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

Ευστάθεια υπό Διαταραχές για Σχεδιασμό Μηχανισμών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Φικιώρης Ιωάννης

Επιβλέπων: Δημήτρης Φωτάκης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

Αθήνα, Οκτώβριος 2019

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

Ευστάθεια υπό Διαταραχές για Σχεδιασμό Μηχανισμών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Φικιώρης Ιωάννης

Επιβλέπων: Δημήτρης Φωτάκης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την $23^{η}$ Οκτωβρίου 2019.

....	....	....
Δημήτρης Φωτάκης	Άρης Παγουρτζής	Ευάγγελος Μαρκάκης
Αν. Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ο.Π.Α.

Εργαστήριο Λογικής και Επιστήμης Υπολογισμών
Αθήνα, Οκτώβριος 2019

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

Περίληψη

Σε αυτή τη διπλωματική, εισάγουμε την έννοια της *Ευστάθειας υπό Διαταραχές* για *Συνδυαστικές Δημοπρασίες*, εμπνευσμένοι από την πρόσφατη δουλειά στο *Endowment Effect* και στην ευστάθεια υπό διαταραχές σε προβλήματα όπως το *Minimum Multiway Cut* και το *k-Median*. Αυτό βοηθάει στο να ξεπεράσουμε τα διάφορα φράγματα για φιλαλήθεις και μη-φιλαλήθεις μηχανισμούς σε πιο γενικές κλάσεις συναρτήσεων valuation, όπως submodular και subadditive.

Στα πλαίσια του σχεδιασμού μηχανισμών, μέσω του ορισμού μας για την ευστάθεια υπό διαταραχές, ο οποίος εκφράζει με φυσιολογικό τρόπο ότι η λύση ενός στιγμιότυπου μιας συνδυαστικής δημοπρασίας δεν πρέπει πρέπει να αλλάζει υπό μικρές διαταραχές, αποδείξαμε ότι με μια απλή παράλληλη μη-φιλαλήθη δημοπρασία μπορούμε να βρούμε εύκολα την βέλτιστη λύση σε μια ευσταθή δημοπρασία με subadditive valuations. Επίσης για ευσταθείς submodular δημοπρασίες, αποδείξαμε ότι η Παράλληλη Δημοπρασία Δεύτερης Τιμής, βρίσκει την βέλτιστη λύση, καθώς και είναι φιλαλήθης, με την έννοια της Ισορροπίας Nash. Επίσης βελτιώσαμε το προηγούμενο κάτω φράγμα του $\frac{1}{2}$ για τη δημοπρασία Kelso-Crawford για submodular συναρτήσεις, δείχνοντας ότι αν ένα στιγμιότυπο μιας submodular δημοπρασίας είναι αρκετά ευσταθές, τότε η δημοπρασία Kelso-Crawford βρίσκει πάντα την βέλτιστη λύση.

Επιπλέον, στον τομέα του *Τιμήματος της Αναρχίας* σε απλές δημοπρασίες, δείξαμε ότι το κάτω φράγματα του $\frac{1}{2}$ για *Παράλληλες Δημοπρασίες Δεύτερης Τιμής* για submodular συναρτήσεις παραμένει tight ακόμα και για λίγο ευσταθείς δημοπρασίες, έως ότου για κάποια τιμή ευστάθειας και πάνω, παίρνει εγγυημένα την τιμή 1. Για την περίπτωση των *Παράλληλων Δημοπρασιών Πρώτης Τιμής* για XOS συναρτήσεις, βελτιώσαμε το προηγούμενο κάτω φράγμα του $1 - \frac{1}{e}$, δείχνοντας ότι όσο πιο ευσταθής γίνεται ένα στιγμιότυπο μιας δημοπρασίας, τόσο αυξάνει το κάτω φράγμα του Τιμήματος της Αναρχίας, έως ότου πάρει την τιμή 1, ασυμπτωτικά.

Λέξεις Κλειδιά: Συνδυαστικές Δημοπρασίες, Σχεδιασμός Μηχανισμών, Ανάλυση πέρα από τη Χειρότερη Περίπτωση, Ευστάθεια υπό Διαταραχές, Τίμημα της Αναρχίας

Abstract

In this thesis, we introduce the notion of *Perturbation Stability* for *Combinatorial Auctions*, inspired by the recent works on the *Endowment Effect* and on the Perturbation Stability of problems like *Minimum Multiway Cut* and k*-Median*. This helps overcome the various impossibility results for truthful and non-truthful auctions for the more general valuation classes, like submodular and subadditive valuations.

In this work, for the case of pure mechanism design, via our definition for perturbation stability that conveys in a natural way that the solution of an instance of a combinatorial auction should not change under small perturbations, we proved that by using a simple non-truthful Parallel Auction, one can find the optimal allocation for stable subadditive instances. Also for stable submodular instances we showed that a Parallel Second Price Auction finds the optimal allocation, while at the same time being truthful in an *Ex-Post Equilibrium* way. We also improved the previous approximation ratio of $\frac{1}{2}$ of the Kelso-Crawford Auction for submodular valuations, by proving that, if a submodular instance is stable enough, then the Kelso-Crawford Auction finds the optimal allocation.

Additionally, for the case of *Price of Anarchy* in simple auctions, we showed that the lower bound of $\frac{1}{2}$ for *Parallel Second Price Auctions* with submodular valuations remains tight even for highly-stable instances, until it "jumps" to a guaranteed value of 1. For the case of *Parallel First Price Auctions* for XOS valuations, we improved the previous bound of $1 - \frac{1}{e}$, by showing that as an instance of a combinatorial auction becomes more stable, the lower bound for the Price of Anarchy increases, reaching the value of 1 asymptotically.

Keywords: Combinatorial Auctions, Mechanism Design, Beyond Worst Case Analysis, Perturbation Stability, Price of Anarchy

Ευχαριστίες

Φτάνοντας στο τέλος των φοιτητικών μου χρόνων, θα ήθελα να ευχαριστήσω τους ανθρώπους που με βοήθησαν να φτάσω εδώ που είμαι τώρα.

Πρώτα, θα ήθελα να ευχαριστήσω τον πατέρα και τον παππού μου που ήταν πάντα για μένα πρότυπα ως άνθρωποι και ως επιστήμονες και με ενέμπνευσαν να κάνω ό,τι έχω κάνει μέχρι σήμερα. Επίσης, θα ήθελα να ευχαριστήσω τη μητέρα μου και τον αδερφό μου που ήταν πάντα εκεί για μένα, όποτε τους χρειάστηκα.

Στη συνέχεια, θα ήθελα να ευχαριστήσω τον επιβλέποντά μου, κ. Φωτάκη, ο οποίος με ενέμπνευσε να ασχοληθώ με τον τομέα της Θεωρητικής Πληροφορικής από το πρώτο μάθημα στο οποίο τον είχα και ο οποίος με καθοδήγησε με τις συμβουλές του σε αυτό το πρώτο μου ερευνητικό έργο.

Επιπλέον, θα ήθελα να ευχαριστήσω όλους τους φίλους που έχω κάνει κατά τη διάρκεια της φοίτησής μου, οι οποίοι έκαναν τα τελευταία πέντε χρόνια αξέχαστα.

Τέλος, θα ήθελα πάνω από όλους να ευχαριστήσω την Αλεξία, με την οποία πέρασα όλες μου τις φοιτητικές στιγμές, εύκολες ή δύσκολες, και χωρίς την οποία αυτό το ταξίδι των πέντε χρόνων θα ήταν πολύ πιο δύσκολο και πολύ πιο βαρετό.

Γιάννης

Contents

Chapter 1

Εκτεταμένη Ελληνική Περίληψη

Στο κεφάλαιο αυτό, συνοψίζουμε το περιεχόμενο της παρούσας διπλωματικής, δίνοντας βασικούς ορισμούς και θεωρήματα, χωρίς αποδείξεις.

1.1 Εισαγωγή

Με τον όρο 'Σχεδιασμός Μηχανισμών' (Mechanism Design) εννοούμε τον κλάδο της Θεωρίας Παιγνίων, στον οποίο αντί να προσπαθούμε απλά να προβλέψουμε την εγωιστική συμπεριφορά των παιχτών, προσπαθούμε να την ρυθμίσουμε μέσω μιας διαδικασίας-μηχανισμού. Πιο συγκεκριμένα, σκοπός μας είναι, μέσω κάποιας αυτοματοποιημένης διαδικασίας (δηλαδή του μηχανισμού), τα εγωιστικά τους συμφέροντα να παραλληλίζονται με το κοινωνικό καλό.

Σε κάθε στιγμιότυπο ενός μηχανιμσού έχουμε ένα σύνολο N από παίχτες, με πληθάριθμο n. Επίσης υπάρχει ένα σύνολο $\mathcal{O}$ από πιθανές εκβάσεις του μηχανισμού. Κάθε παίχτης i έχει μια ιδιωτική συνάρτηση συνόλου v_i, η οποία λέγεται *valuation function* και αντιστοιχεί κάθε πιθανή έκβαση σε μια μη αρνητική πραγματική τιμή:

$$v_i : \mathcal{O} \rightarrow \mathbb{R}^+$$

Ένα μηχανισμός δέχεται κάποια είσοδο από τους παίχτες και αποφασίζει την τελική έκβαση ω από το σύνολο $\mathcal{O}$, καθώς και κάποια πληρωμή που πρέπει να κάνει κάθε παίχτης. Σκοπός των πληρωμών είναι να υποχρεώσουν τους παίχτες να 'βοηθήσουν' στην μεγιστοποίηση της κοινωνικής ευημερίας (*Social Welfare*), που είναι ίσο με το το άθροισμα $\sum_i v_i(\omega)$. Σκοπός των παιχτών είναι να μεγιστοποιήσουν την χρησιμότητά τους (*utility*) που είναι ίση με το valuation της έκβασης μείον την πληρωμή που τους ζητήθηκε να πληρώσουν.

Η πιο απλή κατηγορία μηχανισμών είναι αυτή της άμεσης αποκάλυψης (*Direct Revelation*). Εδώ οι παίχτες δίνουν στον μηχανισμό ως είσοδο κάποια συνάρτηση, όπου άμα πουν την αλήθεια θα είναι το πραγματικό valuation τους. Σε αυτή την κατηγορία μηχανισμών σκοπός του Σχεδιασμού Μηχανισμών είναι η υλοποίηση μιας διαδικασίας-μηχανισμού, στην οποία μέσω των πληρωμών που θα υπολογιστών μετά την αποκάλυψη των συναρτήσεων των παιχτών, οι παίχτες να 'υποχρεώνονται' να πουν την αλήθεια. Ένας μηχανισμός που το πετυχαίνει αυτό λέγεται φιλαλήθης και αυτό που πετυχαίνει είναι η στρατηγική των παιχτών που προσδίδει τη μέγιστη χρησιμότητα να είναι να πουν την αλήθεια, δηλαδή να αποκαλύψουν την πραγματική συνάρτησή τους.

Ο πιο γνωστός μηχανισμός που καταφέρνει να είναι ταυτόχρονα και φιλαλήθης και να μεγιστοποιεί την κοινωνική ευημερία είναι ο VCG, ο οποίος παρουσιάζεται αναλυτικά στην ενότητα 3.4. Μέσω κάποιων περίπλοκων πληρωμών η μεγιστοποίηση της χρηστικότητα των παιχτών συμβαίνει μόνο όταν μεηιστοποιείται και η κοινωνική ευημερία, το οποίο σημαίνει ότι συμφέρει τους παίχτες να αποκαλύψουν τα πραγματικά τους valuations. Δυστυχώς ο μηχανιμσός αυτός είνα αδύνατο να εφαρμοστεί στην πράξη διότι η υλοποίησή του συνήθως απαιτεί εκθετικό χρόνο.

Ο τρόπος με τον οποίο περιγράφηκαν πριν οι μηχανισμοί είναι πολύ γενικός. Σε αυτή την εργασία μας ενδιαφέρουν οι συνδυαστικές δημοπρασίες (*Combinatorial Auctions*) όπου θεωρούμε ότι έχουμε ένα σύνολο από διαφορετικά αντικείμενα M, πληθικότητας m, και το valuation function κάθε παίκτη δέχεται σαν όρισμα ένα υποσύνολο αντικειμένων του M, που είναι τα αντικείμενα που δέχεται στο τέλος της δημοπρασίας. Εδώ, η ποσότητα της κοινωνικής ευημερίας ισούται με

$$\sum_{i=1}^{n} v_i(S_i)$$

όπου κάθε S_i είναι ένα υποσύνολο του M, ξένο από τα υπόλοιπα: $S_i \cap S_k = \emptyset$, όταν $i \neq k$.

Μια πολύ σημαντική έννοια στην Σχεδιασμό Μηχανισμών και γενικότερα στην οικονομική θεωρία είναι αυτή της Walrasian Ισορροπίας (*Walrasian Equilibrium*).

Ορισμός 1 (Walrasian Ισορροπία)**.** Walrasian Ισορροπία ονομάζεται ένα διάνυσμα τιμών $\mathbf{p}$ και μια διανομή των αντικειμένων $S_1, ..., S_n$, τέτοια ώστε κάθε παίχτης i να είναι ὲυχαριστημένος΄, δηλαδή με βάση το διάνυσμα τιμών $\mathbf{p}$ το σύνολο S_i που το έχει δωθεί να μεγιστοποιεί την χρηστικότητά του

$$v_i(S_i) - \sum_{j \in S_i} \geq v_i(S) - \sum_{j \in S}$$

για κάθε υποσύνολο αντικειμένων $S \subseteq M$.

Το γιατί αυτή η έννοια είναι τόσο σημαντική είναι εύκολο να το καταλάβει κανείς: Άμα βρούμε κάποιο διάνυσμα τιμών που ικανοποιεί την συνθήκη αυτή, τότε για όλους τους παίχτες η πιο απλή και φιλαλήθης στρατηγική είναι απλά να πάρουν το υποσύνολο αντικείμενων που προτιμούν. Το επόμενο θεώρημα αναδεικνύει πραγματικά πόσο δυνατή είναι η Walrasian Ισορροπία.

Θεώρημα 1 (1ο Θεώρημα Φιλαληθίας)**.** *Έστω μια Walrasian Ισορροπία, με διάνυσμα τιμών* $\mathbf{p}$ *και διανομή* $S_1, ..., S_n$*. Τότε η διανομή* $S_1, ..., S_n$ *μεγιστοποιεί την κοινωνική ευημερία.*

Έτσι μας γίνεται εύκολα κατανοητό ποια είναι η δύναμη της Walrasian Ισορροπίας και πόσο απλό είναι να χρησιμοποιήσουμε τις τιμές της άμα τις ξέρουμε. Δυστυχώς το μεγαλύτερο πρόβλημα της είναι ότι η ύπαρξή της είναι εγγυημένη μόνο σε περιορισμένο σύνολο valuation function, γνωστό ως *Gross-Substitutes.*

Πριν προχωρήσουμε στο επόμενο κεφάλαιο, θα ορίσουμε τώρα τις 3 βασικές κλάσεις από συναρτήσεις valuation, με τις οποίες θα ασχοληθούμε αργότερα.

Ορισμός 2 (Submodular Συναρτήσεις)**.** Μια συνάρτηση $v : M \rightarrow \mathbb{R}^+$ λέγεται *submodular*, αν για κάθε υποσύνολα αντικειμένων $S \subseteq T \subseteq M$ και για κάθε αντικείμενο $j \notin S$ ισχύει

$$v(S \cup \{j\}) - v(S) \geq v(T \cup \{j\}) - v(T)$$

Ορισμός 3 (XOS Συναρτήσεις)**.** Μια συνάρτηση $v : M \to \mathbb{R}^+$ λέγεται *XOS*, αν υπάρχουν διανύσματα $(\mathbf{a}_1, ..., \mathbf{a}_r)$ με μη αρνητικές τιμές, όπου για κάθε l, $\mathbf{a}_l = (a_{l1}, ..., a_{lm})$, ώστε να ισχύει για κάθε $S \subseteq M$

$$v(S) = \max_{l=1}^{r} \sum_{j \in S} a_{lj}$$

Ορισμός 4 (Subadditive Συναρτήσεις)**.** Μια συνάρτηση $v : M \to \mathbb{R}^+$ λέγεται *subadditive*, αν για κάθε υποσύνολα αντικειμένων $S, T \subseteq M$ ισχύει

$$v(S) + v(T) \geq v(S \cup T)$$

1.2 Πέρα της Ανάλυσης Χειρότερης Περίπτωσης

Αυτό είναι ένα σύντομο κεφάλαιο όπου απλά θα αναφέρουμε την έννοια της Ανάλυσης πέρα από τη Χειρότερη Περίπτωση (*Beyond Worst Case Analysis*). Μέσω της ανάλυσης αυτής προσπαθούμε να ξεπεράσουμε προβλήματα που δημιουργεί η ανάλυση της χειρότερης περίπτωσης. Το πιο χαρακτηριστικό παράδειγμα τέτοιου προβλήματος είναι η μέθοδος Simplex που χρησιμοποιείται για την επίλυση προβλημάτων Γραμμικού Προγραμματισμού. Παρόλο που για τη μέθοδο αυτή έχουν βρεθεί παραδείγματα για τα οποία η επίλυσή τους χρειάζεται εκθετικό χρόνο, είναι η πιο ευρύτερα χρησιμοποιούμενη μέθοδος για το πρόβλημα αυτό. Οπότε γιατί χρησιμοποιούμε αυτή την μέθοδο και όχι κάποια άλλη που έχει αποδειχθεί ότι λύνει οποιοδήποτε πρόβλημα σε πολυωνυμικό χρόνο·

Η απάντηση σε αυτό το ερώτημα δίνεται μέσω μιας μεθόδου ανάλυσης που λέγεται Εξομαλυμένης Ανάλυσης (*Smoothed Analysis.*) Με αυτή την ανάλυση μας ενδιαφέρει όχι ο χρόνος εκτέλεσης ενός αλγορίθμου στη χειρότερη περίπτωση, αλλά αλλά σε μια τυχαία περιοχή γύρω από τη χειρότερη περίπτωση. Πιο συγκεκριμένα, στη μέθοδο Simplex μας ενδιαφέρει ο μέσος χρόνος που χρειάζεται ένα στιγμιότυπο να τρέξει άμα του προσθέσουμε στα στοιχεία της εισόδου τυχαίο Γκαουσσιανό Θόρυβο με τυπική απόκλιση σ. Έχει αποδειχθεί το παρακάτω θεώρημα

Θεώρημα 2. *Σε ένα πρόβλημα Γραμμικού Προγραμματισμού με d μεταβλητές και n περιορισμούς, άμα προστεθεί θόρυβος Γκάους τυπικής απόκλισης σ, στα στοιχεία της εισόδου για τους περιορισμούς, ο εκτιμώμενος χρόνος τρεξίματος είναι*

$$O(d^2 \sigma^{-2} \sqrt{\log n} + d^3 \log^{3/2} n)$$

Βλέπουμε με αυτό το θεώρημα ότι η πιθανότητα να χρειαστεί η μέθοδος Simplex εκθετικό χρόνο είναι πολύ μικρή άμα τροποποιήσουμε λιγάκι την αρχική είσοδο. Έτσι γίνεται εύκολα κατανοητό γιατί η μέθοδος Simplex προτιμάται από άλλους μεθόδους, παρόλο που με την κλασσική έννοια είναι χειρότερη.

Μια άλλη μέθοδος ανάλυσης πέρα από την ανάλυση χειρότερης περίπτωσης, είναι αυτή της Ευστάθειας υπό Διαταραχές (*Perturbation Stability*). Με αυτή την μέθοδο προσπαθούμε να χαρακτηρίσουμε κάποια στιγμιότυπα χωρίς νόημα και κάποια με νόημα. Γενικά στιγμιότυπα με νόημα θεωρούνται αυτά που έχουν δυνατή βέλτιστη λύση. Για να γίνει αυτή η έννοια πιο κατανοητή ας σκεφτούμε το πρόβλημα του Clustering.

Στο πρόβλημα του clustering προσπαθούμε να χωρίσουμε σημεία κάποιου χώρου σε ομάδες, ώστε σημεία σε ίδια ομάδα να έχουν μικρή απόσταση, ενώ σημεία σε διαφορετικές ομάδες να έχουν μεγαλύτερες αποστάσεις. Αυτό το πρόβλημα είναι NP-Hard. Παρόλο της δυσκολίας όμως να βρούμε τη λύση σε ένα οποιοδήποτε στιγμιότυπο μπορούμε να βρούμε εύκολα την βέλτιστη λύση σε στιγμιότυπα με νόημα. Για να ποσοτικοποιήσουμε πόσο νόημα έχει ένα στιγμιότυπο, λέμε ότι είναι γ-ευσταθές αν μπορούμε να μειώσουμε την απόσταση οποιοδήποτε 2 σημείων κατά παράγοντα γ και η βέλτιστη λύση να παραμείνει ίδια. Αυτή η έννοια ποσοτικοποιεί πόσο ευσταθής είναι η βέλτιστη λύση. Ισχύει το παρακάτω θεώρημα.

Θεώρημα 3. *Αν ένα στιγμιότυπο clustering είναι 2-ευσταθές, τότε μπορούμε να βρούμε την βέλτιστη λύση σε πολυωνυμικό χρόνο.*

1.3 Ευστάθεια υπό Διαταραχές σε Συνδυαστικές Δημοπρασίες

Τώρα θα εφαρμόσουμε τις έννοιες της προηγούμενης ενότητας σε συνδυαστικές δημοπρασίες. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε την έννοια της Ευστάθειας υπό Διαταραχές. Πριν από αυτό όμως θα ορίσουμε την έννοια του (*Endowment Effect*) σε συνδυαστικές δημοπρασίες. Σύμφωνα με το Endowment Effect, άμα κάποιος παίχτης έχει συνάρτηση valuation $v(\cdot)$, και του δώσουμε κάποιο υποσύνολο αντικειμένων S, τότε αξία του για τα αντικείμενα αυτά θα αυξηθεί και η καινούργια του συνάρτηση θα γίνει $v^{S,\alpha}(\cdot)$, όπου

$$v^{S,\alpha}(T) = v(T) + (\alpha - 1) \cdot v(S \cap T)$$

όπου $T \subseteq M$ και το α δείχνει πόσο ισχυρό είναι το Endowment Effect: Άμα $\alpha = 1$, τότε οι συναρτήσεις παραμένουν ίδιες και πρακτικά δεν υπάρχει Endowment Effect, ενώ όσο μεγαλώνει το α η επιπρόσθετη αξία που αποκτούν τα αντικείμενα του συνόλου S ως προς το οποίο γίνεται το endowment, αυξάνεται.
Τώρα με την έννοια του Endowment Effect, μπορούμε να ορίσουμε πότε ένα στιγμιότυπο μιας συνδιαστικής δημοπρασίας έχει νόημα, δηλαδή τον ορισμό του πότε είναι γ-ευσταθές.

Ορισμός 5. Μια συνδυαστική δημοπρασία λέγεται γ-ευσταθής αν η βέλτιστη διαμέριση των αντικειμένων είναι μοναδική και παραμένει μοναδική άμα κάνουμε endow σε έναν αυθαίρετο παίχτη ένα αυθαίρετο αντικείμενο κατά γ.

Με απλούστερα λόγια, για όποιον παίχτη $i \in N$ και αντικείμενο $j \in M$, η βέλτιστη λύση στις συναρτήσεις $v_1(\cdot), ..., v_n(\cdot)$, παραμένει βέλτιστη στις συναρτήσεις $v'_1(\cdot), ..., v'_n(\cdot)$, όπου:

- Αν $k \neq i$, τότε $v'_k(S) = v_k(S)$, $\forall S \subseteq M$
- Για τον παίχτη i, $v'_i(S) = v_i(S) + (\gamma - 1) \cdot v_i(S \cap \{j\})$, $\forall S \subseteq M$

Το πιο άμεσο αποτέλεσμα του παραπάνω ορισμού είναι το λήμμα που ακολουθεί, που δείχνει τον τρόπο με τον οποίο οι ευσταθείς δημοπρασίες είναι πιο εύκολες από τις γενικές δημοπρασίες.

Λήμμα 1. *Σε μια συνδυαστική δημοπρασία, η οποία είναι γ-ευσταθής, για κάθε ζευγάρι παιχτών i, k και για κάθε αντικείμενο j που ανήκει στο σύνολο O_i, που είναι τα αντικείμενα που δίνονται στον παίχτη i στην βέλτιστη διανομή, ισχύει*

$$v_i(O_i) - v_i(O_i - \{j\}) > (\gamma - 1) \cdot v_k(j)$$

Με αυτό το λήμμα θα μπορέσουμε να κάνουμε όλες τις αποδείξεις των επόμενων κεφαλαίων, όπου θα δείξουμε τι πλεονέκτημα έχουν οι ευσταθείς δημοπρασίες σε σχέση με τις γενικές δημοπρασίες.

1.4 Μηχανισμοί με ή χωρίς ευστάθεια

Σε αυτό το κεφάλαιο θα δούμε κάποια από τα πιο διάσημα αποτελέσματα για μηχανισμούς για συνδυαστικές δημοπρασίες, καθώς και πώς αυτά μπορούν να βελτιωθούν για ευσταθείς μηχανισμούς.

Αρχικά, θα ορίσουμε την έννοια του αιτήματος αξίας (*Value Query*), στο οποίο παρουσιάζουμε σε κάποιον παίχτη ένα υποσύνολο από αντικείμενα και αυτός μας απαντάει με την αξία του για αυτά τα αντικείμενα. Χρειαζόμαστε τα αιτήματα αξίας γιατί αν θέλαμε να υλοποιήσουμε έναν μηχανισμό άμεσης αποκάλυψης χωρίς αυτά, θα έπρεπε να μάθουμε όλη τη συνάρτηση αξίας κάθε παίχτη, της οποίας το μέγεθος είναι εκθετικό ως προς το m.

Θα αναφερθούμε εδώ σε δύο μηχανισμούς που χρησιμοποιούν αιτήματα αξίας. Ο πρώτος είναι για subadditive συναρτήσεις και χωρίς να είναι φιλαλήθης πετυχαίνει λόγο προσέγγισης το πολύ 2, ενώ ο δεύτερος είναι φιλαλήθης για submodular συναρτήσεις και πετυχαίνει λόγο προσέγγισης το πολύ $\sqrt{m}$. Και για τους 2 έχει αποδειχθεί ότι δεν μπορούν να βελτιωθούν και να έχουν καλύτερους λόγους προσέγγισης.

Εμείς, χρησιμοποιώντας την έννοια της ευστάθειας σπάσαμε τα παραπάνω φράγματα με τον εξής απλό μηχανισμό:

Algorithm 1 Παράλληλη Δημοπρασία

1: Θέσε $S_1 = S_2 = ... = S_n = \emptyset$
2: **for** $j \in M$ **do**
3:     Έστω i ο παίχτης που μεγιστοποιεί το $v_i(j)$
4:     Δώσε το αντικείμενο j στον παίχτη i, δηλαδή θέσε $S_i \leftarrow S_i \cup \{j\}$
5: **end for**
6: **επέστρεψε** τη διανομή $(S_1, ...S_n)$

Αποδείξαμε ότι για subadditive δημοπρασίες που είναι 2-ευσταθείς, ο παραπάνω μηχανισμός (χωρίς να είναι φιλαλήθης), βρίσκει την βέλτιστη λύση.

Θεώρημα 4. *Έστω μια συνδυαστική δημοπρασία με subadditive συναρτήσεις. Τότε ο μηχανισμός 1 βρίσκει την βέλτιστη διανομή των αντικειμένων σε πολυωνυμικό χρόνο, κάνοντας μόνο $n \cdot m$ αιτήματα αξίας.*

Άμα στον μηχανισμό 1 προσθέσουμε και πληρωμές, μπορούμε να τον κάνουμε φιλαλήθη για την πιο περιορισμένη κλάση των submodular συναρτήσεων. Κάθε παίχτης i που παίρνει ένα σύνολο S_i, πληρώνει το άθροισμα των μέγιστων αξιών των άλλων παιχτών για τα αντικείμενα του S, δηλαδή αν ο παίχτης k για το αντικείμενο j κατέθεσε τιμή b_{kj}, τότε ο παίχτης i που πήρε το σύνολο S_i πληρώνει

$$P_i = \sum_{j \in S_i} \max_{k \neq i} b_{kj}$$

Αυτός ο μηχανισμός είναι γνωστός ως Παράλληλη Δημοπρασία Δεύτερης Τιμής *Parallel Second Price Auction.*

Θεώρημα 5. *Η Παράλληλη Δημοπρασία Δεύτερης Τιμής για 2-ευσταθείς συνδυαστικές δημοπρασίες με submodular είναι φιλαλήθης με την έννοια του ex-Post Incentive Compatibility, δηλαδή αν όλοι οι παίχτες λένε την αλήθεια, δεν συμφέρει κανέναν να αλλάξει στρατηγική.*

Προηγουμένως αναφέραμε τα αιτήματα αξίας, τα οποία είναι ένας τρόπος για τον μηχανισμό να αλληλεπιδρά με τους παίχτες. Υπάρχει ένας άλλος γνωστός τρόπος για τον μηχανισμό να αλληλεπιδρά με τους παίχτες, τα *demand queries.* Εδώ παρουσιάζουμε σε έναν παίχτη μια τιμή για κάθε αντικείμενο και αυτός μας απαντάει με το σύνολοπου μεγιστοποιεί την χρηστικότητά του, δηλαδή αν $v(\cdot)$ είναι η συνάρτηση αξίας του και η τιμή που δίνουμε σε κάθε αντικείμενο j είναι p_j, τότε η απάντησή του είναι το σύνολο S, για το οποίο ισχύει

$$v(S) - \sum_{j \in S} p_j \geq v(T) - \sum_{j \in T} p_j, \forall T \subseteq M$$

Ένας διάσημος μηχανισμός που χρησιμοποιεί demand queries είναι ο Kelso-Crawford. Αυτός ο μηχανισμός αυξάνει σταδιακά τις τιμές για τα αντικείμενα που θέλουν οι παίχτες: Σε κάθε γύρο κάθε παίχτης έχει κάποια αντικείμενα και δηλώνει (μέσω ενός demand query) ποια επιπλέον αντικείμενα θέλει για να μεγιστοποιήσει την χρηστικότητά του. Στη συνέχεια για τα επιπρόσθετα αντικείμενα κάποιου παίχτη αυξάνονται οι τιμές τους και δίνονται στον παίχτη. Αυτό επαναλαμβάνεται μέχρι οι τιμές να γίνουν αρκετά ψηλές ώστε κανένας παίχτης να μην θέλει να πάρει επιπλέον αντικείμενα.

Algorithm 2 Η δημοπρασία Kelso-Crawford

Θέσε $S_1 = S_2 = ... = S_n = \emptyset$
Θέσε $p_1 = p_2 = ... = p_m = 0$
while true **do**
    Ρώτα κάθε παίχτη i για το T_i, που μεγιστοποιεί $v_i(S_i \cup T_i) - \sum_{j \in S_i} p_j - \sum_{j \in T_i} (p_j + \epsilon)$
    Αν για κάθε παίχτη i ισχύει $T_i = \emptyset$, **επέστρεψε** τη διανομή $(S_1, ..., S_n)$
    Διαφορετικά **διάλεξε** τυχαία παίχτη i για τον οποίο $T_i \neq \emptyset$.
    Θέσε $S_i \leftarrow S_i \cup T_i$
    Για κάθε $k \neq i$, **θέσε** $S_k \leftarrow S_k - T_i$
    Για κάθε $j \in T_i$ **θέσε** $p_j \leftarrow p_j + \epsilon$
end while

Παρόλο που αυτός ο αλγόριθμος δεν είναι φιλαλήθης, είναι αρκετά διάσημος, διότι υπολογίζει για την κλάση των Gross-Substitutes συναρτήσεων μια Walrasian Ισορροπία, αποδεικνύοντας έτσι την ύπαρξή τους σε αυτή την κλάση. Για submodular συναρτήσεις υπολογίζει μια λύση που είναι 2-προσεγγιστική της βέλτιστης.

Εμείς δείξαμε ότι και αυτός ο μηχανισμός βελτιώνεται αν η το στιγμιότυπο είναι ευσταθές. Πιο συγκεκριμένα, αν είναι 3-ευσταθές πάντα βρίσκει την βέλτιστη διανομή των αντικειμένων.

Θεώρημα 6. *Άμα χρησιμοποιήσουμε τη δημοπρασία Kelso-Crawford σε μια συνδυαστική δημοπρασία με submodular συναρτήσεις που είναι 3-ευσταθής, τότε αντί για μία 2 προσεγγιστικά λύση, βρίσκουμε την βέλτιστη.*

Σε αυτό το κεφάλαιο είδαμε πώς αρκετοί απλοί και διαισθητικοί μηχανισμοί που δεν εγγυώνται τόσο καλά αποτελέσματα, μπορούν να βρουν την βέλτιστη διανομή των αντικειμένων.

1.5 Το Τίμημα της Αναρχίας σε Δημοπρασίες

Σε αυτό το κεφάλαιο θα δούμε το Τίμημα της Αναρχίας (*Price Of Anarchy*) σε συνδυαστικές δημοπρασίες. Πριν ορίσουμε τι είναι αυτό, θα πρέπει να δώσουμε τον ορισμό για 2 βασικές ισορροπίες Nash. Η πρώτη είναι η αμιγής ισορροπία Nash(*Pure Nash Equilibrium*) στην οποία οι παίχτες δεν επιτρέπεται να έχουν τυχαίες στρατηγικές.

Ορισμός 6 (Αμιγής Ισορροπία Nash)**.** Σε μια συνδυαστική δημοπρασία ένα προφίλ από bids $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_n)$ αποτελεί Αμιγή Ισορροπία Nash, σε μια παράλληλη δημοπρασία Πρώτης ή Δεύτερης Τιμής όπου κάθε αντικείμενο καταλήγει με τιμή $p_j(\mathbf{b})$, αν κάθε παίχτης i, για κάθε διαφορετικό διάνυσμα από bids $\mathbf{b}'_i$, δεν τον συμφέρει να ποντάρει $\mathbf{b}'_i$, δηλαδή

$$v_i(S_i(\mathbf{b})) - \sum_{j \in S_i(\mathbf{b})} p_j(\mathbf{b}) \geq v_i(S_i(\mathbf{b}'_i, \mathbf{b}_{-i})) - \sum_{j \in S_i(\mathbf{b}'_i, \mathbf{b}_{-i})} p_j(\mathbf{b}'_i, \mathbf{b}_{-i})$$

Στη συνέχεια έχουμε την Ανάμεικτη Ισορροπία Nash(*Mixed Nash Equilibrium*), στην οποία επιτρέπονται στρατηγικές με τυχαιότητα

Ορισμός 7 (Ανάμεικτη Ισορροπία Nash)**.** Σε μια συνδυαστική δημοπρασία ένα προφίλ από κατανομές από bids $\mathbf{D} = (D_1, ..., D_n)$, αποτελεί μια Ανάμεικτη Ισορροπία Nash, σε μια παράλληλη δημοπρασία Πρώτης ή Δεύτερης Τιμής όπου κάθε αντικείμενο καταλήγει με τιμή $p_j(\mathbf{b})$, αν κάθε παίχτης i, για κάθε διαφορετική κατανομή D'_i, δεν συμφέρει τον i να αλλάξει την στρατηγική του σε D'_i, δηλαδή

$$\mathop{\mathbb{E}}_{\mathbf{b} \sim \mathbf{D}} \left(v_i(S_i(\mathbf{b})) - \sum_{j \in S_i(\mathbf{b})} p_j(\mathbf{b}) \right) \geq \mathop{\mathbb{E}}_{\mathbf{b} \sim (D'_i, \mathbf{D}_{-i})} \left(v_i(S_i(\mathbf{b})) - \sum_{j \in S_i(\mathbf{b})} p_j(\mathbf{b}) \right)$$

Έχοντας ορίσει τις διάφορες ισορροπίες Nash, μπορούμε να ορίσουμε το τίμημα της Αναρχίας: Είναι ίσο με τον ελάχιστο λόγο της κοινωνικής ευημερίας όλων των Ισορροπιών, προς τη βέλτιστη κοινωνική ευημερία. Για Ανάμεικτες Ισορροπίες στον αριθμητή έχουμε τη μέση τιμή της ισοροπίας. Πιο συγκεκριμένα:

$$POA = \frac{\mathbb{E}_{\mathbf{b} \sim \mathbf{D}} \left(\sum_{i \in N} v_i(S_i(\mathbf{b})) \right)}{\sum_{i \in N} v_i(O_i)}$$

Αρχικά θα αναφερθούμε σε Παράλληλες Δημοπρασίες Πρώτης τιμής, που είναι παρόμοιες με αυτές της Δεύτερης Τιμής, αλλά κάθε παίχτης για κάθε αντικείμενο που πήρε πληρώνει την τιμή που πόνταρε. Για αυτές τις δημοπρασίες, έχει αποδειχθεί ότι για Αμιγείς Ισορροπίες το Τίμημα της Αναρχίας είναι πάντα 1, το οποίο σημαίνει ότι βρίσκεται πάντα βέλτιστη διανομή των αγαθών. Δυστυχώς, Αμιγείς Ισορροπίες σε Παράλληλες Δημοπρασίες Πρώτης τιμής υπάρχουν αν και μόνο αν υπάρχει Walrasian Ισορροπία, το οποίο όπως αναλύσαμε και προηγουμένως δεν είναι εγγυημένο για τις περισσότερες κλάσεις συναρτήσεων.

Όσον αφορά Ανάμεικτες Ισορροπίες σε Παράλληλες Δημοπρασίες Πρώτης τιμής, τα πράγματα είναι πιο ενδιαφέροντα: Έχει αποδειχθεί ότι το τίμημα της Αναρχίας για XOS συναρτήσεις είναι τουλάχιστον $1 - \frac{1}{e}$, καθώς και ότι υπάρχουν ισορροπίες που έχουν αυτό το Τίμημα. Αυτό σημαίνει ότι χωρίς κάποιο παραπάνω στοιχείο δεν γίνεται να εγγυηθούμε μεγαλύτερο Τίμημα Αναρχίας για αυτήν την περίπτωση.

Τα πράγματα αλλάζουν αν υποθέσουμε ότι οι δημοπρασίες μας είναι ευσταθείς. Αποδείξαμε ότι ασυμπτωτικά ως προς το γ οι Παράλληλες Δημοπρασίες Πρώτης τιμής είναι βέλτιστες. Πιο συγκεκριμένα

Θεώρημα 7. *Σε κάθε γ-ευσταθή συνδυαστική δημοπρασία με XOS συναρτήσεις το τίμημα της Αναρχίας για Μικτές Στρατηγικές είναι τουλάχιστον $\frac{\gamma-2}{\gamma-1}$.*

Μεγαλύτερο ενδιαφέρον για το κομμάτι της ευστάθειας παρουσιάζουν οι Παράλληλες Δημοπρασίες Δεύτερης τιμής. Εδώ επειδή οι στρατιγικές των παικτών είναι πιο απλές, θα αναφερθούμε μόνο σε Αμιγείς Ισορροπίες. Στο γενικότερο πλαίσιο έχει αποδειχθεί ότι για XOS συναρτήσεις το τίμημα της Αναρχίας είναι πάντα μεγαλύτερο από $\frac{1}{2}$, καθώς και έχει βρεθεί παράδειγμα με τέτοιο Τίμημα.

Άμα υποθέσουμε ότι οι δημοπρασίες μας είναι και γ-ευσταθείς, μπορούμε για XOS συναρτήσεις να αποδείξουμε λόγο παρόμοιο με αυτόν που αποδείξαμε και στις Παράλληλες Δημοπρασίες Πρώτης τιμής, δηλαδή ότι το Τίμημα την Αναρχίας είναι τουλάχιστον $\frac{\gamma-2}{\gamma-1}$.

Τα πιο εντυπωσιακά αποτελέσματα μας όμως είναι για γ-ευσταθείς δημοπρασίας με submodular συναρτήσεις. Αποδείξαμε ότι το Τίμημα της Αναρχίας συνεχίζει να μπορεί να είναι $\frac{1}{2}$, όσο $\gamma < 3$. Τα πράγματα όμως αλλάζουν ραγδαία όμως όταν το γ γίνει 3 και πάνω.

Θεώρημα 8. *Σε κάθε 3-ευσταθή συνδυαστική δημοπρασία με submodular συναρτήσεις το τίμημα της Αναρχίας είναι πάντα 1.*

Chapter 2

Introduction

Imagine that you have in possession of an important painting, for which you no longer have any use of, maybe because your walls are now full of even more important paintings. Because the painting is important and you have no use for more money, you want to make sure that it is taken by someone who really values its worth, despite how much he pays. Basically, you wish to sell it to the person who values it more.

In terms of *Mechanism Design*, this is called an auction, where the goal is to maximize the *Social Welfare*, meaning the total value produced when a good is allocated. In a more complex case, there might be many different goods, which depend highly on each other: For example, when auctioning two paintings, a bidder might not value both paintings together higher than what he values a single painting. A simple explanation for this is that he might not have enough space for both paintings and simply have use of only one of them. The opposite is also a probable scenario: Each painting alone might have small value, but together their value may be tenfold the sum of their independent value, because together form some part of a collection.

Unfortunately, although this problem might sound easy, another difficulty has to be taken into account. To illustrate this, consider a naive solution to the first problem with the single painting: Simply ask each bidder for their value and give the painting for free to the one that says the highest number. This would simply not work, because as the "algorithm" implies, the painting is not given to the bidder with the highest value, but to the bidder who can come up with a number higher than that of the other bidders, i.e. there is no guarantee that anyone will say the truth.

To counter this problem, bidders are required to give some sort of payment in order to show that their value is as high as they claim it to be. This can be modeled by imagining that each player wants to maximize his *utility*, i.e. the value he gets from the items he acquires, minus the payment he is asked to make.

All the above ideas, are the primary concerns in the field of *Game Theory*, where any setting involves a set of players who try to selfishly satisfy their own goals. *Mechanism Design*, the science of rule making, tries to regulate this selfish behavior, in order to achieve a common goal, like the maximization of Social Welfare. Mechanism Design could also be characterized as reverse game theory, because instead of analyzing and predicting the selfish behavior of the players in a system, it tries to create a system in which the best strategy of every player is to be truthful.

2.1 Single-Parameter Auctions

In this section we are going into a little more detail about auctions, where the whole information that each bidder has private and is not publicly known can be represented with a single real number. These are the simplest environments one can have. For more details see section 3.2.

The setting is the one described above: We have 1 item and n bidders and we want to give the item to the bidder with the highest value for the item, information which is hidden from everyone else. In order to do that, we are going to ask everyone to submit a *bid*, which if the players play truthfully should be their hidden value. The first naive solution is to give the item for free to the bidder who bids the highest value. As explained before this auction is doomed to fail, as every bidder is going to be completely dishonest and try to simply outbid everyone else no matter his value.

The second solution is the following: Simply give the item to the bidder with the highest bid and have him pay his bid. This may sound like a good and natural solution at first, but it has a number of problems: Even the bidder who should win the item has no clear strategy, because he does not know the values of the other bidders. He does not know whether the other bidders have values close to his, in which case he should bid close to his value, or if the other bidders have values much lower than him, in which case he should bid low. This auction, known as *First Price Auction*, favors bidder who are likely to take risk and not necessarily the bidder who should win the item.

The solution to this problem came from Vickrey, in 1961. [1] shows that the *Vickrey-Auction* has all the properties that one would want: It "forces" the bidders to truthfully bid their value, by making this strategy strictly better than any else, and also finds the bidder whose value is the highest and allocates him the item. All these are achieved simply by having the bidder with the highest bid, pay the second highest bid. Because of this payment rule, the Vickrey auction is often called *Second-Price Auction*.

In 1981, Myerson generalized this result for auctions where more than one items can be allocated and the bidders' private information again can be represented with a real number. In order to make this setting more understood, think of an item for which we have several copies and each bidder is interested in only one copy. *Myerson's Lemma* ([6]) states that a mechanism "forces" truthfulness and achieves maximum social welfare if and only if it is monotone and makes each bidder pay his externality. By monotone we mean a mechanism in which if a bidder increases his bid he is not going to receive less items. To calculate the externality of a bidder an explicit formula is used, which can be found in section 3.2. Myerson's Lemma simply states that for each setting (examples will be given below), there is a unique payment that checks our goals (maximum welfare and truthfulness), thus also showing that Vickrey's Auction is the only auction that achieves all these goals.

In order to better understand the multi-unit, single-parameter environments for which Myerson's Lemma is, let us present an auction more complex than that described before: We have a total capacity W and n bidders, each with a known size w_i and a private value v_i. We can think each w_i as the length of a TV ad, while W as the total time we are allowed for advertisements. Our goal is to use the ads whose total size fit our capacity and also achieve maximum welfare over all other feasible solutions. This is basically a knapsack problem, where the values of the items are private. Myerson's Lemma gives an explicit mechanism that guarantees truthfulness and maximum welfare (unfortunately in pseudo-polynomial time).

The last result of this section involves *Sponsored Search Auctions*, where we want to illustrate the usefulness of Myerson's Lemma in the real world. Here our items are slots for ads in a search result page, which people have a probability to click (it is only logical to assume that the higher the slot the higher the probability). Again, each bidder has a private value that represents the profit if his add is clicked. Our goal is to auction off these slots, where the highest slot goes to the bidder with the highest value. Each time a search is performed by a user, an auction takes place in order to be decided what ads are going to appear on each slot. The auction format can only be the one that Myerson's Lemma dictates. It is estimated that sponsored search accounted for 98% of Googles revenue in 2006 [18].

2.2 Multi-Parameter Auctions

In this section we are going to analyze the generalization of Single-Parameter Auctions, *Combinatorial Auctions*. In the previous setting we presented Myerson's Lemma, which in most cases provides a mechanism that checks all our goals (maximum social welfare and truthful bidding). Unfortunately in Combinatorial Auctions mechanisms either are not that simple or they simply do not exist. Sometimes, in order to achieve a good approximation ratio we are required to drop the requirement for truthfulness.

In this setting we generally assume that there is a set M that includes m distinct items, for which each bidder has a valuation that is represented by a set function, which maps each subset of the items to a positive real number, i.e. $v : 2^M \to \mathbb{R}^+$. We always assume that each valuation function is non-decreasing, meaning that the value of a subset does not decrease when more items are added, while also that each valuation function is normalized, meaning that the value for the empty set is always 0.

Because the definition above is too general, in order to hope for better results we need to restrict out valuation functions. We are going to analyze 3 classes of valuation. The first and more strict of the 3 are *submodular* valuations, where for any sets S, T it holds that $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$. The most strict class is the one of *subadditive* valuations, where the inequality constraint is like the one in submodular valuations but more slack: $v(S) + v(T) \geq v(S \cup T)$. The valuation class that lies in-between the two are *XOS* valuations (also known as *fractionally subadditive*), where each function is the maximum of some additive functions, i.e. $v(S) = \max_r \left(\sum_{j \in S} a_{r,j} \right)$.

Previously we stated that in Multi-Parameter Auctions there is not something as good as Myerson's Lemma, that can solve most of our problems. This is not completely true, as there exists a mechanism that works for every auction with general valuations, but has one major downside: It usually runs in $\Omega(2^m)$ time. This mechanism is known as the *VCG Mechanism* and it was introduced by Vickrey [1], Clarke [3] and Groves [4]. It is a generalization of Myerson's Lemma, although this time it is not the only mechanism that meets the desired properties. The key to the VCG mechanism is again on the pricing policies: Each bidder is charged his externality, see section 3.4 for more details.

Someone may have already noticed that even if every bidder was guaranteed to be truthful, simply the input might be too large for us: Each bidder's support for his valuation function is exponential in m. In order to solve this problem, we assume that we can ask bidders queries, which gives us the necessary information for their valuation. Now we are going to see the 2 most common queries, *Value Queries* and *Demand Queries*.

2.2.1 Value Queries

Value Queries are the simplest queries one can think of. A bidder is given a set of items and is asked to answer with his value for said set. Value queries have been used on many mechanisms in order to approximate the optimal value, as it has been proven that even for submodular auctions where the valuations are considered publicly known, finding the optimal allocation is NP-hard [8]. One of the most popular algorithms that 2-approximates the optimal allocation is that of [8], which will also be analyzed here section 6.1.1. This algorithm again assumes that valuations are publicly known. In the same setting is the algorithm of Vondrak [20], providing an approximation ratio of $\frac{e}{e-1}$, which is optimal for the value query model[12].

Regarding mechanisms that use value queries and also "force" bidders to be truthful, the optimal one was made in [21] and provides an approximation bound of $\sqrt{m}$. In 2011 it was shown in [22] that value queries cannot achieve better approximation, even with randomized protocols. Having seen these tight bounds for value queries, we hope that the next type of queries are going to provide better results.

2.2.2 Demand Queries

Demand queries are slightly more complicated than value queries. In a demand query a bidder is presented with a price for each item and is asked to pick the set of items that maximizes his utility. It turns out that demand queries are strictly stronger than value queries, because one can simulate a value queries with polynomial demand queries, but the opposite requires an exponential number of value queries. For this reason we believe that with demand queries we are going to achieve better approximation results.

A known result that shows that demand queries are stronger, is in the case where bidders' valuations are publicly known. The approximation ration of $\frac{e}{e-1}$ that was first proven with demand queries [14], can be decreased to $\frac{e}{e-1} - 10^{-6}$, as shown by Feige and Vondrak [15]. The strength of the demand queries can also be seen by the lower bound for this problem: In [24] it was shown that it is impossible to get a better approximation ratio than $\frac{2e}{2e-1}$ for problems where the valuations are submodular. Let us also add the results of [16], in which a non-truthful algorithm for the more general subadditive valuations was found, with an approximation ration of $\frac{1}{2}$.

Now we are going to see how demand queries affect our ability to make mechanism that force truthfulness. The first non-trivial results where that of [13], in 2006, proving an expected approximation ratio of $O(\log^2 m)$. Later, in 2007 a better ratio of $O(\log m \log\log m)$ was found in [17] and then in 2012 an even better ration of $O(\log m)$ for XOS valuations in [26]. The current state of the art was found in 2016 and is the mechanism of Dobzinski [33], which guarantees an expected cost of $O(\sqrt{\log m})$ for XOS valuations, while also making truth-telling strictly an optimal strategy for each bidder.

All these results have much better approximation ratios than the optimal $O(\sqrt{m})$ ratio that value queries provide, proving how better demand queries are.

2.3 Price of Anarchy in Auctions

Price of Anarchy (POA) is very important nation of Algorithmic Game Theory. It was first used by Papadimitriou in [9] and it shows the inefficiency of equilibria: It is equal to the ratio of the gain/cost of the equilibria with the worst outcome and the the gain/cost of the optimal solution. It has been used in many cases, one of the most known being in congestion games, where in [10] it was shown that probably the simplest non-trivial network has the worst POA.

In our setting we are going to use POA to characterize how "good" a simple auction is, if we leave the bidders bid selfishly. *Simple Auction* usually refers to *Parallel First or Second Price Auctions*, where each bidder submits a bid for each item and each items goes to the bidder with the highest bid, for which he pays either his bid (first price) or the second highest bid (second price).

In general, first price auctions are considered better, because they have higher Price of Anarchy: When the bidders are allowed only pure strategies, the equilibria always achieves maximum welfare, but unfortunately such equilibria doe not always exist. For randomized strategies in [28] it was shown that POA for First Price Auctions is at least $1-\frac{1}{e}$ for XOS valuation, while for subadditive valuations it is at least $\frac{1}{2}$, proven in Feldman et al.

On the other hand Second Price Auctions have worse results: For XOS valuations it was proven in [31] that the POA is at least $\frac{1}{2}$, a result which can easily be verified as tight. For subadditive valuations, in [27] a lower bound of $\frac{1}{4}$ was found for the price of anarchy in Second Price Auctions, which is twice as bad as the one in first Price Auctions.

2.4 Beyond Worst Case Analysis

In this section we are going to do a complete turn and talk about a completely different field: *Beyond Worst Case Analysis*. In this field the goal is to analyze algorithms, usually for hard problems, in a different manner than that of the classical analysis, which is to simply find the worst case input. The reason for this is that we want to uncover some information about the problem/algorithm, which is hidden because of our simplistic worst-case view.

There are many ways to analyze a problem other than the worst case. In this work we will focus more on *Perturbation Stability*, but we are also going to see *Smoothed Analysis*, which is more easily understandable. In *Smoothed Analysis* we look at a hybrid of worst-case and average-case analysis: An adversary picks a "bad" input and then some random process perturbs it a little. What we are interested in is the expected running time of the perturbed input. If this expected running time, for every starting input, is guaranteed to be small, e.g. polynomial, then we can assume that any input that yields large running time, e.g. exponential, is not natural and unlikely to come across.

To illustrate this notion let us present the results of [11], concerning Dantzig's famous simplex algorithm, that has been used since the 1940s to solve Linear Programs, but for which infinitely many examples have been found that run in exponential time. Spielman and Teng [11] showed that given any LP instance for which the input matrixes have been perturbed slightly, the simplex method has polynomial expected running time. One of the corollaries of this is that given a "hard" instance, we can perturb it slightly, thus slightly changing its solution, and solve it with the simplex method in polynomial time. This result reveals the reason why simplex is the dominating algorithm in practice, even though by the classical sense its running time is exponential.

In order to make Smoothed Analysis a bit more understood let us also present the problem of *Local Max Cut*, where given a graph we are required to find a partition of the vertexes in which if we move any vertex to the other part, the total weight of the edges whose vertexes are in different partitions will decrease. To find such a partition, in the worst case we are required to do an exponential number of iterations. However, if we perturb the edges of the graph with random noise, then if the graph has a maximum degree of $O(\log n)$ the expected running time is polynomial [23] and for general graphs the expected running time is quasi-polynomial, i.e. $n^{O(\log n)}$ [35]. Both results look at the problem at a different angle, to show that empirically it is easy to find a Local Max Cut.

Now let us see *Perturbation Stability*. Here we try to focus our analysis only on instances of a problem that matter. To quantify when an instance matters, we usually look at how stable the optimal solution is: If for any small perturbation of the original instance the final solution changes, then the optimal allocation of the original instance is not stable and thus not meaningful. Just like before, let us present a few examples to make Perturbation Stability better understood.

First consider the problem of clustering, where we are given some items and we are required to partition them into a fixed number of clusters, such that items in the same cluster are more similar to each other, than items in other clusters. This problem is again a well-known NP-Hard problem. However in [34] they proved that if a clustering instance is 2-stable, i.e. we can distort the similarity of any pair of elements up to a factor of 2, then we can find the optimal solution in polynomial time. This results shows that it is easy to find partitions when they are distinct enough, which is often the case at real world instances of clustering problems.

Finally, let us present another example for Perturbation Stability, that of *Minimum Multiway Cut*, where given a graph and k vertexes and we are required to partition the graph into k partitions, each containing exactly one of the k vertexes. Even for $k \geq 3$ this is a NP-Hard problem. Makarychev, Makarychev, and Vijayaraghavan in [30] showed that if a Minimum Multiway Cut instance is 4-stable, i.e. by dividing any edge up to a factor of 4 the optimal partition stays the same, we can recover the optimal solution simply be solving a Linear Program. This LP generally does not have an integral solution, meaning that its solution does not directly reveal the optimal solution, but in the 4-stable case it does exactly that: Its solution is integral and it encodes in a natural way the optimal solution.

2.5 Our Contribution

Our aim is to use ideas from Beyond Worst Case Analysis in Mechanism Design, because as we saw in the worst case, most auction problems are hard. Because in combinatorial auctions our aim is to partition the items to the bidders, just like in clustering and minimum cut problems, we are going to use Perturbation Stability in order to characterize some instances as meaningful and other some non-meaningful. In particular, we will call an instance stable when the optimal allocation remains uniquely optimal if we perturb the instance in the following way:

1. Pick one arbitrary bidder i and one arbitrary item j.
2. Increase for bidder i the value of the bundles that contain j proportionally to his value for item j as a singleton.

Thus if we pick a bidder i and an item j such that in the optimal allocation i does not get j, then we weaken the optimal allocation because now bidder i want j more. This notion of stability was inspired by the *Endowment Effect* (see section 5.2). The exact definition (stated formally here section 5.3), is that when the instance is γ-stable, a bidder's valuation changes from $v(\cdot)$ to:

$$v'(S) = v(S) + (\gamma - 1) \cdot v(S \cap \{j\})$$

where S is any subset of the items. Now, when an instance is stable, intuitively, the optimal allocation has to stand out in some way which makes it have the compelling properties required to improve the results above. Specifically, the main property of γ-stable instances is that the marginal value for any item that a bidder receives in the optimal allocation, is greater than $(\gamma - 1)$ times the values of that item as a singleton for any other bidder, i.e. if an item j belongs to O_i, where O_i is the bundle that i receives in the optimal allocation, then for any other bidder k it holds

$$v_i(O_i) - v_i(O_i - j) > (\gamma - 1) \cdot v_k(\{j\})$$

Using this property we have shown a number of interesting results. First we improved the previous bound of $\sqrt{m}$ for submodular mechanisms that are truthful and use value queries. We showed that when the instance is 2-stable and bidders have submodular valuations, then in a parallel Second Price Auction bidding truthfully is always an ex-post Nash Equilibrium, i.e. if all other bidders bid truthfully, then the dominating strategy is to bid truthfully as well.

For 2-stable combinatorial auctions with subadditive valuations we showed a result similar, but weaker, with the previous one: If an instance is 2-stable, then one can find the optimal solution with polynomial many value queries, in polynomial time, but there are no truthful guarantees.

For 2-stable submodular valuations we also showed that there always exists a Walrasian Equilibrium, a well-known notion in economics, which in general is not guaranteed for submodular valuations. This result is similar to that of multiway max cut [30], because in both cases, if the instance is stable enough then each LP, which under normal circumstances has a fractional optimal solution, is integral.

For the setting of Price of Anarchy, in Parallel Second Price Auctions we found that the upper bound of $\frac{1}{2}$ is tight even for $(3-\epsilon)$-stable submodular instances, while for submodular instances that are 3-stable the Price of Anarchy is always 1. In Parallel First Price Auctions, we showed that the POA of Mixed Equilibria is equal to 1 asymptotically for XOS valuations. Specifically, we showed that γ-stable XOS instances have guaranteed POA at least $\frac{\gamma-2}{\gamma-1}$.

2.6 Organization of the Thesis

First in chapter 3 - Basic of Mechanism Design we are going to properly define all the basics of Mechanism Design that one needs to read this thesis. Specifically, we are going to start with Single-Parameter Environments and then we are going to analyze Multi-Parameter Environments. In the latter we are going to see more specific notions than what we saw in the introduction. In particular first we are going to present many more valuation classes, then analyze Linear Programming for Combinatorial Auctions and finally see what a Walrasian Equilibrium is, along with the famous *First* and *Second Welfare Theorems* and their proofs.

In chapter 4 - Beyond Worst Case Analysis we will define and analyze in more detail the different tools of Beyond Worst Case Analysis that we saw earlier: First we are going to see Smoothed Analysis and then Perturbation Stability.

In chapter 5 - Perturbation Stability in Combinatorial Auctions we define exactly what Stability in Combinatorial Auctions is, along with the important lemmas that directly reveal the usefulness of stability in the settings that it is used. Before we do that, at the start of Perturbation Stability in Combinatorial Auctions, we analyze what the *Endowment Effect* is, which is an important scheme that inspired our definition for stable combinatorial auction instances.

In chapter 6 - Mechanisms we analyze the basic mechanisms that exist in each problem for each valuation category, along with our stability-based mechanisms, which provide better guarantees than the previous ones. First we analyze Direct Revelation Mechanisms and then Auctions with demand queries.

Finally in chapter 7 - Price of Anarchy in Auctions we define the different kinds of equilibria and then present all the state of the art theorems for POA for First and Second Price Auctions. After this we present our results for POA for stable combinatorial auctions.

Chapter 3

Basic of Mechanism Design

In this chapter we are going to give some fundamental definitions for the area of *Mechanism Design*. First we are going to talk about general settings. Then we are going to talk more specifically about single-parameter environments, where each bidder has only 1 private piece of information and finally we are going to talk about the much more general and complex setting, multi-parameter auctions where bidders have many private information.

3.1 Preliminaries

Mechanism Design involves a set of players and an auctioneer. We always play the part of the auctioneer, who is trying to allocate some goods to the players. Our goal is to design a mechanism, which gives some kind of guarantee. There are many guarantees, which we will talk about later. In order to achieve these guarantees, we will sometimes ask the players for some kind of payment.

Definition 3.1 (The basic Setup)**.** We assume that we have n bidders and a set of feasible outcomes $\mathcal{O}$. Each bidder has a private valuation $v_i : \mathcal{O} \to \mathbb{R}^+$, which is unknown to us and a set of actions $\mathcal{A}_i$. After collecting a vector of actions $\mathbf{a} = (a_1, ..., a_n)$, where for each i, $a_i \in \mathcal{A}_i$, the mechanism uses a function $f : (\mathcal{A}_1, ..., \mathcal{A}_n) \to \mathcal{O}$, that maps each vector of actions to a feasible outcome. The mechanism also uses a pricing policy, which is a vector of functions $\mathbf{p} = (p_1, ..., p_n)$, where for each i, $p_i : (\mathcal{A}_1, ..., \mathcal{A}_n) \to \mathbb{R}^+$. Finally each bidder i has a utility u_i, which gives the final gain of i and depends on his private valuation, the outcome and the result of the pricing policy. The bidder's goal is to maximize their utility. The pair $(f, \mathbf{p})$ defines a mechanism.

The definition above describes a mechanism as general as possible. We are going to start simplifying the setting. First we are going to restrict ourselves to bidders out have quasi-linear utility functions, meaning their utility is simply their valuation minus the price they have to pay.

Definition 3.2 (Quali-Linear Utility Function)**.** Let $v_i : \mathcal{O} \to \mathbb{R}^+$ be a valuation of a bidder i, $(f, \mathbf{p})$ a mechanism and $\mathbf{a}$ the vector of the actions of the bidders. We will say that bidder i has *quasi-linear utility*, if

$$u_i = v_i(f(\mathbf{a})) - p_i(\mathbf{a})$$

From now on we are going to only talk about bidders who have quasi linear utility functions. Another simplification that often happens is that bidders do not report actions, something which is very general, but bids. A bid is what a bidder claims his private valuations to be. A mechanism that ask from each bidder a bid, is called direct revelation mechanism, because each player is asked to directly reveal his valuation.

Definition 3.3 (Direct Revelation Mechanism)**.** A mechanism $(f, \mathbf{p})$ is called a *Direct Revelation Mechanism*, if each bidder's action is to report a bid b_i, where $b_i = \mathcal{O} \rightarrow \mathbb{R}^+$.

In this chapter we are going to assume that each mechanism is a direct revelation mechanism, unless stated otherwise.

Now we are going to talk about one of the first goals of our mechanisms: To predict what kind of strategies the players should play. First we will define what a dominant strategy is. A dominant strategy is simply a bid (or more generally an action) that maximizes the utility of a player, regardless of the bids of the other players.

Definition 3.4 (Dominant Strategy)**.** Let $(f, \mathbf{p})$ be a Direct Revelation Mechanism. For a player i, a bid b_i is called a *Dominant Strategy*, if for any other bid b_i' and bids $\mathbf{b}_{-i}$

$$v_i(f(b_i, \mathbf{b}_{-i})) - p_i(b_i, \mathbf{b}_{-i}) \geq v_i(f(b_i', \mathbf{b}_{-i})) - p_i(b_i', \mathbf{b}_{-i})$$

One of our goals is for our mechanism to be truthful. A mechanism is called truthful if for each bidder his dominant strategy is to bid truthfully, i.e. report his true valuation.

Definition 3.5 (Truthful Mechanism)**.** A mechanism $(f, \mathbf{p})$ is called *Truthful*, if for every bidder i and any valuation function v_i, reporting as bid v_i is a dominant strategy, i.e. for every bid b_i and every vector of bids $\mathbf{b}_{-i}$

$$v_i(f(v_i, \mathbf{b}_{-i})) - p_i(v_i, \mathbf{b}_{-i}) \geq v_i(f(b_i, \mathbf{b}_{-i})) - p_i(b_i, \mathbf{b}_{-i})$$

If a mechanism is truthful, each bidder has nothing to lose by reporting his true valuation, thus helping us, the auctioneer, learn the whole picture in order to fulfill our goals. For completeness, let us define truthful mechanisms, in a randomized context where we use a set of deterministic mechanisms.

Definition 3.6 (Universal Truthful)**.** A randomized mechanism $\{f^k, \mathbf{p}^k\}_{k=1,\ldots}$, which is a set of deterministic mechanisms $(f^1, \mathbf{p}^1), \ldots$, is called *Universally Truthful* if every one of his deterministic mechanisms is truthful.

Another goal of our mechanism is that every bidder should not regret telling the truth. This means that if he reports his true valuations, his utility should not be negative. This property is called individually rationality.

Definition 3.7 (Individually Rationality)**.** A mechanism $(f, \mathbf{p})$ is *Individually Rational* if for every bidder i, every valuation v_i and any vector of bids $\mathbf{b}_{-i}$

$$v_i(f(v_i, \mathbf{b}_{-i})) - p_i(f(v_i, \mathbf{b}_{-i})) \geq 0$$

Definition 3.8 (DSIC)**.** A mechanism $(f, \mathbf{p})$ is called *DSIC*, if it is both truthful and individually rational.

Mechanisms that are DSIC are (part of) the holy grail: Every bidder always wants to tell the truth, no matter matter what the other bidders are doing and also has no incentive not to participate, since his utility cannot be negative.

Up until now our only goal is DSIC. This can be trivially achieved, if our mechanism outputs a fixed outcome and 0 payments. As one might have guessed this has no interest, meaning we should raise our bar a little. What follows is some quantities that are of great interest for us, the auctioneer.

Definition 3.9 (Social Welfare)**.** The *Social Welfare* of an outcome $\omega \in \mathcal{O}$ is defined as

$$\text{SW} = \sum_{i=1}^{n} v_i(\omega)$$

The Social Welfare is what is best for the bidders as a whole, because it maximizes their total valuations.

Definition 3.10 (Revenue)**.** The *Revenue* of a mechanism $(f, \mathbf{p})$, for a given vector of bids $\mathbf{b}$ is defined as

$$\text{Rev} = \sum_{i=1}^{n} p_i(\mathbf{b})$$

The Revenue is the total of the payments of all the bidders and generally has the purpose to please the auctioneer. We will not go into more detail about the revenue of an auction, as from now on our goal is going to maximize the social welfare.

In order to summarize this chapter, our goals (plus a new one) are the following: Create mechanisms that maximizes the social welfare, while being DSIC, all in polynomial time. This means that the mechanism is going to force each bidder to bid his true valuation (because this will be his best action), while at the same time ensuring that the output outcome maximizes the sum of the bidders' values, while calculating it in polynomial time.

3.2 Single-Parameter Environments

In this section we are going to talk about the simpler setting, where each bidder's private valuation is not as complicated as a function, but can simply be represented by a single real number which is called his value, denoted $v_i \in \mathbb{R}^+$. In this setting we are first going to look at the simplest auction, where we simply have one item, and we want to allocate it to the bidder with the highest valuation. The first solution that comes to mind seems to be to allocate it to the bidder with the highest bid and have him pay his bid. This is a first price auction and it simply does not work. The reason is because it cannot possibly be DSIC, as the final utility is always 0 if a bidder bids truthfully. The best (and as will be stated later) only solution is a Vickrey auction.

Definition 3.11 (Vickrey or Second Price Auction)**.** A *Vickrey* or *Second Price Auction* is an auction where every bidder reports a bid and the item goes to the player with the highest bid. Each player pays 0, except for the winning player who pays the second highest bid.

This auction may seem a little a bit weird, but it can be proven that it checks all our desirable goals.

Theorem 3.1 ([1]). *For any vector of valuations* $\mathbf{v} = (v_1, ..., v_n)$*, the Vickrey auction has the following 3 properties*

1. *It can be calculated in polynomial time.*
2. *It is DSIC.*
3. *If every bidder bids truthfully, it maximizes social welfare.*

The first and third properties are easy enough to understand. The second, although more complex, also seems logical: The winning bidder, since his valuation is the highest, is guaranteed to pay less than his valuation since he will pay the second highest bid. The other bidders, in order to win would have to bid higher than their valuation, thus achieving negative utility if doing so.

Proof. The first property is trivial since both the allocation rule and the pricing policy are simple find-the-maximum-element problems. The third is also trivial, since if we are reported the true valuations, allocating the item to the bidder with the highest valuation is indeed the option that maximizes social welfare.
For the second property, fix a bidder i, his valuation v_i and consider any bid vector $\mathbf{b}_{-i}$ played by the other players. Denote with $B = \max(\mathbf{b}_{-i})$. Consider 2 cases: $v_i > B$ or $v_i \leq B$. All we have to do is show that in both cases bidder i has nothing to lose by bidding v_i.
In the first case, by bidding v_i, his utility is $v_i - B > 0$. By bidding higher he is still guaranteed the same utility, but by bidding lower, his bid will either stay above B, keeping the utility the same, or will drop at B or below, thus risking the utility to drop to 0. This completes the proof for the first case.
In the second case, by bidding v_i his utility will always be 0. By bidding lower, his utility does not increase, while by bidding higher he is in danger of bidding above B. This will lead to utility $v_i - B$ which is non-positive. This completes the proof for the second case. ■

Now we are going to present a simplified version of a very important theorem, proven in [6], known as Myerson's Lemma.

Theorem 3.2 (Myerson's Lemma [6]). *A mechanism* $(\mathbf{f}, \mathbf{p})$*, where* $\mathbf{f}$ *is a vector of allocation functions, is DSIC if and only if*

- *The function* $f_i(\cdot, \mathbf{b}_{-i})$ *is non-decreasing for every bidder* i *and bid vector* $\mathbf{b}_{-i}$*.*
- *Given a bid vector* $\mathbf{b}$ *the payment rule for bidder* i *is given by the following formula*

$$p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} z \cdot \frac{d}{dz} f_i(z, \mathbf{b}_{-i}) dz \tag{3.1}$$

Myerson's lemma can be summarized in the following sentence: Given that we want to make a DSIC mechanism, all we have to do is make sure that we do not allocate a bidder less if he bids more, and that the payments are given by that function in eq. (3.1). To understand the payment function, let us present how it works in the Vickrey auction.

In the Vickrey auction, the allocation function for a certain bidder i is a step function: $f_i(b_i, \mathbf{b}_{-i}) = step(b_i - \max(\mathbf{b}_{-i}))$, where $f_i(\cdot) = 0$ denotes that bidder i does not get allocated anything, while $f_i(\cdot) = 1$ means that i is allocated the good. The derivative of f_i is a delta function: $\frac{d}{dz} f_i(z, \mathbf{b}_{-i}) = \delta(z - \max(\mathbf{b}_{-i}))$. This means that the payment rule is

$$p_i^{Vickrey}(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} z \cdot \delta\big(z - \max(\mathbf{b}_{-i})\big) dz = \max(\mathbf{b}_{-i}) \cdot \mathbb{1}[b_i \geq \max(\mathbf{b}_{-i})]$$

Given Myerson's Lemma we can easily create DSIC mechanisms for single-parameter environments. Another example of its application is a generalization of the previous setting, where we have k identical items and we want to allocate them to the k bidders who have the highest valuation. Myerson's Lemma says to allocate them to the k bidders who have the highest bids and have them pay the $(k+1)$-highest bid, which is a generalization of the Vickrey auction.

3.3 Multi-Parameter Environments

In the previous section each bidder had a single parameter, which characterized his valuation. However what happens when there are multiple distinct items sold? A simple answer is that each bidder has a value for each item and when he gets multiple items then his total value is simply the sum of the value of each item. This turns out to simplify reality too much. Consider a player Bob, who wants to buy a house. Lets say that there are 2 identical houses, each of a value of 1 to Bob. However, giving Bob both houses does not necessarily mean that his total valuation will be 2, simply because Bob might not have use for the second house. This example shows that we need to create a more complex model to describe environments like the one above.

Definition 3.12 (Combinatorial Auction)**.** A c*Combinatorial Auction* is described by a triplet $(N, M, \mathbf{v})$. N is a set of bidders with cardinality $|N| = n$, M is a set of items with cardinality $|M| = m$ and $\mathbf{v}$ is a vector of n set functions, $\mathbf{v} = (v_1, ..., v_n)$, each from a subset of M to a non-negative real number, i.e. for each $i \in N$, $v_i : 2^M \mapsto \mathbb{R}^+$. Each function v_i is assumed to be normalized to 0, i.e. $v_i(\emptyset) = 0$ and non-decreasing, i.e. if $S \subseteq T \subseteq M$ then $v_i(S) \leq v_i(T)$.

In this setting let us define what a feasible allocation is.

Definition 3.13 (Feasible Allocation)**.** In an auction $(N, M, \mathbf{v})$, a *Feasible Allocation* $(S_1, ..., S_n)$ is a vector of n disjoint sets, i.e. $\forall i \neq k,\ S_i \cap S_k = \emptyset$.

What we care about is the optimal allocation which maximizes $\sum_{i \in N} v_i(S_i)$. From now on an optimal allocation is going to be symbolized with the vector $\mathbf{O}$, whose elements are going to be $(O_1, ..., O_n)$.

3.3.1 Valuation Classes

In definition 3.12, each valuation needed only be normalized and non-decreasing. It turns out that when the valuations are that general, finding an efficient allocation can be quite hard. To solve this problem we define classes of valuations. Let v be a valuation function and M the set of items.

Definition 3.14 (Additive Function)**.** A valuation function $v : 2^M \mapsto \mathbb{R}^+$ is *additive* if for every $S \subseteq M$

$$v(S) = \sum_{j \in S} v(\{j\})$$

where for every $j \in M$, $v(\{j\}) \geq 0$.

This is one of the least general classes of valuations, as items are viewed independently.

Definition 3.15 (Unit-Demand Function)**.** A valuation function $v : 2^M \mapsto \mathbb{R}^+$ is called *unit-demand* if for every $S \subseteq M$

$$v(S) = \max_{j \in S} v(\{j\})$$

where for every $j \in M$, $v(\{j\}) \geq 0$.

This is also one of the least general classes, because, just like additive valuations, each bundle of items is easily calculable from the values of each singleton.

Definition 3.16 (Submodular Function)**.** A valuation function $v : 2^M \mapsto \mathbb{R}^+$ is called *submodular* if for every $S, T \subseteq M$, with $S \subseteq T$ and any item $j \notin S$

$$v(S \cup \{j\}) - v(S) \geq v(T \cup \{j\}) - v(T)$$

Submodular functions include both additive and unit-demand functions. They are the analogous to concave function and they represent a form of diminishing returns: the larger the set, the less is the additional value when a new item is added. Submodular functions can sometimes be characterized as general enough, as they capture various general settings.

Definition 3.17 (XOS Function)**.** A valuation function $v : 2^M \mapsto \mathbb{R}^+$ is called *XOS* if there exist a set $A = \{a_1(\cdot), a_2(\cdot), ...\}$ of additive valuation functions, such that for every $S \subseteq M$

$$v(S) = \max_{a \in A} a(S) = \max_{a \in A} \sum_{j \in S} a(\{j\})$$

XOS functions are kind of hard to understand, but are the class that is slightly more general that submodular and because of their definition they are sometimes easier to work with, compared to submodular valuations.

Definition 3.18 (Subadditive Function)**.** A valuation function $v : 2^M \mapsto \mathbb{R}^+$ is called *Subadditive* if for every $S, T \subseteq M$

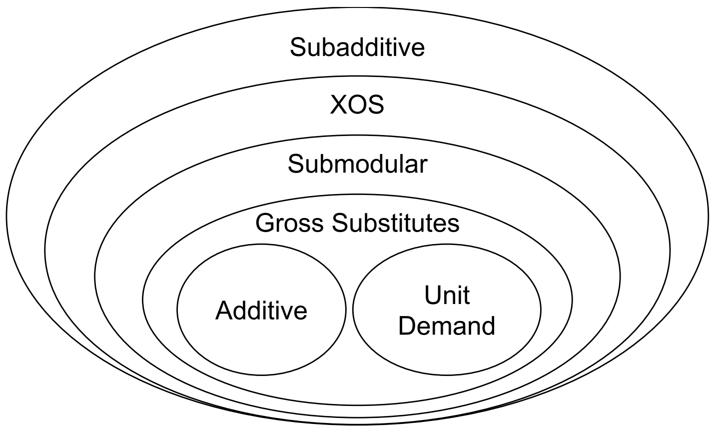
$$v(T) + v(S) \geq v(T \cup S)$$

Subadditive functions are so general that in most cases they are the most general class used. They include XOS functions and are sometimes called *complement-free*, because they capture the notion that a combination of 2 bundles does not increase the value of each bundle. As an example of non-subadditivity, a nail and a hammer independently might have little value, but together their value is greater than the sum of the two independent values.

To sum up the relation of the aforementioned classes

$$\textit{Additive, Unit-Demand} \subset \textit{Gross-Subtitutes} \subset \textit{Submodular} \subset \textit{XOS} \subset \textit{Subadditive}$$

The class of Gross-Subtitutes mentioned above is also a very important class, but because the definition is more technical, it can be found in . Below is the above relation in a diagram.



3.3.2 Combinatorial Auctions and Linear Programming

As many other combinatorial problems, maximizing the total welfare can be formulated as an Integer Linear Program. We present the Linear Program Relaxation, where each variable, instead of taking a value from $\{0, 1\}$, takes a value in $[0, 1]$.

Definition 3.19 (The Linear Programming Relaxation (LPR))**.** Let $(N, M, \mathbf{v})$ be a combinatorial auction. The *Linear Programming Relaxation* is

$$\begin{aligned}
\textbf{Maximize} \quad & \sum_{i=1}^{n} \sum_{S \subseteq M} x_{iS} \cdot v_i(S) & \\
\textbf{s.t.} \quad & \sum_{i=1}^{n} \sum_{S | j \in S} x_{iS} \leq 1 & \forall j \in M \\
& \sum_{S \subseteq M} x_{iS} \leq 1 & \forall i \in N \\
& x_{iS} \geq 0 & \forall i \in N,\ S \subseteq M
\end{aligned}$$

The LP's variables x_{iS}, take values from 0 to 1. $x_{iS} = 1$ denotes that bidder i receives bundle S in the final allocation. The objective function is maximizing the social welfare, the first set of inequalities ensure that each item is allocated at most 1 time and the second set of inequalities is to ensure that each bidder gets at most 1 bundle. Note that, in terms of n and m, the LRP has exponentially many variables, but linearly many constraints. The dual of the above LP is the following.

Definition 3.20 (The Dual Linear Programming Relaxation (DLPR))**.** Let $(N, M, \mathbf{v})$ be a combinatorial auction. The *Dual Linear Programming Relaxation* is

$$\begin{aligned} \textbf{Minimize} \quad & \sum_{i=1}^{n} u_i + \sum_{j \in M} p_j & \\ \textbf{s.t.} \quad & u_i + \sum_{j \in S} p_j \geq v_i(S) & \forall i \in N,\ S \subseteq M \\ & u_i \geq 0,\ p_j \geq 0 & \forall i \in N,\ j \in M \end{aligned}$$

The usage of the variables u_i and p_j is intentional, as we will see later that they can be interpreted as utilities and prices, respectively.

We would like to now provide an example, to show that the solution to the LRP can be fractional.

Example 3.1 (LRP can be fractional)**.** Consider two players Alice and Bob and two items a and b. Alice has a value of 2 for each non-empty set and Bob only cares for the whole bundle ab, for which he has a value of 3. The optimal allocation allocates both items to Bob for a welfare of 3. However, the optimal fractional solution has welfare equal to 3.5: Allocate half the bundle ab to Bob for a value of 1.5. This means that Alice can have half the bundle $\{a\}$ and half the bundle $\{b\}$, for a welfare of 2. To be more clear we provide the positive variables of the optimal allocation, where bidders 1 and 2 are Alice and Bob, respectively: $x_{1\,a} = 0.5$, $x_{1\,b} = 0.5$, $x_{2\,ab} = 0.5$. This means that the fractional solution can have higher welfare than the integral one.

3.3.3 Walrasian Equilibrium

A natural mechanism that one may come up with, is the following: Pick a price for each item and the let each bidder take what he prefers. The idea behind this simple mechanism is simple. If the prices are picked right and are not too high, then each item will be picked by someone who has high value for it, and therefore we will achieve high total welfare. In addition, it would be best that no matter the order in which the bidders pick items, each bidder would always pick the same bundle. This idea can easily be formulated. Firstly we need to define what is the best preference of a bidder.

Definition 3.21 (Demand of a bidder)**.** Let i be a bidder, $v_i(\cdot)$ his valuation and $p_1, ..., p_m$ prices for each item. A bundle $S \subseteq M$ is called a *Demand* of bidder i, if for every bundle $T \subseteq M$, bidder i does not gain utility by picking T, i.e.

$$v_i(S) - \sum_{j \in S} p_j \geq v_i(T) - \sum_{j \in T} p_j.$$

Now that we have defined what each bidder prefers we are ready to define these market-clearing prices.

Definition 3.22 (Walrasian Equilibrium)**.** Let $(N, M, \mathbf{v})$ be a combinatorial auction. A vector of non-negative prices $p_1^*, ..., p_m^*$ and an allocation $S_1^*, ..., S_n^*$ is called Walrasian Equilibrium, if for every player i, S_i^* is his demand at prices $p_1^*, ..., p_m^*$, and for any item j not allocated ($j \notin \cup_i S_i^*$), it holds that $p_j^* = 0$.

Setting walrasian equilibrium prices for the items and simply letting the bidders take what they demand, seems like a very simple and truthful mechanism. Unfortunately, a walrasian equilibrium does not always exist, as we demonstrate with the following example.

Example 3.2 (Non-Existance of Walrasian Equilibrium)**.** Consider the same setting as in example 3.1. Consider two cases.
First case is Bob's demand does not contain the empty set. This means that he demands the whole bundle and Alice demands the empty set. For him to demand the whole bundle, means that the sum of the two prices is at most 3, which means that at least one item has price at most 1.5. But if an item has price less than 1.5, Alice would demand that item and not the empty set, which leads to a contradiction.
Second case is that Bob's demand has the empty set, which means that the sum of the prices is greater than 3. This implies that Alice cannot demand the bundle ab as her value for it is 2. This means that Alice demand is one item, w.l.o.g. item a, for a price at most 2. Item b is now left out of any demand, which means that since we have a walrasian equilibrium, its price is 0. This contradicts with the fact that the sum of the two prices is at least 3.

The fact that we used the same example twice in a row is not a coincidence. It turns out that existence of a walrasian equilibrium and the integrality of the LRP are one and the same. We see that through two of the most fundamental theorems of Mechanism Design.

The first is the *First Welfare Theorem*. It states that an allocation of an walrasian equilibrium maximizes the social welfare, even over fractional solutions of the LRP.

Theorem 3.3 (The First Welfare Theorem [19])**.** *Let $(N, M, \mathbf{v})$ be a combinatorial auction and $(p_1^*, ..., p_m^*)$ and $(S_1^*, ..., S_m^*)$ a Walrasian Equilibrium. Then $(S_1^*, ..., S_m^*)$ maximizes welfare over all fractional solutions i.e. $\sum_{i=i}^n v_i(S_i^*) \geq \sum_{i=i}^n \sum_{S \subseteq M} x_{is} \cdot v_i(S)$ for any feasible fractional solution $\{x_{is}\}_{i,S}$.*

Proof. Fix a feasible solution to the LRP $\{x_{is}\}_{i,S}$, a bidder i and a bundle $S \subseteq M$. Because we have a walrasian equilibrium, we know that

$$v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq v_i(S) - \sum_{j \in S} p_j^* \tag{3.2}$$

By multiplying inequality 3.2 with x_{iS} and summing over all i and $S \subseteq M$ we get

$$\sum_{i=1}^{n} \sum_{S \subseteq M} x_{iS} \cdot \big(v_i(S_i^*) - \sum_{j \in S_i^*} p_j^*\big) \geq \sum_{i=1}^{n} \sum_{S \subseteq M} x_{iS} \cdot \big(v_i(S) - \sum_{j \in S} p_j^*\big) \tag{3.3}$$

Since the solution is feasible, using the fact that $1 \geq \sum_{S \subseteq M} x_{is}$ for all i in inequality 3.3 we get

$$\sum_{i} v_i(S_i^*) - \sum_{i} \sum_{j \in S_i^*} p_j^* \geq \sum_{i,S} x_{iS}\, v_i(S) - \sum_{i,S} x_{iS} \sum_{j \in S} p_j^* \tag{3.4}$$

To prove the theorem we now simply need to prove that $\sum_i \sum_{j \in S_i^*} p_j^* \geq \sum_{i,S} x_{iS} \sum_{j \in S} p_j^*$. Notice that the LHS equals $\sum_{j \in M} p_j^*$, since every item appears at most once and for items j that $j \notin \cup_i S_i^*$ it holds that $p_j^* = 0$. In the RHS, each p_j^* is multiplied by $\sum_i \sum_{S|j \in S} x_{iS}$, which is at most 1, because of the first set of inequalities in the LRP. This concludes the proof. ■

Now we know that if an auction has a walrasian equilibrium, then posting the right price to each item gives a truthful mechanism, that also finds the optimal allocation. However it is time to wonder how we can find if a walrasian equilibrium exists and if so, how to calculate the prices. We only know that if a walrasian equilibrium exists, then the LRP is integral. However the opposite is also true: A walrasian equilibrium exists if and only if the LRP is integral. This is formulated by the *Second Welfare Theorem*.

Theorem 3.4 (The Second Welfare Theorem [19]). *Let $(N, M, \mathbf{v})$ be a combinatorial auction. If the LRP of $(N, M, \mathbf{v})$ has an integral optimal solution, then a walrasian equilibrium also exists.*

Proof. Let $O_1, ..., O_n$ be the allocation that maximizes social welfare over all fractional allocations. Also consider the optimal solution to the DLRP, $p_1^*, ..., p_m^*$ and $u_1^*, ..., u_n^*$. We will show that $O_1, ..., O_n$ together with $p_1^*, ..., p_m^*$ form a Walrasian Equilibrium.
Fix a bidder i. Because of the complementary slackness conditions and $x_{iO_i} = 1 > 0$, the second constraint in the DLRP must hold with an equality

$$u_i^* = v_i(O_i) - \sum_{j \in O_i} p_j^*$$

Substituting u_i^* in the second constraint of the DLPR, for any $S \subseteq M$ we get

$$v_i(O_i) - \sum_{j \in O_i} p_j^* \geq v_i(S) - \sum_{j \in S} p_j^* \tag{3.5}$$

Because of inequality 3.5 we have completed half the proof of the existence of a Walrasian Equilibrium. Now we need to show that if an item is unsold then its price is 0. This also comes from the complementary slackness conditions: In the LRP, for $j \in M$ such that the first constraint is not strict (here meaning that j is not allocated) we have that $p_j^* = 0$. Thus we have completed the proof. ■

3.4 VCG Mechanism

In this section we are going to talk about the VCG mechanism, which belongs more in section 3.1, but is presented here because it is too complex to have been presented before. The VCG mechanism is the generalization to Myerson's lemma, for any setting. It provides a mechanism that is both DSIC and maximizes social welfare over all feasible outcomes. Unfortunately, just like Myerson's Lemma it does not guarantee polynomial running time.

Theorem 3.5 (Vickrey-Clarke-Groves (VCG) Mechanism [1, 3, 4]). *For any setting, no matter how general it is, there is always a DSIC direct-revelation welfare-maximizing mechanism.*

Now we are going to analyze the mechanism itself. The allocation rule is simple: Given that everyone is telling the truth, pick the outcome that maximizes the bids that the bidders made

$$f^{VCG}(\mathbf{b}) = \underset{\omega \in \mathcal{O}}{argmax} \sum_{i=1}^{n} b_i(\omega)$$

This completes the proof of welfare maximization, since we assume that the bidders are truthful. Now we need to give payments that make truth telling a dominant strategy. Denote with ω^* the outcome of the VCG mechanism. The payment policy is the following

$$p_i^{VCG}(\mathbf{b}) = \max_{\omega \in \mathcal{O}} \sum_{\substack{k=1 \\ k \neq i}}^{n} b_k(\omega) - \sum_{\substack{k=1 \\ k \neq i}}^{n} b_k(\omega^*)$$

This means that each bidder is asked to pay his externality, since the first sum is how much the maximum welfare would have been without i and the second sum represents how much the social welfare is now, without calculating i. Notice that the payment is always non-negative. In order for this to make more sense, let us look at the utility of bidder i, if the bids of the other players are $\mathbf{b}_{-i}$

$$u_i^{VCG}(\mathbf{b}) = v_i(\omega^*) + \sum_{\substack{k=1 \\ k \neq i}}^{n} b_k(\omega^*) - \max_{\omega \in \mathcal{O}} \sum_{\substack{k=1 \\ k \neq i}}^{n} b_k(\omega)$$

Let us denote with $B_{-i}(\omega) = \sum_{k \neq i} b_k(\omega)$. The only thing that i can affect with his bid in his utility is the outcome ω^*, meaning that we can ignore the third term, since it is independent of i's bid. Let us denote with ω^* the outcome if i is truthful and with ω' the outcome if i bids v_i'. Then, always ignoring the third term, i's utility if he bids truthfully is

$$u_i^* = v_i(\omega^*) + B_{-i}(\omega^*)$$

Because VCG picks ω^* in order to maximize the exact quantity above we have that

$$u_i^* = v_i(\omega^*) + B_{-i}(\omega^*) \geq v_i(\omega') + B_{-i}(\omega')$$

We notice that latter term is i's utility if he bids v_i', for any v_i'. This concludes the proof because we showed that truthful bidding dominates every other strategy.

Although the VCG mechanism is pretty awesome it lacks a very important property: Guarantee that it can be calculated in polynomial time. Specifically, for unit-demand, additive and Gross Substitutes valuations, one can calculate the VCG payments in polynomial time, but for submodular valuations, it is NP-Hard to do so.

Chapter 4

Beyond Worst Case Analysis

4.1 Introduction

When creating an algorithm for a certain problem, our goal is to make that algorithm good. But what does "good" mean? The first thought that comes to mind is that the complexity of our algorithm should always be at most some function of the size of the input, in the best case polynomial. However this is not always what good means.

The most well-known such case is the problem of Linear Programming or LP, where the goal is to minimize a linear objective function, subject to some linear constraints. It has been proven that any LP problem can be solved in polynomial time in the number of variables and constraints, for example with the ellipsoid method. Yet, the algorithm that is most commonly used in practice to solve LP problems, is the simplex method, developed by Dantzig in the 1940s. The problem with this algorithm is the fact that for very specific instances, it can take exponential time to find the solution, as shown by Klee and Minty in 1970[2].

So why do we use the simplex algorithm, when there are plain better algorithms? The reason is because the term "better" applies only to the worst case instances of each algorithm, which are very hard to come across, making the simplex algorithm far better in practice. However, the last sentence is still vague. In the next sections we are going to define what "in practice" or "better" means in a formal sense.

4.2 Smoothed Analysis - Simplex Method

In this section we are going to analyze the example that we mentioned previously: How and why is the simplex method good in practice. First lets us explain what *Smoothed Analysis* is. In Smooth Analysis we think of an adversary, who comes up with an instance of the problem I. Then, that instance is perturbed slightly by a random process and becomes $I' = I + N$, when N can be thought as some form of noise with mean 0. Then the *Smoothed Complexity* of an algorithm is the maximum expected time that takes to solve instance I', over all instances I. The definition of Smoothed Complexity is trying to capture that when solving a problem, it is very unlikely to have a specific adversarial worst case instance, but rather some instance that is close to that.

Now let us now remind that the input to a linear program are a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and 2 vectors $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{c} \in \mathbb{R}^d$, where we are required to either maximize or minimize the function $\mathbf{c}^\top \cdot \mathbf{x}$, subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. In [11], Spielman and Teng showed that bad examples for the simplex method are basically non-existent.

The perturbation that they proposed is the following: Independently for each entry of $\mathbf{A}$ and $\mathbf{b}$, add a Gaussian random variable with mean 0 and standard deviation σ. When $\sigma = 0$ then the instance remains unchanged and we are doing a pure worst case analysis, while when $\sigma \to \infty$, the input is uniformly random and we have a pure average case analysis. Spielman and Teng showed that the expected running time of the simplex method is at most polynomial, meaning that bad examples are very rare.

Theorem 4.1. *[11] For every initial linear program, in expectation over the perturbation to the program, the running time of the simplex method is polynomial in the input size and* $\frac{1}{\sigma}$.

This theorem basically tells us that if we make a small perturbation to our initial input, then the simplex algorithm is going to find a solution near the optimal (since the objective function is unchanged) in polynomial time. This shows why the simplex algorithm in practice is not considered an exponential algorithm: The chances that the running time is going to be exponential are very small.

Another similar but simpler result, that also shows how fast the simplex algorithm is, is that of [38]. Dadush and Huiberts show that when the elements of the input are bounded and the simplex method uses the shadow vertex pivot rule, then the expected running time of the input perturbed by Gaussian Random Noise with deviation σ is at most

$$O(d^2\sigma^{-2}\sqrt{\log n} + d^3 \log^{3/2} n)$$

These results are a first example of how Beyond Worst Case Analysis helps to analyze algorithms in a more detailed manner than that of the worst case. Next we will see a diffent way to analyze algorithms.

4.3 Perturbation Stability

In this section we are going to talk about *Perturbation Stability*. We are going to go into more detail than the previous one, because this is the type of analysis we are going to use on mechanism design. Perturbation Stability, in our settings, captures the following property: when trying to assign items into certain categories, an instance where miss-assigning an item has impact on the optimal solution, does not have much meaning. Another way to understand this, is that we would like the optimal solution to be meaningful and far away from suboptimal solutions. This will become more specific and understood as we analyze examples of known problems where this has been used.

4.3.1 Stable Clustering

Clustering in general is the problem where we are trying to partition points of a space into categories that are similar to each other. In particular, we are going to study the

k-median problem. In this problem we are given a set of points X and a distance function between the points $d(\cdot,\cdot)$, which is symmetric and satisfies the triangular inequality. The objective is to find k points from X, called centers, such that each center is a close representative of a distinct subset of the points from X, i.e. we are trying to find points $c_1, ..., c_k \in X$, such that, if $C_i = \{x \in X : i = \arg\min_k d(x, c_k)\}$, then the following function is minimized

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

A reasonable assumption here is to assume that if a clustering is not clearly optimal, then some points are not clearly assigned to a certain cluster, thus making the choice of their assignment not important. This, in order makes the instance not meaningful. First we need a way to generate instances that are close to ours, in order to compare their optimal solutions.

Definition 4.1 (γ-Perturbation for Clustering Problems)**.** For $\gamma \geq 1$, a *γ-perturbation* of an instance (X, d) is another instance (X, d') with the same set of points and a new distance function such that for any 2 points $x, y \in X$

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

Thus by perturbing an instance we get another instance where each pair of items are at most γ times closer to each other. Now we are ready to define what makes an instance have a clearly optimal solution, i.e. makes it *stable*.

Definition 4.2 (Perturbation Stability for Clustering Problems)**.** A k-median instance is called *γ-perturbation-stable*, or simply *γ-stable* if its optimal clusters $C_1^*, ..., C_k^*$ remains uniquely optimal in any γ-perturbation of (X, d).

Now what we want is that if an instance is stable enough, then it should be easy to recover its meaningful solution, since it is far away from other solutions. That is exactly what Angelidakis, Makarychev, and Makarychev proved, by constructing a new algorithm called *single-link++*. This algorithm, instead of finding the optimal clustering in the original graph, which is complete, calculates the optimal solution in the Minimum Spanning Tree of the original graph, where it can be done in polynomial time.

Theorem 4.2 ([34])**.** *In every* 2*-perturbation-stable k-median instance, the single-link++ algorithm recovers the optimal solution in polynomial time.*

This theorem seems to make the best out of a bad situation: Even though k-median is a NP-Hard problem, when the solution is meaningful, then we can find it easily.

4.3.2 Stable Minimum Cut

This time we are going to look at the *Minimum Cut* problem. As it is probably known, in Minimum Cut we are given a graph and we are asked to partition its vertexes into 2 sets in order to minimize the sum of the weights of the edges that have vertexes in both sets. However it is also known that minimum cut is not a hard problem, as we can solve it in polynomial time. What is of interest to us is the linear program that encodes an instance of minimum cut.

Solving a problem via its linear program relaxation (where we get a solution that is not necessarily discrete as we would want it to be) and then doing some kind of rounding is a very popular way to solve or even approximate hard problems. But again in minimum cut, this is easy. One can show that if the optimal solution to a minimum cut instance is unique, then the LP relaxation has an integral solution. This means that minimum cut can also easily be solved by a LP.

Thus we have established that minimum cut problem is easy. However, its generalization *Minimum Multiway Cut* is not. Here we are given a graph and k vertexes called terminals. We must partition the vertexes into k sets, where each set contains exactly 1 terminal, so that the sum of the weights of the edges whose vertexes belong to different sets is minimized. It turns out that this problem is much harder. To be precise, even when $k = 3$, the problem is NP-Hard. This also means that we should not hope that by solving the LP relaxation of the multiway cut problem we can guarantee that we will find the optimal integral solution.

Now that we have found a difficult problem we would like to apply the same setting that we applied for the k-medians problem. As before, we can contemplate that if the optimal solution to the Minimum Multiway Cut problem is not clearly optimal (and thus not stable), then the instance is not meaningful. So let us define what a stable instance of multiway cut is.

Definition 4.3 (Perturbation Stability for Graph Problems)**.** For $\gamma \geq 1$, an instance of multiway cut is γ-stable if its optimal partition of the vertexes is the unique optimal partition to any γ-perturbation, which is the original problem, but the weight of each edge w_e is replaced by edge weight $w'_e \in [\frac{1}{\gamma} w_e, w_e]$.

This definition basically says that an instance is stable if the optimal solution does not change when the weights have changed a little. Calling stable instances meaningful makes sense, since it is only natural for the weights of the edges to not be properly calculated and contain some error in them. Now we can show the main theorem, shown by Makarychev, Makarychev, and Vijayaraghavan, which states that if a multiway cut instance is stable enough then finding its solution is as easy as solving its LP relaxation problem.

Theorem 4.3 ([30])**.** *For every 4-perturbation stable multiway way cut instance, the optimal solution to the linear program relaxation of the multiway cut, is integral, meaning that the optimal solution can be found in polynomial time.*

As in the k-medians problem, we have that if the instance is clearly optimal, then solving the multiway cut problem is easy: all we have to do is use our favorite Linear Programming solver.

4.4 Conclusion

In this chapter we saw problems where in the worst case we knew that their solution is intractable. However, by using Beyond Worst Case Analysis, we showed that these worst cases are not really important, as they either almost never exist (Smoothed Analysis), or they are not meaningful (Perturbation Stability). In the next chapters, we are going to use Perturbation Stability in mechanism design, where many of the problems are intractable, to show that meaningful instances can have the desired properties that we talked in the precious chapter.

Chapter 5

Perturbation Stability in Combinatorial Auctions

5.1 Introduction

In the previous section we saw how Beyond Worst Case Analysis can help analyze problems to show that the worst cases, in which exist unwanted properties, are either not interesting or scarce in a way that makes them not important. In this section we are going to define the notion of perturbation stability in auctions, in order to later show that interesting-stable instances of auctions have a number of very important properties, which in the worst cases do not exist. To do that we need to define how to perturb an instance of a combinatorial auction. However, because combinatorial auctions are complicated, perturbing is not as easy as it was in the other problems that we saw before. In order to correctly define perturbation we first going to look at something called the *Endowment Effect*.

5.2 The Endowment Effect

The Endowment Effect was first used as a term by Thaler in [5], to explain situations where simply possessing an item, makes us value it more. This concept was generalized and used by Babaioff, Dobzinski, and Oren in [37] to show that an interesting result for combinatorial auctions: If we assume that after we have assigned the items to the bidders, their value for their newly possessed items increases, then it is easier to find allocations that make everyone happy. In order to illustrate their findings, let us first define what how they modeled and used the endowment effect.

Definition 5.1 (Endowment Effect[37]). Let i be a bidder, $v_i(\cdot)$ his valuation and S a bundle of items. Then if bidder i is endowed with set S by α, then his new valuation is

$$v_i^{S,\alpha}(T) = v_i(T) + (\alpha - 1) \cdot v_i(S \cap T)$$

Studying the above definitions we can make the following remarks. Firstly, the endowed value of a set that is a subset of S is simply multiplied by α. Secondly, if a set is disjoint with S then its endowed value is equal to its original value. With this definition we have a way to increase the valuation function of a bidder, but only "around" a desired set.

Using this definition Babaioff, Dobzinski, and Oren showed that if almost any allocation is endowed enough, then this allocation becomes a Walrasian equilibrium, meaning not only that it maximizes social welfare at the endowed valuations, but that there exists market clearing prices. However, their most interesting result is the following, concerning combinatorial auctions with submodular valuations.

Theorem 5.1 ([37]). *Let $(N, M, \mathbf{v})$ be a submodular combinatorial auction and $S_1, ..., S_n$ a local optimum. Then for any $\alpha \geq 2$, for the endowed valuations $v_1^{\alpha,S_1}, ..., v_n^{\alpha,S_n}$, the allocation $S_1, ..., S_n$ forms a Walrasian equilibrium.*

In order to understand this theorem, first we need to define what a local optimum is. As one might expect, a local optimum is an allocation whose welfare does not increase if we allocate 1 item to a different player, i.e. for any $i, k \in N$ and $j \in S_i$

$$v_i(S_i) + v_k(S_k) \geq v_i(S_i - \{j\}) + v_k(S_k \cup \{j\})$$

Now, what theorem 5.1 says is that if we found a local optimum at the non-endowed valuations, then allocation the local optimum creates endowed valuations, where we have a Walrasian equilibrium. This is useful, because taking into account the endowment effect, which says that after we acquire some items then we value them more, we can simply allocate the local optimum and then have a welfare maximizing allocation.

Since the proof of theorem 5.1 is quite simple, we are going to present it here. The proof uses prices $p_j = v_i(j|S_i - j)$ if $j \in S_i$ to show the existence of the Walrasian Equilibrium for the endowed valuations and breaks down into two parts. In the first part we prove that no bidder is willing to drop any items for which he has been endowed. This is only natural because each bidder's valuation has been increased at least by $(\alpha - 1) \cdot p_j$ for $j \in S_i$ because of submodularity. In the second part we show that no bidder is willing to obtain any items that are not in his endowed set, because of local optimality. The technical proof follows.

Proof. Fix a bidder i. First we are going to prove that for any set S that does not contain an item $j \in S_i$, $S \cup j$ provides not less utility for i. Fix such a set S and an item $j \in S_i$. Bidder i's utility by adding j to S increases by

$$\begin{aligned} I &= v_i^{a,S_i}(S \cup j) - v_i^{a,S_i}(S) - p_j \\ &= v_i(S \cup j) + (\alpha - 1)v_i\big((S \cap S_i) \cup j\big) - v_i(S) - (\alpha - 1)v_i(S \cap S_i) - p_j \end{aligned}$$

All we have to do now is prove that I is not negative. To do that we use monotonicity, $v_i(S \cup j) \geq v_i(S)$ and submodularity, $v_i\big((S \cap S_i) \cup j\big) - v_i(S \cap S_i) \geq v_i(S_i) - v_i(S_i - j)$. This, because of $\alpha - 1 \geq 0$ gives

$$I \geq (\alpha - 1) \cdot \big(v_i(S_i) - v_i(S_i - j)\big) - p_j \tag{5.1}$$

Because $\alpha \geq 2$ and the definition of p_j, from inequality 5.1 we have $I \geq 0$. Inductively, this gives us than bidder i has nothing to lose by adding the whole bundle S_i.

Now we simply need to prove that bidder i, who always demands a superset of S_i, has nothing to lose by discarding items not in S_i. Fix a bundle S, for which $S_i \subset S$ that contains an item j for which $j \in S_k$, for some $k \neq i$. By discarding j from S, bidder i's utility would increase by

$$\begin{aligned} I &= v_i^{a,S_i}(S-j) - v_i^{a,S_i}(S) + p_j \\ &= v_i(S-j) + (\alpha-1)v_i(S_i) - v_i(S) - (\alpha-1)v_i(S_i) + p_j \end{aligned}$$

Now because the allocation is a local optimum and by the definition of p_j, for which $j \in S_k$, we have that $p_j = v_k(S_k) - v_k(S_k - j) \geq v_i(S_i \cup j) - v_i(S_i)$. This gives us

$$I \geq v_i(S-j) - v_i(S) + v_i(S_i \cup j) - v_i(S_i)$$

Because of $S_i \cup j \subseteq S$ and submodularity $v_i(S_i \cup j) - v_i(S_i) \geq v_i(S) - v_i(S-j)$, which proves that $I \geq 0$. Inductively this gives us that i has nothing to lose by discarding items not in S_i.

To sum all the above up, we have that i has nothing to lose by gaining the whole bundle S_i and also nothing to lose by discarding items not in S_i. This means that S_i is his demand at prices p_j, proving the existence of a Walrasian Equilibrium at the endowed valuations. ■

Overall theorem 5.1 is quite nice and as said before seems quite practical: By allocating any local optimum and putting the prices of the equilibrium, because of the endowment effect, everyone has his demand and is happy. However there is a major flaw: Finding a local optimum is no easy task. Babaioff, Dobzinski, and Oren proved that even for 2 submodular players, finding a local optimum requires exponential many value queries.

A lot of also interesting results about the endowment effect were also found by Ezra, Feldman, and Friedler in [39], where they generalized the results of [37]. Their main result was that if we use a different and stronger form of endowment, then 2-endowing XOS valuations according to a welfare maximizing allocation, creates a Walrasian Equilibrium. The endowment that they used, given a valuation function $v(\cdot)$ and a bundle S was the following

$$v^{\alpha,S}(T) = v(T) + (\alpha-1) \cdot \big(v(S) - v(S-T)\big)$$

5.3 Definition of Stability in auctions

In this section we are going give the basic definitions for Perturbation Stability in Combinatorial Auctions. First we need to define how to perturb an instance of an auction. For this we are going to use the endowment effect that we talked about above, in order to be able to increase a certain valuation function. We will use the same endowment as in [37], although it does not matter as the definitions from [39] produce the same final result. A γ-perturbation of the original instance allows one bidder to endow his valuation for one item by γ.

Definition 5.2 (γ-Perturbation of Combinatorial Auctions)**.** For $\gamma \geq 1$, a *γ-perturbation of a Combinatorial Auction* $(N, M, \mathbf{v})$ is another combinatorial auction $(N, M, \mathbf{v}')$, where for one bidder $i \in N$, one item $j \in M$ and some $\sigma \in [1, \gamma]$:

- $\mathbf{v}_{-i} = \mathbf{v}'_{-i}$
- $v'_i(S) = v_i(S) + (\sigma - 1) \cdot v_i(S \cap \{j\})$, for every $S \subseteq M$

Simply put, the above perturbation keeps every valuation the same, except for one, which is endowed for one item.

Having defined how to perturb an auction, we are ready to define what is a perturbation stable combinatorial auction.

Definition 5.3 (Perturbation-Stability of Combinatorial Auctions)**.** For $\gamma \geq 1$, an instance of a combinatorial auction is called γ-*Perturbation Stable*, or simply γ-*Stable*, if its optimal allocation is unique, and remains unique under any γ-Perturbation.

To add some intuition to the definition above, it simply states that the optimal allocation is strong enough, so that it remains optimal even if any single item is endowed for any single bidder, by at most γ. Let us present an example to make the definition even clearer.

Example 5.1. Let Alice and Bob be our 2 bidders, in an auction of 2 items a and b. Alice and Bob are unit-demand. Alice values item a for 2 and item b for 1. Bob is symmetrical, meaning he values b for 2 and a for 1. Clearly the optimal allocation is $O_A = \{a\}, O_B = \{b\}$. Let us calculate how stable this auction is.
First we indeed notice that the optimal allocation is unique, which means that the auction is at least 1-stable. In order to check what is the maximum value of γ, we need to find how much we need to endow some item for some bidder, in order for the allocation (O_A, O_B) to be suboptimal. We easily notice that if we endow either Alice's value for item a, or Bob's value for item b the optimal allocation will only get stronger. Since the bidders are symmetrical we only need to check what happens when we endow Alice's value for b.
If we endow by γ, her new valuation is $v'_A(b) = \gamma$ and $v'_A(ab) = \gamma + 1$. Now the optimal allocation still has a social welfare of 4 and any other allocation that could become optimal (giving ab to Alice or b to Alice and a to Bob) has welfare equal to $\gamma + 1$. Thus in order for the optimal allocation to remain uniquely optimal it must hold that $4 > \gamma + 1$. This concludes that the auction is $(3 - \epsilon)$-stable or simply γ-stable, where $\gamma < 3$.

5.4 Properties of Perturbation-Stable Auctions

Having defined what makes an auction stable, we can start exploring some of the properties that derive from the definition of γ-stability. First we will see a very basic, but very fundamental property of stable auctions.

Claim 5.1. *Let $(N, M, \mathbf{v})$ be a γ-stable auction, for any $\gamma \geq 1$ and $O_1, ..., O_n$ the optimal allocation. Then it holds that*

$$M = \cup_{i \in N} O_i$$

Proof. We are going to show this by contradiction. Suppose that for some $j \in M$, $j \notin \cup_{i \in N} O_i$. Then by giving j to any bidder, the total social welfare is not going to change, because the allocation is already optimal. But this means that an allocation different from the optimal one, has the same welfare as the optimal. Because of the uniqueness of the optimal allocation in stable auctions, this leads to a contradiction. ■

Now we will prove a general lemma, that does not restrict to any valuation classes and is the foundation of γ-stable auctions.

Lemma 5.1 (Basic Propery of Stable Auctions)**.** *Let $(N, M, \mathbf{v})$ be a γ-stable auction. Then for every bidder $i \in N$, every item in i's optimal bundle $j \in O_i$ and any other bidder $k \neq i$ it holds that*

$$v_i(O - i) - v_i(O_i - j) > (\gamma - 1) \cdot v_k(j)$$

Proof. Fix an arbitrary bidder i, an item $j \in O_i$ and another bidder k, whose optimal set is O_k. Since the auction is γ-stable the optimal allocation will not change if we endow k's value for item j by γ, changing k's valuation from $v_k(\cdot)$ to $v'_k(\cdot)$. In the perturbed auction we compare the optimal allocation with the allocation which is the same us the optimal, but j goes to k instead of i:

$$\sum_{l \neq i,k} v_l(O_l) + v_i(O_i) + v'_k(O_k) > \sum_{l \neq i,k} v_l(O_l) + v_i(O_i - j) + v'_k(O_k \cup j)$$

$$v_i(O_i) + v_k(O_k) > v_i(O_i - j) + v_k(O_k \cup j) + (\gamma - 1) \cdot v_k(j)$$

$$v_i(O_i) - v_i(O_i - j) > v_k(O_k \cup j) - v_k(O_k) + (\gamma - 1) \cdot v_k(j) \tag{5.2}$$

Because v_k is non decreasing $v_k(O_k \cup j) - v_k(O_k) \geq 0$. Combining this and inequality 5.2 yields the lemma. ■

Now that we have proven a general property of stable auctions, we can restrict ourselves to more specific classes of valuations.

Corollary 5.1 (Property of Stable Subadditive Auctions)**.** *Let $(N, M, \mathbf{v})$ be a γ-stable auction with subadditive valuations. Then for every bidder $i \in N$, every item in i's optimal bundle $j \in O_i$ and any other bidder $k \neq i$*

$$v_i(j) > (\gamma - 1) \cdot v_k(j)$$

Proof. Because of lemma 5.1 we already know that

$$v_i(O - i) - v_i(O_i - j) > (\gamma - 1) \cdot v_k(j) \tag{5.3}$$

and because of subadditivity that

$$v_i(j) + v_i(O_i - j) \geq v_i(O_i) \tag{5.4}$$

Combining inequalities 5.3 and 5.4 yields the corollary. ■

Corollary 5.2 (Property of Stable XOS Auctions)**.** *Let $(N, M, \mathbf{v})$ be a γ-stable auction with XOS valuations. Denote with $q_i(\cdot)$ the additive function that supports bidder i's optimal bundle O_i. Then for every bidder $i \in N$, every item in i's optimal bundle $j \in O_i$ and any other bidder $k \neq i$*

$$q_i(j) > (\gamma - 1) \cdot v_k(j)$$

Proof. Because of lemma 5.1 we already know that

$$v_i(O - i) - v_i(O_i - j) > (\gamma - 1) \cdot v_k(j) \tag{5.5}$$

and because valuations are XOS

$$v_i(O_i - j) \geq \sum_{t \in O_i - j} q_i(t) \tag{5.6}$$

Combining $v_i(O_i) = \sum_{t \in O_i} q_i(j)$, inequalities 5.5 and 5.6 yields the corollary. ■

Corollary 5.3 (Property of Stable Submodular Auctions)**.** *Let* $(N, M, \mathbf{v})$ *be a* γ*-stable auction with submodular valuations. Then for every bidder* $i \in N$*, every item in i's optimal bundle* $j \in O_i$*, any subset* $S \subseteq O_i$ *and any other bidder* $k \neq i$

$$v_i(S) - v_i(S - j) > (\gamma - 1) \cdot v_k(j)$$

Proof. Because of lemma 5.1 we already know that

$$v_i(O_i) - v_i(O_i - j) > (\gamma - 1) \cdot v_k(j) \tag{5.7}$$

and because valuations are submodular and $S \subseteq O_i$

$$v_i(S) - v_i(S - j) \geq v_i(O_i) - v_i(O_i - j) \tag{5.8}$$

Combining inequalities 5.7 and 5.8 yields the corollary. ■

With this corollaries we are going to prove the majority of the theorems that will follow in the next two chapters. First we are going to improve the results for general auctions and secondly we are going to improve the bounds associated with Price of Anarchy in simple auctions.

Chapter 6

Mechanisms

In this chapter we are going to look at different mechanisms, which either find the optimal allocation or try to approximate the optimal allocation, usually up to a constant factor. This includes two different sections: First we are going to look at direct revelation mechanisms with use value queries and then at mechanisms that use demand queries. At each of these two sections, at the beginning we will focus on general settings, while at the end we will analyze mechanism that take advantage of the stability hypothesis.

6.1 Direct Revelation Mechanisms

In this section we are going to look at mechanisms that are direct revelation, meaning that the bidders are required to answer *Value Queries*. The definition of a value query follows.

Definition 6.1 (Value Query)**.** Let S be a set of items, i a bidder and $v_i(\cdot)$ his valuation function. Then bidder i's answer to a *Value Query* for the set S is value for the set S, meaning $v_i(S)$.

This kind of mechanisms are considered restrictive, because demand queries (see section 6.2) are strictly more powerful than value queries, which we will prove later. Despite that fact, there are many mechanisms that achieve great results using only value queries. First we are going see some of these mechanisms, without the assumption that the auction is stable.

6.1.1 Direct Revelation without Stability

First we take a look at a well known algorithm made by Lehmann, Lehmann, and Nisan from [8], that approximates the optimal allocation up to a factor of 2, when the valuations are submodular. As we also did in the introduction, let us state that for submodular valuations with value queries, there exists a better, but more complicated algorithm, that of [20], with a ratio of $\frac{e}{e-1}$.
The algorithm that performs on the simple following idea: Give the items sequentially, each time at the bidder with the largest marginal contribution, i.e. to the bidder who has the highest increase in his valuation, given his current set.

Algorithm 3 2-Approximation Algorithm for Submodular valuations [8]

1: Set $S_1 = S_2 = ... = S_n = \emptyset$
2: **for** $j \in M$ **do**
3: Let i be the bidder that maximizes $v_i(S_i \cup \{j\}) - v_i(S_i)$
4: Allocate item j to bidder i, i.e. set $S_i \leftarrow S_i \cup \{j\}$
5: **end for**
6: **return** allocation $(S_1, ...S_n)$

We can easily tell that the algorithm uses a polynomial number of value queries and to be exact, in each iteration $2n$ value queries are made. Next follows the main result concerning the algorithm.

Theorem 6.1 ([8]). *Algorithm 3 provides a 2-approximation to the optimal allocation.*

Proof. Let $(O_1, ..., O_n)$ denote the optimal allocation and $(S_1, ..., S_n)$ the allocation of the algorithm. Without loss of generality let $M = \{1, ..., m\}$ the items in the order they were iterated in step 2. Also denote with Δ_j the value that item j added to the total allocation in step 4. Fix a random bidder i and denote with X_t the first t items of set $O_i - S_i$. Note that $X_0 = \emptyset$ and $X_{|O_i - S_i|} = O_i - S_i$. Now we have that

$$\begin{aligned} v_i(O_i \cup S_i) - v_i(S_i) &= \sum_{t=0}^{|O_i - S_i|-1} \big(v_i(O_i \cup S_i - X_t) - v_i(O_i \cup S_i - X_{t+1})\big) \\ &\leq \sum_{t=0}^{|O_i - S_i|-1} \big(v_i(S_i - X_t) - v_i(S_i - X_{t+1})\big) \\ &\leq \sum_{j \in O_i - S_i} \Delta_j \\ &\leq \sum_{j \in O_i} \Delta_j \end{aligned}$$

where the first inequality holds because of submodularity and the second because in the algorithm we take the item with the largest marginal contribution Δ_j. Summing the above inequality over all bidders and using the facts that $v_i(O_i) \leq v_i(O_i \cup S_i)$ and $\sum v_i(S_i) = \sum_i \sum_{j \in O_i} \Delta_j$

$$\sum_i v_i(O_i) - \sum_i v_i(S_i) \leq \sum_i v_i(S_i)$$

we prove the theorem. ∎

The algorithm above provides a constant factor approximation for auctions with submodular valuations. A natural question that one might ask is if we can calculate the optimal solution in polynomial time. The answer is no, as the following theorem suggest.

Theorem 6.2 ([8]). *Finding the optimal allocation in a combinatorial auction with two valuations that are additive and budget-additive is NP-Hard.*

In the theorem, instead of submodular valuations, additive and budget-additive are used. We have stated that additive valuations are a subclass of submodular and the same can be proved for budget additive valuations. The proof that follows is slightly different than the proof provided by Lehmann, Lehmann, and Nisan et. al.

Proof. We will reduce the *Subset Sum* problem: Given a set of non-negative integers $U = \{a_1, ..., a_m\}$ and an integer t, find if exists a subset $S \subseteq U$ such that $\sum_{j \in S} a_j = t$. The auction to which the problem will be reduced has two bidders. The first is additive: $v_1(S) = \sum_{j \in S} a_j$ and the second is budget-additive: $v_2(S) = 2\, min\left(t, \sum_{j \in S} a_j\right)$. Denote with F the sum of the integers $a_1, ..., a_m$. Note that a calculation of a value query can be done in polynomial time, using only the subset-sum's input. Fix an optimal allocation S_1, S_2, where all the items are allocated. Its total welfare is

$$v_1(S_1) + v_2(S_2) = \sum_{j \in S_1} a_j + 2\, min(t, \sum_{j \in S_2} a_j) \leq \sum_{j \in S_1} a_j + t + \sum_{j \in S_2} a_j = F + t$$

This inequality proves that $F + t$ is an upper bound for the optimal allocation and that the equality holds only when $\sum_{j \in S_2} a_j = t$. This means that by finding the optimal allocation, we can check if its social welfare is less or equal to $F + t$, which immediately tells us if there is a subset of the integers that have sum equal to t, thus solving the subset sum problem. ■

The theorem above covers the subject of exact calculation of the optimal allocation. Let us also note that approximating the problem only with value queries with a factor of $\frac{e}{e-1}$ is also an NP-Hard problem [12]. What we didn't cover is if algorithm 3 can be truthful. The answer is no, which can be shown by a simple example where no pricing scheme can make the bidders be truthful. However, something even more powerful holds. Dobzinski showed in [22] than best approximation ratio that a truthful polynomial mechanism with value queries can achieve is $\sqrt{m}$.

Theorem 6.3 ([22])**.** *Let A be a randomized universally truthful mechanism for combinatorial auctions with submodular valuations that provides an approximation ratio of $m^{0.5-\epsilon}$, for some constant $\epsilon > 0$. Then, A makes exponentially many value queries.*

The above theorem suggests that the approximation ratio of algorithm 3 is unachievable when truthfulness is an issue. Let us add that an algorithm with approximation ratio of $O(\sqrt{m})$ has been fund in [21].

6.1.2 Direct Revelation with Stability

In this section we are going to present our algorithms that have surprisingly good results, provided that the auction is stable. First we start with a mechanism for subadditive valuations.

The algorithm that follows is similar to algorithm 3, but instead of greedily trying to maximize the marginal contribution of each item, it greedily maximizes the singleton contribution of each item, i.e. it allocates each item to the bidder who has the highest contribution for the item as a singleton.

Algorithm 4 Optimal Algorithm for 2-stable Subadditive Auctions

Set $S_1 = S_2 = ... = S_n = \emptyset$
for $j \in M$ **do**
  Let i be the bidder that maximizes $v_i(j)$
  Allocate item j to bidder i, i.e. set $S_i \leftarrow S_i \cup \{j\}$
end for
return allocation $(S_1, ...S_n)$

Theorem 6.4. *Let $(N, M, \mathbf{v})$ be 2-stable combinatorial auction with subadditive valuations. Then algorithm 4 outputs the optimal allocation $(O_1, ..., O_n)$ with polynomial number of value queries.*

In order to prove theorem 6.4(the polynomial part is trivial) one has to simply use corollary 5.1 that states exactly what we need: The bidder who gets each item at the optimal allocation, is the one with the largest singleton value.

Proof. Fix an item j and suppose that i is the bidder for who $j \in O_i$. Because of corollary 5.1 and the fact that the auction is 2-stable we know that $v_i(j) > v_k(j)$, for any other bidder $k \neq i$. Because j is allocated to the bidder with the higher singleton value, i will receive item j. This concludes the proof. ■

This is immediately a better result than the 2-approximation algorithm 3, because it both finds the optimal solution and works for a much larger class of of valuation functions. We should also add that algorithm 3 also finds the optimal allocation when the valuations are submodular and the auction is 2-stable. However, what the above algorithm lucks is truthfulness.

In order to extend our algorithm to be truthful, we must come up with some sort of payment that the bidders must pay. First, we need to we restrict ourselves to submodular valuations and run the same algorithm, while charging each bidder the second highest singleton value. This is actually a *Parallel Second Price Auction*, which generally does very poorly, and only works on restricted classes of valuations, e.g. additive.

Algorithm 5 Parallel Second Price Auction (P2A)

Set $S_1 = S_2 = ... = S_n = \emptyset$
for $j \in M$ **do**
  Collect n bids for item j, $(b_{1j}, ..., b_{nj})$
  Let i be the bidder that maximizes b_{ij}
  Allocate item j to bidder i, i.e. set $S_i \leftarrow S_i \cup \{j\}$
  Set $p_j = \max_{k \neq i} b_{kj}$
end for
For each i set $P_i = \sum_{j \in S_i} p_j$
return allocation $(S_1, ...S_n)$ and payments $(P_1, ...P_n)$

We should note that during the run of this mechanism, bidders do not learn what they are allocated, meaning that in every iteration we expect $b_{ij} = v_i(j)$. A simple implementation to achieve this is to collect all the bids at once.

P2A, given that bidders bid truthfully, finds the optimal allocation, just like algorithm 4 does. What's left is to prove its truthfulness.

Theorem 6.5. *Mechanism 5 is EPIC (Ex-Post Incentive Compatible) for* 2*-stable combinatorial auctions.*

This theorem might seem more like a corollary from corollary 5.3: For a certain bidder, if other bidders bid truthfully, since his singleton values are strictly less for items not in his optimal set, he strictly loses utility by obtaining them. On the other hand, for the items in his optimal set, since his minimum marginal value is greater than the other bidders' singleton values, he only gains utility by obtaining them.

Proof. Fix a bidder i, his valuation $v_i(\cdot)$ and suppose that any other bidder k bids his singleton value $v_k(j)$ for each item $j \in M$. We need to prove that i has nothing to gain if he bids untruthfully.

We will show this by contradiction. Suppose that i makes a different bid and ends up with higher utility. In order to change his utility he must change either his allocated set or the payments. Since the payments are independent of his bids he must to be allocated a set S different than O_i. First we prove that it must hold that $S \subseteq O_i$.

Let's assume the opposite, that there exists a j for which $j \in S$ and $j \notin O_i$. Then i's utility by dropping j would decrease by $V = v_i(S) - v_i(S - j)$, but it would increase by $P = \max_{k \neq i} v_k(j)$. Because j belongs to some O_k, by corollary 5.1, and 2-stability $P > v_i(j)$. Also because of submodularity, $V < v_i(j)$. This shows that i has strictly higher utility if he drops any items not in O_i. Now we have to disprove that $S \subset O_i$.

By contradiction, suppose that there exists a j such that $j \in O_i$, but $j \notin S$. Then i's utility by acquiring j would increase by $V = v_i(S \cup j) - v_i(S)$ and decrease by $P = \max_{k \neq i} v_i(j)$. From corollary 5.3, we know that $P < v_i(S \cup j) - v_i(S)$, which means that bidder i gains strictly positive utility by obtaining any item in O_i.

The theorem is now proven. We have shown that bidder i strictly loses utility from items outside of O_i and strictly gains utility from items in O_i. ■

Algorithm 5 and theorem 6.5 provide a simple, elegant and truthful mechanism for submodular 2-stable auctions. This both solves the problem of non-approximability with value queries under a factor of $\sqrt{m}$, as well as the inability to find the optimal allocation in polynomial time.

Let us also state the analogy of our results, with the results presented in section 4.3.1: In both cases, a simple and intuitive algorithm, which in the general setting had no guaranteed results, in the stable setting achieved optimal results.

6.2 Auctions with demand queries

In the previous section we analyzed mechanisms that used value queries. As we discussed, there is also another type of queries, called *Demand Queries*. In this section we are going to describe auctions that use demand queries, to either find the optimal allocation or approximate it. First though we need to describe what a demand query is. Simply put, when a bidder is presented with a vector of prices and is asked to answer a demand query, we expect him to answer with a bundle that maximizes his utility.

Definition 6.2 (Demand Querry)**.** Given a bidder with valuation $v(\cdot)$ and a vector of prices $\mathbf{p} = (p_1, ..., p_m)$, the *Demand Query* $D(v, \mathbf{p})$ is the demand of the bidder, i.e. the bundle $S \subseteq M$ that maximizes $v(S) - \sum_{j \in S} p_j$.

As we have noted before, demand queries are strictly stronger than value queries. The reason for that is that a value query can be simulated with a polynomial number of demand queries. Intuitively, one can use binary search on the prices to find the value of a certain set.

Lemma 6.1 ([19])**.** *The value query for a bundle S can be simulated by $t \cdot |S|$ demand queries, where t is the number of bits of precision in the representation of a bundle's value.*

Proof. W.l.o.g let $S = \{1, ..., m\}$, where $m = |S|$ and let S_i denote the first i items of S, i.e. $S_i = \{1, ..., i\}$. By calculating every $v(S_i) - v(S_{i-1})$, we can find $v(S) = \sum_{i=1}^{m} v(S_i) - v(S_{i-1})$.
To calculate $v(S_i) - v(S_{i-1})$, we set $p_j = 0$, for $j \in S_{i-1}$, $p_j = \infty$, for $j \notin S_i$ and run a binary search on p_i to find at which price the bidder starts preferring $v(S_i)$ to $v(S_{i-1})$. That price is equal to $v(S_i) - v(S_{i-1})$.
We need to find $|S|$ marginal values and each binary search requires t value queries, giving a total of $t \cdot |S|$ value queries. ■

It is clear now that demand queries are stronger than value queries, as we can simulate the latter with the first. But can we also do the opposite? Then answer is no, and that is the reason why demand queries are strictly stronger.

Lemma 6.2 ([19])**.** *An exponential number of value queries may be required for simulating a single demand query.*

In order to see the first mechanism that utilizes demands queries, first we are going to give the proper definition for the gross-substitutes class of valuations. If a valuation function is gross-substitutes, then given some prices and a demand set for that prices, increasing the prices for some items, will not make the bidder not demand the items for which the prices remain unchanged. The proper definition follows.

Definition 6.3 (Gross-Substitutes Function)**.** A set function $v : 2^M \mapsto \mathbb{R}^+$ satisfies the *gross-substitutes* condition, if for any price vectors $\mathbf{p}, \mathbf{q}$ such that $\mathbf{p} \geq \mathbf{q}$ (pairwise comparison) and any set $S \in D(v, \mathbf{p})$, there exists a bundle $T \subseteq M$ such that

$$T \in D(v, \mathbf{q}) \text{ and } S - \Lambda \subseteq T$$

where $\Lambda = \{j \in M : q(j) > p(j)\}$

6.2.1 Demand queries without Stability

Seeing the above definition one might start coming up with ways to calculate the optimal bundle for auctions where the valuations satisfy the gross-subtitutes condition. For instance a good idea is that if a bidder has his demand, and we take an arbitrary subset of his demand and raise the prices, he can always keep the items that he already has and get some more to again stay satisfied with a new set that maximizes his utility. This is the idea that the *Kelso-Crawford Auction* utilizes.

Algorithm 6 The Kelso-Crawford Auction[7]

Set $S_1 = S_2 = ... = S_n = \emptyset$
Set $p_1 = p_2 = ... = p_m = 0$
while true **do**
  Ask each bidder i for a set T_i, that maximizes $v_i(S_i \cup T_i) - \sum_{j \in S_i} p_j - \sum_{j \in T_i}(p_j + \epsilon)$
  If for every bidder i it holds that $T_i = \emptyset$, **output** the allocation $(S_1, ..., S_n)$
  Otherwise **pick** arbitrarily a bidder i for whom $T_i \neq \emptyset$.
  Set $S_i \leftarrow S_i \cup T_i$
  For each $k \neq i$, **set** $S_k \leftarrow S_k - T_i$
  For each $j \in T_i$ **set** $p_j \leftarrow p_j + \epsilon$
end while

The Kelso-Crawford mechanism can be summarized in 2 key points:

1. Once a bidder takes an item, he cannot lose it, unless another bidder demands it.
2. In every round, some of the items that are demanded by some bidder, have their prices increases.

Taking into account the definition of the gross-subtitutes and the two key points of the Kelso-Crawford, one might begin to notice that the these two fit perfect: Even if a gross-subtitutes bidder loses some items, he can keep what he already owns, maybe take some more items, and end up with a bundle that maximizes his utility either way. Indeed, we can easily prove that the Kelso-Crawford auction outputs a $m\epsilon$-Walrasian equilibrium, meaning that unsold items have price 0 and every bidder gets a bundle that yields utility within $m\epsilon$ of his preferred bundle.

Theorem 6.6. *Let $(N, M, \mathbf{v})$ be a combinatorial auction, where all bidder's valuations satisfy the gross-subtitutes condition. Then the Kelso-Crawford Auction terminates with prices and bundles that form a $m\epsilon$-Walrasian equilibrium.*

Proof. Fix a bidder i. Using induction we are going to show that in each iteration of the auction, after bidder i has picked his set T_i, the bundle $S_i \cup T_i$ is the demand of i at prices p_j for $j \in S_i$ and $p_j + \epsilon$ for $j \notin S_i$.
At the start of the auction the invariant holds because $S_i = \emptyset$, which means that the set $S_i \cup T_i$ can be whatever bidder i wants.
For the inductive step, consider an arbitrary iteration, where at some previous point bidder i chose and took his demand S_i, but has now lost a subset Λ of S_i because the prices for items in Λ used to be p_j, but now are q_j. We notice that this perfectly fits the definition of the gross-subtitutes: Since S_i was the demand of bidder i at prices $\mathbf{p}$ and now he has $S_i - \Lambda$ at the same prices and can also choose from the rest of the items at prices $\mathbf{q}$, there is a set T_i, such that $T_i \cup S_i$ is the demand of i.
Thus we have shown that at each iteration where a bidder changes his bundle, he ends up with his demand. This means that this also holds at the end of the mechanism. Lastly

we need to state that the final allocation and prices are not a Walrasian equilibrium, but a $m\epsilon$-Walrasian equilibrium, because bidder i has his demand if the items outside of S_i have their price increase be ϵ. ■

Having found a good mechanism that can calculate the best solution is pseudo polynomial time, one may ask if the algorithm is better than simply calculating the optimal solution: Is the algorithm truthful? The answer unfortunately is no, even for 2 bidders. The intuition is that a bidder might be willing to drop some items from his optimal set, in order to make another bidder satisfied and keep him from raising the prices on the other items. This is exactly what happens in the example that follows.

Example 6.1. Let Alice and Bob be our 2 bidders, in an auction of m items. Alice is additive and values each item the same, equal to a, with $a > 1$. Bob is unit demand and also values each item the same, equal to 1. Notice that both are gross-substitutes. One can easily tell that the unique optimal solution is for Alice to receive all the items, for a total value of $a \cdot m$. We should also note that the minimum price on each item in order to have a Walrasian equilibrium is exactly 1. This is also the solution that Kelso-Crawford finds in the end, because in each iteration, either will Alice hold all the items and Bob is going to pick the one with the lowest price, if that is lower than 1, or will Bob have an item, which Alice is going to take.
If Alice is truthful, then she will take all the items, each at price 1, for a total utility of $(a - 1) \cdot m$. However Alice has a better strategy: Let Bob keep the first item, which will keep him happy and take the rest of the items at price 0. This yields a total utility of $a \cdot (m - 1)$. As long as $a < m$, Alice is going to chose the latter strategy, giving a sub optimal solution.

An immediate result from theorem 6.6, is that gross-substitutes always have a Walrasian equilibrium, something that is not true for the more general class of submodular valuations. However the Kelso-Crawford auction can be used for submodular valuations, giving a good approximate result. If the valuations are submodular, then each bidder will never have negative valuation by taking some items at some prices, because after he loses some of them, the marginal contribution of each item will be even larger that before. Beside this point, we can also achieve an approximation ratio of 2.

Theorem 6.7 ([25]). *Let $(N, M, \mathbf{v})$ be an auction with submodular valuations. Then running the Kelso-Crawford auction on $(N, M, \mathbf{v})$ yields a 2-approximation of the optimal solution, up to $m\epsilon$, and each bidder gains non negative utility.*

Proof. First we will prove the second part of the theorem using induction: At every step of the auction, for every bidder i and his set S_i, it holds that for every $T \subseteq S_i$

$$v_i(T) \geq \sum_{j \in T} p_j$$

Meaning that he has non zero utility for any subset of the items he holds. In the base case where every bidder has not been allocated any items yet, the proof is trivial. For the inductive step we need to prove that the invariant holds, when a bidder is picked to increase his bundle and when a bidder's bundle gets smaller because some of his items are picked. In the second case the invariant holds, simply because the prices stay the same and the invariant held for a superset of items.

In the case where i increases his bundle from S_i to $S_i \cup T_i$ (where we assume that S_i and T_i are disjoint), because $S_i \cup T_i$ maximizes his utility it must hold that for every $T \subseteq T_i$

$$v_i(S_i \cup T_i) - \sum_{j \in T_i} p_j \geq v_i(S_i \cup T) - \sum_{j \in T} p_j \tag{6.1}$$

Where we assume that the p_j for the items in T_i have already been raised. Now imagine that the invariant does not hold: $v_i(A) < \sum_{j \in A} p_j$, for some bundle A. W.l.o.g. label the items in $A = \{1, ..., a\}$ where the first l items are the ones that are in S_i and the items from $l + 1$ to a are the ones that are in T_i. Now denote with X_j the first j items if A, i.e. $X_j = \{1, ..., j\}$. Note that $X_0 = \emptyset$ and $X_a = A$. Using this notation, because A breaks the invariant we have that

$$v_i(X_l) + \sum_{j=l+1}^{a} \big(v_i(X_j) - v_i(X_{j-1})\big) < \sum_{j=1}^{l} p_j + \sum_{j=l+1}^{a} p_j \tag{6.2}$$

Because $X_l \subseteq S_i$ and the invariant holds for subsets of S_i we have that $v_i(X_l) \geq \sum_{j=1}^{l}$. This means that in inequality 6.2 for some $j \geq l + 1$, it must hold that

$$v_i(X_j) - v_i(X_{j-1}) < p_j \tag{6.3}$$

otherwise inequality 6.2 is false. Since $j > l$ it holds that $j \in T_i$, which used in inequality 6.3, because of submodularity, produces the following

$$v_i(S_i \cup T_i) - v_i(S_i \cup T_i - j) < p_j$$

This inequality contradicts with inequality 6.1 if we use $T = T_i - j$. Thus we proved that the invariant always holds.

Now we are going to prove the first part of the theorem, that the final allocation, from now on denoted with $S_1, .., S_n$, is a 2-approximation of the optimal one. Fix a bidder i. Bidder i has nothing to gain by extending his set to $S_i \cup O_i$. This means that

$$v_i(S_i) \geq v_i(S_i \cup O_i) - \sum_{j \in O_i - S_i} (p_j + \epsilon) \geq v_i(O_i) - \sum_{j \in O_i} (p_j + \epsilon) \tag{6.4}$$

where the second inequality holds because of monotonicity and the fact that the prices are non-negative. Summing inequality 6.4 over all i gives

$$\sum_i v_i(S_i) \geq \sum_i v_i(O_i) - \sum_i \sum_{j \in O_i} (p_j + \epsilon) \tag{6.5}$$

Now because $M = \cup_i O_i$ and $p_j = 0$ for the items that were not allocated, inequality 6.5 becomes

$$\sum_i v_i(S_i) \geq \sum_i v_i(O_i) - \sum_i \sum_{j \in S_i} p_j - m\epsilon \tag{6.6}$$

Now we make the following simple note: Because of the invariant, $v_i(S_i) \geq \sum_{j \in S_i} p_j$. Using this in inequality 6.6 gives us the theorem. ■

This completes the analysis of the Kelso-Crawford Auction, which as stated before is a non-truthful auction. For completeness, let us show the state of the art for truthful auctions with demands queries. We would like to remind the results of [22], where a lower approximation bound of $\sqrt{m}$ was shown, for truthful auctions with submodular valuations, using only value queries. Because of that, we would expect that when using the stronger demand queries the approximation would be much better.

This is indeed the case. One of the most known and simple algorithms that provides an expected approximation ratio of $O(\log m)$ for XOS bidders in truthful auctions with demand queries, is that of Krysta and Vöcking in [26]. However, the state of the art for the same setting is the algorithm of Dobzinski. In [33] he proves that there exists a mechanism that is truthful and has expected approximation ratio $O(\sqrt{\log m})$. Unfortunately, this mechanism is too complicated to even state here.

6.2.2 Demand queries with Stability

In this section we are going to revisit some of the mechanisms of the previous section and we are going to show their advanced potential, under the assumption that the auction is stable.

First we are going to prove a very basic fact about 2-stable auctions with submodular valuations: The very important fact that they always admit a Walrasian equilibrium. To prove this we are going to us once again corollary 5.3. For each item j that $j \in O_i$, at price $p_j = v_i(O_i) - v_i(O_i - j) > v_k(j)$, every other bidder does not want the item since it is higher than his singleton value and bidder i wants it because the price is equal to the smallest marginal value.

Theorem 6.8. *Let $(N, M, \mathbf{v})$ be a 2-stable auction with submodular valuations. Then the prices $p_1, ..., p_m$, where if $j \in O_i$ then $p_j = v_i(j|O_i - j) = v_i(O_i) - v_i(O_i - j)$, form a Walrasian equilibrium.*

Proof. Fix a bidder i and his optimal bundle O_i. Suppose that his demand is the set S. First we are going to prove that if $j \notin O_i$, then $j \notin S$. Using lemma 5.1 we know that for some k, $p_j = v_k(j|O_k - j) > v_i(j)$. This means that if bidder i dropped item j from S, then his utility would decrease by $v_i(S|S - j)$, which by submodularity is at most $v_i(j)$, but it would also increase by p_j, which is strictly larger than $v_i(j)$. This means that dropping item j, i's utility would strictly increase, which means that j cannot be in his demand.
Now we need to prove that if $j \in O_i$, then i has nothing to lose by gaining j. Fix such an item j and a demand S of bidder i. Note that we have already proven that $S \subseteq O_i$. We will show that by adding j to S, i's utility would not decrease, making $S \cup j$ also the demand of i. Adding j to S, increases i's utility by $v_i(j|S)$ which is more than $v_i(j|O_i - j)$, because of submodularity and $S \subseteq O_i - j$. By adding j to S, bidder i's utility also decreases by $p_j = v_i(j|O_i - j)$. This shows that i's utility cannot decrease by adding j to his demand. By induction, this concludes that $S = O_i$ and also completes the proof of the theorem. ∎

Thus we have shown that stability breaks the barrier of the non existence of Walrasian equilibrium for submodular valuations. This also means that we can find the optimal solution by solving the LPR with demand queries and the ellipsoid algorithm. This also gives a polynomial time algorithm that finds the optimal solution, just like when the valuations satisfy the gross-substitutes condition. We can also find another similarity between gross-substitutes and stable submodular auctions: The Kelso-Crawford auction always finds the optimal solution.

Theorem 6.9. *Let* $(N, M, \mathbf{v})$ *be a* 3*-stable auction with submodular valuations. Then the Kelso-Crawford auction, by taking the* ϵ *term sufficiently small, outputs the optimal allocation.*

Proof. For every item j, that $j \in O_i$, let $w_j = \max_{k \neq i} v_k(j)$, i.e. the second highest singleton value for item j. It is easy to show that for every item j, its price p_j will exceed w_j, only when j is allocated to i in the final allocation, where $j \in O_i$. That is because only i can yield an addition to his utility by such a price.
Now let $(S_1, ..., S_n)$ be the final allocation of the Kelso-Crawford Auction and suppose that it is different than the optimal one. Because $(S_1, ..., S_n)$ is the final allocation, any bidder i has no gain to switch from S_i to $S_i \cup O_i$:

$$v_i(S_i) - \sum_{j \in S_i} p_j \geq v_i(S_i \cup O_i) - \sum_{j \in S_i} p_j - \sum_{j \in O_i - S_i} p_j + \epsilon$$

From now on we will assume that the ϵ terms are small enough, so that they can be ignored. Because $v_i(\cdot)$ is non-decreasing and submodular, we know that $v_i(S_i \cup O_i) \geq v_i(O_i)$ and $\sum_{j \in S_i - O_i} v_i(j) + v_i(S_i \cap O_i) \geq v_i(S_i)$. Using those in the above inequality

$$\sum_{j \in S_i - O_i} v_i(j) + v_i(S_i \cap O_i) \geq v_i(O_i) - \sum_{j \in O_i - S_i} p_j$$

But, as we argued before, for each item in $O_i - S_i$, the price must be lower than w_j, because no bidder other that i would have the incentive to increase the price above w_j. Also for the items $j \in S_i - O_i$ it holds that $w_j \geq v_i(j)$, from the definition of the prices w_j and the fact that $j \notin O_i$. All these in the above inequality yield:

$$\sum_{j \in S_i - O_i} w_j + \sum_{j \in O_i - S_i} w_j \geq v_i(O_i) - v_i(S_i \cap O_i) \tag{6.7}$$

Summing inequality 6.7 for all i, and using the fact that $\cup_i \big(S_i - O_i\big) = \cup_i \big(O_i - S_i\big)$ (both unions represent the items not optimally allocated) we get that

$$\sum_i \sum_{j \in O_i - S_i} (2 \cdot w_j) \geq \sum_i \big(v_i(O_i) - v_i(S_i \cap O_i)\big) \tag{6.8}$$

Now because of lemma 5.1 we have that $v_i(O_i) - v_i(O_i - j) > 2 \cdot w_j$ for items $j \in O_i$. This makes inequality 6.8

$$\sum_i \sum_{j \in O_i - S_i} \big(v_i(O_i) - v_i(O_i - j)\big) > \sum_i \big(v_i(O_i) - v_i(S_i \cap O_i)\big) \tag{6.9}$$

Thus because it holds that $\sum_{j \in O_i - S_i} \big(v_i(O_i) - v_i(O_i - j)\big) \leq v_i(O_i) - v_i(S_i \cap O_i)$ for submodular valuations, we have that inequality 6.9 can't hold. Thus we have reached a contradiction, which completes the proof of the theorem. ■

Chapter 7

Price of Anarchy in Auctions

In this chapter we are going to look at the Price of Anarchy, or POA, in simple mechanisms. By simple we mean a mechanism where each bidder i makes a bid for each item b_{ij}. Then each item j goes to the bidder with the highest bid, for some price. Let us formally define what we mean by simple mechanism.

Definition 7.1. Let $(N, M, \mathbf{v})$ be a combinatorial auction. A mechanism is called *simple*, when each bidder $i \in N$ reports a bid vector $\mathbf{b}_i = (b_{i1}, ..., b_{im})$, gets allocated the set of items for which his bids are the highest, i.e. $S_i = S_i(\mathbf{b}) = \{j \in M : b_{ij} = \max_k b_{kj}\}$ and for each item he gets pays a function of all the bids $p_j = p_j(\mathbf{b})$, where $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_n)$. Ties are broken arbitrarily.

7.1 Introduction

We are going to study the price of anarchy of some simple auctions. POA quantifies how bad is the social welfare of an equilibrium, compared to the optimal social welfare. First let us define the different kinds of equilibrium, from the perspective of auctions.

First we have the Pure Nash Equilibrium, where each bidder has a non-randomized bid vector, which is the best response to the bids of the other players.

Definition 7.2 (Pure Nash Equilibrium (PNE))**.** Let $(N, M, \mathbf{v})$ be a combinatorial auction. A bid profile $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_n)$ is a *Pure Nash Equilibrium (PNE)* at a simple mechanism with pricing policy $p_j(\mathbf{b})$, when for every bidder i and any other bid vector $\mathbf{b}'_i$, i has no incentive to change his bid tp $\mathbf{b}'_i$, i.e.

$$v_i(S_i(\mathbf{b})) - \sum_{j \in S_i(\mathbf{b})} p_j(\mathbf{b}) \geq v_i(S_i(\mathbf{b}'_i, \mathbf{b}_{-i})) - \sum_{j \in S_i(\mathbf{b}'_i, \mathbf{b}_{-i})} p_j(\mathbf{b}'_i, \mathbf{b}_{-i})$$

Secondly we have the Mixed Nash Equilibrium, where each bidder is allowed to have a randomized bid, which maximizes the expected utility of the bidder.

Definition 7.3 (Mixed Nash Equilibrium (MNE))**.** Let $(N, M, \mathbf{v})$ be a combinatorial auction. A profile of bid vector distributions $\mathbf{D} = (D_1, ..., D_n)$, is called a *Mixed Nash Equilibrium (MNE)* at a simple mechanism with pricing policy $p_j(\mathbf{b})$, if for every bidder

i and any other bid distribution D'_i, i has no incentive to play according to D'_i, i.e.

$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}}\left(v_i(S_i(\mathbf{b})) - \sum_{j\in S_i(\mathbf{b})} p_j(\mathbf{b})\right) \geq \mathop{\mathbb{E}}_{\mathbf{b}\sim(D'_i,\mathbf{D}_{-i})}\left(v_i(S_i(\mathbf{b})) - \sum_{j\in S_i(\mathbf{b})} p_j(\mathbf{b})\right)$$

These are two of the basic concepts of Nash Equilibriums in auctions. There are more general equilibria, but we are not going to mention them here. Now we are ready to define what Price of Anarchy is. We will define it for MNE, but the definition is similar for PNE.

Definition 7.4 (Price of Anarchy (POA) for MNE)**.** Let $(N, M, \mathbf{v})$ be a combinatorial auction and $\mathbf{D} = (D_1, ..., D_n)$ a MNE for that auction. Then the *Price of Anarchy* is the ratio between the expected welfare of the equilibrium and the welfare of the optimal allocation, i.e.

$$POA = \frac{\mathbb{E}_{\mathbf{b}\sim\mathbf{D}}\left(\sum_{i\in N} v_i(S_i(\mathbf{b}))\right)}{\sum_{i\in N} v_i(O_i)}$$

In the following sections we are going to see 2 simple auctions: Parallel 1st Price Auctions (P1A) and Parallel 2nd Price Auctions (P2A) and study their POA.

7.2 Parallel 1st Price Auctions (P1A)

In this section we are going for the first time see mechanisms that make bidders pay their bid. Formally, P1A is a simple mechanism where $p_j(\mathbf{b}) = \max_i b_{ij}$. In general, 1st price auctions are considered much more complicated than 2nd price auctions, because in first price auctions one needs to know information about the bids of the other bidders to bid successfully. However, in this setting, where we assume that a bidder knows the valuations of the other bidders, as well as the bid strategy that they are following, we can show that P1A do better that P2A. We will talk more about that at the next section, where we will compare the 2 types of auctions directly.

7.2.1 Price of Anarchy for Pure Equilibrium

First we are going to study PNE in P1A. Interestingly, PNE in P1A are always, no matter the valuation class, optimal: The output is always the optimal allocation.

Theorem 7.1. *Let $(N, M, \mathbf{v})$ be a combinatorial auction and let $\mathbf{b}$ be bids that form a PNE of P1A. Then the POA of $\mathbf{b}$ is always 1, i.e. the allocation $S_1(\mathbf{b}), ..., S_n(\mathbf{b})$ achieves optimal welfare.*

Proof. Fix a bidder i and an optimal allocation $O_1, ..., O_n$. We construct an alternative bid $\mathbf{b}'_i$ for him: Bid 0 for items not in O_i, and $p_j(\mathbf{b}) + \epsilon$ for items in O_i. Using this bidding strategy we see that i will receive only items in O_i. For simplicity we will omit the arguments in $S_i(\cdot)$ and $p_j(\cdot)$. Because we had a PNE, i should not gain anything by switching to $\mathbf{b}'_i$

$$v_i(S_i) - \sum_{j\in S_i} p_j \geq v_i(O_i) - \sum_{j\in O_i} (p_j + \epsilon) \tag{7.1}$$

By adding inequality 7.1 for all i we get

$$\sum_i v_i(S_i) - \sum_i \sum_{j \in S_i} p_j \geq \sum_i v_i(O_i) - \sum_i \sum_{j \in O_i} (p_j + \epsilon) \tag{7.2}$$

W.l.o.g we can assume that both allocations **S** and **O** leave no items unassigned. This means that in inequality 7.2 the term with the prices cancel each other out and by taking the ϵ term infinitesimally small we get

$$\sum_i v_i(S_i) \geq \sum_i v_i(O_i)$$

which proves the theorem. ■

The above theorem gives a very interesting corollary: The prices of the PNE form a walrasian equilibrium, because of inequality 7.1: By replacing O_i with an arbitrary set S we get that at prices $p_1, ..., p_m$ the set S_i is the demand of bidder i.

Corollary 7.1. *Let $(N, M, \mathbf{v})$ be a combinatorial auction and let* **b** *be bids that form a PNE of P1A. Then the prices $p_1(\mathbf{b}), ..., p_m(\mathbf{b})$ form a walrasian equilibrium.*

We can also notice that the opposite is true as well: Using the prices $p_1, .., p_m$ from a walrasian equilibrium we can construct a PNE by having each bidder bid for item j, either p_j if he receives j at the walrasian equilibrium, either infinitesimally less that p_j other wise.

Corollary 7.2. *Let $(N, M, \mathbf{v})$ be a combinatorial auction and let $p_1, ..., p_m$ be the prices of a walrasian equilibrium. Then the following bidding vectors form a PNE of P1A.*

$$b_{ij} = \begin{cases} p_j, & \text{if } j \text{ goes to } i \text{ in the walrasian equilibrium} \\ p_j - \epsilon, & \text{otherwise} \end{cases}$$

To conclude all that we saw, PNE in P1A achieve maximum welfare for all valuation classes, but have one major flaw: They exist only when walrasian equilibrium exist, which as we know are guaranteed only for gross-substitutes valuations. Under the scope of stability, the above also means that PNE become better only when stability guarantees the existence of walrasian equilibrium.

7.2.2 Price of Anarchy for MNE without Stability

Having talked about the more specific type of Nash Equilibrium, we are ready to move to a more general setting, Mixed Nash Equilibrium, where bidders are allowed randomized strategies. Unfortunately the welfare guarantees are not as good as in PNE. Specifically, there is an example where the POA of P1A is $1 - \frac{1}{e}$. Christodoulou et al. created an auction with OXS bidders, a class that contains unit-demand but is contained by gross-substitutes, where the POA goes to $1 - \frac{1}{e}$, as $n \to \infty$.

Theorem 7.2. *In an auction where bidders have OXS valuations, the POA of a MNE in a P1A can be $1 - \frac{1}{e} \approx 0.6321$.*

This means that for every valuation class more general than OXS, we can not guarantee POA larger than $1 - \frac{1}{e}$. Showing the proof for the above theorem, as well as simply presenting the example that Christodoulou et al. construct is far too complicated for the purpose of this thesis.

Having found an upper bound for the POA, we are now going to prove a lower bound. More specifically we are going to show that for XOS valuations (see definition 3.17) the POA of any MNE is at least $1 - \frac{1}{e}$, closing completely the gap between the lower and the upper bound.

Theorem 7.3 ([28]). *Let $(N, M, \mathbf{v})$ be a combinatorial auction with XOS valuations and $\mathbf{D} = (D_1, ..., D_n)$ a profile of bid vector distributions that forms a MNE for the P1A. Then the POA of $\mathbf{D} = (D_1, ..., D_n)$ is at least $1 - \frac{1}{e} \approx 0.6321$.*

Proof. Fix a bidder i and his valuation $v_i(\cdot)$. Let $q_i(\cdot)$ denote the additive valuation that represents i's valuation at set O_i. This means that $v_i(O_i) = q_i(O_i)$ and $v_i(S) \geq q_i(S)$ for every $S \subseteq M$. Thus by using $q_i(\cdot)$ instead of $v_i(\cdot)$ for bidder i, we can only underestimate his value.
We will construct a deviating bid $\mathbf{b}'_i$ for i. For every item $j \notin O_i$, $b'_{ij} = 0$. For every item $j \in O_i$, i is going to draw a bid b'_{ij} from a distribution with density function $f_j(x) = \frac{1}{q_i(j)-x}$ and support $[0, (1 - \frac{1}{e})q_i(j)]$.
Now consider an arbitrary realization of the players' bids $\mathbf{b} \sim \mathbf{D}$. We will focus on an arbitrary item $j \in O_i$, by underestimating i's valuation with $q_i(\cdot)$, meaning that if i bids x for item j and gets it, his utility will be $q_i(j) - x$. Thus, i's expected utility from item j by bidding as above is

$$u'_{ij} \geq \begin{cases} 0, & \textit{if } \max_{k \neq i} b_{kj} > (1 - \frac{1}{e})q_i(j) \\ \int_{\max_{k \neq i} b_{kj}}^{(1-\frac{1}{e})q_i(j)} (q_i(j) - x) f_j(x) dx = & \\ = (1 - \frac{1}{e})q_i(j) - \max_{k \neq i} b_{kj}, & \textit{otherwise} \end{cases} \tag{7.3}$$

By looking more closely at inequality 7.3, in the first case $0 > (1 - \frac{1}{e})q_i(j) - \max_{k \neq i} b_{kj}$. Thus, we can see that in both cases, $u'_{ij} \geq (1 - \frac{1}{e})q_i(j) - \max_{k \in N} b_{kj}$, simply by adding bidder i to the max term. Since this is a 1st price auction, $p_j(\mathbf{b}) = \max_{k \in N} b_{kj}$ and by adding the inequality for all items we get an underestimate of i's total expected utility at bid realizations $\mathbf{b}$

$$u'_i(\mathbf{b}) \geq \sum_{j \in O_i} u'_{ij} \geq (1 - \frac{1}{e}) \sum_{j \in O_i} q_i(j) - \sum_{j \in O_i} p_j(\mathbf{b}) = (1 - \frac{1}{e}) v_i(O_i) - \sum_{j \in O_i} p_j(\mathbf{b}) \tag{7.4}$$

Where the last equality holds because $q_i(O_i) = v_i(O_i)$. Now by taking the expectation over $\mathbf{b}$ of inequality 7.4, we get a lower bound of i's utility if he uses the deviation described at the start. Because $\mathbf{D}$ was a MNE, we have that his expected utility $u_i(\mathbf{D})$ before deviating will be higher than the deviating utility

$$u_i(\mathbf{D}) \geq \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathbf{D}} u'_i(\mathbf{b}) \geq (1 - \frac{1}{e}) v_i(O_i) - \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathbf{D}} \sum_{j \in O_i} p_j(\mathbf{b}) \tag{7.5}$$

By adding inequality 7.5 over all i and replacing each utility u_i with its expression

$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}}\left(\sum_i v_i(S_i(\mathbf{b})) - \sum_i \sum_{j\in S_i(\mathbf{b})} p_j(\mathbf{b})\right) \geq (1-\frac{1}{e})\sum_i v_i(O_i) - \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}}\sum_i\sum_{j\in O_i} p_j(\mathbf{b}) \quad (7.6)$$

By simplifying inequality 7.6

$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}}\left(\sum_i v_i(S_i(\mathbf{b}))\right) \geq (1-\frac{1}{e})\sum_i v_i(O_i)$$

which proves the theorem. ■

Thus we see that the Price of Anarchy in P1A always lies in $[0.6321, 1]$ and that the lower bound cannot be increased. Now we are going to see how this bounds are affected by stability guarantees in auctions.

7.2.3 Price of Anarchy for MNE with Stability

In this section we are going to see how the results of the previous section can be improved when we assume that the auction is stable. We have proven in *corollary* 5.1 that if $\gamma \geq 2$ then the bidder with the highest singleton value for an item is the one who gets that item in the optimal allocation. From now on we are going to denote with w_j the maximum of the singleton values of the bidders that do not get item j at the optimal allocation, i.e. if $j \in O_i$, then $w_j = \max_{k\neq i} v_k(j)$.
Because in stable auctions the singleton of the bidders are very important, we would like to prove that in a P1A no bidder is going to bid higher than w_j for item j. However this is not the case. Consider a really simple example with 2 bidders and 1 item. The first bidder has value 1 and the second 2. A PNE is for the first bidder to bid 1.5 and the second bidder to bid $1.5 + \epsilon$.
Since the above does not hold, we are going to prove something a little bit more convoluted: A bidder will bid higher that his singleton value only if he is sure that he is not going to receive that item.

Claim 7.1. *Let $(N, M, \mathbf{v})$ be a combinatorial auction with subadditive valuations and $\mathbf{D} = (D_1, ..., D_n)$ a profile of bid vector distributions that forms a MNE for P1A. Then for every $j \in M$ and any $i \in N$ the following entailment is true*

$$\textit{If } \mathop{\mathbb{P}}_{\mathbf{b}\sim\mathbf{D}}[j \in S_i(\mathbf{b})] > 0, \ \textit{then } \mathop{\mathbb{P}}_{\mathbf{b}\sim\mathbf{D}}[b_{ij} > v_i(j)] = 0$$

This claims that if a bidder gets a certain item with positive probability, then he, under no circumstances, bids higher than his singleton value for that item. The proof is quite easy, as we just need to prove that a bidder would only lose utility by bidding above his singleton value if he got that item.

Proof. Fix a bidder i and an item j. We will show this by contradiction: Suppose that at some realization of D_i bidder i bids higher than $v_i(j)$ for item j and that at some other realization of $\mathbf{D}$ bidder i gets item j. Since the distributions in $\mathbf{D}$ are independent, this means that there exists a realization of $\mathbf{D}$ where bidder i both gets j and bids for it higher than $v_i(j)$. Now we need to show that if bidder i lowered his bid to $v_i(j)$ he would gain utility.
In the realizations where i bids high for j and he does not get j, he has nothing to lose by lowering his price.
In the realizations where i bids high for j and gets j, if he lowered his price to $v_i(j)$ he would either lose the item, yielding a utility increase of $b_{ij} - v_i(j|S-j)$ or he would simple pay less, increasing his utility by $b_{ij} - v_i(j)$. Both quantities are strictly positive, the first because of subadditivity - $v_i(j|S-j) < v_i(j)$ -, meaning that $\mathbf{D}$ is not a MNE, which completes the contradiction. ■

Having proven the claim above, we are ready to show how the POA increases when the auction is stable.

Theorem 7.4. *Let $(N, M, \mathbf{v})$ be a γ-stable auction, $\gamma \geq 2$, with XOS bidders. Let $\mathbf{D}$ be a profile of bid vector distributions which forms a MNE for P1A. Then the POA is strictly greater than $\frac{\gamma-2}{\gamma-1}$.*

The proof that follows is quite easy: As γ becomes larger, the value that a bidder gets from each item in his optimal set gets further away from the second highest singleton. This means that if we have him bid the second highest singleton for each item, his utility should be large enough to prove that the allocation of the equilibrium has high enough welfare.

Proof. Fix a bidder i. We are going to construct a deviating bid for i. For any items $j \notin O_i$, bid 0. Denote with A the subset of O_i that i gets with probability 1 at the MNE. For the items in A, keep the same bidding strategy and for the other items j in $O_i - A$ bid the second highest singleton $w_j + \epsilon$, i.e. $\max_{k \neq i} v_k(j)$. This bidding strategy gets i the whole bundle O_i. This is obvious for items in A. For items $j \in O_i - A$, player i only needs to outbid any other player k that has positive probability to get j. Because of claim 7.1 their bid is at most w_j, which means that our deviating bids achieve getting these items from them.
Before using our deviating bid let us analyze the expected payment of i at the equilibrium:

$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} \sum_{j \in S_i(\mathbf{b})} p_j(\mathbf{b}) = \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} \sum_{j \in S_i(\mathbf{b})-A} p_j(\mathbf{b}) + \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} \sum_{j \in A} p_j(\mathbf{b}) \geq \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} \sum_{j \in A} p_j(\mathbf{b}) \tag{7.7}$$

Where the equality holds because for every realization of bids $\mathbf{b}$, $A \subseteq S_i(\mathbf{b})$ and the inequality because the payments are always non-negative. Now we use the fact that if i uses the deviating bids, he is not going to lose any utility. For the payment at the equilibrium we use inequality 7.7 as it is a lower bound for the real payment of i

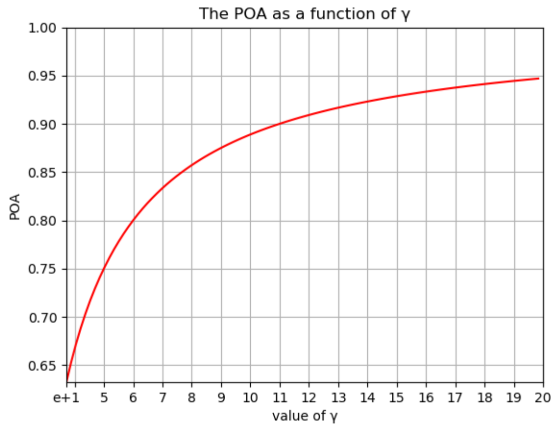
$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} v_i(S_i(\mathbf{b})) - \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} \sum_{j \in A} p_j(\mathbf{b}) \geq v_i(O_i) - \sum_{j \in O_i - A} w_j - \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} \sum_{j \in A} p_j(\mathbf{b}) \tag{7.8}$$

Now by rearranging and using corollary 5.2 ($w_j < \frac{1}{\gamma-1} q_i(j)$) on inequality 7.8 we get

$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathbf{D}} v_i(S_i(\mathbf{b})) > v_i(O_i) - \frac{1}{\gamma-1}\sum_{j\in O_i - A} q_i(j) \geq v_i(O_i) - \frac{1}{\gamma-1}\sum_{j\in O_i} q_i(j) \tag{7.9}$$

Where in the second inequality we simply make the sum contain more items. Using now the fact that $v_i(O_i) = \sum_{j\in O_i} q_i(j)$ and by adding inequality 7.9 for all i, we complete the proof of the theorem. ■

This theorem does not really say anything, unless $\gamma > e+1$, where the POA is guaranteed above $1 + \frac{1}{e}$. Asymptotically the POA becomes equal to 1. We also provide a graph to show how fast is the growth of the POA.



7.3 Parallel 2nd Price Auctions (P2A)

In this section we are going to study *Parallel 2nd Price Auctions* (P2A), where the price of each item j is the second highest bid, i.e. if $j \in S_i(\mathbf{b})$ then $p_j(\mathbf{b}) = \max_{k\neq i} b_{kj}$. Traditionally, second price auctions are considered better, because it is easier for bidders to make their bids, especially when they don't know the information about the other players. However, as we are going to see now, first price auctions provide better guarantees for the POA. In P2A, we are going to restrict ourselves only to PNE, as the bidders have a lot of freedom, even at this restricted setting. As we did in the previous section, first we are going to see P2A, without any assumptions of stability.

7.3.1 Price of Anarchy without Stability

It has been proven by Vickrey in [1] that 2nd price auctions for single parameter settings are awesome in a sense. However, if 2nd price auctions are left totally unrestricted they provide no POA guarantees whatsoever. As the following example demonstrates, 2nd price auctions, even for 1 item, can have unbounded POA.

Example 7.1. Consider 2 bidders and 1 item. Alice has value 1 for the item and Bob has value $\epsilon < 1$. The optimal allocation has welfare 1. However, the bids 0 and 100, for Alice and Bob respectively, forms a PNE: If Alice overbids Bob she will have utility of

-99, while Bob couldn't be happier since he gets the item for free. This PNE has POA equal to ϵ, which can be arbitrarily close to 0.

The problem with this example is that bidders with low value can bid arbitrarily high, making other bidders with high values uninterested in the item that they would have wanted. In order to achieve some kind of lower bound in the POA, we need to set some no-overbidding rules.

First we are going to see the strongest assumption of no-overbidding, that forces each player to bid below his value, no matter the outcome of the P2A.

Definition 7.5 (Strong No Overbidding (SNO))**.** A bid profile $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_n)$ satisfies SNO if for every bidder i

$$v_i(S) \geq \sum_{j \in S} b_{ij} \tag{7.10}$$

for every bundle of items $S \subseteq M$.

Next we see a weaker no-overbidding assumption, that forces each player to bid below his value only for the bundle that he will receive.

Definition 7.6 (Weak No Overbidding (WNO))**.** A bid profile $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_n)$ satisfies WNO if for every bidder i

$$v_i(S_i(\mathbf{b})) \geq \sum_{j \in S_i(\mathbf{b})} b_{ij} \tag{7.11}$$

for every bundle of items $S \subseteq M$.

Having defined our two notions of no-overbidding we are ready to explore what values takes the POA in P2A. We will begin with an example, that shows that POA can be $\frac{1}{2}$.

Example 7.2 (POA in P2A can be $\frac{1}{2}$)**.** Consider 2 unit-demand bidders, Alice and Bob and 2 items a and b. Alice's values for a and b are 2 and 1, respectively. Bob is symmetrical, meaning his values are 1 and 2 instead. We can conclude that the optimal welfare is 4.

Consider now that Alice's bids for a and b are 0 and 1, while Bob's are 1 and 0. These bids actually form an equilibrium. Since everything is symmetrical we will only analyze Alice's point of view. Alice has no reason to change her to bid for item b, since she gets it for free, for a total utility of 1. If she decides to bid for item a, her total value will increase to 2, but she will also be required to pay 1, bringing her total utility again to 1. Thus we have concluded that Alice can not gain any utility by deviating. Thus we have an equilibrium with welfare of 2, making POA equal to $\frac{1}{2}$.

Notice that both the bids of the equilibrium, as well as the deviating bids conform to SNO.

Now our hope is to prove that this example is the worst case scenario and that POA in P2A is always above $\frac{1}{2}$. This is indeed the case. Surprisingly we do not have to restrict ourselves only to bidding profiles that conform to SNO, as the assumption of WNO is enough.

Theorem 7.5 ([31]). *Let $(N, M, \mathbf{v})$ be a combinatorial auction with XOS valuations and $\mathbf{b}$ a bidding profile that forms a PNE for P2A. Then the POA of $\mathbf{b}$ is at least $\frac{1}{2}$.*

The proof that follows is quite simple. If each bidder for the set that he gets allocated in the optimal allocation bids high enough, then he will pay the bids of the other players, which in turn because of WNO are most the welfare of the equilibrium.

Proof. Fix a bidder i and denote with $q_i(\cdot)$ the additive valuation that represents $v_i(O_i)$. Because we have XOS, this means that $v_i(O_i) = q_i(O_i)$ and $v_i(S) \geq q_i(S)$ for any $S \subseteq M$. We will create a deviating bid for i: Bid 0 for items not in O_i and for items $j \in O_i$ bid $b'_{ij} = q_i(j)$. Then i's utility by deviating in such a way would be

$$\begin{aligned} u_i(\mathbf{b}_i, \mathbf{b}_{-i}) &= v_i(S_i(\mathbf{b}'_i, \mathbf{b}_{-i})) - \sum_{j \in S_i(\mathbf{b}'_i, \mathbf{b}_{-i})} \max_{k \neq i} b_{kj} \\ &\geq \sum_{j \in S_i(\mathbf{b}'_i, \mathbf{b}_{-i})} \left(q_i(j) - \max_k b_{kj} \right) \\ &\geq \sum_{j \in O_i} \left(q_i(j) - \max_k b_{kj} \right) = v_i(O_i) - \sum_{j \in O_i} \max_k b_{kj} \end{aligned}$$

Where in the first inequality we put more terms in the max term and used that $v_i(S_i) \geq q_i(S_i)$ and in the second we added the terms $q_i(j) - \max_k b_{kj}$ for the items that i did not get, because such terms are negative. Finally the equality holds because $v_i(O_i) = q_i(O_i)$. Now we can combine this inequality with the fact that $u_i(\mathbf{b}_i, \mathbf{b}_{-i})$ is not going to be more than the utility of i at the equilibrium, which in turn is less than the value of i's bundle at the equilibrium

$$v_i(S_i(\mathbf{b})) \geq u_i(\mathbf{b}) \geq u_i(\mathbf{b}_i, \mathbf{b}_{-i}) \geq v_i(O_i) - \sum_{j \in O_i} \max_k b_{kj} \tag{7.12}$$

Now by adding inequality 7.12 for all i and using the fact that $\cup_i S_i = \cup_i O_i$ (because we can suppose that both allocate all items), we get

$$\sum_i v_i(S_i(\mathbf{b})) \geq \sum_i v_i(O_i) - \sum_i \sum_{j \in S_i(\mathbf{b})} \max_k b_{kj} \tag{7.13}$$

Now because of WNO we know that $\sum_{j \in S_i(\mathbf{b})} \max_k b_{kj} \leq v_i(S_i(\mathbf{b}))$. Using this in inequality 7.13 and by rearranging we prove the theorem. ∎

This concludes the study of Price of Anarchy in Parallel 2nd Price Auctions, as we have found both the lower and the upper bound. Next we are going to study P2As under stability conditions, where we will show that things are a lot more interesting.

7.3.2 Price of Anarchy with Stability

In this section we are going to study P2A under the assumption of stability. First we would like to find some bound to show how good can we expect our results to be. Let's look at example 7.2, in which POA was $\frac{1}{2}$. Notice that this is the same example as example 5.1, where we showed that this auction is $(3-\epsilon)$-stable. This means that we cannot hope to show better bounds than the previous section when γ is less than 3.

One idea is to make adjustments to theorem 7.4 and show that POA in P2A is also greater than $\frac{\gamma-2}{\gamma-1}$. This makes sense as for $\gamma = 3$ we get the bound of $\frac{1}{2}$, of the previous auction. This idea indeed works with minor adjustments to the proof of theorem 7.4, but the reality is that we can prove something much stronger. For 3-stable auctions, POA is always 1, meaning that the only allocation that achieves an equilibrium is the optimal one. To prove this we are going to assume a different overbidding method, that allows bidders more freedom than SNO. We call this *Singleton No-Overbidding (SiNO)* and it simply states that each bidder should not bid more for each item than his singleton value.

Definition 7.7 (Singleton No Overbidding (SiNO))**.** A bid profile $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_n)$ satisfies *Singleton No Overbidding (SiNO)* if for every bidder i and item j

$$v_i(j) \geq b_{ij} \tag{7.14}$$

Just for concreteness let us prove the fact that SiNO allows bidders more bidding options than SNO.

Claim 7.2 (SNO $\subseteq$ SiNO)**.** *If a bidding profile* $\mathbf{b}$ *satisfies SNO, then it also satisfies SiNO.*

Proof. Fix a bidder i and an item j. Since $\mathbf{b}_i$ satisfies SNO, then for the bundle $S = \{j\}$ we have that $v_i(j) \geq b_{ij}$. This completes the proof. ■

Now we are ready to properly state that with SiNO, POA is always 1.

Theorem 7.6. *Let* $(N, M, \mathbf{v})$ *be a* 3*-stable combinatorial auction with submodular valuations and* $\mathbf{b}$ *a bidding profile that forms a PNE for P2A and satisfies SiNO. Then the allocation of the equilibrium is the optimal allocation.*

The proof that follows is similar to the proof of theorem 6.9. Intuitively, because the gap between the prices and the values has gotten large enough, the bidder who is optimally allocated an item has much more incentive to outbid the other bidders for that item.

Proof. We will show this by contradiction. First, denote with w_j the second highest singleton value of item j, i.e. if $j \in O_i$, $w_j = \max_{k \neq i} v_k(j)$ (we know that i has the highest singleton because of corollary 5.1). Fix a bidder i for who $O_i \not\subseteq S_i(\mathbf{b})$. Let A_i be the items that i is allocated in the equilibrium that are also in O_i, i.e. $A_i = O_i \cap S_i(\mathbf{b})$. We construct a deviating bid for i: For items not in O_i, bid 0, for items A_i bid as before (which means that the prices for these items will be the same as before) and for items in $O_i - A_i$ bid $w_j + \epsilon$. This bid vector guarantees that i will receive the whole bundle O_i, because of SiNO. His new utility is

$$u_i' = v_i(O_i) - \sum_{j \in A_i} p_j(\mathbf{b}) - \sum_{j \in O_i - A_i} \max_{k \neq i} b_{kj} \geq v_i(O_i) - \sum_{j \in A_i} p_j(\mathbf{b}) - \sum_{j \in O_i - A_i} w_j \tag{7.15}$$

where the inequality holds because of SiNO: Every bid must be below every corresponding singleton value, which in turn is less than the maximum of the singleton values. Now we use that this new utility can't be greater than the utility of the bidder at the equilibrium (for simplicity we denote $S_i = S_i(\mathbf{b})$)

$$v_i(S_i) - \sum_{j\in S_i} p_j(\mathbf{b}) \geq v_i(O_i) - \sum_{j\in A_i} p_j(\mathbf{b}) - \sum_{j\in O_i - A_i} w_j \tag{7.16}$$

In inequality 7.16 we can eliminate the term $\sum_{j\in A_i} p_j(\mathbf{b})$ because it is contained in the term $\sum_{j\in S_i} p_j(\mathbf{b})$. Also we can ignore the rest of the prices as they are positive. All these make inequality 7.16

$$v_i(S_i) \geq v_i(O_i) - \sum_{j\in O_i - A_i} w_j \tag{7.17}$$

Now using the facts that $\sum_{j\in S_i - A_i} v_i(j) + v_i(A_i) \geq v_i(S_i)$ and $w_j \geq v_i(j)$ for items such that $j \in S_i - A_i$, inequality 7.17 becomes

$$\sum_{j\in S_i - A_i} w_j + \sum_{j\in O_i - A_i} w_j \geq v_i(O_i) - v_i(A_i) \tag{7.18}$$

Using the fact that $\cup_i(S_i - A_i) = M - \cup_i A_i = \cup_i(O_i - A_i)$ and adding inequality 7.18 for all i and we get

$$2\sum_i \sum_{j\in O_i - A_i} w_j \geq \sum_i \big(v_i(O_i) - v_i(A_i)\big) \tag{7.19}$$

Because of lemma 5.1 and 3-stability, for any item $j \in O_i$ we have that $v_i(j|O_i - j) > \frac{1}{2}w_j$. This makes *inequality* 7.19

$$\sum_i \sum_{j\in O_i - A_i} \big(v_i(O_i) - v_i(O_i - j)\big) > \sum_i \big(v_i(O_i) - v_i(A_i)\big) \tag{7.20}$$

Now if for every i, $\sum_{j\in O_i - A_i} \big(v_i(O_i) - v_i(O_i - j)\big) \leq v_i(O_i) - v_i(A_i)$, inequality 7.20 is a contradiction. The latter inequality is true because of submodularity: W.l.o.g. we order items in $O_i - A_i = \{1, 2, 3, ..., |O_i - A_i|\}$ and denote $X_j = \{1, 2, ..., j\}$, where $X_0 = \emptyset$ and $X_{|O_i - A_i|} = O_i - A_i$. This makes the RHS in the inequality that we want to prove equal to

$$v_i(O_i) - v_i(A_i) = \sum_{j=1}^{|O_i - A_i|} \big(v_i(O_i - X_{j-1}) - v_i(O_i - X_j)\big) \tag{7.21}$$

Because of submodularity $v_i(O_i - X_{j-1}) - v_i(O_i - X_j) \geq v_i(O_i) - v_i(O_i - j)$, which completes the proof. ■

This completes the section of POA, having proven that without stability guarantees P1A are better that P2A, but stable P2A are better than stable P1A, when the auction is at least 3-stable.

Bibliography

[1] William Vickrey. "Counterspeculation, Auctions, and Competitive Sealed Tenders". In: *The Journal of Finance* 16.1 (1961), pp. 8–37. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2977633.

[2] Victor Klee and George J. Minty. "How Goog Is The Simplex Algorithm". In: 1970.

[3] Edward H. Clarke. "Multipart pricing of public goods". In: *Public Choice* 11.1 (Sept. 1971), pp. 17–33. ISSN: 1573-7101. DOI: 10.1007/BF01726210. URL: https://doi.org/10.1007/BF01726210.

[4] Theodore Groves. "Incentives in Teams". In: *Econometrica* 41.4 (1973), pp. 617–631. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/1914085.

[5] Richard Thaler. "Toward a positive theory of consumer choice". In: *Journal of Economic Behavior and Organization* 1.1 (1980), pp. 39–60. URL: https://EconPapers.repec.org/RePEc:eee:jeborg:v:1:y:1980:i:1:p:39-60.

[6] Roger B. Myerson. "Optimal Auction Design". In: *Mathematics of Operations Research* 6.1 (1981), pp. 58–73. ISSN: 0364765X, 15265471. URL: http://www.jstor.org/stable/3689266.

[7] A. Kelso, P. Crawford, and Vincent Crawford. "Job matching, coalition formation, and gross substitutes". In: *Econometrica* 50 (Jan. 1982), pp. 1483–1504. DOI: 10.2307/1913392.

[8] Benny Lehmann, Daniel Lehmann, and Noam Nisan. "Combinatorial Auctions with Decreasing Marginal Utilities". In: *Proceedings of the 3rd ACM Conference on Electronic Commerce*. EC '01. Tampa, Florida, USA: ACM, 2001, pp. 18–28. ISBN: 1-58113-387-1. DOI: 10.1145/501158.501161. URL: http://doi.acm.org/10.1145/501158.501161.

[9] Christos Papadimitriou. "Algorithms, Games, and the Internet". In: *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*. STOC '01. Hersonissos, Greece: ACM, 2001, pp. 749–753. ISBN: 1-58113-349-9. DOI: 10.1145/380752.380883. URL: http://doi.acm.org/10.1145/380752.380883.

[10] Tim Roughgarden. "The Price of Anarchy is Independent of the Network Topology". In: *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*. STOC '02. Montreal, Quebec, Canada: ACM, 2002, pp. 428–437. ISBN: 1-58113-495-9. DOI: 10.1145/509907.509971. URL: http://doi.acm.org/10.1145/509907.509971.

[11] Daniel A. Spielman and Shang-Hua Teng. “Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time”. In: *J. ACM* 51.3 (May 2004), pp. 385–463. ISSN: 0004-5411. DOI: 10.1145/990308.990310. URL: http://doi.acm.org/10.1145/990308.990310.

[12] Subhash Khot et al. “Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions”. In: *Proceedings of the First International Conference on Internet and Network Economics*. WINE'05. Hong Kong, China: Springer-Verlag, 2005, pp. 92–101. ISBN: 3-540-30900-4, 978-3-540-30900-0. DOI: 10.1007/11600930_10. URL: https://doi.org/10.1007/11600930_10.

[13] Shahar Dobzinski, Noam Nisan, and Michael Schapira. “Truthful Randomized Mechanisms for Combinatorial Auctions”. In: *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*. STOC '06. Seattle, WA, USA: ACM, 2006, pp. 644–652. ISBN: 1-59593-134-1. DOI: 10.1145/1132516.1132607. URL: http://doi.acm.org/10.1145/1132516.1132607.

[14] Shahar Dobzinski and Michael Schapira. “An Improved Approximation Algorithm for Combinatorial Auctions with Submodular Bidders”. In: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*. SODA '06. Miami, Florida: Society for Industrial and Applied Mathematics, 2006, pp. 1064–1073. ISBN: 0-89871-605-5. URL: http://dl.acm.org/citation.cfm?id=1109557.1109675.

[15] U. Feige and J. Vondrak. “Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e”. In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. Oct. 2006, pp. 667–676. DOI: 10.1109/FOCS.2006.14.

[16] Uriel Feige. “On Maximizing Welfare when Utility Functions Are Subadditive”. In: *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*. STOC '06. Seattle, WA, USA: ACM, 2006, pp. 41–50. ISBN: 1-59593-134-1. DOI: 10.1145/1132516.1132523. URL: http://doi.acm.org/10.1145/1132516.1132523.

[17] Shahar Dobzinski. “Two Randomized Mechanisms for Combinatorial Auctions”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Ed. by Moses Charikar et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 89–103. ISBN: 978-3-540-74208-1.

[18] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. “Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords”. In: *American Economic Review* 97.1 (Mar. 2007), pp. 242–259. DOI: 10.1257/aer.97.1.242. URL: http://www.aeaweb.org/articles?id=10.1257/aer.97.1.242.

[19] Noam Nisan et al. *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007. ISBN: 0521872820.

[20] Jan Vondrak. “Optimal Approximation for the Submodular Welfare Problem in the Value Oracle Model”. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. Victoria, British Columbia, Canada: ACM, 2008, pp. 67–74. ISBN: 978-1-60558-047-0. DOI: 10.1145/1374376.1374389. URL: http://doi.acm.org/10.1145/1374376.1374389.

[21] Shahar Dobzinski, Noam Nisan, and Michael Schapira. "Approximation Algorithms for Combinatorial Auctions with Complement-Free Bidders". In: *Mathematics of Operations Research* 35.1 (2010), pp. 1–13. ISSN: 0364765X, 15265471. URL: http://www.jstor.org/stable/40538414.

[22] Shahar Dobzinski. "An Impossibility Result for Truthful Combinatorial Auctions with Submodular Valuations". In: *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*. STOC '11. San Jose, California, USA: ACM, 2011, pp. 139–148. ISBN: 978-1-4503-0691-1. DOI: 10.1145/1993636.1993656. URL: http://doi.acm.org/10.1145/1993636.1993656.

[23] Robert Elsässer and Tobias Tscheuschner. "Settling the Complexity of Local Max-Cut (Almost) Completely". In: *Automata, Languages and Programming*. Ed. by Luca Aceto, Monika Henzinger, and Jiří Sgall. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 171–182. ISBN: 978-3-642-22006-7.

[24] Shahar Dobzinski and Jan Vondrak. "The Computational Complexity of Truthfulness in Combinatorial Auctions". In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. EC '12. Valencia, Spain: ACM, 2012, pp. 405–422. ISBN: 978-1-4503-1415-2. DOI: 10.1145/2229012.2229044. URL: http://doi.acm.org/10.1145/2229012.2229044.

[25] Hu Fu, Robert Kleinberg, and Ron Lavi. "Conditional Equilibrium Outcomes via Ascending Price Processes with Applications to Combinatorial Auctions with Item Bidding". In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. EC '12. Valencia, Spain: ACM, 2012, pp. 586–586. ISBN: 978-1-4503-1415-2. DOI: 10.1145/2229012.2229055. URL: http://doi.acm.org/10.1145/2229012.2229055.

[26] Piotr Krysta and Berthold Vöcking. "Online Mechanism Design (Randomized Rounding on the Fly)". In: *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part II*. ICALP'12. Warwick, UK: Springer-Verlag, 2012, pp. 636–647. ISBN: 978-3-642-31584-8. DOI: 10.1007/978-3-642-31585-5_56. URL: http://dx.doi.org/10.1007/978-3-642-31585-5_56.

[27] Michal Feldman et al. "Simultaneous Auctions Are (Almost) Efficient". In: *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*. STOC '13. Palo Alto, California, USA: ACM, 2013, pp. 201–210. ISBN: 978-1-4503-2029-0. DOI: 10.1145/2488608.2488634. URL: http://doi.acm.org/10.1145/2488608.2488634.

[28] Vasilis Syrgkanis and Eva Tardos. "Composable and Efficient Mechanisms". In: *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*. STOC '13. Palo Alto, California, USA: ACM, 2013, pp. 211–220. ISBN: 978-1-4503-2029-0. DOI: 10.1145/2488608.2488635. URL: http://doi.acm.org/10.1145/2488608.2488635.

[29] Matthias Englert, Heiko Röglin, and Berthold Vöcking. "Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP". In: *Algorithmica* 68.1 (Jan. 2014), pp. 190–264. ISSN: 1432-0541. DOI: 10.1007/s00453-013-9801-4. URL: https://doi.org/10.1007/s00453-013-9801-4.

[30] Konstantin Makarychev, Yury Makarychev, and Aravindan 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 '14. Portland, Oregon: Society for Industrial and Applied Mathematics, 2014, pp. 890–906. ISBN: 978-1-611973-38-9. URL: http://dl.acm.org/citation.cfm?id=2634074.2634141.

[31] George Christodoulou, Annamária Kovács, and Michael Schapira. "Bayesian Combinatorial Auctions". In: *J. ACM* 63.2 (Apr. 2016), 11:1–11:19. ISSN: 0004-5411. DOI: 10.1145/2835172. URL: http://doi.acm.org/10.1145/2835172.

[32] George Christodoulou et al. "Tight Bounds for the Price of Anarchy of Simultaneous First-Price Auctions". In: *ACM Trans. Econ. Comput.* 4.2 (Jan. 2016), 9:1–9:33. ISSN: 2167-8375. DOI: 10.1145/2847520. URL: http://doi.acm.org/10.1145/2847520.

[33] Shahar Dobzinski. "Breaking the Logarithmic Barrier for Truthful Combinatorial Auctions with Submodular Bidders". In: *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*. STOC '16. Cambridge, MA, USA: ACM, 2016, pp. 940–948. ISBN: 978-1-4503-4132-5. DOI: 10.1145/2897518.2897569. URL: http://doi.acm.org/10.1145/2897518.2897569.

[34] Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. "Algorithms for Stable and Perturbation-resilient Problems". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2017. Montreal, Canada: ACM, 2017, pp. 438–451. ISBN: 978-1-4503-4528-6. DOI: 10.1145/3055399.3055487. URL: http://doi.acm.org/10.1145/3055399.3055487.

[35] Michael Etscheid and Heiko Röglin. "Smoothed Analysis of Local Search for the Maximum-Cut Problem". In: *ACM Trans. Algorithms* 13.2 (Mar. 2017), 25:1–25:12. ISSN: 1549-6325. DOI: 10.1145/3011870. URL: http://doi.acm.org/10.1145/3011870.

[36] Tim Roughgarden, Vasilis Syrgkanis, and Éva Tardos. "The Price of Anarchy in Auctions". In: *J. Artif. Int. Res.* 59.1 (May 2017), pp. 59–101. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=3176788.3176790.

[37] Moshe Babaioff, Shahar Dobzinski, and Sigal Oren. "Combinatorial Auctions with Endowment Effect". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. EC '18. Ithaca, NY, USA: ACM, 2018, pp. 73–90. ISBN: 978-1-4503-5829-3. DOI: 10.1145/3219166.3219197. URL: http://doi.acm.org/10.1145/3219166.3219197.

[38] Daniel Dadush and Sophie Huiberts. "A Friendly Smoothed Analysis of the Simplex Method". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2018. Los Angeles, CA, USA: ACM, 2018, pp. 390–403. ISBN: 978-1-4503-5559-9. DOI: 10.1145/3188745.3188826. URL: http://doi.acm.org/10.1145/3188745.3188826.

[39] Tomer Ezra, Michal Feldman, and Ophir Friedler. "A General Framework for Endowment Effects in Combinatorial Markets". In: *CoRR* abs/1903.11360 (2019). arXiv: 1903.11360. URL: http://arxiv.org/abs/1903.11360.