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

Προσεγγίσεις της Ωφέλειας Κανονικοποιημένης με Βάση τη Ρευστότητα σε Συνδυαστικές Δημοπρασίες

Διπλωματική Εργασία

της

Ποδηματά Χαρίκλειας

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

Εργαστήριο Λογικής και Επιστήμης Υπολογισμών
Αθήνα, Φεβρουάριος 2016

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

Προσεγγίσεις της Ωφέλειας Κανονικοποιημένης με Βάση τη Ρευστότητα σε Συνδυαστικές Δημοπρασίες

Διπλωματική Εργασία

της

Ποδηματά Χαρίκλειας

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Φεβρουαρίου 2016.

....	....	....
Ευάγγελος Μαρκάκης	Άρης Παγουρτζής	Δημήτρης Φωτάκης
Επίκουρος Καθηγητής ΑΣΟΕΕ	Αναπληρωτής Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.

Εργαστήριο Λογικής και Επιστήμης Υπολογισμών
Αθήνα, Φεβρουάριος 2016

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

Περίληψη

Στην παρούσα Διπλωματική Εργασία μελετούμε φιλαλήθεις μηχανισμούς για Συνδυαστικές Δημοπρασίες, δημοπρασίες δηλαδή όπου διατίθενται πολλά αντικείμενα και οι παίκτες ενδιαφέρονται στο να αποκτήσουν κάποιο υποσύνολο αυτών, με τον επιπλέον περιορισμό ότι ο κάθε παίκτης έχει πλέον κάποιον κρυφό περιορισμό ρευστότητας (budget). Στόχος μας ήταν να μελετήσουμε τη συμπεριφορά μίας σχετικά πρόσφατα εισαχθείσας αντικειμενικής συνάρτηση στη διεθνή βιβλιογραφία: τη κοινωνική ωφέλεια υπό περιορισμούς ρευστότητας (Liquid Welfare). Το Liquid Welfare είναι ένα μέτρο που προτάθηκε από τους Dobzinski και Paes Leme και καταφέρνει να εξισορροπήσει την θέληση ενός παίκτη να αγοράσει ένα αντικείμενο και την ικανότητά του να το αγοράσει (λόγω των περιορισμών ρευστότητας). Για την ακρίβεια, το Liquid Welfare είναι το ελάχιστο μεταξύ της αξίας που έχει ένας παίκτης για κάποια ανάθεση αντικειμένων και του budget του παίκτη αυτού. Το ενδιαφέρον μας εστιάζεται στο να επεκτείνουμε και να βελτιώσουμε πρόσφατα αποτελέσματα της προσέγγισης που δίνει το Liquid Welfare για δημοπρασίες όπου έχουμε ένα διαιρέσιμο αγαθό και για submodular συνδυαστικές δημοπρασίες. Διαισθητικά, συνδυάζουμε ιδέες από τον γνωστό αλγόριθμο των Πολλαπλασιαστικών Συντελεστών του πεδίου της Μηχανικής Εκμάθησης, που μπορούν να παράξουν φιλαλήθεις μηχανισμούς, καθώς και ιδέες από τη δουλειά των Dobzinski και Paes Leme, οι οποίοι καταφέρνουν να σχεδιάσουν μηχανισμούς για την περίπτωση όπου έχουμε παίκτες με budgets.

Για τους σκοπούς αυτής της διπλωματικής εργασίας πραγματοποιούμε εκτενή ανάλυση της σχετικής βιβλιογραφίας και παρουσιάζουμε στο τέλος τα δικά μας συμπεράσματα καθώς και προτάσεις για μελλοντική εργασία.

Λέξεις Κλειδιά – Φιλαλήθεις Μηχανισμοί, Combinatorial Auctions, Liquid Welfare, Multiplicative Weights Update

Abstract

In this diploma thesis we study truthful mechanisms for Combinatorial Auctions; auctions where they are many items and the bidders are interested in getting a subset of these items (bundles), with the additional constraint that each bidder has a private budget. Our goal was to study the behavior of a recently proposed new objective function: the Liquid Welfare function. Liquid Welfare is a metric which was proposed by Dobzinski and Paes Leme and manages to balance between the willingness of a user to pay for an item and his/her ability to pay for it, due to budget constraints. In other words, Liquid Welfare is the minimum between the bidder's valuation for a certain allocation and the budget of this bidder. We focus our interests in extending and improving recent results of the approximation of Liquid Welfare for single item multi unit auctions and submodular combinatorial auctions. Intuitively, we combine ideas from the known Multiplicative Weights Update Method from the field of Machine Learning, which can produce truthful mechanisms together with ideas from the work of Dobzinski and Paes Leme, who manage to design mechanisms for budgeted settings.

For the purposes of this diploma thesis, we study and analyze extensively the relevant bibliography and we present, at the end, our conclusions together with suggestions for future work.

Keywords – Truthful Mechanisms, Combinatorial Auctions, Liquid Welfare, Multiplicative Weights Update Method

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κύριο Ζάχο, τον κύριο Παγουρτζή και τον κύριο Φωτάκη, οι οποίοι ως οι καθηγητές του Εργαστηρίου Λογικής και Επιστήμης των Υπολογιστών μου γνώρισαν ένα μεγάλο φάσμα ερευνητικών πεδίων της Θεωρητικής Πληροφορικής.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον κύριο Φωτάκη. Τον κύριο Φωτάκη τον γνώρισα ήδη από το πρώτο έτος. Στο τέλος αυτής της διπλωματικής εργασίας, που σημαίνει και το τέλος των προπτυχιακών μου σπουδών, θέλω να τον ευχαριστήσω που μου κέντρισε το ενδιαφέρον για τομείς της Επιστήμης των Υπολογιστών και των Αλγορίθμων, των οποίων αγνοούσα την ύπαρξη. Θέλω ακόμη να τον ευχαριστήσω γιατί από το τέταρτο εξάμηνο ήταν πάντοτε πολύ υποστηρικτικός και πίστευε σε εμένα από την πρώτη στιγμή. Μου έβαλε τις βάσεις για τον ερευνητικό τρόπο σκέψης και με καθοδήγησε ως μέντορας για δύο (ίσως και παραπάνω) χρόνια.

Θέλω επίσης να ευχαριστήσω τα παιδιά του Εργαστήριου Λογικής και Επιστήμης των Υπολογιστών για το κλίμα συνεργασίας και το περιβάλλον εργασίας που μου προσέφεραν.

Ιδιαίτερη μνεία θα ήθελα να κάνω στους φίλους μου που με ανέχονται και με αντέχουν τόσα χρόνια. Σας ευχαριστώ για τις στιγμές χαλάρωσης που μου έχετε χαρίσει που μου έχετε χαρίσει: χωρίς αυτές θα ήμουν πολύ διαφορετική.

Ένα μεγάλο ευχαριστώ θα ήθελα να πω και στην οικογένεια μου, στη μαμά μου τη Ρόζα, το μπαμπά μου το Θωμά και την αδερφή μου τη Στέλλα που είναι πάντοτε δίπλα μου σε ό,τι και αν κάνω. Τους ευχαριστώ γιατί όσο περνάνε τα χρόνια η στήριξή τους και η πίστη τους σε εμένα μεγαλώνει.

Τέλος, ευχαριστώ τις δύο μου γάτες, τη Νάλα και τον Ιάκωβο, οι οποίες ακόμη και όταν απλά κοιμούνται δίπλα μου, μου δίνουν αρκετή δύναμη να συνεχίσω να προσπαθώ, παρά την κούρασή μου.

Χαρά Ποδηματά

Περιεχόμενα

List of Algorithms

Κεφάλαιο 1

Εισαγωγή

1.1 Από τα Οικονομικά στους Αλγορίθμους

Για πολλές δεκαετίες, Οικονομολόγοι και Επιστήμονες των Υπολογιστών δεν είχαν καμία ιδιαίτερη διάδραση. Οι μεν σχεδιάζαν θεωρίες που δεν ήξεραν ποτέ που μπορεί να χρησιμοποιηθούν και οι δε, ακόμη και στην πιο θεωρητική τους ενασχόληση, ασχολούνταν με εφαρμόσιμες θεωρίες. Όλα αυτά άλλαξαν με την απαρχή του Internet. Τα τελευταία χρόνια έχει υπάρξει μία εκρηκτική άνοδος στην έρευνα που βρίσκεται στην τομή της Επιστήμης των Υπολογιστών και της Θεωρίας Παιγνίων, ενός τομέα μέχρι πρότινος αμιγώς Οικονομικού. Οι Επιστήμονες των Υπολογιστών χρησιμοποίησαν σε πρακτικά περιβάλλοντα ήδη γνωστά συμπεράσαματα της Θεωρίας Παιγνίων και έδωσαν νέα άνθηση στο πεδίο αυτό, με τις δικές τους συμβολές.

Το Internet, λοιπόν, δημιούργησε μία νέα οικονομία. Η υπολογιστική φύση του Internet επέτρεπε τη χρήση υπολογιστικών εργαλείων σε αυτή τη νέα αναδυόμενη οικονομία. Από την άλλη μεριά, το Internet το ίδιο είναι το αποτέλεσμα των δράσεων πολλών ανθρώπων. Και αυτό ακριβώς ήταν ένα πολύ καινούριο στοιχείο στην κλασσική «top-down» προσέγγιση του υπολογισμού, που χρησιμοποιούσαμε ως τότε. Έτσι, η Θεωρία Παιγνίων εμφανίστηκε σαν φυσικός τρόπος να δούμε το Internet και τις διαδράσεις μέσα σε αυτό: διαδράσεις τόσο ανθρώπινες, όσο και μηχανικές.

Η Θεωρία Παιγνίων μελετά τις ισορροπίες (equilibria), όπως για παράδειγμα η ισορροπία του Nash. Μία ισορροπία χαρακτηρίζεται ως η κατάσταση κατά την οποία κανένας παίκτης δεν έχει κίνητρο να αλλάξει τη στρατηγική του. Οι ισορροπίες μπορούν να βρεθούν σε διάφορα πεδία που σχετίζονται με το Internet, όπως για παράδειγμα στις οικονομικές συναλλαγές. Η Θεωρία Παιγνίων προμηθεύει τα εργαλεία για την ανάλυση των ισορροπιών και μία συνήθης προσέγγιση αμέσως μετά είναι να «βρούμε το παίγνιο», δηλαδή να περιγράψουμε φορμαλιστικά τις διαδράσεις στο Internet σαν ένα παίγνιο και στη συνέχεια να εξάγουμε τις ισορροπίες.

Το να επαναδιατυπώσουμε τα προβλήματα με όρους παιγνίων μας επιτρέπει την ανάλυση των διαδράσεων που στηρίζονται στο Internet καθώς και την κατασκευή μηχανισμών που ανταπεξέρχονται σε συγκεκριμένη ζήτηση. Εάν μία ισορροπία αποδειχθεί ότι υπάρχει, γεννάται ένα νέο ερώτημα: μπορεί αυτή η ισορροπία να βρεθεί σε κάποιο (λογικό) χρονικό διάστημα; Αυτό το ερώτημα οδηγεί στην ανάλυση των αλγορίθμων για την έρευση των ισορροπιών.

Κρατώντας όλα τα παραπάνω κατά νου, είναι εύκολο κανείς να συμπεράνει πώς η Αλγοριθμική Θεωρία Παιγνίων προσφέρει ένα πολύ μεγάλο πεδίο έρευνας, τα αποτελέσματα της οποίας τα συναντάμε στην καθημερίνοτητά μας (χωρίς, πολλές φορές, να το καταλαβαίνουμε). Παρακάτω σε αυτή την εργασία θα αναφερθούμε πιο στοχευμένα σε συγκεκριμένα «κομμάτια» αυτής της έρευνας.

1.2 Προσεγγιστικοί Αλγόριθμοι

Οι αλγόριθμοι που θα μας απασχολήσουν σε αυτή τη διπλωματική εργασία είναι αλγόριθμο *βελτιστοποίησης*. Τέτοιοι αλγόριθμοι έχουν μία *αντικειμενική συνάρτηση* (objective function), την οποία προσπαθούν να την μεγιστοποιήσουν ή να την ελαχιστοποιήσουν. Η τιμή της συνάρτησης αυτής στο τέλος του αλγορίθμου είναι το κόστος του αλγορίθμου. Συνήθως, δεν μπορούμε να υπολογίσουμε *ακριβώς* τη βέλτιστη λύση (την οποία συμβολίζουμε με OPT), αλλά κάποια προσέγγιση αυτής.

Ορισμός 1.1 (Προσεγγιστικός Αλγόριθμος)**.** *Προσεγγιστικός Αλγόριθμος* ονομάζεται ένας αλγόριθμος, ο οποίος βρίσκει μία εφικτή λύση για το πρόβλημα βελτιστοποίησης, της οποίας το κόστος είναι τουλάχιστον (ή το πολύ, ανάλογα με την αντικειμενική συνάρτηση) $\frac{1}{\rho}OPT$ (ή αντίστοιχα ρOPT) αν είναι πρόβλημα μεγιστοποίησης (και πρόβλημα ελαχιστοποίησης αντίστοιχα). Η παράμετρος ρ καλείται προσεγγιστικός λόγος και εξαρτάται συχνά από άλλες παραμέτρους της εισόδου.

Ένας άλλος τρόπος να αναπαριστούμε προβλήματα βελτιστοποίησης είναι μέσω ενός γραμμικού προγράμματος (LP). Ένα LP σε κανονική μορφή είναι του τύπου:

$$\begin{aligned} \max\ c^T x \\ Ax \leq b \\ x \geq 0 \end{aligned}$$

Και το σύνολο των εφικτών λύσεων σχηματίζει ένα *πολύεδρο*. Τα πιο χαρακτηριστικά προβλήματα είναι προβλήματα Ακέραιου Προγραμματισμού (IP), όπου το x, η εφικτή λύση δηλαδή, μπορεί να πάρει μόνο ακέραιες τιμές. Εφόσον τα IP είναι τα πιο δύσκολα επιλύσιμα προβλήματα, συνήθως λύνουμε το αντίστοιχο LP πρόβλημα με χαλαρωμένο τον τελευταίο περιορισμό και στη συνέχεια στρογγυλοποιούμε τη λύση που βρέθηκε (προκειμένου να την κάνουμε ακέραια) μέσω κάποιας randomized (τυχαιοποιημένης) διαδικασίας.

Ορισμός 1.2 (Shadow Price)**.** Ονομάζουμε *σκιώδη τιμή* (shadow price) ενός συγκεκριμένου περιορισμού την αλλαγή στη βέλτιστη λύση της αντικειμενικής συνάρτησης ανά μονάδα αύξησης της τιμής που βρίσκεται στο δεξί μέλος του περιορισμού αυτού, αν όλα τα υπόλοιπα δεδομένα του προβλήματος παραμείνουν τα ίδια.

Η ερμηνεία των shadow-prices ενός Προβλήματος Γραμμικού Προγραμματισμού είναι μία πολύ χρήσιμη έννοια. Πρώτον, αυτές οι shadow prices μας δίνουν άμεσα την *οριακή αξία* μίας επιπλέον μονάδας σε οποιοδήποτε από τα resources που έχουμε στο πρόβλημα. Δεύτερον, όταν μία «δράση» (στα πλαίσια του προβλήματος) έχει τιμολογηθεί χρησιμοποιώντας τις shadow prices, μπορούμε να καθορίσουμε το ευκαιριακό κόστος που προκύπτει αν όλες οι άλλες δράσεις είναι καθορισμένες. Η Δυϊκότητα στον Γραμμικό Προγραμματισμό (Duality in Linear Programming) είναι ουσιαστικά μία ενοποιητική θεωρία, η οποία ερευνά τις σχέσεις μεταξύ δύο ενός δεδομένου

Προβλήματος Γραμμικού Προγραμματισμού και ενός άλλου σχετικού Προβλήματος το οποίο εκφράζεται έχοντας ως μεταβλητές τις ίδιες τις shadow prices. Η σημασία της δυϊκότητας είναι διπλή. Πρώτα και κύρια, το να καταλάβουμε την ερμηνεία με τις shadow prices του αρχικού μας προβλήματος, μπορεί να αποδειχθεί πάρα πολύ χρήσιμη στο να καταλάβουμε εφαρμογές και επεκτάσεις ενός ολόκληρου μοντέλου γραμμικού προγραμματισμού. Δεύτερον, είναι συχνά δυνατό να λύσουμε το συσχετιζόμενο (Δυϊκό) πρόβλημα γραμμικού προγραμματισμού με τις shadow prices ως τις μεταβλητές, αντί για να επιλύσουμε το αρχικό πρόβλημα, χρησιμοποιώντας κάποια πιθανή καλύτερη υπολογιστική συμπεριφορά του Dual Linear Program. Η σημασία της δυϊκότητας γίνεται περισσότερο εμφανείς στις εφαρμογές που ακολουθούν. Παρακάτω, μάλιστα, επιχειρούμε να λύσουμε μία επέκταση ενός δεδομένου και λυμένου προβλήματος γραμμικού προγραμματισμού χρησιμοποιώντας το δυϊκό του.

Ορισμός 1.3 (Τυχαιοποιημένος-Randomized)**.** Ονομάζουμε *τυχαιοποιημένο (randomized)* έναν αλγόριθμο που χρησιμοποιεί ομοιόμορφα τυχαία bits μαζί με την είσοδό του, ώστε να πάρει αποφάσεις. Επομένως, οι τελικές αποφάσεις ενός τυχαιοποιημένου αλγορίθμου δεν μπορούν να είναι γνωστές από την αρχή.

Τέλος, θα μιλήσουμε για μία άλλη κατηγορία αλγορίθμων, τους οποίους θα χρησιμοποιήσουμε σε αυτή την εργασία, τους online αλγορίθμους.

Ορισμός 1.4 (Online Algorithms)**.** Online ονομάζονται οι αλγόριθμοι οι οποίοι προσπελάζουν την είσοδό τους σειριακά, με την έννοια ότι μαθαίνουν σταδιακά την είσοδο καθώς προχωράει ο αλγόριθμος και δεν την γνωρίζουν εξαρχής. Η σύγκριση μεταξή της απόδοσης του online αλγορίθμου και του αντίστοιχου offline ονομάζεται *ανταγωνιστικός λόγος (competitive ratio).*

Οι online αλγόριθμοι είναι ιδιαιτέρως χρήσιμοι όταν πρέπει οι αποφάσεις να λαμβάνονται «ακαριαία» ή όταν έχουμε περιορισμένη γνώση σχετικά με το μέλλον. Παραδείγματα περιπτώσεων όπου θα μας ήταν χρήσιμο να κατασκευάσουμε online αλγόριθμο είναι *η ανάθεση πόρων (όπως ο χρόνος στη CPU), ή το εύρος για κάποιες διεργασίες ή η on-demand ζήτηση ραδιοφωνικού εύρους.*

Αξίζει τέλος να αναφέρουμε εδώ ότι εξετάζουμε κάθε φορά διάφορους τρόπους να δίνεται η είσοδος στον online αλγόριθμο με πιο χαρακτηριστικούς το *μοντέλο της γραμματέως (secretary model)* (το οποίο έχει μελετηθεί αναλυτικά στα [8, 9, 10]), όπου ουσιαστικά η είσοδος δίνεται με τυχαίο τρόπο, και το *μοντέλο του αντιπάλου (adversarial model)* ([11]), όπου, όπως φαίνεται και από το όνομά του, η είσοδος δίνεται με τον χειρότερο δυνατό τρόπο στον αλγόριθμο.

1.3 Σχεδιασμός Μηχανισμών (Mechanism Design)

Μεγάλο ποσοστό της διεθνούς βιβλιογραφίας έχει εστιαστεί στη μελέτη της αποδοτικότητας, καλύτερα της μή-αποδοτικότητας (inefficiency) της εγωιστικής (selfish) συμπεριφοράς των παικτών κατά τη διάρκεια λήψης κάποιας απόφασης. Τις περισσότερες φορές, οι εγωιστικές κινήσεις των παικτών, που στόχο έχουν την επικράτησή τους έναντι των συμπαικτών τους, οδηγούν σε αποτελέσματα τα οποία είναι συγκρίσιμα ή και άλλοτε, μή-συγκρίσιμα σε σχέση με τη βέλτιστη λύση.

Με τον όρο «Σχεδιασμός Μηχανισμών» (Mechanism Design) αναφερόμαστε ουσιαστικά σε μία διαδικασία Reverse Engineering ως προς το Game Theory. Με άλλα λόγια, μοντελοποιούμε την αλληλεπίδραση των παικτών μας με βάση κάποιους μηχανισμούς, με δεδομένη τη συμπεριφορά που θα έχουν οι παίκτες αυτοί στο παιχνίδι. Σε έναν μηχανισμό ένας παίκτης έχει έναν χώρο από στρατηγικές για να επιλέξει και οι αποφάσεις που λαμβάνει εξαρτώνται από κάποια συνάρτηση πάνω σε αυτό το χώρο. Για παράδειγμα, στο σχεδιασμό ενός μηχανισμού για Δημοπρασίες (auction), ο χώρος των στρατηγικών του κάθε παίκτη είναι όλα τα δυνατά bids που σκέφτεται να υποβάλει και οι αποφάσεις είναι συνάρτηση των outcomes που λαμβάνει ανάλογα με το κάθε bid. Υπάρχουν πολλές υποθέσεις και συνθήκες κάτω από τις οποίες μπορούμε να μελετήσουμε και να σχεδιάσουμε μηχανισμούς: για παράδειγμα, θα μπορούσαν οι παίκτες να έχουν κρυφό ή φανερό budget, θα μπορούσαμε να δημοπραττούμε ένα ή περισσότερα αντικείμενα, οι παίκτες θα μπορούσαν να θέλουν ολόκληρα αντικείμενα ή έστω μερίδια αυτών και πολλά άλλα. Στόχος μας είναι να καταφέρουμε να εναρμονίσουμε τα ιδιωτικά συμφέροντα των παικτών μας, με το συμφέρον του σχεδιαστή της δημοπρασίας ή του παιγνίου γενικότερα.

Παρακάτω θα ακολουθήσουν μερικοί βασικοί ορισμοί και κάποια από τα σημαντικότερα θεωρήματα της περιοχής του Mechanism Design. ([1])

Βασικό Πλαίσιο. Συμβολίζουμε με O το σύνολο των δυνατών outcomes. Για κάθε παίκτη i αντιστοιχίζουμε μία τιμή $v_i : O \rightarrow \mathbb{R}$, η οποία είναι το *valuation* του παίκτη για το κάθε outcome του παιγνίου και ορίζουμε επίσης ως *valuation profile* το διάνυσμα: $\mathbf{v} = (v_1, v_2, \dots, v_n)$. Μπορούμε να σκεφτόμαστε το *valuation* του παίκτη ως την ευχαρίστηση που παίρνει από την απόκτηση του αντικειμένου. Ο κάθε παίκτης μπορεί, ωστόσο, να υποβάλει bids για το αντικείμενο. Έτσι, συμβολίζουμε με $b_i \in \mathbb{R}$ το bid του παίκτη i και με $\mathbf{b} = (b_1, b_2, \dots, b_n)$ τα bids όλων των παικτών. Ονομάζουμε συνάρτηση ανάθεσης (*allocation function*) τη συνάρτηση $f : b_1 \times b_2 \times \cdots \times b_n \rightarrow \mathbb{R}^n$ με βάση την οποία καθορίζεται τι ακριβώς παίρνει ο κάθε παίκτης από τη δημοπρασία και κανόνα πληρωμής (*payment rule*) τη συνάρτηση $p(b_1, b_2, \dots, b_n) \in \mathbb{R}^n$ που καθορίζει το ποσό που πληρώνει ο κάθε παίκτης ανάλογα με την ανάθεση που του έγινε. Τέλος, ονομάζουμε *ωφέλεια (utility)* τη συνάρτηση $u_i(b_1, b_2, \dots, b_n) = v_i(f(b_1, b_2, \dots, b_n)) - p_i(b_1, b_2, \dots, b_n)$.

Σχόλια. Συμβολίζουμε πολύ συχνά, για λόγους οικονομίας, ως v_{-i} το διάνυσμα των valuations χωρίς το valuation του παίκτη i, δηλαδή $v_{-i} = (v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n)$ και αντίστοιχα για το u_{-i}.

Με βάση αυτά που αναφέραμε παραπάνω, ο παίκτης έχει το valuation και το bid του. Το ενδιαφέρον εδώ είναι ότι αυτές οι δύο τιμές δεν ταυτίζονται απαραίτητα. Θεωρούμε, λοιπόν, ότι το valuation είναι πάντα αληθινό, δηλαδή εκφράζει επακριβώς την ευχαρίστηση του παίκτη για το κάθε outcome, αλλά το bid μπορεί να μην ισούται με το valuation. Με άλλα λόγια, μπορεί ο παίκτης όταν μας αποκαλύψει το bid του να μας *πει ψέμματα (misreport)*. Συνεπώς, είναι λογικό να μας απασχολεί η έννοια της φιλαλήθειας (*truthfulness or incentive compatibility*).

Ορισμός 1.5 (Direct Revelation)**.** [1] Ένας *μηχανισμός άμεσης αποκάλυψης (direct revelation mechanism)* αποτελείται από μία allocation function $f : v_1 \times v_2 \times \cdots \times v_n \rightarrow A$ και ένα payment rule $p_i : v_1 \times v_2 \times \cdots \times v_n \rightarrow \mathbb{R}$.

Ορισμός 1.6 (Truthful or Incentive Compatible)**.** [1] Ένας μηχανισμός $f, p_1, p_2, \ldots, p_n$ ονομάζεται *φιλαλήθης (incentive compatible or truthful)*, αν για κάθε παίκτη i, $v_1 \in V_1, \ldots, v_n \in V_n$ και κάθε $v_i' \in V_i$ ίσχυει ότι

$$v_i(\alpha) - p_i(v_i, v_{-i}) \geq v_i(\alpha') - p_i(v_i, v_{-i})$$

όπου $\alpha = f(v_i, v_{-i})$ και $\alpha' = f(v_i', v_{-i})$.

Διαισθητικά, αυτό που μας λέει ο παραπάνω ορισμός είναι ότι ο παίκτης i, με valuation v_i θα προτιμούσε να πει το αληθινό v_i στον μηχανισμό, αντί για κάποιο ψεύτικο v_i' προκειμένου να μεγιστοποιήσει το utility του.

Ορισμός 1.7 (Individually Rational)**.** [1] Ένας μηχανισμός ονομάζεται *ατομικά λογικός (individually rational)* αν οι παίκτες πάντοτε έχουν μη-αρνητικό utility. Δηλαδή, για κάθε $i \in N$ και για $v_1, v_2, \ldots, v_n$ έχουμε ότι:

$$v_i(f(v_1, v_2, \ldots, v_n)) - p_i(v_1, v_2, \ldots, v_n) \geq 0$$

Ορισμός 1.8 (Dominant Strategy)**.** Ένα προφίλ $b \in V_i$ είναι *κυρίαρχη στρατηγική (dominant strategy)* για τον παίκτη i, αν $\forall b_{-i} \in V_{-i}$, $b' \in V_i$ ισχύει:

$$u_i(f(b_{-i}, b)) \geq u_i(f(b_{-i}, b'))$$

Όπως βλέπουμε από τον συμβολισμό του ορισμού για την dominant strategy, δεν είναι απαραίτητο το $b \in V_i$ να *ταυτίζεται* με το πραγματικό valuation, v_i, του παίκτη i. Ως εκ τούτου, γεννάται το φυσικό ερώτημα: *γίνεται μία στρατηγική να είναι dominant strategy αλλά ο μηχανισμός να μην είναι truthful*; Η απάντηση σε αυτό το ερώτημα είναι **όχι** και έχουμε ένα πολύ ισχυρό θεώρημα που το αποδεικνύει: το θεώρημα Revelation Principle. Διαισθητικά, το Θεώρημα μας λέει ότι για κάθε μηχανισμό που κάθε παίκτης έχει dominant strategy, υπάρχει ένας αντίστοιχος άμεσης αποκάλυψης φιλαλήθης (direct revelation truthful) μηχανισμός.

Θεώρημα 1.1 (Revelation Principle)**.** *Κάθε μηχανισμός που υλοποιεί μία allocation function, f, σε κυρίαρχες στρατηγικές (dominant strategies) μπορεί να μετατραπεί σε έναν άμεσο φιλαλήθη (direct truthful) μηχανισμό που να υλοποιεί την f.*

Απόδειξη. Έστω ότι κάθε παίκτης i έχει ένα σύνολο από στρατηγικές S_i και μία συνάρτηση s_i, τέτοια ώστε το $s_i(b_i)$ να καθορίζει την dominant strategy του παίκτη. Έστω ακόμη ότι ο allocation rule είναι μία συνάρτηση $g : \Pi_i S_i \to O$ πάνω στις στρατηγικές των παικτών, έτσι ώστε: $g(s_1(b_1), \ldots, s_n(b_n)) = f(b_1, \ldots, b_n)$. Εφόσον οι συναρτήσεις s_i είναι γνωστές, μπορούμε να δημιουργήσουμε έναν άμεσο (direct) μηχανισμό h, που ορίζεται ως: $h(b_1, \ldots, b_n) = g(s_1(b_1), \ldots, s_n(b_n))$. Με βάση τους ορισμούς που έχουμε δώσει, αν ο κανόνας g υλοποιεί την f, τότε η h επίσης υλοποιεί την f. Επιπλέον, εφόσον η h προσομοιώνει επακριβώς τη συμπεριφορά των παικτών στην εκτέλεση του g, οι παίκτες δεν έχουν κανένα κίνητρο να πουν ψέμματα στην h, γιατί αυτό θα ήταν ισοδύναμο με το να πουν ψέμματα στον ίδιο τους τον εαυτό κατά το σχηματισμό της g.

■

Αρκεί, λοιπόν, χωρίς βλάβη της γενικότητας να εστιάσουμε την προσοχή μας σε direct truthful μηχανισμούς. Το παρακάτω Θεώρημα αποκλείει πρακτικά την ύπαρξη μη-τετριμμένων λύσεων σε περιπτώσεις παιγνίων, όπου υπάρχουν περισσότερα των 3 outcomes.

Θεώρημα 1.2 (Gibbard-Satterthwaite, [2, 3])**.** *Δεν υπάρχουν φιλαλήθεις κανόνες ανάθεσης (truthful allocation rules) που ικανοποιούν ταυτόχρονα τα ακόλουθα:*

i. $|O| \geq 3$

ii. *ο allocation rule είναι «επί», δηλαδή* $\forall o \in O, \exists$ *valuation profile* $v \in V : f(v) = o$

iii. *ο κανόνας δεν είναι δικτατορικός (dictatorial), δηλαδή δεν υπάρχει παίκτης* i*, με* $v_i \in V_i$ *τέτοιος ώστε για οποιοδήποτε* $v_{-i} \in V_{-i}$ *να ισχύει ότι:* $f(v_i, v_{-i}) = \arg\max_o v_i(o)$

Διαισθητικά, το παραπάνω θεώρημα μας λέει ότι στην πιο γενική του μορφή, ένα πρόβλημα στο mechanism design δεν έχει μη-τετριμμένες λύσεις. Για να αντιμετωπίσουμε τις δυσκολίες που θέτει το παραπάνω θεώρημα, εισάγουμε στους μηχανισμούς μας την έννοια των πληρωμών, του αντιτίμου δηλαδή που καλούνται να δώσουν οι παίκτες σαν αντίβαρο στην ευχαρίστηση που παίρνουν αποκτώντας το κάθε outcome. Ένας άλλος τρόπος να αντιμετωπίσουμε τις δυσκολίες αυτές είναι να περιορίσουμε το χώρο των συναρτήσεων ωφέλειας (utility functions) των παικτών. Αξίζει εδώ να αναφέρουμε ότι μία πολύ συνήθης υπόθεση είναι ότι οι utility functions είναι ημιγραμμικές (quasi-linear).

Αντικειμενικές Συναρτήσεις. Προκειμένου να αξιολογήσουμε την απόδοση των μηχανισμών και να τους συγκρίνουμε μεταξύ τους, χρειαζόμαστε τις αντικειμενικές συναρτήσεις (objective functions). Στη βιβλιογραφία υπάρχουν αρκετές objective functions με πιο γνωστή την **Κοινωνική Ευημερία (Social Welfare)**. Διαισθητικά, το Social Welfare αντιστοιχεί στο «ποσό» της ευχαρίστησης που συγκεντρώνουν αθροιστικά όλοι οι παίκτες του παιγνίου, εξαιτίας κάποιου συγκεκριμένου αποτελέσματος. Μία άλλη πολύ γνωστή αντικειμενική συνάρτηση είναι τα **Έσοδα (Revenue)**, η οποία αντιστοιχεί στο ποσό των χρημάτων που συγκεντρώνει ο σχεδιαστής του παιγνίου από τις πληρωμές που γίνονται από όλους τους παίκτες για κάποιο outcome. Στοχεύουμε στο να μεγιστοποιούμε ή να ελαχιστοποιούμε τις αντικειμενικές συναρτήσεις σχεδιάζοντας κατάλληλους αλγορίθμους. Παρακάτω ακολουθούν οι ορισμοί των δύο αντικειμενικών συναρτήσεων που αναφέραμε.

Ορισμός 1.9 (Revenue)**.** Έστω ο μηχανισμός $M = (f, p)$. Ορίζουμε ως *Έσοδα (Revenue)* του μηχανισμού, την ποσότητα:

$$Rev(v) = \sum_i p_i(v)$$

Ορισμός 1.10 (Social Welfare)**.** Για ένα outcome $o \in O$, ονομάζουμε *Κοινωνική Ευημερία (Social Welfare)* την ποσότητα:

$$SW[o] = \sum_i v_i(o)$$

Σε αυτή την εργασία θα εστιάσουμε την προσοχή μας σε μια πιο πρόσφατη αντικειμενική συνάρτηση, τη **Ρευστή Ευημερία (Liquid Welfare)** για την οποία θα αναφερθούμε εκτενώς σε επόμενο κεφάλαιο.

Στη συνέχεια θα αναφερθούμε σε ένα πολύ ισχυρό θεώρημα του Σχεδιασμού Μηχανισμών, το *Λήμμα του Myerson (Myerson's Lemma)*, το οποίοι εν ολίγοις δίνει τις συνθήκες που πρέπει να ισχύουν ώστε ένας allocation rule να είναι υλοποιήσιμος (implementable) και επιπλέον, καθορίζει και πώς θα ορίζεται το σχήμα πληρωμών σε έναν τέτοιο μηχανισμό.

Ορισμός 1.11 (implementable)**.** Ένας μηχανισμός είναι *υλοποιήσιμος (implementable)* εάν υπάρχει ένα σχήμα πληρωμών (payments scheme), p, τέτοιο ώστε ο μηχανισμός (f, p) να είναι φιλαλήθης (truthful).

Ορισμός 1.12 (Μονότονος)**.** Ένας κανόνας ανάθεσης $\mathbf{f}$ (allocation rule) είναι *μονότονος* εάν για κάθε παίκτη i και bids $\mathbf{b}_{-i}$ από τους άλλους bidders, η ανάθεση $f_i(z, \mathbf{b}_{-i})$ για τον παίκτη i δεν είναι φθίνουσα ως προς το bid z.

Διαισθητικά, το παραπάνω σημαίνει ότι σε έναν μονότονο κανόνα ανάθεσης, όσο μεγαλώνεις το bid που αναφέρεις, θα παίρνεις μεγαλύτερο μερίδιο ενός outcome.

Θεώρημα 1.3 (Myerson's Lemma, [4])**.** *Ένας μηχανισμός για ένα πρόβλημα μίας παραμέτρου είναι φιλαλήθης κυρίαρχη στρατηγική (dominant strategy truthful) ανν για κάθε παίκτη i και καθορισμένα bids $\mathbf{b}_{-i}$ για τους υπόλοιπους παίκτες ισχύουν τα εξής:*

- *η $f_i(b_i)$ είναι μη-φθίνουσα συνάρτηση του bid του παίκτη i, b_i*

- *ο κανόνας των πληρωμών είναι ο εξής:*

$$p_i(b_i, b_{-i}) = \int_0^{b_i} z \cdot \frac{d}{dz} f_i(z, b_{-i}) \tag{1.1}$$

Bayesian Analysis. Το μοντέλο που θα χρησιμοποιήσουμε για την ανάλυση αυτή είναι ελαφρώς διαφορετικό από το προηγούμενο. Συγκεκριμένα, έχουμε:

- Ένα περιβάλλον μίας παραμέτρου (single-parameter) [1], όπως ακριβώς είχαμε και μέχρι τώρα.
- Η κρυφή (private) valuation v_i του παίκτη i προκύπτει από μία κατανομή F_i με συνάρτηση πυκνότητας πιθανότητας f_i μέσα στο διάστημα $[0, v_{\max}]$. Υποθέτουμε ότι οι κατανομές $F_1, \ldots, F_n$ είναι ανεξάρτητες, αλλά όχι απαραίτητα ίδιες. [2]
- Οι κατανομές $F_1, \ldots, F_n$ είναι γνωστές εξαρχής στον σχεδιαστή του μηχανισμού. Εφόσον εστιάζουμε την προσοχή μας σε dominant strategy truthful μηχανισμούς, οι παίκτες δεν χρειάζεται να γνωρίζουν τις κατανομές $F_1, \ldots, F_n$.

Σε ένα τέτοιο Bayesian περιβάλλον είναι προφανές ότι η «βέλτιστη» δημοπρασία είναι αυτή που μας αποφέρει τα υψηλότερα αναμενόμενα έσοδα (expected revenue), δεδομένων πάντα των κατανομών $F_1 \times \cdots \times F_n$ πάνω στα valuation profiles $\mathbf{v}$.

Ορισμός 1.13 (Virtual Valuation)**.** Η *Εικονική Ευχαρίστηση (Virtual Valuation)* ενός παίκτη i με valuation v_i η οποία προκύπτει από κάποια κατανομή F_i ορίζεται ως:

$$\phi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

Αξίζει εδώ να σημειώσουμε ότι η virtual valuation ενός παίκτη εξαρτάται, όπως άλλωστε φαίνεται και από τη μαθηματική της έκφραση, από τη valuation του παίκτη και την κατανομή του *και μόνο*. Δηλαδή, δεν εξαρτάται από τους άλλους παίκτες. Οι virtual valuations έχουν εξέχουσα θέση στον σχεδιασμό βέλτιστων Bayesian δημοπρασιών. Διαισθητικά, ο πρώτος όρος της μαθηματικής έκφρασης του $\phi_i(v_i)$ αντιστοιχεί στα μέγιστα έσοδα που μπορούμε να πάρουμε από τον bidder i και ο δεύτερος όρος είναι το ποσό που καλούμαστε να «πληρώσουμε» προκειμένου να μάθουμε την κατανομή του valuation (information rent). Μία δεύτερη και πιο ακριβής ερμηνεία της $\phi_i(v_i)$ είναι ως η κλίση μιας «καμπύλης εσόδων» στο v_i, όπου η καμπύλη εσόδων αναπαριστά τα αναμενόμενα έσοδα που αποκτώνται από έναν παίκτη με valuation που προκύπτει από μία κατανομή F_i σαν συνάρτηση της πιθανότητας κάποιας πώλησης.

[1] Όταν αναφερόμαστε σε περιβάλλοντα μίας παραμέτρου (single parameter environment) ουσιαστικά κάνουμε λόγο για προβλήματα όπου οι παίκτες έχουν μόνο **μία** κρυφή πληροφορία: το valuation τους.

[2] Στην πράξη, αυτές οι κατανομές προκύπτουν κατά κάποιον τρόπο από προηγούμενα data, με κάποιου είδους sampling.

Ορισμός 1.14 (Virtual Welfare)**.** Για ένα valuation profile $\mathbf{v}$ ονομάζουμε *virtual welfare* την ποσότητα:

$$VW[\mathbf{v}] = \sum_{i=1}^{n} \phi_i(v_i) \cdot x_i(\mathbf{v})$$

Ακολουθώντας τα βήματα της απόδειξης του Myerson Lemma, αλλά αυτή τη φορά στο Bayesian πλαίσιο, προκύπτει ότι *για κάθε δημοπρασία, τα αναμενόμενα έσοδα ισούνται με το αναμενόμενο virtual welfare.* Με άλλα λόγια, ο υπολογισμός των μέγιστων εσόδων πάνω στις dominant strategy truthful δημοπρασίες ανάγεται ουσιαστικά στη μεγιστοποίηση του αναμενόμενου virtual welfare.

1.4 VCG vs Posted Prices

Σκοπίμως, σε όσα έχουμε αναφέρει ως τώρα έχουμε κάνει μία λεπτή διάκριση, καθόλου εμφανή στον αναγνώστη: έχουμε «κρύψει» πότε αναφερόμαστε σε single-parameter περιβάλλοντα και πότε σε multi-parameter περιβάλλοντα. Για να κάνουμε πιο εύκολη την κατανόηση θα εστιάσουμε για λίγο σε *δημοπρασίες*.

- [**single-parameter**] *Περιβάλλον μίας παραμέτρου* ονομάζεται μία δημοπρασία με n bidders, όπου κάθε bidder i έχει το κρυφό του valuation, v_i, το οποίο αντιστοιχεί στην «ανά μονάδα» αντικειμένου αξία που λαμβάνει από τη δημοπρασία. Σε ένα τέτοιο περιβάλλον υπάρχει επίσης ένα *εφικτό (feasible)* σύνολο X. Το X είναι ένα διάνυσμα: $X = (x_1, x_2, \ldots, x_n)$, όπου το x_i αντιστοιχεί στο ποσό του «αντικειμένου» που δίνεται στον παίκτη i.
- [**multi-parameter**] *Περιβάλλον πολλών παραμέτρων* εν γένει ονομάζεται ένα mechanism design πρόβλημα με n στρατηγικούς παίκτες, ένα πεπερασμένο σύνολο από outcomes Ω και κρυφές valuations $v_i(\omega)$ των παικτών για κάθε outcome $\omega \in \Omega$. [3]

Παράδειγμα single-parameter περιβάλλοντος. Θα αναφερθούμε εδώ στην πολύ γνωστή δημοπρασία του *Vickrey (Vickrey auction)*, ή αλλιώς *sealed bid second-price auction*. Ο αλγόριθμος των sealed bid auctions είναι ο ακόλουθος.

1. Each bidder i privately communicates his/her bid, b_i, to the auctioneer.
2. Auctioneer decides who gets the item.
3. Auctioneer decides the price that the bidder will pay.

Algorithm 1: Sealed Bid Auction

Προφανώς, είναι μάλλον εύκολο να καταλάβουμε το βήμα (2): το αγαθό θα το πάρει ο bidder που έχει ποντάρει τα πιο πολλά, δηλαδή, που έχει το μεγαλύτερο bid. Ο σχεδιασμός του βήματος (3), δηλαδή οι πληρωμές, ήταν η συμβολή του Vickrey σε αυτές τις δημοπρασίες. Σύμφωνα με τον Vickrey προκειμένου να έχουμε truthful μηχανισμό πρέπει ο νικητής να πληρώσει το ποσό του δεύτερου μεγαλύτερου bidder. [4] Η απόδειξη αυτού είναι απλή, αρκεί να σκεφτούμε τι εναλλακτικές στρατηγικές έχει ο κάθε παίκτης: είτε να ποντάρει το αληθινό του valuation ή κάποιο ψεύτικο (*μεγαλύτερο ή μικρότερο*) και τι αποτελέσματα επιφέρει η κάθε στρατηγική στο utility του.

Διαισθητικά, αυτό που μας λέει η Vickrey auction είναι ότι προκειμένου να έχουμε truthfulness σε ένα **single-parameter** περιβάλλον, αρκεί να χρεώνουμε τον νικητή του αγαθού (σ.σ: αυτόν με το μεγαλύτερο bid) την *επήρεια (externality)* που έχει η συμμετοχή του στη δημοπρασία για την utility των άλλων παικτών [5]. Ο μηχανισμός Vickrey-Clarke-Groves αποτελεί ουσιαστικά

[3] Επομένως, οι ορισμοί που δώσαμε στην προηγούμενη ενότητα αναφέρονταν στη γενικότερη περίπτωση, δηλαδή, στο multi-parameter περιβάλλον, εκτός από το Myerson Lemma που διατυπώθηκε στο single-parameter.

[4] Εξ ου και το όνομα αυτών των δημοπρασιών: *second-price or Vickrey.*

[5] Με άλλα λόγια το πόσο μειώνεται η ευημερία των υπολοίπων παικτών από την είσοδο του μεγαλύτερου bidder.

μία επέκταση-γενίκευση της second-price δημοπρασίας για multi-parameter περιβάλλοντα. Η ιδέα που παραμένει αναλλοίωτη είναι η «εξήγηση» που δώσαμε παραπάνω: κάθε bidder χρεώνεται το externality που προκαλεί στους υπόλοιπους.

Θεώρημα 1.4 (Vickrey-Clarke-Groves (VCG), [7, 5, 6])**.** *Ονομάζουμε* VCG *μηχανισμό έναν μηχανισμό* (f, p) *με:*

- συνάρτηση ανάθεσης*:* $f(\mathbf{b}) = \arg\max_{\omega\in\Omega} \sum_{i=1}^{n} b_i(\omega)$
- σχήμα πληρωμών*:* $p_i(\mathbf{b}) = \max_{\omega\in\Omega} \sum_{j\neq i} b_i(\omega) - \sum_{j\neq i} b_j(f(\mathbf{b}))$

Εύκολα αποδεικνύεται ότι ο παραπάνω μηχανισμός είναι individually rational, δηλαδή ότι κάθε bidder θα έχει μη-αρνητικό utility αν παίξει ειλικρινώς. Θα αποδείξουμε, ωστόσο, το truthfulness του σχήματος πληρωμών.

Θεώρημα 1.5. *Ο μηχανισμός VCG είναι truthful.*

Απόδειξη. Διαλέγουμε ένα οποιοδήποτε i και το αντίστοιχο b_{-i} και έστω ω^* η ανάθεση που αποφασίζεται από τη συνάρτηση ανάθεσης του μηχανισμού. Τότε, το utility του παίκτη i θα είναι:

$$v_i(\omega^*) - p_i(\mathbf{b}) = \underbrace{\left[v_i(\omega^*) + \sum_{j\neq i} b_j(\omega^*) \right]}_{\text{(A)}} - \underbrace{\left[\max_{\omega\in\Omega} \sum_{j\neq i} b_j(\omega) \right]}_{\text{(B)}}$$

Είναι εύκολο να δούμε ότι το (Β) δεν αλλάζει και είναι ανεξάρτητο από το b_i. Επομένως, εφόσον θέλουμε να μεγιστοποιήσουμε το u_i, αρκεί να μεγιστοποιήσουμε το (Α). Ας υποθέσουμε τώρα κάτι ακόμη ισχυρότερο από το δεδομένο μας: ότι ο παίκτης i έχει τη δύναμη να *καθορίσει όπως θέλει αυτός* το outcome ω^*, και όχι απλά να συμμετέχει στον σχηματισμό του, όπως έχουμε υποθέσει ως τώρα. Σε μία τέτοια περίπτωση, ο (Α) θα προσπαθούσε να μεγιστοποιήσει το (Α). Παρατηρούμε όμως, ότι αν θέσουμε $b_i = v_i$ στη συνάρτηση ανάθεσης του θεωρήματος, προκύπτει ότι η ποσότητα που προσπαθεί να μεγιστοποιήσει ο μηχανισμός *ισούται* με την ποσότητα (Α), που θέλει να μεγιστοποιήσει ο παίκτης. Πράγμα που σημαίνει ότι αν ο παίκτης ποντάρει το πραγματικό του valuation, τότε και μόνο τότε μεγιστοποιείται το utility του και άρα, ο μηχανισμός VCG είναι truthful. ∎

Τα προβλήματα του VCG. Ο μηχανισμός VCG φαντάζει ιδανικός: μας εξασφαλίζει truthful μηχανισμούς που μέσω του σχήματος πληρωμών εξασφαλίζουν μεγιστοποίηση του welfare. Ωστόσο, αυτό δεν αντικατοπτρίζει την πραγματικότητα. Τα μειονεκτήματα του VCG γίνονται εμφανή σε μία ιδιαίτερη οικογένεια προβλημάτων: τις *Συνδυαστικές Δημοπρασίες (Combinatorial Auctions).*

Ορισμός 1.15 (Combinatorial Auctions)**.** Μία *Συνδυαστική Δημοπρασία (Combinatorial Auction)* (CA) είναι ένα μοντέλο δημοπρασίας με τα εξής χαρακτηριστικά:

- n παίκτες (bidders)
- Ένα σύνολο, M, από m αντικείμενα, *όχι* ίδια
- Το σύνολο των outcomes Ω αντιστοιχεί σε n-διανύσματα $(S_1, S_2, \ldots, S_n)$, όπου το S_i αντιστοιχεί στα αντικείμενα που ανατίθενται στον παίκτη i. Κανένα αντικείμενο δεν ανατίθεται (στη γενική περίπτωση) 2 φορές. [α´] Συμπεραίνουμε εύκολα ότι το πλήθος των δυνατών outcomes είναι $(n+1)^m$.
- Κάθε bidder έχει valuation $v_i(S)$ για κάθε $S \subseteq M$ που μπορεί να του ανατεθεί.

[α´] Το σύνολο των αντικειμένων που ανατίθενται σε κάποιον παίκτη ονομάζεται στη βιβλιογραφία bundle.

Από τα παραπάνω συμπεραίνουμε εύκολα ότι ο κάθε bidder έχει 2^m κρυφές παραμέτρους, όσα δηλαδή και τα κρυφά valuations του για κάθε πακέτο αντικειμένων που μπορεί να λάβει. Επίσης, συνήθως υποθέτουμε ότι $v_i(\emptyset) = 0$ και ότι $v_i(S) \leq v_i(T)$, αν $S \leq T$.

Μία Συνδυαστική Δημοπρασία μπορεί να εκφραστεί ως ένα πρόβλημα Γραμμικού Προγραμματισμού (LP). Το primal πρόβλημα είναι το εξής:

$$\max \sum_{i \in N, S \subseteq M} x_{i,S} v_i(S) \tag{1.2}$$

$$\textbf{s.t}: \sum_{i \in N, S | j \in M} x_{i,S} \leq 1, \forall j \in M \tag{1.3}$$

$$\sum_{S \subseteq M} x_{i,S} \leq 1, \forall i \in N \tag{1.4}$$

$$x_{i,S} \geq 0, \forall i \in N, S \subseteq M \tag{1.5}$$

Η παραπάνω μαθηματική γραφή προκύπτει άμεσα από τον ορισμό των Combinatorial Auctions. Η 1.2 ουσιαστικά μας λέει ότι στόχος μας είναι να μεγιστοποιήσουμε το social welfare των παικτών. Η 1.3 μας λέει ότι κάθε παίκτης θα πάρει το πολύ ένα bundle αντικειμένων και η 1.4 μας λέει ουσιαστικά ότι κάθε αντικείμενο δεν πρέπει να πωλείται παραπάνω φορές από όσες διατίθεται (δεν γίνεται overselling).

Έχοντας εξηγήσει προηγούμενως τι σημαίνει Δυϊκό Πρόβλημα καθώς και την αξία που μπορεί να έχει η Δυϊκότητα, προχωράμε τώρα στην περιγραφή των συνδυαστικών δημοπρασιών ως Dual Linear Programs.

$$\min \sum_{i \in N} u_i + \sum_{j \in M} p_j \tag{1.6}$$

$$\mathbf{s.t}: u_i + \sum_{j \in S} p_j \geq v_i(S), \forall i \in N, S \subseteq M \tag{1.7}$$

$$u_i \geq 0, p_j \geq 0, \forall i \in N, j \in M \tag{1.8}$$

Στο Δυϊκό Πρόβλημα πρέπει να παρατηρήσουμε ότι είμαστε από τη μεριά του auctioneer και όχι των παικτών. Έτσι, ο auctioneer θέλει να διαθέσει τα αντικείμενα έχοντας τη λιγότερη δυνατή ικανοποίηση των παικτών, κάτι που θα του αποφέρει μεγαλύτερα κέρδη (1.6). Ο auctioneer θέλει επίσης να ισχύει η ανισότητα 1.7 προκειμένου να έχει κίνητρο ο παίκτης να αγοράσει το bundle.

Ήδη από την απλή περιγραφή του μοντέλου καταλαβαίνουμε ότι είναι πολύ πιο περίπλοκο σε σχέση με τα πιο απλά μοντέλα δημοπρασιών που είχαμε συζητήσει ως τώρα. Στην πραγματικότητα, οι Συνδυαστικές Δημοπρασίες μας θέτουν 3 βασικές δυσκολίες, που πρέπει να αντιμετωπίσουμε.

Δυσκολία 1. Η πρώτη δυσκολία εστιάζεται στο πλήθος των πληροφοριών. Κάθε bidder έχει $2^m - 1$ κρυφές παραμέτρους, οι οποίες αντιστοιχούν στα valuations που έχει για κάθε διαφορετικό bundle που μπορεί να του ανατεθεί. Ένα τέτοιο τεράστιο μέγεθος παραμέτρων είναι δύσκολο και για τον bidder να το απαριθμήσει, αλλά και για τον σχεδιαστή της δημοπρασίας να το ακούσει (communication complexity). Αυτό το εκθετικό μέγεθος παραμέτρων μας δείχνει τα πρώτα προβλήματα του VCG μηχανισμού, καθώς και όλων των άλλων direct-revelation μηχανισμών.

Δυσκολία 2. Ακόμη και αν θεωρήσουμε, όμως, ότι με κάποιον τρόπο ξεπερνάμε την πρώτη δυσκολία, η δεύτερη δυσκολία εστιάζεται στο να αναθέσουμε τα αντικείμενα της δημοπρασίας με τέτοιον τρόπο, ώστε να πετυχαίνουμε μεγιστοποίηση του welfare ή έστω μία καλή προσέγγιση αυτού. Είναι αρκετά δύσκολο έως αδύνατο να ελέγξουμε ευθέως το παραπάνω, εφόσον τα valuations των bidders είναι άγνωστα και οι δημοπρασίες που χρησιμοποιούνται στην πράξη προσφέρουν κάποιες δυνατότητες στρατηγικής (ξεφεύγουν, με άλλα λόγια από το κλασικό dominant strategy incentive compatible που έχουμε εξετάσει).

Δυσκολία 3. Ακόμη και αν αγνοήσουμε τις προηγούμενες δύο δυσκολίες, το τρίτο μας πρόβλημα αφορά στον μηχανισμό VCG, ο οποίος μπορεί να έχει πολύ κακές ιδιότητες, παρότι είναι dominant strategy incentive compatible [12]. Για παράδειγμα, έστω ότι έχουμε δύο bidders που θέλουν να αποκτήσουν δύο αντικείμενα: τα A, B. Ο πρώτος bidder θέλει και τα δύο αντικείμενα και άρα έχει $v_1(AB) = 1$ και $v_1 = 0$ για κάθε άλλη περίπτωση. Ο δεύτερος bidder θέλει μόνο το αντικείμενο A, άρα $v_2(AB) = v_2(A) = 1$. Εύκολα μπορεί να δει κανείς ότι τα έσοδα του VCG μηχανισμού σε αυτό το παράδειγμα είναι 1. Έστω τώρα, όμως, ότι προσθέτουμε έναν τρίτο bidder, ο οποίος θέλει μόνο το αντικείμενο B, άρα $v_3(AB) = v_3(B) = 1$. Τώρα, το μέγιστο welfare είναι 2, ενώ το welfare που θα πάρουμε από τον VCG μηχανισμό έχει πέσει στο 0.

Τέλος, ένα χαρακτηριστικό των combinatorial auctions είναι πολύ συχνά πραγματοποιούνται επαναληπτικά. Με άλλα λόγια, πραγματοποιούνται σε n γύρους. Έχοντας τις τρεις δυσκολίες κατά νου, καθώς και το χαρακτηριστικό της επανάληψης, θα παρουσιάσουμε τώρα έναν άλλο τρόπο να φτιάχνουμε «σχήματα πληρωμών»: τις *δημοσιευμένες τιμές (posted prices)*.

Ορισμός 1.16 (Posted Price Mechanism)**.** *Ένας μηχανισμός δημοσιευμένων τιμών (posted price)* δίνει σε κάθε παίκτη μία τιμή για το αντικείμενο (μπορεί η τιμή αυτή να διαφέρει από παίκτη σε παίκτη) και ο παίκτης μπορεί είτε να δεχτεί την τιμή αυτή και να αγοράσει το αντικείμενο ή να την απορρίψει.

Διαισθητικά, για να καταλάβουμε έναν posted price μηχανισμό, αρκεί να φανταστούμε πως πάνω στο κάθε αντικείμενο υπάρχει ένα post-it με μια τιμή και η συμφωνία μας με τους παίκτες είναι ότι είτε θα το πάρουν σε αυτή την τιμή που τους το δίνουμε ή θα το αφήσουν και δεν θα έχουν δυνατότητα να το αγοράσουν σε καμία άλλη φάση της δημοπρασίας (take-it-or-leave-it). Στη συνέχεια αυτής της εργασίας θα δούμε πώς οι posted-price μηχανισμοί είναι οι μηχανισμοί που χρησιμοποιούνται κατά κύριο λόγο στις συνδυαστικές δημοπρασίες.

Θα αναφερθούμε τώρα σε ένα πολύ πρόσφατο αποτέλεσμα του Shengwu Li [13]. Ως τώρα έχουμε εστιάσει στις second price δημοπρασίες και στους posted prices μηχανισμούς. Μεταξύ των δύο μηχανισμών παρατηρούμε διαισθητικά το εξής: αν προσπαθήσουμε να εξηγήσουμε σε κάποιον (όχι ειδικό) τη στρατηγική που πρέπει να ακολουθήσει ώστε να έχει το μέγιστο δυνατό κέρδος του σε μία second price δημοπρασία [6] είναι δυσκολότερο από το να του εξηγήσουμε τη στρατηγική για το βέλτιστο κέρδος σε έναν posted-price μηχανισμό. Ο Shengwu Li έδωσε έναν ορισμό, για ποιο λόγο συμβαίνει το παραπάνω.

Ορισμός 1.17 (Προφανώς Μη-Στρατηγικός - Obviously Strategy-Proof)**.** *Ένας μηχανισμός ονομάζεται προφανώς μη-στρατηγικός (obviously strategy-proof (OSP))* όταν η βελτιστότητα του να έχει ο παίκτης ως στρατηγική την αλήθεια (truthtelling) μπορεί να εξαχθεί χωρίς *δικαιολόγηση ενδεχομένων (contigent reasoning)*.

Διαισθητικά, η *δικαιολόγηση ενδεχομένων* σημαίνει ότι πρέπει να κρατάμε πληροφορίες για προηγούμενες αποφάσεις και προηγούμενα αποτελέσματα, προκειμένου να μπορέσουμε να βασίσουμε σε αυτά την τωρινή μας απόφαση. Με βάση τα παραπάνω, είναι εύκολο να συμπεράνει κανείς ότι η οικογένεια των posted price μηχανισμών εξ' ορισμού δεν απαιτεί δικαιολόγηση ενδεχομένων, αφού οι αποφάσεις μας για τα αντικείμενα είναι take-it-or-leave-it ανάλογα με την τιμή που έχουν εκείνη την ώρα που έχει έρθει η σειρά μας. Επομένως, εξ' ορισμού και πάλι, η οικογένεια των posted-price μηχανισμών είναι OSP, εν αντιθέσει με τις second-price δημοπρασίες.

[6] Ουσιαστικά, να του εξηγήσουμε γιατί το να πει το αληθινό του valuation ως bid είναι η βέλτιστη στρατηγική.

1.5 Οργάνωση της Εργασίας

Σε αυτή την ενότητα θα παραθέσουμε την οργάνωση της διπλωματικής αυτής εργασίας.

Το Κεφάλαιο 1 είναι μία γενική εισαγωγή. Παραθέτουμε πολύ βασικά θεωρήματα και αποδείξεις για Προσεγγιστικούς Αλγορίθμους, Online Αλγόριθμους, καθώς και τα βασικότερα θεωρήματα και συμπεράσματα από την Αλγοριθμική Θεωρία Παιγνίων. Επιπλέον, δίνουμε το γενικό πλαίσιο περιγραφής προβλημάτων Γραμμικού Προγραμματισμού, καθώς και μία πρώτη περιγραφή Συνδυαστικών Δημοπρασιών (Combinatorial Auctions).

Στο Κεφάλαιο 2 δίνουμε το πρώτο βασικό βιβλιογραφικό συστατικό της διπλωματικής αυτής: τα Multiplicative Updates. Για το σκοπό αυτό, αναλύουμε αρχικά το γνωστό meta-algorithm για τα Multiplicative Weights Update, ένα αποτέλεσμα δομικό για τη Θεωρία της Εκμάθησης. Στη συνέχεια, περιγράφουμε αρκετές διάσημες εφαρμογές του αλγορίθμου αυτού προκειμένου να καταδείξουμε τη σημασία του με βάση τη διεθνή βιβλιογραφία. Στο τέλος αυτού του Κεφαλαίου βρίσκεται μία πρόσφατη εφαρμογή της ιδέας των Multiplicative Updates σε Combinatorial Auctions, μία εφαρμογή που έδωσε ώθηση και για τη δική μας έρευνα στο πεδίο αυτό.

Στο Κεφάλαιο 3 δίνουμε τον δεύτερο βιβλιογραφικό πυλώνα: τα budgeted settings και το Liquid Welfare. Εστιαζόμαστε σε έναν μηχανισμό του Dobzinski ο οποίος δίνει πολύ καλό approximatio guarantee για την περίπτωση που θέλουμε να δημοπρατήσουμε ένα διαιρέσιμο αντικείμενο σε bidders με private budgets. Αναλύουμε τον μηχανισμό αυτό και δίνουμε αναλυτικό παράδειγμα εκτέλεσης. Η ιδέα του Dobzinski χρησιμοποιείται στη συνέχεια της διπλωματικής αυτής εργασίας προκειμένου να προσεγγίσουμε το ίδιο καλά Combinatorial Auctions.

Τέλος, στο Κεφάλαιο 4 αναλύουμε τη δική μας έρευνα και τα συμπεράσματά μας στην προσέγγιση του Liquid Welfare σε Combinatorial Auctions και παραθέτουμε τις σκέψεις σχετικά με τις πιθανές επεκτάσεις της έρευνάς μας.

Κεφάλαιο 2

Η μέθοδος των Πολλαπλασιαστικών Συντελεστών

2.1 Εισαγωγή

Ας υποθέσουμε ότι έχετε ένα πρόβλημα και πρέπει να πάρετε μία δύσκολη απόφαση. Πώς παίρνετε αυτή την απόφαση; Χωρίς να ξέρετε το ακριβές αντίκτυπο της κάθε μίας πιθανής απόφασης, διαλέγετε αυτή που *σε εκείνη τη χρονική στιγμή* **μοιάζει** να είναι η καλύτερη. Αλλά είναι **όντως** η καλύτερη· Και ακόμη και αν αποδειχτεί ότι ήταν, *πώς* μπορούσατε να το ξέρετε αυτό εξαρχής, χωρίς καμία προηγούμενη γνώση των αντικτύπων (outcomes); Αυτή είναι μία δύσκολη ερώτηση, οπότε ας την επαναδιατυπώσουμε λίγο: αν είχατε τη δυνατότητα να μην αποφασίσετε κατευθείαν, αλλά να επαναλάβετε τη διαδικασία της απόφασης για κάποιους γύρους και σε κάθε γύρο να αποκτάτε περισσότερη γνώση σχετικά με την απόφαση, θα επηρέαζε αυτή η επανάληψη την ικανότητά σας να πάρετε την καλύτερη απόφαση; Η απάντηση είναι «ναι» και έχει διαπιστωθεί πολλές φορές, από πολλούς ερευνητές. Αυτή ακριβώς είναι η ιδέα πίσω από τους πολλαπλασιαστικούς συντελεστές (multiplicative updates): σε κάθε γύρο, έχετε τη δυνατότητα να *ανανεώσετε* την κρίση σας για κάποια απόφαση, πολλαπλασιάζοντας μία ορισμένη ποσότητα, η οποία αναπαριστά τη γνώση σας. Ή, σε πιο μαθηματική ανάλυση, μία κατανομή πιθανοτήτων διατηρείται πάνω σε κάποιο σύνολο και σε κάθε βήμα, η πιθανότητα που έχει ανατεθεί στο i πολλαπλασιάζεται ή διαιρείται με το $(1 + \epsilon C(i))$, όπου το $C(i)$ είναι το *κέρδος* για το στοιχείο i.

Πολλοί αλγόριθμοι σε διαφορετικά επιστημονικά πεδία χρησιμοποιούν την ιδέα της διατήρησης μίας κατανομής πιθανοτήτων πάνω σε ένα συγκεκριμένο σύνολο και χρησιμοποιούν τον Κανόνα των Πολλαπλασιαστικών Συντελεστών (multiplicative update rule) ώστε να ανανεώνουν επαναληπτικά τα βάρη της κατανομής αυτής. Το ενδιαφέρον εδώ είναι ότι η ανάλυση των αλγορίθμων αυτών, αν και ανήκουν σε διαφορετικά επιστημονικά πεδία, είναι πολύ παρόμοια και συχνά βασίζεται σε μία εκθετική συνάρτηση δυναμικού (potential function). Οι Sanjeev Arora, Elad Hazan και Satyen Kale σχέδιασαν έναν απλό γενικό μετα-αλγόριθμο που ενοποιεί όλες τις διαφορετικές προσεγγίσεις ([14]) και θα τον παρουσιάσουμε σε αυτό το κεφάλαιο.

Σε αυτό το κεφάλαιο, θα αναλύσουμε αρχικά τον αλγόριθμο της μεθόδου αυτής ([14]) καθως και μερικές από τις πιο γνωστές εφαρμογές της στη βελτιστοποίηση (optimization) και στη μηχανική εκμάθηση (learning).

Προκειμένου να παρουσιάσουμε τον αλγόριθμο είναι απαραίτητο να εισάγουμε πρώτα τον Αλγόριθμο της Πλειοψηφίας με Βάρη (Weighted Majority Algorithm) [15].

2.2 Ο Αλγόριθμος της Πλειοψηφίας με Βάρη (Weighted Majority Algorithm)

Για να περιγράψουμε τον αλγόριθμο, θα ασχοληθούμε το *πρόβλημα της πρόβλεψης κατόπιν συμβουλής ειδικών* (*Prediction from Expert Advice Problem*), όπου θέλουμε να προβλέψουμε την κίνηση μιας μετοχής («πάνω» ή «κάτω»), μην έχοντας καμία άλλη γνώση πέραν των προβλέψεων κάποιων ειδικών (experts). Για τις ανάγκες της ανάλυσής μας, ας υποθέσουμε ότι κάθε φορά που κατόπιν των συμβουλών προβλέπουμε λάθος την κίνηση της μετοχής θα χάνουμε 1 ευρώ. Ο αλγόριθμος που θα παρουσιαστεί παρακάτω μας επιτρέπει ουσιαστικά να περιορίσουμε τα χρήματα που χάνουμε, ώστε να είναι περίπου ίσα με αυτά που χάνει ο **καλύτερος** από όλους τους experts. Αξίζει να παρατηρήσουμε εδώ ότι αυτό ίσως φαίνεται αρκετά δύσκολο έως ακατόρθωτο στην αρχή, δεδομένου ότι δεν μπορούμε να ξέρουμε *ποιος* είναι ο καλύτερος expert.

Ο αλγόριθμος, λοιπόν, αυτό που κάνει είναι να διατηρεί μία κατανομή βαρών (weighting) πάνω στους experts. Αρχικά, όλοι οι experts έχουν το ίδιο βάρος. Όσο, όμως, κάποιοι experts προβλέπουν σωστά την κίνηση της μετοχής, τόσο αυξάνουμε το βάρος τους, δηλώνοντας ότι τους εμπιστευόμαστε περισσότερο, ενώ όσοι προβλέπουν λάθος την κίνηση, μειώνουμε το βάρος τους, κάτι που δηλώνει ότι τους εμπιστευόμαστε λιγότερο.

Θεωρούμε ότι w_i^t είναι το βάρος του i-οστού expert (όπου $i = 1, 2, \ldots, n$) στον γύρο t (όπου $t = 1, 2, \ldots, T$).

Initialization: Fix an $\epsilon \le 1/2$. Initial weights of experts $w_i^1 = 1, \forall i \in \{1, 2, \ldots, n\}$
.

At every round t:

1. If the total weight of all experts predicting "up" is at least $\sum_i w_i^t/2$ then we predict "up" as well and otherwise we predict "down". Break ties arbitrarily.
   Our prediction is the opinion of a weighted majority of the experts.

2. For each i such that expert i predicted incorrectly the movement of the stock, we set

$$w_i^{t+1} = (1 - \epsilon) w_i^t$$

Algorithm 2: Weighted Majority Algorithm

Θεώρημα 2.1. *Μετά από t βήματα, έστω m_i^t ο αριθμός των λαθών του expert i και m^t ο αριθμός των λαθών που έχει κάνει ο αλγόριθμος. Τότε, έχουμε το ακόλουθο φράγμα, για κάθε i:*

$$m^t \le \frac{2 \ln n}{\epsilon} + 2(1+\epsilon) m_i^t$$

Συγκεκριμένα, το παραπάνω ισχύει και για τον καλύτερο expert, δηλαδή ισχύει για το ελάχιστο δυνατό m_i.

Παρατηρήσεις. Βλέπουμε στο παραπάνω φράγμα, ότι αν το m_i^t είναι μηδενικό (στην ακραία περίπτωση όπου ο καλύτερος expert δεν κάνει κανένα λάθος), τότε κυριαρχεί ο όρος $\frac{2 \ln n}{\epsilon}$. Ο όρος αυτός εξαρτάται μόνο από την παράμετρο ϵ και το πλήθος των experts και αναπαριστά τα λάθη, τα οποία δεν μπορεί να αποφύγει να κάνει ο αλγόριθμος στη χειρότερη περίπτωση.

Η απόδειξη του θεωρήματος βασίζεται στη χρήση της συνάρτησης δυναμικού (potential function) και παρουσιάζεται παρακάτω.

Απόδειξη. Αρχικά, μία απλή αναγωγή μπορεί να μας δώσει:

$$w_i^t = (1-\epsilon)^{m_i^t} \tag{2.1}$$

Έστω $\Phi^t = \sum_i w_i^t$ η potential function. Λόγω των αρχικών βαρών που δώσαμε στους experts έχουμε ότι: $\Phi^1 = n$. Κάθε φορά που κάνουμε ένα λάθος, τουλάχιστον το *μισό* από το συνολικό βάρος μειώνεται κατά έναν παράγοντα $1-\epsilon$ και συνεπώς, η συνάρτηση δυναμικού μειώνεται κατά τουλάχιστον $(1-\epsilon/2)$. Πράγματι, αν σε κάποιο γύρο το πλήθος των experts που κάνουν λανθασμένη πρόβλεψη είναι $k \geq n/2$ τότε για την potential function έχουμε:

$$\Phi^{(t+1)} \leq \Phi^t \left(\frac{n-k}{n} + \frac{k}{n}(1-\epsilon)\right) = \Phi^t \left(1 - \frac{k}{n}\epsilon\right) \leq \Phi^t \left(1 - \frac{\epsilon}{2}\right) \tag{2.2}$$

Συνεπώς, από την παραπάνω σχέση, με μία αναγωγή έχουμε

$$\Phi^t \leq \Phi^1 (1-\epsilon/2)^{m^t} = n(1-\epsilon)^{m^t} \tag{2.3}$$

Για κάθε expert i θα ισχύει:

$$\begin{aligned} \Phi^{(t+1)} &\geq w_i^{(t+1)} \xRightarrow{2.1,2.2} \\ n(1-\epsilon/2) &\geq (1-\epsilon)^{m_i^t} \xRightarrow{\ln(\cdot)} \\ m^t \ln(1-\epsilon/2) + \ln(n) &\geq m_i^t \ln(1-\epsilon) \Rightarrow \\ m^t(-\ln(1-\epsilon/2)) &\leq \ln(n) + m_i^t(-\ln(1-\epsilon)) \end{aligned} \tag{2.4}$$

Για να περάσουμε από τη σχέση 2.4 στο Θεώρημα, πρέπει να συγκρίνουμε τα δύο μέλη της παραπάνω σχέσης. Για το αριστερό μέλος, παρατηρούμε ότι για το ανάπτυγμα της σειράς Taylor του $\ln(1-x)$ έχουμε: $\ln(1-x) = -x - \frac{1}{2}x^2 - \frac{1}{3}x^3 - \ldots$, για $|x| < 1$. Άρα, δεδομένου ότι $|\epsilon| < 1/2$ αντικαθιστούμε το αριστερό μέλος με το ανάπτυγμα Taylor και έχουμε ότι:

$$m^t(-\ln(1-\epsilon/2)) = m^t \left(\frac{\epsilon}{2} + \frac{1}{2}\frac{\epsilon^2}{4} + \ldots\right) \geq m^t \left(\frac{\epsilon}{2}\right) \tag{2.5}$$

Για το δεξί μέλος, τώρα, θα κάνουμε χρήση της γνωστής ανισότητας: $-\ln(1-x) \leq x + x^2$, για $x < 1/2$. Άρα, δεδομένου και πάλι ότι $|\epsilon| < 1/2$ έχουμε ότι:

$$\ln(n) + m_i^t(-\ln(1-\epsilon)) \leq \ln(n) + m_i^t(\epsilon + \epsilon^2) \tag{2.6}$$

Άρα, η σχέση 2.4 με τη χρήση των 2.5, 2.6 γίνεται:

$$m^t(\epsilon/2) \leq \ln(n) + m_i^t(\epsilon+\epsilon^2) \Rightarrow m^t \leq \frac{2\ln(n)}{\epsilon} + 2\ln(1+\epsilon)m_i^t$$

■

Αξίζει να τονιστεί εδώ ότι η παραπάνω ανάλυση δεν χρησιμοποιεί καμία υπόθεση σχετικά με τη σειρά με την οποία γίνονται τα γεγονότα: θα μπορούσαν, δηλαδή, να είναι αυθαίρετα συσχετισμένα και ακόμη και να εξαρτώνται από το τρέχον βάρος των experts.

2.3 Ο Αλγόριθμος της Μεθόδου Πολλαπλασιαστικών Συντελεστών (Multiplicative Weights Update Algorithm)

Στο γενικό πρόβλημα, και πάλι έχουμε n experts, οι οποίοι κάνουν προβλέψεις. Ωστόσο, αυτό που διαφέρει εδώ είναι ότι το σύνολο των πιθανών αποτελεσμάτων δεν είναι πια δυαδικό («πάνω»/«κάτω») και μπορεί μάλιστα να είναι μη-πεπερασμένο. Έστω, λοιπόν, ότι σε κάθε γύρο ακολουθούμε την πιο απλοϊκή και αφελής στρατηγική του να διαλέξουμε *τυχαία* έναν expert από το σύνολο των n και να ακολουθήσουμε την πρόβλεψή του. Τότε, η αναμενόμενη χρηματική μας απώλεια θα είναι τόση όση αυτή του «μέσου» (average) expert. Αν, όμως, κάποιοι experts είναι εμφανώς καλύτεροι στις προβλέψεις τους από ό, τι κάποιοι άλλοι, τότε δεν είναι λογικό να θέλουμε αυτοί να έχουν μεγαλύτερη πιθανότητα να επιλεγούν από εμάς, για να μας δώσουν την πρόβλεψή τους; Ο κανόνας των πολλαπλασιαστικών συντελεστών (multiplicative weight update rule) είναι, με άλλα λόγια, ο τρόπος μας φτιάχνουμε την κατανομή πιθανοτήτων στους experts.

Θα συμβολίζουμε με $\mathbf{P}$ το σύνολο όλων των αποτελεσμάτων/γεγονότων και με $\mathbf{M}$ τον πίνακα του οποίου η θέση $\mathbf{M}(i,j)$ θα συμβολίζει την *ποινή* (penalty) που ο expert i θα πληρώνει όταν το αποτέλεσμα είναι $j \in \mathbf{P}$. Υποθέτουμε ότι για κάθε expert i και για κάθε γεγονός j, το $\mathbf{M}(i,j)$ θα βρίσκεται στο διάστημα $[-l, \rho]$, όπου $l \le \rho$ [1]

At every step t, there is a weight w_i^t assigned to expert i.
Initialization: Weight $w_i^1 = 1$ for all experts.
For each time step t:

1. Assosiate the distribution $\mathcal{D}^t = \{p_1^t, p_2^t, \ldots, p_n^t\}$ on the experts, where
$$p_i^t = \frac{w_i^t}{\sum_k w_k^t}$$
2. Pick an expert according to the distribution $\mathcal{D}^t$ and use it to make the prediction.
3. Based on the outcome $j \in \mathbf{P}$ in round t, at step $t+1$, the weight of expert i is updated as follows for each i:
$$w_i^{(t+1)} = \begin{cases} w_i^t(1-\epsilon)^{\mathbf{M}(i,j^t)/\rho} & \text{if } \mathbf{M}(i,j^t) \ge 0 \\ w_i^t(1+\epsilon)^{-\mathbf{M}(i,j^t)/\rho} & \text{if } \mathbf{M}(i,j^t) < 0 \end{cases}$$

Algorithm 3: Multiplicative Weights Update Algorithm

Παρατηρήσεις. Όπως είναι προφανές από τον παραπάνω αλγόριθμο η κατανομή των πιθανοτήτων πάνω στους experts αλλάζει σε κάθε γύρο. Κάθε p_i^t ουσιαστικά συμβολίζει το πόσο έχει επηρεάσει ο expert i την απόφαση που θα ληφθεί.

Η αναμενόμενη ποινή (penalty) για το αποτέλεσμα $j^t \in \mathbf{P}$ είναι
$$\sum_i p_i^t \mathbf{M}(i,j^t) = \sum_i w_i^t \mathbf{M}(i,j^t) / \sum_i w_i^t$$

[1] Η ανάλυση που θα παρουσιαστεί, ωστόσο, δουλεύει ακόμη και αν υπάρχουν δύο είδη από experts: κάποιοι για τους οποίους όλα τα κέρδη (payoffs) είναι στο διάστημα $[-l, \rho]$ και κάποιοι για τους οποίους όλες οι ποινές (penalties) είναι στο διάστημα $[-\rho, l]$.

το οποίο το συμβολίζουμε ως $\mathbf{M}(\mathcal{D}^t, j^t)$.

Θεώρημα 2.2. *Έστω $\epsilon \leq 1/2$. Μετά από T γύρους, για κάθε expert i, έχουμε:*

$$\sum_i \mathbf{M}(\mathcal{D}^t, j^t) \leq \frac{\rho \ln n}{\epsilon} + (1+\epsilon) \sum_{\geq 0} \mathbf{M}(i, j^t) + (1-\epsilon) \sum_{<0} \mathbf{M}(i, j^t)$$

όπου οι δείκτες ≥ 0 και < 0 αναφέρονται στους γύρους t όπου $\mathbf{M}(i, j^t)$ είναι ≥ 0 και < 0 αντίστοιχα.

Απόδειξη. Χρησιμοποιούμε τις ακόλουθες γνωστές ανισότητες, οι οποίες προκύπτουν από την κυρτότητα της εκθετικής συνάρτησης:

$$(1-\epsilon)^x \leq (1-\epsilon x), \text{ if } x \in [0,1] \tag{2.7}$$

$$(1+\epsilon)^{-x} \leq (1-\epsilon x), \text{ if } x \in [-1,0] \tag{2.8}$$

Η απόδειξη ακολουθεί τα βήματα της προηγούμενης απόδειξης και χρησιμοποιεί τη συνάρτηση δυναμικού $\Phi^t = \sum_i w_i^t$. Εφόσον $\mathbf{M}(i, j^t)/\rho \in [-1, 1]$ χρησιμοποιώντας τις σχέσεις 2.7 και 2.8 έχουμε:

$$\begin{aligned}
\Phi^{t+1} &= \sum_i w_i^{t+1} \\
&= \sum_{i:\mathbf{M}(i,j^t)\geq 0} w_i^t (1-\epsilon)^{\mathbf{M}(i,j^t)/\rho} + \sum_{i:\mathbf{M}(i,j^t)<0} w_i^t (1+\epsilon)^{-\mathbf{M}(i,j^t)/\rho} \\
&\leq \sum_i w_i^t (1 - \epsilon \mathbf{M}(i, j^t)/\rho) \\
&= \Phi^t - \frac{\epsilon \Phi^t}{\rho} \sum_i p_i^t \mathbf{M}(i, j^t) \\
&= \Phi^t (1 - \epsilon \mathbf{M}(\mathcal{D}^t, j^t)/\rho) \\
&\leq \Phi^t e^{-\epsilon \mathbf{M}(\mathcal{D}^t, j^t)/\rho}
\end{aligned} \tag{2.9}$$

όπου και χρησιμοποιήσαμε το γεγονός ότι $p_i^t = w_i^t/\Phi^t$, το οποίο προκύπτει από το πώς το ορίσαμε στον αλγόριθμο. Μετά από T γύρους, έχουμε $\Phi^T \leq \Phi^1 e^{-\epsilon \sum_t \mathbf{M}(\mathcal{D}^t, x^t)/\rho}$. Επιπλέον, για κάθε i:

$$\Phi^T \geq w_i^T = (1-\epsilon)^{\sum_{\geq 0} \mathbf{M}(i,j^t)/\rho} \cdot (1+\epsilon)^{-\sum_{<0} \mathbf{M}(i,j^t)/\rho} \tag{2.10}$$

Χρησιμοποιώντας τις ανισώσεις: $\ln\left(\frac{1}{1-\epsilon} \leq \epsilon + \epsilon^2\right)$ και $\ln(1+\epsilon) \geq \epsilon - \epsilon^2$ για $\epsilon \leq 1/2$ και εφαρμόζοντας τον λογάριθμο και στα δύο μέλη προκύπτει το ζητούμενο. [2] ■

[2] Είναι προφανές από την απόδειξη ότι ο update rule θα μπορούσε κάλλιστα να είναι και

$$w_i^{t+1} = w_i^t (1 - \epsilon \mathbf{M}(i, j^t))$$

άσχετα από το πρόσημο του $\mathbf{M}(i, j^t)$. Ένας τέτοιος κανόνας ίσως να ήταν πιο πρακτικός στην υλοποίηση και επίσης χρησιμοποιείται στην ανάλυση ορισμένων αλγορίθμων, όπως το SET COVER.

Πόρισμα 2.1. *Έστω $\delta > 0$ μία παράμετρος λάθους. Τότε, για $\epsilon \leq \min\{\frac{\delta}{4l}, \frac{1}{2}\}$, μετά από $T = \frac{2\rho \ln n}{\epsilon\delta}$ γύρους, έχουμε το ακόλουθο (αθροιστικό και πολλαπλασιαστικό) (additive and multiplicative) φράγμα για την μέση αναμενόμενη απώλεια (average expected loss): Για κάθε expert i:*

$$\frac{\sum_t \mathbf{M}(\mathcal{D}^t, j^t)}{T} \leq \delta + (1 \pm \epsilon)\frac{\sum_t \mathbf{M}(i, k^t)}{T} \tag{2.11}$$

όπου το πρόσημο $+$ ή $-$ εξαρτάται από το αν $\mathbf{M}(i,j) \in [-l, \rho]$ ή $[-\rho, l]$ αντίστοιχα.

Απόδειξη. Θα αποδείξουμε την περίπτωση $\mathbf{M}(i,j) \in [-l, \rho]$ [3] Σε αυτή την περίπτωση, έχουμε:

$$(1-\epsilon)\sum_{<0}\mathbf{M}(i,j^t) \leq (1+\epsilon)\sum_{<0}\mathbf{M}(i,j^t) + 2\epsilon l T \tag{2.12}$$

Αντικαθιστώντας την 2.12 στην ανισότητα του Θεωρήματος και διαιρώντας δια T προκύπτει:

$$\frac{\sum_t \mathbf{M}(\mathcal{D}^t, j^t)}{T} \leq \frac{\rho \ln n}{\epsilon T} + 2\epsilon l + (1+\epsilon)\frac{\sum_t \mathbf{M}(i, j^t)}{T} \tag{2.13}$$

Το φράγμα που ψάχνουμε προκύπτει στη συνέχεια από την κατάλληλη επιλογή των ϵ και T. ■

Πόρισμα 2.2. *Έστω δ μία παράμετρος λάθους. Τότε, για $\epsilon = \min\left\{\frac{\delta}{4\rho}, \frac{1}{2}\right\}$ μετά από $T = \frac{16\rho^2 \ln n}{\delta^2}$ γύρους, έχουμε το ακόλουθο αθροιστικό (additive) φράγμα πάνω στην μέση αναμενόμενη απώλεια (average expected loss): Για κάθε expert i:*

$$\frac{\sum_t \mathbf{M}(\mathcal{D}^t, j^t)}{T} \leq \delta + \frac{\sum_t \mathbf{M}(i, j^t)}{T} \tag{2.14}$$

Ένα αντίστοιχο δυαδικό αποτέλεσμα (dual) μπορεί να επιτευχθεί για την περίπτωση που ο πίνακας $\mathbf{M}$ συμβολίζει *κέρδη* αντί για *απώλειες* (gains instead of losses). Σε αυτή την περίπτωση, το μόνο που πρέπει να κάνουμε είναι να τρέξουμε τον προηγούμενο αλγόριθμο με τον πίνακα $\mathbf{M}' = -\mathbf{M}$ αντί για τον πίνακα $\mathbf{M}$. Και έτσι με την ίδια αντικατάσταση καταλήγουμε στο δυαδικό του προηγούμενου θεωρήματος, το οποίο είναι το εξής:

$$\sum_t \mathbf{M}(\mathcal{D}^t, j^t) \geq -\frac{\rho \ln n}{\epsilon} + (1-\epsilon)\sum_{\geq 0}\mathbf{M}(i,j^t) + (1+\epsilon)\sum_{<0}\mathbf{M}(i,j^t)$$

[3] Η άλλη περίπτωση αποδεικνύεται ανάλογα.

2.4 Εφαρμογές της Μεθόδου Πολλαπλασιαστικών Συντελεστών

Ο τρόπος με τον οποίο χρησιμοποιείται συνήθως η μέθοδος των multiplicative updates είναι ο ακόλουθος: Έστω ότι η απόδοση ενός αλγορίθμου εξαρτάται από το πώς αυτός ο αλγόριθμος υλοποιεί κάποιους συγκεκριμένους περιορισμούς. Μπορούμε να το σκεφτόμαστε αυτό το πλαίσιο, όπως τα προβλήματα LP που είδαμε στην Εισαγωγική Ενότητα. Σε τέτοια προβλήματα, ο κάθε **expert** αναπαριστά και από έναν **περιορισμό** και τα **γεγονότα** αντιστοιχούν σε **σημεία** στο πεδίο που μας ενδιαφέρει. Η **ποινή** κάθε expert είναι ανάλογη με το πόσο καλά ικανοποιείται ο κάθε περιορισμός στο εκάστοτε σημείο (που αναπαρίσταται από ένα γεγονός). Αυτό φαίνεται να είναι αντίθετο με τη διαίσθησή μας, αλλά ας θυμηθούμε εδώ ότι στον αλγόριθμο μειώνουμε το βάρος του expert ανάλογα με την ποινή του και εάν ο περιορισμός ενός expert είναι καλά ικανοποιημένος πάνω στα έως-τώρα γεγονότα, εμείς θα θέλαμε το βάρος του να γίνει μικρότερο, έτσι ώστε ο αλγόριθμος να μπορεί να εστιάσει στους experts των οποίων οι περιορισμοί δεν έχουν ικανοποιηθεί επαρκώς. Με αυτά τα βάρη, ο αλγόριθμος παράγει ένα *μεγιστικά αντίπαλο (maximally adversarial)* γεγονός, εφόσον μεγιστοποιεί την αναμενόμενη τιμή.

Όπως μπορούμε να συμπεράνουμε αυτή η πορεία σκέψης μας οδηγεί σε πολλές εφαρμογές του αλγορίθμου μερικές από τις οποίες θα συζητήσουμε παρακάτω.

2.4.1 Προσεγγιστική Επίλυση των Παιχνιδιών Μηδενικού Αθροίσματος (Zero-Sum Games)

Η μέθοδος των multiplicative updates έχει πολλές εφαρμογές στην Αλγοριθμική Θεωρία Παιγνίων. Μία από αυτές είναι στα παιχνίδια μηδενικού αθροίσματος (zero-sum games).

Ορισμός 2.1 (Zero-Sum Games)**.** Στη θεωρία παιγνίων, ένα παιχνίδι *μηδενικού αθροίσματος (zero-sum)* είναι μία μαθηματική αναπαράσταση ενός παιγνίου όπου το κέρδος (ή η ζημία) του utility ενός παίκτης εξισορροπούνται απόλυτα από τη ζημία (ή το κέρδος αντίστοιχα) των άλλων παικτών. Αν προσθέσουμε, δηλαδή, τα συνολικά κέρδη και αφαιρέσουμε τις συνολικές ζημίες, θα έχουμε μηδενικό αποτέλεσμα.

Θα δείξουμε στη συνέχεια ότι ο γενικός αλγόριθμος που παρουσιάστηκε προηγούμενως, μπορεί να χρησιμοποιηθεί προκειμένου να λύσουμε προσεγγιστικά zero-sum παιχνίδια. Στο ίδιο αποτέλεσμα κατέληξαν και οι Freud και Schapire ([16]) απλώς χρησιμοποιήσαν διαφορετική απόδειξη σύγκλισης.

Έστω, λοιπόν, ότι $\mathbf{M}$ είναι ο πίνακας των κερδών ενός πεπερασμένου παιχνιδιού zero-sum δύο παικτών, έτσι ώστε όταν ο παίκτης-γραμμή παίζει τη στρατηγική i και ο παίκτης-στήλη τη στρατηγική j, τότε το κέρδος για τον παίκτη-στήλη είναι $\mathbf{M}(i,j)$. Έστω επίσης, ότι $\mathbf{M}(i,j) \in [0,1]$. Στόχος μας είναι τώρα να υπολογίσουμε προσεγγιστικά την αξία του παιχνιδιού, η οποία με βάση το θεώρημα MinMax του von Neumann ισούται με:

$$\lambda^* = \min_{\mathcal{D}} \max_{j} \mathbf{M}(\mathcal{D}, j) = \max_{\mathcal{P}} \min_{i} \mathbf{M}(i, \mathcal{P}) \tag{2.15}$$

όπου το $\mathcal{D}$ (και αντίστοιχα, το $\mathcal{P}$ κινείται πάνω όλες τις κατανομές των γραμμών (αντίστοιχα, των στηλών) και το j (αντίστοιχα το i) κινείται πάνω σε όλες τις στήλες (αντίστοιχα, γραμμές) και τέλος, ο συμβολισμός $\mathbf{M}(\mathcal{D}, j)$ αντιστοιχεί στο $\mathbb{E}_{i\in\mathcal{D}}[\mathbf{M}(i,j)]$.

Έστω, τώρα, δ μία παράμετρος λάθους. Αυτό που θέλουμε εμείς ουσιαστικά είναι να λύσουμε το παιχνίδι zero-sum με το πολύ αθροιστικό λάθος του δ. Δηλαδή, να βρούμε mixed στρατηγικές [4] γραμμών-στηλών $\mathcal{D}_{final}, \mathcal{P}_{final}$ αντίστοιχα, τέτοιες ώστε:

$$\lambda^* - \delta \leq \min_i \mathbf{M}(i, \mathcal{P}_{final}) \tag{2.16}$$

$$\max_j \mathbf{M}(\mathcal{D}_{final}, j) \leq \lambda^* + \delta \tag{2.17}$$

Προκειμένου τώρα να αντιστοιχήσουμε το υπάρχον μας πρόβλημα με τον αλγόριθμο που δείξαμε στην προηγούμενη ενότητα, θεωρούμε αρχικά ότι οι **experts** μας είναι πλέον οι **αμιγείς (pure) στρατηγικές του παίκτη-γραμμή**. Συνεπώς, μία **κατανομή πάνω στους experts** αντιστοιχεί **μεικτή στρατηγική-γραμμής**. Τα **γεγονότα** αντιστοιχούν σε **αμιγείς στρατηγικές του παίκτη-στήλη**. Η ποινή που πληρώνει ένας expert i, όταν συμβαίνει το γεγονός j είναι $\mathbf{M}(i, j)$. Η αλγοριθμική υπόθεση σχετικά με το παιχνίδι είναι ότι δεδομένης οποιασδήποτε κατανομής $\mathcal{D}$ πάνω στους experts, εμείς μπορούμε να βρούμε με αποδοτικό τρόπο το καλύτερο δυνατό γεγονός, δηλαδή την αμιγή στρατηγική-στήλη j, ώστε να μεγιστοποιείται το $\mathbf{M}(\mathcal{D}, j)$. Αυτή η ποσότητα είναι τουλάχιστον λ^* με βάση την 2.17. Οι ποινές για τους experts (οι οποίες είναι ίδιες με τα κέρδη) βρίσκονται στο διάστημα $[0, 1]$. Τότε, από το 2.2 αν χρησιμοποιήσουμε για παραμέτρους τα $l = 0$, $w = 1$ και αν θέσουμε $\epsilon = \delta/4$. Τρέχουμε το παιχνίδι για $T = \frac{16\ln(n)}{\delta^2}$ γύρους, όπως ακριβώς στο Πόρισμα 2.2.

Για οποιαδήποτε κατανομή $\mathcal{D}$ πάνω στις στρατηγικές-γραμμές, έχουμε ότι $\sum_t \mathbf{M}(\mathcal{D}, j^t) \geq \min_i \sum_t \mathbf{M}(i, j^t)$. Επιπλέον, για κάθε t, $\mathbf{M}(\mathcal{D}^t, j^t) \geq \lambda^*$. Συνεπώς, μετά από T γύρους, έχουμε ότι για οποιοδήποτε $\mathcal{D}$:

$$\lambda^* \leq \frac{\sum_{t=1}^T \mathbf{M}(\mathcal{D}^t, j^t)}{T} \leq \delta + \min_i \left\{ \frac{\sum_{i=1}^T \mathbf{M}(i, j^t)}{T} \right\} \leq \delta + \frac{\sum_{t=1}^T \mathbf{M}(\mathcal{D}, j^t)}{T} \tag{2.18}$$

Αν θέσουμε $\mathcal{D} = \mathcal{D}^*$, η οποία αντιστοιχεί στη βέλτιστη στρατηγική-γραμμή, έχουμε ότι $\mathbf{M}(\mathcal{D}, j) \leq \lambda^*$, για κάθε j. Τότε, η ανισότητα 2.18 γίνεται:

$$\lambda^* \leq \frac{\sum_{t=1}^T \mathbf{M}(\mathcal{D}^t, j^t)}{T} \leq \delta + \lambda^* \tag{2.19}$$

Άρα, το $\frac{\sum_{t=1}^T \mathbf{M}(\mathcal{D}^t, j^t)}{T}$ είναι μία (αθροιστική) δ-προσέγγιση στο λ^*.

Ορίζουμε τώρα το $\mathcal{D}_{final}$ να είναι η κατανομή $\mathcal{D}^t$, η οποία έχει το ελάχιστο $\mathbf{M}(\mathcal{D}^t, j^t)$ πάνω σε όλα τα t. Έστω ότι j_{final} είναι η βέλτιστη απάντηση του παίκτη-στήλη στο $\mathcal{D}_{final}$. Από την ανισότητα 2.18 έχουμε:

$$\mathbf{M}(\mathcal{D}_{final}, j_{final}) \leq \frac{\sum_{t=1}^T \mathbf{M}(\mathcal{D}, j^t)}{T} \leq \lambda^* + \delta \tag{2.20}$$

Εφόσον για οποιοδήποτε t, το j^t μεγιστοποιεί το $\mathbf{M}(\mathcal{D}^t, j)$ πάνω σε όλα τα j, συμπεραίνουμε ότι η $\mathcal{D}_{final}$ είναι προσεγγιστικά βέλτιστη μικτή στρατηγική-γραμμή.

[4] Μία μικτή στρατηγική (mixed strategy) είναι η ανάθεση πιθανότητας σε κάθε αμιγή στρατηγική. Αυτή η ανάθεση πιθανότητας επιτρέπει στον κάθε παίκτη να διαλέξει τυχαία κάποια αμιγή στρατηγική. Εφόρον οι πιθανότητες είναι συνεχείς, υπάρχουν μη-πεπερασμένο πλήθος μικτών στρατηγικών διαθέσιμες στον κάθε παίκτη.

Ορίζουμε, τέλος, το $\mathcal{P}_{final}$ να είναι η κατανομή η οποία αναθέτει την πιθανότητα $\frac{1}{T} \times \#$(φορές που το j παίζεται στον αλγόριθμο) στην στήλη j. Και πάλι από την 2.18 έχουμε ότι για οποιαδήποτε κατανομή-γραμμής $\mathcal{D}$

$$\lambda^* - \delta \leq \frac{\sum_{t=1}^{T} \mathbf{M}(\mathcal{D}, j^t)}{T} = \mathbf{M}(\mathcal{D}, \mathcal{P}_{final}) \tag{2.21}$$

το οποίο αποδεικνύει ότι η $\mathcal{P}_{final}$ είναι προσεγγιστικά μία βέλτιστη μικτή στρατηγική-στήλη.

2.4.2 Multiplicative Updates και Συνδυαστικές Δημοπρασίες

Σε αυτή τη δουλειά ([17]) οι Krysta και Vocking σχεδίασαν truthful μηχανισμούς για Συνδυαστικές Δημοπρασίες σε ένα online μοντέλο με bidders που έρχονται σειριακά και ο τρόπος με τον οποίο διατάσσονται είναι είτε τυχαίος ή adversarial. Τα valuations των bidders δίνονται από Μαντεία Ζήτησης (Demand Oracles).

Ουσιαστικά όλες οι προηγούμενες ιδέες και προσεγγίσεις στην online ανάθεση πόρων ([18, 19, 20]) βασίζονται στην ακόλουθη ιδέα: Ο online αλγόριθμος καθορίζει τις τιμές για τα αντικείμενα και πουλάει τα αντικείμενα στους bidders στις τιμές αυτές. Αρχικά, όλα τα αντικείμενα έχουν την ίδια τιμή. Οι τιμές για τα διαφορετικά αντικείμενα αυξάνονται με το πέρασμα του χρόνου με διαφορετικές ταχύτητες ανάλογα με το πόσο «δημοφιλές» είναι το εκάστοτε αντικείμενο. Κάθε φορά που πωλείται ένα αντικείμενο, η τιμή του αυξάνεται με έναν πολλαπλασιαστικό τρόπο. Οπότε, ο αλγόριθμος «μαθαίνει» την τιμή του αντικειμένου, πωλώντας αντίγραφα αυτού. Αυτός είναι και ο λόγος για τον οποίο μια σχετικά μεγάλη πολλαπλότητα των αντικειμένων, οδηγεί σε έναν καλό λόγο προσέγγισης.

Οι συγγραφείς δίνουν σε αυτή τη δουλειά τους πρώτους online μηχανισμούς με καλούς ανταγωνιστικούς λόγους για διάφορα είδη από Συνδυαστικές Δημοπρασίες και μικρές πολλαπλότητες. Για να πετύχουν να καλύψουν ακόμη και την περίπτωση της μοναδικής πολλαπλότητας ($b = 1$) συνδυάζουν την online ανάθεση αντικειμένων, χρησιμοποιώντας αυξανόμενες τιμές με την ιδέα του oblivious randomized rounding ([21]). Η ιδεά εδώ είναι να χαλαρώσουμε τους περιορισμούς υποθέτοντας ότι το κάθε αντικείμενο είναι διαθέσιμο σε κάποια συγκεκριμένη «εικονική πολλαπλότητα». Έτσι, ο μηχανισμός μας μαθαίνει την τιμή των αντικειμένων, πωλώντας «εικονικά» αντίγραφα. Στη συνέχεια, χρησιμοποιούμε το randomized rounding (τυχαιοποιημένη στρογγυλοποίηση) προκειμένου να δούμε ποια «εικονικά» αντίγραφα, θα αντιστοιχηθούν σε πραγματικά αντίγραφα και όλη αυτή η διαδικασία γίνεται online, δηλαδή δεν εξαρτάται καθόλου από τα valuations. Αυτή η τελευταία ιδιότητα είναι κομβική για το truthfulness του μηχανισμού.

Τυπική Περιγραφή του Προβλήματος

Όπως έχουμε αναλύσει και παραπάνω, θα έχουμε ένα σύνολο U από m αντικείμενα, τα οποία θα ανατεθούν σε n bidders. Κάθε $e \in U$ έχει πολλαπλότητα $b \geq 1$, δηλαδή, μπορεί να ανατεθεί σε το πολύ b bidders. Οι bidders παίρνουν «πακέτα» (bundles) του U. Ο κάθε bidder i έχει valuation function $v_i : 2^U \to \mathbb{R}_{\geq 0}$, η οποία πληροί τις κλασσικές προϋποθέσεις, ότι δηλαδή: $v_i(\cdot)$ είναι *μη-φθίνουσα* και ότι $v_i(\emptyset) = 0$. Οι συναρτήσεις των valuations υποθέτουμε ότι δίνονται από μαύρα κουτιά (μαντεία (oracles», τα οποία ερωτώνται από τον μηχανισμό. Τα μαύρα, αυτά, κουτιά απαντούν σε *Ερωτήσεις Ζήτησης (Demand Queries)*, δηλαδή: *Με δεδομένο ένα διάνυσμα τιμών για τα αντικείμενα, $(p_e)_{e \in U}$, ποιο είναι το υποσύνολο $S_i \subseteq U$, το οποίο μεγιστοποιεί το utility και ποιο το valuation του, $v_i(S_i)$;* Το utility για το S_i κάτω από το διάνυσμα τιμών $p = (p_e)_{e \in U}$ είναι $u_i(p, S_i) = v_i(S_i) - \sum_{e \in S_i} p_e$. Οι αλγόριθμοι που θα παρουσιαστούν χρησιμοποιούν περιορισμένα oracles $D_i(U', p)$, για $U' \subseteq U$, τα οποία επιστρέφουν ένα υποσύνολο $S'_i \subseteq U'$ το οποίο μεγιστοποιεί το utility. Ένα περιορισμένο demand oracle μπορεί να προσομοιωθεί από ένα μη-περιορισμένο θέτοντας τις τιμές των αντικειμένων $U \setminus U'$ να είναι επαρκώς μεγάλοι αριθμοί. Ο τρόπος μας να πετύχουμε το truthfulness είναι με το ασχολούμαστε με τον κάθε παίκτη έναν-έναν στη σειρά και προσφέροντας στον κάθε παίκτη τα αντικείμενα σε καθορισμένες τιμές. Συγκεκριμένα, εάν το $U_i \in U$ είναι το σύνολο των αντικειμένων που ο μηχανισμός αποφασίζει να προσφέρει στον παίκτη i, τότε ο μηχανισμός καθορίζει τις τιμές $p = (p_e)_{e \in U}$ και υπολογίζει το S_i καλώντας το demand oracle $D_i(U_i, p)$. Οι τιμές στις οποίες προσφέρονται τα αντικείμενα στον παίκτη i, δεν καθορίζονται από το valuation του παίκτη i. Αυτή η προσέγγιση οδηγεί σε truthfulness με βάση τον άμεσο χαρακτηρισμό των truthful μηχανισμών ([22, 1]).

Overselling MPU Algorithm

Παρουσιάζουμε τώρα τον αρχικό αλγόριθμο. Η διαφορά του με τις προηγούμενες έρευνες ήταν ότι εδώ οι συγγραφείς επιτρέπουν να γίνεται υπερ-πώληση κάποιων αντικειμένων. Για την ακρίβεια, παραβιάζεται η προσφορά b των αντικειμένων κατά έναν παράγοντα $O(\log bm)$. Ο τελικός αλγόριθμος θα διορθώσει αυτό το «λάθος» μέσω της τυχαιοποιημένης διαδικασίας.

Θεωρούμε ότι στον αλγόριθμο δίνονται δύο παράμετροι $\mu \geq 1$ και $L > 0$, τέτοιο ώστε το L είναι το κάτω φράγμα στο μέγιστο valuation μεταξύ όλων των bidders και όλων των bundles και ότι υπάρχει το πολύ ένας bidder του οποίου το valuation να ξεπερνά το μL.

Παρατήρηση 2.1. *Για το παραπάνω πρόβλημα, ισχύει:* $L \leq v(opt)$.

Initialization: Set all items' prices at $p_0 = \frac{L}{4bm}$.

1. For each good $e \in U$ do $p_e^1 = p_0$.
2. For each bidder $i = 1, 2 \ldots, n$ do:
3. Set $S_i = D_i(U_i, p^i)$, for a suitable $U_i \subseteq U$.
4. Update for each good $e \in S_i : p_e^{i+1} = p_e^i \cdot 2^{1/b}$.

End: Bidder i pays: $p^i = \sum_{e \in S_i} p_e^i$.

Algorithm 4: Overselling MPU Algorithm

Βλέπουμε, λοιπόν, ότι αλγόριθμος αυτός συνδυάζει 3 τεχνικές που έχουμε δει και αναλύσει προηγουμένως:

i Αρχικά, καταφέρνει να παράξει έναν truthful μηχανισμό για Combinatorial Auctions.

ii Κατά δεύτερον, χρησιμοποιεί τις τεχνικές των Multiplicative Price Updates προκειμένου να αυξάνει την τιμή των εναπομείναντων αντιγράφων, κάθε φορά που πωλείται κάποιο συγκεκριμένο αντικείμενο. Διαισθητικά, αυτή η αύξηση στην τιμή προκύπτει από τον Νόμο της Προσφοράς και της Ζήτησης: όσο αυξάνεται η ζήτηση, τόσο πρέπει να αυξηθεί και η τιμή (διότι μειώνεται η προσφορά) προκειμένουν να φτάσουμε στην επιθυμητή ισορροπία Προσφοράς-Ζήτησης.

iii Τέλος, χρησιμοποιείται εγγενώς posted-price μηχανισμός και μάλιστα, έχουμε ξεπεράσει το σκόπελο του τεράστιου μεγέθους πληροφορίας (λόγω combinatorial auction).

Υπενθυμίζουμε, ωστόσο, εδώ ότι η λύση που παράγει ο αλγόριθμός μας είναι μη-εφικτή και αυτό γιατί μπορεί να πουλήσει παραπάνω αντικείμενα από ό, τι διαθέτει αρχικά. Παρακάτω θα δείξουμε πώς να παραμετροποιούμε τα Demand Oracles, δηλαδή πώς να επιλέγουμε κατάλληλα τα υποσύνολα $U_1, U_2, \ldots, U_n$, ώστε να παίρνουμε μία εφικτή λύση.

MPU Algorithm with Oblivious Randomized Rounding

Η ιδέα για τον τροποποιημένο αλγόριθμο είναι να χρησιμοποιούμε «εικονικά» αντίγραφα αντικειμένων και να τα πουλάμε ακολουθώντας τον multiplicative price update rule. Ο αριθμός των «εικονικών» αντιτύπων που πωλούνται ξεπερνά τα διαθέσιμα (πραγματικά) αντίτυπα κατά έναν παράγοντα $O(\log(\mu bm))$, το οποίο αντιστοιχεί στο άνω φράγμα της υπερ-πώλησης του αλγορίθμου 8. Στον αλγόριθμο που ακολουθεί θα χρησιμοποιήσουμε μία τυχαιοποιημένη διαδικασία στρογγυλοποίησης (randomized rounding) προκειμένου να αποφασίσουμε ποια από τα «εικονικά» αντίγραφα που αρχικά ανατίθενται στους παίκτες, θα γίνουν πραγματικά bundles που θα πωληθούν. Συγκεκριμένα, για κάθε bidder i, ο αλγόριθμος θέτει $R_i = S_i$ με πιθανότητα q και $R_i = \emptyset$ με πιθανότητα $1-q$, όπου $0 \leq q \leq 1$ είναι μία παράμετρος. Για διαφορετικά προβλήματα από combinatorial auctions μπορούμε να προσδιορίσουμε διαφορετικό q, αλλά αυτό δεν είναι κάτι που θα εξετάσουμε σε αυτή την εργασία.

Initialization: Set all items' prices at $p_0 = \frac{L}{4bm}$.

1. For each good $e \in U$ do $p_e^1 = p_0$, $b_e^1 = b$.
2. For each bidder $i = 1, 2 \ldots, n$ do:
3. Set $S_i = D_i(U_i, p^i)$, for $U_i = \{e \in U | b_e^i > 0\}$
4. Update for each good $e \in S_i : p_e^{i+1} = p_e^i \cdot 2^{1/b}$.
5. With probability q set $R_i = S_i$, else $R_i = \emptyset$.
6. Update for each good $e \in R_i : b_e^{i+1} = b_e^i - 1$.

End: Bidder i pays: $p^i = \sum_{e \in R_i} p_e^i$.

Algorithm 5: MPU Algorithm with Oblivious Randomized Rounding

Παρατηρούμε ότι ο ορισμός του U_i στη γραμμή 3 του αλγορίθμου εξασφαλίζει ότι για κάθε αντικείμενο e, ο αριθμός των αντιγράφων του e στο $R_1 \cup \cdots \cup R_n$ δεν υπερβαίνει τον αριθμό b των διαθέσιμων αντιγράφων. Συνεπώς, η ανάθεση του R είναι εφικτή. Επιπλέον, εάν η πιθανότητα q στο βήμα 5 οριστεί στο 0, τότε η «προσωρινή» ανάθεση των S που υπολογίζεται από το αλγόριθμο 5 είναι ταυτόσημη με την ανάθεση S από τον αλγόριθμο 8 με τα μη-περιορισμένα Demand Oracles. Προφανώς, ωστόσο, πρέπει να θέσουμε $q > 0$, έτσι ώστε η τελική ανάθεση R να μην είναι κενή.

2.4.3 Το πλαίσιο των Plotkin, Shmoys, Tardos

Μία από τις σημαντικότερες εφαρμογές της μεθόδου των multiplicative updates βρίσκεται στον πλαίσιο των Plotkin, Shmoys, Tardos ([23]) σχετικά με την προσεγγιστική επίλυση των fractional packing and covering προβλημάτων. Ο αλγόριθμός τους είναι μία ποσοτική έκδοση της κλασικής ιδέας χαλάρωσης του Lagrange και εφαρμόζεται επίσης στον γενικό γραμμικό προγραμματισμό. Εμείς θα ασχοληθούμε με τον αλγόριθμο για τα γενικά LP.

Το βασικό μας πρόβλημα εδώ είναι να ελέγξουμε την εφικτότητα του ακόλουθου προγράμματος LP:

$$Ax \geq b \,, x \in P \tag{2.22}$$

όπου A είναι ένας $m \times n$ πίνακας, $x \in \mathbb{R}^n$ και το P είναι ένα κυρτό σύνολο στο $\mathbb{R}^n$. Διαισθητικά το σύνολο P εκφράζει τους περιορισμούς που είναι «εύκολοι» στο να ικανοποιηθούν, όπως για παράδειγμα η μη-αρνητικότητα και ο A αναπαριστά τους «δύσκολους» περιορισμούς. Οι Plotkin, Shmoys και Tardos υποθέτουν την ύπαρξη ενός Μαντείου (Oracle), το οποίο λύνει το ακόλουθο πρόβλημα εφικτότητας:

$$\exists ?x \in P : c^T x \geq d \tag{2.23}$$

όπου $c = \sum_i p_i A_i$ και $d = p_i b_i$ για κάποια κατανομή $p_1, p_2, \ldots, p_m$. Είναι λογικό να περιμένουμε να υπάρχει μία τέτοια διαδικασία βελτιστοποίησης, εφόσον εδώ το μόνο που χρειαζόμαστε είναι να ελέγξουμε την εφικτότητα ενός περιορισμού και όχι των m.

Χρησιμοποιώντας αυτό το μαντείο, θα περιγράψουμε έναν αλγόριθμο ο οποίος είτε οδηγεί σε μία προσεγγιστικά εφικτή λύση, δηλαδή σε μία λύση $x \in P$ τέτοια ώστε $A_i x \geq b_i - \delta$ για κάποιο μικρό $\delta > 0$ είτε (εάν αποτύχει να υπολογίσει μία τέτοια λύση) αποδεικνύει ότι το σύστημα είναι μη-εφικτό.

Για να αντιστοιχίσουμε το γενικό μας πλαίσιο σε αυτή την κατάσταση, έχουμε ότι ο κάθε expert αντιστοιχίζεται σε κάποιον από τους m περιορισμούς. Τα γεγονότα αντιστοιχίζονται σε διανύσματα λύσεων $x \in P$. Η ποινή του expert που αντιστοιχεί στον περιορισμό i για το γεγονός x είναι $A_i x - b_i$ [5] Έστω παράμετρος ρ [6]γνωστή στον αλγόριθμό μας και υποθέτουμε ότι οι απαντήσεις του oracle ικανοποιούν τη σχέση $A_i x - b_i \in [-\rho, \rho]$ για κάθε i. Επομένως, οι ποινές βρίσονται στο διάστημα $[-\rho, \rho]$. Τρέχουμε, λοιπόν, τον Αλγόριθμο των Multiplicative Updates για T γύρους (όπως στο Πόρισμα 2.2), χρησιμοποιώντας $\epsilon = \delta/4\rho$. Αξίζει εδώ επίσης να σημειώσουμε ότι εάν $p_1, p_2, \ldots, p_m$ είναι οι κατανομή κάθε χρονική στιγμή, τότε καλούμε το oracle με $c = \sum_i p_i A_i$ και $d = \sum_i b_i$.

Έχουμε τις ακόλουθες περιπτώσεις:

Περίπτωση 1. Το oracle επιστρέφει ένα εφικτό x για το 2.23 σε κάθε επανάληψη. Τότε, από το Πόρισμα 2.2, έχουμε για κάθε i:

$$\frac{\sum_{t=1}^{T} \sum_j p_j^t [A_j x^t - b_j]}{T} \leq \delta + \frac{\sum_{t=1}^{T} [A_i x^t - b_i]}{T} \tag{2.24}$$

Το αριστερό μέλος είναι μεγαλύτερο ή ίσο του μηδενός, από την υπόθεση. Έστω τότε ότι $\bar{x} = \sum_t x^t / T$ η τελική απάντηση, εφόσον η προηγούμενη γραμμή υπονοεί ότι για κάθε γραμμή i, $A_x \bar{x} \leq b_i - \delta$. Έτσι, έχουμε μία προσεγγιστικά εφικτή λύση $\bar{x}$.

Περίπτωση 2. Σε κάποια επανάληψη, το oracle δηλώνει ανεφικτότητα να επιλύσει την 2.23. Σε αυτή την περίπτωση, καταλήγουμε ότι το αρχικό σύστημα είναι ανέφικτο. Και αυτό είναι

[5] Διαισθητικά, η μαθηματική αυτή έκφραση υποδηλώνει πόσο απέχει η λύση του γεγονότος από τον εκάστοτε περιορισμό.

[6] Η παράμετρος αυτή μπορεί να βρεθεί με δυαδική αναζήτηση.

σωστό γιατί αν υπήρχε κάποια εφικτή λύση x, τότε θα ίσχυε $Ax \geq b$ και παίρνουντας το γραμμικό συνδυασμό των ανισοτήτων που δίνεται από την κατανομή $p_1, p_2, \ldots, p_m$ στην τρέχουσα επανάληψη έχουμε ότι: $\sum_i p_i A_i x \geq \sum_i p_i b_i$, το οποίο σημαίνει ότι το oracle λανθασμένα δήλωσε ανεφικτότητα επίλυσης, άτοπο.

Τα παραπάνω τα διατυπώνει καλύτερα το παρακάτω Πόρισμα.

Πόρισμα 2.3. *Μετά από $T = O(\frac{\rho^2}{\epsilon^2} \ln m)$ γύρους, ο αλγόριθμος βρίσκει μία λύση, η οποία ικανοποιεί τους περιορισμούς με ένα αθροιστικό παράγοντας λάθους, δηλαδή $A_i \bar{x} \geq b_i - \epsilon$, εάν υπάρχει λύση ή αλλιώς καταλήγει στο συμπέρασμα ότι το σύστημα είναι αδύνατο.*

Διαισθητικά, αυτό που μας λέει αυτό το πόρισμα είναι ότι πρέπει να υπάρξει κάποιο tradeoff μεταξύ των γύρων που θα τρέξει ο αλγόριθμος και του λάθους που θα έχει κάνει. Πρακτικά, αν αφήσουμε τον αλγόριθμο να τρέξει για πολλούς γύρους, θα βρεθούμε πολύ κοντά στη βέλτιστη λύση.

2.4.4 Ενδυνάμωση - Boosting

Η ενδυνάμωση (boosting) [24] είναι κεντρική ιδέα στην Τεχνητή Νοημοσύνη (Artificial Intelligence) (AI) σήμερα. Ο αλγόριθμος *AdaBoost* των Freund και Schapire ([25]) χρησιμοποιεί τον Κανόνα των Πολλαπλασιαστικών Συντελεστών (Multiplicative Weights Update Rule) και ταιριάζει στο πλαίσιο που έχουμε αναπτύξει ως τώρα. Παρακάτω θα εξηγήσουμε την κύρια ιδέα, χρησιμοποιώντας κάπως απλοποιημένες υποθέσεις.

Έστω X είναι κάποιο σύνολο (domain) και ας υποθέσουμε ότι προσπαθούμε να μάθουμε κάποια άγνωστη συνάρτηση (concept) $c : X \rightarrow \{0, 1\}$ που επιλέγεται από κάποια concept κλάση $\mathcal{C}$. Με δεδομένη μία ακολουθία από παραδείγματα εκπαίδευσης $(x, c(x))$, όπου το x παράγεται από μία καθορισμένη, αλλά άγνωστη κατανομή $\mathcal{D}$ πάνω στο domain X, ο αλγόριθμος εκμάθησης (learning) πρέπει να βγάλει ως έξοδο μία *υπόθεση* $h : X \rightarrow \{0, 1\}$. Το *σφάλμα* της υπόθεσης ορίζεται ως $\mathbb{E}_{x \sim \mathcal{D}}[|h(x) - c(x)|]$.

Ένας *δυνατός αλγόριθμος εκμάθησης* είναι ένας αλγόριθμος, ο οποίος για κάθε κατανομή $\mathcal{D}$ και για κάθε $\epsilon, \delta > 0$ δίνει στην έξοδο με πιθανότητα $1 - \delta$ μία υπόθεση, της οποίας το σφάλμα είναι το πολύ ϵ. Ένας *γ-ασθενής αλγόριθμος εκμάθησης* για $\gamma > 0$ είναι παρόμοιος με τον δυνατό αλγόριθμο εκμάθησης, απλά το σφάλμα του μπορεί να είναι το πολύ $1/2 - \gamma$. Η ενδυνάμωση μας δείχνει ότι εάν υπάρχει ένας γ-ασθενής αλγόριθμος εκμάθησης για μία concept κλάση, τότε υπάρχει και ένας δυνατός αλγόριθμος εκμάθησης. [7]

Αποδεικνύουμε αυτό το αποτέλεσμα σε αυτό που αποκαλούμε *ενδυνάμωση από το πλαίσιο δειγματοληψίας (boosting by sampling framework)*, το οποίο χρησιμοποιεί ένα καθορισμένο σύνολο εκπαίδευσης από N παραδείγματα. Η ιδέα είναι να τρέξουμε επαναληπτικά τον ασθενή αλγόριθμο εκμάθησης πάνω σε διαφορετικές κατανομές που καθορίζεται από αυτό το σύνολο. Η τελική υπόθεση έχει σφάλμα ϵ' κάτω από την ομοιόμορφη κατανομή πάνω στο σύνολο εκμάθησης και μπορούμε με κάποια βοήθεια από τη βιβλιογραφία να καταλήξουμε ότι με πιθανότητα $1 - \delta$ το σφάλμα της υπόθεσης πάνω σε ολόκληρο το domain X είναι ϵ εάν το μέγεθος του συνόλου εκμάθησης, N, έχει επιλεγεί κατάλληλα.

Χρησιμοποιούμε τον αλγόριθμο Multiplicative Weights Update, αλλά για να αποφύγουμε οποιοδήποτε μπέρδεμα με τον συμβολισμό, θα χρησιμοποιήσουμε το α, αντί για το ϵ για τους

[7] Ο χρόνος τρεξίματος του αλγορίθμου καθώς και ο αριθμός των δειγμάτων μπορεί να εξαρτώνται από το γ.

multiplicative update παράγοντες. Οι experts αντιστοιχούν σε δείγματα του συνόλου εκπαίδευσης και τα γεγονότα αντιστοιχούν στο σύνολο όλων των υποθέσεων που μπορούν να παραχθούν από τον ασθενή αλγόριθμο εκμάθησης. Εάν το γεγονός που αντιστοιχεί στην υπόθεση h πραγματοποιηθεί, η ποινή για τον expert x είναι 1 ή 0, ανάλογα με το εάν $h(x) = c(x)$ ή όχι.

Σε κάθε επανάληψη, ο αλγόριθμος παρουσιάζει την τρέχουσα κατανομή $\mathcal{D}^t$ πάνω στα παραδείγματα στον ασθενή αλγόριθμο εκμάθησης και σε αντάλλαγμα παίρνει μία υπόθεση h^t της οποίας το σφάλμα, με δεδομένη την κατανομή $\mathcal{D}^t$, είναι το πολύ $1/2 - \gamma$. Με άλλα λόγια, η αναμενόμενη ποινή, $\mathbf{M}(\mathcal{D}^t, h^t)$ σε κάθε επανάληψη είναι τουλάχιστον $1/2 + \gamma$. Ο αλγόριθμος εκτελείται για T γύρους. Η τελική υπόθεση, h_{final}, καθορίζει το $x \in X$, ανάλογα με την ψήφο της πλειοψηφίας μεταξύ των $h^1(x), h^2(x), \ldots, h^T(x)$.

Έστω S το σύνολο των $x \in X$ που έχουν λανθασμένα υποδειχθεί από το h_{final}. Η ποινή για κάθε $x \in X$, $\sum_t \mathbf{M}(x, h^t)$ είναι το πολύ $T/2$, εφόσουν η ψήφος της πλειοψηφίας για αυτό είναι λανθασμένη. Προσαρμόζουμε τώρα την απόδειξη του Θεωρήματος 2.2: Όπως και στην απόδειξη, έχουμε:

$$\Phi^T \leq \Phi^1 e^{-\alpha \sum_t \mathbf{M}(\mathcal{D}^t, h^t)} \leq n e^{-\alpha T(1/2+\gamma)}$$

Τώρα έχουμε:

$$\Phi^T \geq \sum_{x \in S} (1-\alpha)^{\sum_t \mathbf{M}(x,h^t)} \geq |S|(1-\alpha)^{T/2} \geq |S| e^{-(\alpha+\alpha^2)(T/2)}$$

[8] Επομένως, καταλήγουμε ότι το σφάλμα του h_{final} πάνω στο σύνολο εκπαίδευσης κάτω από την ομοιόμορφη κατανομή είναι $\frac{|S|}{n} \leq e^{-\alpha(\gamma - \alpha/2)T}$. Διαλέγοντας $\alpha = \gamma$ και $T = \frac{2}{\gamma^2} \ln \frac{1}{\epsilon'}$ παίρνουμε ότι το σφάλμα είναι το πολύ ϵ', όπως επιθυμούσαμε.

2.4.5 Άλλες εφαρμογές

Πέρα από τις εφαρμογές που αναφέραμε παραπάνω, η μέθοδος των Multiplicative Updates βρίσκει εφαρμογή και σε πολλές άλλες περιπτώσεις όπως είναι: η προσέγγιση των Multicommodity Flow προβλημάτων ([26]), η $O(\log n)$-προσέγγιση αρκετών NP-δύσκολων προβλημάτων ([27, 28, 29]), ο έλεγχος των Congestion Networks, η προσεγγιστική επίλυση μερικών Semidefinite προβλημάτων ([30, 31, 14]), η προσέγγιση Graph Separators και το Online Convex Optimization ([32, 33]), ενώ χαρακτηριστική και εξαιρετικά ενδιαφέρουσα είναι η συμβολή της μεθόδου αυτής στην Θεωρία της Εξέλιξης.

[8] Εδώ έχουμε χρησιμοποιήσει την ιδιότητα: $(1-x) \geq e^{-(x+x^2)}$, για $x < 1/2$.

Κεφάλαιο 3

Περιορισμοί στη Ρευστότητα

Ως τώρα έχουμε κάνει μία κρυφή παραδοχή: παντού συζητάμε για τα valuations των παικτών, τις αξίες δηλαδή που έχουν για αυτούς τα αγαθά. Εάν η τιμή ενός αγαθού είναι το πολύ ίση με το valuation του παίκτη, τότε ο παίκτης μπορεί να αποκτήσει το αντικείμενο, σε ό, τι αφορά τους posted-price μηχανισμούς, ενώ η χρέωση του παίκτη στους VCG μηχανισμούς ήταν ίση με το externality που επέβαλε στα valuations των άλλων παικτών το bid του. Αυτό που δεν έχουμε συμπεριλάβει πουθενά στις αναλύσεις μας είναι το εάν ο παίκτης *μπορεί πράγματι να πληρώσει την τιμή του αγαθού*, με άλλα λόγια αν έχει την κατάλληλη *ρευστότητα (budget)*. Η εισαγωγή των περιορισμών των budget των παικτών στα προβλήματά μας δημιουργεί ένα ολόκληρο νέο πεδίο ανάλυσης. Γίνεται, δηλαδή, εμφανής μία διαφορά μεταξύ της *θέλησης να πληρώσουμε για κάποιο αγαθό (willingness to pay)* και της *ικανότητάς μας να πληρώσουμε για αυτό (ability to pay)*.

Ανάλογα με το πρόβλημα, το budget των παικτών μπορεί να είναι *κοινό*, να προέρχεται από κάποια *κατανομή*, να είναι *public* ή *private*, ενώ μπορεί ακόμη το budget constraint να το έχει ένας παίκτης που προσπαθεί να αγοράσει υπηρεσίες από τους υπόλοιπους (procurement auctions). Προκύπτει, λοιπόν, η ανάγκη να μελετήσουμε αυτά τα νέα προβλήματα κάτω από την ήδη γνωστή μας μετρική, το Social Welfare, αλλά και κάτω από μία νέα μετρική, την οποία θα εισάγουμε και θα μελετήσουμε σε αυτό το κεφάλαιο, το Liquid Welfare.

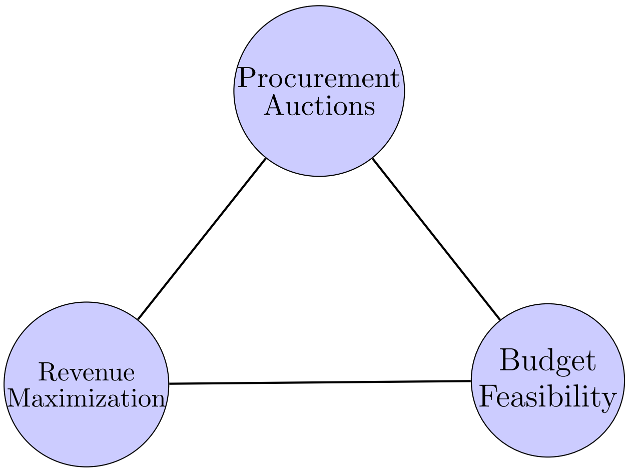
Ας μιλήσουμε, όμως, πρώτα για τις συνέπειες που έχει η ύπαρξη των budgets στις *δημοπρασίες προμήθειας (procurement auctions)*.

3.1 Δημοπρασίες Προμήθειας - Procurement Auctions

Ορισμός 3.1 (Δημοπρασίες Προμήθειας)**.** Ονομάζουμε *δημοπρασίες προμήθειας (procurement auctions)* δημοπρασίες στις οποίες ο ρόλος του αγοραστή και του πωλητή αντιστρέφονται [α΄]. Σε αυτές τις δημοπρασίες, οι πωλητές ανταγωνίζονται για να αγοράσουν αγαθά ή υπηρεσίες από τον αγοραστή και οι τιμές συνήθως μειώνονται, καθώς οι πωλητές αποκλείουν ο ένας τον άλλον.

[α΄] Εξ ου και στη βιβλιογραφία πολύ συχνά τις συναντάμε ως «αντίστροφες» δημοπρασίες.

Έχοντας μιλήσει στο Κεφάλαιο της Εισαγωγής για μηχανισμούς που οδηγούν σε Revenue Maximization ή αλλιώς, για μηχανισμούς που βελτιστοποιούν το Social Welfare ερχόμαστε τώρα αντιμέτωποι με το παρακάτω τρίγωνο, στο οποίο μπορούν μόνο 2 από τους 3 κόμβους να ισχύουν για τους μηχανισμούς μας.



Υπενθυμίζουμε τα εξής:

- **Procurement Auctions**: 1 αγοραστής και n πωλητές που έρχονται με σειριακό τρόπο. Ο αγοραστής προμηθεύει την υπηρεσία/αγαθό και οι τιμές διαμορφώνονται με βάση τα budgets των πωλητών που έρχονται να αγοράσουν την υπηρεσία.
- **Revenue Maximization**: n αγοραστές και 1 σχεδιαστής της δημοπρασίας. Στόχος μας είναι ο καθορισμός των πληρωμών, ώστε να μεγιστοποιηθούν τα χρήματα που θα συγκεντρώσει ο σχεδιαστής.
- **Budget Feasible**: Γενικό πλαίσιο σχεδιασμού μηχανισμών, όπου ο εκάστοτε αγοραστής μπορεί να αγοράσει ό, τι θέλει, αρκεί να μην ξεπερνά ποτέ το budget του.

Όταν αναφερόμαστε σε Αγορά Προμηθειών (Procurement Market) πρέπει να κρατάμε κατά νου ότι έχουμε δύο παραμέτρους στο πρόβλημα: τον τρόπο που έρχονται οι agents (secretary, adversarial [1]) και την αντικειμενική συνάρτηση του πωλητή. Ο στόχος μας είναι να φτιάξουμε

[1] Όπως και σε κάθε online μηχανισμό, όπως έχουμε ξαναδεί.

truthful μηχανισμούς, οι οποίοι να είναι budget feasible και να μεγιστοποιούν το utility του πωλητή. Βλέπουμε, λοιπόν, ότι εστιάζουμε στην ικανοποίηση των 2 εκ των 3 κόμβων, ενώ ο κόμβος του Revenue Maximization αντιτίθεται σε αυτούς τους στόχους μας. Διαισθητικά, αν προσπαθήσουμε να αποκτήσουμε όσο το δυνατόν μεγαλύτερη ωφέλεια, με σεβασμό πάντα στο budget μας, θα καταλήξουμε να επιθυμούμε «χαμηλότερες» τιμές. Επομένως, τα έσοδα από μία τέτοια δημοπρασία δεν είναι (στη γενική περίπτωση) τα μέγιστα δυνατά.

Αν, ωστόσο, εστιάσουμε στο Revenue Maximization ενός Procurement Market, αυτό σημαίνει ότι θα πρέπει να κοστολογούμε κάθε αγαθό/υπηρεσία πολύ ψηλά. Επομένως, (εκτός και αν τα budgets είναι πολύ μεγάλα, μία υπόθεση που δεν είναι πολύ συνηθισμένη, ούτε και λογική) θα πρέπει είτε να μην αγοράσουν τίποτα οι παίκτες ή να αγοράσουν πέρα από το budget τους, καταστρατηγώντας με αυτό τον τρόπο το Budget Feasibility.

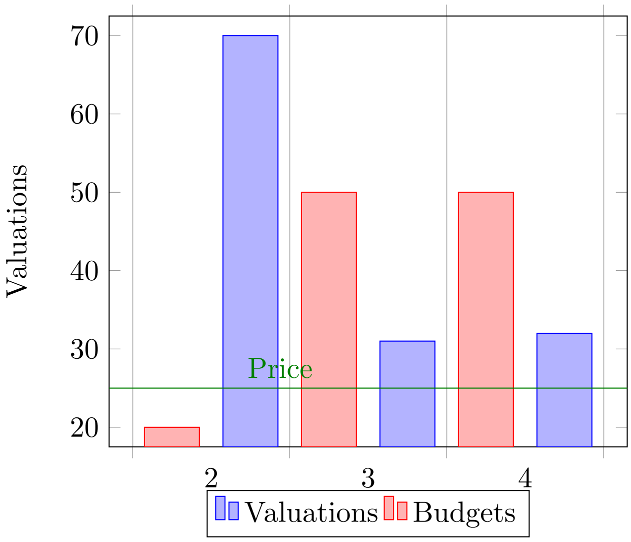
Τέλος, μπορούμε να έχουμε δημοπρασίες όπου θα επιτυγχάνουμε Revenue Maximization ή τουλάχιστον, μία καλή προσέγγιση αυτού, ταυτόχρονα με το Budget Feasibility, αλλά αυτές οι δημοπρασίες δεν εμπίπτουν στην κατηγορία των Procurement Auctions. Για να δούμε την επίδραση που έχει το budget στα έσοδα των κλασικών δημοπρασιών παραπέμπουμε στα: [34, 35] και για τον σχεδιασμό μηχανισμών που βελτιστοποιούν (ή τουλάχιστον προσεγγιστικά βελτιστοποιούν) τα έσοδα παραπέμπουμε στα: [36, 37, 38, 39, 40].

3.2 Μετρικές Απόδοσης

3.2.1 Pareto Efficiency vs Budgets

Ωστόσο, τα πράγματα είναι πολύ διαφορετικά όταν αντί για το Revenue, μας ενδιαφέρει το social welfare. Ένα πολύ γνωστό αποτέλεσμα μας δείχνει ότι ακόμη και αν ο σχεδιαστής της δημοπρασίας γνωρίζει τα budgets, η καλύτερη δυνατή truthful δημοπρασία πετυχαίνει n-προσέγγιση στο social welfare, όπου n ο αριθμός των παικτών.

Προτού διατυπώσουμε το Λήμμα επακριβώς και το αποδείξουμε, ας δούμε ένα αντιπαράδειγμα το πόσο κακή συμπεριφορά μπορούν να έχουν τα budgets για το social welfare. Έστω ότι στο παρακάτω παράδειγμα, θέλουμε να δημοπρατήσουμε 1 αντικείμενο (σε όσα αντίτυπα θέλουμε [2]) με τιμή $p = 25$ και έχουμε τώρα 3 bidders με valuations και budgets όπως φαίνονται στο παρακάτω διάγραμμα. Η διαδικασία της δημοπρασίας είναι απλή: όποιος έχει $v > p$ και $b > p$



Προφανώς, προκειμένου να εξασφαλίσουμε το individual rationality, οι παίκτες δεν μπορούν να πληρώσουν παραπάνω από το budget τους. Με αυτό κατά νου και επίσης, με το (παραδοσιακό) δεδομένο ότι οι παίκτες αγοράζουν μόνο αν η τιμή είναι πάνω από το valuation τους, έχουμε καταρχάς ότι το μοντέλο αυτής της φανταστικής δημοπρασίας είναι truthful και μπορούμε να το δούμε σχετικά εύκολα. Η δημοπρασία μας θα πουλήσει το αντικείμενο (καλύτερα, ένα αντίτυπο του αντικειμένου) στους παίκτες 3 και 4 και θα συγκεντρώσει social welfare: $SW[p = 25] = 63$. Το βέλτιστο social welfare που θα μπορούσαμε να μαζέψουμε (αν δεν είχαμε budgets), ωστόσο, θα έπρεπε να περιλαμβάνει και το valuation του bidder 2.

Λήμμα 3.1. *Έστω δημοπρασίες divisible-multi-unit και αθροιστικοί (additive) bidders. Δεν υπάρχει κανένας α-προσεγγιστικός, truthful και individually rational μηχανισμός $x(v), p(v)$ με $\alpha < n$. Για $\alpha = n$ υπάρχει ένας μηχανισμός που αναθέτει το αντικείμενο στην τύχη και δεν χρεώνει τίποτα.*

Απόδειξη. Έστω ότι όλοι οι agents έχουν uniform budget, $B_i = 1$. Παρατηρούμε, αρχικά, ότι:

$$\lim_{v_i \to \infty} x_i(v_i, v_{-i}) \geq \alpha^{-1} \tag{3.1}$$

[2] Επειδή εδώ έχουμε 3 bidders, έστω ότι έχουμε 3 αντίτυπα.

και αυτό γιατί:

$$\sum_j v_j \cdot x_j(v) \geq \alpha^{-1} \max_i v_i \geq \alpha^{-1} v_i \tag{3.2}$$

Επομένως:

$$x_i(v) + \sum_{j \neq i} \frac{v_j}{v_i} \cdot x_j(v) \geq \alpha^{-1} \tag{3.3}$$

Παίρνοντας το όριο $v_i \to \infty$, έχουμε τη σχέση 3.1.

Θέλουμε τώρα να αποδείξουμε το truthfulness, δηλαδή θα δείξουμε ότι: $v_i x_i(v) - p_i(v) \geq v_i x_i(v'_i, v_{-i}) - p_i(v'_i, v_{-i})$. Με δεδομένο ότι για τις τιμές (και άρα και για τις πληρωμές) ισχύει η ανισότητα: $0 \leq p_i \leq 1$ έχουμε ότι:

$$v_i x_i(v) \leq v_i x_i(v'_i, v_{-i}) - 1 \tag{3.4}$$

και διαιρώντας κατά μέλη με το v_i:

$$x_i(v) \geq x_i(v'_i, v_{-i}) - \frac{1}{v_i} \xRightarrow[3.1]{v'_i \to \infty} x_i(v) \geq \alpha^{-1} - \frac{1}{v_i} \tag{3.5}$$

Αθροίζοντας πάνω σε όλους τους παίκτες παίρνουμε:

$$1 \geq \sum_i x_i(v) \geq n\alpha^{-1} - \sum_i \frac{1}{v_i} \xRightarrow{v_i \to \infty, \forall i} n\alpha^{-1} \geq 1 \tag{3.6}$$

κάτι το οποίο προφανώς ισχύει. ■

Αυτό το impossibility αποτέλεσμα αποτέλεσε το κίνητρο για την έρευνα πάνω σε ασθενέστερες μετρικές για τη μελέτη δημοπρασιών με budgets. Αρχικά, οι Dobzinski, Lavi και Nisan ([41]) προτείνουν τη μελέτη των Pareto-efficient δημοπρασιών.

Ορισμός 3.2 (Pareto Efficient)**.** Το αποτέλεσμα μίας δημοπρασίας ονομάζεται *Pareto αποδοτικό (Pareto Efficient)* εάν δεν υπάρχει κανένα εναλλακτικό αποτέλεσμα (σχήμα ανάθεσης και σχήμα πληρωμών) όπου κανένας agent (είτε bidder είτε auctioneer) να μην έχει χειρότερη απόδοση και τουλάχιστον ένας agent να έχει καλύτερη απόδοση.

Σε αυτή τη δουλειά, οι συγγραφείς σχεδιάζουν μία truthful και Pareto-efficient δημοπρασία, βασισμένη στο πλαίσιο του Ausubel για την clinching δημοπρασία, με την προϋπόθεση ότι τα budgets είναι δημόσια. Επιπλέον, αποδεικνύουν ότι αυτή είναι η μοναδική truthful δημοπρασία που παράγει πάντα Pareto-Efficient λύση. Η μελέτη των Pareto-βέλτιστων δημοπρασιών για παίκτες με περιορισμένο budget έχει επεκταθεί σε διαφορετικά settings σε μία ακολουθία από μελέτες που ακολούθησαν [42, 43, 44, 45, 46].

Ωστόσο, ακριβώς αυτό το αποτέλεσμα «μοναδικότητας» της Clinching Auction είναι το πρόβλημα. Πολύ σπάνια στην πράξη έχουμε ως μοναδικό στόχο του σχεδιαστή το Pareto Optimality. Επιπλέον, ένα βασικό πρόβλημα με το Pareto-optimality είναι ότι είναι μία δυαδική μετρική: μία ανάθεση είτε είναι Pareto-efficient είτε δεν είναι, δεν υπάρχει περισσότερο ή λιγότερο Pareto-efficient. Αυτό διαισθητικά έρχεται σε αντίθεση με τα όσα ξέραμε για το Social Welfare.

Ήταν, λοιπόν, λογικό να αναζητηθεί ένα νέο μέτρο αποδοτικότητας, το οποίο να εφαρμόζεται στα προβλήματα με budgets. Τα βασικά χαρακτηριστικά που θα θέλαμε να έχει αυτό το μέτρο ήταν:

i να είναι μετρήσιμο, δηλαδή να αντιστοιχίζει μία τιμή για κάθε πιθανό αποτέλεσμα

ii να είναι επιτεύξιμο, δηλαδή να μπορεί να προσεγγιστεί από truthful μηχανισμούς και τέλος

iii να επιτρέπει νέους σχεδιασμούς μηχανισμών, που προσεγγίζουν το welfare

3.2.2 Ρευστή Ευημερία - Liquid Welfare

Το Liquid Welfare είναι η μετρική που ικανοποιεί όλες τις παραπάνω προϋποθέσεις και προτάθηκε από τους Dobzinski και Leme στο [47].

Ορισμός 3.3 (Liquid Welfare). Δεδομένου ενός outcome x_i, ορίζουμε τη *ρευστή ευημερία (liquid welfare)* ως εξής:

$$\bar{W}(x) = \sum_i \min(v_i(x_i), B_i)$$

όπου $v_i(x_i)$ είναι το valuation του παίκτη i για το outcome v_i και B_i το budget του παίκτη. Συμβολίζουμε με $\bar{W}^*$ το βέλτιστο Liquid Welfare.

Διαισθητικά, αυτό που καταφέρνει το liquid welfare είναι να μετρήσει τη διάθεση ενός παίκτη να πληρώσει για ένα αγαθό: το ελάχιστο, δηλαδή, μεταξύ της ικανότητάς του να πληρώσει (budget) και της θέλησής του να πληρώσει (valuation). Από την οπτική γωνία του σχεδιαστή της δημοπρασίας, το Liquid Welfare αποτελεί το μέγιστο κέρδος που μπορεί να αποκτήσει από τη δημοπρασία, δεδομένου ότι οι παίκτες μπορούν να πληρώσουν το πολύ όσο είναι το budget τους.

Η μετρική αυτή, επίσης, ικανοποιεί τις 3 προϋποθέσεις που θέσαμε παραπάνω. Καταρχάς, είναι προφανώς μετρήσιμη, αφού σε εξαρτάται από το outcome βάσει ορισμού. Κατά δεύτερον, μπορεί να προσεγγιστεί, κάτι που θα το αποδείξουμε εκτενέστερα παρακάτω με συγκεκριμένα παραδείγματα. Και τρίτον, επιτρέπει σχεδιασμούς νέων μηχανισμών το οποίο έπεται και αυτό για τη συνέχεια.

3.3 Clinching Auction

Η δημοπρασία αυτή αναπτύχθηκε αρχικά στο [48] και προσαρμόστηκε στο budgeted πρόβλημα στο [41]. Θα αναφερθούμε σε αυτή την ενότητα στην *Προσαρμοστική Clinching Auction (Adaptive Clinching Auction)*.

Διαισθητικά η Adaptive Clinching Auction είναι μία διαδικασία αύξησης τιμής: υπάρχει ένα ρολόι τιμής (price clock), του οποίου η τιμή p ξεκινάει από το 0 και σταδιακά ανεβαίνει. Ας πούμε ότι το ρολόι αυξάνεται με ρυθμό ϵ. Αρχικά, το αγαθό δεν έχει ανατεθεί σε κανέναν και ο κάθε παίκτης έχει στο ακέραιο το αρχικό του budget.

Το πλαίσιο περιγραφής μίας Clinching Auction περιλαμβάνει:

1. Ορίζουμε ως *προσφορά (supply)* το εναπομείναν αγαθό, από την αρχή της δημοπρασίας, στην τιμή p, $S(p)$. Προφανώς, αρχικά $S(0) = 1$. Σε κάθε τιμή p, θα ισχύει για την προσφορά:
$$S(p) = 1 - \sum_i x_i(p) \tag{3.7}$$
2. Το ποσό που κάνει clinch ο παίκτης i: $x_i(p)$. Αρχικά, $x_i(0) = 0$.
3. Το ποσό που χρεώνουμε σε κάθε παίκτη i: $p_i(p)$. Αρχικά, $p_i(p) = 0$. Το ποσό που χρεώνουμε στον παίκτη είναι η τιμή/μονάδα αγαθού που λαμβάνει. Επομένως,
$$\partial_p p_i(p) = p \cdot \partial_p x_i(p) \tag{3.8}$$
4. Το εναπομείναν budget του κάθε παίκτη i: $b_i(p)$. Αρχικά, $b_i(p) = B_i$. Το budget του κάθε παίκτη θα μειώνεται ανάλογα με το πόσο πληρώνει για το αγαθό:
$$b_i(p) = B_i - p_i(p) \tag{3.9}$$

Από τα παραπάνω συμπεραίνουμε ότι η ζήτηση που θα «επιβάλει» ο κάθε παίκτης i είναι:
$$D_i(p) = \begin{cases} \frac{b_i}{p} & \text{if } p < v_i \\ 0 & \text{if } otherwise \end{cases} \tag{3.10}$$

Ουσιαστικά, χωρίζουμε το budget σε κομμάτια-τιμή και όσα μπορούμε να πάρουμε, τα αγοράζουμε. Τόση ακριβώς είναι και η ζήτηση που επιβάλουμε.

Ορισμός 3.4 (Clinching Auction)**.** Θεωρούμε ότι έχουμε μία μονάδα διαιρέσιμου αγαθού (divisible) και n agents με valuations $v_1, v_2, \ldots, v_n$ και budgets $B_1, B_2, \ldots, B_n$. Έστω ότι $v_i \neq v_j$, εάν $i \neq j$. Ονομάζουμε *ενεργό σύνολο (Active Set)* το σύνολο:
$$A(p) = \{i; p < v_i\}$$
και *τερματικό σύνολο (Clinching Set)* το σύνολο:
$$C(p) = \left\{ i \in A(p); S(p) = \sum_{j \in A(p), j \neq i} \frac{B_j - p_j(p)}{p} \right\} \tag{3.11}$$

Καθορίζουμε τώρα τα $x_i(p)$ και $p_i(p)$ με βάση τους ακόλουθους κανόνες:

- Για $p \notin \{v_1, v_2, \ldots, v_n\}$, οι $x_i(p)$, $p_i(p)$ είναι διαφορίσιμες και οι παράγωγοι για $i \in C(p)$ δίνονται από τις σχέσεις:

$$\partial_p x_i(p) = \frac{S(p)}{p} \tag{3.12}$$

$$\partial_p p_i(p) = S(p) \tag{3.13}$$

και για $i \notin C(p)$:

$$\partial_p x_i(p) = \partial_p p_i(p) = 0 \tag{3.14}$$

- Με δεδομένη μία συνάρτηση f και μια τιμή p ορίζουμε τα εξής: $f(p-) = \lim_{x\uparrow p} f(x)$ και $f(p+) = \lim_{x\downarrow p} f(x)$. Έτσι, για $p = v_i$, ορίζουμε:

$$\delta_j^i = \left[S(v_i-) - \sum_{k\in A(v_i), k\neq j} \frac{B_k - p_j(v_i-)}{v_i} \right]$$

και

$$x_j(v_i+) = x_j(v_i) = x_j(v_i-) + \delta_j^i$$
$$p_j(v_i+) = p_j(v_i) = p_j(v_i-) + \delta_j^i$$

για $j \in A(v_i)$ ενώ για $j \notin A(v_i)$ ορίζουμε:

$$x_j(v_i+) = x_j(v_i) = x_j(v_i-)$$
$$p_j(v_i+) = p_j(v_i) = p_j(v_i-)$$

Οι τελικές αναθέσεις και οι πληρωμές της clinching δημοπρασίας δίνονται από τα όρια $\lim_{p\to\infty} x_i(p)$ και $\lim_{p\to\infty} p_i(p)$ αντίστοιχα. Εφόσον οι τιμές των $x_i(p)$ και $p_i(p)$ είναι σταθερές για $p > \max_i v_i$, φτάνουμε στα παραπάνω όρια για πεπερασμένη τιμή.

Διαισθητικά, το δ_j^i εκφράζει την ασυνέχεια για κάποιον agent με valuation v_j, εάν κάποιος άλλος agent έχει valuation v_i και εμείς απλά ρίξουμε τον bidder στην τιμή v_j.

Με βάση το [41], η Clinching Auction είναι η μοναδική η οποία είναι truthful, individually rational (σέβεται τα budgets) και είναι Pareto Optimal, όταν τα budgets είναι public. Οι αποδείξεις για τα παραπάνω, βρίσκονται στο ίδιο paper.

Η Clinching Auction, ωστόσο, εμφανίζει μία ενδιαφέρουσα συμπεριφορά, η οποία αποτελεί το δεύτερο χαρακτηριστικό που θα θέλαμε να έχει το Liquid Welfare: να μπορεί να προσεγγίζει γνωστά προβλήματα.

Θεώρημα 3.1. *Η Clinching Auction είναι 2-approximation στο Liquid Welfare. Με άλλα λόγια, με δεδομένους n agents με αξίες ανά τιμή v_i και budgets B_i, έστω x, p το αποτέλεσμα της clinching auction για μία τέτοια είσοδο. Τότε,*

$$\bar{W} \geq \frac{1}{2}\bar{W}^*$$

3.4 Ισορροπία Αγοράς και Liquid Welfare

Σε αυτή την ενότητα θα δείξουμε αρχικά ότι μπορεί να σχεδιαστεί εκ νέου δημοπρασία, η οποία είναι 2-προσεγγιστική στο Liquid Welfare και να υλοποιήσουμε με αυτόν τον τρόπο και την 3 προϋπόθεση που θέλαμε για αυτή τη μετρική και στη συνέχεια θα δείξουμε έναν άλλον αλγόριθμο, ο οποίος τυπικά αποτελεί στιγμιότυπο των Multiplicative Updates που, όμως, θεωρούμε ότι συγκεντρώνουν μεγάλες ομοιότητες οι οποίες χρήζουν περαιτέρω μελέτης.

Έστω, για τις ανάγκες της δημοπρασίας μας, ότι έχουμε μία αγορά με n αγοραστές, καθένας εκ των οποίων έχει budget B_i και valuation v_i ανά μονάδα ενός και μόνο διαιρέσιμου αγαθού. Για αυτή την περίπτωση:

Ορισμός 3.5 (Market Clearing Price)**.** Μία τιμή p ονομάζεται *Τιμή που Καθαρίζει την Αγορά (Market Clearing Price)* εάν ανατίθεται σε κάθε αγοραστή ένα βέλτιστο bundle αγαθών (στη συγκεκριμένη περίπτωση, ένα βέλτιστο «κομμάτι» του αγαθού) έτσι ώστε να μην υπάρχει ούτε απόθεμα, ούτε έλλειψη κανενός αγαθού.

Παρατήρηση 3.1. *Υπάρχει μόνο μία Market Clearing Price.*

Απόδειξη. Έστω ότι υπάρχουν δύο market clearing prices, p_1^*, p_2^*. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι $p_1^* < p_2^*$. Τότε, όμως, δεν θα μπορέσουμε να πουλήσουμε ορισμένες ποσότητες του προϊόντος στην τιμή p_2^*, άρα θα υπάρχει πλεόνασμα, ενώ αντίστοιχα στην τιμή p_1^* θα έχουμε περισσότερη ζήτηση και άρα θα έχουμε έλλειμα. Άρα, καμία από τις δύο δεν είναι πραγματικά market clearing price και καταλήγουμε σε άτοπο. ∎

Εύκολα επίσης συμπεραίνουμε ότι μόλις βρεθεί η Market Clearing Price μετά μπορούμε να βρούμε και ακριβώς τις αναθέσεις που πρέπει να γίνουν. Και ακριβώς αυτό είναι αυτό που κάνει η Δημοπρασία Ομοιόμορφης Τιμής (Uniform Price Auction). Ωστόσο, **προσοχή**: με αυτόν τον τρόπο βρίσκουμε μόνο τις **αναθέσεις**. Οι πληρωμές καθορίζονται από το Myerson lemma.

Ορισμός 3.6 (Uniform Price Auction)**.** Έστω n agents με valuations $v_1 \geq v_2 \geq \cdots \geq v_n$ και budgets B_i. Ορίζουμε μία δημοπρασία η οποία αναθέτει μία μονάδα ενός (διαιρέσιμου) αγαθού ως εξής: έστω k είναι ο μεγαλύτερος ακέραιος, τέτοιος ώστε: $\sum_{j=1}^{k} B_j \leq v_k$, τότε:

- Περίπτωση I: εάν $\sum_{j=1}^{k} B_j > v_{k+1}$ αναθέτουμε $x_i = \frac{B_i}{\sum_{j=1}^{k} B_j}$, για $i = 1, 2, \ldots, k$ και τίποτα σε όλους τους εναπομείναντες agents.
- Περίπτωση II: εάν $\sum_{j=1}^{k} B_j \leq v_{k+1}$ αναθέτουμε $x_i = \frac{B_i}{v_{k+1}}$ για $i = 1, 2, \ldots, k$ και τίποτα σε όλους τους εναπομείναντες agents.

  Οι πληρωμές καθορίζονται με βάση το ολοκλήρωμα του Myerson (1.1).

Στην περίπτωση I, δηλαδή, market clearing price είναι $p^* = \sum_{j=1}^{k} B_j$, ενώ στην περίπτωση II είναι $p^* = v_{k+1}$. Ουσιαστικά, όταν θέλουμε να μην υπάρχει πλεόνασμα, θέλουμε να ελέγχουμε ότι οι ζητήσεις των agents για τα επιμέρους κομμάτια του αγαθού, δεν θα υπερβαίνουν την προσφορά, δηλαδή το 1. Άρα, έχουμε ότι:

$$\sum_{i} D_i \leq S \tag{3.15}$$

Θέλουμε επίσης, τα budgets όσων των valuation ξεπερνάει την market clearing price. Άρα, έχουμε ότι:

$$\sum_{i:v_i \geq p} B_i \leq p \tag{3.16}$$

Τέλος, αρκεί να κρατάμε πάντα κατά νου ότι ο τρόπος να μειώσουμε τη ζήτηση ενός αγαθού είναι:

i Είτε αυξάνοντας την τιμή του, αλλά κρατώντας την κάτω από το valuation (περίπτωση I)

ii Είτε αυξάνοντας τόσο την τιμή του, ώστε να υπερβαίνει το valuation (περίπτωση II)

Στο paper [47] αποδεικνύεται επίσης, ότι η Uniform Price Auction είναι *μονότονη*, είναι *budget feasible*.

Θεώρημα 3.2. *Η Uniform Price Auction είναι truthful* 2- *προσεγγιστική ως προς το Liquid Welfare.*

3.5 Μία $O(log^2 n)$-προσεγγιστική δημοπρασία για το Liquid Welfare

Σε αυτή την ενότητα, θα μιλήσουμε για μία δημοπρασία που παρουσιάστηκε στο [47]. Η συγκεκριμένη δημοπρασία ήταν εμπνευσμένη από μία τεχνική των Bartal, Gonen, Nisan στο [49]. Το πλαίσιο που θα εργαστούμε περιλαμβάνουμε 1 διαιρέσιμο αντικείμενο και παίκτες με υποαθροιστικές (subadditive)[3] valuations, και private budgets. Αξίζει εδώ να σημειώσουμε επίσης, ότι εάν το πλαίσιο εργασίας μας περιλαμβάνει είτε subadditive valuations ή private budgets, τότε είναι (αποδεδειγμένα) αδύνατο να επιλυθεί ως προς το Pareto Optimality. Τέλος, με την παρουσίαση και αυτής της δημοπρασίας κλείνει ο κύκλος των 3 προϋποθέσεων που θέλαμε να πληροί η νέα μας μετρική, το Liquid Welfare, εφόσον θα έχουμε βρει μια νέα δημοπρασία που θα προσεγγίζει το Liquid Welfare.

Επαναλαμβάνουμε, λοιπόν, το πλαίσιο μας είναι: 1 διαιρέσιμο αγαθό, κάθε παίκτης έχει subadditive valuation $v_i : [0,1] \to \mathbb{R}_+$ και budget B_i. Η ουσιαστική μας διαφορά με το προηγούμενο πρόβλημα που μελετήσαμε είναι ότι τώρα, τα budgets των παικτών μας είναι private.

Ορισμός 3.7 (Liquid Valuation). Ονομάζουμε *ρευστή ευχαρίστηση (liquid valuation)* το ελάχιστο μεταξύ του valuation ενός παίκτη για κάποιο συγκεκριμένο allocation και του budget του, δηλαδή:

$$\bar{v_i} = \min\{v_i(x_i), B_i\}$$

Προτού ξεκινήσουμε με την περιγραφή της δημοπρασίας θα ήθελα να σταθώ στο εξής: Εύλογα γεννάται το ερώτημα, πώς (εφόσον τα budgets αλλά και τα valuations είναι private) θα μπορούσαμε να κάνουμε τη δημοπρασία μας truthful και να δώσουμε κίνητρα στους παίκτες μας να παίξουν με τις αληθινές τους τιμές; Η απάντηση σε αυτή την ερώτηση, μοιάζει κάπως με την απάντηση που θα έδινε κάποιος posted-price μηχανισμός: Θα τους αφήσουμε να διαλέξουν μόνοι τους ποιο κομμάτι του αγαθού (δηλαδή, ποιο allocation) τους μεγιστοποιεί το utility. Με αυτό τον τρόπο, αν κάποιος προσπαθήσει να πει ψέμματα, θα πει ψέμματα στον ίδιο του τον εαυτό και θα οδηγηθεί να πληρώσει παραπάνω από το budget του ή να πάρει κάποιο κομμάτι αντικειμένου που δεν θα θέλει τόσο, κάτι που είναι αντίθετο με τις αρχές του individual rationality.

[3] Μια συνάρτηση ονομάζεται subadditive όταν για τη συνάρτηση αυτή ισχύει το εξής: $v_i(x_1 + x_2) \leq v_i(x_1) + v_i(x_2)$

Έστω ότι ο r είναι κάποιος παίκτης. Θεωρούμε τον επόμενο μηχανισμό, προκειμένου να πουλήσουμε *το μισό* αγαθό, στους παίκτες $i \neq r$, χρησιμοποιώντας μόνο την πληροφορία για το $\bar{v}_r\left(\frac{1}{2}\right)$.

1. Διαιρούμε τώρα το κομμάτι $[0, \frac{1}{2}]$ σε $k = 8\log(n)$ κομμάτια, μεγέθους $\frac{1}{2k}$ έκαστο.

2. Συνδέουμε κάθε κομμάτι $i = 1, \ldots, k$ με τιμή ανά μονάδα $p_i = \frac{2^i}{8}\bar{v}_r\left(\frac{1}{2}\right)$.

3. Ταξινομούμε αυθαίρετα όλους τους παίκτες εκτός από τον παίκτη r.

4. Ο κάθε παίκτης, εκτός από τον r, όταν έρθει η σειρά του, παίρνει το υποσύνολο του $[0, \frac{1}{2}]$ το οποίο τον συμφέρει περισσότερο και δεν έχει ανατεθεί ακόμη σε κανέναν άλλον παίκτη, κάτω από τις προκαθορισμένες τιμές. Οι παίκτες δεν επιτρέπει να πληρώσουν περισσότερο από το budget τους.

   Πιο συγκεκριμένα, έστω $p : [0, \frac{1}{2}] \to \mathbb{R}_+$ τέτοιο ώστε: $x \in \left[\frac{1}{2k}(i-1), \frac{1}{2k}i\right]$, $p(x) = p_i = \frac{2^i}{8}\bar{v}_r\left(\frac{1}{2}\right)$. Τώρα, για $i = 1, \ldots, r-1, r+1, \ldots, n$ έστω ότι το x_i μεγιστοποιεί $v_i(x_i) - \int_{z_i}^{z_i+x_i} p(t)dt$, όπου $z_i = \sum_{j<i} x_j$, με την προϋπόθεση βέβαια ότι οι πληρωμές είναι μικρότερες από το budget, δηλαδή ότι $\int_{z_i}^{z_i+x_i} p(t)dt \leq B_i$. Ορίζουμε, τότε, την πληρωμή να είναι: $\pi_i = \int_{z_i}^{z_i+x_i} p(t)dt$.

Algorithm 6: Sell-Without-r

Ο αλγόριθμος 6 θα χρησιμοποιηθεί στον τελικό μας μηχανισμό που ακολουθεί.

1. Με δεδομένο ένα διαιρέσιμο αγαθό και n παίκτες με valuations $v_i(\cdot)$ και budgets B_i, θεωρούμε την ακόλουθη δημοπρασία: Έστω $r_1 = \arg\max_i \bar{v}_i\left(\frac{1}{2}\right)$ και $r_2 = \arg\max_{i \neq r_1} \bar{v}_i\left(\frac{1}{2}\right)$. Ονομάζουμε τον r pivot παίκτη. Έστω ότι (x, π) είναι το αποτέλεσμα του Sell-Without-r_1(6) για τους παίκτες $[x] \setminus r_1$ και (x', π') το αποτέλεσμα του Sell-Without-r_2 για τους παίκτες $[n] \setminus r_2$.

2. Για τους παίκτες $i \neq r_1$, ανάθεσέ τους x_i και χρέωσέ τους π_i. Για τον r_1:

$$(x_{r_1}, \pi_{r_1}) = \begin{cases} (x'_{r_1}, \pi'_{r_1}) & \text{if } v_{r_1}(x'_{r_1}) - \pi'_{r_1} \geq v_{r_1}(\frac{1}{2}) - 2\bar{v}_{r_2}\left(\frac{1}{2}\right) \\ (\frac{1}{2}, 2\bar{v}_{r_2}) & \text{if } \textit{not} \end{cases} \tag{3.17}$$

Algorithm 7: Estimate-and-Price

Παρατήρηση 3.2. *Η παραπάνω δημοπρασία είναι feasible εφόσον ο r_1 μπορεί να πάρει το πολύ το μισό αντικείμενο, ενώ μισό αντικείμενο το πολύ μπορούν να πάρουν και οι υπόλοιποι $[n] \setminus r_1$ παίκτες αθροιστικά.*

Προτού προχωρήσουμε στην ανάλυση της δημοπρασίας είναι αρκετά ενδιαφέρον να δώσουμε τη διαίσθηση πίσω από τις λεπτομέρειες του αλγορίθμου. Καταρχάς, ας παρατηρήσουμε την τιμή του κάθε κομματιού $p_i = \frac{2^i}{8}\bar{v}_r\left(\frac{1}{2}\right)$. Η τιμή αυτή βλέπουμε ότι για το πρώτο κομμάτι είναι $p_1 = \frac{1}{4}\bar{v}_r\left(\frac{1}{2}\right)$. Εδώ λοιπόν, θεωρούμε ότι εφόσον ο r έχει το μεγαλύτερο valuation για το μισό αντικείμενο, είναι λογικό να ξεκινήσουμε με starting price στο 1/4 του μέγιστου valuation. Επιπλέον, βλέπουμε ότι έχει τελική τιμή $p_k = \bar{v}_r\left(\frac{1}{2}\right)$, τιμή στην οποία κανείς δεν μπορεί να αγοράσει πλην του r. Βλέπουμε, επίσης, ότι καθώς προχωράμε στα κομμάτια, η τιμή ανεβαίνει. Η ιδέα πίσω από αυτή τη σκέψη πηγάζει από το Νόμο της Αγοράς και της Ζήτησης, σύμφωνα με τον οποίον πρακτικά όσο μειώνονται τα διαθέσιμα αποθέματα, τόσο θα ανεβαίνει η τιμή των αποθεμάτων. Τέλος, αξίζει εδώ να παρατηρήσουμε την καθοριστική συμβολή ενός κανόνα Multiplicative Updates που κρύβεται στον τρόπο που αυξάνεται η τιμή ανά κομμάτι: κάθε επόμενο κομμάτι έχει δύο φορές την τιμή του προηγούμενου.

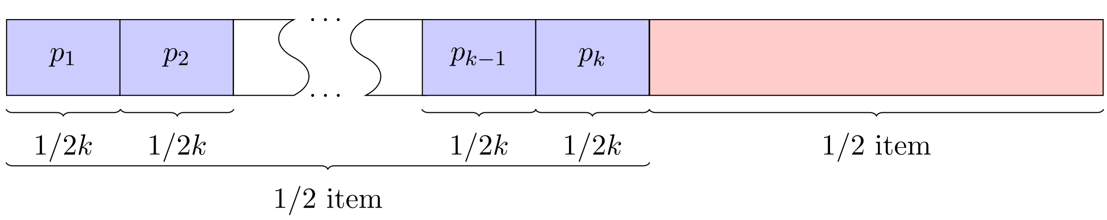
Προχωράμε τώρα στην μαθηματική ανάλυση της απόδοσης της δημοπρασίας.

Λήμμα 3.2. *Η Υπολόγισε-και-Τιμολόγησε (Estimate-and-Price) δημοπρασία είναι truthful για παίκτες με private budget.*

Απόδειξη. Η ιδέα της απόδειξης είναι να επιχειρηματολογήσουμε σχετικά με το ότι δεν υπάρχει καμία απόκλιση όπου ο παίκτης να αλλάζει το valuation ή το budget του και να τον συμφέρει περισσότερο. Για τον παίκτη $i \neq r_1$, ενώ ο i αποκλίνει και **δεν** γίνει ο pivot παίκτης, τότε παραμένουν σταθερά και το allocation και η πληρωμή του (οι παίκτες εκτός του r_1 ταξινομούνται αυθαίρετα). Τώρα, ο εν λόγω παίκτης θα μπορούσε να προσπαθήσει να αποκλίνει και να γίνει ο pivot παίκτης. Σε αυτή την περίπτωση, ωστόσο, είτε έχει το ίδιο allocation και πληρωμή όπως προηγουμένως, είτε παίρνει το μισό αντικείμενο $\left(\frac{1}{2}\right)$ και πληρώνει $2\bar{v}_{r_1}\left(\frac{1}{2}\right) > \bar{v}_i\left(\frac{1}{2}\right)$ καταλήγοντας σε αρνητικό utility (άτοπο, γιατί έχουμε individually rational παίκτε) ή με έλλειμμα στο budget του.

Όσο για τον pivot παίκτη r_1, δεν μπορεί να ωφεληθεί από το να μειώσει το $\bar{v}_{r_1}\left(\frac{1}{2}\right)$ ώστε να μην είναι πλέον ο pivot παίκτης. Αν καταφέρει και το μειώσει τόσο, ώστε να μην είναι ο pivot, τότε θα γίνει pivot ο r_2 και ο r_1 θα πάρει όσο κομμάτι του αντιστοιχεί με βάση τον μηχανισμό Sell-Without-r_2, κάτι το οποίο του είναι άχρηστο. Αν, παίζοντας truthfully, έπαιρνε μισό αγαθό, τότε θα είχε φτάσει το μέγιστο που θα μπορούσε να πάρει και καμία άλλη στρατηγική δεν θα του επέφερε το ίδιο αποτέλεσμα. Αν, παίζοντας truthfully, έπαιρνε ό,τι του αναλογούσε από τον Sell-Without-r_2, τότε η απόκλισή του δεν θα του αποφέρει καμία διαφορά στο αποτέλεσμα. ■

Σχηματικά, μπορούμε να δούμε το διαχωρισμό που κάνουμε πάνω στο αντικείμενο ως εξής:



Μία $O(\log n)$-προσέγγιση για τους submodular παίκτες

Θα δείξουμε, πρώτα, ότι η δημοπρασία Estimate-and-Price είναι $O(\log n)$-προσέγγιση, όταν οι bidders μας αποτελούν μία υποκλάση των subadditive (submodular). Στη συνέχεια, παραλλάσσοντας ελαφρώς αυτή την απόδειξη και χάνοντας μόνο έναν $O(\log n)$ παράγοντα, θα προκύψει το επιθυμητό αποτέλεσμα για τους subadditive.

Μια συνάρτηση valuation είναι submodular όταν είναι subadditive και κοίλη (concave). Αποδεικνύουμε στη συνέχεια ότι στην Sell-Without-r_1 δεν πωλούνται όλα τα κομμάτια του αντικειμένου.

Λήμμα 3.3. *Έστω x το αποτέλεσμα της Sell-Without-r_1. Τότε, ισχύει ότι: $\sum_{i \neq r_1} x_i < \frac{1}{2}$.*

Απόδειξη. Για να αποδείξουμε το παραπάνω λήμμα, αρκεί να συγκρίνουμε δύο τιμές: το μέγιστο συνολικό ποσό που πληρώνουν οι παίκτες και πόσο θα κόστιζε να έχει πωληθεί ολόκληρο το $\frac{1}{2}$ κομμάτι του αντικειμένου.

Μέγιστο ποσό που πληρώνουν:

$$\begin{aligned}\sum_{i \neq r_1} \pi_i &\leq \sum_{i \neq r_1} \bar{v}_i(x_i) \quad \text{(λόγω individual rationality)} \\ &\leq \sum_{i \neq r_1} \bar{v}_i\left(\frac{1}{2}\right) \\ &\leq n \cdot \bar{v}_{r_1}\left(\frac{1}{2}\right) \quad \text{(από τον ορισμό του } r_1\text{)}\end{aligned}$$

Ποσό που θα έπρεπε να πληρώνουν αν αγόραζαν το μισό αντικείμενο:

$$\begin{aligned}\sum_{i \neq r_1} \pi_i &= \int_0^{1/2} p(t)dt \\ &\leq \int_{k-1/2k}^{1/2} p(t)dt \\ &= \frac{1}{2k} \cdot \frac{2^k}{8} \bar{v}_{r_1}\left(\frac{1}{2}\right) \\ &= \frac{n^8}{16 \cdot 8 \cdot \log n} \bar{v}_{r_1}\left(\frac{1}{2}\right) \\ &> n\bar{v}_{r_1}\left(\frac{1}{2}\right) \quad \text{γιατί } n^7 > 16 \cdot 8 \log n, \text{ για } n \geq 1.7\end{aligned}$$

Επομένως, αν μπορούσαν να αγοράσουν ολόκληρο (το μισό) αντικείμενο θα έπρεπε να πληρώσουν παραπάνω από το μέγιστο που έχουν δυνατότητα να πληρώσουν. ■

Λήμμα 3.4. *Έστω $x^{\#}$ η λύση του $\max \bar{\mathbf{W}}(x^{\#})$, τέτοια ώστε να ισχύει: $x^{\#}_{r_1} = 0$ και $\sum_{i \neq r_1} x^{\#}_i = \frac{1}{2}$, τότε: $\bar{v}_{r_1}\left(\frac{1}{2}\right) + \bar{\mathbf{W}}(x^{\#}) \geq \frac{1}{2}\mathbf{W}^*$.*

Απόδειξη. Έστω x^* είναι το βέλτιστο αποτέλεσμα, έτσι ώστε: $\bar{\mathbf{W}}^* = \bar{\mathbf{W}}(x^*)$. Επιπλέον, ισχύει το εξής:

$$\frac{1}{2} \geq \frac{1}{2} x^*_{r_1} \tag{3.18}$$

Καθώς επίσης και ότι το $\frac{1}{2}x^*_{-r_1}$ ικανοποιεί τις απαιτήσεις του προγράμματος που καθορίζει το $x^\#$. Τότε:

$$\begin{aligned} \bar{v}_{r_1}\left(\frac{1}{2}\right) + \bar{\mathbf{W}}(x^\#) &\leq \bar{\mathbf{W}}\left(\frac{1}{2}x^*\right) \quad \text{λόγω του 3.18} \\ &\leq \frac{1}{2}\bar{\mathbf{W}}(x^*) \quad \text{λόγω της κοιλότητας του } \bar{\mathbf{W}} \end{aligned}$$

■

Θα διατυπώσουμε και θα αποδείξουμε τώρα ένα ακόμη λήμμα, το οποίο διαισθητικά μας λέει ότι κανένα άλλο allocation δεν με συμφέρει σε ό,τι αφορά το Liquid Welfare, ακόμη και αν δοκιμάσω να αγοράσω το φθηνότερο απούλητο προϊόν.

Λήμμα 3.5. *Για submodular παίκτες, έστω x το αποτέλεσμα της δημοπρασίας Estimate-and-Price και $\bar{p}$ η τιμή του φθηνότερου απούλητου προϊόντος, δηλαδή: $\bar{p} = \lim_{t\downarrow(\sum_{i\neq r_1})} x_i p(t)$. Έστω x' οποιαδήποτε άλλο allocation. Τότε, για κάθε $i \neq r_1$: $\bar{v}(x_i) \geq \bar{v}(x'_i) - \bar{p}x'_i$.*

Απόδειξη. Εάν $x_i \geq x'_i$ τότε το παραπάνω αποτέλεσμα ισχύει, λόγω της μονοτονίας του $\bar{v}_i$. Εάν δεν $x_i < x'_i$, τότε ο i δεν πήρε παραπάνω αντικείμενα για κάποιον από τους δύο λόγους [4]:

I. Εξάντλησε το budget του. Δηλαδή, θα ισχύει ότι $\pi_i = B_i \Rightarrow \bar{v}_i(x_i) = B_i \geq \bar{v}_i(x'_i) \geq \bar{v}_i(x'_i) - \bar{p}x'_i$.

II. Η τιμή του αγαθού ξεπερνά την οριακή του αξία. Σε αυτή την περίπτωση, δηλαδή: $v_i(x_i) \geq v_i(x_i + \epsilon) - p'\epsilon$, για κάθε $\epsilon \in [0, \epsilon_0]$, για κάποιο μικρό ϵ_0. Λόγω της κοιλότητας του v_i έχουμε ότι:

$$v_i(x'_i) - v_i(x_i) \leq p'(x'_i - x_i) \leq \bar{p}x'_i \tag{3.19}$$

Συνεπώς:

$$\begin{aligned} \bar{v}_i(x_i) &= v_i(x_i) \\ &\geq v_i(x'_i) - \bar{p}x'_i \quad 3.19 \\ &\geq \bar{v}_i(x'_i) - \bar{p}x'_i \quad \text{ορισμός liquid valuation} \end{aligned}$$

■

Θεώρημα 3.3. *Για submodular παίκτες, η Estimate-and-Price δημοπρασία είναι $O(\log n)$-προσέγγιση στο Liquid Welfare.*

Απόδειξη. Θα δώσουμε τη γενική ιδέα της απόδειξης και αφήσουμε τις λεπτομέρειες, οι οποίες μπορούν να βρεθούν στο paper [47]. Η απόδειξη στηρίζεται στον χωρισμό δύο περιπτώσεων, ανάλογα με το τι μερίδιο του βέλτιστου Liquid Welfare παράγει ο παίκτης r_1 και χρησιμοποιώντας τα Λήμματα 3.3, 3.4, 3.5. Οι δύο περιπτώσεις που εξετάζονται είναι οι:

[4]Αξίζει να σημειώσουμε εδώ ότι με βάση το Λήμμα 3.3 υπάρχουν προϊόντα τα οποία είναι διαθέσιμα προς πώληση.

I. $\bar{v}_{r_1}\left(\frac{1}{2}\right) \geq 4\bar{\mathbf{W}}^*$

II. $\bar{v}_{r_1}\left(\frac{1}{2}\right) < 4\bar{\mathbf{W}}^*$

■

Θεώρημα 3.4. *Για subadditive παίκτες, η δημοπρασία Estimate-and-Price είναι $O(\log^2 n)$-προσέγγιση του Liquid Welfare.*

Απόδειξη. Παρατηρούμε ότι το μόνο σημείο στο οποίο χρησιμοποιήσαμε την κοιλότητα της v_i ήταν στην απόδειξη του Λήμματος 3.5. Για οποιαδήποτε άλλη απόδειξη διατυπώσαμε, αρκούσε το ότι η v_i ήταν subadditive. Για να αποδείξουμε τώρα το θεώρημα για subadditive παίκτες, θα επανακαθορίσουμε το $x^{\#}$ ως το $\arg\max \bar{\mathbf{W}}(x^{\#})$ με δεδομένο ότι: $x_i^{\#} = 0$, $\sum_i x_i^{\#} = \frac{1}{2}$ και $x_i^{\#} \leq \frac{1}{2k}$. Η τελευταία συνθήκη υποδηλώνει ουσιαστικά ότι κανείς παίκτης στην απόδειξη του Λήμματος 3.5 δεν χρειάζεται να πληρώσει περισσότερο από $2\bar{p}$ ανά μονάδα. Επομένως, μπορούμε να δείξουμε ότι: $\bar{v}_i(x_i) \geq \bar{v}_i(x_i^{\#}) - 2\bar{p}x_i^{\#}$.

Η μόνη διαφορά είναι ότι κάτω από αυτόν τον νέο ορισμό, το κενό μεταξύ $\bar{\mathbf{W}}^*$ και $\bar{v}_{r_1}\left(\frac{1}{2}\right) + \bar{\mathbf{W}}^*$ δεν είναι πια σταθερό, αλλά $O(k) = O(\log n)$. Αυτή η διαφορά μας κάνει να χάσουμε άλλον έναν $O(\log n)$ παράγοντα στον λόγο προσέγγισης. ■

Για να αποκτήσουμε καλύτερη διαίσθηση γύρω από το πώς ακριβώς τρέχει ο αλγόριθμος και πώς παίρνει τις αποφάσεις στο κάθε βήμα, θα παρουσιάσουμε τώρα ένα παράδειγμα εκτέλεσής του. Έστω ότι έχουμε $n = 4$ παίκτες με valuation functions και budgets που φαίνονται παρακάτω:

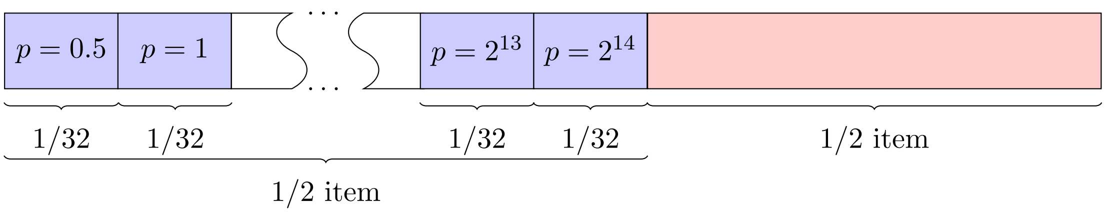
Players	Valuations	Budgets
1	$v_1(x) = \sqrt{x}$	$B_1 = 3$
2	$v_2(x) = \sqrt{2x}$	$B_2 = 2$
3	$v_3(x) = \sqrt{9x}$	$B_3 = 1.5$
4	$v_4(x) = \sqrt{16x}$	$B_4 = 2$

Παρατήρηση 3.3. *Παρατηρούμε ότι οι valuation functions των παικτών μας πληρούν την προϋπόθεση του subadditivity, εφόσον για την συνάρτηση της τετραγωνικής ρίζα ισχύει εξ ορισμού ότι $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$.*

Με βάση τον Αλγόριθμο 6 θα χωρίσουμε το μισό αντικείμενο σε $k = 8\log(4) = 16$ κομμάτια, μεγέθους $\frac{1}{32}$ το καθένα. Το επόμενο βήμα είναι να βρούμε τον pivot player, δηλαδή τον παίκτη για τον οποίο ισχύει ότι έχει $\max \bar{v}(\frac{1}{2})$. Ο παρακάτω πίνακας μας δείχνει ακριβώς πώς τον βρίσκουμε αυτόν τον παίκτη:

Players	Liquid Welfare
1	$\bar{v}_1(\frac{1}{2}) = \min\left\{\frac{\sqrt{2}}{2}\right\} = \frac{\sqrt{2}}{2}$
2	$\bar{v}_2(\frac{1}{2}) = \min\{1, 2\} = 1$
3	$\bar{v}_3(\frac{1}{2}) = \min\left\{\sqrt{\frac{9}{2}}, 1.5\right\} = 1.5$
4	$\bar{v}_4(\frac{1}{2}) = \min\{2.828, 2\} = 2$

Βλέπουμε ότι ο pivot player είναι ο 4 και οι τιμές για το κάθε κομμάτι διαμορφώνονται ως $p_i = \frac{2^i}{8} \cdot 2$:



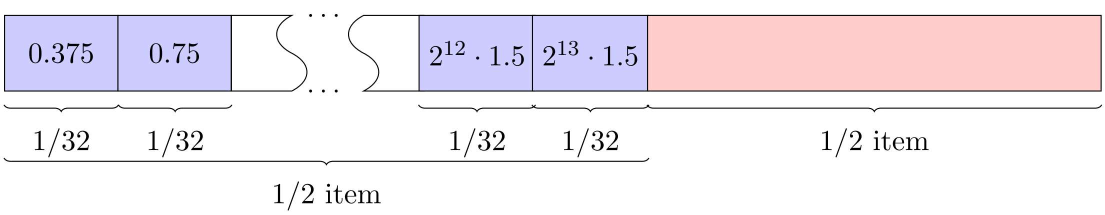
Στη συνέχεια ορίζουμε μία σειρά για την έλευση των παικτών. Για ευκολία (και χωρίς βλάβη της γενικότητας) διαλέγουμε οι παίκτες να έρθουν με τη σειρά: $1, 2, 3, 4$.

1. Ο παίκτης 1 λόγω του budget $B_1 = 3$ μπορεί να αγοράσει μόνο κάποιους συνδυασμούς από τα 3 πρώτα κομμάτια του αντικειμένου, με τιμές $p_1 = 0.5, p_2 = 1, p_3 = 2$. Παρατηρούμε, όμως, ότι το utility του είναι αρνητικό για κάθε συνδυασμό (λόγω μικρής valuation) και επομένως, λόγω του individual rationality, επιλέγει να μην αγοράσει τίποτα και μην πληρώσει τίποτα. Άρα, $(x_1, \pi_1) = (0, 0)$

2. Ο παίκτης 2 λόγω του budget $B_2 = 2$ μπορεί να αγοράσει και πάλι μόνο κάποιους συνδυασμούς από τα 3 πρώτα κομμάτια του αντικειμένου. Ο παίκτης 2 για να αγοράσει το κομμάτι $[0, 1/32]$ έχει utility $v_2(1/32) - p_1 = 0.25 - 0.5 < 0$, επομένως δεν το αγοράζει. Ομοίως, προκύπτει και για αυτόν τον παίκτη ότι δεν τον συμφέρει να αγοράσει κάποιο κομμάτι αγαθού. Επομένως, $(x_2, \pi_2) = (0, 0)$.

3. Ο παίκτης 3 λόγω του budget $B_3 = 1.5$ μπορεί να αγοράσει μόνο συνδυασμούς από τα 2 πρώτα κομμάτια του αντικειμένου. Για να αγοράσει το κομμάτι $[0, 1/32]$ έχει utility $v_3(1/32) - 0.5 = 0.53 - 0.5 = 0.03 > 0$, επομένως θα μπορούσε να το αγοράσει αυτό το κομμάτι. Για το κομμάτι $[[1/32, 1/16]$ έχει αρνητικό utility και δεν θα μπορούσε να το αγοράσει. Για το κομμάτι $[0, 16]$ έχει utility $v_3(1/16) - (p_1 + p_2) < 0$, επομένως, ο παίκτης 3 παίρνει το κομμάτι 1. Άρα, $(x_3, \pi_3) = ([0, 32], 0.5)$.

4. Για τον παίκτη 4 τώρα, επειδή ήταν ο pivot player για τον 6 πρέπει να τρέξουμε τον αλγόριθμο 6 με pivot player τον παίκτη για τον οποίον ισχύει $r_2 = \arg\max_{n \setminus r_1} \min\{v_i, B_i\}$, δηλαδή τον παίκτη 3. Τώρα, ισχύει ότι $\bar{v}_r = 1.5$ και οι τιμές διαμορφώνονται ως εξής:

   Για τη Sell-Without-r_2 δημοπρασία ο παίκτης 1 μπορεί να αγοράσει λόγω του budget του κάποιον συνδυασμό από τα 4 πρώτα κομμάτια. Έχει, όμως, αρνητικό utility για όλα, οπότε δεν αγοράζει κάτι. Ο παίκτης 2 μπορεί να αγοράσει κάποιον συνδυασμό από τα 3 πρώτα κομμάτια αλλά έχει και αυτός αρνητικό utility, οπότε δεν αγοράζει κάτι. Ο παίκτης 4 μπορεί και αυτός να αγοράσει κάποιον συνδυασμό από τα 3 πρώτα κομμάτια. Για το κομμάτι $[0, 1/32]$ έχει utility $\sqrt{\frac{16}{32}} - 0.375 = 0.707 - 0.375 = 0.332$, για το κομμάτι $[0, 1/16]$ έχει $1 - (0.75 + 0.375) < 0$

ενώ για τα υπόλοιπα κομμάτια έχει αρνητικό utility. Επομένως, $(x'_{r_1}, \pi'_{r_1}) = ([0, 1/32], 0.375)$. Άρα, εφόσον ισχύει η ανισότητα: $v_{r_1}(x'_{r_1}) - \pi'_{r_1} \geq v_{r_1}(1/2) - 2\bar{v}_{r_2}(1/2)$, ο παίκτης 4 παίρνει τελικώς $[0, 1/32]$ και πληρώνει 0.375.

Κεφάλαιο 4

Προσεγγίζοντας το **Liquid Welfare** σε γνωστά προβλήματα

Σε αυτό το κεφάλαιο, θα παρουσιάσουμε τα αποτελέσματα της δικής μας έρευνας. Στόχος μας ήταν να σχεδιάσουμε truthful μηχανισμούς για Combinatorial Auctions και παίκτες με budget και να τους μελετήσουμε ως προς την αντικειμενική συνάρτηση του Liquid Welfare. Θέλαμε να επεκτείνουμε και να γενικεύσουμε προηγούμενα αποτελέσματα τα οποία υπήρχαν για την περίπτωση του *single item multi unit auction* καθώς και για *submodular combinatorial auctions.*

4.1 Συνδυαστικές Δημοπρασίες με Περιορισμούς Ρευστότητας

Προτού προχωρήσουμε στα συμπεράσματα στα οποία οδηγηθήκαμε από την έρευνά μας αξίζει να δώσουμε το μαθηματικό πλαίσιο πίσω από τις Combinatorial Auctions με budgets ως primal-dual Προβλήματα Γραμμικού Προγραμματισμού.

$$\max \sum_{i \in N} \bar{v}_i \tag{4.1}$$

$$\bar{v}_i \leq B_i \tag{4.2}$$

$$\bar{v}_i \leq \sum_{S \in U} v_i(S) x_i(S) \tag{4.3}$$

$$\sum_{S \in U} x_i(S) \leq 1, \forall i \in N \tag{4.4}$$

$$\sum_{i \in N} \sum_{S: j \in S} x_i(S) \leq 1, \forall \text{good } j \tag{4.5}$$

Το παραπάνω Primal LP είναι μία παραλλαγή του αρχικού LP (1.2) και αντιπροσωπεύει το πρόβλημα των combinatorial auctions με την προσθήκη των budgets. Θέλουμε να μεγιστοποιήσουμε το Liquid Welfare των παικτών μας (4.1). Οι ανισότητες 4.2 και 4.3 προκύπτουν από τον ορισμό του liquid welfare [1].

$$\min \sum_i B_i \alpha_i + \sum u_i + \sum_j p_j \tag{4.6}$$

$$\alpha_i + \beta_i \geq 1, \forall i \in N \tag{4.7}$$

$$u_i + \sum_{j \in S} p_j \geq \beta_i v_i(S), \forall i \in N \text{ and } \forall S \subseteq U \tag{4.8}$$

$$\tag{4.9}$$

Το οποίο μπορεί να απλοποιηθεί στο εξής Dual LP.

$$\min \sum_{i \in N} B_i \alpha_i + \sum_{i \in N} u_i + \sum_{j \in S} p_j \tag{4.10}$$

$$u_i + \sum_{j \in S} p_j \geq (1 - \alpha_i) v_i(S), \forall i \in N \text{ and } S \subseteq U \tag{4.11}$$

Παρατηρούμε εδώ ότι το Dual LP στο budgeted πλαίσιο μοιάζει πολύ με το κλασικό LP των Combinatorial Auctions. Η διαφορά από τα κλασικά Combinatorial Auctions στο Primal είναι ο πρώτος περιορισμός (4.2) και στο dual είναι οι μεταβλητές α_i, β_i. Παρατηρούμε ότι αν δεν υπάρχει περιορισμός budget, τότε το α_i είναι μηδέν και έχουμε το κλασικό combinatorial auction. Όμως, καθώς το budget μειώνεται, το α_i *αυξάνεται* στη βέλτιση dual λύση και έτσι, οδηγούμαστε σε «εκπτώσεις» στις τιμές των αντικειμένων, γιατί οι παίκτες δεν έχουν να πληρώσουν.

Στο paper [17] που αναλύσαμε σε προηγούμενο κεφάλαιο, είδαμε πώς μπορούμε να προσεγγίσουμε αρκετά καλά το Social Welfare για Combinatorial Auctions (και submodular bidders). Η ιδέα ήταν να συνδυάσουμε τις ιδέες των multiplicative updates και των posted-price online μηχανισμών προκειμένου να πετύχουμε αφενός το truthfulness και αφετέρου την καλή προσέγγιση στο Social Welfare.

Σε αυτό το πλαίσιο, η προσθήκη ενός επιπλέον περιορισμού (των budgets των παικτών), ίσως φαντάζει μία αμελητέα προσθήκη. Η πραγματικότητα είναι ότι με την προσθήκη του budget, αφενός μεν πρέπει να αλλάξουμε την αντικειμενική συνάρτηση που μας ενδιαφέρει και αφετέρου δε, η προσθήκη των budgets μπορεί να καταστήσει πολύ κακό τον μηχανισμό μας. Ο τρόπος μας να «εισάγουμε» την πληροφορία των budgets στο υπάρχον μας πλαίσιο είναι μέσω της δημιουργίας μίας καινούριας οικογένειας μαντείων, τα οποία θα λαμβάνουν υπόψιν και τόσο το valuation, όσο και το budget του εκάστοτε παίκτη.

4.1.1 Η αποτυχία του Demand Oracle

Σε αυτή την ενότητα, θα διαπιστώσουμε τους λόγους για τους οποίους το Demand Oracle αποτυγχάνει στην περίπτωση που οι παίκτες έχουν budgets. Η ιδέα είναι να καταφέρουμε να χτίσουμε τις εξής ανισότητες:

[1] Το Liquid Welfare πρέπει να είναι μικρότερο ή ίσο με το valuation και το budget των παικτών.

i. $v_i(S_i) \geq v_i(T_i) - \sum_{e \in T_i} p_e^i$

ii. $\bar{v}_i(S_i) \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^i$

όπου S_i το bundle που διαλέγει ο μηχανισμός μας και T_i κάθε άλλο εφικτό bundle που θα μπορούσε να διαλέξει. Στην πρώτη περίπτωση, αν καταφέρουμε να χτίσουμε μία τέτοια ανισότητα, είμαστε πολύ κοντά στο να προσεγγίσουμε το Social Welfare [2] ενώ στη δεύτερη περίπτωση, είμαστε πολύ κοντά στο να καταφέρουμε να προσεγγίσουμε το Liquid Welfare.

Θα δείξουμε αρχικά ότι το Demand Oracle στην περίπτωση που θέλουμε οι παίκτες να μην ξεπερνούν το budget τους δεν μπορεί να παράξει utility maximizing bundle. Με άλλα λόγια θα μελετήσουμε τη συμπεριφορά του $D_i'(U_i, p) = \max\left\{v_i(x_i) - \sum_{e \in U_i} p_e^i | \sum_{e \in U_i} p_e^i \leq B_i\right\}$.

Λήμμα 4.1. *Στην περίπτωση που χρησιμοποιούμε το oracle*

$$D_i'(U_i, p) = \max\left\{v_i(x_i) - \sum_{e \in U_i} p_e^i | \sum_{e \in U_i} p_e^i \leq B_i\right\}$$

δεν μπορούμε να παράξουμε utility maximizing bundle. Με άλλα λόγια αν T_i είναι οποιοδήποτε εφικτό bundle που θα μπορούσε να παράξει ο μηχανισμός και S_i το bundle που πραγματικά παράγει, δεν ισχύει η ανισότητα: $v_i(S_i) - \sum_{e \in S_i} p_e^i \geq v_i(T_i) - \sum_{e \in T_i} p_e^i$

Απόδειξη. Θα αποδείξουμε το λήμμα με ένα αντιπαράδειγμα. Έστω οποιοδήποτε εφικτό bundle T_i. Αν για το bundle αυτό ισχύει ότι $v_i(T_i) \to \infty$, αλλά $\sum_{e \in T_i} = B_i + \epsilon$, τότε ο μηχανισμός μας δεν θα επιλέξει το bundle T_i γιατί κοστίζει περισσότερο από όσα μπορεί ο χρήστης να πληρώσει. Παρολαυτά, ισχύει η ανισότητα $v_i(T_i) - \sum_{e \in T_i} \geq v_i(S_i) - \sum_{e \in S_i}$. ∎

Ωστόσο, θα πρέπει να ελέγξουμε επιπλέον τι ισχύει στην περίπτωση που η αντικειμενική μας συνάρτηση δεν είναι το Utility, αλλά το Liquid Welfare. Δηλαδή, να ελέγξουμε κατά πόσο ισχύει (ή θα μπορούσε να ισχύει) η ανισότητα $\bar{v}_i(S_i) \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^i$, που έχουμε θίξει ήδη (και θα αποδείξουμε μαθηματικά σε επόμενη ενότητα) μας οδηγεί στην προσέγγιση του βέλτιστου Liquid Welfare.

Λήμμα 4.2. *Στην περίπτωση που χρησιμοποιούμε το oracle*

$$D_i'(U_i, p) = \max\left\{v_i(x_i) - \sum_{e \in U_i} p_e^i | \sum_{e \in U_i} p_e^i \leq B_i\right\}$$

δεν μπορούμε να παράξουμε liquid welfare maximizing bundle. Με άλλα λόγια αν T_i είναι οποιοδήποτε εφικτό bundle που θα μπορούσε να παράξει ο μηχανισμός και S_i το bundle που πραγματικά παράγει, δεν ισχύει η ανισότητα: $\bar{v}_i(S_i) - \sum_{e \in S_i} p_e^i \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^i$

Απόδειξη. Για να αποδείξουμε το Λήμμα αυτό θα δείξουμε είτε ότι η ανισότητα δεν ικανοποιείται ποτέ, είτε ότι ικανοποιείται κάτω από παράλογες προϋποθέσεις ή ότι ικανοποιείται σε πολύ ειδικές περιπτώσεις. Διακρίνουμε τις εξής περιπτώσεις:

[2] Αντικαθιστώντας την «οποιαδήποτε εφικτή λύση» με την βέλτιστη λύση.

Περίπτωση I. Εάν $\bar{v}_i(S_i) = v_i(S_i)$ και $\bar{v}_i(T_i) = v_i(T_i)$ τότε είναι η ανισότητα δεν μπορεί να ικανοποιηθεί και αντιστοιχεί στην περίπτωση που καλύψαμε στο Λήμμα 4.1.

Περίπτωση II. Εάν $\bar{v}_i(S_i) = B_i$ και $\bar{v}_i(T_i) = B_i$, τότε η ανισότητα ισχύει τετριμμένα, διότι: $B_i \geq B_i - \sum_{e \in T_i} p_e^i$.

Περίπτωση III. Εάν $\bar{v}_i(S_i) = B_i$ και $\bar{v}_i(T_i) = v_i$, τότε η ανισότητα ισχύει και πάλι, διότι $B_i \geq v_i(T_i) - \sum_{e \in T_i} p_e^i$, εφόσον $\bar{v}_i = \min\{v_i, B_i\}$.

Περίπτωση IV. Εάν $\bar{v}_i(S_i) = v_i$ και $\bar{v}_i(T_i) = B_i$, τότε προκειμένου να ισχύει η ανισότητα πρέπει να υποθέσουμε ότι $\sum_{e \in T_i} p_e^i \geq B_i - v_i(S_i)$ κάτι το οποίο είναι μία πολύ ειδική υπόθεση.

Παρατηρούμε, λοιπόν, ότι οι 3 περιπτώσεις για τις ισχύει η ανισότητα είναι ειδικές περιπτώσεις που καταστρατηγούν τη γενικότητα της επιλογής του bundle T_i, γενικότητα που μας επιτρέπει χωρίς βλάβη να υποθέσουμε ότι ένα τέτοιο T_i θα μπορούσε κάλλιστα να είναι και το βέλτιστο. ∎

4.1.2 Το νέο Oracle

Άμεσα προκύπτει, συνεπώς, η ανάγκη να αλλάξουμε δομικά το ίδιο μας το Oracle.

Ορισμός 4.1 (Oracle)**.** Ονομάζουμε *Μαντείο* 1 *(Oracle 1)* το μαντείο το οποίο παίρνει ως είσοδο το τρέχον price vector p^i, καθώς και το budget του παίκτη B_i και επιστρέφει το bundle $S_i \subseteq U_i$ για το οποίο ισχύει:

$$O_1(U_i, p^i, B_i) = \max \left\{ \bar{v}_i(S_i) - \sum_{e \in S_i} p_e^i \right\}$$

Εύκολα μπορούμε να καταλάβουμε ότι αυτό το Oracle εκφράζει το αντίστοιχο του Social Utility για την περίπτωση του Liquid Welfare.

Υποθέτουμε και πάλι ότι στον αλγόριθμο δίνονται δύο παράμετροι, $\mu \geq 1$ και $L \geq 0$, έτσι ώστε το L να είναι το κάτω φράγμα του μέγιστου *liquid valuation*[3] μεταξύ όλων των παικτών και όλων των bundles και ότι υπάρχει το πολύ ένας παίκτης, του οποίου το valuation να ξεπερνά το μL.

Παρατήρηση 4.1. $L \leq \bar{v}(opt)$

Ο νέος αλγόριθμος διαμορφώνεται, λοιπόν, ως εξής:

[3]Θα μπορούσαμε να το προσδιορίσουμε αυτό με sampling, αλλά θα αναφερθούμε εκτενέστερα σε αυτό σε επόμενη ενότητα.

Initialization: Set all items' prices at $p_0 = \frac{L}{4bm}$.

1. For each good $e \in U$ do $p_e^1 = p_0$.
2. For each bidder $i = 1, 2 \ldots, n$ do:
3. Set $S_i = O(U_i, p^i, B_i)$, for a suitable $U_i \subseteq U$.
4. Update for each good $e \in S_i : p_e^{i+1} = p_e^i \cdot 2^{1/b}$.

End: Bidder i pays: $p^i = \sum_{e \in S_i} p_e^i$.

Algorithm 8: Updated Overselling MPU Algorithm

Παρατηρούμε εδώ ότι στο καινούριο μας πλαίσιο, για να μπορέσει κάποιος παίκτης να αγοράσει κάποιο bundle θα πρέπει εκτός από το να έχει valuation μεγαλύτερο από την τιμή του bundle να έχει και το απαραίτητο budget για να το πληρώσει. Από αυτή την ιδιότητα, προκύπτει το ακόλουθο λήμμα σχετικά με το overselling που θα κάνει ο μηχανισμός.

Λήμμα 4.3. *Για οποιαδήποτε επιλογή $U_1, \ldots, U_n$, η ανάθεση S αντιστοιχίζει το πολύ sb αντίγραφα του κάθε αντικειμένου στους παίκτες, όπου $s = \log(4\mu bm) + \frac{2}{b}$, όπου η βάση του λογάριθμου είναι 2.*

Απόδειξη. Έστω ένα οποιοδήποτε αντικείμενο $e \in U$. Έστω ότι μετά από κάποιο βήμα $\lceil sb - 2 \rceil \geq b \log(4\mu bm)$ αντίγραφα του αντικειμένου e έχουν ανατεθεί στους παίκτες. Τότε, η τιμή του e είναι μεγαλύτερη από $p_0 \cdot 2^{\log(4\mu bm)} \geq \mu L$. Μετά από αυτό το βήμα, ο αλγόριθμος μπορεί να δώσει αντίγραφα του αντικειμένου μόνο στους παίκτες, των οποίων το μέγιστο valuation ξεπερνά το μL **και που τους το επιτρέπει το budget τους**. Ωστόσο, ακόμη και αν το budget όλων των εναπομείναντων παικτών είναι αρκετό, υπάρχει μόνο ένας το πολύ παίκτης (από την υπόθεση) ο οποίος μπορεί να το αγοράσει. Άρα, πωλώνται το πολύ $\lceil sb - 1 \rceil \leq sb$ αντίγραφα του e, το οποίο αποδεικνύει το λήμμα. ∎

Είναι εύκολο να δει κανείς ότι το παρακάτω λήμμα ισχύει για το νέο μας oracle, αφού στηρίζεται εγγενώς στην λογική του μηχανισμού και όχι στις λεπτομέρειες των μαντείων. Το ίδιο ισχύει και για το ακόλουθο λήμμα, το οποίο δίνει ένα κάτω φράγμα για τα (πιθανώς) περιορισμένα oracles.

Συμβολισμός. Έστω l_e^i ο αριθμός των αντιγράφων του αντικειμένου e τα οποία έχουν ανατεθεί σε όλους τους bidders που προηγούνται του i και $l_e^* = l_e^{n+1}$ συμβολίζει τον συνολικό αριθμό αντιγράφων του αντικειμένου e που έχουν ανατεθεί. Έστω, επίσης, $p_e^* = p_0 \cdot r^{l_e^*}$ η τιμή του αντικειμένου e στο τέλος του αλγορίθμου.

Λήμμα 4.4. *Για οποιαδήποτε επιλογή από $U_1, \ldots, U_n \subseteq U$, $\bar{v}(S) \geq b \sum_{e \in U} p_e^* - bmp_0$.*

Απόδειξη. Έστω $r = 2^{1/b}$. Οι παίκτες μας είναι individually rational και συνεπώς $\bar{v}_i(S_i) \geq \sum_{e \in S_i} p_e^i$. Άρα,

$$\bar{v}(S) \geq \sum_{i=1}^{n} \sum_{e \in S_i} p_e^i = \sum_{i=1}^{n} \sum_{e \in S_i} p_0 r^{l_e^i} = p_0 \sum_{e \in U} \sum_{k=0}^{l_e^*-1} = p_0 \sum_{e \in U} \frac{r^{l_e^*-1}}{r-1} \tag{4.12}$$

Τώρα, εφαρμόζοντας τα $p_e^* = p_0 r^{l_e^*}$ και $\frac{1}{r-1} = \frac{1}{2^{1/b}-1} \geq b$ στην 4.12 μας δίνει ότι:

$$\bar{v}(S) \geq b \sum_{e \in U} p_e^* - bmp_0$$

■

Στον ανανεωμένο μας μηχανισμό, θα χρειαστεί να κάνουμε μία ακόμη υπόθεση. Θα υποθέσουμε, εκτός του ότι $p_0 = \frac{L}{4bm}$, ότι ισχύει ότι

$$p_0 \geq \frac{B_i - v_i(S_i)}{m} \tag{4.13}$$

για οποιοδήποτε bundle S_i. Δηλαδή, θα διαλέξουμε κατάλληλα το p_0, έτσι ώστε να ισχύει η 4.13. Σημειώνουμε εδώ, αν και θα αναφερθούμε εκτενέστερα σε αυτό παρακάτω, ότι ένα τέτοιο p_0 μπορεί να επιλεγεί κατάλληλα μέσω sampling.

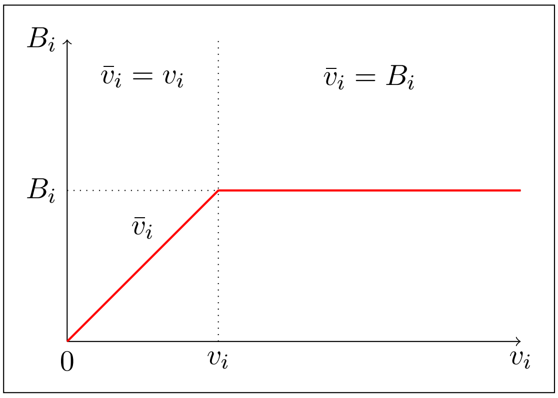
Η υπόθεση αυτή δεν είναι τόσο ανεξήγητη. Βλέπουμε ότι για να ισχύει, θα πρέπει:

$$p_0 \geq \frac{B_i - v_i(S_i)}{m} \Rightarrow \frac{L}{4bm} \geq \frac{B_i - v_i(S_i)}{m} \Rightarrow \tag{4.14}$$

$$\frac{L}{4b} \geq B_i - v_i(S_i) \tag{4.15}$$

Δηλαδή, αυτό που μας λέει η ανισότητα 4.15 είναι ότι πρέπει το κάτω φράγμα του maximum valuation (L) ανά το πλήθος των αντιγράφων των αντικειμένων (b) πρέπει να μας προσφέρει μεγαλύτερη ικανοποίηση από το budget που μας απομένει, αφού διαλέξουμε το κατάλληλο bundle S_i.

Επιπλέον, υπενθυμίζουμε εδώ ότι ενδιαφερόμαστε για submodular valuations. Επομένως, το ακόλουθο διάγραμμα μας δίνει το Liquid Valuation ($\bar{v}_i(S_i) = \min\{v_i(S_i), B_i\}$) ενός παίκτη i.



Στις ενότητες που ακολουθούν, αποδεικνύουμε για το κάθενα ένα από τα τρία Oracles, των οποίων δώσαμε τον ορισμό παραπάνω, το κομβικό Λήμμα, που οδηγεί και στην απόδειξη της προσέγγισης του Liquid Welfare που πετυχαίνουν.

Λήμμα 4.5. *Έστω μία οποιαδήποτε εφικτή ανάθεση από bundles αντικειμένων $T = (T_1, \ldots, T_n)$ που αναθέτει το πολύ b αντίγραφα του κάθε αντικειμένου. Ο αλγόριθμος χρησιμοποιεί μή-*

περιορισμένα oracles και ισχύει ότι:

$$\bar{v}_i(S_i) - \sum_{e \in S_i} p_e^i \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^i$$

ή αλλιώς

$$\bar{v}_i(S_i) \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^i$$

Απόδειξη. Το λήμμα προκύπτει άμεσα από τον Ορισμό του Μαντείου και του individual rationality των παικτών. ■

Έχοντας αποδείξει το λήμμα 4.5 προκύπτει το ακόλουθο:

Λήμμα 4.6. *Για* $U_1 = \cdots = U_n$, $\bar{v}(S) \geq \bar{v}(opt) - b\sum_{e \in U} p_e^*$.

Απόδειξη. Αποδείξαμε ότι και για το νέο μαντείο ισχύει η ανισότητα: $\bar{v}_i(S_i) \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^i$. Εφόσον, $p_e^* \geq p_e^i$ για κάθε αντικείμενο e και για κάθε bidder i, προκύπτει ότι: $\bar{v}_i(S_i) \geq \bar{v}_i(T_i) - \sum_{e \in T_i} p_e^*$. Αθροίζοντας πάνω σε όλους τους bidders έχουμε ότι:

$$\bar{v}(S) = \sum_{i=1}^{n} \bar{v}_i(S_i) \geq \sum_{i=1}^{n} \bar{v}_i(T_i) - \sum_{i=1}^{n} \sum_{e \in T_i} p_e^* \geq \bar{v}(T) - b\sum_{e \in U} p_e^* \tag{4.16}$$

Όπου η τελευταία εξίσωση κάνει χρήση του γεγονότος ότι αν το σύνολο T είναι εφικτό, θα πρέπει κάθε αντικείμενο να πωλείται σε b αντίτυπα. Το αποτέλεσμα προκύπτει αν θεωρήσουμε ότι το bundle T_i ανατίθεται στον παίκτη i στη βέλτιστη λύση *opt* και στη συνέχεια αθροίζοντας την ανισότητα και πάλι για όλους τους bidders. ■

Θεώρημα 4.1. *Ο αλγόριθμος 8 με* $p_0 = \frac{L}{4bm}$ *παράγει μία ανέφικτη ανάθεση* S *κατά την οποία το πολύ* $b\log(4\mu bm) + 2$ *αντίγραφα του κάθε αντικειμένου ανατίθενται στους bidders. Εάν* $U_1 = \cdots = U_n = U$, *τότε:* $\bar{v}(S) \geq \frac{1}{4}\bar{v}(opt)$.

Απόδειξη. Το φράγμα στο πλήθος των αντιγράφων που θα πωληθούν από τον Αλγόριθμο, έχει αποδειχθεί στο Λήμμα 4.3. Από το Λήμμα 4.4 προκύπτει ότι: $b\sum_{e \in U} \leq \bar{v}(S) + bmp_0$. Αντικαθιστώντας αυτό το άνω φράγμα στο άθροισμα των τελικών τιμών του Λήμματος 4.6 μας δίνει:

$$\bar{v}(S) \geq \bar{v}(opt) - \bar{v}(S) - bmp_0 \Rightarrow 2\bar{v}(S) \geq \bar{v}(opt) - bmp_0$$

Επιπλέον, εφόσον $\bar{v}(opt) \geq L$ προκύπτει ότι:

$$p_0 = \frac{L}{4bm} \leq \frac{\bar{v}(opt)}{4bm} \Rightarrow bmp_0 \leq \frac{1}{4}\bar{v}(opt) \Rightarrow \bar{v}(S) \geq \frac{1}{4}\bar{v}(opt)$$

■

Για την πληρότητα της εργασίας, παρουσιάζουμε παρακάτω το πώς γίνεται η τυχαιοποίηση του μηχανισμού, ώστε να μπορέσουμε να μην έχουμε υπερ-πώληση των αντικειμένων. Η ουσία τόσο της απόδειξης όσο και του αλγορίθμου, παραμένει ίδια με την αντίστοιχη στο ;;.

Initialization: Set all items' prices at $p_0 = \frac{L}{4bm}$.

1. For each good $e \in U$ do $p_e^1 = p_0$, $b_e^1 = b$.
2. For each bidder $i = 1, 2 \ldots, n$ do:
3. Set $S_i = O_i(U_i, p^i)$, for $U_i = \left\{e \in U | b_e^i > 0\right\}$
4. Update for each good $e \in S_i : p_e^{i+1} = p_e^i \cdot 2^{1/b}$.
5. With probability q set $R_i = S_i$, else $R_i = \emptyset$.
6. Update for each good $e \in R_i : b_e^{i+1} = b_e^i - 1$.

End: Bidder i pays: $p^i = \sum_{e \in R_i} p_e^i$.

Algorithm 9: Updated MPU Algorithm with Oblivious Randomized Rounding

Λήμμα 4.7. *Έστω ότι η πιθανότητα $q > 0$ στον αλγόριθμο 9 έχει επιλεγεί επαρκώς μικρή, έτσι ώστε για οποιοδήποτε $1 \leq i \leq n$ και για οποιοδήποτε bundle $T \subseteq U$:*

$$\mathbb{E}[\bar{v}_i(T \cup U_i)] \geq \frac{1}{2}\bar{v}_i(T)$$

Τότε, $\mathbb{E}(\bar{v}(S)) \geq \frac{1}{8}\bar{v}(opt)$ και $\mathbb{E}[\bar{v}(R)] \geq \frac{q}{8}\bar{v}(opt)$.

4.1.3 Το πρόβλημα του μηχανισμού

Ως εδώ, φαίνεται ότι ο μηχανισμός έχει ένα πολύ καλό approximation ratio. Ωστόσο, έχουμε αποκρύψει κάτι: σε κανένα σημείο της ανάλυσή μας δεν έχουμε ασχοληθεί με το κατά πόσο ο μηχανισμός μας είναι truthful. Η αλήθεια είναι ότι παρά το γεγονός ότι το μοντέλο παραμένει online, εξαιτίας του νέου Oracle ο μηχανισμός δεν είναι πλέον incentive compatible.

Κρίνουμε εδώ σκόπιμο να επαναλάβουμε τον αυστηρό ορισμό του incentive compatibility όπως διατυπώνεται στο [1].

Ορισμός 4.2 (Incentive Compatible)**.** Ένας μηχανισμός $(f, p_1, \ldots, p_n)$ ονομάζεται incentive compatible αν για κάθε παίκτη i $v_1 \in V_1, \ldots, v_n \in V_n$ και για κάθε $v_i' \in V_i$, εάν συμβολίσουμε με $\alpha = f(v_i, v_{-i})$ και $\alpha' = f(v_i', v_{-i})$, τότε:

$$v_i(\alpha) - p_i(v_i, v_{-i}) \geq v_i(\alpha') - p_i(v_i, v_{-i})$$

Πρακτικά, δηλαδή, για να έχουμε incentive compatibility πρέπει και αρκεί σε κάθε βήμα ο κάθε παίκτης να διαλέγει κάποιο utility maximizing bundle. Ο δικός μας αλγόριθμος, ωστόσο, διαλέγει με στόχο τη μεγιστοποίηση του Liquid Welfare των παικτών και όπως έχουμε δει ήδη από το Παράδειγμα 3.2.1 αυτά τα δύο είναι ασυμβίβαστα.

Παρατήρηση 4.2. *Ο μόνος τρόπος να κάνουμε τον μηχανισμό μας incentive compatible θα ήταν να χρησιμοποιήσουμε τις τιμές που θα παρήγαγε το Demand Oracle.*

Το πρόβλημα που εγείρει η παραπάνω παρατήρηση του μηχανισμού είναι προφανές και άμεσο: όπως έχουμε δείξει αναλυτικά, αν δοκιμάσουμε να χρησιμοποιήσουμε το Demand Oracle, λόγω των περιορισμών στα budgets δεν μπορούμε να προσεγγίσουμε το Liquid Welfare παρά μόνο σε λίγες και πολύ συγκεκριμένες περιπτώσεις. Τέλος:

Παρατήρηση 4.3. *Λόγω της γενικής φύσης των budgets τα αποτελέσματα που προκύπτουν από τα Liquid Welfare maximizing Oracles είναι ασυσχέτιστα με τα αποτελέσματα από τα Demand Oracles.*

4.2 Ο μηχανισμός Top_k

Στη συνέχεια, θα περιγράψουμε έναν μηχανισμό (καθώς και το πλαίσιο που τον συνοδεύει) ο οποίος αγγίζει το κάτω φράγμα του $o(\log m)$ για το Social Utility. Ο μηχανισμός αυτός προτάθηκε από τους Fotakis, Tsipras, Tzamos, Zampetakis σε μία πρόσφατη δουλειά ([50]). Οι λόγοι που επιλέγουμε να ασχοληθούμε μαζί του, θα εξηγηθούν εκτενέστερα παρακάτω.

Το πλαίσιο. Για οποιοδήποτε ακέραιο m, $[m] \equiv \{1, 2, \dots, m\}$. Ως j- οστή συντενταγμένη ενός διανύσματος x συμβολίζουμε το x_j. Για ένα διάνυσμα $x = (x_1, \dots, x_m)$ και $i \in [m]$, x_{-i} είναι το διάνυσμα x, χωρίς τη συντεταγμένη i. Για ένα διάνυσμα $x \in \mathbb{R}^m$ και κάποιο $l \geq 0$, $x^l = (x_1^l, \dots, x_m^l)$ είναι η δύναμη του x σε συντεταγμένες $||x||_l = \left(\sum_{j=1}^m x_j^l\right)^{1/l}$ είναι η l-νόρμα του x. Για ευκολία, θα συμβολίζουμε $||q||_1 = |q|$. Επιπλέον, $||x||_\infty \max_{j\in[m]}\{x_j\}$. Έστω ότι m είναι το σύνολο των δυνατών outcomes που έχουμε για το πρόβλημά μας. Θεωρούμε ένα σύνολο από n στρατηγικούς παίκτες, ο καθένας εκ των οποίων έχει μία κρυφή και μή μηδενική αξία για κάθε outcome. Για τον παίκτη i, ορίζουμε το valuation του σαν ένα διάνυσμα $x_i \in \mathbb{R}_+^m$, το οποίο υποδηλώνει ότι ο παίκτης i έχει valuation x_{ij} για το outcome j. Ονομάζουμε το διάνυμα από όλα τα valuations των παικτών, $x = (x_1, \dots, x_n)$ ένα valuation προφίλ. Για το valuation profile x, $w(x) = x_1 + x_2 + \dots, x_n$ είναι το διάνυμα των βαρών των outcomes. Θα γράφουμε w αντί για $w(x)$ και w_{-i} αντί για $w(x_{-i})$.

Από τη δουλειά των Hartline και Roughgarden ([]) προκύπτει το ακόλουθο πόρισμα.

Πόρισμα 4.1. *Κανένας truthful μηχανισμός δεν μπορεί να προσεγγίσει το Social Utility με παράγοντα $o(\log m)$.*

Για να αποδειχθεί αυτό το Πόρισμα χρησιμοποιούμε ουσιαστικά Bayesian Analysis και τον ορισμό των VCG payments. Στη συνέχεια, θα περιγράψουμε τον μηχανισμό, ο οποίος αντιστοιχεί σε αυτό το κάτω φράγμα.

Ορισμός 4.3. Για κάποιο $k \in [m]$, ο κανόνας ανάθεσης Top_k με είσοδο x, ταξινομεί τα outcomes σε φθίνουσα σειρά με βάση το βάρος τους, δηλαδή, $w_1 \geq w_2 \geq \cdots w_m$ (σπάζοντας τις ισότητες αυθαίρετα) και αναθέτει πιθανότητα $\frac{1}{k}$ στα πρώτα k. Τυπικά, $Top_k(x) = \arg\max_{s\in S_k} s \cdot w$, όπου S_k είναι το σύνολο των διανυσμάτων στο $\mathbb{R}_+^m$ με ακριβώς k συντεταγμένες ίσες με $1/k$ και ακριβώς $m - k$ ίσες με 0.

Εφόσον οι Top_k είναι αυτοί που μεγιστοποιούν το Welfare θα μπορούσαν να μετατραπούν σε truthful και individually rational μηχανισμούς μέσω του VCG σχήματος πληρωμών. Συμβολίζουμε μηχανισμούς αυτής της οικογένειας ως $\mathcal{M}_k = (Top_k, p_k)$. Καθένας από αυτούς τους μηχανισμούς μπορεί να πετύχει διαφορετικά approximation guarantees σχετικά με το social welfare και το social utility σε διαφορετικά πλαίσια, ανάλογα με το k. Επομένως, κάνοντας χρήση της τυχαιοποίησης πάνω σε αυτά, μπορούμε να παρέχουμε worst-case guarantees. Στον μηχανισμό που ακολουθεί πετυχαίνουμε ένα βέλτιστο social utility approximation guarantee τυχαιοποιώντας πάνω σε εκθετικά αυξανόμενες τιμές του k. Για απλότητα, υποθέτουμε ότι το m είναι δύναμη του 2.

1. Choose j uniformly at random from $0, 1, 2, \ldots, \log m$.
2. Let $k \leftarrow 2^j$.
3. Output the probability distribution $Top_k(x)$ over outcomes.
4. Charge agent i the amount $w_{-i}Top_k(x_{-i}) - w_{-i}Top_k(x)$.

Algorithm 10: A $\log m$ approximate mechanism for Social Utility

Διαισθητικά, λοιπόν, ο Top_k μηχανισμός, δίνει πιθανότητα εμφάνισης-εξέτασης μόνο στα πρώτα k outcomes και χρεώνει στον κάθε παίκτη το externality του στους άλλους παίκτες. Παρατηρούμε ότι ο τρόπος με τον οποίον αυτός ο μηχανισμός βρίσκει το βέλτιστο outcome (για την ακρίβεια, μία πολύ καλή προσέγγιση αυτού) θα μπορούσε να χρησιμοποιηθεί στο πρόβλημά μας με το διαιρέσιμο αγαθό και τους παίκτες με budget περιορισμούς.

Η αναγωγή

Για να κάνουμε αυτή την αναγωγή καταρχάς υποθέτουμε ότι ο χώρος των outcomes είναι λογαριθμικός. Επιπλέον, υποθέτουμε ότι δεν υπάρχει communication complexity στον τρόπο με τον οποίο θα μάθουμε το κάθε outcome. Σε αυτή την περίπτωση, τα outcomes αντιστοιχούν σε μοιράσματα των διαθέσιμων αντικειμένων και δεν αντιπροσωπεύουν μόνο έναν παίκτη, αλλά πολλούς. Στην πραγματικότητα, το κάθε outcome μπορεί να αντιστοιχεί σε μη-μηδενικό allocation για όλους τους παίκτες. Η ιδέα είναι να ορίσουμε μία γενική πληρωμή για κάθε outcome, η οποία να αυξάνεται πολλαπλασιαστικά. Αυτή η γενική πληρωμή είναι το άθροισμα όλων των payments που ο σχεδιαστής της δημοπρασίας θα περίμενε να συγκεντρώσει πωλώντας τα αντικείμενα σε όλους τους παίκτες.

Η κατεύθυνση στην οποία εργαζόμαστε σε αυτό το πρόβλημα είναι κατά πόσο μπορούμε, ακόμη και αν εξασφαλίσουμε το budget feasibility για τους παίκτες μας να τους δώσουμε κίνητρο να παίξουν σε μία τέτοια δημοπρασία. Για να γίνει κάτι τέτοιο, πρέπει να προσδιορίσουμε με αυστηρό μαθηματικό τρόπο πώς θα γίνει ο διαμοιρασμός των πληρωμών. Με άλλα λόγια, αν αποφασίσουμε μέσω του Top_k ότι το O_i είναι το βέλτιστο και αυτό που σέβεται τα budgets, τότε αυτόματα σημαίνει ότι όλοι οι παίκτες θα μπορούν να πληρώσουν τη γενική πληρωμή αυτού του outcome. Αλλά πώς μοιράζεται αυτή η πληρωμή στον κάθε παίκτη και τι ποσό θα πρέπει να πληρώσει ο καθένας προκειμένου ο μηχανισμός να είναι πραγματικά incentive compatible; Αυτό είναι το ερώτημα που θέλουμε να ερευνήσουμε σε επόμενο στάδιο.

4.3 Συμπεράσματα και Μελλοντική Δουλειά

Η τεχνική των Multiplicative Updates σε Online μηχανισμούς μπορεί να δώσει πολύ καλά approximation guarantees για Combinatorial Auctions με budgets. Το πρόβλημα που έχει αυτή η τεχνική είναι ότι ως τώρα φαίνεται να μην είναι incentive compatible. Μία από τις μελλοντικές κατευθύνσεις είναι να διερευνηθεί εκτενέστερα το incentive compatibility αυτών των δημοπρασιών. Για το σκοπό αυτό, θα μπορούσαμε να ελέγξουμε τι γίνεται στις περιπτώσεις που τα budgets των παικτών δεν είναι γενικής φύσης, αλλά υπακούουν σε συγκεκριμένους περιορισμούς, για παράδειγμα είναι uniform ή προκύπτουν ως συγκεκριμένη κατηγορία ασκήσεων. Εγείρεται φυσικά το ερώτημα αν θα μπορούσαμε με uniform budgets να βρούμε μία σχέση μεταξύ των Demand Oracles και μίας άλλη οικογένειας Oracles. Μία ακόμη κατεύθυνση που πρέπει να διερευνηθεί περαιτέρω είναι τι θα μπορούσαμε να κάνουμε, αν ο σκόπελος του incentive compatibility δεν μπορεί να ξεπεραστεί. Με άλλα λόγια, με δεδομένο ότι ο μηχανισμός μας δεν είναι φιλαλήθης, μπορούμε να τον επεκτείνουμε ώστε η μη-φιλαλήθεια να μην παίζει κανένα σημαντικό ρόλο;[4] Αξίζει, επιπλέον, να μελετήσουμε τα Combinatorial Auctions με budgets ως προβλήματα γραμμικού προγραμματισμού με χρήση της δουλειάς των Plotkin, Shmoys, Tardos.

[4]Έχει νόημα να αναρωτηθούμε προς αυτή την κατεύθυνση, καθώς οι σημαντικότεροι σύγχρονοι αλγόριθμοι, λόγω του τεράστιου όγκου πληροφοριών που χειρίζονται, δεν είναι φιλαλήθεις και όμως, χρησιμοποιούνται ευρέως.

Βιβλιογραφία

[1] Nisan/Roughgarden/Tardos/Vazirani (eds), Algorithmic Game Theory, Cambridge University, 2007.

[2] A. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica: journal of the Econometric Society*, pages 587–601, 1973.

[3] M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of economic theory*, 10(2) : 187–217, 1975.

[4] Roger B Myerson. Optimal Auction Design. *Mathematcs of operations research*, 6(1) : $58 - 73$, 1981

[5] E. Clarke. Multipart pricing of public goods. *Public Choice*, 8 : 19–33, 1971.

[6] T. Groves. Incentives in teams. *Econometrica*, 41 : 617–631, 1973.

[7] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16 : 8–27, 1961.

[8] E. B Dynkin. The optimum choice of the instant for stopping a markov process. Sov. Math. Dokl. 4 (1963).

[9] R. D Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In SODA, pp. $630 - 631$ (2005).

[10] M. Babaioff, N. Immorlica, R. Kleinberg. Matroids, secretary problems and online mechanisms. In FOCS, pp. $434 - 443$ (2007).

[11] A. Borodin, R. El-Yaniv. Online computation and competitive analysis. Cambridge University Press (1998).

[12] T. Roughgarden. Algorithmic Game Theory (CS364A), Lecture Notes, 2013

[13] Shengwu Li. Obviously Strategy-Proof Mechanisms, 2016.

[14] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta algorithm and applications. Technical report, Princeton University, 2005.

[15] N. Littlestone and M. K. Warmuth. The Weighted Majority Algorithm. In $FOCS$, pages 256–261. $IEEE$ Computer Society, 1989.

[16] Y. Freund and R. E. Schapire. A decision-theoretic generalisation of on-line learning and application to boosting. *Journal of computer and system science*, 55 : 119–139, 1997.

[17] P. Krysta and B. Vocking. Online Mechanism Design (Randomized Rounding on the Fly). In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *ICALP*(2), volume 7392 of *LectureNotesinComputerScience*, pages 636–647. Springer, 2012.

[18] B. Awerbuch, Y. Azar, S. A Plotkin. Throughput-competitive on-line routing. In *FOCS*, pp. 32 – 40 (1993)

[19] Buchbinder, N., Naor, J.: Online Primal-Dual Algorithms for Covering and Packing Problems. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 689–701. Springer, Heidelberg (2005)

[20] Buchbinder, N., Naor, J.: Improved bounds for online routing and packing via a primal-dual approach. In: *FOCS*, pp. 293–304 (2006)

[21] Feige, U.: On maximizing welfare when utility functions are subadditive. SIAM J. Comput. 39(1), 122–142 (2009)

[22] Bartal, Y., Gonen, R., Nisan, N.: Incentive compatible multi unit combinatorial auctions. In The Proc. of the 9th TARK, pp. 72–87 (2003)

[23] S. A. Plotkin, D. B. Shmoys, and E. Tardos. Fast Approximation Algorithms for Fractional Packing and Covering Problems. In *FOCS*, pages 495–504. *IEEE* Computer Society, 1991.

[24] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5 : 197 – 227, 1990.

[25] Yoav Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1) : 119 – 139, August 1997.

[26] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science* (FOCS-98), pages 300–309, Los Alamitos, CA, November8–11 1998. *IEEE* Computer Society.

[27] P. Raghavan and C. D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica, 7(4) : 365–374, 1987.

[28] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. In *27th Symposium on Foundations of Computer Sciences*, pages 10–18. *IEEE*, 1986.

[29] Neal E. Young. Randomized rounding without solving the linear program. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 170–178, San Francisco, California, 22–24 January 1995.

[30] Philip Klein and Hsueh-I. Lu. Efficient approximation algorithms for semidefinite programs arising from MAX CUT and COLORING. In *Proceedings of the twenty-eighth annual ACM Symposium on the Theory of Computing*, pages 338–347, 1996.

[31] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM, 42(6) : 1115–1145, 1995.

[32] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

[33] Elad Hazan. New Techniques in Online Convex Optimization and their Applications. PhD thesis, Princeton University, 2006.

[34] Y.-K. Che and I. Gale. Standard auctions with financially constrained bidders. *Review of Economic Studies*, 65(1) : 1–21, January 1998.

[35] J.-P. Benoit and V. Krishna. Multiple-object auctions with budget constrained bidders. *Review of Economic Studies*, 68(1) : 155–79, January 2001.

[36] J.-J. Laffont and J. Robert. Optimal auction with financially constrained buyers. *Economics Letters*, 52(2) : 181–186, August 1996.

[37] A. Malakhov and R. V. Vohra. Optimal auctions for asymmetrically budget constrained bidders. Discussion Papers 1419, Northwestern University, Center for Mathematical Studies in Economics and Management Science, Dec. 2005.

[38] M. M. Pai and R. Vohra. Optimal auctions with financially constrained bidders. Discussion papers, Northwestern University, Center for Mathematical Studies in Economics and Management Science, Aug 2008.

[39] C. Borgs, J. T. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *ACM Conference on Electronic Commerce*, pages 44–51, 2005.

[40] S. Chawla, D. L. Malec, and A. Malekian. Bayesian mechanism design for budget-constrained agents. In *ACM Conference on Electronic Commerce*, pages 253–262, 2011.

[41] S. Dobzinski, R. Lavi, and N. Nisan. Multi-unit auctions with budget limits. *Games and Economic Behavior*, 74(2) : 486–503, 2012.

[42] S. Bhattacharya, V. Conitzer, K. Munagala, and L. Xia. Incentive compatible budget elicitation in multi-unit auctions. In *SODA*, pages 554–572, 2010.

[43] A. Fiat, S. Leonardi, J. Saia, and P. Sankowski. Single valued combinatorial auctions with budgets. In *ACM Conference on Electronic Commerce*, pages 223–232, 2011.

[44] R. Colini-Baldeschi, M. Henzinger, S. Leonardi, and M. Starnberger. On multiple keyword sponsored search auctions with budgets. In *ICALP (2)*, pages 1–12, 2012.

[45] G. Goel, V. S. Mirrokni, and R. Paes Leme. Polyhedral clinching auctions and the adwords polytope. In *STOC*, pages 107–122, 2012.

[46] G. Goel, V. S. Mirrokni, and R. Paes Leme. Clinching auctions with online supply. In *SODA*, 2013.

[47] S. Dobzinski and R. P. Leme. Efficiency Guarantees in Auctions with Budgets. *CoRR*, abs/1304.7048, 2013.

[48] L. M. Ausubel. An efficient ascending-bid auction for multiple objects. *American Economic Review*, pages 1452–1475, 2004.

[49] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In $TARK$, pages 72–87, 2003.

[50] D. Fotakis, D. Tsipras, C. Tzamos, E. Zampetakis. Efficient Money Burning in General Domains. In $SAGT$, pages $85 - -97$, 2015.

[51] J.D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *Proc. of the 40th ACM Symposium on Theory of Computing(STOC '08)*, pages 75–84, 2008.