal location. We therefore call it the Minimum Bounding Box mechanism.

Mechanism 12 Minimum Bounding Box mechanism for maximum cost in two dimensions.

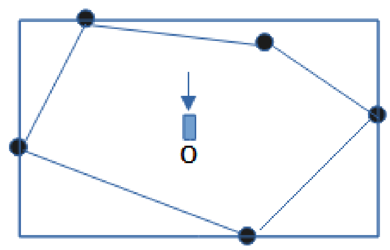
Input: points $((x_1, y_1), ..., (x_n, y_n)) \in \mathbb{R}^{2n}$, prediction $(x_{\hat{o}}, (y_{\hat{o}}) \in \mathbb{R}^2$
$x_f = MinMaxP((x_1, ..., x_n), x_{\hat{o}})$
$y_f = MinMaxP((y1, ..., y_n), y_{\hat{o}})$
return (x_f, y_f)

We will now prove some of the properties of *Minimum Bounding Box Mechanism*. The mechanism is strategyproof, 1-consistent, which means that we choose the optimal facility location when we are provided with the correct prediction, while also achieving a (1+$\sqrt{2}$)-robustness when the prediction is arbitrarily wrong. Although the best achievable approximation is 2 , we will prove that we can not create a mechanism that provides (1+$\sqrt{2} - \epsilon$)-robustness, while also maintaining 1-consistency.

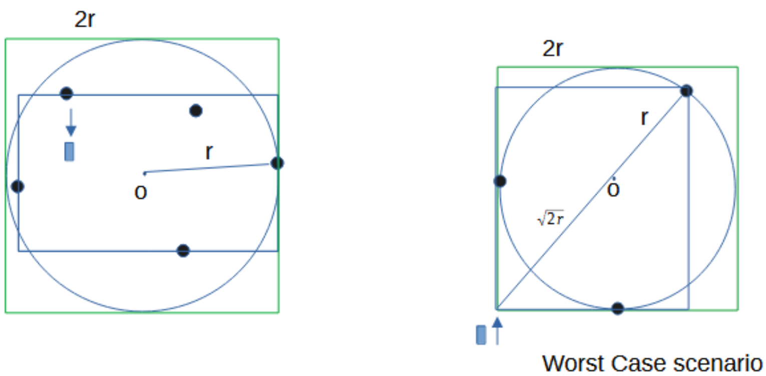
Theorem 5.4. *The Minimum Bounding Box mechanism is deterministic, strategyproof and anonymous. It is 1-consistent and (1 +$\sqrt{2}$)-robust for the maximum cost objective.*

Proof. Initially, we will tackle the issue of the algorithm's strategyproofness. We can observe that *Minimum Bounding Box Mechanism* runs *MinMaxP* separately for each dimension, which implies the strategyproofness of our Mechanism since we have already proved the strategyproofness of *MinMaxP*. Alternatively, we can view *Minimum Bounding Box Mechanism* as *GCM* mechanism, the same way we did with *MinMaxP*, by constructing a set P' that contains n-1 copies of $\hat{o}$.

We will now prove the consistency guarantee. Let's denote as o, the optimal location of the facility, and since our prediction is correct, $\hat{o} = o$. Moreover o is always in the convex hull of the points in P. Obviously, the convex hull is contained within the minimum axis-parallel bounding box, so for any instance where the prediction $\hat{o}$ is correct, this prediction is in the bounding box, and is thus the location returned by the mechanism, verifying its 1-consistency.



Visual Representation of Mechanism's Consistency



Visual Representation of Mechanism's Robustness

Finally, we will prove the robustness guarantee with a geometrical approach. Consider any instance with set P of agents' locations, optimal facility location o, and MC(o,P) as the maximum cost an agent of P can pay if we set the facility location on o. Consider, also, a circle C with center on o and radius

= MC(o,P). Obviously, C contains all points of P , since MC(o,P) is the distance between o and the furthest point of P from o. Furthermore, C is inscribed to a square T with a side length of 2MC(o,P), which also, contains all points of P. As a result, the minimum bounding box is contained inside T, which means that the facility location f that our mechanism produces is also inside T. The most distant point of T from o, is one of its vertices, meaning that $cost(o, f) \leq \sqrt{2}MC(o, P)$. From triangle inequality :

$$max_{p_i \in P}\{cost(f, p_i)\} \leq cost(o, f) + max_{p_i \in P}\{cost(o, p_i)\} \leq (1 + \sqrt{2})MC(o, P)$$

□

5.2.3 Optimality of Minimum Bounding Box

With Minimum Bounding Box, we have managed to achieve 1-consistency, but that comes at the cost of the robustness guarantee, which weakens from 2 to $1 + \sqrt{2}$. The coordinatewise median CM mechanism achieves a 2-approximation for the maximum cost over all instances in two dimensions [Goel and Hann-Caruthers, 2021], meaning that it is 2-consistent and 2-robust. A natural question, that comes up is whether the Minimum Bounding Box mechanism is tight and the trade-off between 1-consistency and $(1 + \sqrt{2})$-robustness is the best we can attain. It turns out that ensuring that there is no middle ground between the coordinatewise median CM mechanism and Minimum Bounding Box mechanism.

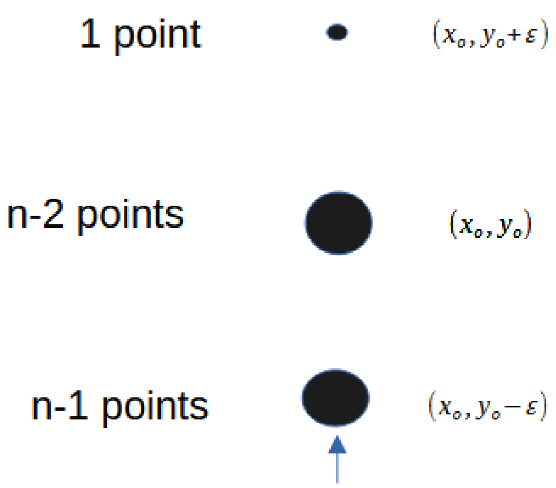
Theorem 5.5. *There is no deterministic, strategyproof, anonymous Mechanism that guarantees (2-ϵ) consistency and $(1 + \sqrt{2} - \epsilon)$-robustness, with respect to the maximum cost objective for any ϵ > 0.*

Proof. Our proof sketch will follow 3 steps.

- Prove that every mechanism with the desired properties (determistic, strategyproof, anonymous, bounded robustness) can be modeled to a GCM mechanism with n-1 constant points in P'.
- Create an instance in which these mechanisms need to have all the n-1 constant points on $\hat{o}$, in order to achieve (2-ϵ) consistency
- Create an instance in which a GCM mechanism with n-1 constant points on $\hat{o}$, have at least $(1 + \sqrt{2})$-robustness

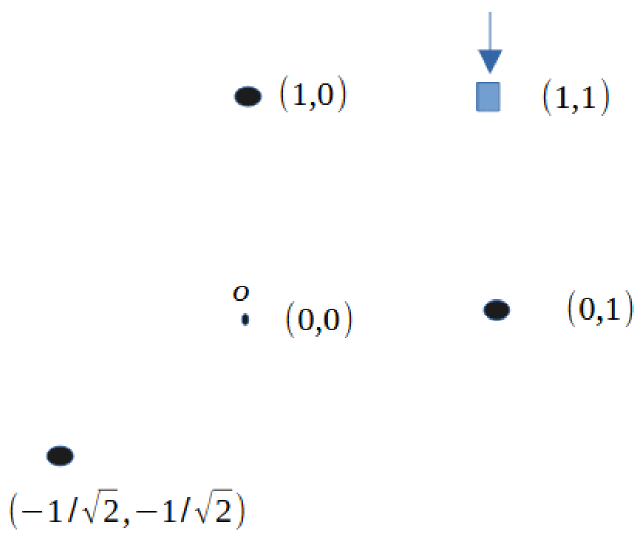
To fulfill our first objective we just need to observe that any mechanism f with bounded robustness needs to be unanimous, i.e., given a set of points P where all the points are at the same location $(p_i = p_j \forall i, j \in [n])$, the mechanism needs to place the facility at that same location, i.e., $f(P) = p_i$. If not, then its cost would be positive, while the optimal cost is zero, by placing the facility at the same location as all the points. Therefore, we can restrict our attention to mechanisms that are unanimous. Using the characterization of Peters et al. [1993], we know that any deterministic, strategyproof, anonymous, and unanimous mechanism in our setting takes the form of a generalized coordinatewise median (GCM) mechanism with n-1 constant points in P'.

For the second objective, consider any GCM mechanism with at least one of its n-1 constant points not located on $\hat{o} = (x_o, y_o)$. Without loss of generality, assume that this point is directly below $\hat{o}$ at



$(x_o, y_o - \epsilon)$. We create the following instance : n-1 agents at $(x_o, y_o - \epsilon)$ and 1 agent at $(x_o, y_o + \epsilon)$. So we have 1 point at $(x_o, y_o + \epsilon)$, n-2 points at (x_o, y_o) and n points at $(x_o, y_o - \epsilon)$. Their median is at $(x_o, y_o - \epsilon)$, however, the optimal facility location for the maximum cost objective is at (x_o, y_o). Our mechanism has at least 2-consistency. So if we wish to have lower consistency we need to place n-1 constant points at $\hat{o}$

For our final objective, we conclude the proof by showing that the robustness of the GCM mechanism that uses the prediction point $\hat{o}$ for all the $\mathrm{n}-1$ constant points in P' is no better than $(1+\sqrt{2})$. Assume that the prediction is located at (1,1) and consider and instance with n = 3 agents located at (0, 1), (1, 0), and $(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})$. The set P' will contain 2 points located at (1,1). The GCM mechanism will place the facility at (1,1), however, the optimal facility location is at (0,0). The maximum cost of GCM is $1 + \sqrt{2}$, while the optimal is 1, meaning that no GCM mechanism that uses the prediction point $\hat{o}$ for all the $\mathrm{n} - 1$ constant points in P' can achieve better robustness than $(1 + \sqrt{2})$.



□

5.3 Conclusion

We can conclude this section by mentioning that in [1] Gkatzelis et al introduced a learning-augmented mechanism for the social cost objective on the two-dimensions plane using a deterministic, anonymous, and strategyproof mechanism, that achieves a $\sqrt{2}$-approximation, which is optimal for this class of mechanisms. They provide a family of mechanisms, parameterized by a "confidence value" c $\in$ [0, 1) that the designer can choose depending on how much they trust the prediction. If the designer is confident that the prediction is of high quality, then they can choose a higher value of c, which provides stronger consistency guarantees, at the cost of deteriorating robustness guarantees. Specifically, we prove that our deterministic and anonymous mechanism is $\sqrt{2c^2 + 2}$/(1 + c)-consistent and $\sqrt{2c^2 + 2}$/(1 − c)-robust. However, it still remains to be seen how we can utilize the addition of predictions in our facility locations mechanisms in the general case of K facilities.

Chapter 6

Learning-Augmented Mechanism Design with Predictions for Stable Instances of Facility Location Games

We can now integrate all the previous ingredients that were introduced in previous chapters (facility location games, γ-stable instances, and learning-augmented mechanism design) and try to come up with an elegant mechanism that incorporates them. Our goal is to locate k facilities on the line. We will create mechanism $M(\vec{x}, \hat{o})$, which receives the tuple $(\vec{x}, \hat{o})$ as input. Our instance $\vec{x}$ is a vector consisting of each agent's location on the line and $\hat{o}$ is a vector with the predicted locations of the facilities, produced by an external system. Our work may seem easier since we already have mechanisms that are applied successfully in γ-stable instances and with the addition of external predictions, it seems right to hope for an even better outcome. However, a problem arises from the fact that the agents can now exploit the mechanism through the predictions' location, adding another layer of complexity.

Let's examine the different kinds of deviations that an agent can produce in the simple case of k-Facility Location Game in a γ-stable instance. The agent can merge clusters, the agent can split clusters (if he creates a new singleton cluster or if he merges other clusters), the agent can create singleton clusters and he can finally change its own cluster's length. In the OPTIMAL mechanism, we do not deal with singleton clusters, while the change of the agent's cluster length is not an optimal strategy. So, we are left to deal with merges and splits, where, as seen in the strategyproofness analysis, we can make a distinction between two cases: one in which the Cluster Separation Property is violated and one where the optimal clustering does not change, despite the deviation. In ALMOSTRIGHTMOST, we allow the existence of singleton clusters. The analysis of merges and splits is the same as in OPTIMAL, since both of these strategies are infeasible or suboptimal, due to the structure of a γ-stable instance, with $\gamma \geq 2 + \sqrt{3}$. The inclusion of singleton clusters can be dealt with with the increase of stability on the instance. If $\gamma \geq 5$, then from 4.22 no agent can gain from being served by a singleton cluster, that he creates.

With the addition of predictions, our mechanism is obliged to utilize them in its decision-making process, making it susceptible to manipulation. One natural approach is to generalize the rule of MINMAXP, that [1] introduced, where each prediction is attached to a cluster and according to its position in relation to the cluster's bounds, we allocate the facility. However, in MINMAXP, we had to deal with only one cluster and only one prediction, while in the general case, we had to assign k predictions to k clusters in a strategyproof way. This is the new kind of deviation that an agent can use to profit. He can manipulate the structure of the instance and choose a more suitable prediction to be assigned to his cluster. We can make the following categorization of our deviations:

1. Case 1 - Splits \ Merges that maintain the initial assignment of predictions to clusters

2. Case 2 - Splits \ Merges that do not maintain the assignment of predictions to clusters.

3. Case 3 - Agent deviation that changes only its own cluster's length.

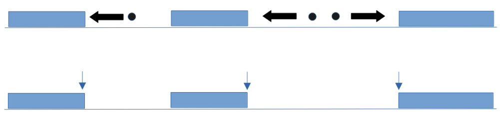
Case 2 is already examined in the ALMOSTRIGHTMOST \ OPTIMAL analysis. Our goal is to investigate ways to deal with Cases 1 and 3.

6.1 Creating our Mechanism

The Mechanism receives the input, runs a check on the cluster-separation property, and checks if a singleton cluster exists. If the instance passes both of these tests, we output k facilities. For our mechanism to work, we need our instance to have at least 5 stability and its optimal clustering to not include singleton clusters. If the Cluster Separation Property is not violated and there is no singleton cluster in our optimal clustering, we can match the i-th prediction $\hat{o}_i$ to the i-th cluster C_i. Since our instance is γ-stable, we can treat each cluster as a single instance, which is completely divided by the rest of the clusters, and apply the MinMaxP Mechanism [1] on each cluster. Unfortunately, we need to include the restrictions of no singleton cluster in the optimal clustering, since if we allow an agent to deviate and create a singleton cluster, without disturbing the stability of the instance, then he can possibly isolate a distant prediction and change the enumeration of the predictions to his gain.

Mechanism 13 Mechanism $M(\vec{x}, \hat{o})$:Deterministic Mechanism for 5-stable instances with no singleton clusters

Result: An allocation of k-facilities
Input: A k-Facility Location instance $\vec{x}$ and k-vector of predictions on facilities locations $\hat{o}$ Find the optimal clustering $\vec{C} = (C_1, ..., C_k)$ of $\vec{x}$.
for i $\in$ 1, . . . ,k **do**
  Match $\hat{o}_i$ to i-th cluster, C_i.
  if $\hat{o}_i \in [x_{i,l}, x_{i,r}]$ **then**
    Allocate a facility to $\hat{o}_i$.
  end if
  if $\hat{o}_i < x_{i,l}$ **then**
    Allocate a facility to $x_{i,l}$
  end if
  if $\hat{o}_i > x_{i,r}$ **then**
    Allocate a facility to $x_{i,r}$
  end if
end for
Output: The k-facility allocation that was previously defined.



Blue rectangles are clusters, black cycles are predictions, and blue arrows are allocated facilities.

6.1.1 Consistency for Maximum Cost

We know that $MC^*(\vec{x}) = \frac{\max_i |C_i|}{2}$. If we want to check the consistency of the mechanism, then we assume that the predictions are correct $\hat{o} = o$ = center of each cluster.

$$MC(\vec{x}, M(\vec{x}, \hat{o})) = \frac{\max_i |C_i|}{2} \Rightarrow$$
$$MC(\vec{x}, M(\vec{x}, \hat{o})) = MC^*(\vec{x})$$

We conclude that M is 1-consistent for the Maximum Cost objective.

6.1.2 Robustness for Maximum Cost

If we want to check the robustness of the mechanism, then we assume that the predictions are arbitrarily wrong. In our case it is one of the two edges so $o_i = x_{i,r}$

$$MC(\vec{x}, M(\vec{x}, \hat{o})) = \max_i |C_i| \Rightarrow$$
$$MC(\vec{x}, M(\vec{x}, \hat{o})) = 2 * MC^*(\vec{x})$$

We conclude that M is 2-robust for the Maximum Cost objective.

6.1.3 Consistency for Social Cost

If we want to check the consistency of the mechanism, then we assume that the predictions are correct $\hat{o} = o^*$ = median of each cluster. So $\hat{o} = (med(C_1), ..., med(C_k)) = o^*$

$$SC(\vec{x}, M(\vec{x}, \hat{o})) = \sum_{i=1}^{n} d(x_i, o^*) \Rightarrow$$
$$SC(\vec{x}, M(\vec{x}, \hat{o})) = SC^*(\vec{x})$$

We conclude that M is 1-consistent for the Social Cost objective.

6.1.4 Robustness for Social Cost

If we want to check the robustness of the mechanism, then we assume that the predictions are arbitrarily wrong. In our case, it is one of the two edges so $o_i = x_{i,r}$. Suppose that we have only one cluster C, to make our computations simpler.

$$
\begin{aligned}
SC(\vec{x}, M(\vec{x}, \hat{o})) &= \sum_{i=1}^{k} \sum_{j \in C_i} d(x_j, o_i) \Rightarrow \\
SC(\vec{x}, M(\vec{x}, \hat{o})) &= \sum_{i=1}^{k} \sum_{j \in C_i} d(x_j, x_{i,r}) \Rightarrow \\
SC(\vec{x}, M(\vec{x}, \hat{o})) &\leq \sum_{i=1}^{k} \sum_{j \in C_i} D(C_i) \Rightarrow \\
SC(\vec{x}, M(\vec{x}, \hat{o})) &\leq (n-1) * \sum_{i=1}^{k} D(C_i) \Rightarrow \\
SC(\vec{x}, M(\vec{x}, \hat{o})) &\leq (n-1) * SC^*(\vec{x})
\end{aligned}
$$

We conclude that M is (n-1)-robust for the Social Cost objective.

6.1.5 Strategyproofness

Suppose that $x_i \in C_j$ and agent i declares false position y. With that lie, he creates clustering $\vec{Y}$. We can make 3 categories for the agents' deviations.

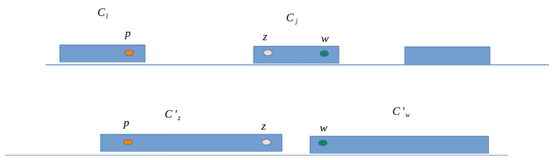
Splits. C_j is split into two clusters in $\vec{Y}$. Hence, the leftmost agents, originally in C_j , are served by a different facility than the rest of the agents originally in C_j We next show that in any profitable deviation of x_i where C_j is split, either the deviation is not feasible or the cluster-separation property is violated.

1. Agents in C_j are clustered together with some agents of $x \backslash Cj$ in $\vec{Y}$. By hypothesis, there are agents $z, w \in C_j$ placed in different clusters of $\vec{Y}$, and at least one of them, say z, is clustered together with an agent $p \in C_l$, with $l \neq j$, in $\vec{Y}$. For brevity, we refer to the (different) clusters in which z and w are placed in clustering $\vec{Y}$ as C'_z and C'_w, respectively. Then, $D(C'_z) \geq d(p, z) > D(C_j)$, by Lemma 1. But also $d(C'_z, C'_w) < d(z, w) \leq D(C_j)$, consequently $D(C'_z) > d(C'_z, C'_w)$, which implies that the cluster-separation property is violated and Mechanism does not allocate any facilities in this case.

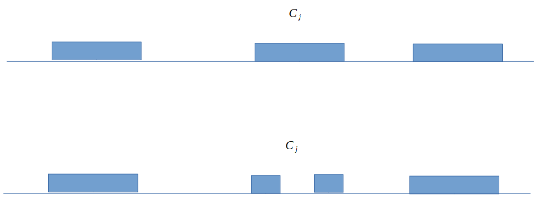
2. Agents in C_j are split and are not clustered together with any agents of $x \backslash C_j$ in $\vec{Y}$. Hence, y is not clustered with any agents in $x \backslash C_j$ in $\vec{Y}$. Otherwise, i.e., if y is not clustered with agents of C_j in $\vec{Y}$, it would be suboptimal for clustering C_j to allocate more than one facility to agents

of $C_j \backslash x_i$ and at most k-2 facilities to $(\vec{x} \cup y) \backslash C_j$, while the optimal clustering $\vec{C}$ allocates a single facility to C_j and k-1 facilities to $\vec{x} \backslash C_j$. But again if y is only clustered with agents of C_j, it is suboptimal for clustering $\vec{Y}$ to allocate more than one facility to agents of $(C_j \cup y) \backslash x_i$ and at most k-2 facilities to $\vec{x} \backslash C_j$, while the optimal clustering $\vec{C}$ allocates a single facility to C_j and k - 1 facilities to $\vec{x} \backslash C_j$.

The above restrictions for the **Splits** deviations are independent of the mechanism, that we choose to use. They are produced by the structure of the instance. So they exist, regardless of the way, we will place the facilities.



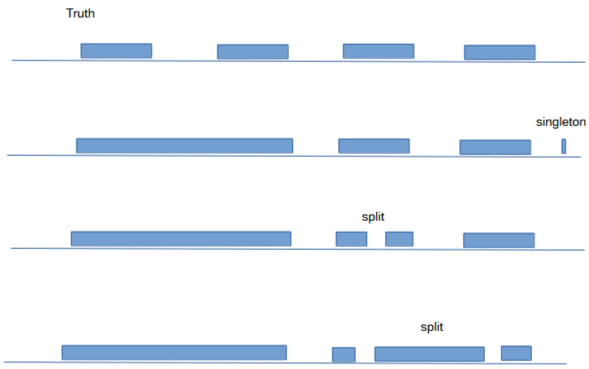
In the split case 1, we can see that the CSP is violated.



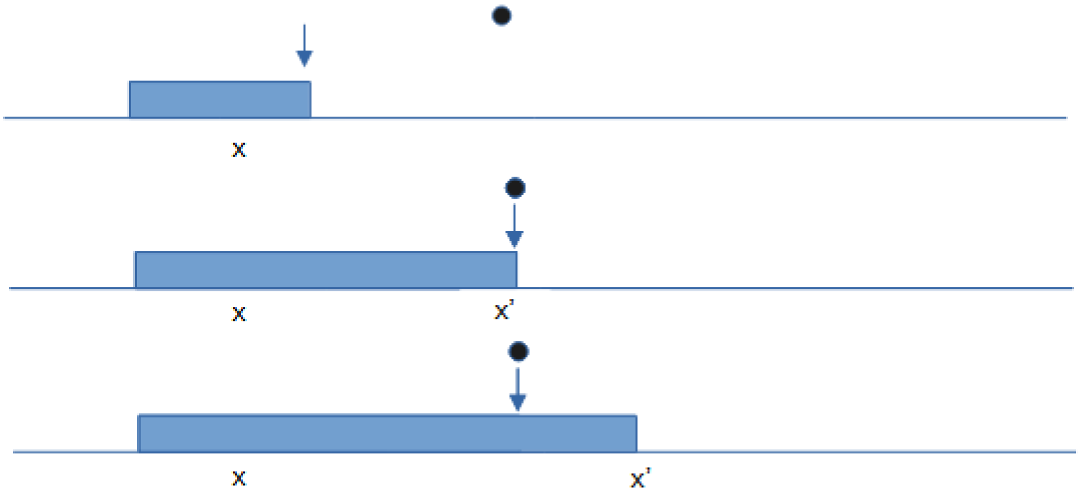
In the split case 2, we can see that it is suboptimal to ”waste” two facilities on C_j agents since the optimal clustering has proven that we only need one.

Merges. The only way for an agent to make profitable merges is to manipulate his cluster’s enumeration in the instance. What are the ways to manipulate your cluster’s position? All of the merges produce instances that are ”banned”. The first one is an instance with a singleton, which fails the mechanism singleton test. The second and third instances include splits that are either not feasible or fail the cluster-separation property. This means that there exists no profitable merge deviation. The proof plan for this case is the same as in Case 1b,2b of the OPTIMAL mechanism.

Altering cluster length, without changing its enumeration. In this case, the agent’s deviation does not modify any other cluster. He tries to include or exclude inside his cluster, the prediction that is matched to his cluster. Consider any agent i and, without loss of generality, assume that $x_i \le \hat{o}_i$, i.e., that the agent’s true preferred location is weakly on the left of the prediction. We consider two cases, depending on whether x_i is weakly greater than all the locations reported by the other agents or not. If it is, this means that if i reported truthfully, the mechanism would place the facility at x_i and i would clearly have no incentive to lie. If, on the other hand, x_i is not weakly greater than all the other reported locations, then the returned location f if i reported the truth would be on the right of x_i, i.e., $f > x_i$. However, it is easy to verify that if agent i reported a false point $x_i' < x_i$, this would not affect the outcome, and if he reported a false point $x_i' > x_i$, this could only move f further away from x_i. Therefore, it is a dominant strategy for i to report the truth.



The leftmost agent of the third cluster wants to belong in the second cluster of a modified instance. He can do that with 3 different merges. He can merge the first and second clusters, while he creates a singleton on his right. He can merge the first and second clusters, while he splits his own cluster. He can merge the first and second clusters, while he splits the fourth and his own cluster.



6.2 Conclusion and Future Work

The result of [21] has led researchers to focus on different ways to approach k-Facility Location Games, constantly trying to combine seemingly different fields in their effort to fully understand all the limitations that may exist. Although stability and prediction-enhanced algorithms are not entirely new concepts, there is still work to be done when they are applied to k-Facility Location Games. In my thesis, i have presented a mechanism that achieves 1-consistency and n-robustness but has two shortcomings. The mechanism needs at least 5 stability, like the ALMOSTRIGHTMOST mechanism and it does not examine instances that have singleton clusters, like the OPTIMAL mechanism.

One natural direction would be to examine if we can achieve lower stability or if we can find a way to deal with the strategyproof issue that is created by the existence of singleton clusters. The stability was bounded in 5, since we based much of our proof on ALMOSTRIGHTMOST. However, ALMOSTRIGHTMOST is not restricted by singleton clusters like our mechanism. Moreover, the addition of predictions was introduced, due to the gap of stability that exists between ALMOSTRIGHT-

MOST and OPTIMAL, since it still remains to be seen, if we can find a mechanism with $\gamma < 5$ stability, which also deals with singleton clusters in the instance. Finally, in [23] the impossibility result for k-Facility Location games with $k \geq 3$ was extended, proving 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 $\delta > 0$. There still remains to be seen if this bound is tight.

Bibliography

[1] P. Agrawal, E. Balkanski, V. Gkatzelis, T. Ou, and X. Tan, "Learning-augmented mechanism design: Leveraging predictions for facility location," in *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, D. M. Pennock, I. Segal, and S. Seuken, Eds. ACM, 2022, pp. 497–528. [Online]. Available: https://doi.org/10.1145/3490486.3538306

[2] N. Alon, M. Feldman, A. Procaccia, and M. Tennenholtz, "Strategyproof approximation of the minimax on networks," *Mathematics of Operations Research*, vol. 35, no. 3, pp. 513–526, 2010.

[3] N. Alon, M. Feldman, A. D. Procaccia, and M. Tennenholtz, "Strategyproof approximation of the minimax on networks," *Math. Oper. Res.*, vol. 35, no. 3, pp. 513–526, 2010. [Online]. Available: https://doi.org/10.1287/moor.1100.0457

[4] H. Angelidakis, K. Makarychev, and Y. Makarychev, "Algorithms for stable and perturbation-resilient problems," in *Proc. of the 49th ACM Symposium on Theory of Computing (STOC 2017)*, 2017, pp. 438–451.

[5] A. Antoniadis, T. Gouleakis, P. Kleer, and P. Kolev, "Secretary and online matching problems with machine learned advice," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/5a378f8490c8d6af8647a753812f6e31-Abstract.html

[6] P. Awasthi, A. Blum, and O. Sheffet, "Center-based clustering under perturbation stability," *Inf. Process. Lett.*, vol. 112, no. 1-2, pp. 49–54, 2012.

[7] Y. Azar, D. Panigrahi, and N. Touitou, "Online graph algorithms with predictions," in *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, J. S. Naor and N. Buchbinder, Eds. SIAM, 2022, pp. 35–66. [Online]. Available: https://doi.org/10.1137/1.9781611977073.3

[8] M. Balcan, N. Haghtalab, and C. White, "k-Center Clustering Under Perturbation Resilience," in *Proc. of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, ser. LIPIcs, vol. 55, 2016, pp. 68:1–68:14.

[9] M.-F. Balcan, A. Blum, and A. Gupta, "Clustering under approximation stability," *Journal of the ACM*, vol. 60, no. 2, 2013.

[10] M. Balcan and Y. Liang, "Clustering under perturbation resilience," *SIAM Journal on Computing*, vol. 45, no. 1, pp. 102–155, 2016.

[11] É. Bamas, A. Maggiori, and O. Svensson, "The primal-dual method for learning augmented algorithms," in *Advances in Neural Information Processing Systems 33: Annual*

Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/e834cb114d33f729dbc9c7fb0c6bb607-Abstract.html

[12] S. Banerjee, V. Gkatzelis, A. Gorokh, and B. Jin, "Online nash social welfare maximization with predictions," in *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, J. S. Naor and N. Buchbinder, Eds. SIAM, 2022, pp. 1–19. [Online]. Available: https://doi.org/10.1137/1.9781611977073.1

[13] Y. Bilu and N. Linial, "Are Stable Instances Easy?" in *Proc. of the 1st Symposium on Innovations in Computer Science (ICS 2010)*. Tsinghua University Press, 2010, pp. 332–341.

[14] Y. Bilu, A. Daniely, N. Linial, and M. E. Saks, "On the practically interesting instances of MAXCUT," in *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, ser. LIPIcs, N. Portier and T. Wilke, Eds., vol. 20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013, pp. 526–537.

[15] Z. Chen, K. C. 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.

[16] E. Dokow, M. Feldman, R. Meir, and I. Nehama, "Mechanism design on discrete lines and cycles," in *Proc. of the 13th ACM Conference on Electronic Commerce (EC '12)*, 2012, pp. 423–440.

[17] P. Dütting, S. Lattanzi, R. P. Leme, and S. Vassilvitskii, "Secretaries with advice," in *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, P. Biró, S. Chawla, and F. Echenique, Eds. ACM, 2021, pp. 409–429. [Online]. Available: https://doi.org/10.1145/3465456.3467623

[18] 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, p. 49, 2020.

[19] M. Feldman and Y. Wilf, "Randomized strategyproof mechanisms for Facility Location and the mini-sum-of-squares objective," *CoRR abs*, vol. 1108.1762, 2011.

[20] D. Fotakis and C. Tzamos, "Winner-imposing strategyproof mechanisms for multiple facility location games," *Theoretical Computer Science*, vol. 472, pp. 90–103, 2013.

[21] ——, "On the power of deterministic mechanisms for facility location games," *ACM Transactions on Economics and Computation*, vol. 2, no. 4, pp. 15:1–15:37, 2014.

[22] D. Fotakis, E. Gergatsouli, T. Gouleakis, and N. Patris, "Learning augmented online facility location," *CoRR*, vol. abs/2107.08277, 2021. [Online]. Available: https://arxiv.org/abs/2107.08277

[23] D. Fotakis and P. Patsilinakos, "Strategyproof facility location in perturbation stable instances," in *WINE*, ser. Lecture Notes in Computer Science, vol. 13112. Springer, 2021, pp. 95–112.

[24] D. Fotakis and C. Tzamos, "Strategyproof facility location for concave cost functions," *Algorithmica*, vol. 76, no. 1, pp. 143–167, 2016.

[25] S. Goel and W. Hann-Caruthers, "Coordinate-wise median: Not bad, not bad, pretty good," *CoRR*, vol. abs/2007.00903, 2020. [Online]. Available: https://arxiv.org/abs/2007.00903

[26] ——, “Optimality of the coordinate-wise median mechanism for strategyproof facility location in two dimensions,” *Soc. Choice Welf.*, vol. 61, no. 1, pp. 11–34, 2023. [Online]. Available: https://doi.org/10.1007/s00355-022-01435-1

[27] 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

[28] S. Im, R. Kumar, M. M. Qaem, and M. Purohit, “Online knapsack with frequency predictions,” in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 2733–2743. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/161c5c5ad51fcc884157890511b3c8b0-Abstract.html

[29] S. H. Jiang, E. Liu, Y. Lyu, Z. G. Tang, and Y. Zhang, “Online facility location with predictions,” *CoRR*, vol. abs/2110.08840, 2021. [Online]. Available: https://arxiv.org/abs/2110.08840

[30] M. Kyropoulou, C. Ventre, and X. Zhang, “Mechanism design for constrained heterogeneous facility location,” in *Proceedings of the 12th International Symposium on Algorithmic Game Theory (SAGT 2019)*, ser. Lecture Notes in Computer Science, vol. 11801. Springer, 2019, pp. 63–76.

[31] M. Li, P. Lu, Y. Yao, and J. Zhang, “Strategyproof mechanism for two heterogeneous facilities with constant approximation ratio,” in *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, 2020, pp. 238–245.

[32] P. Lu, X. Sun, Y. Wang, and Z. Zhu, “Asymptotically Optimal Strategy-Proof Mechanisms for Two-Facility Games,” in *Proc. of the 11th ACM Conference on Electronic Commerce (EC ’10)*, 2010, pp. 315–324.

[33] P. Lu, Y. Wang, and Y. Zhou, “Tighter bounds for facility games,” in *Proc. of the 5th Workshop on Internet and Network Economics (WINE ’09)*, ser. LNCS, vol. 5929, 2009, pp. 137–148.

[34] T. Lykouris and S. Vassilvitskii, “Competitive caching with machine learned advice,” *J. ACM*, vol. 68, no. 4, pp. 24:1–24:25, 2021. [Online]. Available: https://doi.org/10.1145/3447579

[35] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan, “Bilu-linial stable instances of max cut and minimum multiway cut,” in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, C. Chekuri, Ed. SIAM, 2014, pp. 890–906. [Online]. Available: https://doi.org/10.1137/1.9781611973402.67

[36] L. Mei, M. Li, D. Ye, and G. Zhang, “Facility location games with distinct desires,” *Discrete Applied Mathematics*, vol. 264, pp. 148–160, 2019.

[37] M. Mihalák, M. Schöngens, R. Srámek, and P. Widmayer, “On the complexity of the metric TSP under stability considerations,” in *SOFSEM 2011: Theory and Practice of Computer Science - 37th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, January 22-28, 2011. Proceedings*, ser. Lecture Notes in Computer Science, I. Cerná, T. Gyimóthy, J. Hromkovic, K. G. Jeffery, R. Královic, M. Vukolic, and S. Wolf, Eds., vol. 6543. Springer, 2011, pp. 382–393. [Online]. Available: https://doi.org/10.1007/978-3-642-18381-2_32

[38] M. Mitzenmacher and S. Vassilvitskii, “Algorithms with predictions,” *CoRR*, vol. abs/2006.09123, 2020. [Online]. Available: https://arxiv.org/abs/2006.09123

[39] E. Miyagawa, “Locating libraries on a street,” *Social Choice and Welfare*, vol. 18, pp. 527–541, 2001.

[40] H. Moulin, “On strategy-proofness and single-peakedness,” *Public Choice*, vol. 35, pp. 437–455, 1980.

[41] A. Muñoz Medina and S. Vassilvitskii, “Revenue optimization with approximate bid predictions,” *CoRR*, vol. abs/1706.04732, 2017. [Online]. Available: http://arxiv.org/abs/1706.04732

[42] K. Nissim, R. Smorodinsky, and M. Tennenholtz, “Approximately optimal mechanism design via Differential Privacy,” in *Proc. of the 3rd Conference on Innovations in Theoretical Computer Science (ITCS ’12)*, 2012, pp. 203–213.

[43] A. Procaccia and M. Tennenholtz, “Approximate mechanism design without money,” in *Proc. of the 10th ACM Conference on Electronic Commerce (EC ’09)*, 2009, pp. 177–186.

[44] M. Purohit, Z. Svitkina, and R. Kumar, “Improving online algorithms via ML predictions,” in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 9684–9693. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html

[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