ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Μηχανικής Μάθησης για την Εκτίμηση της Ωφέλειας των Παικτών σε Ηλεκτρονικές Δημοπρασίες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ανδρέα Ματζόρι

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

Αθήνα, Οκτώβριος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Μηχανικής Μάθησης για την Εκτίμηση της Ωφέλειας των Παικτών σε Ηλεκτρονικές Δημοπρασίες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ανδρέα Ματζόρι

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 25 Οκτωβρίου 2017.

.....	.....	.....
Δημήτρης Φωτάκης	Νικόλαος Παπασπύρου	Αριστείδης Παγουρτζής
Επίκουρος Καθηγητής	Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής
Ε.Μ.Π.	Ε.Μ.Π.	Ε.Μ.Π .

Αθήνα, Οκτώβριος 2017

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

Ευχαριστίες

Μιας και αυτή η διπλωματική σηματοδοτεί το τέλος των προπτυχιακών μου σπουδών θα ήταν άδικο να μην ευχαριστήσω θερμά όσους συντέλεσαν και βοήθησαν στην ολοκλήρωση τους. Θα ήθελα να ευχαριστήσω θερμά τον κ.Φωτάκη, ο οποίος εκτός από την συνεχή βοήθεια που μου έδωσε στην εκπόνηση της παρούσας διπλωματικής, αποτέλεσε και αποτελεί πρότυπο επιστήμονα για μένα καθώς χάρη σε αυτόν αγάπησα τη Θεωρητική Πληροφορική.

Επίσης οφείλω να ευχαριστήσω την οικογένεια μου και την Αλεξάνδρα που και με ανέχονται και με αντέχουν όλα αυτά τα χρόνια, τους θαυμάζω ιδιαίτερα γι΄ αυτό καθώς αντιλαμβάνομαι το μέγεθος του άθλου τους. Τέλος δεν μπορώ να μην αναφερθώ και να ευχαριστήσω τους φίλους μου που πάντα είναι δίπλα μου οποτεδήποτε και αν τους χρειαστώ.

Στη Δανάη και στον Ηλία

Περίληψη

Οι ηλεκτρονικές δημοπρασίες διαφημίσεων είναι ηλεκτρονικές δημοπρασίες που συνήθως επικεντρώνονται στην πώληση διαφημιστικού χώρου πλάι στα αποτελέσματα κάποιας αναζήτησης. Η ραγδιαία αύξηση της χρήσης τους τα τελευταία χρόνια οδήγησε την επιστημονική κοινότητα στο να αρχίσει να μελετά τις ιδιότητές τους. Ένα από τα βασικά προβλήματα σε τέτοιου είδους συνεχώς επαναλαμβανόμενες δημοπρασίες είναι η εκτίμηση της ωφέλειας που έχει ένας παίκτης για το αντικείμενο στο οποίο ποντάρει παρατηρώντας μόνο την εξέλιξη των ποντάρισμάτων του. Κάποιες κλασσικές μέθοδοι αντιμετωπίζουν το πρόβλημα αυτό θεωρώντας ότι οι παίκτες έχουν καταλήξει σε κάποιου είδους ισσοροπία, βασιζόμενοι σε αυτή την υπόθεση καταλήγουν στο να εκτιμούν την αξία που δίνει ο παίκτης στη διαφήμιση ως την αξία που καλύτερα ερμηνεύει την υποτιθέμενη ισσοροπία. Δυστυχώς όμως σε γρήγορα μεταβαλλόμενα συστήματα αυτή η υπόθεση δεν είναι πάντοτε εύλογη.

Σε αυτή την διπλωματική αξιολογούμε πειραματικά μία καινούργια μέθοδο εκτίμησης , η οποία βασίζεται στην υπόθεση ότι οι παίκτες είναι ευφυείς πράκτορες. Επίσης κατασκευάσαμε ένα σύστημα προσομοίωσης του οποίου ο στόχος είναι να αποτελέσει τη βάση για την περαιτέρω μελέτη συστημάτων δημοπρασιών και συγχρόνως να καλύψει την έλλειψη δημοσιευμένων και δωρεάν δεδομένων

Στην δεύτερη παράγραφο εισάγουμε τον αναγνώστη σε κάποιες βασικές αρχές της σχεδίασης μηχανισμών. Στην τρίτη παράγραφο αναλύουμε τις ιδιότητες της πιο ευρέως διαδεδομένης δημοπρασίας στις ηλεκτρονικές δημοπρασίες διαφημίσεων, τη Γενικευμένη Δεύτερης Τιμής δημοπρασία. Στην τέταρτη παράγραφο παρουσιάζουμε κάποιες βασικές αρχές σχετικά με ευφυείς αλγορίθμους οι οποίοι θα αποτελέσουν τη βάση πάνω στην οπόια θα μελετήσουμε τη συμπεριφορά των παικτών στις ηλεκτρονικές δημοπρασίες. Τέλος στην πέμπτη παράγραφο παρουσιάζουμε και αναλύουμε την πρόταση των Nekipelov, Syrgkanis , Tardos σχετικά με την εκτίμηση της ωφέλειας των παικτών, παρουσιάζουμε το σύστημα προσομοίωσης και μια νέα ανάλυση σχετικά με την ευρωστία της προτεινόμενης μέθοδου εκτίμησης.

Λέξεις-Κλειδιά: αλγοριθμική θεωρία παιγνίων, σχεδίαση μηχανισμών, ηλεκτρονικές δημοπρασίες, ευφυείς πράκτορες, μηχανιστική μάθηση, εκτίμηση ωφέλειας

Abstract

Online ad auctions are internet auctions which usually sell advertising space placed nearby the results of a search query. The rapidly increasing market of these auctions led the scientific community to study their properties. A major problem in this type of auctions is to inference the valuation of a bidder, that is the particular value that each bidder has per amount of space he gets. Classical work to face the problem of valuation inferencing rely on the assumption that the players somehow reached an mixed nash equilibrium. However in dynamic settings this is not always the case.

In this thesis we experimentally analyse the new valuation inference method proposed by Syrgkanis, Nekipelov and Tardos which assumes that the players are learning agents. Furthermore we constructed an implementable simulation system whose purpose is to provide a basis for the further study of these types of auctions because of the public data's lack.

In the second chapter we introduce the reader to some basic mechanism design concepts. In the third chapter we analyse the properties of the particular auction format usually used in online ad auctions, the Generalised Second Price auction. In the fourth chapter we introduce basic concepts and algorithms used by players in online decision settings which will constitute our basic assumption about how a bidder should behave in an online ad auctions context. In the fifth chapter we describe the new valuation inference method proposed by Nekipelov, Syrgkanis and Tardos , we describe the simulation system constructed and we present our experimental results with new robustness evidences of the evaluation method.

Keywords: algorithmic game theory, mechanism design, sponsored search auctions, GSP, online convex optimization, bandit convex optimization, online learning, valuation inference

Contents

Chapter 1

Introduction

The impact of the Internet on economic growth is undeniable. Specially marketing is rapidly evolving from traditional advertising methods such as newspaper or television advertisements to digital online advertisements. A particular domain of digital marketing called sponsored search advertising are the advertisements that appear side to the results of a search query in the widely known search engines Yahoo and Google. The important percentage of the market that these type of advertisements maintain made the necessitude of their study clear. What is the best way to sell this advertisements? How can search engines improve their income? How companies should behave to buy these advertising places? These are apparently innocuous but in reality deep questions that need to be answered.

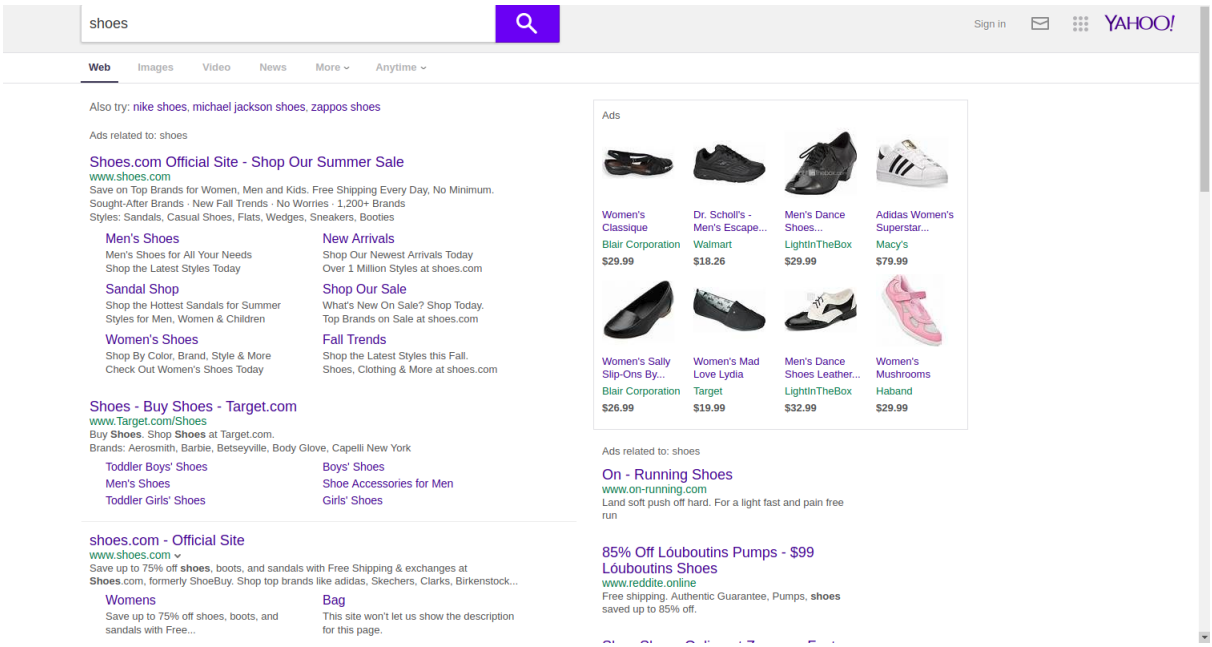


Figure 1.1: Sponsored search advertising

These advertisements are sold via known online auction formats trying to face all the problems mentioned above in the best possible way. The most widely used auction format is the Generalised Second Price(GSP) auction.

In the GSP auction the auctioneer sells advertisement spots that have different click probability coefficients, let $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$ to be the vector of these coefficients. These coefficients are positive and non increasing, this means that the first position has a higher probability of getting clicked than the second, the second than the third etc. Moreover each bidder is associated with a click probability coefficient which reflects the proclivity of a person to click the particular advertisement associated with this bidder, let $\gamma = (\gamma_1, \gamma_2, ..., \gamma_n)$ be the vector of these coefficients. Thus when bidder i gets the jth position, the probability of being clicked is $p_{ij} = \gamma_i \alpha_j$. Each time there is a search query adjacent to the players' preferences of advertising, the auctioneer asks the interested bidders to place their bid. Then the bids are collected and the most valuable spot in terms of click probability coefficient is given to the player who posted the higher bid, the second most valuable spot to the player with the second highest bid etc. Every time an advertisement is clicked the bidder is charged with the minimum bid that would have permitted him to maintain his position, that is the next lower bid. As for the bidders, every bidder has a valuation per unit of item he gets, that is v_i, and reflects how much his average income is when his advertisement is clicked. Therefore if we assume wlog. that the bids are in decreasing order $b_1 > b_2 > ... > b_n$, then bidder i who gets the ith position has an average profit of :

$$\alpha_i \gamma_i \left(v_i - b_{i+1}\right) \tag{1.1}$$

It turns out that there is not an obvious way for a player to bid in order to maximize his profit. So to analyse his behaviour is essential to make good predictions about him in terms of rationality. Moreover in order to predict this behaviour and test what economic theory suggest us, one major problem is to understand from the bids made what is the valuation of each player. This constitutes the major problem analysed throughout this thesis. Thus, the data available to us are the auction system's details and the complete bidding history of a particular keyword auction. Our goal is to estimate the private valuation that each player has.

A lot of work has been made to propose efficient inference methods at [1] , [2] , [3]. However one major flaw in all these methods is the silent assumption that a mixed nash equilibrium has been reached. In fact all these methods try to best explain the bidding history of the auction assuming that the valuations are in concordance with a certain type of equilibrium. This means that supposing a tuple of valuations $v_1, v_2, ..., v_n$ then the bids observed form

an equilibrium.In such dynamic settings this is not always the case for many different reasons. Many bidders appear and disappear changing the possible equilibria, also as the auction is performed under a non full information context it is quixotic to assume that the bidders will make it to reach an equilibrium. Obviously in many cases they make it to reach an equilibrium, however this assumption do not stand generally and provoke a big valuation estimation error when it do not hold even approximately.

A newer inference method proposed at [4] assume that bidders are learning agents, that is, bidding is implemented by algorithms which can learn in a repeated setting what is the best way to bid. That intuition reflects a completely different approach to econometrics in general. While often we analyse an economic system supposing that somehow players reached an equilibrium, we do not usually assume nothing about how they make it to reach an equilibrium. The key point of this new inference method is to assume the way players behave instead of supposing a final steady state. Proceeding in this line of thought it is important to assume correctly how players would behave, in fact, what does it mean to be a learning agent.

Online advertisement auctions can be structured as a repeated game. The player submit a bid to the auctioneer and then accumulate the profit that this bid created. Therefore the main question for a bidder is how can he bid optimally throughout the auction to maximize his profit. This task can be done efficiently using learning algorithms like the Multiplicative Weights Update. This algorithm maintains a probability distribution over the possible bids and update this probability distribution according to the performance of the different bids. So this procedure consists in a high level three steps approach

- submit a bid b according to a probability distribution
- suffer the loss or profit of this particular bid
- modify the probability distribution by observing the various bids performance

This particular algorithm can be slightly modified to work also in non full information contexts [21] . In fact in these type of contexts the bidder only receives the profit provoked by his bid and he does not know how other bids would have behave. The performance in such competitive and dynamic settings is evaluated using the notion of regret. Let Alg be the learning algorithm applied to bid optimally, then we define as the regret of Alg the difference between the profit produced by Alg and the profit that the best fixed bid would have produced. It can be proven that in general setting there do not exist an algorithm that achieves a Regret of $\Omega(\sqrt{T})$ where T

is the number of iterations. However there are simple algorithm that meet up to a constant term this error bound. The simplicity in addition with the optimal performance of these algorithms enforce the assumption that bidders would and should make use of it.

So the key point of this method assuming that bidders are learning agents is to inference their valuations supposing that their average regret vanishes over time. This property is called no-regret assumption. In fact following the assumption that bidders are learning agents and they achieve a regret of $\sqrt{T}$ it follows immediately that $\frac{\sqrt{T}}{T} \to 0$

Although this method has been proposed in the context of online auctions it is possible to consist a way more general approach to study economics. As an advantage of this method we can consider the fact that they capture the dynamic evolution of the system rather than only considering the final outcome.

Contribution

In this thesis we constructed a general auction simulation system. The purpose of this simulation system is to became a free tool available to everyone who wish to implement auctions' variations and test the predictions of the theory in dynamic environments. This tool aims to fill the gap created by the lack of free offered data in the context of online auctions and permit the study of problems related to auctions independently of industry societies. Moreover we implemented some econometric and statistical libraries. The econometric library provides useful functions to test the no regret assumption in the context of online auctions and aims to develop in a more general library whose purpose is to study no-regret economics. The statistical library offers functions that create statistical metrics with respect of online ad auctions

In addition to that we evaluated the new valuation inference method proposed by Nekipelov, Syrgkanis and Tardos at [4]. Apart from the general evaluation , we propose some critic which we hope will be the base for a further development of this brilliant idea. Furthermore we evaluate its robustness in settings when the auction implemented is not fair with respect to the bidders [5]. In fact Parkes et al. proposed a model under which the payments in the GSP auction are calculated with erroneous parameters and are not in concordance with the incomes, they proved that the GSP auction is in a sense of truthfulness of the outcome (higher valuation players should achieve a better placement) particularly robust. This robustness suggests a good behaviour when the inference valuation method is applied to non fair

auctions which is experimentally confirmed

Structure

In the **second chapter** we introduce the reader to the basic concepts of mechanism design, the domain of algorithmic game theory which concentrates its study to the analysis and study of auctions' mechanisms. We present some properties of auctions which are demanded by the classical theory and we critic their meaning.

In the **third chapter** we concentrate our analysis to the study of the auctions' format used in the online advertising context. We explain why these auctions deviate from the classical economic theory and why notwithstanding this fact they are used.

The **fourth chapter** explains how a learning agent should behave in dynamic environments. Although the main purpose of the chapter is to explain how a bidder should bid in online auctions in order to maximize his profit, the context of learning algorithms is way more general. In the first part of the chapter we present a discrete model and its optimization algorithms for dynamic repeated games. We then proceed by switching our analysis to continuous domains. Finally we present a swift of the classical analysis of learning agents to models where the player has not only to face the dynamic aspect of the environment but also a non full information context.

In the **final chapter** we describe our simulation system. Its properties and its possible further development. In addition to that we explain exactly how a bidder should use the algorithms presented in the precedent chapter to maximize his profit. Moreover we evaluate the newest proposed inference valuation method with an extended critic of his weaknesses and strengths. Finally we present some additional evidence about the robustness of the valuations inference method under nontruthful mechanisms and we enlighten the further work's possibilities.

Chapter 2

Game Theory and Mechanism Design Warm up

2.1 Introductory Examples

Mechanism design is a science that lies in the intersection between game theory, economy and computer science. His main purpose is to design mechanisms that guarantee good performance even where they are used by strategic players. The importance of good mechanisms is revealed by observing the failures of bad mechanisms.

Incentive Example: 2012 Olympic Games women's badminton tournament

The tournament has two phases, In the first phase there are four groups A,B,C,D with four teams each. The first two teams of each group advance to the second phase after a round robin series of matches. The second phase is composed by three direct elimination rounds where a team is either eliminated or advances, until one only team remains. In the first elimination round of the second face a team that ends second in the first phase faces a team that ended first in his group.

Now that we know the template, let's describe what happened. In the group D there was the best team of the tournament, the Chinese team of Qing and Wunlei. In their last match of the first phase they got beaten by the Danish team ending in such a way second. As a result the first of the group A would face the Chinese team. However in the last match of the A group two teams with a score of 2-0 each were playing against each other. For each of the two teams a win would have meant the first place and in the next round a match with the top Chinese team. So that's

what happened, they tried both to lose intentionally the match https://www.youtube.com/watch?v=7mq1ioqiWEo

There is no doubt that this is a scandal. But to understand more deeply what happened we should find what are the incentives of the players. What does a player want? Obviously to proceed as much as he can in the tournament. What the designer of the tournament template wanted? That each player tried as much as he could to win every match. So our paradigm shows that in the rules context of this tournament the incentives of the participants are not always aligned with the incentives of the organizers. For a more profound analysis we recommend Kleinberg's article at https://agtb.wordpress.com/2012/08/01/olympic-badminton-is-not-incentive-compatible-6/

This example shows how important is the design of good rules. The outcomes when the rules are not selected appropriately could be very curious.

So mechanism design is used to design rules that guarantee in a sense "good" outcomes.

2.2 First Price and Second Price Auction

The best way to start analysing mechanism design is by introducing some models for single item auctions. In this section we will present some basic auction formats and we will prove the first useful lemmas.

Single Item Auction: First of all let's see what is a simple single item auction. Suppose a seller has an item he wants to sold and there are n byers that are disposed to give money for this item. Each buyer has a private parameter called valuation, that represents how much the buyer wants the item and how much is disposed to pay for it. Then the auction starts and each buyer (we will start calling them players) bid. The seller decides based on the bids who gets the item and how much he has to pay for it. Now it remains to see what is the utility of a player and the utility of the seller. As for the utility of the seller we will see that is no such an obvious answer, the seller can be the government so that his purpose is (or it should be) to maximize the commonwealth of the society or it could be a private society whose purpose is to maximize revenues. As for the players we will use, throughout this theses, the quasilinear model of utility. The quasilinear model is as follows: If a player doesn't get the item then his utility is zero, on the contrary if it gets the item at a price p and his private valuation is v_i then his utility is equal to $v_i - p$.

Sealed-Bid Auctions: The auction format under which all the different auctions will be is the sealed bid auction format:

1. Each player i tells privately his bid b_i to the auctioneer (so the other bidders don't know his bid).
2. The auctioneer decides who gets the item (it can be no one)
3. The auctioneer decides the selling price

Now, we should start thinking how to implement steps 2 and 3. Step 2 is simple, just give the item at the bidder with the highest bid. This decision seems obvious, but as we will see later, when our main purpose is revenue maximization it is not always the correct choice. Step 3 is more difficult to think about, one simple idea, if for example our goal is to give the item at who wants it more without caring about revenue, is to give it for free. This is a bad idea, because the players are strategic and the auction will develop into a game of who announces the biggest number.

First Price Auctions: When the winning bidder pays his bid the auction is called First Price Auction. Although this auction seems simple, from the bidder's perspective is not so simple. To understand why just think the following experiment: there are two players, the first player is controlled by us, with valuation $v_i \in [0, 1]$ and the second player has a valuation X wich is a uniform random variable at $[0, 1]$. How we should bid to maximize our utility. If the other player just bids his valuation the answer is simple, we find our bid by maximizing the function $(v - b)(1 - b)$, but if the other player play in a different way (which is obvious because for him bidding every time his value will give him zero utility) how we should bid? This simple example shows that maybe this auction format is not so good. It is difficult for the auctioneer to make assumptions about how the bidders will behave and therefore to predict the outcome of the auction.

Second Price Auctions: An other very common in practice auction format is the second price auction first suggested by Vickrey at [6]. In the second price auction the highest bidder gets the item and pays the bid of the second highest bidder (plus a negligible ϵ). The intuition behind this auction format is simple, the winner has not to pay his bid, but just the least possible bid that would gave to him the win anyway.We will proceed by proving two very important theorems that makes this auction format so

widely used.

Theorem 2.2.1. *In a single item second price auction every bidder with valuation v_i has a dominant bid strategy, setting his bid b_i equal to his private valuation.*

This property is very important. Gives us a strong clue about bidders behaviour. If we have a system of rational players that we want to examine and a player has a dominant strategy, the assumption that this particular player will play according to this action is the weakest possible assumption in this type of setting.

Proof. Fix an arbitrary player i with valuation v_i and the vector of bids made by the other players b_{-i}.
Now we have to prove that no matter what the vector b_{-i} is, the utility of player i is maximized by setting $b_i = v_i$.

Let the maximum bid of the other players to be $B = max_{j \neq i} b_j$. Then we identify the cases:

1. If $b_i \geq B$ then $utility_i = v_i - B$
2. If $b_i \leq B$ then $utility_i = 0$

So the best we can hope is to make a bid b_i that in every situation assures as $utility_i = max\{v_i - B, 0\}$.
If $v_i = b_i$ then:

1. If $b_i \geq B \Rightarrow v_i \geq B$ then $utility_i = v_i - B = max\{v_i - B, 0\}$
2. If $b_i \leq B \Rightarrow v_i \leq B$ then $utility_i = 0 = max\{v_i - B, 0\}$

□

Theorem 2.2.2. *In a single item second price auction, every bidder who bids his valuation is guaranteed to have non-negative utility.*

Before proceeding to the proof of the previous theorem. It is important to highlight his meaning. Imagine an auction that a player has a dominant strategy but his pay off is negative. What we would do if we were in his shoes? Of course leave the auction.So an important property of an auction is the ability to maintain his players. This property among other things, surely requires that players have a non-negative utility.

Proof. Recall from the previous theorem that bidding his valuation assures to every player $utility_i = max\{v_i - B, 0\} \geq 0$. □

2.3 Myerson's lemma

Myerson's lemma constituted the basic tool upon which auctions were designed for many years. Myerson's at [7] provided a simple design approach to create DSIC auctions in more general settings than the one of the single item auction.

Before presenting it, let's generalize our auction model beyond single item auctions.

Single-parameter environment

1. The auction has a constant number n of bidders
2. Each bidder has a constant private valuation v_i that represents the value per unit of good that a bidder gets
3. X is the feasible set of n-vectors of good allocations $(x_1, x_2, \ldots, x_n)$

To spot the descriptive ability of this model let's give some examples.

1. In a single item auction $X = \{(x_1, x_2, \ldots, x_n) \,|x_i \in \{0, 1\}, \sum_{i=1}^{n} x_i \leq 1\}$
2. If the auction sells k identical goods then $X = \{(x_1, x_2, \ldots, x_n) \,|x_i \in \{0, 1\}, \sum_{i=1}^{n} x_i \leq k\}$

Sealed bid auction template: A mechanism designer in a sealed bid auction must implement two basic functions. Firstly the allocation rule function who takes as input the vector of bids and outputs the allocation ,of goods, vector. Secondly the payment rule function who takes as input the vector of bids and outputs the payments vector.Formally the sealed bid auction who is based on a particular single parameter environment is described by three steps:

1. collect the vector of bids $b = (b_1, b_2, ..., b_n)$
2. choose a feasible allocation using a function $x : \mathrm{Re}^n \to X$
3. get the payment of each player using a function $p : \mathrm{Re}^n \to \mathrm{Re}^n$

The sealed bid auction format is important to preserve the anonymity of the bidders. In real world situations where the environment is competing, bidders have a bounded budget and multiple single item are sold continuously a good tactic for the second highest bidder, even if the auction implemented is the second price auction is to overbid his true valuation. By this, he will inflict a bigger acquisition price for the first bidder leading him to finish his budget faster. In addition to that, even in other auction formats, bidding reflects a marketing tactic to which enterprises are very attentive to reveal.

For the player's utility we will use as always the quasilinear utility model. If an auction has an allocation rule x and a payment rule p, then the utility of player i under the bid profile b is: $utility_i = v_i x_i(b) - p_i(b)$
We will restrict ourselves to payments rules that satisfy the following equation: $0 \leq p_i(b) \leq b_i x_i(b)$
On the one hand the LHS of the inequality restrict our study to mechanisms where the seller doesn't pay the bidders. On the other hand the RHS ensures that always a truthtelling bidder have a non-negative utility (so we restrict ourselves directly to incentive compatible mechanisms).

Now we are almost ready to present Myerson's lemma. Although we should present two auxiliary definitions.

Definition 2.3.1. *An allocation rule x for a single parameter environment is implementable if there is a payment rule p such that the sealed bid auction (x, p) is DSIC.*

Just to explain further the definition, it is important to notice that if we project the space of DSIC mechanisms onto their allocation rules, we get the space of implementable allocation rules. So if our purpose is to design DSIC mechanisms we are obliged to restrict ourselves to allocation rules that are implementable.

Definition 2.3.2. *An allocation rule x for a single parameter environment is monotone if for every bidder i and other players bids b_{-i}, the function $x_i(z, b_{-i})$ is non decreasing in z.*

So if x is a monotone allocation rule, this means that bidding more can only give more stuff to the player that is bidding. Imagine for just a moment an allocation rule that is non monotone, like the second highest bidder wins. Intuitively is very strange, in fact it is not only intuitively a strange choice, also form the mechanism design perspective it is a bad choice as we will prove that if an allocation rule is not monotone then we are sure that is neither implementable.

Theorem 2.3.3. *Fix a single parameter environment*

1. *An allocation rule x is implementable if and only if it is monotone*
2. *If x is monotone, then there is a unique function p such that the sealed bid auction (x, p) is DSIC*
3. *The payment rule of DSIC mechanisms with an allocation function x is given by a precise formula*

Myerson's lemma is the foundation of mechanism design. Before proceeding to the proof, let's understand what it says. It says that if you want to create a DISC auction, the first step is to find an allocation rule that is monotone. Then you have not to think for a payment rule at all. It is unique and a precise formula exists for it.

Proof. DSIC condition states that for every bidder i and b_{-i}. The utility of bidder i have to be maximum when he bids his true valuation v_i.
Let (x, p) be a DSIC mechanism. Fix a particular bidder i and bids b_{-i}, as a shortcut let $x_i(z, b_{-i})$ be $x(z)$ and $p_i(z, b_{-i})$ be $p(z)$.
Let $0 \leq y < z$
Suppose now that bidder i has valuation y and bids z, then by the DSIC property we have that:

$$yx(y) - p(y) \geq yx(z) - p(z) \Rightarrow y(x(y) - x(z)) \geq p(y) - p(z) \tag{2.1}$$

Suppose now that bidder i has valuation z and bids y, then by the DSIC property we have that:

$$zx(z) - p(z) \geq zx(y) - p(y) \Rightarrow p(y) - p(z) \geq z(x(y) - x(z)) \tag{2.2}$$

Combining the two above inequalities we get that:

$$y(x(y) - x(z)) \geq p(y) - p(z) \geq z(x(y) - x(z)) \quad \forall y, z \quad 0 \leq y < z \tag{2.3}$$

Since the above inequality stands $\forall y, z \quad 0 \leq y < z$ then the function x must be non decreasing. Indeed suppose that $\exists z_1, z_2$ with $0 \leq z_1 < z_2$ and $x(z_1) > x(z_2)$.Then is must hold that:
$z_1(x(z_1) - x(z_2)) \geq z_2(x(z_1) - x(z_2)) \Rightarrow z_1 \geq z_2$ that is a contradiction.
So, since now we have prove that $(x, p) \quad DSIC \Rightarrow x \quad monotone$.

Now we have to figure out how the payment rule must be. Suppose that x is differentiable. Then dividing by $y - z$ and taking the limit of z approaching y. We get $p^{'}(z) = zx^{'}(z)$
So:

$p(b_i) - p(0) = \int_0^{b_i} zx^{'}(z)$

Then by noticing that $p(0) = 0$ since the auction is incentive compatible even for a bidder with a valuation of zero and to assure that bidding truthfully has not negative utility his payment must be zero.

$p(b_i) = \int_0^{b_i} zx^{'}(z)$

This is an exact formula for the payment rule. The only possible formula that provided (x, p) is DSIC, works.
Of course the idea of Myerson's doesn't require the function x to be differentiable. Indeed if it is stepwise constant and we redefine the derivative of such a function as the jump in the position of discontinuity the formula works in the same way. The simplified equation for stepwise constant functions is the following:
$p_i(b_i, b_{-i}) = \sum_{j=1}^{l} z_l(x_i(z_l^+, b_{-i}) - x_i(z_l, b_{-i}))$
where the points $z_1, z_2, ..., z_l$ are the point of discontinuity of the allocation function $x_i(z, b_{-i})$ from zero to b_i.

Now it rests to prove that if x is a monotone allocation rule then it is implementable. Let's suppose that x is monotone and differentiable.We will prove that the sealed bid auction with a payment rule $p_i(b_i, b_{-i}) = \int_0^{b_i} zx_i^{'}(z, b_{-i})$ for every bidder i is DSIC.
Let's see what is the utility of bidder i (we will follow the previous notation).
$utility_i(b) = v_i x_i(b) - p_i(b) = v_i x_i(b) - \int_0^{b_i} zx^{'}(z)$
Taking the partial derivative we get:
$\frac{\partial utility_i}{\partial b} = v_i \frac{\partial x_i}{\partial b} - bx^{'}(b)$

And, we get the maximum of the utility by taking his derivative equal to zero
$\frac{\partial utility_i}{\partial b} = 0 \Rightarrow b = v_i$

So we just proved that playing his valuation is a dominant strategy for the player. Now to conclude the proof we will prove that being truthful never leads to a negative utility. Or equivalently we will prove that:

$v_i x_i(v_i) \geq \int_0^{v_i} zx^{'}(z) \Rightarrow$
$v_i x_i(v_i) \geq \int_0^{v_i} (zx(z))^{'} - \int_0^{v_i} z^{'}x(z) \Rightarrow$
$\int_0^{v_i} x(z) \geq 0$
which holds since $x(z)$ is obviously non negative.
Equivalent arguments hold even when the function x is not differentiable but only stepwise constant.

□

2.4 Hierarchy of Equilibrium Concepts

In this section we will present and compare the different notions of equilibria in games more in detail discussed at [8]. Since now we focused our attention to auctions in which the players had a dominant strategy action. This is not always the case, as in many games there is not a dominant strategy action. We will enlarge the set of "logical" outcomes by permitting the players to converge in different types of equilibria.

In the auction setting we will see that these equilibria are quite useful. First of all because the notions allow us to interpret the outcome in different ways and conclude different things. Secondly because the dynamic nature of auctions does not allow us to conclude in different settings which equilibrium convergence is more "logical".

Before proceeding to the analysis of the different equilibrium concepts we will present the game setting in which we will define them.

Cost Minimization Game :

1. a finite number k of players
2. a finite strategy S_i for each player
3. a cost function $C_i : S_1 \times S_2 \times, ..., \times S_k \to R$ for each player

By s we will denote an outcome of the game (i.e. $s \in S_1 \times S_2 \times, ..., \times S_k$) and for every player i we will denote a particular action that he took by s_i and the other players action by s_{-i}

Now we are ready to present the equilibrium concepts. Obviously the first equilibrium concept is due to Nash at [9]

Definition 2.4.1. ***Pure Nash Equilibrium***
A strategy profile s of a cost minimization game is a pure Nash equilibrium if for every player i and every unilateral deviation $s_i^{'} \in S_i$:
$C_i(s) \leq C_i(s_i^{'}, s_{-i})$

The above definition states in simple word that an outcome s is a pure nash equilibrium if no player wants to change his action, assuming the other players will do the same. Although this is a very basic definition it turns out it does not always exist. So now we are obliged to extent our definition by permitting randomization over actions.

Definition 2.4.2. ***Mixed Nash Equilibrium***
Distributions $\sigma_1, \sigma_2, ..., \sigma_k$ over strategy sets $S_1, S_2, ..., S_k$ of a cost minimization game constitute a mixed Nash equilibrium if for every player i and every unilateral deviation $s_i^{'} \in S_i$:
$E_{s\sim\sigma}\left[C_i(s)\right] \leq E_{s\sim\sigma}\left[C_i(s_i^{'}, s_{-i})\right]$
where σ is the product distribution $\sigma_1 \times \sigma_2 \times \times ... \times \sigma_k$

This definition is the same as the definition of a pure nash equilibrium except from the fact that it permits to have a distribution over the actions and play according to this distribution. The good news are that in cost minimization games as we defined them previously such an equilibria always exist. However it turns out that calculate an equilibria of this type can be computationally hard. How we can suppose that player will reach such an equilibrium if we are not capable even to calculate one. This problem is faced ,as always, by enlarging the set of possible equilibria and hoping that this new set will be computationally easier to find.

Definition 2.4.3. ***Correlated Equilibrium***
A distribution σ over the set of outcomes $S_1 \times S_2 \times ... \times S_k$ of a cost minimization game constitute a correlated equilibrium if for every player i , every strategy s_i and every unilateral deviation $s_i^{'} \in S_i$:
$E_{s\sim\sigma}\left[C_i(s)|s_i\right] \leq E_{s\sim\sigma}\left[C_i(s_i^{'}, s_{-i})|s_i\right]$

Note that in the last definition the distribution σ needs not to be a product distribution. In addition to that the expectations are conditional to the action s_i.

The intuition behind this weird definition is very simple. Suppose you know a common distribution σ and a trusted third party suggests you the action you might take according to this distribution, then if σ is a correlated equilibria you must follow the advice.

To make it even more clear just think at the traffic light. We know that the distribution from which the traffic light picks his proposals gives green to the one road and red to the other. Now, minimizing our loss is achieved by following the advice of the traffic light.The traffic light distribution to the green red signs is a perfect example of what a correlated equilibria is.

This new type of equilibria is tractable is the sense that exists and can be calculated efficiently. We will introduce the last type of equilibria which is a further generalisation of the equilibrium concept.

Definition 2.4.4. ***Coarse Correlated Equilibrium***
A distribution σ over the set of outcomes $S_1 \times S_2 \times ... \times S_k$ of a cost minimization game constitute a coarse correlated equilibrium if for every player i and every unilateral deviation $s_i^{'} \in S_i$:
$E_{s\sim\sigma}\left[C_i(s)\right] \leq E_{s\sim\sigma}\left[C_i(s_i^{'}, s_{-i})\right]$

Note that this equilibria is a generalization of the previous one, in the sense that it does not requires protection when the suggested outcome is known. The only thing that the agent knows here is the distribution from which the outcomes are picked.It turns out that also this equilibria can be explained by the example of a traffic light. However now imagine that we must decide before the advice of the traffic light if we will follow his distribution. This is quite different. In the previous equilibrium concept the traffic light tell us what we should do and it turns out that the best thing we can do knowing what the outcome was is to follow his advice. However in this type of equilibrium we should decide before the outcome of the distribution of we will follow what the traffic light we will tell us.

To conclude this section we present an example constructed by Tim Roughgarden at [8] that illustrates all the equilibrium concepts together and enlighten their relation as a strictly increasing series of sets.

Example Suppose we have a graph with two nodes one source s and one destination t.The graph has 6 parallel edges. There are 4 players that choose an edge to "travel" form s to t. The cost is proportional to how many other players are using the same edge to travel. For example when 3 out of 4 players choose the same edge then each of them will suffer a cost equal of 3. When a player is the only one who uses a particular edge then he suffers a cost of 1 etc.Let's present an equilibrium of each type into this setting.

1. The $\binom{6}{4}$ outcomes in which every player chooses a different edge to travel constitutes obviously a nash equilibrium

2. If each player selects uniformly and independently an edge then it suffers an expected loss of $\frac{3}{2}$. The product of these uniform distributions is a mixed nash equilibrium because no player has any incentive to deviate in an other distribution.

3. Suppose now that the distribution over the outcomes is uniform over the following set: one edge with two players and two edges with one player each. Then playing according to the distribution knowing what

our outcome is makes us suffer an expected loss of $\frac{3}{2}$. Playing an other action continues to give us the same expected cost. Thus this correlated distribution constitutes a correlated equilibria

4. The uniform distribution over the subset of the previous outcomes in which the edges are chosen either by the set $\{0, 1, 2\}$ or by the set $\{3, 4, 5\}$ is a coarse correlated equilibria making each player suffer a cost of $\frac{3}{2}$. Note that this is not a correlated equilibria since knowing our outcome will lead us to change our decision to an arbitrary edge of the other set and suffer a loss of 1.

The previous example underlines that the sets of equilibria form a series of strictly increasing sets. Moreover ,in making an assumption about where the players will converge, we must be careful about two parameters, the tractability to reach an equilibrium and how simple it is to implement distributed algorithms that reach this equilibrium.

Chapter 3

Online Ad Auctions

In this chapter we will present the principal used model for online advertising auctions. Subsequently we will compare under a truthfulness point of view what Myerson's lemma suggest and what is actually implemented.

3.1 Introduction

Online ad auctions are auctions that are used to sell space from an internet search query to a company which uses it for publicity. In order to ensure the reader for the importance of the study of these mechanisms, it is important to mention that a large part of Yahoo's and Google's revenues are created by these auctions. Just to give some numbers, Google's 2005 total revenue was about 6.14 billion of dollars. Over the 98 percent of this sum was created by online ad auctions. On the other hand Yahoo's revenue in 2005 was about 5.26 billions of dollars and it is believed that over the half of these revenues were highly connected with this type of advertising. [10]

In addition to that, the last years, this market exploded as the global online advertising market is continuously increasing `http://www.wordstream.com/articles/google-earnings`.

Before proceeding to the formal definition of the model we will give a high level description of how this system works. Each time an internet user enters a search query an online auction runs in real time. Along with the results of the search query, some advertisements are displayed. Behind each of these advertisements there is a company who bids on this particular search query to obtain the possibility to display his advertisement on the result page of the user. The bid made by the company reflects his maximum willingness to pay if his advertisement is clicked. So every time a user clicks a particular advertisement, the company that made the advertisement is charged by a quantity less or equal to the bid submitted. Hence the company bids his maximum payment per click. Obviously more than one places on the

result page are sold each time. As expected, the highest an advertisement is displayed, the more probable is that the ad displayed in this position will be clicked. The mechanism has then to choose two different things. Firstly the allocation rule and secondly the payments rule.

Let's give a particular example of this situation. Imagine a company who sells flowers. Each time a user type to the search engine the phrase "buy flowers" the company submit a bid. Then the mechanism decides in which position to display the advertisement of the flower's company. If the user clicks this advertisement then the company is charged.

3.2 Model Description

Each bidder is associated with a click probability γ_i. Each position has a click probability α_i. So, the probability of bidder i who is displayed in the j position to be clicked is denoted by p_{ij} and equals the product $\alpha_i\gamma_j$.The bidders are asked by the system to submit a bid b_i. Then the bids are collected and the mechanism decides the assignment of positions to the bidders using a particular allocation rule. Each time a particular advertisement is clicked the system charges the bidder a quantity less or equal to his bid using a payments function.

Thus we denote by:

1. m the number of positions sold each time
2. n the number of bidders
3. u_i is the valuation of bidder i per click
4. $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$ the vector of position's coefficients, wlog we will assume that $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_m$
5. $\gamma = (\gamma_1, \gamma_2, ..., \gamma_n)$ the vector of bidder's coefficients
6. $p_{ij} = \alpha_i\gamma_j$ the probability of being clicked of bidder i when placed to position j
7. r is the reserve price of the auction
8. $b = (b_1, b_2, ..., b_n)$ is the vector of bids submitted by the players

Following the previous notation we introduce some additional symbols. Let $\sigma_i(b)$ to be the position where player i is allocated under the bid profile b. Also denote by $\pi(j, b)$ the index of the bidder placed in position j under a particular bid profile b .In addition to that let $c_{ij}(b)$ to be the cost per click of bidder i when placed to position j.
The expected cost of player i under the profile bid b is then:

$$C_i(b) = \alpha_{\sigma_i(b)}\gamma_i c_{i\sigma_i(b)}(b) \tag{3.1}$$

Furthermore the click probability as a function of the bid profile b is:

$$P_i(b) = \alpha_{\sigma_i(b)}\gamma_i \tag{3.2}$$

For the player's utility we will use as always the quasilinear model, so the expected utility of player i under bid profile b is denoted by $U_i(b)$ and is given by the equation:

$$U_i(b) = u_i P_i(b) - C_i(b) \tag{3.3}$$

3.3 Myerson's lemma versus GSP

In this section we will present the two basic implementations of the online ad auctions model.The implementation which uses Myerson's lemma 2.2 to calculate the payments function and the most widely used mechanism, the Generalised Second Price auction. As always the functions that needs to be implemented are the allocation and the payments function.

Allocation The allocation function used in common by the two mechanism is the most obvious possible. Give the positions from the most valuable (α_1) to the less valuable (α_m) to the m biggest bidders respectively assuming that at least m bidders submitted a bid higher of the reserve price r. Otherwise give the $k \leq m$ first positions to the k bidders who bid more than the reserve price and do not allocate the rest of the positions. Note that this allocation function is increasing for each player i. Thus from Myerson's lemma we know that it can be implemented to a DSIC mechanism.

Myerson's derived auction For notation simplicity we will assume that in the bids vector b the bids are non increasing (i.e. $b_1 \geq b_2 \geq ... \geq b_n$), thus, using Myerson's lemma we can derive the payment per click function which is:

$$c_{i\sigma_i(b)}(b) = \sum_{j=i}^{min(m,biddersallocated)} \frac{a_j - a_{j+1}}{a_i} b_{j+1} \tag{3.4}$$

Note that the payment function derived by Myerson's lemma suggests that the i-th bidder should pay each time is clicked, a suitable convex combination of the smaller bids. This auction is obviously DSIC as the allocation function is non-decreasing for every bidder and the payments function is the one provided by Myerson's lemma.

From the last equation we can easily deduce that the average cost of bidder i under the profile bid b (continue assuming bids are in a non increasing order) is:

$$C_i(b) = \sum_{j=i}^{min(m,biddersallocated)} \gamma_i(a_j - a_{j+1}) b_{j+1} \tag{3.5}$$

GSP The GSP auction is the most widely used model for online advertising auctions. The main reason why is used instead of the optimal (in a DSIC sense) Myerson's derived auction is its simplicity. Furthermore other good properties of this mechanism make it extremely useful.Before proceeding to the comparison of the two mechanisms we shall describe the GSP implementation. As the name suggests it generalizes the second price idea of Vickrey [6] in such a way that if a bidder is placed in the j-th position then when is clicked he has to pay the minimum amount of money he should have bided to maintain his position, thus the next smaller bid. The cost per click function is then (here the formulas are way simpler and we do not need to suppose that the vector of bids is non-increasing):

$$c_{ij}(b) = max\{b_{\pi_{j+1}}, r\} \mathbb{1}\{b_i \geq r\} \tag{3.6}$$

Thus the expected cost of bidder i is:

$$C_i(b) = \alpha_{\sigma_i(b)} \gamma_i max\{b_{\pi_{\sigma_i(b)+1}}, r\} \mathbb{1}\{b_i \geq r\} \tag{3.7}$$

However it seems counterintuitive that the GSP payments function does not form a DSIC mechanism, this is the sad reality. We know that the payments function given by Myerson's lemma is the only one which creates a DSIC mechanism. As an example, suppose there are three bidders with valuations $v_1 = 10$, $v_2 = 8$,$v_1 = 3$ and there are two positions available, the first one with $\alpha_1 = 1$ and the second one with $\alpha_2 = 0.99$ (γ are omitted). Then if every bidder bids truthfully, the first bidder's utility is $1(10 - 8) = 2$. By deviating to a bid of 4 his utility would increase as $0.99(10 - 3) = 6.93 > 2$. So being sincere is not a dominant strategy for the first bidder and consequently the auction is not DSIC.

An interesting question arises here. If the vector of bids stabilizes in a particular vector b. Then what properties should this vector have? This question was answered by Ostrovsky et al. at [10] and by Variant at [2]. Obviously the first condition that it has to fulfil is the nash equilibrium requirement. This means that every bidder is best responding to the other bids. Assuming once again for notation simplicity that the vector of bids is non-increasing and wlog that all the bids are bigger than the reserve price we get that a bid profile b constitute a nash equilibrium ($a_k = 0$ for $k > m$) iff $\forall i$:

$$\alpha_i\gamma_i(v_i - b_{i+1}) \geq \alpha_j\gamma_i(v_i - b_{j+1}) \quad j > i \tag{3.8}$$

$$\alpha_i\gamma_i(v_i - b_{i+1}) \geq \alpha_j\gamma_i(v_i - b_j) \quad j < i \tag{3.9}$$

Presenting an example will give us more clue about how an equilibrium vector should look like. Imagine there are two bidders bidding for one object. The first bidder has a valuation of 100 and the second one a valuation of 40. The bid vector (30,120) constitutes a nash equilibrium for the one shot game, however in a repeated game is quite irrational to assume that the bids vectors will stabilize to this instance as the first bidder can increase his bid and oblige the second bidder to abandon the auction. Following this idea we will restrict ourselves to a more specific case of equilibria called locally envy-free equilibria. The equilibria of this type are characterised by the following relation.
$\forall i$

$$\alpha_i\gamma_i(v_i - b_{i+1}) \geq \alpha_j\gamma_i(v_i - b_{j+1}) \tag{3.10}$$

We have strengthen the second inequality of the general nash equilibria. This relation does not give us the exact characterization of the equilibrium that players will reach as there are multiple bid profiles that satisfy the

equation. Although if the bid profile stabilizes somewhere then we must have reached an locally envy-free equilibrium.

Under the assumption that the players will reach a locally envy-free equilibrium the gsp auction enjoy some interesting properties. We present maybe the most important one.

Theorem 3.3.1.
If the number of advertisers is greater than the number of positions available then the auction revenue of any locally envy-free equilibrium under the GSP auction is at least the revenue of the Myerson's derived auction under the assumption that bidders are truthful

The proof of the previous theorem was made by Ostrovsky et al. at [10].

What does the previous paragraph tells us? Simple, when you are the auctioneer and you want to increase your incomes then you should better use the GSP auction rather than the Myerson's derived auction.

It is obvious that this is an appealing property for an auction designer. As well as this property, the GSP auction is preferred for some additional robustness properties that we will spot later. In addition to that, observe that from a bidder point of view in some sense the GSP auction preserves its privacy by not revealing directly to the auctioneer his private valuation. Hence among other reasons, GSP achieves at the same time three important goals, it preserves bidder's privacy, it achieves greater revenue and it is extremely simple.

However GSP payments function is different from the unique function that suggests Myerson's lemma to obtain a DSIC mechanism. Therefore for bidders there is not an obvious way to bid. The question to how bidders should bid is extremely interesting and will be examined through an entire chapter.

Chapter 4

Online Learning

In the previous chapter we presented the GSP auction which constitutes the most used online ad auction format. This auction is not DSIC and does not provide bidders with any obvious bid tactic. Therefore the main aim of this chapter is to resolute the question of how these bidders should bid.

In the first part of the chapter we will present the Multiplicative Weights Update template proposed by Arora et al. at [11]. The idea of the multiplicative weights update algorithm has been discovered and proposed many times in different sectors.In the first studies of the Weight Update idea an additive update rule has been proposed at [12] and the FTL (follow the leader algorithm) at [13].Furthermore in the machine learning bibliography Freud and Schapire proposed the well known Adaboost algorithm at [14]. The very first ideas were developed in the game theory community. So, the first implementation of this idea in economics were presented at [15] and later were reintroduced at [16].

In the second part of the chapter, we introduce the general Online Convex Optimization framework and the analysis of its basic algorithms. Moreover, finally we introduce describe the basic algorithms which generalize the Multiplicative Weight Update idea in non full information settings.

The application of these algorithms in our framework is straightforward. We state that the main problem is how bidders should bid in non-DSIC mechanisms. These algorithms step by step learn what is the best tactic for each bidder.

4.1 Introductory Algorithms

4.1.1 Multiplicative Weights Updates

The Multiplicative Weights Updates Algorithm is more than an algorithm, it constitutes a general method of solving problems.The idea behind the Multiplicative Weights Updates algorithm is that whenever i have a set of different possible actions, maintaining a distribution over the actions permits me ,in the future, to choose an action accordingly to the accumulated experience.

In a sense this is what humans do. When there is a problem, everyone of us evaluates the different actions that can take, the evaluation of these actions is based on some accumulated experience of how these actions performed in the past. Of course actions that performed well in the past, are at least in our minds more likely to perform well in the future.

We will follow the general template proposed by Arora et al. in at [11] Let's define some useful quantities:

1. $\mathbf{A} = \{1, ..n\}$ is the set of possible actions

2. **P** is the set of possible outcomes

3. **M** is the cost matrix, such that **M(i,j)** represents the cost of taking the action i when the outcome is j.

4. $\forall i \in A, \forall j \in P \; M(i,j) \in [-l, r,] \; ,where \; l \leq r$

5. $w_i^t \; , \forall i \in A$ is the weight assigned to action i at time t

6. $p_i^t \; , \forall i \in A$ is the selection probability assigned to action i at time t

7. $D^t = \{p_1^t, p_2^t, .., p_n^t\}$ is the distribution over actions at time t

Algorithm 1 MWU algorithm

Initialize $w_1^t = 1, \quad \forall i \in A$
for $t = 1 \quad to \quad T$ **do**
  pick an action $i_t \sim D^t$
  suffer the outcome $j^t \in P$
  for $i = 1 \quad to \quad n$ **do** ▷ **weigths update**
    if $M(i,j) \geq 0$ **then**
      $w_i^{t+1} = w_i^t(1-\epsilon)^{\frac{M(i,j^t)}{r}}$
    else if $M(i,j) < 0$ **then**
      $w_i^{t+1} = w_i^t(1+\epsilon)^{-\frac{M(i,j^t)}{r}}$
  $\Phi^{t+1} = \sum_{i=1}^{n} w_i^{t+1}$
  for $i = 1 \quad to \quad n$ **do** ▷ normalize to create a distribution
    $p_i^{t+1} = \frac{w_i^{t+1}}{\Phi^{t+1}}$
  $D^{t+1} = \{p_1^{t+1}, p_2^{t+1}, .., p_n^{t+1}\}$

Intuitive Explanation Before proceding to the analysis of the algorithm, it is worth giving an explanation to the weights update rule.

$M(i,j^t) \geq 0$ means that in this particular time t the outcome of nature is j^t and our action i suffers a loss of magnitude $M(i,j^t)$.So what we should do? Of course diminishing the probability of choosing the action i to the next iteration. Selecting $\epsilon \in (0,1)$ does what we want.

Following the same reasoning we should augment the probability given to an action who suffers a negative loss $M(i,j^t)$ (i.e it has a positive payoff). The update rule does what we expect.

Since the MWU algorithm is probabilistic, the framework with which we will compare the performance is the expected penalty.So let's denote as $M(D^t, j^t) = \sum_{i=1}^{n} p_i^t M(i,j^t)$ the expected penalty for the outcome $j^t \in P$ of the probability distribution D^t of our algorithm at time t.

Theorem 4.1.1. *if* $\epsilon \leq \frac{1}{2}$ *then* $\forall i \in A$*:*
$\sum_t M(D^t, j^t) \leq \frac{rln(n)}{\epsilon} + (1+\epsilon) \sum_{t:M(i,j^t)\geq 0} M(i,j^t) + (1-\epsilon) \sum_{t:M(i,j^t)<0} M(i,j^t)$

Proof.

we will use three basic inequalities

1. $(1-\epsilon)^x \leq (1-\epsilon x) \quad x \in [0,1]$
2. $(1+\epsilon)^{-x} \leq (1-\epsilon x)$, $x \in [0,1]$

3. $e^x \geq x+1$, $\forall x \in \mathrm{Re}$

from ϕ_t definition we have

$$\phi_{t+1} = \sum_{i=1}^{n} w_i^{t+1} = \sum_{i:M(i,j^t)\geq 0} w_i^t(1-\epsilon)^{\frac{M(i,j^t)}{r}} + \sum_{i:M(i,j^t)<0} w_i^t(1+\epsilon)^{-\frac{M(i,j^t)}{r}}$$
$$\leq$$
$$\leq \sum_{i:M(i,j^t)\geq 0} w_i^t(1-\epsilon\frac{M(i,j^t)}{r}) + \sum_{i:M(i,j^t)<0} w_i^t(1-\epsilon\frac{M(i,j^t)}{r}) =$$
$$= \sum_{i=1}^{n} w_i^t(1-\epsilon\frac{M(i,j^t)}{r}) = \sum_{i=1}^{n} w_i^t - \epsilon\sum_{i=1}^{n} w_i^t\frac{M(i,j^t)}{r} = \phi_t - \frac{\epsilon\phi_t}{r}\sum_{i=1}^{n}\frac{w_i^t}{\phi_t}M(i,j^t) =$$
$$\phi_t - \frac{\epsilon\phi_t}{r}\sum_{i=1}^{n} p_i^t M(i,j^t) = \phi_t - \frac{\epsilon\phi_t}{r}M(D^t,j^t) = \phi_t(1-\frac{\epsilon}{r}M(D^t,j^t)) \leq$$
$$\leq \phi_t e^{-\frac{\epsilon}{r}M(D^t,j^t)}$$

by induction on the number of iterations and from the fact that $\phi_1 = n$ we easily get that

$$\phi_T \leq n e^{-\frac{\epsilon}{r}\sum_{t=1}^{T-1} M(D^t,j^t)} \tag{4.1}$$

In addition to that, since the weights remain positives during the entire algorithm, we have that:

$$\phi_T \geq w_i^T = (1-\epsilon)^{\sum_{t:M(i,j^t)>0}\frac{M(i,j^t)}{r}}(1+\epsilon)^{\sum_{t:M(i,j^t)<0}-\frac{M(i,j^t)}{r}} \quad \forall i \in A \tag{4.2}$$

combining the two inequalities we get

$$n e^{-\frac{\epsilon}{r}\sum_{t=1}^{T-1} M(D^t,j^t)} \geq (1-\epsilon)^{\sum_{t:>0}\frac{M(i,j^t)}{r}}(1+\epsilon)^{\sum_{t:<0}-\frac{M(i,j^t)}{r}} \quad \forall i \in A \tag{4.3}$$

now using the inequalities:

1. $ln(\frac{1}{1-\epsilon}) \leq \epsilon + \epsilon^2 \quad \forall \epsilon \in \left(0, \frac{1}{2}\right)$
2. $ln(1+\epsilon) \geq \epsilon - \epsilon^2 \quad \forall \epsilon \in \left(0, \frac{1}{2}\right)$

we easily conclude the proof

□

Corollary 4.1.1.1. *For $T > 2ln(n)$ and $\epsilon = \sqrt{\frac{ln(n)}{2T}}$ we have that :*

$$\frac{\sum_t M(D^t, j^t)}{T} \leq O(\frac{1}{\sqrt{T}}) + \frac{\sum_t M(i, j^t)}{T} \quad \forall i \in A$$

Proof.
We first observe that

$$(1+\epsilon) \sum_{t:M(i,J^t)\geq 0} M(i, j^t) \leq \sum_{t:M(i,J^t)\geq 0} M(i, j^t) + \epsilon r T$$

$$(1-\epsilon) \sum_{t:M(i,J^t)<0} M(i, j^t) \leq \sum_{t:M(i,J^t)\geq 0} M(i, j^t) + \epsilon l T$$

$$l \leq r$$

so from the previous theorem we get that

$$\sum_t M(D^t, j^t) \leq \frac{r ln(n)}{\epsilon} + 2\epsilon r T + \sum_t M(i, j^t) \quad \forall i \in A$$

Finally by substituting ϵ we get the desired expression

□

Remarks: First of all we should notice that the theorems above make no assumptions about the knowledge an adversary could have. It is possible that the adversary knows our distribution over the actions and picks an event according to this.This property is very important particularly in repeated games like auctions. It is lucid that after many iterations our bids provide private information to our adversaries. Therefore the algorithm protect us against even the most powerful adversaries which knows everything about us. Secondly, the initial uniform distribution over the actions reflect our starting ignorance.Finally maybe the most interesting result is that the average difference between the losses of our actions and the losses of a fixed action vanishes as $O(\frac{1}{\sqrt{T}})$.

4.1.2 Hedge Algorithm

The Hedge algorithm constitutes a slightly different version of the Weights Updates Algorithm by differentiating the update rule.Was first proposed by Freud and Schapire in [17]. Although it is not as general, the performance achieved for particular instances is similar,despite the fact that the analysis is simpler.

We first present the algorithm and then we proceed to the technical analysis. The framework remains the same despite the fact that the losses are now bounded in the $[0, \infty)$ range. So:

$M(i, j) \in [0, \infty) \quad \forall i \in A, j \in P$

Algorithm 2 Hedge algorithm

Initialize $w_1^t = 1, \quad \forall i \in A$
for $t = 1 \quad to \quad T$ **do**
    pick an action $i_t \sim D^t$
    suffer the outcome $j^t \in P$
    for $i = 1 \quad to \quad n$ **do** ▷ **weigths update**
        $w_i^{t+1} = w_i^t e^{-M(i,j^t)}$
    $\Phi^{t+1} = \sum_{i=1}^{n} w_i^{t+1}$
    for $i = 1 \quad to \quad n$ **do** ▷ normalize to create a distribution
        $p_i^{t+1} = \frac{w_i^{t+1}}{\Phi^{t+1}}$
    $D^{t+1} = \{p_1^{t+1}, p_2^{t+1}, .., p_n^{t+1}\}$

Theorem 4.1.2. *Let* $M(D^t, j^t)^2 = \sum_{i=1}^{n} p_i^t M(i, j^t)^2$ *and as before* $M(D^t, j^t) = \sum_{i=1}^{n} p_i^t M(i, j^t)$.

Then $\forall i^* \in A$:

$$\sum_t M(D^t, j^t) - \sum_t M(i^*, j^t) \leq \epsilon \sum_t M(D^t, j^t)^2 + \frac{ln(n)}{\epsilon}$$

Proof.

We will use two basic inequalities:

1. $e^x \geq x + 1 \quad \forall x \in \mathrm{Re}$
2. $e^{-x} \leq 1 - x - x^2 \quad \forall x \in [0, \infty)$

From the definition of ϕ_t we have that:
$\phi_{t+1} = \sum_i w_i^t e^{-\epsilon M(i,j^t)} = \phi_t \sum_i p_i^t e^{-\epsilon M(i,j^t)} \leq \phi_t \sum_i p_i^t (1 - \epsilon M(i, j^t) - \epsilon^2 M(i, j^t)^2) =$

$= \phi_t(1 - \epsilon M(D^t, j^t) - \epsilon^2 M(D^t, j^t)^2) \leq \phi_t e^{-\epsilon M(D^t, j^t) - \epsilon^2 M(D^t, j^t)^2}$

By induction it follows that after T iterations:
$\phi_{T+1} \leq n e^{-\epsilon \sum_{t=1}^{T} M(D^t, j^t) - \epsilon^2 \sum_{t=1}^{T} M(D^t, j^t)^2}$

In addition to that, since the weights remain positive through the entire algorithm, we have that:
$\phi_{T+1} \geq w_{i^*}^{T+1} = e^{-\sum_{t=1}^{T} M(i^*, j^t)} \quad \forall i^* \in A$
combining the two last inequalities we get the desired result. □

Remarks: An attentive reader will notice that following the same reasoning as we did in the Multiplicative Weights Updates algorithm and noticing that $\sum_{t=1}^{T} M(D^t, j^t)^2 \leq T$. We get the same asymptotic behaviour by selecting $\epsilon = \sqrt{\frac{ln(n)}{T}}$. In addition to that, before we define formally the quantity that we are obsessively bounding let's give some intuition. The quantity $\sum_t M(D^t, j^t) - \sum_t M(i^*, j^t)$ quantifies the performance of our algorithm by comparing it to the best constant action that we could have taken in hindsight. This value is the regret of our online algorithm and we will understand its importance later on.

To conclude it is important to understand that our framework is equivalent to an adversary that after each iteration selects a function l_t which gives a cost to each of our possible actions, the equivalence comes from the manner we have defined our outcome set P (i.e without restrictions), so by selecting $M(i, l^t) = l^t(i)$, the equivalence becomes clear.

4.2 Offline Convex Optimization and Gradient Descent

Before proceeding to the online framework it is useful to present some basic background of offline convex optimization and it's most basic algorithm.

Our Goal: Minimize a continuous and convex function over a convex subset of the Euclidean Space.

Basic definitions and properties:

- A set K is convex iff $\forall x, y \in K$, then $ax + (1-a)y \in K \ \forall a \in [0,1]$
- A function f over a convex set K is convex iff $\forall x, y \in K$, then $f(ax + (1-a)y) \leq af(x) + (1-a)f(y) \ \forall a \in [0,1]$
- if f is differentiable and ∇f exists $\forall x \in K$ then is convex iff $\forall x, y \in K$ $f(y) \geq f(x) + \nabla f(x)^T (y - x)$
- We denote by D an upper bound on the diameter of K: $\forall x, y \in K \quad \|x - y\| \leq D$
- We denote by G an upper bound to the norm of the gradient: $\|\nabla f(x)\| \leq G \quad \forall x \in K$
- projection onto a convex set (i.e the closest point into the convex set of a given point): $\Pi_K(y) = argmin_{x \in K} \|x - y\|$

Now before proceeding in the presentation of the gradient descent algorithm it will be useful to highlight two important theorems.

Theorem 4.2.1 (Pythagora's Theorem)**.** *Let* $K \subseteq \mathrm{Re}^d$ *be a convex set,* $y \in \mathrm{Re}^d$ *and* $x = \Pi_K(y)$*, then*
$\forall z \in K \quad \|y - z\| \ \geq \|x - z\|$

Theorem 4.2.2 (Karush-Kuhn-Tucker)**.** *Let* $K \subseteq \mathrm{Re}^d$ *be a convex set and* $x^* \in argmin_{x \in K} f(x)$ *then* $\forall y \in K$*:*
$\nabla f(x^*)^T (y - x^*) \geq 0$

Now we proceed to the presentation of the gradient descent algorithm. The idea behind the simplest algorithm in optimization is that whenever I want to minimize some function f, a good idea is to take a step towards the direction of the steepest descent of his value. Obviously the magnitude of the step must be carefully selected to permit a good convergence. We will

focus also on these details later.

Algorithm 3 gradient descent algorithm

1: **Input** f, T, initial point $x_1 \in K, \{\eta_t\}$
2: **for** $t = 1 \quad to \quad T$ **do**
3: $\quad y_{t+1} = x_t - \eta_t \nabla f(x_t)$
4: $\quad x_{t+1} = \Pi_K(y_{t+1})$
5: return $x_{avg} = \frac{\sum_{t=1}^{T} x_t}{T}$

Theorem 4.2.3. *Convergence of gradient descent Let f be a convex function on a convex set K.If D is an upper bound on the diameter of K and G an upper bound of the norm of the gradient of f on K. Then by selecting* $\eta_t = \frac{D}{G}\frac{1}{\sqrt{t}}$. *We have that after T iterations:*

$f(x_{avg}) - argmin_{x\in K} f(x) \leq \frac{3DG}{2\sqrt{T}}$

Proof.

Let $x^* \in argmin_{x\in K} f(x)$
From jensen inequality we have that:

$$f(x_{avg}) - f(x^*) = f(\sum_{t=1}^{T} \frac{x_t}{T}) - f(x^*) \leq \sum_{t=1}^{T} \frac{f(x_t) - f(x^*)}{T} \tag{4.4}$$

Let $h_t = f(x_t) - f(x^*)$
From the convexity of the function we have that:

$$h_t \leq \nabla f(x_t)^T (x_t - x^*) \tag{4.5}$$

We know proceed by bounding the RHS of the last equation:

$$\|x_{t+1} - x^*\|^2 \leq \|x_t - \eta_t - x^*\|^2 = \|x_t - x^*\|^2 + \eta_t^2 \|\nabla f(x_t)\|^2 - 2\eta_t \nabla f(x_t)^T (x_t - x^*) \Rightarrow$$

$$\Rightarrow 2\nabla f(x_t)(x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f(x_t)^2\| \Rightarrow$$

$$\Rightarrow 2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T} \{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f(x_t)^2\|\} \xrightarrow[\frac{1}{\eta_0}=0]{\eta_t = \frac{D}{G\sqrt{t}}}$$

$$2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T} \{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f(x_t)^2\|\} \leq \sum_{t=1}^{T} \|x_t - x^*\|^2 (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) +$$

$$\sum_{t=1}^{T} \eta_t \|\nabla f(x_t)^2\| \leq D^2 \sum_{t=1}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + G^2 \sum_{t=1}^{T} \eta_t = D^2 \frac{1}{\eta_T} + G^2 \sum_{t=1}^{T} \eta_t =$$

$$= DG\sqrt{T} + DG \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 3DG\sqrt{T}$$

from equations 4.4 and 4.5 we get that:

$f(x_{avg}) - f(x^*) \leq \frac{3DG}{\sqrt{T}}$

□

4.3 Online Convex Optimization

4.3.1 The Online Convex Optimization model

The OCO (Online Convex Optimization) model tries to face the problem of optimizing a sequence of choices rather than find a single best choice. Intuitively, for non static problems, when new information is revealed as the process goes on, this is what we need. Classic methods of algorithmic game theory and mathematical optimization are not robust to the uncertainty of the environment. Therefore we need algorithms that gain experience from the past and perform well under each circumstance of the future.

In contrast with the Multiplicative Weights Update algorithm the main difference is that the algorithms proposed in these section does not maintain a distribution over a limited space of actions but work in a convex set and therefore in continuous space.

The OCO framework ,as proposed at [18], can be structured as a repeated game.

Let K be the convex set of actions available to the online player and F be a bounded (or somehow structured) set of cost functions available to the adversary.

at each iteration t:

1. the online player choose to perform an action $x_t \in K$
2. a convex function $f_t \in F : K \to \text{Re}$ is revealed
3. the online player suffers a cost of $f_t(x_t)$

After defining the general framework, we must define quantities to evaluate the performance of an algorithm that selects the actions of the step (1) of our framework. It comes up that the appropriate performance metric is called regret and it is the difference between the cost that our algorithm suffers and the cost that we would have suffered if we had play the best fixed action throughout the game.

Let's define formally the regret metric.

Definition 4.3.1. *Let Alg be the algorithm for our OCO model, which maps a particular game history to an action. The regret of Alg after T iterations is defined as:*

$$regret_T(Alg) = sup_{\{f_1,f_2,...,f_T\}\subseteq F}\{\sum_{t=1}^{T} f_t(x_t) - min_{x\in K}\sum_{t=1}^{T} f_t(x)\}$$

It is important to underline the width of problems that the OCO framework covers. Recall the Multiplicative Weights Updates algorithm. By defining K as the n-dimensional simplex and $f_t(x_t) = M(x_t, f_t)$ it is clear that the problem it solves is just a special case of the general OCO model. In addition to that, going back to the 4.1.1.1 it is clear that the regret achieved by the Multiplicative Weights Updates algorithm is bounded by $\sqrt{T}$.

4.3.2 Online Gradient Descent

As the name suggests, the simplest algorithm in the OCO framework is the online version of the standard gradient descent. It obviously follows the idea of the simple offline gradient descent and it was first proposed by Zinkevich at [19]

In the offline version each new point was updated following the direction of the negative gradient of the function at the previous point. A generalization of this idea implies that our new point should follow the negative gradient of the cost function revealed at iteration t at the previous selected point.

The surprising thing is that however the functions may be completely different from time to time (so it makes no sense to follow the negative gradient of the previous function), the regret achieved is $O(\sqrt{T})$.

Let's present the Online Gradient Descent algorithm:

Algorithm 4 online gradient descent algorithm (OGD)

1: **Input:** a convex set K, T, initial point $x_1 \in K, \{\eta_t\}$
2: **for** $t = 1$ *to* T **do**
3: $\quad y_{t+1} = x_t - \eta_t \nabla f_t(x_t)$
4: $\quad x_{t+1} = \Pi_K(y_{t+1})$

Theorem 4.3.2. *Regret of online gradient descent algorithm If D is an upper bound on the diameter of K and G an upper bound of the norm of the gradients of evry function in F. Then by selecting* $\eta_t = \frac{D}{G}\frac{1}{\sqrt{t}}$. *We have that*

after T iterations:

$regret_T(OGD) \leq \frac{3}{2}DG\sqrt{T}$

Proof. The proof is almost identical to the proof of the convergence of the simple gradient descent algorithm. Nevertheless we present it for completeness.

Let $f_1, f_2, ..., f_T \in F$ be the cost functions that by selecting them our adversary , creates our biggest regret.

Let $x^* \in argmin_{x \in K} \sum_{t=1}^{T} f_t(x)$

$regret_T(OGD) = \sum_{t=1}^{T}(f_t(x_t) - f_t(x^*))$

Let $h_t = f_t(x_t) - f_t(x^*)$

From the convexity of the functions in F we have that:

$$h_t \leq \nabla f_t(x_t)^T(x_t - x^*) \tag{4.6}$$

We know proceed by bounding the RHS of the last equation:

$\|x_{t+1} - x^*\|^2 \leq \|x_t - \eta_t - x^*\|^2 = \|x_t - x^*\|^2 + \eta_t^2\|\nabla f_t(x_t)\|^2 - 2\eta_t \nabla f_t(x_t)^T(x_t - x^*) \Rightarrow$

$\Rightarrow 2\nabla f_t(x_t)(x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t\|\nabla f_t(x_t)^2\| \Rightarrow$

$\Rightarrow 2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T}\{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t\|\nabla f_t(x_t)^2\|\} \xrightarrow[\frac{1}{\eta_0}=0]{\eta_t = \frac{D}{G\sqrt{t}}}$

$2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T}\{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t\|\nabla f_t(x_t)^2\|\} \leq \sum_{t=1}^{T}\|x_t - x^*\|^2(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) +$

$\sum_{t=1}^{T}\eta_t\|\nabla f_t(x_t)^2\| \leq D^2\sum_{t=1}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + G^2\sum_{t=1}^{T}\eta_t = D^2\frac{1}{\eta_T} + G^2\sum_{t=1}^{T}\eta_t =$

$= DG\sqrt{T} + DG\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 3DG\sqrt{T}$

finally from equation 4.6 we get that:

$regret_T(OGD) \leq \frac{3DG\sqrt{T}}{2}$

□

Now it is useful to underline how this algorithm should be used by a bidder. In contrast with the Multiplicative Weights Update algorithm the above algorithm achieves something greater. It minimizes regret in a continuous space. A bidder who wish to create a bidding algorithm when using the MWU algorithms should first divide its bidding space and then running the MWU algorithm with the selected actions (i.e. bid).

In contrast, when using the OGD algorithm two different approaches are possible. The first one is to divide the bidding space in n actions and then run the algorithm on the n-dimensional simplex. Although this is a good tactic it is pellucid that the algorithm evolution mimics the MWU idea by maintaining a probability distribution over a set of actions and continuously moving into the simplex to guarantee a better performance. The other option is to run the OGD in the support of the bidding space. Of course this is just a one dimensional line and it is convex. However the problem here is that the gradients are unbounded and a direct implementation is infeasible. An idea to use this approach would be to create a smooth version of the Utility function and by this provide the algorithm with a bounded family of gradients. An other possibility would be to update our bidding not in every iteration but less frequently. This will create Utilitity functions that are possibly continuous and differentiable.

4.3.3 Lower Bounds in the OCO model

After presenting the basic algorithm for online convex optimization we proved a bound of $O(\sqrt{T})$ on regret. So here an interesting question arise. In the general case, what is the best we can hope for? The next theorem proves that in fact $O(\sqrt{T})$ is a lower bound on the best possible achieved regret in the most general setting of any OCO algorithm.

Theorem 4.3.3. *Any algorithm in the OCO model suffers in the general case a regret of* $\Omega(DG\sqrt{T})$

Proof.
Let K be the n-dimensional hypercube, $K = \{x \in Re^n, \|x\|_\infty \leq 1\}$
Let $f_v(x) = v^T x$ be the form of the 2^n linear cost functions, one for each vertex of K.

Firstly, notice that D and G are bounded:

$D = \sqrt{\sum_{i=1}^{n} 2^2} = 2\sqrt{n}, \quad G = \sqrt{\sum_{i=1}^{n} 1^2} = \sqrt{n}$

Let the adversary choose at each iteration a cost function uniformly at random from the 2^n available.

By independence, at each iteration we have that:

$$E_{v_t}\left[f_t(x_t)\right] = E_{v_t}\left[v_t x_t\right] = 0$$

Now, let x^* be the best fixed action for a particular sequence of functions, we have that:

$$E_{v_1,v_2,\ldots,v_t}\left[\sum_{t=1}^{T} f_t(x^*)\right] = E_{v_1,v_2,\ldots,v_t}\left[\sum_{t=1}^{T}\sum_{i=1}^{n} v_t(i)x^*(i)\right] = E_{v_1,v_2,\ldots,v_t}\left[\sum_{i=1}^{n}\sum_{t=1}^{T} v_t(i)x^*(i)\right] =$$
$$E_{v_1,v_2,\ldots,v_t}\left[-|\sum_{i=1}^{n}\sum_{t=1}^{T} v_t(i)|\right] = E_{v_1}\left[-n|\sum_{t=1}^{T} v_t(1)|\right] = = -n\Omega(\sqrt{T})$$

The last equality comes from the fact that the sum involved represents the point of a random walk after T steps.

□

Remarks: At this point it is important to underline two important facts. First the simple online gradient descent algorithm achieves the best possible regret in the general setting. Secondly in the last proof there is a paradox. If we had to prove the lower bound from scratch maybe the adversary we would have constructed would not be a stationary distribution. How is possible that a stationary distribution obtains the worst regret possible. The response to this question comes form the difference between the quantity of regret and the loss that our algorithm suffers. It is obvious that an adversary who uses prior knowledge would impose more loss to our algorithm, however in the same time the performance of the best fixed action would be bad, here is explained the paradox.

Now we know that the best possible tactic for a bidder is to use no regret algorithms as in a general setting they assure almost an optimal regret bound.

4.4 Bandit Convex Optimization

4.4.1 The Bandit Convex Optimization Model

Since now, we have presented the Multiplicative Weights Updates and the Hedge algorithm. Then by generalizing the setting we have derived the online gradient descent algorithm. All these algorithms have one common characteristic, they require a full information model. In fact the Multiplicative Weights Updates and the Hedge algorithm perform a weights updates procedure that requires the knowledge of the loss function in the entire action space. The online gradient descent requires the knowledge of the gradient of the loss function which is also a strong requirement to fulfil .

In many scenarios this is quite unrealistic. Imagine for example the routing problem in an unknown network, we select a path and then we suffer the time that our data needs to arrive in the desired destination, nobody tells us what we would had suffered if we had selected a different path. The only feedback available to us, is the loss we suffered from the particular action that we chose.

In our problem (i.e. the GSP auction) the bidders do not know neither the allocation function nor the particular position coefficients of the particular GSP auction they are participating. In this setting the bidders place a bid and then they get the pay off of their bid. However they do not know how much the pay off would have been if they had placed an other bid. Obviously not knowing the performance of its different bid does not permit the classic implementation of the MWU algorithm.

In order to cover these types of settings we introduce the Bandit Convex Optimization Model.

The BCO framework ,as proposed at [**?**] , can be structured as a repeated game.

Let K be the convex set of actions available to the online player and F be a bounded (or somehow structured) set of cost functions available to the adversary.

at each iteration t:

1. the online player choose to perform an action $x_t \in K$
2. the online player suffers a cost of $f_t(x_t)$

As an attentive reader may notice, the framework is similar to the one in the OCO model. However the adversary doesn't reveal the cost function to the player. By doing this the only feedback available is the loss suffered by the online player.

The quantity which will help us to evaluate the performance of our algorithms is as always the regret:

Definition 4.4.1. *Let Alg be the algorithm for our BCO model, which maps a particular game history to an action. The regret of Alg after T iterations is defined as:*

$$regret_T(Alg) = sup_{\{f_1,f_2,...,f_T\}\subseteq F}\{\sum_{t=1}^{T} f_t(x_t) - min_{x\in K}\sum_{t=1}^{T} f_t(x)\}$$

Remark: However the problems modelled become apparently more difficult. The regret follows the same definition.

4.4.2 Multi Armed Bandit (MAB) model

In this section we will introduce the most simple model of bandit optimization. Which is just a subcategory of the more general BCO model. This model has been introduced by Robbins at [20]

Let's describe the MAB model. As always here we are with the repeated game:

at each iteration t:

1. the online player choose to perform an action $i_t \in \{1, 2, ...n\}$
2. the online player suffers a cost of $l_t(i_t) \in [0, 1]$

This simple repeated game, remember us the problem that the Multiplicative Weight Updates solves. Actually it is just the bandit analogue of this problem. Notice that here any algorithm which try to figure out what action should choose in the next iteration doesn't know the previous loss functions, but only their values.

The problems modelled are just specific problems of the more general BCO model. To understand this we will transform the MAB model into a BCO problem.

1. Take the action set from $\{1, 2, ..., n\}$ to be the set of distributions over these n actions. So $K = \Delta^n$ is the n-dimensional simplex.

2. Take the loss functions to be $f_t(x_t) = \sum_{i=1}^{n} l_t(i)x_t(i)$ (i.e the expected loss of our distribution)

And the equivalence follows easily.

4.4.3 MAB algorithms

In this section we will introduce two basics algorithms that achieve low regret in the MAB model

First of all, let's discuss what properties an algorithm should have in order to tackle with these types of problems. Here, every time we pick an action, we suffer it's loss, so we have no clues about the performance of the other actions in this particular iteration. Here comes the first property that a MAB algorithm should have. It has to efficiently explore the action space, meaning that it should try different possible actions in order to get a clue about how they perform on average.

Secondly, since the MAB algorithm is basically a prediction algorithm it has from the accumulated experience to give us a prediction about which action will perform well in the next iteration. This is the exploitation step.

To summarize, MAB algorithms must do two things. Explore the action space and from their exploration predict good action in the future.

Now we will present the simplest MAB algorithm template as proposed by Hazan at [18]. It separates completely the exploration step with the exploitation step. So at each iteration with some probability explores the actions space, meaning that it wastes some loss to acquire useful information and with the rest of the probability performs the exploitation step that consists on predicting a good action.

Algorithm 5 simple MAB algorithm

Input: OCO algorithm A, parameter δ
for $t = 1 \quad to \quad T$ **do**
  Let b_t be a Bernoulli variable,with $P(b_t = 1) = \delta$
  if $b_t = 1$ **then**
    choose $i_t \sim$ uniformly at random from $\{1, 2, ...n\}$
    play i_t
    for $i = 1 \quad to \quad n$ **do**
      $\hat{l}_t(i) = \frac{n}{\delta} l_t(i) \mathbb{1}\{i = i_t\}$
    Let $\hat{f}_t(x) = \hat{l}_t^T x$
    $x_{t+1} = A(\hat{f}_1, \hat{f}_2, ..., \hat{f}_t)$
  else if $b_t = 0$ **then**
    choose $i_t \sim x_t$
    play i_t
    $\hat{f}_t(x) = \mathbf{0}$
    $x_{t+1} = A(\hat{f}_1, \hat{f}_2, ..., \hat{f}_t)$

Theorem 4.4.2. *Let A be the online gradient descent algorithm and* $\delta = \sqrt{n}T^{-\frac{1}{4}}$ *then,*
$E\left[\sum_{t=1}^{T} l_t(i_t) - min_i \sum_{t=1}^{T} l_t(i)\right] \leq O(T^{\frac{3}{4}}\sqrt{n})$

Proof.
Before proceeding to more technical details, notice that:

1. for a constant action $i \in \{1, 2, ..., n\}$, we have that:
   $E\left[\hat{l}_t(i)\right] = Pr\left[b_t = 1\right] Pr\left[i_t = i | b_t = 1\right] \frac{n}{\delta} l_t(i) = l_t(i)$
   so the estimation of the loss vector is unbiased
2. $\|\hat{l}_t\|^2 \leq \frac{n}{\delta}$
3. for a constant distribution over actions, we have that:
   $E\left[\hat{f}_t(x)\right] = E\left[\sum_{i=1}^{n} \hat{l}_t(i) x(i)\right] = \sum_{i=1}^{n} x(i) E\left[\hat{l}_t(i)\right] = \sum_{i=1}^{n} x(i) l_t(i) = f_t(x)$

Now we are ready to proceed to the main part of the proof
Let $S_T \subseteq [T]$ be those iterations in which our algorithm explores the action space (i.e. $b_t = 1$), then:

$$E\left[regret_T\right] = = E\left[\sum_{t=1}^{T} f_t(x_t) - min_{x\in\Delta^n}\sum_{t=1}^{T} f_t(x)\right] =$$
$$= E\left[\sum_{t=1}^{T} l_t(i_t) - min_i \sum_{t=1}^{T} l_t(i)\right] =$$

let i^* achieve the minimum

$$= E\left[\sum_{t=1}^{T} l_t(i_t) - \sum_{t=1}^{T} l_t(i^*)\right] =$$
$$= E\left[(\sum_{t\notin S_T} l_t(i_t) - \sum_{t\notin S_T} l_t(i^*)) + (\sum_{t\in S_T} l_t(i_t) - \sum_{t\in S_T} l_t(i^*))\right] \leq$$
$$\leq E\left[\sum_{t\notin S_T} l_t(i_t) - \sum_{t\notin S_T} l_t(i^*) + \sum_{t\in S_T} 1\right] =$$

i^* is independent from t

$$= E\left[\sum_{t\notin S_T} l_t(i_t) - \sum_{t\notin S_T} \hat{l}_t(i^*) + \sum_{t\in S_T} 1\right] \leq$$

$$l_t(i_t) \leq \hat{l}_t(i_t) \quad \forall t$$

$$\leq E\left[\sum_{t\notin S_T} \hat{l}_t(i_t) - \sum_{t\notin S_T} \hat{l}_t(i^*) + \sum_{t\in S_T} 1\right] \leq$$

Since A is the online gradient descent algorithm we have that, it achieves a regret of $O(\frac{3}{2}GD\sqrt{T})$

$$\leq \tfrac{3}{2}GD\sqrt{T} + \delta T \leq$$

$$D \leq 2, G \leq \tfrac{n}{\delta}, \delta = \sqrt{n}T^{-\frac{1}{4}}$$

$\leq 4T^{\frac{3}{4}}\sqrt{n}$, which concludes the proof $\square$

Remarks: Although this is a very simple algorithm, his main idea is very powerful. The idea is to create unbiased estimators of the loss functions while interchanging between exploration and exploitation steps. It turns out that using this idea it is possible to create a general method to turn OCO algorithms into BCO algorithms. In addition to that it is important to point an other property of this simple algorithm. It achieves sublinear regret, so the average regret vanishes as $O(\frac{1}{\sqrt{T}})$, a property will be useful throughout this thesis.

Now we will proceed by presenting an other MAB algorithm. The algorithm is called EXP3 algorithm and was designed by Cesa-Bianchi et al at [21] . EXP3 stands for Exponential-weight algorithm for Exploration and Exploitation. His main difference with the simple MAB algorithm is that it combines the two steps (exploration and exploitation) achieving in this manner better regret bounds.

It is remarkable how such a simple algorithm achieves near optimal regret bounds. It's time to present it:

Algorithm 6 EXP3 algorihtm

Input: learning parameter ϵ
$x_1 = \frac{1}{n}\mathbf{1}$
for $t = 1 \quad to \quad T$ **do**
  $i_t \sim x_t$
  play i_t
  for $i = 1 \quad to \quad n$ **do**
    $\hat{l_t}(i) = \frac{l_t(i)}{x_t(i_t)}\mathbb{1}\{i = i_t\}$
    $y_{t+1}(i) = x_t(i)e^{-\epsilon\hat{l_t}(i)}$
  $x_{t+1} = \frac{y_{t+1}}{\|y_{t+1}\|_1}$ ▷ Normalize to create a distribution

Theorem 4.4.3. *EXP3 algorithm with losses in the* $[0, 1]$ *range and* $\epsilon = \sqrt{\frac{ln(n)}{Tn}}$ *guarantees the following regret upper bound:*
$E\left[\sum_{t=1}^{T} l_t(i_t) - min_i \sum_{t=1}^{T} l_t(i)\right[\leq 2\sqrt{Tnln(n)}$

Proof.
First of all, let's notice that:

1. for a constant action $i \in \{1, 2, ..., n\}$, we have that:
   $E\left[\hat{l_t}(i)\right] = Pr\left[i_t = i\right]\frac{l_t(i)}{x_t(i)} = l_t(i)$
   so the estimation of the loss vector is unbiased
2. $E\left[x_t^T l_t^2\right] = \sum_{i=1}^{n} x_t(i)^2 l_t(i)^2 \leq n$

3. so as before, for a constant distribution over actions, we have that:
$E\left[\hat{f}_t(x)\right] = E\left[\sum_{i=1}^{n} \hat{l}_t(i)x(i)\right] = \sum_{i=1}^{n} x(i)E\left[\hat{l}_t(i)\right] = \sum_{i=1}^{n} x(i)l_t(i) = f_t(x)$

Before proceeding to the technical part of the proof, let's underline an important correlation. The update rule of the EXP3 algorithm is identical with the update rule of the Hedge algorithm. In fact, EXP3 is just the application of the Hedge algorithms to the sequence of loss functions $\hat{l_1}, \hat{l_2}, ..., \hat{l_t}$, that represents an unbiased estimation of the true loss functions $l_1, l_2, ..., l_t$. So from the Hedge algorithm theorem we have an important relation:
$\forall i^* \in \{1, 2, ..n\}$
$E\left[\sum_{t=1}^{T} \hat{l}_t(i_t) - \sum_{t=1}^{T} \hat{l}_t(i^*)\right] \leq E\left[\epsilon \sum_{t=1}^{T}\sum_{i=1}^{n} \hat{l}_t(i^*)^2 x_t(i^*) + \frac{ln(n)}{\epsilon}\right]$

$$E\left[regret_T\right] = = E\left[\sum_{t=1}^{T} l_t(i_t) - min_i \sum_{t=1}^{T} l_t(i)\right] =$$

let i^* achieve the minimum

$$= E\left[\sum_{t=1}^{T} l_t(i_t) - \sum_{t=1}^{T} l_t(i^*)\right] =$$

i^* is independent from t

$$= E\left[\sum_{t=1}^{T} l_t(i_t) - \sum_{t=1}^{T} \hat{l}_t(i^*)\right] \leq$$

$l_t(i_t) \leq \hat{l}_t(i_t) \quad \forall t$

$$\leq E\left[\sum_{t=1}^{T} \hat{l}_t(i_t) - \sum_{t=1}^{T} \hat{l}_t(i^*)\right] \leq$$

From the Hedge theorem regret bound

$$\leq E\left[\epsilon \sum_{t=1}^{T}\sum_{i=1}^{n} \hat{l}_t(i^*)^2 x_t(i^*) + \frac{ln(n)}{\epsilon}\right]$$

Since the loss functions are bounded in the $[0, 1]$ region

$$\leq \epsilon nT + \frac{ln(n)}{\epsilon}$$

finally, by substituting ϵ we get the desired bound

□

Chapter 5

Valuation Inference in Online Ad Auctions

In this chapter we will study and evaluate the new valuation inference method proposed lately by Eva Tardos, Vasilis Syrgkanis and Denis Nekipelov at [4] in the context of GSP auctions. Moreover we will describe the simulation system constructed to evaluate these methods, its advantages and its future development possibilities.

In addition to that, we will study the robustness of their valuation model under Non-Truthful mechanisms as a continuation of the robustness study of Dutting, Parkes and Fischer at [5].

5.1 Problem Statement

The problem of inferencing the players valuations in auctions is very important because knowing players valuations permits an accurate study of their bidding behaviour and consequently the evaluation of the predictions made by theory. In our setting the inferencer is provided by the following:

- the number of items sold
- the maximum number of players in every iteration
- the players importance coefficients
- the value of every item sold
- the reserve price
- the complete auction's bidding history
- the type of auction implemented

The goal is to conclude the private valuation per unit of item that each player has.

We will concentrate our study to the GSP auction implementation. Thus the item sold are the advertising positions, their value are the respective position coefficients, the importance of each player is the click coefficients etc. Obviously the allocation and cost functions are the one described previously in 3.2 .

Before proceeding to the technicalities it is useful to describe the idea behind this model. The standard econometric approach to deal with environments when strategic agents interact is to assume that somehow players have reached a mixed nash equilibrium, thus they are best responding to the distribution they are facing [3]. Other methods rely on the assumption that players will reach a particular equilibrium characterized by bidding functions of their valuations. Through an approximation of the bidding distribution is possible to inverse the particular functions and conclude the valuation distribution [1]. Also methods conclude a particular tuple of valuations which best explain the approximate equilibrium reached [2]. A different approach which is pervasive to the auction is described at [22] .

The new method's objective is to describe rapidly changing environments as the one of online auctions in a more accurate way. Online auctions are rapidly changing over time as new bidders appear and disappear, thus the agents try to adequate their behaviour to the adversities faced, constantly. Furthermore an other flaw of these method relies on the assumption that bidders make it to reach a mixed nash equilibrium. The problem here is not so much relevant from a complexity theory point of view (because auctions are relatively simple interaction mechanisms) as it is for an information theoretic aspect. Bidders do not interact under full information environments and consequently the amount of information needed to reach such an equilibrium is maybe too large to assert such an assumption. All these observations lead us to weaken the assumption about what players achieve.

The new inferencing method suppose that bidders are learning agents in a constantly changing environment. Thus they use learning algorithms to decide what is the best strategy they can follow every time. This assumption makes us suppose that bidders, if they want to maximize their utility over time have to use no-regret algorithms. There are no-regret algorithms that obtain the best possible regret bounds and it is logic to assume that bidders will use such algorithms. Moreover as we previously presented, these algorithms are quite simple to implement, so our assumption is unequivocally strengthen by this observation.

5.2 Inference Method

We will proceed by describing the method in detail.

Each player has a strategy space B_i. Thus a bid profile $b \in B_1 \times B_2 \times ... \times B_n$. Furthermore we will denote by b^t the bid profile at the t-th iteration. Assuming that each learning player has a regret of ϵ_i and a valuation of v_i we have that:

$$\forall b' \in B_i \quad \frac{1}{T}\sum_{t=1}^{T} U_i(b^t, v_i) \geq \frac{1}{T}\sum_{t=1}^{T} U_i(b', b^t_{-i}, v_i) - \epsilon_i \tag{5.1}$$

Definition 5.2.1. *(Rationalizable set) A pair (ϵ_i, v_i) of valuation v_i and regret ϵ_i is rationalizable if it satisfies the previous equation. The set of such pairs is called the rationalizable set of player i and is denoted by NR.*

Now we will analyse the properties of the rationalizable set NR. We suppose that $B_i \subseteq \mathrm{Re}_+$. Also from the quasilinear model we have that:

$$\forall b' \in B_i \quad \frac{1}{T}\sum_{t=1}^{T}(v_i P_i(b^t) - C_i(b^t)) \geq \frac{1}{T}\sum_{t=1}^{T}(v_i P_i(b', b^t_{-i}) - C_i(b', b^t_{-i})) - \epsilon_i \tag{5.2}$$

By denoting as:

$$\Delta P(b') = \frac{1}{T}\sum_{t=1}^{T}\left(P_i(b', b^t_{-i}) - P_i(b^t)\right) \tag{5.3}$$

The average increase in the click probability if a player chooses to bid constantly b'.

$$\Delta C(b') = \frac{1}{T}\sum_{t=1}^{T}\left(C_i(b', b^t_{-i}) - C_i(b^t)\right) \tag{5.4}$$

The average increase in the cost that a player suffers if he chooses to bid constantly b'.

Hence the starting equation turns to be equivalent to the:

$$\forall b' \in B_i \quad v_i \Delta P(b') - \Delta C(b') \leq \epsilon_i \tag{5.5}$$

Lemma 5.2.2. *The rationilizable set for each player is a closed convex set*

Proof Each inequality is linear in the valuations and the regret. Thus it divides the space in halfspaces. The intersection of these halfspaces is obviously a convex set. Moreover the inequalities are not strict. So the set is closed.

It can be proven that under additional logical assumptions the rationalizable has other useful properties that permit to approximately calculate it efficiently.To mention them briefly we will underline that this particular convex set is fully determined by the functions $\Delta P(\cdot)$ and $\Delta C(\cdot)$ as they determine the support function of the convex set.The entire analysis of the inference method is given by Syrgkanis, Nekipelov and Tardos at [4].

Now the inference method is quite obvious. Our main goal is to find the valuation of each player that best explains his bids over time.

By best explain we will consider the valuation that is connected with the minimum possible regret.Obviously we will restrict ourselves to valuations that are strictly positive.In addition to that we will consider B_i to be a discretized bounded space which is a subset of a set $[0, b_{max}]$.

By the previous if we set $B_i = \{0, 2e, 3e, ..., b_{max}\}$. Then the inference method can be described as a linear program:

$$\begin{aligned} &minimize \quad \epsilon + 0 \cdot v \\ &subject \quad to \quad v\Delta P(b') - \Delta C(b') \leq \epsilon \quad \forall b' \in B_i \end{aligned} \tag{5.6}$$

The results of this inference method described above suffer from a serious pathology. Smaller additive regrets ϵ tend to be best explained by smaller valuations than the real ones. It turns out that better inference results are achieved by assuming that the learning algorithm used by each agent succeeds in pursuing the smallest possible multiplicative regret.

So now, we slightly modify the initial equation and we get that a multiplicative regret of δ is achieved under the valuation v if:

$$\forall b' \in B_i \quad \frac{1}{T}\sum_{t=1}^{T} U_i(b^t, v_i) \geq (1-\delta)\frac{1}{T}\sum_{t=1}^{T} U_i(b', b^t_{-i}, v_i) \tag{5.7}$$

By defining the average click probability and the average cost that player i suffers under his actual bid sequence as P_0 and C_0 we get that:

$$\forall b' \in B_i \quad v\Delta P(b') \leq \Delta C(b') + \frac{\delta}{1-\delta}(vP_0 - C_0) \tag{5.8}$$

Our aim now is to minimize the multiplicative regret δ. This modification drops out the pathologies of the inference method that are due to the fact that small valuations explain better small additive regrets. Let's explain why.

First notice that a multiplicative regret of δ corresponds to an additive regret of $\frac{\delta}{1-\delta}(vP_0 - C_0)$. Supposing that δ lies in the $[0, 1]$ range we have that the function $f(\delta) = \frac{\delta}{1-\delta}$ is increasing. So the minimization of δ is equivalent to the minimization of $\frac{\delta}{1-\delta}$.In addition to that notice that $\frac{\epsilon}{(vP_0 - C_0)} = \frac{\delta}{1-\delta}$. This means that bigger valuations are now more advantageous than smaller ones. To sum up due to some pathologies we prefer to use the multiplicative regret minimization inference method than the additive one.

Now, the only question that remains to be answered is how we should implement this inference method. First note that if we replace the $\frac{\delta}{1-\delta}$ term with a κ then each constraint is a quadratic convex function. So the intersection of the constraints obviously forms a convex set.Furthermore since minimizing δ is the same as minimizing $\frac{\delta}{1-\delta}$ we conclude that the inference method can be written as a convex program. So the minimization methods that are applicable in general in the convex optimization setting can obviously be used in this particular problem.

$$\begin{aligned} &minimize \quad \kappa + 0 \cdot v \\ &subject \quad to \quad v\Delta P(b') \leq \Delta C(b') + \kappa(vP_0 - C_0) \quad \forall b' \in B_i \end{aligned} \tag{5.9}$$

Now, one option is to use convex optimization methods in order to find the valuation that best explains the minimization of the previous problem. An other option is to divide the valuation space and solve each program separately. Obviously now, since the valuation in each instance of the general convex program is constant the general quadratic convex program turns to a bunch of linear programs. Then we select the instance of the valuation that is best explained (i.e best minimizes κ). More formally, supposing that the valuation is bounded in the $[0, v_{max}]$ range. we select $0 = v^1 < v^2 < ... < v^l = v_{max}$ to divide equally in l segments the valuation space. Letting V to be the set of these valuations we formulate the following program for each $v^j \in V$

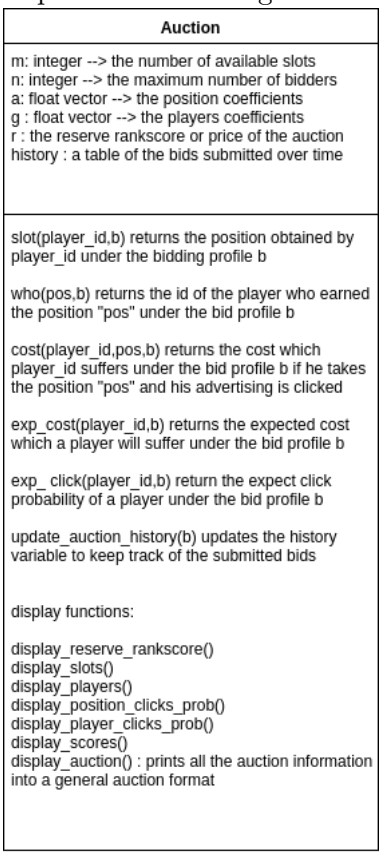
$$\begin{aligned} &minimize \quad \kappa^j \\ &subject \quad to \quad v^j\Delta P(b') \leq \Delta C(b') + \kappa^j(v^j P_0 - C_0) \quad \forall b' \in B_i \end{aligned} \tag{5.10}$$

We must note that this is not a general form linear program as it can be calculated just by looking the intersection of the various inequalities. So, now we predict that the valuation of the particular bidder is the one that achieves the best κ^j.

5.3 System Description

In order to test the valuation inference method proposed. We construct a bidding simulation system which continues to be under construction to enlarge its possibilities. His primary objective is to create an environment suitable to run experiments. The need of such an environment is obvious, as the global community lacks of experimental data that will enable students and researchers to conduct experiments for free without the lenses of a profit organization. The bidding simulator is written in python using object oriented techniques to be easily modified and studied. The code can be found at the following link `https://github.com/andreasr27`

We proceed describing the main classes template:

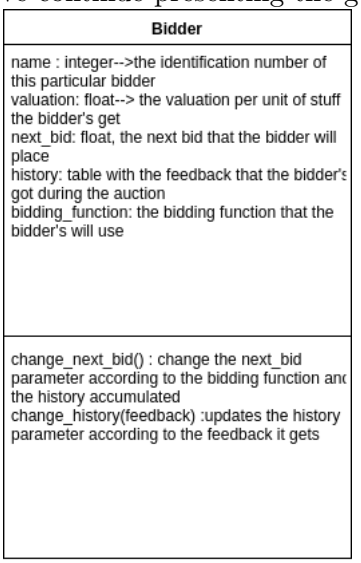


The general class Auction template provides a general template to be implemented analogously to which auction we wish to recreate. To elucidate further this fact we underline how easy it is to implement different auctions just by changing a minimum amount of code. If we wish to recreate the GSP auction then the functions should be implemented according to the model described previously 3.2. If we want to change our experiment to test

some of the properties of the Myerson's lemma derived auction then the only thing that it has to change is the cost function according to the equation 3.3 . If we want to change to a First Price auction then also the only thing to change is the cost function.

Obviously many more features could be studied. As we discuss in the section of 5.7 a dynamic reserve price could be used to augment auctioneer incomes. In fact the history variable of the auction which stores all the bids which were submitted in addition with the valuation inference method could be used under this purpose.

We continue presenting the general template proposed for the bidder.



Now we can vary our study on many aspects. Varying the bidding function parameter to study the effectiveness of different bidding functions. Obviously the bidding functions have to be in concordance with the feedback given to the bidders. By this we can fluctuate between full information and not full information environments. As the feedback could be only the profit that the bidder gets but also the entire bidding vector, which obviously will permit more precise learning algorithms.

An other interesting fact is that the feedback accumulated in the history parameter permits different implementations of the algorithms that exploit them. We could use a myopic ,greedy approach bidding only by responding to the last iterations and thus best responding in a aggressive way or using smooth approaches. In fact it is possible to implement no regret algorithms which update their weights above the actions taking into account only the lasts iterations and omitting the entire auction history.

When the auction is ran it creates an .auction file which constitutes the description of the auction that will be available to the inference methods. In this description the informations that are available are:

- The number of slots
- The maximum number of players
- the position clicks coefficients
- the players click coefficients
- the scores of players (this is a particularity of certain auction which we omitted in the general description but we implemented it in case turned useful for future study)
- the reserve price
- the bid profile of the first iteration
- the bid profile of the second iteration
- etc...

The idea behind the auction format is to provide all the informations that the mechanism will accumulate during the auction and thus study no regret dynamics as if we were the auctioneer.

In addition to that, we created a first version of a library to test no regret dynamics in auctions. The library's name is **regret.py**. The library is provided with the two basic inference valuation approaches, thus the additive regret method and the multiplicative regret method. Furthermore it provides additional functions that could be used in variations of such algorithms. There are functions that evaluate the accumulated regret of a bidder, the average probability of a click through the auction, the average cost etc. The main characteristic and the main objective of the library created is to be the first step toward a more generic and reusable library of functions. We enlighten the fact that the functions created are in concordance with the general auction template created and do not depend on the particular implementation of the auction. So, they could be used also to test the no regret approach in first price auctions and generally every type of repeated dynamic auction.

Finally we created a library of functions that could be used to visualize various metrics of the auctions studied. This is the library **stats.py**. The library at the time contains functions to visualize the rationalizable sets, the average position that a bidder gets, the evolution of the average bid of a particular bidder through the auction, the evolution of the regret of a particular bidder as the auction proceeds etc. We should underline another time that these functions are independent to the particular auction implemented and can be used in many variations.

5.4 Experimental Results

In this section we will describe the system created to evaluate the valuation inference method described in the previous section and then we will present the experimental results.

The system created was as close as possible to the real bidding system used by online ad auctions companies. Especially, the setting of the auction was the one described previously 3.2 and the implementation of the allocation and payment rule was according to the Generalized Second Price auction.

The auction parameters as click probabilities and reserve price are not known to the bidders. The only thing known to the bidders is the type of auction ran, that is the GSP. Thus the only feedback available to the bidders is their profit or loss.

Under this context it is supposed that bidders are learning agents that taking as feedback their profit, understand as the auction is repeated how to bid. The main problem here is that bidders do not know in every auction what the best bid would have been. The only thing that they know is the actual profit of the bid they have submitted. It is lucid that we are in a non full information context and only approaches similar to the ones in the Bandit Convex Optimization framework are likely to succeed.

5.4.0.1 How Bidders Bid

The algorithm chosen to simulate the learning bidders is the EXP3 algorithm [21]. Firstly because of its simplicity which make it particularly attractive and secondly because it achieves almost an optimal regret bound.

However some technicalities needs to be clarified. First of all what is the bidding space for a bidder. As we used the EXP3 algorithm to simulating the bidding behaviour of learning agents it is obvious that a good tactic for a bidder is to equally divide his bidding space and find among all these possible

actions, the sequence that gives him a good regret bound of the profit. In addition to that the support of the bidding space has to be chosen. For the experiments the support of the bidder is considered the $[0, v_i]$ range, where v_i is his valuation. As we are under the GSP model and we are testing an inference method creating various learning bidders, it does not make any sense for a bidder to bid over his valuation as in the majority of cases this behaviour will result to a negative utility. In contrast if the bidder does not overbid then it is assured that his utility will be positive due to payment rule of the GSP auction. In real world scenarios however, this risky tactic could be profitable, overbidding could lead to a negative utility in a first phase of the auction but will also discourage other bidders to bid as their profit will be lesser. Hence it is possible to lead other bidders to leave the auction and finally gain a type of monopole. Obviously experiments have to be rational, so these type of irrational situations must be omitted.

Another fact that need to be elucidated is how the bidders use the EXP3 algorithm into a maximization context. Bidding mostly under his valuation a bidder is assured to have a positive profit. Thus his main objective is to minimize regret of the negative utility.Formally:

$$max_{b' \in B_i} \left(\sum_{t=1}^{T} U_i(b', b_{-i}^t, v_i) - \sum_{t=1}^{T} U_i(b^t, v_i) \right) \tag{5.11}$$

Is equivalent to:

$$min_{b' \in B_i} \left(\sum_{t=1}^{T} \left(-U_i(b', b_{-i}^t, v_i) \right) - \sum_{t=1}^{T} \left(-U_i(b^t, v_i) \right) \right) \tag{5.12}$$

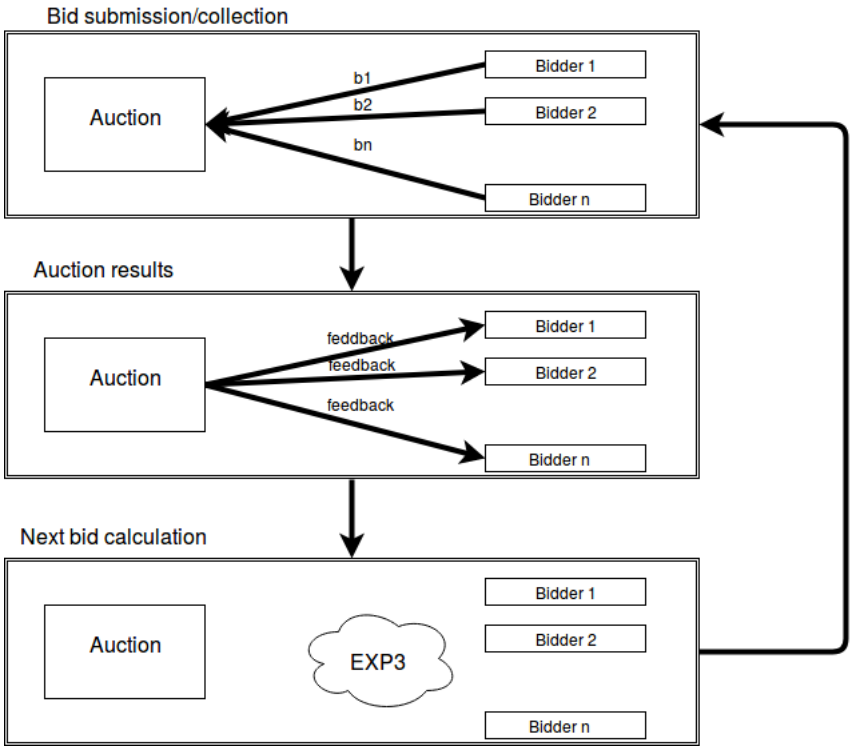
So, as a loss function a bidder should use the negative utility, here two problems arise. First the losses are negative. Applying the multiplicative weights updates idea into non full information settings , as the EXP3 algorithm does, lead to bad convergence behaviour. To elucidate this further we must underline the idea behind the update rule of the MWU algorithm and the one of the EXP3 algorithm. In the MWU update rule we diminish the weight of bad performing actions and we increase the weight of the good ones in each iteration. This is possible because in each moment we know exactly how all the actions are performing, so we have the possibility whenever environment change to adequate to it. In contrast the idea behind the EXP3 algorithm is a little bit different. EXP3 diminish in every iteration the weight of the chosen action.This diminution (for equally probable actions) is bigger when the action has a big loss than when the action has a little loss. Now imagine a problem when the losses can be negative. We will describe a though experiment to make clear the point. One bidder is bidding and has to choose among two possible bids a, b. In the first 100 iterations bid a

has a loss of -10 and b a loss of 0 and for the rest of the experiment action a will have a loss of -1 and action b of -2. Obviously, if the auction iterations are more that 10000, for example 100000 we easily see that the best fixed action is b. However, the sad news are that EXP3 algorithm without attentively choosing the loss function will stabilize to a and will never come back. Following the classic update rule of EXP3 the probability vector will stabilize to a position close to the $(1, 0)$ after a dozen of iterations. So, after these 100 first iterations, in every subsequent iteration a difference of 1 will be accumulated. The algorithm will not overcome and the desired regret bound will not be achieved. To overcome to this difficulty we just use a biased loss to update our algorithm. In fact let u_{max} be the max possible utility then our minimization rule can be modified as :

$$min_{b' \in B_i} \left(\sum_{t=1}^{T} \left(u_{max} - U_i(b', b^t_{-i}, v_i) \right) - \sum_{t=1}^{T} \left(u_{max} - U_i(b^t, v_i) \right) \right) \quad (5.13)$$

The new losses are obviously positive and the minimization problem is totally equivalent, so we solved our first issue. The second issue is due to the fact the EXP3 algorithm pretends losses in the $[0, 1]$ range. This issue is not quite relevant as dividing by a normalizing factor will affect our regrets bounds only by constant factor equal to the max possible loss.Hence the asymptotic vanishing property of the average regret will not be affected.

We present an explanatory diagram of the simulation system:


Bid submission/collection
Auction
b1
b2
bn
Bidder 1
Bidder 2
Bidder n
Auction results
Auction
feddback
feedback
feedback
Bidder 1
Bidder 2
Bidder n
Next bid calculation
Auction
EXP3
Bidder 1
Bidder 2
Bidder n

5.4.0.2 Setting and Results

First of all, we will describe the setting upon which we did our experiments. Although changing the parameters of the experiment does not affect particularly the inference method and the behaviour of the system, for clarity we will present the specific parameters.

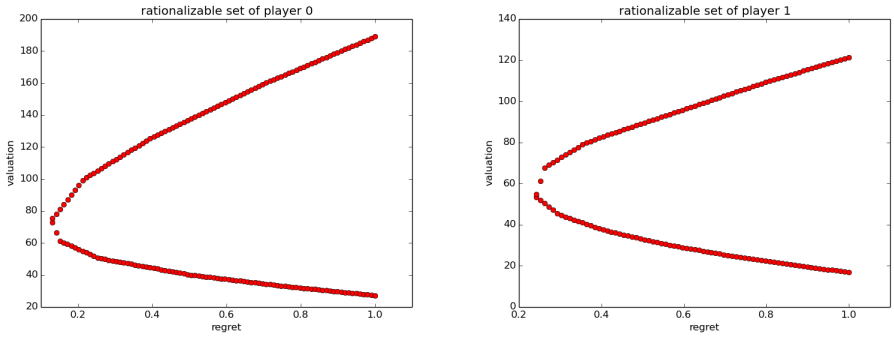
1. number of positions = 5
2. number of players = 6
3. position coefficients $a = [1.0, 0.9, 0.75, 0.55, 0.3]$
4. players coefficients $\gamma = [0.1, 0.08, 0.07, 0.07, 0.06, 0.07]$
5. reserve bid $r = 15$
6. players valuations $v = [72, 61, 53, 46, 39, 33]$ are supposed to be multiple of a penny

In addition to that we must mention that each bidder had in his bidding space twenty different possible actions that divide equally the support of $\left[0, v^i_{max}\right]$.

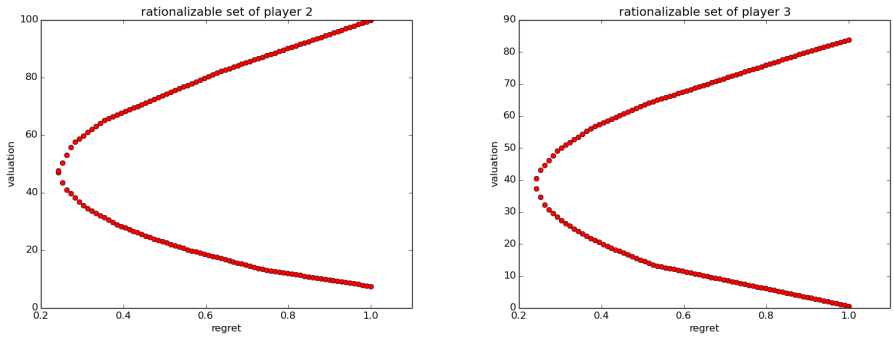
As for the inference method, of course we selected the one that minimizes the multiplicative regret, dividing the possible valuation space into 20 different valuations.

An other important observation we should mention before presenting the diagrams is that the repeated auction ran for a total of T=100,000 iterations. As T increases the valuation inference method performs better because bidders achieve to minimize regret consistently

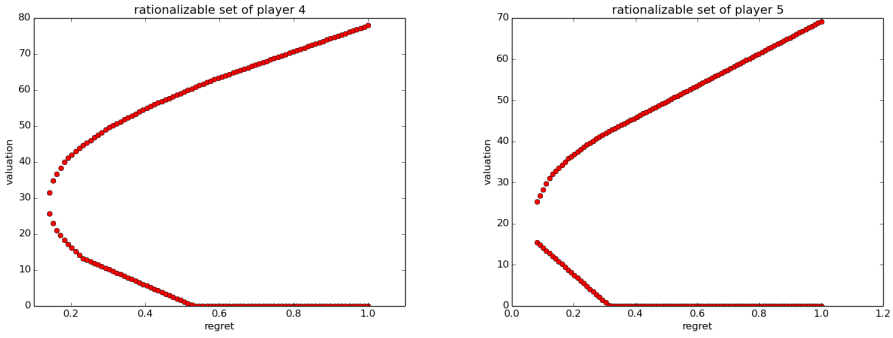
RATIONALIZABLE SETS We present the rationalizable set for each bidder in our experiment. (Definition of rationalizable set 5.2.1)



(a) valuations 72 and 61



(b) valuations 53 and 46

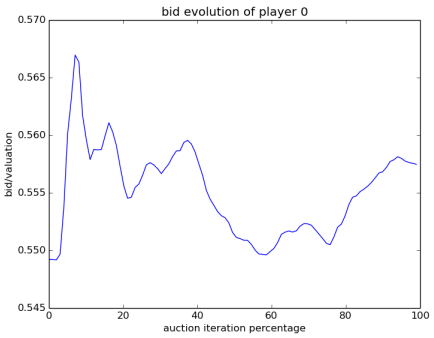


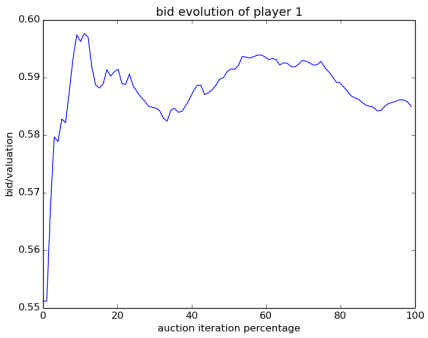
(c) valuations 39 and 33

Observations: First note from the definition of the rationalizable set, the plots concern the additive regret and the valuation of each bidder. Furthermore notice that the left angle of each rationalizable set is very steep. What does this suggest? It suggests that from a particular value of regret and

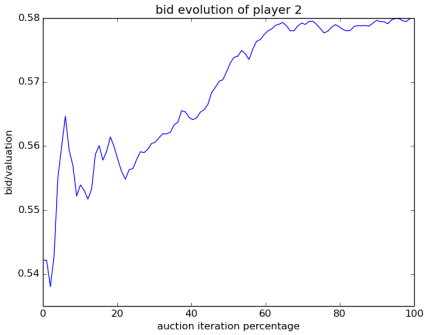
after ,if we want to further minimize regret the only value that we can use to explain it is way lesser. This pathology as previously noted flies away when we use as inference method the one of the multiplicative regret. In fact, using the multiplicative regret we take into account the magnitude of the valuation in such a way that we calculate the trade-off between regret minimization and magnitude of valuation.

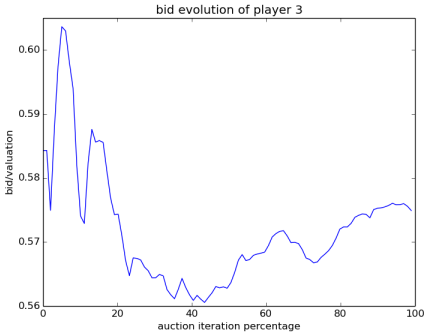
BID EVOLUTION We continue presenting the average bid evolution as a percentage of each bidder's valuation:



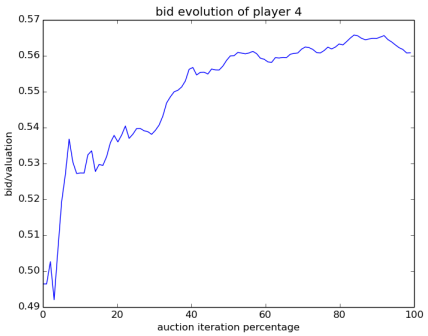


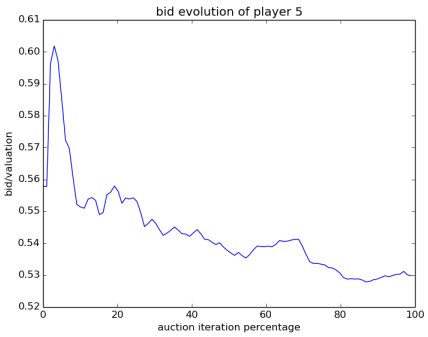
(a) valuations 72 and 61





(b) valuations 53 and 46

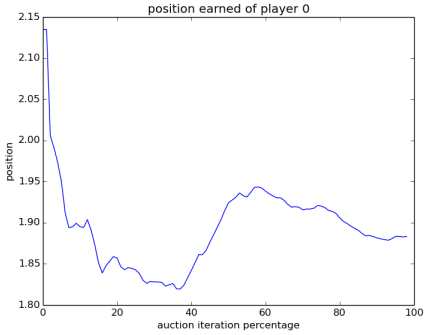


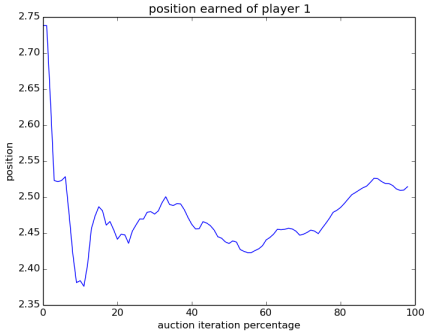


(c) valuations 39 and 33

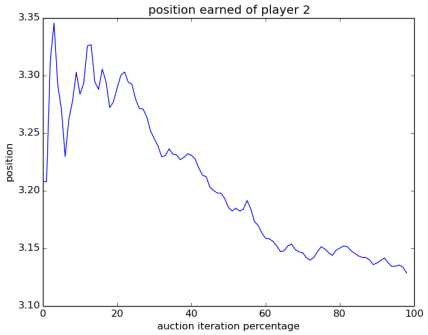
Observations: There are three important observations that need to be enlighten. The first one is that in every bidder two phases of the bidding behaviour are clear. The first one is the learning phase. That is the phase when the bidder has no clue about how he should bid and explores the space. This phase is characterized by an unstable bid. As the auction proceeds the bidder learns which are the bids that perform better and try to concentrate the majority of the probability mass on these bids. This behaviour is clear from the fact that each bidder's bid tend to concentrate to a particular percentage of its valuation. A second important observation is that all the bidders bid concentrate around 50%-60% of their valuation. Although this fact is an artefact of our experiment it turns out that occurs also in real world scenarios `https://www.youtube.com/watch?v=X93omDyYjC4`. Last but not least, it is important to notice that although the bid tend to converge on average, it does not stop fluctuating. These fluctuations are necessary to permit each bidder to be aware of changes in the environment. As previously underlined, one major advantage of learning bidders is that they are able to overcome changes in the environment. Thus, to do so they must continuously explore it.

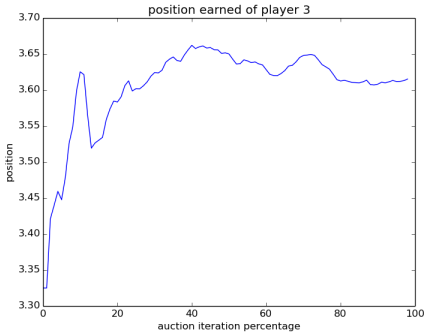
POSITION EVOLUTION We will present the evolution as the auction is repeated of the average position that each bidders get.



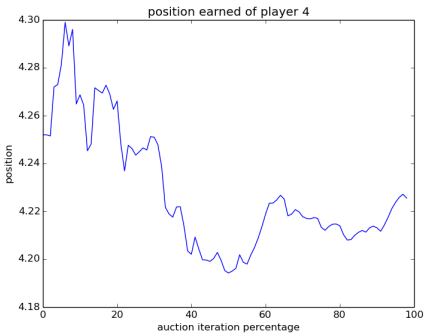


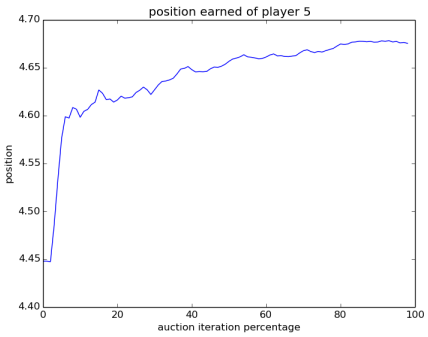
(a) valuations 72 and 61





(b) valuations 53 and 46



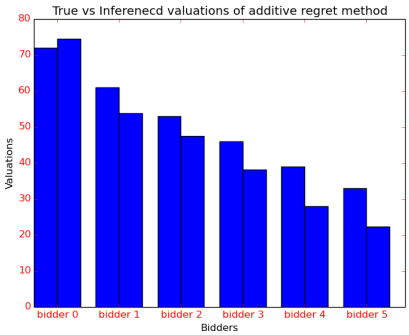


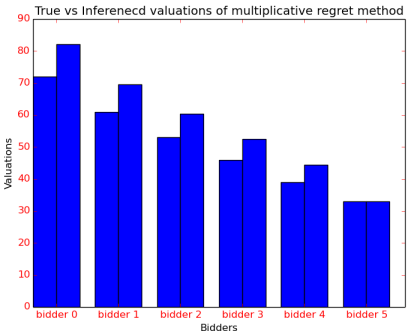
(c) valuations 39 and 33

Observations: In general the same observations made for the bid evolution diagrams apply also here. This comes from the fact that an increase in the average bid in the vast majority of cases leads to a better gained average position. In addition to that an important observation is the correlation

between the bidders valuation and the average position gained. In fact, the outcome is efficient. By efficient we mean an outcome where the average positioning of the bidders is decreasing in their valuation (remember that $\alpha 1 > \alpha_2 > ... > \alpha_m$).It is quite interesting the fact that the first bidder, although it has the best average position, it does not usually get it, as his average position is around 1.9 .This result comes from the competitiveness of the environment. Indeed the bidders are more than the positions available and the rough competition lead to a diminution of each bidder gains.

INFERENCE RESULTS We continue by presenting the result of the inference method minimizing the multiplicative regret and the one minimizing the additive regret





We measured the inference error of the two methods by comparing the average percentage error with respect to the valuations of each bidder. Formally ,defining as $\hat{v_i}$ the inferred valuation of bidder i, the equation used was:

$$\sum_{i=1}^{n} \frac{|v_i - \hat{v_i}|}{v_i} \tag{5.14}$$

1. additive regret method's error = 17.2%
2. multiplicative regret method's error = 11,7%

Observations First, we observe that the additive regret method tends to underestimate the true valuations in contrast with the multiplicative regret method that tends to overestimate them. This difference comes from our previous analysis of the pathologies of the additive method that the multiplicative try to overcome. In addition to that, an important observation is that the error's magnitude in the valuation's estimation for each bidder of the additive method is highly correlated with the magnitude of the gradient in small regret values. To elucidate this point further, just notice that the percentage error is lower for low rank bidders for which the rationilizable set is steepest in the small regret area.

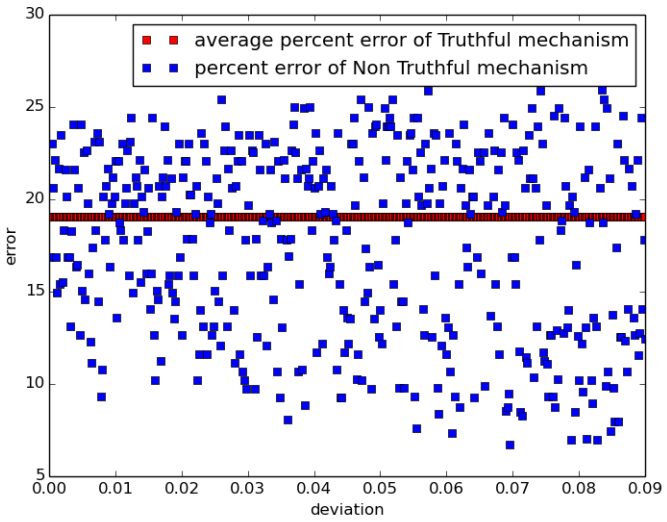
5.5 Robustness of the multiplicative regret inference valuation method

Now we will examine the robustness of the multiplicative regret inference valuation method.

Parkes et al. at [5] examined the robustness of the GSP auction in terms of how much truthful the outcomes were when the auction was not-truthful by itself. If the outcome of an auction is truthful then the bidders had received a position analogue to the magnitude of their valuation (i.e the bidder with the biggest valuation received the first position, the bidder with the second highest valuation the second one etc). Obviously we will concentrate to the properties of outcomes which consist a nash equilibrium. An efficient nash equilibrium is a nash equilibrium in which the outcome is truthful. This concept of truthfulness can obviously be extended also to other types of equilibrium concepts, as for locally envy-free nash equilibria. Imagine that the mechanism has not the true click probabilities perceived by the bidders. Then the payment rule will be based on the click probabilities that the system has. Let these click probabilities be denoted by the position vector $\hat{a}$. Then the profit of the bidders is based on the true position coefficient vector a in contrast with the payments which are based to the $\hat{a}$. This is a more realistic scenario to investigate than the precedent as particularly in online advertising the environment is highly evolving and auction designers are more exposed to faults.So, in general $\hat{a} \neq a$, following the same notation as in 3.2 we redefine our cost function according to the supposed by the auctioneer $\hat{a}$ and we have that the utility of bidder i under a bid profile of b is:

$$U_i(b) = u_i P_i(b) - \hat{C}_i(b) \tag{5.15}$$

We will test our valuation method robustness supposing that the only information available to us is the wrong vector used by the auctioneer $\hat{a}$. To investigate his robustness, we added an error term to each particular coefficient probability. The error term was draw by a normal distribution center to zero. In the following diagram we present the correlation of the error in the inference method to the variance of the added error term. We ran about 500 true α and non true α auctions of about 10,000 iterations each.



Truth based mechanisms have an average inference error of 19.059% while non truth based an average error of 17.733%. While the standard deviations are about 3% and 5%. By this we can deduce that the inference method is quite robust as a significant variation of the untruthfulness does not change the average results much. This is a quite interesting result which relies on the payments rule of the GSP. In fact the payment rule of the GSP is not additive in the sense of the summation of the errors among particular coefficient and thus is more stable in confront with the errors accumulation of other mechanisms.

A paradox fact has to be cleared. How is possible that the inference method of the non truth based mechanisms has a smaller error than the error of the truth ones. The main observation which can explain this fact is that the multiplicative error inference method produce an overestimation of the valuation. This overestimation is present in particular in the first (in terms of valuation) bidders estimation, due to the largest upper bound of valuation for each regret bound. When the untruth diminishes the difference between α_1 and α_2 then the inference is quite similar. But, when the difference enlarges then the bigger bidder "tends to want less the first position" and bids in a more restricted way. This behaviour creates a biased lower estimation that adjust the error term in the inference of the untruthful mechanisms. However we must underline that many times we observe an untruthful inference outcome (i.e two bidders with valuation $v_1 < v_2$ are inferred inversely) when we deal with untruth mechanism, an outcome that we do not observe at all in the inference of truth based mechanisms.

5.6 Under an Equilibrium Prospective

As proved at [8] no regret distributed algorithms converge in an ϵ approximate CCE equilibrium 2.4. Therefore we can deduce that inferencing the valuation of the players in an online ad auction with the [4] method is equivalent to finding the tuple of valuations that best minimizes the regret of the agents. Obviously this inference method can be applied to a particular bidder rather than the entire set of bidders and thus assume that only this particular bidder uses a no regret algorithm. However when this inference method is applied to the entire set of bidders then the only difference with the classic approach of supposing a mixed nash equilibrium is that we enlarge the possible set of equilibria.An interesting question would be if try the inferencing method under different equilibrium concepts what the outcomes would be.

5.7 Future Work

Some interesting ideas for future work are discussed at [23]. Combining the different evaluation methods through machine learning techniques would be an attractive direction. In addition to that, trying through similarity metrics between auctions to inference the player's valuation with classical learning techniques would be an interesting idea. Furthermore there are some dynamic aspects that the no regret inference method does not take into account. For example the evolution of the rationalizable set through time. How much can a particular valuation explain this evolution is a tough question. This suggest a tradeoff between the two different explanations. The one that tries to describe the auction in his entire and the one other that tries to explain the auction behaviour evolution.

A completely different direction is one that suggests ways for the auctioneer to maximize his income. Does the starting information provided to the bidders produce different steady state outcomes? How can we dynamically set a reserve price to maximize the revenue by changing the equilibrium? From the bidders perspective, when switched to full information contexts they implement full information no regret algorithms rather that bandit ones. Does this difference play a relevant role?

Last but not least, an interesting question would be how can the bidders maximize their utility beyond the simple assumption of no regret behaviour. Also our implementation was based on a discrete model for bidding (i.e. the EXP3 algorithm), does a continues space algorithm like the mab template implemented with the online gradient descent outperforms the discrete ones?

Bibliography

[1] Emmanuel Guerre, Isabelle Perrigne, and Quang Vuong. Optimal nonparametric estimation of first-price auctions. *Econometrica*, 68(3):525–574, 2000. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/2999600`.

[2] Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, December 2007. URL `https://ideas.repec.org/a/eee/indorg/v25y2007i6p1163-1178.html`.

[3] Susan Athey and Denis Nekipelov. A structural model of sponsored search advertising auctions. In Sixth ad auctions workshop, May 2010.

[4] Denis v, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, pages 1–18, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3410-5. doi: 10.1145/2764468.2764522. URL `http://doi.acm.org/10.1145/2764468.2764522`.

[5] Paul Dütting, Felix A. Fischer, and David C. Parkes. Truthful outcomes from non-truthful position auctions. *CoRR*, abs/1602.07593, 2016. URL `http://arxiv.org/abs/1602.07593`.

[6] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961. URL `https://EconPapers.repec.org/RePEc:bla:jfinan:v:16:y:1961:i:1:p:8-37`.

[7] Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, February 1981. ISSN 0364-765X. doi: 10.1287/moor.6.1.58. URL `http://dx.doi.org/10.1287/moor.6.1.58`.

[8] Tim Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 1st edition, 2016. ISBN 131662479X, 9781316624791.

[9] John F. Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[10] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, March 2007. doi: 10.1257/aer.97.1.242. URL http://www.aeaweb.org/articles?id=10.1257/aer.97.1.242.

[11] Sanjev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 8:121 – 164, 2012.

[12] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212 – 261, 1994. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1994.1009. URL http://www.sciencedirect.com/science/article/pii/S0890540184710091.

[13] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291 – 307, 2005. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2004.10.016. URL http://www.sciencedirect.com/science/article/pii/S0022000004001394. Learning Theory 2003.

[14] Yoav Freund and Robert 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, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL http://www.sciencedirect.com/science/article/pii/S002200009791504X.

[15] J. Robinson. An iterative method of solving a game. *The Annals of Mathematics*, 54:296–301, September 1951.

[16] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999. URL https://EconPapers.repec.org/RePEc:eee:gamebe:v:29:y:1999:i:1-2:p:79-103.

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

[18] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016. ISSN 2167-3888. doi: 10.1561/2400000013. URL http://dx.doi.org/10.1561/2400000013.

[19] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*,

ICML'03, pages 928–935. AAAI Press, 2003. ISBN 1-57735-189-4. URL http://dl.acm.org/citation.cfm?id=3041838.3041955.

[20] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 09 1952. URL https://projecteuclid.org:443/euclid.bams/1183517370.

[21] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003. ISSN 0097-5397. doi: 10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

[22] Avrim Blum, Yishay Mansour, and Jamie Morgenstern. Learning valuation distributions from partial observation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 798–804, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9550.

[23] Noam Nisan and Gali Noti. An experimental evaluation of regret-based econometrics. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 73–81, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4913-0. doi: 10.1145/3038912.3052621. URL https://doi.org/10.1145/3038912.3052621.

[24] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[25] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007. ISBN 0521872820.

[26] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135.

[27] Brown. *Analysis of Production and Allocation*. Wiley, 1951.

[28] Howard Karloff. *Linear Programming*.

[29] Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, pages 1040–1053, 1982.

[30] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928. URL http://eudml.org/doc/159291.

[31] Shuchi Chawla, Jason Hartline, and Denis Nekipelov. A/b testing of actions. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 19–20, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3936-0. doi: 10.1145/2940716.2940757. URL http://doi.acm.org/10.1145/2940716.2940757.

[32] Edward Clarke. Multipart pricing of public goods. *Public Choice*, 11 (1):17–33, 1971. URL https://EconPapers.repec.org/RePEc:kap:pubcho:v:11:y:1971:i:1:p:17-33.

[33] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:41:y:1973:i:4:p:617-31.

[34] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:68:y:2000:i:5:p:1127-1150.

[35] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007. URL http://dl.acm.org/citation.cfm?id=1314543.

[36] Tim Roughgarden, Vasilis Syrgkanis, and Éva Tardos. The price of anarchy in auctions. *CoRR*, abs/1607.07684, 2016. URL http://arxiv.org/abs/1607.07684.

[37] Shuchi Chawla, Jason D. Hartline, and Denis Nekipelov. Mechanism design for data science. *CoRR*, abs/1404.5971, 2014. URL http://arxiv.org/abs/1404.5971.

[38] Michael Ostrovsky and Michael Schwarz. Reserve prices in internet advertising auctions: A field experiment. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 59–60, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0261-6. doi: 10.1145/1993574.1993585. URL http://doi.acm.org/10.1145/1993574.1993585.

[39] L S. Shapley and Martin Shubik. The assignment game i: The core. 1: 111–130, 12 1971.

[40] Paul Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004. URL https://EconPapers.repec.org/RePEc:cup:cbooks:9780521536721.

[41] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Conference on Algorithms*, ESA'07, pages

253–264, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-75519-5, 978-3-540-75519-7. URL http://dl.acm.org/citation.cfm?id=1778580.1778606.

[42] Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26(2):305–321, 1978. doi: 10.1287/opre.26.2.305.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή στην Θεωρία Παιγνίων και το Σχεδιασμό Μηχανισμών

1.1 Εισαγωγικά Παραδείγματα

Ο Σχεδιασμός Μηχανισμών είναι η επιστήμη που βρίσκεται στην τομή μεταξύ Θεωρίας Παιγνίων, Οικονομικών και Επιστήμης Υπολογιστών. Κύριος σκοπός του είναι ο σχεδιασμός μηχανισμών που εξασφαλίζουν υψηλή απόδοση όταν χρησιμοποιούνται από στρατηγικούς παίκτες. Η σημασία σχεδιασμού καλών μηχανισμών μπορεί να φανεί παρατηρώντας τις αποτυχίες των κακών μηχανισμών.

Διαισθητικό Παράδειγμα: Μπάντμιντον γυναικών στους Ολυμπιακούς Αγώνες του 2012

Η δομή του αγωνίσματος αποτελείται από δύο φάσεις. Στην πρώτη φάση σχηματίζονται τέσσερις όμιλοι (A,B,C,D), ο καθένας εκ των οποίων περιλαμβάνει τέσσερις ομάδες. Οι δύο πρώτες ομάδες κάθε ομίλου προχωρούν στην δεύτερη φάση, μετά από μια ρουνδ-ροβιν αλληλουχία αγώνων. Στη δεύτερη φάση υπάρχουν τρεις αναμετρήσεις άμεσης αποχώρησης (νοκ-άουτ), όπου μία ομάδα μπορεί να αποκλειστεί ή να περάσει στον επόμενο γύρο, μέχρι να αναδειχθεί τελικά ο νικητής. Στην πρώτη νοκ-άουτ αναμέτρηση της δεύτερης φάσης, η δεύτερη ομάδα του πρώτου ομίλου αντιμετωπίζει την πρώτη του δεύτερου ομίλου και αντίστοιχα στον τρίτο και τέταρτο όμιλο.

Αφού παρουσιάσαμε τη δομή του αγωνίσματος, ας δούμε τώρα τι συνέβη. Στην ομάδα D βρισκόταν η καλύτερη, κατά γενική ομολογία, ομάδα, η Κινέζικη

ομάδα των Qing - Wunlei. Στην τελευταία αναμέτρησή τους για την πρώτη φάση ηττήθηκαν από τη Δανέζικη Εθνική ομάδα, τερματίζοντας με αυτό τον τρόπο δεύτερες στον όμιλό τους. Ως αποτέλεσμα, η πρώτη στην κατάταξη ομάδα του ομίλου Α θα έπρεπε να αναμετρηθεί, στην επόμενη φάση, με την Κινέζικη ομάδα. Ωστόσο, στον τελευταίο αγώνα του Α ομίλου, δύο ομάδες που είχαν ως τότε σημειώσει 2-0 (2 νίκες, 0 ήττες) έπαιζαν μεταξύ τους. Για κάθε ομάδα εκ των δύο, η νίκη ισοδυναμούσε με την πρώτη θέση του ομίλου και άρα αναμέτρηση με την πολύ δυνατή Κινέζικη ομάδα. Αυτό που συνέβη, λοιπόν, ήταν ότι προσπάθησαν και οι δύο να χάσουν εσκεμμένα το ματς. `https://www.youtube.com/watch?v=7mq1ioqiWEo`

Χωρίς αμφιβολία, λόγω της αναμέτρησης αυτής ξέσπασε μεγάλο σκάνδαλο. Για να καταλάβουμε όμως σε μεγαλύτερο βάθος τι συνέβη, πρέπει να αντιληφθούμε πρώτα ποια είναι τα κίνητρα των παικτών. Τι επιθυμεί ένας παίκτης· Προφανώς, να προχωρά όσο περισσότερο μπορεί σε κάθε διοργάνωση. Τι επιθυμεί ο διοργανωτής του διαγωνισμού· Να προσπαθεί κάθε παίκτης όσο περισσότερο μπορεί να κερδίσει κάθε αγώνα. Επομένως, καθίσταται εμφανές από το παράδειγμά μας ότι ως προς το περιεχόμενο των κανόνων του διαγωνισμού,τα κίνητρα των συμμετέχοντων δεν είναι πάντα ευθυγραμμισμένα με τα κίνητρα των διοργανωτών. Για μια ανάλυση σε μεγαλύτερο βάθος προτείνουμε το άρθρο του Kleinberg στο `https://agtb.wordpress.com/2012/08/01/olympic-badminton-is-not-incentive-compatible-6/`

Το εν λόγω παράδειγμα υπογραμμίζει πόσο σημαντικός είναι ο σχεδιασμός σωστών κανόνων. Τα αποτελέσματα, όταν οι κανόνες δεν έχουν επιλεχθεί κατάλληλα, είναι πιθανόν να είναι ιδιαίτερα και ανεπιθύμητα.

Επομένως, ο σχεδιασμός μηχανισμών χρησιμοποείται για να σχεδιάσουμε κανόνες που εξασφαλίζουν, κατά μία έννοια, ¨καλά' αποτελέσματα.

1.2 Δημοπρασία Πρώτης και Δεύτερης Τιμής

Ο καλύτερος τρόπος για να ξεκινήσουμε την ανάλυση της σχεδίασης μηχανισμών είναι εισάγοντας ορισμένα μοντέλα για δημοπρασίες όπου υπάρχει μία μονάδα ενός προϊόντος για πώληση (single item auctions). Σε αυτή την ενότητα θα παρουσιάσουμε κάποιες βασικές φόρμες δημοπρασιών και θα αποδείξουμε τα πρώτα χρήσιμα λήμματα.

Δημοπρασία ενός προϊόντος (Single Item Auction): Αρχικά, ας δούμε τι ακριβώς είναι μια απλή δημοπρασία ενός προϊόντος. Ας υποθέσουμε ότι ένας πωλητής έχει ένα αντικείμενο που θέλει να πουλήσει και υπάρχουν ν υποψήφιοι αγοραστές που είναι διατεθειμένοι να δώσουν χρήματα για αυτό το

αντικείμενο. Κάθε αγοραστής έχει μια ιδιωτική παράμετρο που ονομάζεται αξία (αλυατιον), η οποία αντιπροσωπεύει πόσο θέλει ο αγοραστής το αντικείμενο και πόσα χρήματα είναι διατεθειμένος να πληρώσει για αυτό. Τότε, η δημοπρασία ξεκινά και κάθε αγοραστής (από εδώ και στο εξής θα τους αποκαλούμε παίκτες) ποντάρει. Ο πωλητής επιλέγει, βασισμένος στα ποντάρίσματα, ποιος θα πάρει το αντικείμενο και πόσο θα πρέπει να πληρώσει για αυτό. Τώρα, απομένει να δούμε ποιο είναι το κέρδος κάθε παίκτη και το κέρδος του πωλητή. Σε ό,τι αφορά το κέρδος του πωλητή, θα δούμε ότι δεν υπάρχει προφανής απάντηση, καθώς ο πωλητής θα μπορούσε να είναι η Κυβέρνηση, άρα σκοπός της θα ήταν (ή θα έπρεπε να είναι) να πράξει για το κοινό καλό της κοινωνίας μεγιστοποιώντας το κεφάλαιό της, ή θα μπορούσε να είναι μια ιδιωτική εταιρεία, σκοπός της οποίας είναι να μεγιστοποιήσει τα κέρδη της. Ως προς τους παίκτες θα χρησιμοποιήσουμε, σε όλη αυτή τη διπλωματική εργασία, το σχεδόν-γραμμικό (χυασι-λινεαρ) μοντέλο του κέρδους. Το μοντέλο αυτό λειτουργεί ως εξής: Εάν ένας παίκτης δεν πάρει το αντικείμενο, τότε το κέρδος του είναι ίσο με το μηδέν, ενώ αν πάρει το αντικείμενο σε μια τιμή p και η ιδιωτική του αξία είναι v_i, τότε το κέρδος του είναι ίσο με $v_i - p$.

Δημοπρασίες Σφραγισμένων Προσφορών (Sealed-Bid Auctions): Η μορφή της δημοπρασίας που θα χρησιμοποιήσουμε για όλες τις διαφορετικές δημοπρασίες είναι η ακόλουθη:

1. Κάθε παίκτης i λέει ιδιωτικά το ποντάρισμά του (βιδ) b_i στον δημοπράτη (ώστε οι υπόλοιποι παίκτες να μην ξέρουν το ποντάρισμά του).
2. Ο δημοπράτης επιλέγει ποιος θα πάρει το αντικείμενο (μπορεί να μην το πάρει κανείς)
3. Ο δημοπράτης επιλέγει την τιμή πώλησης

Τώρα πρέπει να ξεκινήσουμε να σκεφτόμαστε πώς να υλοποιήσουμε τα βήματα 2 και 3. Το βήμα 2 είναι εύκολο, απλά δίνουμε το αντικείμενο στον υποψήφιο αγοραστή που έχει κάνει το μεγαλύτερο ποντάρισμα. Αυτή η απόφαση μοιάζει προφανής, όμως, όπως θα δούμε αργότερα, όταν ο κύριος σκοπός μας είναι να μεγιστοποιήσουμε τα έσοδα, δεν αποτελεί πάντα τη σωστή επιλογή. Είναι πιο δύσκολο να σκεφτούμε έναν απλό τρόπο να υλοποιήσουμε το βήμα 3. Μια απλή ιδέα, αν για παράδειγμα ο στόχος μας είναι να δώσουμε το αντικείμενο σε αυτόν που το θέλει παραπάνω χωρίς να ενδιαφερόμαστε για τα έσοδα, είναι να το δώσουμε δωρεάν. Αυτή η ιδέα είναι κακή, επειδή οι παίκτες είναι στρατηγικοί και η δημοπρασία θα εξελιχθεί σε ένα παιχνίδι του ποιος θα ανακοινώσει το μεγαλύτερο αριθμό.

Δημοπρασίες Πρώτης Τιμής: Όταν αυτός που κάνει τη μεγαλύτερη προσφορά πληρώνει το ποσό του πονταρίσματός του η δημοπρασία ονομάζεται Δημοπρασία Πρώτης Τιμής. Αν και η συγκεκριμένη δημοπρασία μοιάζει απλή, από την οπτική του υποψήφιου αγοραστή δεν είναι. Για να καταλάβουμε την αιτία, ας σκεφτούμε το εξής πείραμα: υπάρχουν δύο παίκτες, εκ των οποίων ο πρώτος είναι ελεγχόμενος από εμάς, με αξία $v_i \in [0,1]$ και ο δεύτερος παίκτης έχει αξία X ,που είναι μια ομοιόμορφη τυχαία μεταβλητή στο $[0,1]$. Πώς θα έπρεπε να παίξουμε για να μεγιστοποιήσουμε το κέρδος μας· Αν ο άλλος παίκτης ποντάρει απλά την αξία του, η απάντηση είναι απλή, βρίσκουμε το ποντάρισμα μεγιστοποιώντας τη συνάρτηση $(v-b)(1-b)$. Αν όμως ο άλλος παίκτης παίξει με διαφορετικό τρόπο (πράγμα που είναι προφανές γιατί το να ποντάρει κάθε φορά την αξία του θα του αποφέρει μηδενικό κέρδος), πώς πρέπει να ποντάρουμε· Αυτό το απλό παράδειγμα δείχνει ότι ίσως αυτή η μορφή δημοπρασίας να μην είναι τόσο καλή. Είναι δύσκολο για τον δημοπράτη να κάνει υποθέσεις σχετικά με το πώς θα συμπεριφερθούν οι υποψήφιοι αγοραστές και οπότε να προβλέψει το αποτέλεσμα της δημοπρασίας.

Second Price Auctions: Δημοπρασίες Δεύτερης Τιμής:Μια άλλη πολύ συνηθισμένη μορφή δημοπρασίας είναι η Δημοπρασία Δεύτερης Τιμής, την έννοια της οποίας εισήγαγε πρώτος ο ΐσκρεψ στο [6]. Στη δημοπρασία δεύτερης τιμής, ο πλειοδότης, ή αλλιώς ο παίκτης που ποντάρει τα περισσότερα, παίρνει το αντικείμενο και πληρώνει το ποντάρισμα του δεύτερου κατά σειρά πλειοδότη (και ένα μικρό ϵ παραπάνω). Η διαίσθηση πίσω από αυτή τη μορφή δημοπρασίας είναι απλή, ο νικητής δεν είναι αναγκασμένος να πληρώσει το ποσό που ποντάρισε, παρά μόνο το ελάχιστο πιθανό ποντάρισμα που θα του έδινε τη νίκη σε κάθε περίπτωση. Θα συνεχίσουμε αποδεικνύοντας δύο πολύ σημαντικά θεωρήματα που καθιστούν αυτή τη μορφή δημοπρασίας τόσο ευρέως χρησιμοποιούμενη.

Θεωρεμ 1.2.1. *Σε μια δημοπρασία ενός προϊόντος δεύτερης τιμής (single item-second price), κάθε παίκτης με αξία v_i έχει μια κυρίαρχη στρατηγική πονταρίσματος, θέτοντας το ποντάρισμά του b_i σε τιμή ίση με την ιδιωτική του αξία.*

Η συγκεκριμένη ιδιότητα είναι πολύ σημαντική. Μας δίνει ένα δυνατό στοιχείο για τη συμπεριφορά των ατόμων που ποντάρουν. Αν έχουμε ένα σύστημα από λογικούς παίκτες που θέλουμε να εξετάσουμε και κάποιος παίκτης έχει κυρίαρχη στρατηγική, η υπόθεση ότι ο συγκεκριμένος παίκτης θα παίξει σύμφωνα με αυτήν είναι η πιο αδύναμη υπόθεση σε αυτή τη δομή.

Απόδειξη. Έστω τυχαίος παίκτης i με αξία v_i και το διάνυσμα των πονταρισμάτων των υπόλοιπων παικτών b_{-i}.
Τώρα πρέπει να αποδείξουμε ότι ανεξαρτήτως της τιμής του διανύσματος b_{-i},

το κόστος του παίκτη i μεγιστοποιείται θέτοντας $b_i = v_i$.

Έστω ότι το μέγιστο ποντάρισμα των υπόλοιπων παικτών είναι $B = max_{j \neq i} b_j$.
Τότε διακρίνουμε περιπτώσεις:
(Υπενθυμίζουμε ότι υτιλιτψ είναι το κέρδος)

1. Αν $b_i \geq B$ τότε $utility_i = v_i - B$

2. Αν $b_i \leq B$ τότε $utility_i = 0$

Άρα, το καλύτερο που μπορούμε να ελπίζουμε είναι ότι θα κάνουμε ένα ποντάρισμα b_i έτσι ώστε σε κάθε περίπτωση να εξασφαλίζουμε ότι $utility_i = max\{v_i - B, 0\}$.
Αν $v_i = b_i$ τότε:

1. Αν $b_i \geq B \Rightarrow v_i \geq B$ τότε $utility_i = v_i - B = max\{v_i - B, 0\}$

2. Αν $b_i \leq B \Rightarrow v_i \leq B$ τότε $utility_i = 0 = max\{v_i - B, 0\}$

□

Θεορεμ 1.2.2. *Σε μια δημοπρασία ενός προϊόντος δεύτερης τιμής (σινγλε ιτεμ-σεσονδ πριςε), κάθε παίκτης που ποντάρει την αξία του είναι εξασφαλισμένο ότι θα έχει σίγουρα μη-αρνητικό κέρδος.*

Προτού προχωρήσουμε στην απόδειξη του προηγούμενου θεωρήματος, είναι σημαντικό να υπογραμμίσουμε τη σημασία του. Φανταστείτε μία δημοπρασία όπου ένας παίκτης έχει κυρίαρχη στρατηγική αλλά το κέρδος του είναι αρνητικό. Τι θα κάνατε στη θέση του· Μα φυσικά θα αποχωρούσατε από τη δημοπρασία. Επομένως, μία σημαντική ιδιότητα κάθε δημοπρασίας είναι η ικανότητα να διατηρεί τους παίκτες της. Η εν λόγω ιδιότητα σίγουρα απαιτεί ότι οι παίκτες έχουν μη αρνητικό κέρδος.

Απόδειξη. Θυμηθείτε από το προηγούμενο θεώρημα ότι το να ποντάρει την αξία του εξασφαλίζει σε κάθε παίκτη ότι $utility_i = max\{v_i - B, 0\} \geq 0$. □

1.3 Λήμμα του Myerson

Το λήμμα του Μψερσον αποτελούσε το βασικό εργαλείο πάνω στο οποίο σχεδιάζονταν οι δημοπρασίες για πολλά χρόνια. Ο Μψερσον με το [7] παρείχε μία σχεδιαστικά απλή προσέγγιση για τη δημιουργία DSIC (Dominant Strategy Incentive Compatible) δημοπρασιών σε πιο γενικευμένες συνθήκες από αυτές των δημοπρασιών ενός αντικειμένου (single item auctions).

Προτού το παρουσιάσουμε, ας γενικεύσουμε το μοντέλο δημοπρασιών μας πέρα από τις σινγλε ιτεμ δημοπρασίες.

Μονοπαραμετρικό Περιβάλλον

1. Η δημοπρασία έχει έναν σταθερό αριθμό παικτών n
2. Κάθε παίκτης έχει σταθερή αξία v_i που αντιπροσωπεύει την αξία ανά μονάδα αγαθού που παίρνει ο παίκτης
3. Ξ είναι το εφικτό σετ ν-διανυσμάτων καλών διανομών $(x_1, x_2, \ldots, x_n)$

Για να δείξουμε την αναλυτική ικανότητα αυτού του μοντέλου, ας δώσουμε ορισμένα παραδείγματα.

1. Σε μια δημοπρασία μοναδικού προϊόντος (σινγλε ιτεμ) $X = \{(x_1, x_2, \ldots, x_n) \,|x_i \in \{0, 1\}, \sum_{i=1}^{n} x_i \leq 1\}$
2. Αν στη δημοπρασία πωλούνται κ πανομοιότυπα αγαθά, τότε $X = \{(x_1, x_2, \ldots, x_n) \,|x_i \in \{0, 1\}, \sum_{i=1}^{n} x_i \leq k\}$

Μορφή δημοπρασίας σφραγισμένων προσφορών (Sealed bid auction template): Ο σχεδιαστής μηχανισμών σε μία δημοπρασία σφραγισμένων προσφορών πρέπει να υλοποιήσει δύο βασικές συναρτήσεις. Πρώτα πρέπει να υλοποιήσει τη συνάρτηση κανόνα διανομής, η οποία παίρνει ως είσοδο το διάνυσμα των πονταρισμάτων και δίνει ως έξοδο το διάνυσμα της διανομής των αγαθών. Δευτερευόντως πρέπει να υλοποιήσει τη συνάρτηση (κανόνα) πληρωμής που παίρνει ως είσοδο το διάνυσματα των πονταρισμάτων και δίνει ως έξοδο το διάνυσμα των πληρωμών. Τυπικά η δημοπρασία σφραγισμένων προσφορών που βασίζεται σε σε ένα συγκεκριμένο περιβάλλον μίας παραμέτρου περιγράφεται από τρία βήματα:

1. συλλέγουμε το διάνυσμα των πονταρισμάτων $b = (b_1, b_2, ..., b_n)$
2. διαλέγουμε μία εφικτή διανομή χρησιμοποιώντας μία συνάρτηση $x : \mathrm{Re}^n \to X$
3. παίρνουμε την πληρωμή (παψμεντ) κάθε παίκτη χρησιμοποιώντας μία συνάρτηση $p : \mathrm{Re}^n \to \mathrm{Re}^n$

Η μορφή δημοπρασίας σφραγισμένων προσφορών είναι σημαντική για να προστατευτεί η ανωνυμία των παικτών. Σε πραγματικές καταστάσεις όπου το περιβάλλον είναι ανταγωνιστικό, οι παίκτες έχουν οριοθετημένο βυδγετ και πολλαπλά σινγλε ιτεμς πωλούνται συνεχόμενα, μία καλή τακτική για τον δεύτερο (σε ποσό πονταρίσματος) παίκτη, ακόμα και αν η δημοπρασία που πραγματοποιείται είναι η δημοπρασία δεύτερης τιμής, είναι να ποντάρει πάνω από την πραγματική αξία του. Με τον τρόπο αυτό, θα επιβάλει μεγαλύτερη τιμή απόκτησης για τον πρώτο (σε ποσό πονταρίσματος) παίκτη, οδηγώντας τον στο να εξαντλήσει γρηγορότερα το budget του. Επιπρόσθετα, ακόμα και σε άλλα μοντέλα πονταρίσματος, το ποντάρισμα ανταναχλά τακτική μάρκετινγκ την οποία οι εταιρείς είναι πολύ διστακτικές στο να αποκαλύψουν.

Για το κέρδος κάθε παίκτη, θα χρησιμοποιήσουμε όπως πάντα το σχεδόνγραμμικό μοντέλο κέρδους. Αν μία δημοπρασία έχει κανόνα διανομής x και κανόνα πληρωμής p, τότε το κέρδος του παίκτη i όταν αυτός έχει τα χαρακτηριστικά πονταρίσματος b είναι: $utility_i = v_i x_i(b) - p_i(b)$
Θα περιοριστούμε σε κανόνες πληρωμής που ικανοποιούν την εξής συνάρτηση: $0 \leq p_i(b) \leq b_i x_i(b)$
Το αριστερό σκέλος της ανισότητας περιορίζει την έρευνά μας σε μηχανισμούς όπου ο πωλητής δεν πληρώνει τους παίκτες. Το δεξί σκέλος, όμως, διασφαλίζει ότι ένας ειλικρινής παίκτης θα έχει πάντα μη-αρνητικό κέρδος (επομένως περιοριζόμαστε απευθείας σε incentive compatible μηχανισμούς).

Τώρα, είμαστε σχεδόν έτοιμοι να παρουσιάζουμε το λήμμα του Myerson. Ωστόσο, απομένει να παρουσιάσουμε δύο βοηθητικούς ορισμούς.

Δεφινιτιον 1.3.1. *Μια συνάρτηση διανομής x για μονοπαραμετρικό περιβάλλον είναι υλοποιήσιμη αν υπάρχει κανόνας πληρωμής p έτσι ώστε η δημοπρασία σφραγισμένων προσφορών (x, p) να είναι ΔΣΓ.*

Για να εξηγήσουμε περαιτέρω τον ορισμό, είναι σημαντικό να αντιληφθούμε ότι αν προβάλουμε το χώρο των DSIC μηχανισμών στους κανόνες διανομών τους, θα πάρουμε το χώρο των υλοποιήσιμων κανόνων διανομής. Επομένως, αν ο σκοπός μας είναι να σχεδιάσουμε DSIC μηχανισμούς, είμαστε υποχρεωμένοι να περιοριστούμε σε κανόνες διανομής που είναι υλοποιήσιμοι.

Δεφινιτιον 1.3.2. *Μια συνάρτηση διανομής x για μονοπαραμετρικό περιβάλλον είναι μονότονη αν για κάθε παίκτη i και για σταθερό διάνυσμα πονταρισμάτων των υπόλοιπων παικτών b_{-i}, η συνάρτηση $x_i(z, b_{-i})$ είναι μη φθίνουσα ως προς z.*

Άρα, αν η x είναι μονότονη συνάρτηση διανομής, έπεται ότι το να ποντάρει περισσότερο ένας παίκτης μπορεί μόνο να του αποφέρει όλο και πιο πολλά αντικείμενα. Φανταστείτε για μια στιγμή μια συνάρτηση διανομής που δεν είναι

μονότονη, δηλαδή κερδίζει ο δεύτερος κατά σειρά πλειοδότης. Διαισθητικά αυτό είναι πολύ παράξενο. Ταυτόχρονα, πέραν του ότι μας φαίνεται παράλογη επιλογή, θα δούμε ότι και από την οπτική του σχεδιασμού μηχανισμών όντως αποτελεί κακή επιλογή. Θα αποδείξουμε ότι αν μια συνάρτηση διανομής δεν είναι μονότονη, τότε είμαστε σίγουροι ότι δεν είναι υπολογίσιμη.

Θεωρεμ 1.3.3. *Έστω μονοπαραμετρικό περιβάλλον*

1. *Μια συνάρτηση διανομής x είναι υλοποιήσιμη αν και μόνο αν είναι μονότονη*
2. *Αν η συνάρτηση x είναι μονότονη, τότε υπάρχει μοναδική συνάρτηση p τέτοια ώστε η δημοπρασία σφραγισμένων προσφορών (x, p) να είναι DSIC*
3. *Ο κανόνας διανομής DSIC μηχανισμών με μια συνάρτηση διανομής x δίνεται από ακριβή σχέση*

Το λήμμα του Myerson αποτελεί την θεμελιώδη βάση της σχεδίασης μηχανισμών. Προτού προχωρήσουμε στην απόδειξη, ας καταλάβουμε τι ακριβώς λέει. Ουσιαστικά λέει το αν θέλουμε να δημιουργήσουμε μία DISC δημοπρασία, το πρώτο βήμα είναι να βρούμε μία συνάρτηση διανομής που είναι μονότονη. Τότε, δεν χρειάζεται να σκεφτούμε καθόλου πάνω στον κανόνα πληρωμής. Είναι μοναδικός και υπάρχει ακριβής σχέση για αυτόν.

Απόδειξη. Η συνθήκη DSIC απαιτεί ότι για κάθε παίκτη i και b_{-i}, το κέρδος του παίκτη i πρέπει να είναι μέγιστο όταν ποντάρει την πραγματική του αξία v_i. Έστω (x, p) ένας DSIC μηχανισμός. Έστω τώρα ένας συγκεκριμένος παίκτης i και σταθερό διάνυσμα ποντάρισμάτων για τους υπόλοιπους παίκτες b_{-i}, για συντομία ας συμβολίσουμε την συνάρτηση $x_i(z, b_{-i})$ με $x(z)$ και την συνάρτηση $p_i(z, b_{-i})$ με $p(z)$.
Έστω $0 \leq y < z$
Ας υποθέσουμε ότι ο παίκτης i έχει αξία y και ποντάρει z, τότε από την DSIC ιδιότητα έχουμε ότι:

$$yx(y) - p(y) \geq yx(z) - p(z) \Rightarrow y(x(y) - x(z)) \geq p(y) - p(z) \tag{1.1}$$

Ας υποθέσουμε στη συνέχεια ότι ο παίκτης i έχει αξία z και ποντάρει y, τότε από την DSIC ιδιότητα έχουμε ότι:

$$zx(z) - p(z) \geq zx(y) - p(y) \Rightarrow p(y) - p(z) \geq z(x(y) - x(z)) \tag{1.2}$$

Με συνδυασμό των δύο παραπάνω ανισοτήτων παίρνουμε:

$$y(x(y) - x(z)) \geq p(y) - p(z) \geq z(x(y) - x(z)) \quad \forall y, z \quad 0 \leq y < z \tag{1.3}$$

Καθώς η παραπάνω ανισότητα ισχύει $\forall y, z \quad 0 \leq y < z$ τότε η συνάρτηση x πρέπει να είναι μη φθίνουσα. Ας υποθέσουμε όμως ότι $\exists z_1, z_2$ με $0 \leq z_1 < z_2$ και $x(z_1) > x(z_2)$. Τότε πρέπει να ισχύει ότι:
$z_1(x(z_1) - x(z_2)) \geq z_2(x(z_1) - x(z_2)) \Rightarrow z_1 \geq z_2$ το οποίο είναι άτοπο.
Επομένως, ως τώρα έχουμε αποδείξει ότι $(x, p) \quad DSIC \Rightarrow x \quad monotone.$

Τώρα πρέπει να αντιληφθούμε πώς πρέπει να είναι ο κανόνας πληρωμής. Ας υποθέσουμε ότι η συνάρτηση πληρωμής x είναι διαφορίσιμη. Τότε διαιρώντας με $y - z$ και παίρνοντας το όριο όταν ζ τείνει ψ, παίρνουμε ότι $p^{'}(z) = zx^{'}(z)$
Επομένως:
$p(b_i) - p(0) = \int_0^{b_i} zx^{'}(z)$
από τη στιγμή που η δημοπρασία είναι incentive compatible οφείλει να εξασφαλίζει ακόμη και σε ένα παίκτη με μηδενική αξία ότι η ειλικρίνεια του δεν θα του κοστίσει.Αυτο σημαίνει ότι $p(0) = 0$ και άρα η προηγούμενη σχέση γίνεται:

$p(b_i) = \int_0^{b_i} zx^{'}(z)$

Αυτή είναι μια ακριβής σχέση για το πώς πρέπει να είναι ο κανόνας πληρωμής. Επιπλέον, μιας και το σύστημά μας είναι DSIC, είναι η μόνη δυνατή.
Φυσικά η ιδέα του Myerson δεν απαιτεί να είναι διαφορίσιμη η συνάρτηση x. Πράγματι, αν η συνάρτησή μας είναι βηματικά σταθερή, τότε ορίζοντας εκ νέου την παράγωγο ως μέγεθος του άλματος στο σημείο ασυνέχειας, ισχύει η ίδια σχέση. Η απλοποιημένη εξίσωση για βηματικά σταθερές συναρτήσεις είναι η εξής:
$p_i(b_i, b_{-i}) = \sum_{j=1}^{l} z_l(x_i(z_l^+, b_{-i}) - x_i(z_l, b_{-i}))$
όπου τα σημεία $z_1, z_2, ..., z_l$ είναι τα σημεία ασυνέχειας της συνάρτησης διανομής $x_i(z, b_{-i})$ από το μηδέν στο b_i.

Τώρα απομένει να αποδείξουμε ότι αν η συνάρτηση διανομής x είναι μονότονη τότε είναι υπολογίσιμη. Ας υποθέσουμε ότι η x είναι μονότονη και διαφορίσιμη. Θ αποδείξουμε ότι η δημοπρασία σφραγισμένων προσφορών με συνάρτηση πληρωμής $p_i(b_i, b_{-i}) = \int_0^{b_i} zx_i^{'}(z, b_{-i})$ για κάθε παίκτη i είναι DSIC.
Ας δούμε ποιο είναι το κέρδος του παίκτη i (θα ακολουθήσουμε το ίδιο σκεπτικό με προγουμένως).
$utility_i(b) = v_i x_i(b) - p_i(b) = v_i x_i(b) - \int_0^{b_i} zx^{'}(z)$
Πάιρνοντας τη μερική παράγωγο έχουμε:
$\frac{\partial utility_i}{\partial b} = v_i \frac{\partial x_i}{\partial b} - bx^{'}(b)$

Παίρνουμε το μέγιστο του κέρδους αν θέσουμε την παράγωγό του ίση με μηδέν
$\frac{\partial utility_i}{\partial b} = 0 \Rightarrow b = v_i$

Άρα, αποδείξαμε μόλις ότι το να παίζει την αξία του είναι κυρίαρχη στρατηγική για τον παίκτη. Τώρα, για να ολοκληρώσουμε την απόδειξη θα αποδείξουμε ότι

το να στοιχηματίζει ο παίκτης με ειλικρίνεια (bidding truthfully) δεν οδηγεί ποτέ σε αρνητικό κέρδος. Ισοδύναμα, θα αποδείξουμε ότι:

$v_i x_i(v_i) \geq \int_0^{v_i} zx'(z) \Rightarrow$
$v_i x_i(v_i) \geq \int_0^{v_i} (zx(z))' - \int_0^{v_i} z'x(z) \Rightarrow$
$\int_0^{v_i} x(z) \geq 0$
το οποίο ισχύει αφού η $x(z)$ είναι προφανώς μη αρνητική.
Ισοδύναμα επιχρήματα στέκουν ακόμα και όταν η συνάρτηση x δεν είναι διαφορίσιμη αλλά μόνο βηματικά σταθερή.

□

1.4 Έννοιες της Ιεραρχίας της Ισορροπίας (Hierarchy of Equilibrium Concepts)

Σε αυτή την παράγραφο θα παρουσιάσουμε και θα συγκρίνουμε τις διαφορετικές έννοιες της ισορροπίας στα παίγνια, οι οποίες παρουσιάζονται με μεγαλύτερη λεπτομέρεια στο [8]. Ως τώρα επικεντρωθήκαμε σε δημοπρασίες όπου οι παίκτες είχαν κυρίαρχη στρατηγική. Αυτό δε συμβαίνει πάντα, διότι σε πολλά παίγνια δεν υπάρχει κυρίαρχη στρατηγική. Θα επεκτείνουμε αυτό το πλήθος των ¨λογικών' αποτελεσμάτων επιτρέποντας στους παίκτες να συγκλίνουν σε διαφορετικές μορφές ισορροπίας.

Θα δούμε ότι αυτές οι μορφές ισορροπίας είναι αρκετά χρήσιμες στη μορφή της δημοπρασίας. Αρχικά, επειδή οι έννοιες αυτές μας επιτρέπουν να ερμηνεύσουμε το αποτέλεσμα με διάφορους τρόπους και να καταλήξουμε σε διαφορετικά συμπεράσματα. Δευτέρον, επειδή η δυναμική φύση των δημοπρασιών δε μας επιτρέπει να καταλήξουμε σε διαφορετικές συνθήκες όπου η σύγκλιση σε ισορροπία είναι πιο ¨λογική¨.

Προτού προχωρήσουμε στην ανάλυση διαφορετικών εννοιών ισορροπίας, θα παρουσιάσουμε τις παραμέτρους του παίγνιου στις οποίες θα τις ορίσουμε.

Παίγνιο Ελαχιστοποίησης Κόστους (Cost Minimization Game) :

1. πεπερασμένος αριθμός k παικτών
2. πεπερασμένη στρατηγική S_i για κάθε παίκτη
3. συνάρτηση κόστους $C_i : S_1 \times S_2 \times, ..., \times S_k \to R$ για κάθε παίκτη
   Με s θα συμβολίσουμε το αποτέλεσμα του παιγνίου (δηλαδή $s \in S_1 \times$

$S_2 \times, ..., \times S_k)$ και για κάθε παίκτη i η στρατηγική του στο παίγνιο συμβολίζεται με s_i και η στρατηγική όλων των υπόλοιπων παικτών συμβολίζεται με s_{-i}

Τώρα είμαστε έτοιμοι να παρουσιάσουμε τις έννοιες της ισορροπίας. Η πρώτη έννοια της ισορροπίας εισήχθηκε από τον Nash στο [9]

Δεφινιτιον 1.4.1. ***Ισορροπία κατά Nash σε Αγνές Στρατηγικές (Pure Nash Equilibrium)***
Το στρατηγικό προφίλ s ενός παιγνίου ελαχιστοποίησης κόστους είναι μια αγνή ισορροπία Nash, αν για κάθε παίκτη i και για κάθε μονομερή απόκλιση $s_i^{'} \in S_i$:
$C_i(s) \leq C_i(s_i^{'}, s_{-i})$

Ο παραπάνω ορισμός δηλώνει με απλά λόγια ότι το αποτέλεσμα s αποτελεί αγνή ισορροπία Nash αν κανένας παίκτης δε θέλει να αλλάξει τη στρατηγική του, υποθέτοντας ότι οι υπόλοιποι παίκτες θα κάνουν το ίδιο. Αν και αυτός είναι ένας πολύ βασικός ορισμός, δεν ισχύει πάντα. Επομένως, τώρα είμαστε υποχρεωμένοι να επεκτείνουμε τον ορισμό μας επιτρέποντας τυχαιοποίηση πάνω στις στρατηγικές.

Δεφινιτιον 1.4.2. ***Ισορροπία κατά Nash σε Μικτές Στρατηγικές (Mixed Nash Equilibrium)***
Οι κατανομές $\sigma_1, \sigma_2, ..., \sigma_k$ στα στρατηγικά σετ $S_1, S_2, ..., S_k$ ενός παιγνίου ελαχιστοποίησης κόστους αποτελούν μικτή ισορροπία κατά Nash, αν για κάθε παίκτη i και για κάθε μονομερή απόκλιση $s_i^{'} \in S_i$:
$E_{s \sim \sigma}\left[C_i(s)\right] \leq E_{s \sim \sigma}\left[C_i(s_i^{'}, s_{-i})\right]$
όπου σ είναι η κατανομή-γινόμενο $\sigma_1 \times \sigma_2 \times \times ... \times \sigma_k$

Αυτός ο ορισμός είναι ίδιος με τον ορισμό της αγνής ισορροπίας κατά Nash, εκτός από το γεγονός ότι επιτρέπει να έχουμε κατανομή στις στρατηγικές και να παίζουμε σύμφωνα με αυτή την κατανομή. Τα καλά νέα είναι ότι σε παίγνια ελαχιστοποίησης κόστους, όπως τα ορίσαμε προηγουμένως, μια τέτοια ισορροπία υπάρχει πάντα. Ωστόσο, φαίνεται ότι ο υπολογισμός μιας τέτοιας ισορροπίας είναι υπολογιστικά δύσκολος. Πώς μπορούμε να υποθέσουμε ότι ένας παίκτης θα φτάσει σε μια τέτοια ισορροπία, αν δεν μπορούμε ούτε καν να την υπολογίσουμε· Το πρόβλημα αντιμετωπίζεται, όπως πάντα, με τη διεύρυνση των σετ των πιθανών ισορροπιών και με το να ελπίζουμε ότι αυτό το νέο σετ θα είναι πιο εύκολο υπολογιστικά να βρεθεί.

Δεφινιτιον 1.4.3. ***Correlated Equilibrium***
Μία κατανομή σ πάνω στο σετ των αποτελεσμάτων $S_1 \times S_2 \times ... \times S_k$ ενός παιγνίου ελαχιστοποίησης κόστους αποτελεί correlated equilibrium αν για κάθε παίκτη i , κάθε στρατηγική s_i και κάθε μονομερής απόκλιση $s_i^{'} \in S_i$:
$E_{s\sim\sigma}\left[C_i(s)|s_i\right] \leq E_{s\sim\sigma}\left[C_i(s_i^{'}, s_{-i})|s_i\right]$

Παρατηρήστε ότι στον τελευταίο ορισμό της κατανομής σ δεν χρειάζεται αυτή να είναι κατανομή-γινόμενο. Επιπλέον, οι προσδοκίες είναι εξαρτημένες από την στρατηγική s_i.

Η διαίσθηση πίσω από αυτό τον περίεργο ορισμό είναι πολύ απλή. Υποθέστε ότι γνωρίζετε μία κοινή κατανομή σ και μία έμπιστη τρίτη πηγή σας προτείνει την στρατηγική που πρέπει να ακολουθήσετε σύμφωνα με αυτή την στρατηγική, τότε αν η σ είναι correlated equilibrium, πρέπει να ακολουθήσετε τη συμβουλή.

Για να γίνει ακόμα πιο σαφές, σκεφτείτε απλά τους φωτεινούς σηματοδότες. Γνωρίζουμε ότι η κατανομή από την οποία οι φωτεινοί σηματοδότες παίρνουν απόφαση για το χρώμα που θα δείξουν δίνει πράσινο στον ένα δρόμο και κόκκινο στον άλλο.

Αυτή η νέα μορφή ισορροπίας είναι εύκολο να ελεγχθεί, με την έννοια ότι υπάρχει και μπορεί να υπολογιστεί αποδοτικά. Θα εισάγουμε την τελευταία μορφή ισορροπίας, η οποία αποτελεί περαιτέρω γενίκευση της έννοιας της ισορροπίας.

Δεφινιτιον 1.4.4. ***Coarse Correlated Equilibrium***
Μία κατανομή σ πάνω στο σετ των αποτελεσμάτων $S_1 \times S_2 \times ... \times S_k$ ενός παιγνίου ελαχιστοποίησης κόστους αποτελεί coarse correlated equilibrium αν για κάθε παίκτη i και για κάθε μονομερή απόκλιση $s_i^{'} \in S_i$:
$E_{s\sim\sigma}\left[C_i(s)\right] \leq E_{s\sim\sigma}\left[C_i(s_i^{'}, s_{-i})\right]$

Ας σημειωθεί ότι αυτή η μορφή ισορροπίας αποτελεί γενίκευση της προηγούμενης, με την έννοια ότι δεν απαιτεί προστασία όταν το προτεινόμενο αποτέλεσμα είναι γνωστό. Το μόνο πράγμα που γνωρίζει ο agent είναι η κατανομή από την οποία επιλέγονται τα αποτελέσματα. Φαίνεται ότι και αυτή η ισορροπία μπορεί να εξηγηθεί με το παράδειγμα των φωτεινών σηματοδοτών. Ωστόσο, φανταστείτε τώρα ότι πρέπει να αποφασίσουμε πριν την απόφαση του φωτεινού σηματοδότη, αν θα ακολουθήσουμε την κατανομή του. Αυτό είναι αρκετά διαφορετικό. Στην προηγούμενη

έννοια ισορροπίας, ο φωτεινός σηματοδότης μας συμβούλευε για το τι πρέπει να κάνουμε και φαίνεται ότι η καλύτερη πρακτική που μπορούσαμε να ακολουθήσουμε γνωρίζοντας το αποτέλεσμα ήταν να ακολουθήσουμε τη συμβουλή του. Ωστόσο, σε αυτό τον τύπο ισορροπίας πρέπει να αποφασίσουμε πριν το αποτέλεσμα της κατανομής γίνει γνωστό, αν θα ακολουθήσουμε αυτό που θα μας συμβολεύσει ο φωτεινός σηματοδότης.

Για να ολοκληρώσουμε αυτή την ενότητα, παρουσιάζουμε ένα παράδειγμα του Tim Roughgarden στο [8] που δείχνει όλες τις έννοιες ισορροπίας μαζί και διαφωτίζει τη σχέση τους ως αυστηρά αυξανόμενες σειρές απο σετ.

Εξαμπλε Ας υποθέσουμε ότι έχουμε ένα γράφο με δύο κόμβους, έναν κόμβο πηγής s και έναν προορισμού t. Ο γράφος έχει 6 παράλληλες ακμές. Υπάρχουν 4 παίκτες που διαλέγουν μία ακμή για να ¨ταξιδέψουν' από τον s στον t. Το κόστος είναι ανάλογο του πόσοι άλλοι παίκτες χρησιμοποιούν την ίδια ακμή για να ταξιδέψουν. Για παράδειγμα, όταν 3 από τους 4 παίκτες επιλέγουν την ίδια ακμή, τότε ο καθένας από αυτούς θα έχει κόστος ίσο με 3. Όταν ένας παίκτης είναι ο μόνος που χρησιμοποιεί μια συγκεκριμένη ακμή, τότε θα έχει κόστος 1 κλπ. Ας παρουσιάσουμε μια ισορροπία για κάθε τύπο.

(α′) Τα αποτελέσματα της $\binom{6}{4}$, στην οποία κάθε παίκτης διαλέγει διαφορετική ακμή για να ταξιδέψει αποτελούν προφανώς μια ισορροπία Nash.

(β′) Αν κάθε παίκτης επιλέξει ομοιόμορφα και ανεξάρτητα μία ακμή, τότε έχει αναμενόμενο κόστος ίσο με $\frac{3}{2}$. Το γινόμενο αυτών των ομοιόμορφων κατανομών είναι μια μικτή ισορροπία Nash, επειδή κανένας παίκτης δεν έχει συμφέρον να παρεκκλίνει προς μια άλλη κατανομή.

(γ′) Ας υποθέσουμε τώρα ότι η κατανομή πάνω στα αποτελέσματα είναι ομοιόμορφη πάνω στο εξής σετ: μία ακμή με δύο παίκτες και δύο ακμές με έναν παίκτη η κάθε μία. Τότε, παίζοντας ανάλογα με την κατανομή γνωρίζοντας ποιο είναι το αποτέλεσμα μας κάνει να έχουμε αναμενόμενο κόστος $\frac{3}{2}$. Παίζοντας με άλλη τακτική συνεχίζει να μας δίνει το ίδιο αναμενόμενο κόστος. Οπότε, η συσχετισμένη (correlated) κατανομή αποτελεί μία correlated ισορροπία.

(δ′) Η ομοιόμορφη κατανομή πάνω στο υποσύνολο των προηγούμενων αποτελεσμάτων, στο οποίο οι ακμές διαλέγονται είτε από το σετ $\{0, 1, 2\}$ ή από το σετ $\{3, 4, 5\}$ είναι μία coarse correlated ισορροπία, που κάνει κάθε παίκτη να έχει κόστος $\frac{3}{2}$. Σημειώστε ότι αυτή δεν είναι μία correlated ισορροπία, αφού η γνώση του αποτελέσματος θα μας οδηγήσει στο να αλλάξουμε τνη απόφασή μας σε μία τυχαία ακμή του άλλου σετ και να έχουμε κόστος ίσο με 1.

Το προηγούμενο παράδειγμα υπογραμμίζει ότι τα σετ της ισορροπίας σχηματίζουν μια σειρά από αυστηρά αυξανόμενα σετ. Επιπλέον, όταν κάνουμε την υπόθεση για το πού θα συγκλίνουν οι παίκτες, πρέπει να είμαστε προσεκτικοί σχετικά με δύο παραμέτρους, την ευκολία να φτάσει μια ισορροπία και το πόσο εύκολο είναι να υλοποιηθούν κατανεμημένοι αλγόριθμοι για να επιτευχθεί αυτή η ισορροπία.

Κεφάλαιο 2

Online Διαφημιστικές Δημοπρασίες

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το πρωταρχικά χρησιμοποιούμενο μοντέλο για online διαφημιστικές δημοπρασίες. Στη συνέχεια, θα συγκρίνουμε τι υποδεικνύει το λήμμα του Myerson και τι υλοποιείται στην πραγματικότητα.

2.1 Εισαγωγή

Οι online διαφημιστικές δημοπρασίες είναι δημοπρασίες που χρησιμοποιούνται για να πουλήσουν χώρο από μια αναζήτηση (search query) στο internet σε μια εταιρεία που τη χρησιμοποιεί για διαφήμιση. Για να πεισθεί ο αναγνώστης για τη σημασία της μελέτης των συγκεκριμένων μηχανισμών, είναι σημαντικό να αναφέρουμε ότι ένα μεγάλο μέρος των εσόδων της Yahoo και της Google παράγεται από αυτές τις διαφημίσεις. Σε αριθμούς, το 2005 το συνολικό κέρδος της Google ανερχόταν σε 6.14 δις δολλάρια. Πάνω από το 98 τις εκατό του ποσού αυτού προήλθε από online διαφημιστικές δημοπρασίες. Από την άλλη πλευρά, τα έσοδα της Yahoo το 2005 ήταν περίπου 5.26 δις δολλάρια και εκτιμάται ότι πάνω από τα μισά από αυτά είχαν άμεση σχέση με αυτού του είδους τη διαφήμιση.[10]

Επιπρόσθετα, τα τελευταία χρόνια, η παγκόσμια online διαφημιστική αγορά αυξάνεται συνεχώς `ηττπ://ωωω.ωορδστρεαμ.ςομ/αρτιςλες/γοογλε-εαρνινγσ`.

Προτού προχωρήσουμε στον ορισμό του μοντέλου, θα περιγράψουμε πώς δουλεύει αυτό το σύστημα. Κάθε φορά που ένας χρήστης του in-

ternet πληκτρολογεί μία search query, διεξάγεται μία online δημοπρασία σε πραγματικό χρόνο. Παράλληλα με τα αποτελέσματα της search query, εμφανίζονται κάποιες διαφημίσεις. Πίσω από κάθε μία από αυτές τις διαφημίσεις βρίσκεται μία εταιρεία που ποντάρει πάνω στη συγκεκριμένη search query για να έχει την πιθανότητα να δείξει τη διαφήμισή της στη σελίδα αποτελεσμάτων του χρήστη. Το ποντάρισμα που γίνεται από την εταιρεία ανταναλά το μέγιστο ποσό που είναι πρόθυμη να πληρώσει αν πατηθεί αυτή η διαφήμιση. Άρα, κάθε φορά που ο χρήστης πατάει πάνω σε μια συγκεκριμένη διαφήμιση, η εταιρεία που τη δημιούργησε χρεώνεται με ένα ποσό μικρότερο ή ίσο με το ποντάρισμά της. Επομένως, η εταιρεία ποντάρει τη μέγιστη πληρωμή ανά κλικ. Προφανώς, πάνω από μία θέσεις στην σελίδα αποτελεσμάτων πωλούνται κάθε φορά. Όπως είναι αναμενόμενο, όσο πιο ψηλά εμφανίζεται η διαφήμιση, τόσο πιο πιθανό είναι ότι ο χρήστης θα κάνει κλικ σε αυτή. Ο μηχανισμός πρέπει τότε να διαλέξει δύο πράγματα. Πρώτον, τον κανόνα διανομής και δεύτερον, τον κανόνα πληρωμής.

Ας δώσουμε ένα συγκεκριμένο παράδειγμα. Φανταστείτε μία εταιρεία που πουλάει λουλούδια. Κάθε φορά που ένας χρήστης πληκτρολογεί στη μηχανή αναζήτησης τη φράση ¨αγορά λουλουδιών¨, η εταιρεία κάνει ένα ποντάρισμα. Τότε, ο μηχανισμός διαλέγει σε ποια θέση θα δείξει την διαφήμιση της εταιρείας λουλουδιών. Αν ο χρήστης κάνει κλικ σε αυτή τη διαφήμιση, τότε χρεώνεται η εταιρεία.

2.2 Περιγραφή Μοντέλου

Κάθε παίκτης έχει μια πιθανότητα να κάνει κλικ γ_i. Κάθε θέση έχει μια πιθανότητα κλικαριστεί (να την διαλέξει ο χρήστης) α_i. Άρα, η πιθανότητα του παίκτη i που βρίσκεται στη θέση j να πατηθεί δίνεται από το p_{ij} και είναι ίση με το γινόμενο $\alpha_i\gamma_j$. Το σύστημα ζητάει από τους παίκτες να κάνουν ένα ποντάρισμα b_i.Τότε μαζεύονται τα ποναρίσματα και ο μηχανισμός επιλέγει το πώς θα δοθούν οι θέσεις στους παίκτες χρησιμοποιώντας έναν συγκεκριμένο κανόνα διανομής. Κάθε φορά που κλικάρεται μια συγκεκριμένη διαφήμιση, το σύστημα χρεώνει στον παίκτη ένα ποσό μικρότερο ή ίσο με το ποντάρισμά του, χρησιμοποιώντας μία συνάρτηση πληρωμής.

Συμβολίζουμε με:

(α′) m τον αριθμό των θέσεων που πωλούνται κάθε φορά

(β′) n τον αριθμό των παικτών

(γ′) u_iτην αξία του παίκτη i ανά κλικ

(δ′) $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$ το διάνυσμα των συντελεστών της θέσης, χωρίς βλάβη της γενικότητας θα υποθέσουμε ότι $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_m$

(ε′) $\gamma = (\gamma_1, \gamma_2, ..., \gamma_n)$ το διάνυσμα των συντελεστών του παίκτη

(ϛ′) $p_{ij} = \alpha_i \gamma_j$ η πιθανότητα του να κλικαριστεί ο παίκτης i όταν τοποθετείται στη θέση j

(ζ′) r η ελάχιστη τιμή απόκτησης

(η′) $b = (b_1, b_2, ..., b_n)$ το διάνυσμα των πονταρισμάτων των παικτών

Ακολουθώντας τα παραπάνω, εισάγουμε κάποιους επιπλέον συμβολισμούς. Έστω $\sigma_i(b)$ η θέση όπου ο παίκτης i διανέμεται σύμφωνα με το προφίλ πονταρίσματος b. Επίσης ας θέσουμε $\pi(j, b)$ το δείκτη του παίκτη που βρίσκεται στη θέση j με προφίλ πονταρίσματος b. Επιπρόσθετα, έστω $c_{ij}(b)$ το κόστος ανά κλικ του παίκτη i όταν βρίσκεται στη θέση j. Το αναμενόμενο κόστος του παίκτη i με προφίλ πονταρίσματος b είναι τότε:

$$C_i(b) = \alpha_{\sigma_i(b)} \gamma_i c_{i\sigma_i(b)}(b) \tag{2.1}$$

Επιπλέον, η πιθανότητα του κλικ ως συνάρτηση του προφίλ πονταρίσματος b είναι:

$$P_i(b) = \alpha_{\sigma_i(b)} \gamma_i \tag{2.2}$$

Για το κέρδος του χρήστη, θα χρησιμοποιήσουμε όπως πάντα το σχεδόνγραμμικό μοντέλο, οπότε το αναμενόμενο κέρδος του παίκτη i με προφίλ πονταρίσματος b συμβολίζεται $U_i(b)$ και δίνεται από την εξίσωση:

$$U_i(b) = u_i P_i(b) - C_i(b) \tag{2.3}$$

2.3 Το λήμμα του Myerson σε σχέση με την GSP δημοπρασία (Generalised Second Price auction)

Σε αυτή την ενότητα, θα παρουσιάσουμε δυο βασικές υλοποιήσεις του μοντέλου ονλινε διαφημιστικών δημοπρασιών. Την υλοποίηση που χρησιμοποιεί το λήμμα του Myerson 1.2 για να υπολογίσει τη συνάρτηση

πληρωμής και τον πιο ευρέως χρησιμοποιούμενο μηχανισμό, τη γενικευμένη δημοπρασία Δεύτερης Τιμής (Generalised Second Price auction). Όπως πάντα, οι συναρτήσεις που πρέπει να υλοποιηθούν είναι η συνάρτηση διανομής και η συνάρτηση πληρωμής.

Διανομή Η συνάρτηση διανομής που χρησιμοποιείται συνήθως από τους δύο μηχανισμούς είναι η πιο προφανής. Δεδομένων των θέσεων, από τον μεγαλύτερο (α_1) ως τον μικρότερο (α_m) σε αξία στους m πλειοδότες και υποθέτοντας ότι τουλάχιστον m παίκτες έκαναν ένα ποντάρισμα μεγαλύτερο από την ελάχιστη τιμή απόκτησης r. Διαφορετικά, έστω ότι έχουν τις $k \leq m$ πρώτες θέσεις οι k βιδδερς που ποντάρουν περισσότερο από την ελάχιστη τιμή απόκτησης και δεν διανέμουμε τις υπόλοιπες θέσεις. Ας σημειωθεί ότι αυτή η συνάρτηση διανομής είναι αύξουσα για κάθε παίκτη i. Επομένως, από το λήμμα του Myerson γνωρίζουμε ότι μπορεί να υλοποιηθεί με έναν DSIC μηχανισμό.

Η παραγόμενη δημοπρασία του Myerson Για λόγους απλότητας στην κατανόηση, θα υποθέσουμε ότι στο διάνυσμα πονταρισμάτων b, τα πονταρίσματα δεν αυξάνονται (δηλαδή $b_1 \geq b_2 \geq ... \geq b_n$), οπότε, χρησιμοποιώντας το λήμμα του Myerson, μπορούμε να δημιουργήσουμε την συνάρτησης πληρωμής ανά κλικ, που είναι η εξής:

$$c_{i\sigma_i(b)}(b) = \sum_{j=i}^{min(m,biddersallocated)} \frac{a_j - a_{j+1}}{a_i} b_{j+1} \tag{2.4}$$

Ας σημειωθεί ότι η συνάρτηση πληρωμής που παράχθηκε από το λήμμα του Myerson υποδεικνύει ότι ο ι-οστός παίκτης θα πρέπει να πληρώνει κάθε φορά που ένας κατάλληλος κυρτός συνδυασμός μικρότερων πονταρισμάτων κλικάρεται. Αυτή η δημοπρασία είναι προφανώς DSIC, μιας και η συνάρτηση διανομής είναι μη-φθίνουσα για κάθε παίκτη και η συνάρτηση πληρωμής είναι αυτή που παράγεται από το λήμμα του Myerson.

Από την τελευταία εξίσωση, μπορούμε εύκολα να συμπεράνουμε ότι το μέσο κόστος του παίκτη i με προφίλ πονταρίσματος b (συνεχίζοντας να υποθέτουμε ότι τα πονταρίσματα είναι σε μη αύξουσα σειρα) είναι:

$$C_i(b) = \sum_{j=i}^{min(m,biddersallocated)} \gamma_i(a_j - a_{j+1}) b_{j+1} \tag{2.5}$$

GSP Η δημοπρασία GSP αποτελεί το πιο διαδεδομένο μοντέλο για ονλινε διαφημιστικές δημοπρασίες. Ο κύριος λόγος που χρησιμοποιείται αντί της βέλτιστης (κατά DSIC) παραγόμενης από το λήμμα του Myerson δημοπρασίας είναι η απλότητά της. Επιπλέον, υπάρχουν και άλλες καλές ιδιότητες αυτού του μηχανισμού που τον κάνουν εξαιρετικά χρήσιμο. Προτού προχωρήσουμε στη σύγκριση των δύο μηχανισμών, πρέπει να περιγράψουμε την υλοποίηση GSP. Όπως υποδεικνύει το όνομα, αυτός ο τύπος δημοπρασίας γενικεύει την ιδέα Δεύτερης Τιμής του Vickrey [6] με τέτοιο τρόπο που αν ένας παίκτης τοποθετηθεί στην j-οστή θέση και μετά κλικαριστεί, θα πρέπει να πληρώσει το ελάχιστο χρηματικό ποσό που θα έπρεπε να ποντάρει για να διατηρήσει τη θέση του, επομένως το επόμενο κατά σειρά ποντάρισμα. Η συνάρτηση κόστους ανά κλικ είναι τότε (εδώ οι μορφές είναι αρκετά απλουστευμένες και δε χρειάζεται να υποθέσουμε ότι το διάνυσμα των πονταρισμάτων είναι μη-αύξον):

$$c_{ij}(b) = max\{b_{\pi_{j+1}}, r\}\mathbb{1}\{b_i \geq r\} \tag{2.6}$$

Επομένως, το αναμενόμενο κόστος του παίκτη i είναι:

$$C_i(b) = \alpha_{\sigma_i(b)}\gamma_i max\{b_{\pi_{\sigma_i(b)+1}}, r\}\mathbb{1}\{b_i \geq r\} \tag{2.7}$$

Αν και το γεγονός ότι η συνάρτηση πληρωμής της ΓΣΠ δεν αποτελεί DSIC μηχανισμό φαίνεται να αντιτίθεται στη διαίσθησή μας, αποτελεί τη θλιβερή πραγματικότητα. Γνωρίζουμε ότι η συνάρτηση πληρωμής που δίνεται από το λήμμα του Myerson είναι η μόνη που δημιουργεί DSIC μηχανισμό. Επί παραδείγματι, ας υποθέσουμε ότι υπάρχουν τρεις παίκτες με αξίες $v_1 = 10$, $v_2 = 8$,$v_1 = 3$ και υπάρχουν δύο θέσεις διαθέσιμες, η πρώτη με $\alpha_1 = 1$ και η δεύτερη με $\alpha_2 = 0.99$ (τα γ παραλείπονται). Τότε, αν κάθε παίκτης ποντάρει με ειλικρίνεια, το κέρδος του πρώτου παίκτη είναι $1(10-8) = 2$. Αλλάζοντας το ποντάρισμά του σε 4, το κέρδος του θα αυξανόταν μιας και $0.99(10-3) = 6.93 > 2$. Άρα, η ειλικρίνεια δεν αποτελεί κυρίαρχη στρατηγική για τον πρώτο παίκτη και, επομένως, η δημοπρασία δεν είναι DSIC.

Εδώ δημιουργείται μια ενδιαφέρουσα ερώτηση. Αν το διάνυσμα των πονταρισμάτων σταθεροποιείται σε ένα συγκεκριμένο διάνυσμα b, τότε

ποιες ιδιότητες θα πρέπει να έχει αυτό το διάνυσμα· Η ερώτηση αυτή απαντήθηκε από τους Ostrovsky και άλλοι στο [10] και από τον Variant στο [2]. Προφανώς, η πρώτη συνθήκη είναι ότι πρέπει να ικανοποιεί την απαίτηση για ισορροπία κατά Nash. Αυτό σημαίνει ότι κάθε παίκτης αποκρίνεται βέλτιστα στα άλλα πονταρίσματα. Υποθέτοντας για άλλη μια φορά για απλότητα ότι το διάνυσμα των πονταρισμάτων είναι μη-αύξον και,χωρίς βλάβη της γενικότητας, ότι όλα τα πονταρίσματα είναι μεγαλύτερα από την ελάχιστη τιμή απόκτησης, τότε το προφίλ πονταρίσματος b αποτελεί μία ισορροπία Nash. ($a_k = 0$ για $k > m$) ανν $\forall i$:

$$\alpha_i \gamma_i (v_i - b_{i+1}) \geq \alpha_j \gamma_i (v_i - b_{j+1}) \quad j > i \tag{2.8}$$

$$\alpha_i \gamma_i (v_i - b_{i+1}) \geq \alpha_j \gamma_i (v_i - b_j) \quad j < i \tag{2.9}$$

Η παρουσίαση ενός παραδείγματος θα μας δώσει περισσότερα στοιχεία σχετικά με το πώς πρέπει να μοιάζει ένα διάνυσμα ισορροπίας. Φανταστείτε ότι υπάρχουν δύο παίκτες που ποντάρουν για ένα αντικείμενο. Ο πρώτος παίκτης έχει αξία ίση με 100 και ο δεύτερος αξία ίση με 40. Το διάνυσμα πονταρισμάτων (30,120) αποτελεί ισορροπία κατά Nash για το one shot παίγνιο, ωστόσο σε επαναλαμβανόμενο παίγνιο δεν είναι λογικό να υποθέσουμε ότι τα διανύσματα των πονταρισμάτων θα σταθεροποιηθούν σε αυτή την τιμή, καθώς ο πρώτος παίκτης είναι πιθανό να αυξήσει το ποντάρισμά του και να επιβάλει στον δεύτερο παίκτη να εγκαταλείψει τη δημοπρασία. Ακολουθώντας αυτή την ιδέα, θα περιοριστούμε σε μια πιο ειδική περίπτωση ισορροπίας που ονομάζεται τοπικά envy-free (χωρίς φθόνο) ισορροπία. Οι ισορροπίες αυτού του τύπου χαρακτηρίζονται από την εξής σχέση:
$\forall i$

$$\alpha_i \gamma_i (v_i - b_{i+1}) \geq \alpha_j \gamma_i (v_i - b_{j+1}) \tag{2.10}$$

Έχουμε ισχυροποιήσει τη δεύτερη ανισότητα της γενικευμένης ισορροπίας κατά Nash. Αυτή η σχέση δε μας δίνει τον ακριβή χαρακτηρισμό της ισορροπίας στην οποία θα φτάσουν οι παίκτες, καθώς υπάρχουν πολλά προφίλ πονταρίσματος που ικανοποιούν την εξίσωση. Ωστόσο, αν το προφίλ πονταρίσματος σταθεροποιείται κάπου, τότε θα πρέπει να έχουμε φτάσει μία τοπικά envy-free ισορροπία.

Με την υπόθεση ότι οι παίκτες θα φτάσουν σε μία τοπικά envy-free ισορροπία, η GSP δημοπρασία έχει κάποιες ενδιαφέρουσες ιδιότητες. Παρουσιάζουμε την πιο σημαντική παρακάτω.

Θεορεμ 2.3.1.
Αν ο αριθμός των διαφημιστικών εταιρειών είναι μεγαλύτερος από τον αριθμό των διαθέσιμων θέσεων, τότε τα έσοδα της δημοπρασίας οποιασδήποτε τοπικά envy-free δημοπρασίας υπό GSP δημοπρασία είναι τουλάχιστον ίσα με τα έσοδα της δημοπρασίας που παράχθηκε από τον Myerson, υπό την υπόθεση ότι οι παίκτες είναι ειλικρινείς.

Η απόδειξη του προηγούμενου θεωρήματος έγινε από τους Ostrovsky και άλλοι στο [10].

Τι μας δείχνει, λοιπόν, η προηγούμενη παράγραφος· Πολύ απλά, ότι όταν είσαι δημοπράτης και θέλεις να αυξήσεις τα έσοδά σου, τότε καλά θα κάνεις να χρησιμοποιήσεις την GSP δημοπρασία αντί για την δημοπρασία που είναι παραγόμενη από τον Myerson.

Είναι προφανές ότι αυτή είναι μία ελκυστική ιδιότητα για κάθε σχεδιαστή δημοπρασιών. Τόσο αυτή η ιδιότητα, όσο και η GSP δημοπρασία προτιμώνται για επιπρόσθετες ιδιότητες ευρωστίας που θα σημειώσουμε αργότερα. Επιπλέον, παρατηρήστε ότι από την πλευρά του παίκτη, η GSP δημοπρασία προφυλάσσει κατά κάποιον τρόπο την ιδιωτικότητά του, με το να μην αποκαλύπτει ευθέως στον δημοπράτη την προσωπική του αξία. Έτσι, ανάμεσα σε άλλα, η GSP δημοπρασία επιτυγχάνει τρεις σημαντικούς στόχους: προστατεύει την ιδιωτικότητα του κάθε παίκτη, φέρνει μεγαλύτερα έσοδα και είναι εξαιρετικά απλή.

Ωστόσο, η συνάρτηση GSP πληρωμών είναι διαφορετική από την μοναδική συνάρτηση που υποδεικνύει το λήμμα του Myerson για να αποκτηθεί DSIC μηχανισμός. Επομένως, δεν υπάρχει προφανής τρόπος να ποντάρουν οι παίκτες. Η ερώτηση του πώς πρέπει να ποντάρουν οι παίκτες είναι εξαιρετικά ενδιαφέρουσα και θα εξεταστεί σε ένα ολόκληρο κεφάλαιο.

Κεφάλαιο 3

Online Μάθηση

Στο προηγούμενο κεφάλαιο παρουσίασαμε τη δημοπρασία ΓΣΠ, που αποτελεί την πιο ευρέως χρησιμοποιούμενη μορφή ονλινε διαφημιστικής δημοπρασίας. Αυτή η δημοπρασία δεν είναι DSIC και δεν παρέχει στους παίκτες προφανή τακτική πονταρίσματος. Επομένως, ο κύριος σκοπός αυτού του κεφαλαίου είναι να απαντήσει στο ερώτημα του πώς θα έπρεπε να ποντάρουν αυτοί οι παίκτες.

Στο πρώτο τμήμα του κεφαλαίου, θα παρουσιάσουμε τη μορφή ανανέωσης πολλαπλών βαρών (multiplicative weights update algorithm) που εισήχθη από Αρορα και άλλους στο [11]. Η ιδέα του συγκεκριμένου αλγορίθμου παρουσιάστηκε πολλές φορές σε διαφορετικούς τομείς. Αρχικά, ένας επιπλέον κανόνας ανανέωσης προτάθηκε στο [12] και στη συνέχεια στο [13]. Επιπλέον, στην μαςηινε learning βιβλιογραφία, οι Freud και Schapire πρότειναν των πολύ γνωστό Adaboost αλγόριθμο στο [14]. Οι αρχικές ιδέες αναπτύχθηκαν στην κοινότητα της θεωρίας παιγνίων. Η πρώτη υλοποίηση αυτής της ιδέας στο πεδίο των οικονομικών παρουσιάστηκε στο [15] και μετά εισήχθη ξανά στο [16].

Στο δεύτερο τμήμα του κεφαλαίου, παρουσιάζουμε ένα γενικό Online Convex Optimization πλαίσιο και την ανάλυση των βασικών του αλγορίθμων. Τέλος, περιγράφουμε τους βασικούς αλγορίθμους που γενικεύουν την Multiplicative Weight Update ιδέα για νον φυλλ ινφορματιον παραμέτρους.

Η εφαρμογή αυτών των αλγορίθμων στο πλαίσιο μας είναι σαφής. Ισχυριζόμαστε ότι το κύριο πρόβλημα είναι το πώς θα έπρεπε να ποντάρουν οι παίκτες σε όχι- DSIC μηχανισμούς. Αυτοί οι αλγόριθμοι μαθαίνουν βήμα-βήμα ποια είναι η βέλτιστη τακτική για κάθε παίκτη.

3.1 Εισαγωγικοί Αλγόριθμοι

εδώ αλλάζουν ολά

3.1.1 Multiplicative Weights Updates

Ο Multiplicative Weights Updates Αλγόριθμος δεν είναι απλά ένας αλγόριθμος, αλλά αποτελεί μια γενική μέθοδο επίλυσης προβλημάτων. Η ιδέα πίσω από αυτόν είναι ότι, οποτεδήποτε έχουμε ένα σετ διαφορετικών πιθανών πράξεων, το να διατηρούμε μια κατανομή γύρω από τις πράξεις μας επιτρέπει στο μέλλον να επιλέξουμε μία πράξη σύμφωνα με την συσσωρευμένη εμπειρία.

Κατά μία έννοια, έτσι δρουν οι άνθρωποι. Όταν υπάρχει ένα πρόβλημα, ο καθένας εκτιμά τις διαφορετικές στρατηγικές που μπορεί να ακολουθήσει και η εκτίμηση αυτών των στρατηγικών βασίζεται σε κάποια συσσωρευμένη εμπειρία των αποτελεσμάτων που έφεραν αυτές οι στρατηγικές στο παρελθόν. Φυσικά, στρατηγικές που είχαν καλά αποτελέσματα στο παρελθόν είναι, τουλάχιστον στο μυαλό μας, πιο πιθανό να φέρουν ξανά καλά αποτελέσματα στο μέλλον.

Θα ακολουθήσουμε τη γενική μορφή που προτάθηκε από τον Arora και άλλους στο [11] Ας ορίσουμε κάποιες χρήσιμες ποσότητες:

(α′) $\mathbf{A} = \{1, ..n\}$ το σετ των πιθανών στρατηγικών

(β′) $\mathbf{\Pi}$ το σετ των πιθανών αποτελεσμάτων

(γ′) $\mathbf{M}$ ο πίνακας κόστους, τέτοιος ώστε το $\mathbf{M(\iota,\vartheta)}$ αντιπροσωπεύει το κόστος του να ακολουθήσω μία στρατηγική ι όταν το αποτέλεσμα είναι ϑ.

(δ′) $\forall i \in A, \forall j \in P\ M(i,j) \in [-l, r,]\ ,where\ l \leq r$

(ε′) $w_i^t\ , \forall i \in A$ το βάρος της στρατηγικής ι στη χρονική στιγμή τ

(ϛ′) $p_i^t\ , \forall i \in A$ η πιθανότητα επιλογής της στρατηγικής ι στη χρονική στιγμή τ

(ζ′) $D^t = \{p_1^t, p_2^t, .., p_n^t\}$ η κατανομή των στρατηγικών στη χρονική στιγμή τ

Αλγοριτημ 1 ΜΩΥ αλγοριτημ

Αρχικοποίηση $w_1^t = 1, \quad \forall i \in A$
φορ $t = 1 \quad to \quad T$ **δο**
    διάλεξε μία στρατηγική $i_t \sim D^t$
    υπολόγισε το αποτέλεσμα $j^t \in P$
    φορ $i = 1 \quad to \quad n$ **δο** ▷ **weigths update**
        ιφ $M(i,j) \geq 0$ **τηεν**
            $w_i^{t+1} = w_i^t(1-\epsilon)^{\frac{M(i,j^t)}{r}}$
        ελσε ιφ $M(i,j) < 0$ **τηεν**
            $w_i^{t+1} = w_i^t(1+\epsilon)^{-\frac{M(i,j^t)}{r}}$
    $\Phi^{t+1} = \sum_{i=1}^{n} w_i^{t+1}$
    φορ $i = 1 \quad to \quad n$ **δο** ▷ κανονικοποίησε για να δημιουργήσεις μία κατανομή
        $p_i^{t+1} = \frac{w_i^{t+1}}{\Phi^{t+1}}$
    $D^{t+1} = \{p_1^{t+1}, p_2^{t+1}, .., p_n^{t+1}\}$

Διαισθητική εξήγηση Προτού προχωρήσουμε στην ανάλυση του αλγορίθμου, αξίζει να δώσουμε μία εξήγηση για τον κανόνα ανανέωσης βαρών.

Το $M(i,j^t) \geq 0$ σημαίνει ότι στη συγκεκριμένη χρονική στιγμή τ, το φυσικό αποτέλεσμα είναι j^t και η στρατηγική μας i έχει απώλεια μεγέθους $M(i,j^t)$. Άρα, τι πρέπει να κάνουμε· Μα φυσικά, να μειώσουμε την πιθανότητα επιλογής της στρατηγικής i στην επόμενη επανάληψη. Το να επιλέξουμε $\epsilon \in (0,1)$ κάνει αυτό που θέλουμε.

Ακολουθώντας την ίδια λογική, θα πρέπει να αυξήσουμε την πιθανότητα μιας στρατηγικής με αρνητική απώλειας $M(i,j^t)$ (που έχει δηλαδή θετικό κέρδος). Ο κανόνας ανανέωσης δρα όπως περιμέναμε.

Καθώς ο MWU αλγόριθμος είναι πιθανοτικός, το πλαίσιο με το οποίο θα συγκρίνουμε την απόδοση είναι το αναμενόμενο penalty. Άρα, ας συμβολίσουμε με $M(D^t,j^t) = \sum_{i=1}^{n} p_i^t M(i,j^t)$ το αναμενόμενο πεναλτψ του αποτελέσματος $j^t \in P$ της πιθανοτικής κατανομής D^t του αλγορίθμου μας τη χρονική στιγμή t.

Τηεορεμ 3.1.1 (Θεώρημα)**.** *ιφ* $\epsilon \leq \frac{1}{2}$ *τότε* $\forall i \in A$*:*
$\sum_t M(D^t,j^t) \leq \frac{rln(n)}{\epsilon} + (1+\epsilon) \sum_{t:M(i,j^t)\geq 0} M(i,j^t) + (1-\epsilon) \sum_{t:M(i,j^t)<0} M(i,j^t)$

Απόδειξη.

θα χρησιμοποιήσουμε τρεις βασικές ανισότητες

$(\alpha')\ (1-\epsilon)^x \leq (1-\epsilon x) \quad x \in [0,1]$
$(\beta')\ (1+\epsilon)^{-x} \leq (1-\epsilon x),\ x \in [0,1]$
$(\gamma')\ e^x \geq x+1,\ \forall x \in \mathrm{Re}$

από τον ϕ_t ορισμό έχουμε

$$\phi_{t+1} = \sum_{i=1}^{n} w_i^{t+1} = \sum_{i:M(i,j^t)\geq 0} w_i^t (1-\epsilon)^{\frac{M(i,j^t)}{r}} + \sum_{i:M(i,j^t)<0} w_i^t (1+\epsilon)^{-\frac{M(i,j^t)}{r}}$$
$$\leq$$
$$\leq \sum_{i:M(i,j^t)\geq 0} w_i^t (1-\epsilon\frac{M(i,j^t)}{r}) + \sum_{i:M(i,j^t)<0} w_i^t (1-\epsilon\frac{M(i,j^t)}{r}) =$$
$$= \sum_{i=1}^{n} w_i^t (1-\epsilon\frac{M(i,j^t)}{r}) = \sum_{i=1}^{n} w_i^t - \epsilon \sum_{i=1}^{n} w_i^t \frac{M(i,j^t)}{r} = \phi_t - \frac{\epsilon\phi_t}{r} \sum_{i=1}^{n} \frac{w_i^t}{\phi_t} M(i,j^t) =$$
$$\phi_t - \frac{\epsilon\phi_t}{r} \sum_{i=1}^{n} p_i^t M(i,j^t) = \phi_t - \frac{\epsilon\phi_t}{r} M(D^t,j^t) = \phi_t(1 - \frac{\epsilon}{r} M(D^t,j^t)) \leq$$

$$\leq \phi_t e^{-\frac{\epsilon}{r} M(D^t,j^t)}$$

με επαγωγή στον αριθμό των επαναλήψεων και από το γεγονός ότι $\phi_1 = n$ παίρνουμε εύκολα ότι

$$\phi_T \leq n e^{-\frac{\epsilon}{r} \sum_{t=1}^{T-1} M(D^t,j^t)} \tag{3.1}$$

Επιπρόσθετα, μιας και τα βάρη παραμένουν θετικά κατά τη διάρκεια όλου του αλγορίθμου, έχουμε ότι:

$$\phi_T \geq w_i^T = (1-\epsilon)^{\sum_{t:M(i,j^t)>0} \frac{M(i,j^t)}{r}} (1+\epsilon)^{\sum_{t:M(i,j^t)<0} -\frac{M(i,j^t)}{r}} \quad \forall i \in A \tag{3.2}$$

συνδύαζοντας τις δύο ανισότητες έχουμε ότι

$$n e^{-\frac{\epsilon}{r} \sum_{t=1}^{T-1} M(D^t,j^t)} \geq (1-\epsilon)^{\sum_{t:>0} \frac{M(i,j^t)}{r}} (1+\epsilon)^{\sum_{t:<0} -\frac{M(i,j^t)}{r}} \quad \forall i \in A \tag{3.3}$$

τώρα χρησιμοποιώντας τις ανισότητες

$(\alpha')\ ln(\frac{1}{1-\epsilon}) \leq \epsilon + \epsilon^2 \quad \forall \epsilon \in \left(0, \frac{1}{2}\right)$

$(\beta')\ ln(1+\epsilon) \geq \epsilon - \epsilon^2 \quad \forall \epsilon \in \left(0, \frac{1}{2}\right)$

καταλήγουμε εύκολα στην απόδειξη

□

ὅρολλαρψ 3.1.1.1. *Για* $T > 2ln(n)$ *και* $\epsilon = \sqrt{\frac{ln(n)}{2T}}$ *έχουμε ότι :*

$\frac{\sum_t M(D^t, j^t)}{T} \leq O(\frac{1}{\sqrt{T}}) + \frac{\sum_t M(i, j^t)}{T} \quad \forall i \in A$

Απόδειξη.
Παρατηρούμε πρώτα ότι

$$(1+\epsilon) \sum_{t:M(i,J^t)\geq 0} M(i, j^t) \leq \sum_{t:M(i,J^t)\geq 0} M(i, j^t) + \epsilon r T$$

$$(1-\epsilon) \sum_{t:M(i,J^t)< 0} M(i, j^t) \leq \sum_{t:M(i,J^t)\geq 0} M(i, j^t) + \epsilon l T$$

$l \leq r$

επίσης, από το προηγούμενο θεώρημα έχουμε ότι

$$\sum_t M(D^t, j^t) \leq \frac{r ln(n)}{\epsilon} + 2\epsilon r T + \sum_t M(i, j^t) \quad \forall i \in A$$

Τελικά αντικαθιστώντας το ϵ παίρνουμε την επιθυμητή έκφραση

□

Παρατηρήσεις: Αρχικά, θα πρέπει να αντιληφθούμε ότι τα παραπάνω θεωρήματα δεν κάνουν υποθέσεις για τη γνώση που θα είχε ένας αντίπαλος. Είναι πιθανό ότι ο αντίπαλος γνωρίζει την κατανομή μας στις στρατηγικές και επιλέγει μία περίσταση σύμφωνα με αυτό. Αυτή η ιδιότητα είναι πολύ σημαντική, ειδικά σε επαναλαμβανόμενα παίγνια όπως οι δημοπρασίες. Είναι εμφανές ότι, μετά από πολλές επαναλήψεις, τα πονταρίσματα μας παρέχουν ιδιωτικές πληροφορίες πάνω στους αντιπάλους αμς. Επομένως, ο αλγόριθμος μας προστατεύει εναντίον των πιο δυνατών αντιπάλων, οι οποίοι γνωρίζουν τα πάντα σχετικά με εμάς. Δεύτερον, η αρχική ομοιόμορφη κατανομή πάνω στις στρατηγικές αντανακλά την αρχική μας άγνοια. Τελικα, ίσως το πιο ενδιαφέρον αποτέλεσμα είναι ότι

η μέση διαφορά ανάμεσα στις απώλειες των στρατηγικών μας και τις απώλειες μιας συγκεκριμένης στρατηγικής εξαφανίζεται καθώς $O(\frac{1}{\sqrt{T}})$.

3.1.2 Αλγόριθμος Hedge

Ο αλγόριθμος Hedge αποτελεί μία ελαφρώς διαφορετική μορφή του αλγορίθμου ανανέωσης βαρών με το να παραγωγίζει τον κανόνα ανανέωσης. Προτάθηκε αρχικά από τους Freud ανδ Schapire στο [17]. Αν και δεν είναι τόσο γενικός, η απόδοση που επιτυγχάνεται για συγκεκριμένα παραδείγματα είναι παρόμοια, παρά το γεγονός ότι η ανάλυση είναι απλούστερη.

Παρουσιάζουμε πρώτα τον αλγόριθμο και στη συνέχεια προχωράμε στην τεχνική ανάλυση. Το πλαίσιο παραμένει το ίδιο, παρά το γεγονός ότι οι απώλειες είναι τώρα φραγμένες στο $[0, \infty)$. Άρα:

$M(i, j) \in [0, \infty) \quad \forall i \in A, j \in P$

Αλγοριτημ 2 Αλγόριθμος Hedge

Αρχικοποίηση $w_1^t = 1, \quad \forall i \in A$
φορ $t = 1 \quad to \quad T$ **δο**
  διάλεξε μία στρατηγική $i_t \sim D^t$
  υπολόγισε το αποτέλεσμα $j^t \in P$
  φορ $i = 1 \quad to \quad n$ **δο** ▷ **ανανέωση βαρών**
    $w_i^{t+1} = w_i^t e^{-M(i,j^t)}$
  $\Phi^{t+1} = \sum_{i=1}^{n} w_i^{t+1}$
  φορ $i = 1 \quad to \quad n$ **δο** ▷ κανονικοποίησε για να δημιουργήσεις κατανομή
    $p_i^{t+1} = \frac{w_i^{t+1}}{\Phi^{t+1}}$
  $D^{t+1} = \{p_1^{t+1}, p_2^{t+1}, .., p_n^{t+1}\}$

Τηεορεμ 3.1.2. *Έστω* $M(D^t, j^t)^2 = \sum_{i=1}^{n} p_i^t M(i, j^t)^2$ *και όπως πριν* $M(D^t, j^t) = \sum_{i=1}^{n} p_i^t M(i, j^t)$.

Τότε $\forall i^* \in A$:

$$\sum_t M(D^t, j^t) - \sum_t M(i^*, j^t) \leq \epsilon \sum_t M(D^t, j^t)^2 + \frac{ln(n)}{\epsilon}$$

Απόδειξη.

Θα χρησιμοποιήσουμε δύο βασικές ανισότητες

(α') $e^x \geq x + 1 \quad \forall x \in \mathrm{Re}$
(β') $e^{-x} \leq 1 - x - x^2 \quad \forall x \in [0, \infty)$

Από τον ορισμό του ϕ_t έχουμε ότι:
$\phi_{t+1} = \sum_i w_i^t e^{-\epsilon M(i,j^t)} = \phi_t \sum_i p_i^t e^{-\epsilon M(i,j^t)} \leq \phi_t \sum_i p_i^t (1 - \epsilon M(i, j^t) - \epsilon^2 M(i, j^t)^2) =$
$= \phi_t (1 - \epsilon M(D^t, j^t) - \epsilon^2 M(D^t, j^t)^2) \leq \phi_t e^{-\epsilon M(D^t, j^t) - \epsilon^2 M(D^t, j^t)^2}$

Με επαγωγή προκύπτει ότι μετά από T επαναλήψεις:
$\phi_{T+1} \leq n e^{-\epsilon \sum_{t=1}^{T} M(D^t, j^t) - \epsilon^2 \sum_{t=1}^{T} M(D^t, j^t)^2}$

Επιπλέον, αφού τα βάρη παραμένουν θετικά σε όλη τη διάρκεια του αλγορίθμου, έχουμε ότι:
$\phi_{T+1} \geq w_{i^*}^{T+1} = e^{-\sum_{t=1}^{T} M(i^*, j^t)} \quad \forall i^* \in A$
συνδυάζοντας τις δύο τελευταίες ανισότητες παίρνουμε το επιθυμητό αποτέλεσμα □

Παρατηρήσεις: Ένας προσεκτικός αναγνώστης θα αντιληφθεί ότι ακολουθώντας την ίδια λογική όπως στον Multiplicative Weights Updates αλγόριθμο και παρατηρώντας ότι $\sum_{t=1}^{T} M(D^t, j^t)^2 \leq T$, παίρνουμε την ασυμπτωτικά ίδια συμπεριφορά επιλέγοντας $\epsilon = \sqrt{\frac{ln(n)}{T}}$. Επιπλέον, πριν ορίσουμε τυπικά την ποσότητα που φράζουμε, ας δώσουμε κάποια διασθητικά στοιχεία. Η ποσότητα $\sum_t M(D^t, j^t) - \sum_t M(i^*, j^t)$ ποσοτικοποιεί την απόδοση του αλγορίθμου μας με το να τον συγκρίνει με την

καλύτερη στατική στρατηγική που θα μπορούσαμε να έχουμε ακολουθήσει. Αυτή η ποσότητα είναι το ρεγρετ του online αλγορίθμου μας και θα καταλάβουμε αργότερα τη σημασία του.

Για να ολοκληρώσουμε, είναι σημαντικό να καταλάβουμε ότι το πλαίσιό μας είναι ισοδύναμο με έναν αντίπαλο που μετά από κάθε επανάληψη διαλέγει μια συνάρτηση l_t που δίνει ένα κόστος σε κάθε μία από τις πιθανές στρατηγικες μας. Η ισοδυναμία προκύπτει από τον τρόπο με τον οποίο έχουμε ορίσει το σετ αποτελεσμάτων Π (δηλαδή χωρίς περιορισμούς), άρα επιλέγοντας $M(i, l^t) = l^t(i)$, η ισοδυναμία γίνεται σαφής.

3.2 Offline Κυρτή Βελτιστοποίηση και Gradient Descent

Προτού προχωρήσουμε στο online πλαίσιο, είναι χρήσιμο να παρουσιάσουμε ένα βασικό υπόβαθρο offline κυρτής βελτιστοποίησης και τον πιο βασικό της αλγόριθμο.

Ο στόχος μας: Να ελαχιστοποιήσουμε μία συνεχή και κυρτή συνάρτηση πάνω σε ένα κυρτό υποσύνολο του Ευκλείδειου χώρου.

Βασικοί ορισμοί και ιδιότητες:

- Ένα σετ K είναι κυρτό ανν $\forall x, y \in K$, τότε $ax + (1 - a)y \in K$ $\forall a \in [0, 1]$
- Μία συνάρτηση φ σε ένα κυρτό σετ K είναι κυρτή ανν $\forall x, y \in K$, τότε
  $f(ax + (1 - a)y) \leq af(x) + (1 - a)f(y)$ $\forall a \in [0, 1]$
- αν η φ είναι διαφορίσιμη και ∇f υπάρχει $\forall x \in K$ τότε είναι κυρτή ανν $\forall x, y \in K$
  $f(y) \geq f(x) + \nabla f(x)^T(y - x)$
- Συμβολίζουμε με Δ ένα άνω φράγμα στη δίαμετρο του K:
  $\forall x, y \in K \quad \|x - y\| \leq D$

- Συμβολίζουμε με Γ ένα άνω φράγμα στη νόρμα του γραδιεντ:
  $\|\nabla f(x)\| \leq G \quad \forall x \in K$
- προβολή πάνω σε ένα κυρτό σύνολο (δηλαδή το κοντινότερο σημείο δεδομένου σημείου από το κυρτό σύνολο):
  $\Pi_K(y) = argmin_{x \in K} \|x - y\|$

Τώρα, προτού προχωρήσουμε στην παρουσίαση του γραδιεντ δεσσεντ αλγορίθμου, θα είνια χρήσιμο να υπογραμμίσουμε δύο σημαντικά θεωρήματα.

Θεωρεμ 3.2.1 (Πυθαγόρειο Θεώρημα)**.** *Έστω $K \subseteq \mathrm{Re}^d$ κυρτό σύνολο, $y \in \mathrm{Re}^d$ ανδ $x = \Pi_K(y)$, τότε*
$\forall z \in K \quad \|y - z\| \geq \|x - z\|$

Θεωρεμ 3.2.2 (Karush-Kuhn-Tucker)**.** *Έστω $K \subseteq \mathrm{Re}^d$ κυρτό σύνολο και $x^* \in argmin_{x \in K} f(x)$ τότε $\forall y \in K$:*
$\nabla f(x^*)^T (y - x^*) \geq 0$

Τώρα προχωράμε στην παρουσίαση του gradient descent αλογρίθμου. Η ιδέα πίσω από τον πιο απλό αλγόριθμο στη βελτιστοποίηση είναο ότι, όταν θέλουμε να ελαχιστοποιήσουμε κάποια συνάρτηση f, μία καλή ιδέα είναι να κάνουμε ένα βήμα προς την κατεύθυνση της πιο απότομης καθόδου (steepest descent) της τιμής του. Προφανώς, το πλάτος του βήματος πρέπει να επιλεγεί προσεκτικά για να επιτρέψει καλή σύγκλιση. Θα επικεντρωθούμε επίσης σε αυτές τις λεπτομέρειες αργότερα.

Αλγοριτημ 3 gradient descent αλγόριθμος

1: **Είσοδος** f, T, αρχικό σημείο $x_1 \in K, \{\eta_t\}$
2: **φορ** $t = 1 \quad to \quad T$ **δο**
3: $\quad y_{t+1} = x_t - \eta_t \nabla f(x_t)$
4: $\quad x_{t+1} = \Pi_K(y_{t+1})$
5: επιστρέφει $x_{avg} = \frac{\sum_{t=1}^{T} x_t}{T}$

Τηεορεμ 3.2.3. *Σύγκλιση του γραδιεντ δεσςεντ Έστω f κυρτή συνάρτηση σε ένα κυρτό σύνολο K. Αν D είναι ένα άνω φράγμα της διαμέτρου του K και G είναι ένα άνω φράγμα της νόρμας του gradient της f στο K. Τότε, επιλέγοντας $\eta_t = \frac{D}{G}\frac{1}{\sqrt{t}}$, έχουμε μετά από T επαναλήψεις ότι:*

$f(x_{avg}) - argmin_{x\in K} f(x) \leq \frac{3DG}{2\sqrt{T}}$

Απόδειξη.

Λετ $x^* \in argmin_{x\in K} f(x)$
Από την ανισότητα jensen έχουμε ότι:

$$f(x_{avg}) - f(x^*) = f(\sum_{t=1}^{T} \frac{x_t}{T}) - f(x^*) \leq \sum_{t=1}^{T} \frac{f(x_t) - f(x^*)}{T} \qquad (3.4)$$

Έστω $h_t = f(x_t) - f(x^*)$
Από την κυρτότητα της συνάρτησης έχουμε ότι:

$$h_t \leq \nabla f(x_t)^T (x_t - x^*) \qquad (3.5)$$

Τώρα προχωράμε με το να φράξουμε το δεξί σκέλος της τελευταίας εξίσωσης:

$\|x_{t+1} - x^*\|^2 \leq \|x_t - \eta_t - x^*\|^2 = \|x_t - x^*\|^2 + \eta_t^2 \|\nabla f(x_t)\|^2 - 2\eta_t \nabla f(x_t)^T (x_t - x^*) \Rightarrow$

$\Rightarrow 2\nabla f(x_t)(x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f(x_t)^2\| \Rightarrow$
$\Rightarrow 2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T} \{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f(x_t)^2\|\} \xrightarrow[\frac{1}{\eta_0}=0]{\eta_t = \frac{D}{G\sqrt{t}}}$
$2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T} \{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f(x_t)^2\|\} \leq \sum_{t=1}^{T} \|x_t - x^*\|^2 (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) +$
$\sum_{t=1}^{T} \eta_t \|\nabla f(x_t)^2\| \leq D^2 \sum_{t=1}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + G^2 \sum_{t=1}^{T} \eta_t = D^2 \frac{1}{\eta_T} + G^2 \sum_{t=1}^{T} \eta_t =$
$= DG\sqrt{T} + DG \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 3DG\sqrt{T}$

από τις εξισώσεις 3.4 και 3.5 έχουμε ότι:

$f(x_{avg}) - f(x^*) \leq \frac{3DG}{\sqrt{T}}$

□

3.3 Online Convex Optimization

3.3.1 Το Online Convex Optimization μοντέλο

Το OCO (Online Convex Optimization) μοντέλο προσπαθεί να αντιμετωπίσει το πρόβλημα της βελτιστοποίησης μιας σειράς επιλογών αντί να προσπαθεί να βρει τη μοναδική βέλτιστη επιλογή. Διαισθητικά, για μη στατικά προβλήματα, αν αποκαλύπτονται νέες πληροφορίες ενώ η διαδικασία συνεχίζεται, έχουμε αυτό που χρειαζόμαστε. Κλασσικές μέθοδοι της αλγοριθμικής θεωρίας παιγνίων και της μαθηματικής βελτιστοποίησης δεν είναι εύρωστες στην αβεβαιότητα του περιβάλλοντος. Επομένως, χρειαζόμαστε αλγορίθμους που αποκτούν εμπειρία από το παρελθόν και δρουν καλά κάτω από οποιαδήποτε περίσταση του μέλλοντος.

Αντίθετα με τον Multiplicative Weights Update αλγόριθμο, οι αλγόριθμοι που προτείνονται σε αυτή την ενότητα δεν διατηρούν την κατανομή πάνω σε περιορισμένο χώρο στρατηγικών, αλλά δουλεύουν σε ένα κυρτό σύνολο και άρα σε συνεχή χώρο.

Το πλαίσιο OCO ,όπως προτάθηκε στο [18], μπορεί να δομηθεί ως επαναλαμβανόμενο παίγνιο.

Έστω K κυρτό σύνολο στρατηγικών διαθέσιμο στον ονλινε παίκτη και F ένα φραγμένο (ή με κάποιο τρόπο δομημένο) σύνολο συναρτήσεων κόστους που είναι διαθέσιμες στον αντίπαλο.

σε κάθε επανάληψη t:

(α′) ο online παίκτης διαλέγει στρατηγική $x_t \in K$
(β′) μία κυρτή συνάρτηση $f_t \in F : K \to \mathrm{Re}$ αποκαλύπτεται
(γ′) ο ονλινε παίκτης έχει κόστος $f_t(x_t)$

Αφού ορίσουμε το γενικό πλαίσιο, πρέπει να ορίσουμε ποσότητες για να αξιολογήσουμε την απόδοση ενός αλγορίθμου που επιλέγει τις στρατηγικές του βήματος (1) του πλαισίου μας. Φαίνεται ότι η κατάλληλ μετρική απόδοσης ονομάζεται regret και είναι η διαφορά μεταξύ του κόστους που υποφέρει ο αλγόριθμός μας και του κόστους που θα είχαμε υποφέρει αν είχαμε παίξει με την καλύτερη σταθερή στρατηγική στο παίγνιο.

Ας ορίσουμε τυπικά τη μετρική του ρεγρετ.

Δεφινιτιον 3.3.1. *Έστω Alg ο αλγόριθμος του OCO μοντέλου μας, που αντιστοιχίζει την ιστορία ενός συγκεκριμένου παιγνίου σε μία στρατηγική. Το regret του Alg μετά από T επαναλήψεις ορίζεται ως:*
$regret_T(Alg) = sup_{\{f_1,f_2,...,f_T\}\subseteq F}\{\sum_{t=1}^{T} f_t(x_t) - min_{x\in K}\sum_{t=1}^{T} f_t(x)\}$

Είναι σημαντικό να υπογραμμίσουμε την έκταση των προβλημάτων που καλύπτει το OCO πλαίσιο. Θυμηθείτε τον Multiplicative Weights Updates αλγόριθμο. Ορίζοντας το K ως ν-διάστατο σιμπλεξ και $f_t(x_t) = M(x_t, f_t)$, είναι εμφανές ότι το πρόβλημα λύνει μόνο μια ειδική περίπτωση του γενικού ΟΟ μοντέλου. Επιπρόσθετα, επιστρέφοντας στο 3.1.1.1 γίνεται φανερό ότι το regret που επιτυγχάνεται από τον Multiplicative Weights Updates αλγόριθμο είναι φραγμένο από τη $\sqrt{T}$.

3.3.2 Online Gradient Descent

Όπως δείχνει το όνομα, ο απλούστερος αλγόριθμος στο πλαίσιο OCO είναι η online μορφή του τυπικού gradient descent. Αυτό προφανώς ακολουθεί τη λογική του απλού offline gradient descent και προτάθηκε πρώτα από τον Zinkevich στο [19]

Στην offline μορφή, κάθε νέο σημείο ανανεώνεται σύμφωνα με την κατεύθυνση του αρνητικού gradient της συνάρτησης στο προηγούμενο σημείο. Μία γενίκευση αυτής της ιδέας υποθέτει ότι το νέο μας σημείο θα πρέπει να ακολουθεί το αρνητικό gradient της συνάρτησης κόστους που φαίνεται στην t επανάληψη του προηγούμενου σημείου που έχει επιλεχθεί.

Αυτό που μας εκπλήσσει είναι ότι, αν και οι συναρτήσεις μπορεί να είναι τελείως διαφορετικές κάθε φορά (οπότε το να ακολουθούμε το αρνητικό gradient της προηγούμενης συνάρτησης δε βγάζει νόημα), επιτυγχάνεται regret ίσο με $O(\sqrt{T})$.

Ας παρουσιάσουμε τον αλγόριθμο Online Gradient Descent:

Αλγοριτημ 4 online gradient descent αλγόριθμος (OGD)

Είσοδος: Έστω κυρτό σύνολο K, T, αρχικό σημείο $x_1 \in K, \{\eta_t\}$
φορ $t = 1 \quad to \quad T$ **δο**
    $y_{t+1} = x_t - \eta_t \nabla f_t(x_t)$
    $x_{t+1} = \Pi_K(y_{t+1})$

Τηεορεμ 3.3.2. *Regret του online gradient descent αλγορίθμου Αν D είναι ένα άνω φράγμα της διαμέτρου του K και το G ένα άνω φράγμα της νόρμας των gradients κάθε συνάρτησης στο F. Τότε, επιλέγοντας $\eta_t = \frac{D}{G}\frac{1}{\sqrt{t}}$, έχουμε ότι μετά από T επαναλήψεις:*

$regret_T(OGD) \leq \frac{3}{2}DG\sqrt{T}$

Απόδειξη. Η απόδειξη είναι σχεδόν ίδια με την απόδειξη της σύγκλισης του απλού gradient descent αλγορίθμου. Ωστόσο, την παρουσιάζουμε για λόγους πληρότητας.

Έστω $f_1, f_2, ..., f_T \in F$ οι συναρτήσεις κόστους που αν τις επιλέξει ο αντίπαλός μας, μας δημιουργούν το μεγαλύτερο regret.

Έστω $x^* \in argmin_{x \in K} \sum_{t=1}^{T} f_t(x)$

$regret_T(OGD) = \sum_{t=1}^{T}(f_t(x_t) - f_t(x^*))$

Έστω $h_t = f_t(x_t) - f_t(x^*)$

Από την κυρτότητα των συναρτήσεων στο F έχουμε ότι:

$$h_t \leq \nabla f_t(x_t)^T(x_t - x^*) \tag{3.6}$$

Συνεχίζουμε φράζοντας το δεξί σκέλος της τελευταίας εξίσωσης:

$\|x_{t+1} - x^*\|^2 \leq \|x_t - \eta_t - x^*\|^2 = \|x_t - x^*\|^2 + \eta_t^2\|\nabla f_t(x_t)\|^2 - 2\eta_t \nabla f_t(x_t)^T(x_t - x^*) \Rightarrow$

$\Rightarrow 2\nabla f_t(x_t)(x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t\|\nabla f_t(x_t)^2\| \Rightarrow$
$\Rightarrow 2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T}\{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t\|\nabla f_t(x_t)^2\|\} \xrightarrow[\frac{1}{\eta_0}=0]{\eta_t = \frac{D}{G\sqrt{t}}}$

$2\sum_{t=1}^{T} h_t \leq \sum_{t=1}^{T} \{\frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t \|\nabla f_t(x_t)^2\|\} \leq \sum_{t=1}^{T} \|x_t - x^*\|^2 (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) +$
$\sum_{t=1}^{T} \eta_t \|\nabla f_t(x_t)^2\| \leq D^2 \sum_{t=1}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + G^2 \sum_{t=1}^{T} \eta_t = D^2 \frac{1}{\eta_T} + G^2 \sum_{t=1}^{T} \eta_t =$
$= DG\sqrt{T} + DG \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 3DG\sqrt{T}$

τελικά από την εξίσωση 3.6 έχουμε ότι:

$regret_T(OGD) \leq \frac{3DG\sqrt{T}}{2}$

□

Τώρα είναι χρήσιμο να τονίσουμε πώς πρέπει να χρησιμοποιείται αυτός ο αλγόριθμος από ένα παίκτη. Σε αντίθεση με τον Multiplicative Weights Update αλγόριθμο, ο παραπάνω αλγόριθμος επιτυγχάνει κάτι ευρύτερο και καλύτερο. Ελαχιστοποιεί το regret σε συνεχή χώρο. Ένας παίκτης που θέλει να δημιουργήσει έναν αλγόριθμο για πονταρίσματα, όταν χρησιμοποιεί τους MWU αλγορίθμους θα πρέπει πρώτα να διαρέσει το χώρο πονταρισμάτων και μετά, τρέχοντας τον MWU αλγόριθμο με τις συγκεκριμένες πράξεις (δηλαδή πονταρίσματα).

Αντίθετα, όταν χρησιμοποιούμε τον OGD αλγόριθμο, υπάρχουν δύο πιθανές προσεγγίσεις. Η πρώτη είναι να διαιρέσουμε το χώρο πονταρισμάτων σε n πράξεις και να τρέξουμε μετά τον αλγόριθμο στο n διάστατο simplex. Αν και αυτή είναι καλή τακτική, είναι εμφανές ότι η εξέλιξη του αλγορίθμου μιμείται την MWU ιδέα διατηρώντας την κατανομή πιθανότητας πάνω σε ένα σύνολο στρατηγικών και κινούμενη συνεχώς προς το simplex για να εξασφαλίσει καλύτερη απόδοση. Η άλλη επιλογή είναι να τρέξουμε τον OGD υποστηρικτικά στον χώρο πονταρισμάτων. Φυσικά, αυτή είναι απλά μια μονοδιάστατη γραμμή και είναι κυρτή. Ωστόσο, το πρόβλημα εδώ είναι ότι τα gradient είναι μη-φραγμένα και μια άμεση υλοποίηση δεν είναι εφικτή. Μία ιδέα σε αυτή την κατεύθυνση θα ήταν να δημιουργήσουμε μια ομαλή μορφή της συνάρτησης κέρδους και έτσι να δώσουμε στον αλγόριθμο μια φραγμένη οικογένεια από gradients. Μια άλλη πιθανότητα θα ήταν να ανανεώσουμε το ποντάρισμά μας όχι σε κάθε επανάληψη, αλλά λιγότερο συχνά. Αυτό θα δημιουργήσει συναρτήσεις κέρδους που είναι πιθανά συνεχείς και διαφορίσιμες.

3.3.3 Κάτω φράγματα στο OCO μοντέλο

Αφού παρουσιάσαμε τον βασικό αλγόριθμο για online κυρτή βελτιστοποίηση, αποδείξαμε ένα φράγμα για το $O(\sqrt{T})$ του regret. Εδώ προκύπτει μια ενδιαφέρουσα ερώτηση. Στη γενική περίπτωση, ποιο είναι

το καλύτερο φράγμα που μπορούμε να επιτύχουμε· Το επόμενο θεώρημα αποδεικνύει ότι στην πραγματικότητα το $O(\sqrt{T})$ είναι ένα κάτω όριο στο καλύτερο regret που μπορούμε να πετύχουμε σε γενικές συνθήκες οποιουδήποτε OCO αλγορίθμου.

Τηεορεμ 3.3.3. *Κάθε αλγόριθμος στο OCO μοντέλο έχει γενικά regret ίσο με* $\Omega(DG\sqrt{T})$

Απόδειξη.
Έστω K n-διάστατος υπερκύβος, $K = \{x \in Re^n, \|x\|_\infty \leq 1\}$
Έστω $f_v(x) = v^T x$ η μορφή των 2^n γραμμικών συναρτήσεων κόστους, μία για κάθε κορυφή του K.

Αρχικά, σημειώστε ότι οι D και G είναι φραγμένες:

$D = \sqrt{\sum_{i=1}^{n} 2^2} = 2\sqrt{n}, \quad G = \sqrt{\sum_{i=1}^{n} 1^2} = \sqrt{n}$

Έστω ότι ο αντίπαλος επιλέγει σε κάθε επανάληψη ομοιόμορφα και τυχαία μία συνάρτηση κόστους από τις 2^n διαθέσιμες.

Σε κάθε επανάληψη έχουμε ότι:

$$E_{v_t}\left[f_t(x_t)\right] = E_{v_t}\left[v_t x_t\right] = 0$$

Τώρα, έστω x^* η καλύτερη σταθερή στρατηγική για μια συγκεκριμένη ακολουθία συναρτήσεων, θα έχουμε ότι:

$$E_{v_1,v_2,\ldots,v_t}\left[\sum_{t=1}^{T} f_t(x^*)\right] = E_{v_1,v_2,\ldots,v_t}\left[\sum_{t=1}^{T}\sum_{i=1}^{n} v_t(i)x^*(i)\right] = E_{v_1,v_2,\ldots,v_t}\left[\sum_{i=1}^{n}\sum_{t=1}^{T} v_t(i)x^*(i)\right] =$$
$$E_{v_1,v_2,\ldots,v_t}\left[-|\sum_{i=1}^{n}\sum_{t=1}^{T} v_t(i)|\right] = E_{v_1}\left[-n|\sum_{t=1}^{T} v_t(1)|\right] = = -n\Omega(\sqrt{T})$$

Η τελευταία ισότητα προκύπτει επειδή το άθροισμα αυτό αντιπροσωπεύει ένα σημείο σε τυχαίο μονοπάτι μετά απο T βήματα.

□

Παρατηρήσεις: Σε αυτό το σημείο, είναι σημαντικό να υπογραμμίσουμε δύο σημαντικά γεγονότα. Πρώτον, ο απλός online gradient descent αλγόριθμος πετυχαίνει το βέλτιστο πιθανό ρεγρετ σε γενικές συνθήκες. Δεύτερον, η τελευταία απόδειξη αποτελεί παράδοξο. Αν έπρεπε να αποδείξουμε το κάτω φράγμα από την αρχή, ίσως ο αντίπαλος που

θα είχαμε κατασκευάσει δε θα ήταν στατική κατανομή. Πώς είναι δυνατό μία στατική κατανομή να έχει το χειρότερο δυνατό regret· Η απάντηση δίνεται από τη διαφορά μεταξύ της τιμής του regret και του loss που έχει ο αλγόριθμός μας. Είναι προφανές ότι ένας αντίπαλος που χρησιμοποιεί πρότερη γνώση θα επέβαλλε μεγαλύτερο loss στον αλγόριθμό μας, ωστόσο ταυτόχρονα η απόδοση της βέλτιστης σταθερής στρατηγικής θα ήταν κακή. Ας εξηγήσουμε αυτό το παράδοξο:

Τώρα γνωρίζουμε ότι η καλύτερη πιθανή τακτική για έναν παίκτη είναι να χρησιμοποιήσει αλγόριθμο χωρίς regret, καθώς σε γενικές συνθήκες διασφαλίζουν ένα σχεδόν βέλτιστο φράγμα για το regret.

3.4 Bandit Κυρτή Βελτιστοποίηση (Bandit Convex Optimization)

3.4.1 Το Bandit Convex Optimization μοντέλο

Ως τώρα, παρουσιάσαμε τον Multiplicative Weights Updates και τον Hedge αλγόριθμο. Μετά, γενικεύοντας τις συνθήκες εξήγαμε τον online gradient descent αλγόριθμο. Όλοι αυτοί οι αλγόριθμοι έχουν ένα κοινό χαρακτηριστικό, απαιτούν ένα πλήρες μοντέλο πληροφόρησης.Στην πραγματικότητα, τόσο ο Multiplicative Weights Updates όσο και ο Hedge αλγόριθμος πραγματοποιούν μια διαδικασία ανανέωσης βαρών που απαιτεί γνώση της loss συνάρτησης για ολόκληρο το χώρο στρατηγικών (πράξεων). Ο online gradient descent απαιτεί γνώση του gradient της loss συνάρτησης, που είναι επίσης συνθήκη που είναι δύσκολο να ικανοποιηθεί.

Σε πολλά σενάρια, αυτό είναι αρκετά μη-ρεαλιστικό. Για παράδειγμα σκεφτείτε το πρόβλημα δρομολόγησης σε ένα άγνωστο δίκτυο. Επιλέγουμε ένα μονοπάτι και μετά περιμένουμε μέχρι τα δεδομένα μας να φτάσουν στον επιθυμητό προορισμό. Κανείς δε μας λέει πόσο χρόνο θα περιμέναμε αν είχαμε επιλέξει ένα διαφορετικό μονοπάτι. Το μόνο feedback που έχουμε, είναι το loss που είχαμε από μια συγκεκριμένη στρατηγική που ακολουθήσαμε.

Στο πρόβλημά μας (δηλαδή την GSP δημοπρασία), οι παίκτες δε γνωρίζουν ούτε τη συνάρτηση διανομής ούτε τις ειδικές συνιστώσες θέσης της συγκεκριμένης δημοπρασίας όπου συμμετέχουν. Σε αυτές τις συνθήκες, οι παίκτες ποντάρουν και μετά παίρνουν το κέρδος του ποντάρισματός τους. Ωστόσο, δε γνωρίζουν τι κέρδος θα είχαν αν είχαν κάνει διαφορετικό ποντάρισμα. Προφανώς, το να μη γνωρίζει κάποιος την απόδοση

ενός διαφορετικού πονταρίσματος δεν επιτρέπει την κλασσική υλοποίηση του MWU αλγορίθμου.

Για να καλύψουμε αυτές τις συνθήκες, εισάγουμε το Bandit Convex Optimization μοντέλο.

Το BCO πλαίσιο ,όπως προτάθηκε στο [;] , μπορεί να υλοποιηθεί ως επαναλαμβανόμενο παίγνιο.

Έστω K το κυρτό σύνολο των στρατηγικών που είναι διαθέσιμες για τον ονλινε παίκτη και F το φραγμένο (ή με κάποιο τρόπο δομημένο) σύνολο συναρτήσεων κόστους που είναι διαθέσιμες στον αντίπαλο.

σε κάθε επανάληψη t:

(α′) ο online παίκτης επιλέγει στρατηγική $x_t \in K$

(β′) ο online παίκτης έχει κόστος $f_t(x_t)$

Ένας προσεκτικός αναγνώστης θα αντιληφθεί ότι το πλαίσιο είναι παρόμοιο με αυτό του OCO μοντέλου. Ωστόσο, ο αντίπαλος δεν αποκαλύπτει τη συνάρτηση κόστους στον παίκτη. Έτσι, το μόνο διαθέσιμο feedback είναι το loss που είχε ο online παίκτης.

Το μέγεθος που θα μας βοηθήσει να αξιολογήσουμε την απόδοση του αλγορίθμου μας είναι, όπως πάντα, το regret:

Δεφινιτιον 3.4.1. *Έστω Alg ο αλγόριθμος του BCO μοντέλου μας, που αντιστοιχίζει την ιστορία ενός συγκεκριμένου παιγνίου σε μία στρατηγική. Το regret του Alg μετά από T επαναλήψεις ορίζεται ως:*
$regret_T(Alg) = sup_{\{f_1,f_2,...,f_T\}\subseteq F}\{\sum_{t=1}^{T} f_t(x_t) - min_{x\in K}\sum_{t=1}^{T} f_t(x)\}$

Παρατήρηση: Ωστόσο, τα προβλήματα που καλύπτει το μοντέλο γίνονται εμφανώς πιο δύσκολα. Το regret ακολουθεί τον ίδιο ορισμό.

3.4.2 Multi Armed Bandit (MAB) μοντέλο

Σε αυτή την ενότητα, θα εισάγουμε το απλούστερο μοντέλο bandit βελτιστοποίησης, το οποίο αποτελεί απλά υποκατηγορία του πιο γενικού BCO μοντέλου. Αυτό το μοντέλο εισήχθη από τον Robbins στο [20]

Ας περιγράψουμε το MAB μοντέλο. Όπως πάντα, έχουμε επαναλαμβανόμενο παίγνιο:

σε κάθε επανάληψη t:

(α′) ο online παίκτης επιλέγει στρατηγική $i_t \in \{1, 2, ...n\}$

(β′) ο online παίκτης έχει κόστος $l_t(i_t) \in [0, 1]$

Αυτό το απλό επαναλαμβανόμενο παίγνιο μας θυμίζει το πρόβλημα που λύνει ο Multiplicative Weight Updates αλγόριθμος. Στην πραγματικότητα, είναι απλά το bandit ανάλογο αυτού του προβλήματος. Σημειώστε ότι εδώ κάθε αλγόριθμος που προσπαθεί να καταλάβει ποια στρατηγική να επιλέξει στην επόμενη επανάληψη δε γνωρίζει τις προηγούμενες loss συναρτήσεις, αλλά μόνο τις τιμές τους.

Τα προβλήματα που μοντελοποιούνται εδώ είναι απλά ειδικές περιπτώσεις του πιο γενικού BCO μοντέλου. Για να το καταλάβουμε, θα μετατρέψουμε το MAB μοντέλο σε BCO πρόβλημα.

(α′) Έστω σύνολο στρατηγικών από $\{1, 2, ..., n\}$ να σχηματίζει το σύνολο των κατανομών πάνω σε αυτές τις ν στρατηγικές. Άρα $K = \Delta^n$ είναι το n-διάστατο simplex.

(β′) Έστω ότι οι loss συναρτήσεις είναι $f_t(x_t) = \sum_{i=1}^{n} l_t(i)x_t(i)$ (δηλαδή η αναμενόμενη loss της κατανομής μας)

Και η ισοδυναμία προκύπτει εύκολα.

3.4.3 MAB αλγόριθμοι

Σε αυτή την ενότητα, θα εισάγουμε δύο βασικούς αλγορίθμους που επιτυγχάνουν χαμηλό ρεγρετ στο μοντέλο MAB

Αρχικά, ας συζητήσουμε ποιες ιδιότητες πρέπει να έχει ένας αλγόριθμος για να αντιμετωπίσει τέτοιου είδους προβλήματα. Εδώ, κάθε φορά που διαλέγουμε μία στρατηγική, υποφέρουμε το αντίστοιχο loss, άρα δεν έχουμε στοιχεία για την απόδοση των υπόλοιπων στρατηγικών σε αυτή τη συγκεκριμένη επανάληψη. Εδώ εισάγεται η πρώτη ιδιότητα που πρέπει να έχει ένας MAB αλγόριθμος. Πρέπει να εξερευνά αποδοτικά το χώρο στρατηγικών, δηλαδή πρέπει να επιχειρεί διαφορετικές στρατηγικές για να πάρει ένα δείγμα του πώς αποδίδουν κατά μέσο όρο.

Δεύτερον, μιας και ο MAB αλγόριθμος είναι βασικά αλγόριθμος πρόβλεψης, στηρίζεται στην συσσωρευμένη εμπειρία του για να μας δώσει πρόβλεψη σχετικά με το ποια στρατηγική θα έχει καλή απόδοση στην επόμενη επανάληψη. Αυτό είναι το βήμα εκμετάλλευσης.

Συνοπτικά, οι MAB αλγόριθμοι πρέπει να κάνουν δύο πράγματα. Να εξερευνούν το χώρο στρατηγικών και, από αυτή την εξερεύνηση, να προβλέπουν καλές στρατηγικές στο μέλλον.

Τώρα θα παρουσιάσουμε τον απλούστερο τύπο MAB αλγορίθμου που προτάθηκε από τον Hazan στο [18]. Διαχωρίζει τελείως το βήμα εξερεύνησης από το βήμα εκμετάλλευσης. Άρα, σε κάθε επανάληψη, εξερευνά με κάποια πιθανότητα το χώρο στρατηγικών, δηλαδή σπαταλά loss για να πάρει χρήσιμες πληροφορίες και με την υπόλοιπη πιθανότητα κάνει το βήμα εκμετάλλευσης, που συνίσταται σην πρόβλεψη μιας καλής στρατηγικής.

Αλγοριτημ 5 απλός MAB αλγόριθμος

Είσοδος: OCO αλγόριθμος A, παράμετρος δ
φορ $t = 1 \quad to \quad T$ **δο**
  Έστω b_t μεταβλητή Bernoulli, με $P(b_t = 1) = \delta$
  ιφ $b_t = 1$ **τηεν**
    διάλεξε $i_t \sim$ τυχαία ομοιόμορφα από $\{1, 2, ...n\}$
    παίξε i_t
    φορ $i = 1 \quad to \quad n$ **δο**
      $\hat{l_t}(i) = \frac{n}{\delta} l_t(i) \mathbb{1}\{i = i_t\}$
    Έστω $\hat{f}_t(x) = \hat{l_t}^T x$
    $x_{t+1} = A(\hat{f}_1, \hat{f}_2, ..., \hat{f}_t)$
  ελσε ιφ $b_t = 0$ **τηεν**
    διάλεξε $i_t \sim x_t$
    παίξε i_t
    $\hat{f}_t(x) = \mathbf{0}$
    $x_{t+1} = A(\hat{f}_1, \hat{f}_2, ..., \hat{f}_t)$

Κεφάλαιο 4

Εκτίμηση ΩΦέλειας στις **Online** Διαφημιστικές Δημοπρασίες

Σε αυτό το κεφάλαιο θα μελετήσουμε και εκτιμήσουμε τη νέα μέθοδο εκτίμησης ωφέλειας που προτάθηκε πρόσφατα από τους Eva Tardos, Vasilis Syrgkanis και Denis Nekipelov στο [4] σε σχέση με τις GSP δημοπρασίες. Επιπλέον, θα περιγράψουμε το σύστημα προσομοίωσης που σχεδιάστηκε για να εκτιμήσει αυτές τις μεθόδους, τα πλεονεκτήματά του και τις μελλοντικές πιθανότητες ανάπτυξής του.

Επιπλέον, θα εξετάσουμε την ευρωστία του μοντέλου αξίας τους υπό Μη-Ειλικρινείς μηχανισμούς, ως συνέχεια της μελέτης ευρωστίας των Dutting, Parkes και Fischer στο [5].

4.1 Καθορισμός Προβλήματος

Το πρόβλημα της εκτίμησης της αξίας των παικτών σε δημοπρασίες είναι πολύ σημαντικό, επειδή η γνώση των αξιών των παικτών επιτρέπει ακριβή μελέτη της συμπεριφοράς τους στο ποντάρισμα και συνεπώς την εκτίμηση των προβλέψεων που έγιναν θεωρητικά. Στη μορφή που εξετάζουμε εμείς, ο εκτιμητής δίνεται από τα εξής:

- ο αριθμός των αντικειμένων που πωλήθηκαν
- ο μέγιστος αριθμός παικτών σε κάθε επανάληψη
- οι συνιστώσες σημασίας των παικτών
- η αξία κάθε αντικειμένου που πωλήθηκε

- ελάχιστη τιμή απόκτησης
- το πλήρες ιστορικό πονταρίσματος της δημοπρασίας
- η μορφή της δημοπρασίας που υλοποιείται

Ο στόχος είναι να ολοκληρώσουμε την ιδιωτική αξία ανά μονάδα προϊόντος που έχει κάθε παίκτης.

Θα επικεντρωθούμε στην υλοποίηση GSP δημοπρασιών. Επομένως, το αντικείμενο προς πώληση είναι οι θέσης διαφήμισης, η αξία τους είναι οι αντίστοιχες συνιστώσες θέσεων, η σημασία κάθε παίκτη i είναι οι συνιστώσες του πατήματος (κλικ) κλπ. Προφανώς, οι συναρτήσεις διανομής και κόστους είναι αυτές που περιγράφηκαν προηγουμένως .

Προτού προχωρήσουμε στις τεχνικές λεπτομέρειες, είναι χρήσιμο να περιγράψουμε τη βασική ιδέα πίσω από το μοντέλο. Η τυπική οικονομική προσέγγιση για τα περιβάλλοντα όπου αλληλεπιδρούν στρατηγικοί παίκτες είναι να υποθέτουμε ότι κάπως οι παίκτες έχουν φτάσει σε μία μικτή ισορροπία κατά Nash, και άρα αποκρίνονται βέλτιστα στην κατανομή που αντιμετωπίζουν [3]. Άλλες μέθοδοι στηρίζονται στην υπόθεση ότι οι παίκτες θα φτάσουν σε μια συγκεκριμένη ισορροπία που χαρακτηρίζεται από συναρτήσεις πονταρίσματος των αξιών τους. Μέσα από μία προσεγγιστική εκτίμηση της κατανομής πονταρίσματος, είναι πιθανό να αντιστρέψουμε τις συγκεκριμένες συναρτήσεις και να καταλήξουμε στην κατανομή της αξίας [1]. Επιπλέον, άλλες μέθοδοι καταλήγουν σε μια συγκεκριμένη δυάδα (τούπλα) αξιών που εξηγεί βέλτιστα την προσεγγιστική ισορροπία που επιτεύχθηκε [2]. Μια διαφορετική προσέγγιση που διαπερνά την δημοπρασία περιγράφεται στο [22] .

Ο σκοπός της νέας μεθόδου είναι να περιγράψει γρήγορα μεταβαλλόμενα περιβάλλοντα, όπως αυτό των online δημοπρασιών με πιο ακριβή τρόπο. Οι ονλινε δημοπρασίες αλλάζουν γρήγορα όσο νέοι παίκτες εμφανίζονται και εξαφανίζονται και οι παίκτες προσπαθούν να προσαρμόσουν τη συμπεριφορά τους στους εκάστοτε αντιπάλους με σταθερό τρόπο. Επιπλέον, ένα ακόμα ελάττωμα αυτής της μεθόδους είναι η υπόθεση ότι οι παίκτες μπορούν να φτάσουν σε μικτή ισορροπία κατά Nash. Το πρόβλημα δεν είναι τόσο σχετικό ως προς την πολυπλοκότητα, όσο από θεωρητικής πλευράς. Οι παίκτες δεν αλληλεπιδρούν σε full information περιβάλλοντα και έτσι, το πλήθος των πληροφοριών που χρειάζονται για να φτάσουν μία τέτοια ισορροπία είναι ίσως υπερβολικά μεγάλο για να διασφαλίσει μία τέτοια υπόθεση. Όλες αυτές οι παρατηρήσεις μας οδηγούν στο να χαλαρώσουμε την υπόθεση για το τι επτυγχάνουν οι παίκτες.

Η νέα μέθοδος εκτίμησης υποθέτει ότι οι παίκτες μαθαίνουν σε ένα συνεχώς μεταβαλλόμενο περιβάλλον. Επομένως, χρησιμοποιούν αλγορίθμους μάθησης (learning) για να αποφασίσουν ποια είναι η βέλτιστη στρατηγική που πρέπει να ακολουθήσουν κάθε φορά. Η υπόθεση μας κάνει να υποθέτουμε ότι οι παίκτες, αν θέλουν να μεγιστοποιήσουν το κέρδος τους σε βάθος χρόνου, πρέπει να χρησιμοποιήσουν νο-ρεγρετ αλγορίθμους. Υπάρχουν no-regret αλγόριθμοι που παίρνουν τα καλύτερα δυνατά φράγματα για το regret και είναι λογικό να υποθέσουμε ότι οι παίκτες θα χρησιμοποιήσουν τέτοιους αλγορίθμους. Επιπλέον, όπως παρουσιάσαμε προηγουμένως, αυτοί οι αλγόριθμοι είναι αρκετά απλοί στην υλοποίηση, άρα η υπόθεσή μας στηρίζεται από αυτή την παρατήρηση.

4.2 Μέθοδος Εκτίμησης

Θα προχωρήσουμε περιγράφοντας με λεπτομέρεια αυτή τη μέθοδο.

Κάθε παίκτης έχει ένα χώρο στρατηγικής B_i. Άρα ένα προφίλ πονταρίσματος $b \in B_1 \times B_2 \times ... \times B_n$. Επιπλέον, θα συμβολίσουμε με b^t το προφίλ πονταρίσματος στην τ-οστή επανάληψη. Υποθέτοντας ότι κάθε παίκτης έχει ρεγρετ ίσο με ϵ_i και αξία v_i έχουμε ότι:

$$\forall b' \in B_i \quad \frac{1}{T}\sum_{t=1}^{T} U_i(b^t, v_i) \geq \frac{1}{T}\sum_{t=1}^{T} U_i(b', b^t_{-i}, v_i) - \epsilon_i \tag{4.1}$$

Δεφινιτιον 4.2.1. *(Rationalizable σύνολο) Ένας ζεύγος (ϵ_i, v_i) αξίας v_i και regret ϵ_i είναι rationalizable αν ικανοποιεί την προηγούμενη ϵ-ξίσωση. Το σύνολο τέτοιων ζευγαριών ονομάζεται rationalizable σύνολο του παίκτη i και συμβολίζεται με NR.*

Τώρα, θα αναλύσουμε τις ιδιότητες του rationalizable συνόλου NR. Υποθέτουμε ότι $B_i \subseteq \mathrm{Re}_+$. Επίσης από το σχεδόν γραμμικό μοντέλο έχουμε ότι:

$$\forall b' \in B_i \quad \frac{1}{T}\sum_{t=1}^{T}(v_i P_i(b^t) - C_i(b^t)) \geq \frac{1}{T}\sum_{t=1}^{T}(v_i P_i(b', b^t_{-i}) - C_i(b', b^t_{-i})) - \epsilon_i \tag{4.2}$$

Συμβολίζοντας με:

$$\Delta P(b') = \frac{1}{T}\sum_{t=1}^{T}\left(P_i(b', b^t_{-i}) - P_i(b^t)\right) \tag{4.3}$$

Η μέση αύξηση της πιθανότητας του κλικ αν ένας παίκτης επιλέξει να ποντάρει συνεχώς b'.

$$\Delta C(b') = \frac{1}{T}\sum_{t=1}^{T}\left(C_i(b', b^t_{-i}) - C_i(b^t)\right) \tag{4.4}$$

Η μέση αύξηση του κόστους που έχει ένας παίκτης αν επιλέξει να ποντάρει συνεχώς b'.

Άρα η αρχική εξίσωση είναι ισοδύναμη με:

$$\forall b' \in B_i \quad v_i \Delta P(b') - \Delta C(b') \leq \epsilon_i \tag{4.5}$$

Λεμμα 4.2.2. *Το rationalizable σύνολο για κάθε παίκτη είναι ένα κλειστό κυρτό σύνολο*

Απόδειξη Κάθε ανισότητα είναι γραμμική στις αξίες και στο regret. Άρα, διαιρούν το χώρο σε υποχώρους. Η τομή των υποχώρων είναι προφανώς κυρτό σύνολο. Επιπλέον, οι ανισότητες δεν είναι αυστηρές. Άρα, το σύνολο είναι κλειστό.

Μπορεί να αποδειχθεί ότι υπό κάποιες ακόμα λογικές υποθέσεις, το rationalizable σύνολο έχει άλλες χρήσιμες ιδιότητες που επιτρέπουν να το υπολογίσουμε αποδοτικά. Για να τα αναφέρουμε συνοπτικά, θα υπογραμμίσουμε ότι αυτό το συγκεκριμένο κυρτό σύνολο είναι πλήρως καθορισμένο από τις συναρτήσεις $\Delta P(\cdot)$ και $\Delta C(\cdot)$ καθώς καθορίζουν την support συνάρτηση του κυρτού συνόλου. Η πλήρης ανάλυση της μεθόδου εκτίμησης έχει δοθεί από τους Syrgkanis, Nekipelov και Tardos στο [4].

Τώρα, η μέθοδος εκτίμησης είναι αρκετά προφανής. Ο κύριος στόχος μας είναι να βρούμε την αξία κάθε παίκτη που εξηγεί καλύτερα τα ποντάρισματά του σε βάθος χρόνου.

Για να εξηγήσουμε καλύτερα, ας πάρουμε την αξία που συνδέεται με το ελάχιστο δυνατό regret. Προφανώς, θα περιοριστούμε σε αξίες που είναι αυστηρά θετικές. Επιπλέον, θα θεωρήσουμε τον B_i διακριτοποιημένο φραγμένο χώρο που είναι υποσύνολο του συνόλου $[0, b_{max}]$.

Από τα προηγούμενα, θέτουμε $B_i = \{0, 2e, 3e, ..., b_{max}\}$. Τότε, η μέθοδος εκτίμησης μπορεί να περιγραφεί ως γραμμικό πρόγραμμα:

$$\begin{aligned} &elaqistopohse \quad \epsilon + 0 \cdot v \\ &subject \quad to \quad v\Delta P(b') - \Delta C(b') \leq \epsilon \quad \forall b' \in B_i \end{aligned} \tag{4.6}$$

Τα αποτελέσματα της μεθόδου εκτίμησης που περιγράφηκαν παραπάνω έχουν ένα σοβαρό πρόβλημα. Μικρότερα επιπρόσθετα regrets ϵ τείνουν να εξηγούνται καλύτερα από μικρότερες αξίες από ότι τα πραγματικά regrets. Φαίνεται ότι καλύτερα αποτελέσματα εκτίμησης επιτυγχάνονται υποθέτοντας ότι ο learning αλγόριθμος που χρησιμοποιείται από κάθε παίκτη επιτυγχάνει το καλύτερο πιθανό multiplicative regret.

Άρα τώρα μεταβάλλουμε λίγο την αρχική εξίσωση και παίρνουμε ότι το multiplicative regret του δ επιτυγχάνεται υπό την αξία v αν:

$$\forall b' \in B_i \quad \frac{1}{T}\sum_{t=1}^{T} U_i(b^t, v_i) \geq (1-\delta)\frac{1}{T}\sum_{t=1}^{T} U_i(b', b^t_{-i}, v_i) \tag{4.7}$$

Ορίζοντας τη μέση πιθανότητα κλικαρίσματος και το μέσο κόστος που έχει ο παίκτης i υπό την πραγματική αλληλουχία πονταρισμάτων ως P_0 και C_0 παίρνουμε ότι:

$$\forall b' \in B_i \quad v\Delta P(b') \leq \Delta C(b') + \frac{\delta}{1-\delta}(vP_0 - C_0) \tag{4.8}$$

Ο στόχος μας είναι να ελαχιστοποιήσουμε το multiplicative regret δ. Αυτή η αλλαγή λύνει τα προβλήματα της μεθόδου εκτίμησης που οφείλονται στο γεγονός ότι μικρές αξίες εξηγούν καλύτερα μικρά επιπρόσθετα regrets. Ας εξηγήσουμε γιατί.

Αρχικά παρατηρήστε ότι ένα multiplicative regret του δ αντιστοιχεί σε ένα επιπρόσθετο regret $\frac{\delta}{1-\delta}(vP_0 - C_0)$. Υποθέτοντας ότι το δ βρίσκεται στο $[0, 1]$, έχουμε ότι η συνάρτηση $f(\delta) = \frac{\delta}{1-\delta}$ είναι αύξουσα. Άρα, η ελαχιστοποίηση του δ είναι ισοδύναμη με την ελαχιστοποίηση του $\frac{\delta}{1-\delta}$. Επιπρόσθετα, παρατηρήστε ότι $\frac{\epsilon}{(vP_0 - C_0)} = \frac{\delta}{1-\delta}$. Αυτό σημαίνει ότι μεγαλύτερες αξίες είναι τώρα πιο πλεονεκτικές από ότι οι μικρότερες. Συνοψίζοντας, λόγω κάποιων προβλημάτων προτιμάμε να χρησιμοποιούμε την multiplicative regret minimization εκτίμηση ωφέλειας αντί για την προσθετική.

Τώρα, η μόνη ερώτηση που απομένει να απαντηθεί είναι το πώς θα έπρεπε να υλοποιήσουμε αυτή τη μέθοδο εκτίμησης. Αρχικά παρατηρήστε ότι αν αντκαταστήσουμε τον $\frac{\delta}{1-\delta}$ όρο με ένα κ, τότε κάθε περιορισμός είναι τετραγωνική κυρτή συνάρτηση. Άρα, η τομή των περιορισμών προφανώς σχηματίζει κυρτο σύνολο. Επιπλέον, αφού το να ελαχιστοποιήσουμε το δ είναι ταυτόσημο με το να ελαχιστοποιήσουμε το $\frac{\delta}{1-\delta}$, καταλήγουμε ότι η μέθοδος εκτίμησης μπορεί να γραφτεί ως κυρτό πρόγραμμα. Επιπλέον, οι μέθοδοι ελαχιστοποίησης που είναι εφαρμόσιμες γενικά στη συνθήκη κυρτής βελτιστοποίησης μπορούν προφανώς να χρησιμοποιηθούν σε αυτό το συγκεκριμένο πρόβλημα.

$$\begin{aligned} &elaqistopohse \quad \kappa + 0 \cdot v \\ &subject \quad to \quad v\Delta P(b') \leq \Delta C(b') + \kappa(vP_0 - C_0) \quad \forall b' \in B_i \end{aligned} \tag{4.9}$$

Τώρα, μία επιλογή είναι να χρησιμοποιήσουμε μεθόδους κυρτής βελτιστοποίησης για να βρούμε την αξία που εξηγεί καλύτερα την ελαχιστοποίηση του προηγούμενου προβλήματος. Μια άλλη επιλογή είναι να διαιρέσουμε το χώρο αξίας και να επιλύσουμε κάθε πρόγραμα ξεχωριστά. Προφανώς τώρα, καθώς η αξία σε κάθε παράδειγμα του γενικού κυρτού προγράμματος είναι σταθερή, το γενικό τετραγωνικό κυρτό πρόγραμμα μετατρέπεται σε ένα πλήθος γραμμικών προγραμμάτων. Μετά επιλέγουμε το παράδειγμα της αξίας που εξηγείται καλύτερα (δηλαδή ελαχιστοποιεί βέλτιστα το κ). Πιο τυπικά, υποθέτοντας ότι η αξία φράσσεται στο $[0, v_{max}]$, επιλέγουμε $0 = v^1 < v^2 < ... < v^l = v_{max}$ για να διαιρέσουμε εξίσου σε l τμήματα το χώρο αξίας. Θέτουμε ως V το σύνολο των αξιών αυτών που δημιουργούν το εξής πρόγραμμα για κάθε $v^j \in V$

$$\begin{aligned} &elaqistopohse \quad \kappa^j \\ &subject \quad to \quad v^j\Delta P(b') \leq \Delta C(b') + \kappa^j(v^jP_0 - C_0) \quad \forall b' \in B_i \end{aligned} \tag{4.10}$$

Πρέπει να σημειώσουμε ότι αυτό δεν είναι γραμμικό πρόβλημα γενικής μορφής, καθώς μπορεί να υπολογιστεί απλά κοιτώντας την τομή διάφορων ανισοτήτων. Άρα, τώρα προβλέπουμε ότι η αξία ενός συγκεκριμένου παίκτη είναι αυτή που επιτυγχάνει το καλύτεροΣο κ^j.

4.3 Περιγραφή του Συστήματος

Για να τεστάρουμε την μέθοδο εκτίμησης αξίας που προτάθηκε, κατασκευάζουμε ένα σύστημα προσομοίωσης πονταρισμάτων, το οποίο αναπτύσσεται ακόμα για να διευρύνει τις δυνατότητές του. Ο κύριος στόχος

είναι να δημιουργηθεί ένα περιβάλλον κατάλληλο για τη διεξαγωγή πειραμάτων. Η ανάγκη για ένα τέτοιο περιβάλλον είναι προφανής, καθώς λείπουν από την παγκόσμια κοινότητα δεδομένα που θα επιτρέψουν σε φοιτητές και ερευνητές να διεξάγουν πειράματα δωρεάν χωρίς να εμπλέκονται δικαιώματα κερδοσκοπικών οργανισμών. Ο προσομοιωτής ποντάρισμάτων (bidding simulator) γράφτηκε σε Python με χρήση αντικειμενοστρεφών τεχνικών για να διαβάζεται και μελετάται εύκολα. Ο κώδικας μπορεί να βρεθεί στον εξής σύνδεσμο `ηττπς://γιτηυβ.ςομ/ανδρεασρ27`

Προχωράμε περιγράφοντας τη μορφή των κύριων κλάσεων:

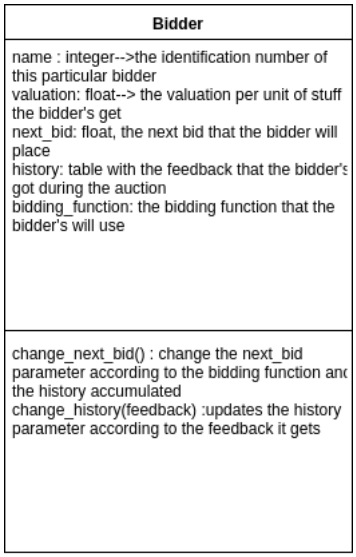
Auction
m: integer --> the number of available slots n: integer --> the maximum number of bidders a: float vector --> the position coefficients g : float vector --> the players coefficients r : the reserve rankscore or price of the auction history : a table of the bids submitted over time
slot(player_id,b) returns the position obtained by player_id under the bidding profile b who(pos,b) returns the id of the player who earned the position "pos" under the bid profile b cost(player_id,pos,b) returns the cost which player_id suffers under the bid profile b if he takes the position "pos" and his advertising is clicked exp_cost(player_id,b) returns the expected cost which a player will suffer under the bid profile b exp_ click(player_id,b) return the expect click probability of a player under the bid profile b update_auction_history(b) updates the history variable to keep track of the submitted bids display functions: display_reserve_rankscore() display_slots() display_players() display_position_clicks_prob() display_player_clicks_prob() display_scores() display_auction() : prints all the auction information into a general auction format

Η μορφή κλάσης Auction παρέχει ένα γενικό τρόπο που μπορεί να υλοποιηθεί ανάλογα με το ποια δημοπρασία θέλουμε να αναπαράγουμε. Για να διευκρινίσουμε περαιτέρω αυτό το γεγονός, υπογραμμίζουμε πόσο εύκολο είναι να υλοποιήσουμε διαφορετικές δημοπρασίες απλά αλλάζοντας ελάχιστες γραμμές κώδικα. Αν θέλουμε να αναπαράγουμε μία GSP δημοπρασία, τότε οι συναρτήσεις θα πρέπει να υλοποιηθούν σύμφωνα με το μοντέλο που περιγράφηκε πριν . Αν θέλουμε να αλλάξουμε το πείραμα

για να τεστάρουμε κάποιες ιδιότητες της δημοπρασίας που παράγεται από το λήμμα του Μψερσον, τότε πρέπει μόνο να αλλάξουμε τη συνάρτηση κόστους σύμφωνα με την γνωστή εξίσωση. Αν θέλουμε να αλλάξουμε σε μία δημοπρασία Πρώτης Τιμής, τότε και πάλι αρκεί να αλλάξουμε την συνάρτηση κόστους.

Προφανώς, πολύ περισσότερα χαρακτηριστικά θα μπορούσαν να μελετηθούν. Όπως αναφέρουμε στην τελευταία ενότητα , μία δυναμική ελάχιστη τιμή απόκτησης θα έπρεπε να χρησιμοποιείται για να αυξήσει τα έσοδα των δημοπρατών. Στην πραγματικότητα, η παράμετρος ιστορικού της δημοπρασίας, η οποία κρατάει όλα τα πονταρίσματα που έχουν γίνει, καθώς και η μέθοδος εκτίμης της αξίας θα μπορούσαν να χρησιμοποιηθούν για αυτό τον σκοπό.

Συνεχίζουμε παρουσιάζοντας τη γενική μορφή που προτάθηκε για κάθε παίκτη.



Τώρα μπορούμε να μεταβάλουμε πολλές παραμέτρους. Μεταβάλλουμε την παράμετρο συνάρτησης πονταρίσματος για να μελετήσουμε την αποδοτικότητα διαφορετικών συναρτήσεων πονταρίσματος. Προφανώς, οι συναρτήσεις πονταρίσματος πρέπει να είναι συμβατές με το feedback των παικτών. Έτσι μπορούμε να κυμανθούμε ανάμεσα σε full information και non full information περιβάλλοντα. Feedback θα είναι το κέρδος του παίκτη αλλά και το πλήρες διάνυσμα πονταρισμάτων, το οποίο προφανώς θα επιτρέψει τη δημιουργία αλγορίθμων για πιο ακριβή μάθηση (precise learning).

Ένα άλλο ενδιαφέρον στοιχείο είναι ότι το feedback που συσσωρεύεται στην παράμετρο του ιστορικού επιτρέπει διαφορετικές υλοποιήσεις των αλγορίθμων που τις εκμεταλλεύονται. Θα μπορούσαμε να χρησιμοποιήσουμε μία greedy προσέγγιση στο ποντάρισμα απαντώντας μόνο στις τελευταίες επαναλήψεις και άρα απαντώντας καλύτερα με επιθετικό τρόπο ή χρησιμοποιώντας ομαλές προσεγγίσεις. Στην πραγματικότητα είναι πιθανό να υλοποιήσουμε no regret αλγορίθμους που ανανεώνουν τα βάρη τους παίρνοντας υπόψη μόνο τις τελευταίες επαναλήψεις και διαγράφοντας τη ιστορικό της δημοπρασίας.

Όταν τρέχουμε την αυςτιον, δημιουργείται ένα .auction αρχείο που αποτελεί την περιγραφή της δημοπρασίας που θα είναι διαθέσιμη στις μεθόδους εκτίμησης. Σε αυτή την περιγραφή, οι πληροφορίες που είναι διαθέσιμες είναι:

- Ο αριθμός των slots
- Ο μέγιστος αριθμός παικτών
- οι συντελεστές κλικαρίσματος θέσης
- οι συντελεστές κλικαρίσματος παικτών
- τα σκορ των παικτών(αυτή είναι μία ιδαιτερότητα μίας συγκεκριμένης δημοπρασίας, την οποία αφαιρέσαμε από τη γενική περιγραφή αλλά την υλοποιήσαμε σε περίπτωση που αποδεικνυόταν χρήσιμη για μελλοντική έρευνα)
- η ελάχιστη τιμή απόκτησης
- το προφίλ πονταρίσματος της πρώτης επανάληψης
- το προφίλ πονταρίσματος της δεύτερης επανάληψης
- κλπ...

Η ιδέα πίσω από τη μορφή της δημοπρασίας είναι να δώσουμε όλες τις πληροφορίες που θα συσσωρεύσει ο μηχανισμός κατά τη δημοπρασία και έτσι η μελέτη των νο ρεγρετ δυναμικών σαν να ήμασταν δημοπράτες.

Επιπρόσθετα, δημιουργήσαμε την αρχική μορφή μίας βιβλιοθήκης για να τεστάρουμε no regret δυναμική σε δημοπρασίες. Το όνομα ης βιβλιοθήκης είναι **regret.py**. Η βιβλιοθήκη παρέχεται με δύο βασικές αξίες εκτίμησης, άρα η επιπλέον μέθοδος regret και η multiplicative regret. Το κύριο χαρακτηριστικό της βιβλιοθήκης αυτής είναι η επιθυμία να αποτελέσει το πρώτο βήμα προς μια πιο γενική και ξρήσιμη library συναρτήσεων.

Τελικά δημιουργήσαμε μια βιβλιοθήκη που θα μπορούσε να χρησιμοποιηθεί για να οπτικοποιήσει διάφορες μετρικές. Αυτή είναι η βιβλιοθήκη **stats.py**. Η βιβλιοθήκη περιείχε συναρτήσεις οπτικοποίησης των rationalizable συνόλων, την μέση θέση του κάθε παίκτη, την εξέλιξη του μέσου πονταρίσματος για ένα συγκεκριμένο παίκτη μέσα στη δημοπρασία.

4.4 Πειραματικά αποτελέσματα

Σε αυτή την ενότητα θα περιγράψουμε το σύστημα το οποίο σχεδιάσαμε για να υλοποιήσουμε την μέθοδο εκτίμησης της αξίας όπως αναλύεται στην προηγούμενη ενότητα και μετά θα παρουσιάσουμε τα πειραματικά αποτελέσματα.

ο σύστημά μας ήταν όσο πιο κοντινό γινόταν σε πραγματικά συστήματα πονταρίσματος που χρησιμοποιούνται από ονλινε αδ εταιρείες. Ειδικά, το setting της δημοπρασία είναι αυτό που περιγράφηκε προηγουμένως και η υλοποίηαη του κανόνα διανομής και του κανόνα πληρωμής ήταν σύμφωνες με την GSP δημοπρασία.

4.4.0.1 Πώς ποντάρουν οι παίκτες

Ο αλγόριθμος που χρησιμοποιείται για να προσομπιώσει τα πονταρίσματα των παικτών [21]. Αρχικά επειδή η απλότητα του τον κάνει πολύ ελκυστικο και δεύτερον επειδή επιτυγχάνει σχεδόν βέλτιστο φράγμα regret.

Ποιος είναι ο χώρος πονταρισμάτον για κάποιο παίκτη· Καθώς χρησιμοποιήσαμε τον EXP3 αλγόριθμο για να προσομοιώσουμε την συμπεριφορά πονταρίσματος των παικτών που μαΘαίνουν, είναι φανερό ότι μία καλή τακτική για έναν παίκτη είναι να μοιράσει εξίσου το χώρο ποναρισμάτων του και να βρεί ανάμεσα σε όλες τις πιθανές πράξεις, την αλληλουχία που του δίνει καλό φράγμα στο regret του κέρδους. Επιπλέον, η υποστήριξη του χώρου πονταρισμάτων πρέπει να επιλεχθεί. Για τα πειράματα, το support του παίκτη είναι $[0, v_i]$, όπου v_i είναι η αξία του. Πιο τυπικά

$$max_{b' \in B_i} \left(\sum_{t=1}^{T} U_i(b', b^t_{-i}, v_i) - \sum_{t=1}^{T} U_i(b^t, v_i) \right) \tag{4.11}$$

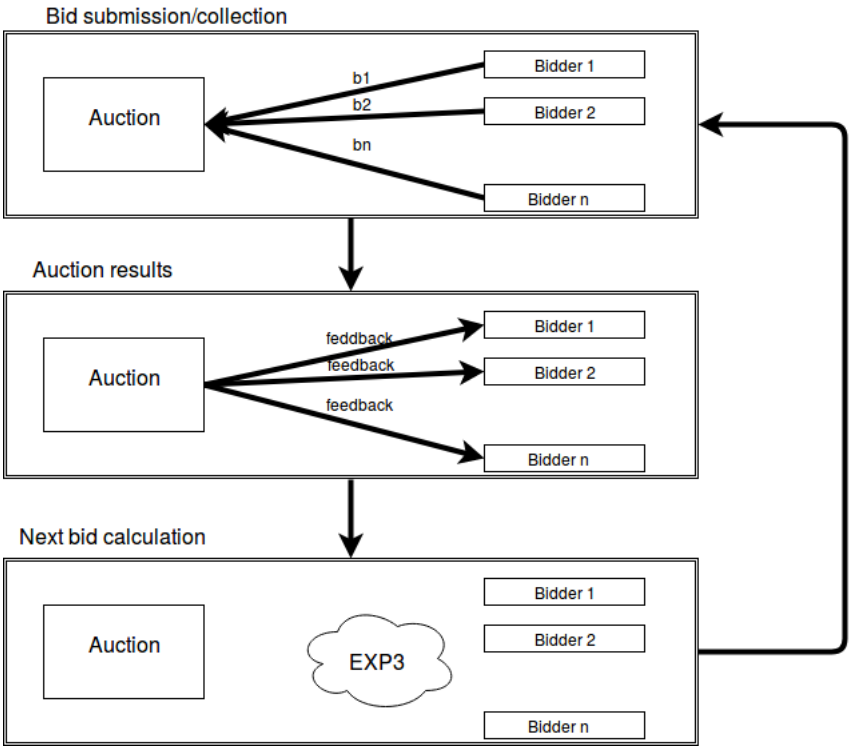
Είναι ισοδύναμο με:

$$min_{b' \in B_i} \left(\sum_{t=1}^{T} \left(-U_i(b', b^t_{-i}, v_i) \right) - \sum_{t=1}^{T} \left(-U_i(b^t, v_i) \right) \right) \tag{4.12}$$

$$min_{b' \in B_i} \left(\sum_{t=1}^{T} \left(u_{max} - U_i(b', b^t_{-i}, v_i) \right) - \sum_{t=1}^{T} \left(u_{max} - U_i(b^t, v_i) \right) \right) \tag{4.13}$$

Τα νέα losses είναι προφανώς θετικά και το πρόβλημα ελαχιστοποίησης είναι τελείως ισοδύναμο, άρα λύσαμε το πρώτο μας ζήτημα. Το δεύτερο σύστημα είναι λόγω του γεγονότος ότι ο αλγόριθμος EXP3 προβλέπει losses στο $[0, 1]$. Αυτό το θέμα δεν είναι πολύ σχετικό, καθώς επηρεάζει μόνο κατά ένα σταθερό παράγοντα ίσο με τη μέγιστη δυνατή loss. Άρα η ασυμπτωτικη vanishing ιδιότητα του μέσου regret δε θα επηρεαστεί.

Επεξηγηματικό διάγραμμα του συστήματος προσομοίωσης:



4.4.0.2 Ρυθμίσεις και αποτελέσματα

Αρχικά, θα παρουσιάσουμε τις ρυθμίσεις πάνω στις οποίες κάναμε το πείραμά μας. Αν και η αλλαγή των παραμέτρων στο πείραμα δεν επηρεάζει ιδιαίτερα την μέθοδο εκτίμησης και τη συμπεριφορά του συστήματος, θα παρουσιάσουμε αυτές τις ειδικές παραμέτρους.

(α′) Ο αριθμός των θέσεων $= 5$

(β′) Ο μέγιστος αριθμός παικτών $= 6$

(γ′) συνιστώσες θέσης $a = [1.0, 0.9, 0.75, 0.55, 0.3]$

(δ′) συνιστώσ· παίκτη $\gamma = [0.1, 0.08, 0.07, 0.07, 0.06, 0.07]$

(ε′) ελάχιστο ποντάρισμα $r = 15$

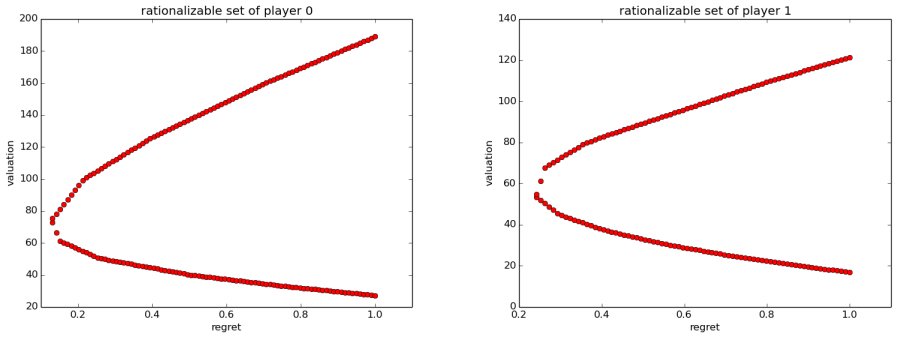
(ϛ′) Οι αξίες των παιδιών $v = [72, 61, 53, 46, 39, 33]$

Επιπλέον, πρέπει να σημειώσουμε ότι κάθε παίκτης είχε στο ποντάρισμά του 20 διαφορετικές πιθανές πράξεις που διαιρούσαν σε n διαστήματα το $\left[0, v^i_{max}\right]$.

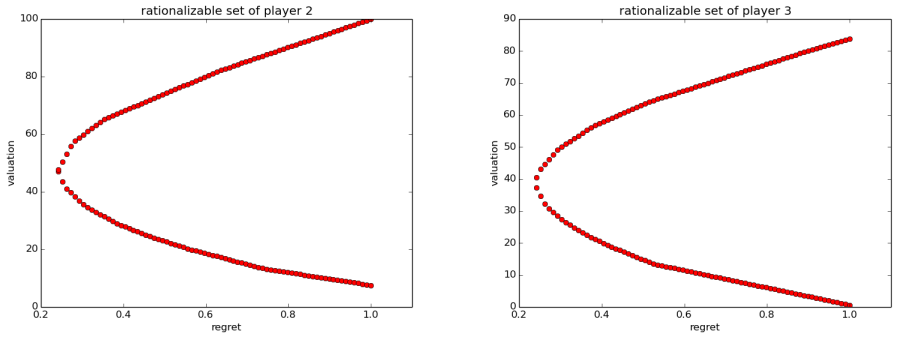
Για τη μέθοδο εκτίμησης, προφανώς επιλέξαμε αυτήν που να ελαχιστοποιεί το multiplicative regret, διαιρώντας το χώρο αξίας σε 20 διαφορετικές αξίες.

Πρόκειται για επαναλαμβανόμενη δημοπρασία που έτρεξε για T=100,000 επαναλήψεις. Όπως αυξάνει το T, η μέθοδος εκτίμησης αξίας δουλεύει καλύτερα επειδή οι παίκτες επιτυγχάνουν minimized regret.

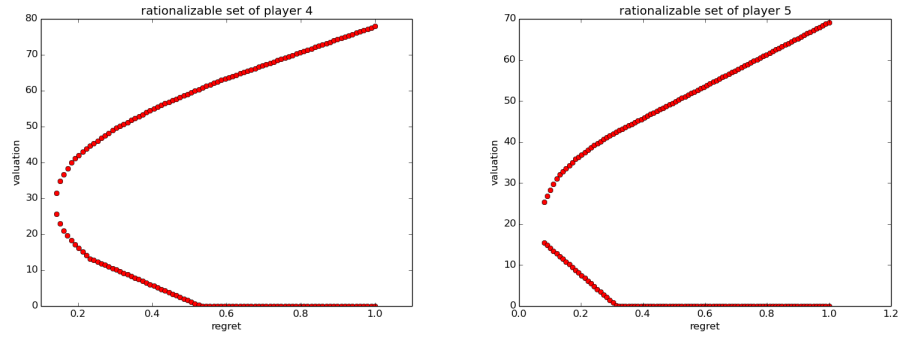
RATIONALIZABLE ΣΥΝΟΛΑ Παρουσιάζουμε το ρατιοναλιζαβλε σύνολο για κάθε παίκτη στο πείραμά μας.



(α′) αξίες 72 και 61



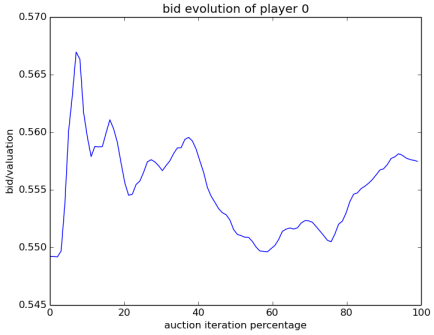
(β′) αξίες 53 και 46

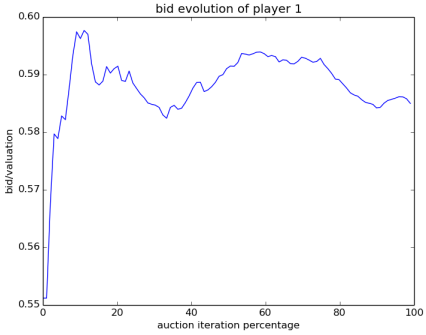


(γ′) αξίες 39 και 33

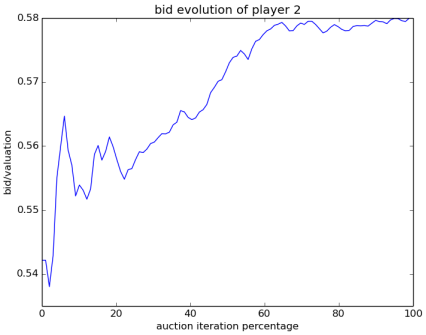
Παρατηρήσεις: Εξετάζουμε μόνο το προσθετικό regret και την αξία κάθε παίκτη.

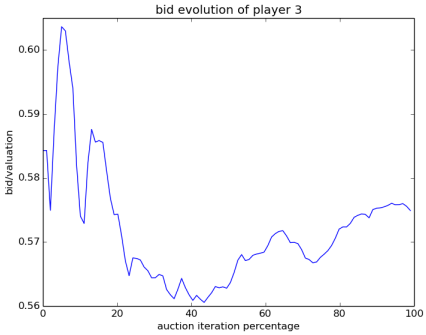
BID EVOLUTION Προχωράμε παρουσιάζοντας την μέση ανάπτυξη πονταρισμάτων σε πίνακα percent για την αξία κάθε παίκτη:



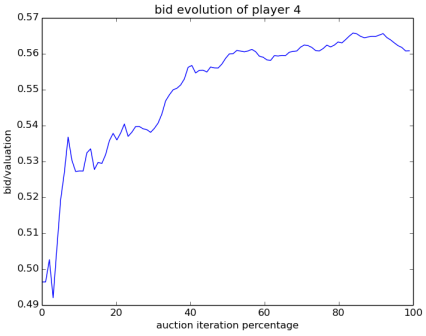


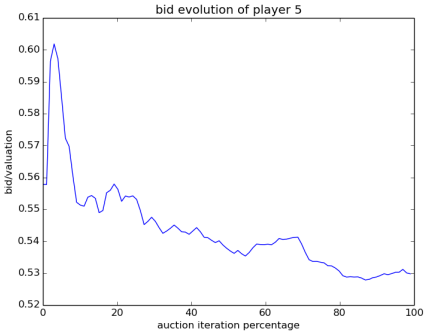
(α′) αλυατιονς 72 ανδ 61





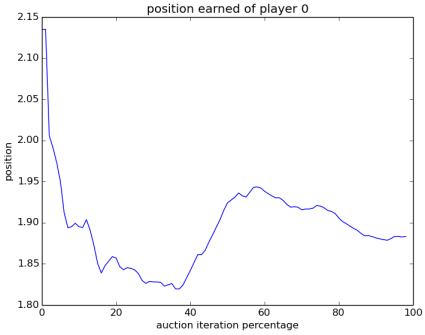
(β′) αλυατιονς 53 ανδ 46

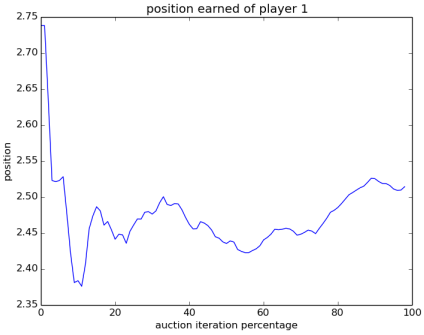




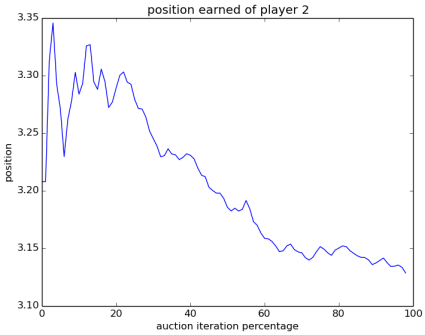
(γ′) αλυατιονς 39 ανδ 33

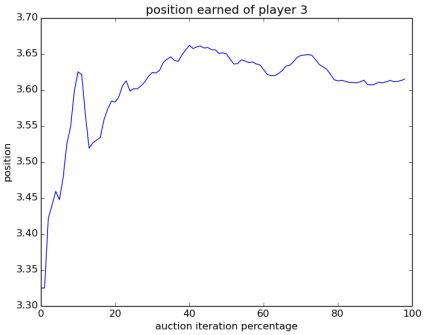
Εξέλιξη Θέσης Θα παρουσιάσουμε την εξέλιξη όσο η δημοπρασία επαναλαμβάνεται στη μέση θέση που κάθε παίκτης παίρνει.



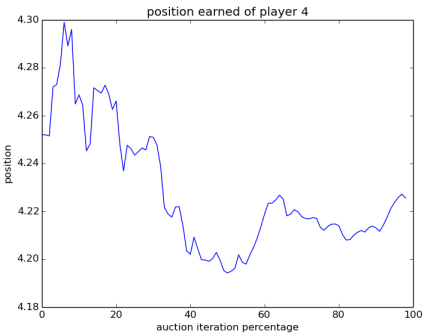


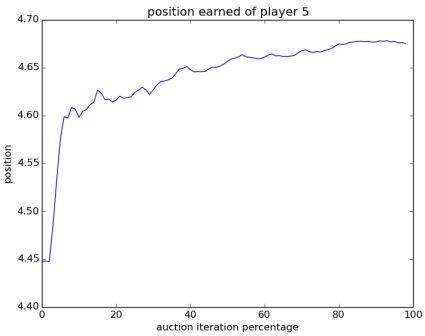
(α′) αλυατιονς 72 ανδ 61





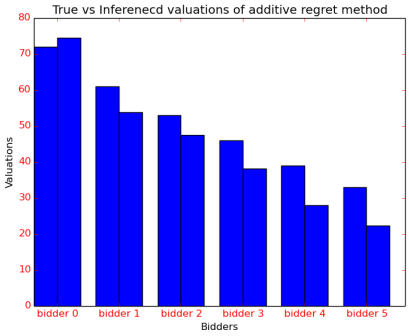
(β′) αλυατιονς 53 ανδ 46

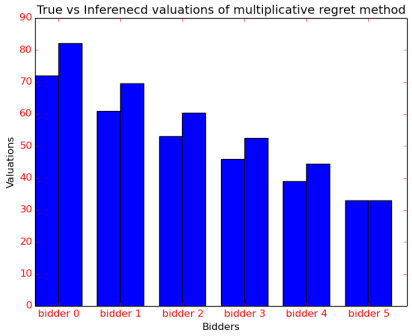




(γ′) αλυατιονς 39 ανδ 33

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΙΜΗΣΗΣ Προχωράμε παρουσιάζοντας το αποτέλεσμα της μεθόδου εκτίμησης ελαχιστοποιώντας το multiplicative regret και αυτό που μηδένιζε το αθροιστικό regret.





Μετρήσαμε το λάθος εκτίμησης των δύο μεθόδων συγκρίνοντας το μέσο ποσοστό σφαλμάτων σε σχέση με τις αξίες κάθε χρήστη. Τυπικά, ορίζοντας ως $\hat{v_i}$ την εκτιμώμενη αξία του i, η εξίσωση ήταν:

$$\sum_{i=1}^{n} \frac{|v_i - \hat{v_i}|}{v_i} \tag{4.14}$$

(α′) αδδιτιε ρεγρετ μετηοδ′ς ερρορ = 17.2%

(β′) μυλτιπλιςατιε ρεγρετ μετηοδ′ς ερρορ = 11,7%

4.5 Με την οπτική της Ισορροπίας

Όπως αποδείχτηκε στο [8] οι κατανεμημένοι αλγόριθμοι no-regret συγκλίνουν σε ένα no regret distributed αλγόριθμο που προσεγγίζει ένα ϵ προσεγγίζοντας την CCE ισορροπία. Άρα μπορούμε να συμπεράνουμε ότι το να εκτιμάμε την αξία του παίκτη σε μία online διαφημιστική δημοπρασία με τη [4] μέθοδο ισοδυναμεί με το να βρεθεί το ζευγάρι των αξιών που μηδενίζει καλύτερα το regret των παικτών. Προφανώς, αυτή η μέθοδος εκτίμησης μπορεί να εφαρμοστεί σε ένα συγκεκριμένο παίκτη αντί για όλο το σύνολο των παικτών και άρα να υποτεθεί ότι μόνο αυτός ο συγκεκριμένος παίκτης χρησιμοποιεί έναν no regret αλγόριθμο. Ωστόσο, όταν αυτή η μέθοδος εκτίμησης εφαρμόζεται σε όλο το σύνολο των παικτών και η μόνη διαφορετική προσέγγιση από την κλασική είναι μία μικτή ισορροπια κατά Ναση,τότε αυξάνουμε το πιθανό σύνολο ισορροπιών. Μία ενδιαφέρουσα ερώτηση θα ήταν αν προσπαθούσαμε να δοκιμάσουμε τη

μέθοδο εκτίμησης κάτω απο διαφορετικές έννοιες ισορροπίας, ποια θα ήταν τα αποτελέσματα.

4.6 Μελλοντική Έρευνα

Κάποιες ενδιαφέρουσες ιδέες για μελλοντική έρευνα συζητιούνται στο [23]. Ο συνδυασμός διαφορετικών μεθόδων αξιολόγησης με τεχνικές machine learning θα ήταν ενδιαφέρουσα ιδέα. Επιπλέον, δοκιμάζοντας διάφορες μετρικές ομοιότητας ανάμεσα στις δημοπρασίες, για να εκτιμηθεί η αξία κάθε παίκτη με κλασσικές τεχνικές λεαρνινγ θα ήταν επίσης μια ενδιαφέρουσα προοπτική. Εκτός αυτού, υπάρχουν κάποιες δυναμικές οπτικές που καμία εκτίμηση regret δε λαμβάνει υπόψη. Κατά πόσο μπορεί μια συγκεκριμένη αξία να εξηγήσει την εξέλιξη·

Μία τελείως διαφορετική κατεύθνση είναι αυτή που προτείνει τρόπους για τον δημοπράτη να μεγιστροποιήσει το κέρδος του. Οι αρχικές πληροφορίες που δόθηκαν στους παίκτες παράγουν διαφορετικά steady state outcomes· Πώς μπορούμε δυναμικά να θέσουμε μία ελάχιστη τιμή απόκτησης για να μεγιστοποιηθούν το κέρδος αλλάζοντας την ισορροπία· Από την οπτική των παικτών, όταν πάνε σε περιβάλλοντα full information, υλοποιούν full information no regret αλγορίθμους αντί για bandit. Παίζει σχετικό ρόλο η διαφορά·

Τέλος, μία ενδιαφέρουσα ερώτηση θα ήταν πώς μπορούν οι παίκτες να μεγιστοποιήσουν το κέρδος τους πέραν της συμπεριφοράς no regret. Επίσης, η υλοποίησή μας βασιζόταν σε ένα διακριτό μοντέλο για bidding (δηλαδή τον EXP αλγόριθμο).μπορεί ένας αλγόριθμος συνεχούς χώρου όπως η μορφή mab υλοποιημένος με gradient descent να ξεπεράσει τον διακριτό;

Bibliography

[1] Emmanuel Guerre, Isabelle Perrigne, and Quang Vuong. Optimal nonparametric estimation of first-price auctions. *Econometrica*, 68(3):525–574, 2000. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/2999600.

[2] Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, December 2007. URL https://ideas.repec.org/a/eee/indorg/v25y2007i6p1163-1178.html.

[3] Susan Athey and Denis Nekipelov. A structural model of sponsored search advertising auctions. In Sixth ad auctions workshop, May 2010.

[4] Denis v, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, pages 1–18, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3410-5. doi: 10.1145/2764468.2764522. URL http://doi.acm.org/10.1145/2764468.2764522.

[5] Paul Dütting, Felix A. Fischer, and David C. Parkes. Truthful outcomes from non-truthful position auctions. *CoRR*, abs/1602.07593, 2016. URL http://arxiv.org/abs/1602.07593.

[6] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961. URL https://EconPapers.repec.org/RePEc:bla:jfinan:v:16:y:1961:i:1:p:8-37.

[7] Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6 (1):58–73, February 1981. ISSN 0364-765X. doi: 10.1287/moor.6.1.58. URL http://dx.doi.org/10.1287/moor.6.1.58.

[8] Tim Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 1st edition, 2016. ISBN 131662479X, 9781316624791.

[9] John F. Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[10] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, March 2007. doi: 10.1257/aer.97.1.242. URL http://www.aeaweb.org/articles?id=10.1257/aer.97.1.242.

[11] Sanjev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 8:121 – 164, 2012.

[12] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212 – 261, 1994. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1994.1009. URL http://www.sciencedirect.com/science/article/pii/S0890540184710091.

[13] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291 – 307, 2005. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2004.10.016. URL http://www.sciencedirect.com/science/article/pii/S0022000004001394. Learning Theory 2003.

[14] Yoav Freund and Robert 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, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL http://www.sciencedirect.com/science/article/pii/S002200009791504X.

[15] J. Robinson. An iterative method of solving a game. *The Annals of Mathematics*, 54:296–301, September 1951.

[16] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999. URL https://EconPapers.repec.org/RePEc:eee:gamebe:v:29:y:1999:i:1-2:p:79-103.

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

[18] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016. ISSN 2167-3888. doi: 10.1561/2400000013. URL http://dx.doi.org/10.1561/2400000013.

[19] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 928–935. AAAI Press, 2003. ISBN 1-57735-189-4. URL http://dl.acm.org/citation.cfm?id=3041838.3041955.

[20] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 09 1952. URL https://projecteuclid.org:443/euclid.bams/1183517370.

[21] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003. ISSN 0097-5397. doi: 10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

[22] Avrim Blum, Yishay Mansour, and Jamie Morgenstern. Learning valuation distributions from partial observation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 798–804, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9550.

[23] Noam Nisan and Gali Noti. An experimental evaluation of regret-based econometrics. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 73–81, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4913-0. doi: 10.1145/3038912.3052621. URL https://doi.org/10.1145/3038912.3052621.

[24] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[25] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007. ISBN 0521872820.

[26] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135.

[27] Brown. *Analysis of Production and Allocation*. Wiley, 1951.

[28] Howard Karloff. *Linear Programming*.

[29] Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, pages 1040–1053, 1982.

[30] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928. URL `http://eudml.org/doc/159291`.

[31] Shuchi Chawla, Jason Hartline, and Denis Nekipelov. A/b testing of actions. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 19–20, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3936-0. doi: 10.1145/2940716.2940757. URL `http://doi.acm.org/10.1145/2940716.2940757`.

[32] Edward Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971. URL `https://EconPapers.repec.org/RePEc:kap:pubcho:v:11:y:1971:i:1:p:17-33`.

[33] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973. URL `https://EconPapers.repec.org/RePEc:ecm:emetrp:v:41:y:1973:i:4:p:617-31`.

[34] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000. URL `https://EconPapers.repec.org/RePEc:ecm:emetrp:v:68:y:2000:i:5:p:1127-1150`.

[35] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007. URL `http://dl.acm.org/citation.cfm?id=1314543`.

[36] Tim Roughgarden, Vasilis Syrgkanis, and Éva Tardos. The price of anarchy in auctions. *CoRR*, abs/1607.07684, 2016. URL `http://arxiv.org/abs/1607.07684`.

[37] Shuchi Chawla, Jason D. Hartline, and Denis Nekipelov. Mechanism design for data science. *CoRR*, abs/1404.5971, 2014. URL http://arxiv.org/abs/1404.5971.

[38] Michael Ostrovsky and Michael Schwarz. Reserve prices in internet advertising auctions: A field experiment. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 59–60, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0261-6. doi: 10.1145/1993574.1993585. URL http://doi.acm.org/10.1145/1993574.1993585.

[39] L S. Shapley and Martin Shubik. The assignment game i: The core. 1:111–130, 12 1971.

[40] Paul Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004. URL https://EconPapers.repec.org/RePEc:cup:cbooks:9780521536721.

[41] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Conference on Algorithms*, ESA'07, pages 253–264, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-75519-5, 978-3-540-75519-7. URL http://dl.acm.org/citation.cfm?id=1778580.1778606.

[42] Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26(2):305–321, 1978. doi: 10.1287/opre.26.2.305.