

# ONLINE MECHANISM DESIGN

YIANNIS GIANNAKOPOULOS

*scholar of*

PROPONDIS FOUNDATION

Supervised by

PROF. ELIAS KOUTSOUIAS

M.Sc. Thesis

*submitted to the*

Graduate Program in Logic, Algorithms and Computation

ΑΠΛ

ATHENS, SEPTEMBER 2008



# Acknowledgements

It was through my supervisor's, Professor Elias Koutsoupias's lectures that I fell in love with Theoretical Computer Science and, for that reason, I will always be indebted to him. Being around him and studying under his supervision has been a privilege. I would also like to thank the other two members of my Advisory Committee, namely Professor Costas Dimitracopoulos and Assoc. Professor Evangelos Raptis, for being always there whenever their students want them and for their sincere efforts to make the University of Athens a better place to live and study. I also feel the need to thank all the people at "Propondis" Foundation and especially director Ioannis Baveas for honoring me with one of their very competitive scholarships, as well as for their constant, invaluable moral support.

Throughout the years, and particularly during the writing of this thesis, some very special friends have been patient enough to try decipher and cope with my "algorithmic" way of life. I am very fortunate to have Kostas Tsirkas, Alexandra Schürmann and, of course, my "fantastic" sister Angeliki around me. Finally, I would like to thank Penny Tzevelekou for being the wonderful person she is and for constituting the dominant strategy equilibrium of my life.

This work is dedicated to my grandparents Varvara and Yiannis K. Giannakopoulos.

YG



# Contents

<b>A Preliminaries</b>	<b>1</b>
<b>1 Game Theory</b>	<b>3</b>
1.1 Introduction - Examples . . . . .	3
1.2 Basic Solution Concepts . . . . .	9
1.3 Mixed Strategies and Nash Equilibria . . . . .	13
1.4 Characterizing Nash Equilibria . . . . .	15
1.5 Notes . . . . .	18
<b>2 Mechanism Design</b>	<b>21</b>
2.1 Social Choices and Mechanisms . . . . .	22
2.2 Direct Revelation Mechanisms and Truthfulness . . . . .	28
2.3 VCG Mechanisms . . . . .	33
2.4 Notes . . . . .	39
<b>3 Competitive Analysis</b>	<b>41</b>
3.1 Online Optimization Problems . . . . .	41
3.2 Competitive Analysis . . . . .	42
3.3 Notes . . . . .	45
<b>B Online Mechanism Design</b>	<b>47</b>
<b>4 Online Mechanism Design</b>	<b>49</b>
4.1 Direct-Revelation Mechanisms . . . . .	52
4.2 Limited Misreports . . . . .	54
4.3 Truthfulness . . . . .	55
<b>5 Single-Valued Online Domains</b>	<b>59</b>
5.1 Basic Definitions . . . . .	60

5.2	Monotonicity . . . . .	61
5.3	Truthfulness . . . . .	65
 <b>C Specific Online Auctions</b>		 <b>71</b>
<b>6</b>	<b>Expiring Items Auctions</b>	<b>73</b>
6.1	The Greedy Auction . . . . .	75
6.2	Upper Bound . . . . .	76
6.3	Lower Bound . . . . .	77
6.4	An Impossibility Result . . . . .	79
6.5	Extensions – Open Problems . . . . .	83
 <b>7</b>	 <b>Adaptive, Limited-Supply Auctions</b>	 <b>85</b>
7.1	The Classical Secretary Problem . . . . .	87
7.2	Adaptive Threshold Auctions . . . . .	91
7.3	Upper Bounds . . . . .	94
7.4	Extensions – Open Problems . . . . .	99
 <b>Bibliography</b>		 <b>101</b>
 <b>Index</b>		 <b>105</b>

**Part A**

**Preliminaries**





# Chapter 1

## Game Theory

### 1.1 Introduction - Some well known Games

In this section we will try to introduce some of the fundamental notions of Game Theory through “natural” examples, building a strong intuition before presenting formal definitions later on at section 1.2 and section 1.3. Our analysis at this section is informal and generally our tone is light and narrative, in contrast to the other parts of this project where the attitude is rigorous and quite formal. After all, Game Theory has its foundations built upon social sciences and that is why in this introduction we intentionally act so as to stimulate that kind of motivation and interest from the reader’s side.

#### 1.1.1 Prisoner’s dilemma

Consider the following scenario: The police has arrested two main suspects for a crime, namely Prisoner 1 and Prisoner 2. However, the evidence are insufficient and only a confession could prove full charges. If both prisoners remain silent (i.e. do not confess) then both will serve a small prison term of 2 years, just for minor charges. In case only one of them confesses, he is used as a witness against the other. The one who confessed receives a reduced 1 year sentence because of his “cooperation” and the one who remained silent serves a full sentence of 10 years. Finally, if both confess, there are enough evidence to incriminate them both, however their cooperation grants them a sentence reduction to 5 years. We must also point out that prisoners are interrogated in different rooms, thus **no cooperation** between them is possible. However, every prisoner is completely informed by the policemen of his possible choices, the other prisoner’s choices, as well as of the various results (imprisonment durations) of their choices. No prisoner is informed about the other prisoner’s decision.

Summarizing, there are two choices for every prisoner, namely *confess* or *silent*. We call them **strategies** and say that each prisoner’s strategy set is  $\{\textit{confess}, \textit{silent}\}$ . Every prisoner’s decision is independent of the other’s, so there are exactly four distinct possible outcomes in our scenario, namely  $(\textit{confess}, \textit{confess})$ ,  $(\textit{silent}, \textit{confess})$ ,  $(\textit{confess}, \textit{silent})$  and  $(\textit{silent}, \textit{silent})$ , where the first component of these tuples is chosen by Prisoner 1 and the second by Prisoner 2. We call such a tuple a **strategy profile**, because it captures all prisoners’ selected strategies.

With each strategy profile there is a numerical outcome associated, the sentence each prisoner must serve. Because these numerical values carry a negative effect to the prisoners, we can assign a negative sign to them and see the resulting values as the “gain”<sup>1</sup> of the prisoners who receive these sentences. In this context, every *rational* prisoner obviously tries maximize his gain, i.e. minimize his imprisonment term. We call each prisoner’s gain at a given outcome the prisoner’s **utility** (or payoff) on this strategy profile.

All the above can be represented in an elegant way by the following  $2 \times 2$  matrix at:

		Prisoner 2	
		<i>confess</i>	<i>silent</i>
Prisoner 1	<i>confess</i>	−5, −5	−1, −10
	<i>silent</i>	−10, −1	−2, −2

**Table 1.1:** Prisoner’s Dilemma

Under this representation we can model our scenario as a “game”<sup>2</sup>: Prisoner 1 chooses a row, Prisoner 2 a column and these choices are made *simultaneously*. At each row-column combination (i.e. strategy profile) the utilities of the prisoners are given in the corresponding cell. The first value is Prisoner’s 1 and the second is Prisoner’s 2 (this is by convention). Prisoners 1 and 2 are the two **players** of the game.

### 1.1.2 What Game Theory is all about. Rationality and equilibria.

Although recognizing the rules, strategies and utilities of a game is very important, all this mathematical modeling is not done only to describe game scenarios but for a much more interesting (and powerful) purpose: *to analyze, and even predict, the players’ behavior*. This

---

<sup>1</sup>Although in no way can years in prison be seen as *gain* (no matter how many minus signs we put in front of them...!), here we use the simplifying (though mathematically natural) interpretation: *negative gain = loss*.

<sup>2</sup>We have not formally defined the term yet, however the reader is encouraged, at this point, to rely on his/her intuition about everyday life “games”.

idea is, of course, the backbone of **Game Theory** and is consequently going to underlie all our results and analysis in this thesis.

In order to analyze such behaviors, it is inevitable to make some solid assumptions about *how* players think and act. These assumptions must be *fundamental* (simple and strong enough to capture the essence of behavior and to be applicable to the various categories of games) and *natural* (being compatible with our intuitions and expectations about behavior in such strategic interactions). Luckily enough, for us there is only one such assumption we are going to need: **rationality**. By saying that our players act rationally we mean one and only one, very specific thing: *players behave so as to maximize their (total) utility*.

Let us analyze the behavior of our players at the Prisoner's Dilemma game. Look at table 1.1 and remember that Prisoner 1 chooses lines and Prisoner 2 columns. Let's think for a moment what is Prisoner's 1 best (i.e. utility maximizing) strategy. If he plays line 1 then (depending on what Prisoner 2 plays<sup>3</sup>) there are two possibilities: If Prisoner 2 plays column 1 then Prisoner 1 will receive a utility of  $-5$  (first component of cell  $(1, 1)$ ) and if he plays column 2 then Prisoner 1 receives utility  $-1$ . But if Prisoner 1 chooses line 2 then his utilities are worse, namely  $-10$  and  $-2$  respectively. This shows us that, whatever Prisoner 2 plays, Prisoner's 1 best strategy is to select line 1, i.e. confess. In an analogous way, it is easy to see that also Prisoner's 2 best strategy is confession. So, based on our fundamental assumption of rationality, it is evident that our game will result in both players selecting to confess (strategy profile  $(confess, confess)$ ).

We refer (informally) to such "*stable*" states, which we believe our games are going to result to, as **solutions**<sup>4</sup>. In the Prisoner's Dilemma game the situation was very clear and the  $(confess, confess)$  solution is the obvious *rational* outcome. Not every game (in fact, very few do) possesses such a strong solution (as we will see at our following examples of subsection 1.1.3 and subsection 1.1.4) and is the job of Game Theory to come up with less demanding, yet natural, solution concepts and apply them to predict various **equilibria**<sup>5</sup>, i.e. strategy profiles which seem reasonable and stable enough to be the actual outcomes of our games.

---

<sup>3</sup>Remember that the two players play simultaneously and independently.

<sup>4</sup>The term is used here rather intuitively and no strict connections are to be drawn with the formal, specific use of the term in the classic paper of Nash [1951].

<sup>5</sup>sing. equilibrium

### 1.1.3 Battle of sexes

Now consider the following situation: A boy and a girl try to decide how to spend their evening. They want to go out together, however there is a disagreement about whether to go to a football (a.k.a. soccer) game or to the movies. Although it is irrelevant to our results, let us say that the football match is the Champion’s League final and the movie is the new George Clooney film, just to make the competition a little more intriguing... The description of the game is given by the following  $2 \times 2$  matrix:

		Boy	
		<i>film</i>	<i>football</i>
Girl	<i>film</i>	10, 7	5, 5
	<i>football</i>	1, 1	7, 10

**Table 1.2:** Battle of Sexes

Their first priority is to attend an event *together*, so the boy is ready (for a utility of 7, instead of the optimum 10) to watch for 2 hours a man admittedly much more beautiful than him charming his girlfriend, just to be with her during the movie. In the same way, the girl is ready to tolerate the view of 22 men chasing a ball along a huge field for 2 hours. In the contrary, even if one player manages to attend the event he/she prefers, this is going to give him/her a utility of only 2 if the other one is not with him. Of course, the absolute disaster would be for the players to be apart and also not attend their preferred events (a situation highly unlikely to occur...).

To summarize, this game is all about the love for each other vs. our personal desires, and the question is what a *rational* player would choose to do. In our analysis there is no space for a philosophical<sup>6</sup> approach: maximization of utility is our only goal! Unfortunately, in this case there is no such a “strong” solution as in the Prisoner’s Dilemma game (see table 1.1). This can be easily verified by a quick inspection of the game’s matrix (table 1.2). That means, no player has a strategy which is optimal regardless of the other player’s choices.

What about strategies that are optimal given a *specific* strategy of the other player? That is, what is the **best response**<sup>7</sup> of a player to the other players’ choices? Surely, this is

<sup>6</sup>However, the reader would definitely benefit from giving some thought to the above fundamental question of human nature...

<sup>7</sup>Note here that “response” has no time significance. Our players act simultaneously and independently. What we mean, is how a player should act, given all other players’ choices fixed (and known). It is sometimes useful to think of these fixed strategies as our player’s *predictions* about how all the other players are going to act.

a relaxed notion of equilibrium, compared to that of the Prisoner’s Dilemma, however it seems natural and stable enough: this “best response” schema captures the essence of strategic behaviour. So, if we assume that the girl chooses to go to the film, then the boy’s optimal response is to join her (for a utility of 7, compared to a utility of 5 if he goes to the football game alone.). In an analogous way, it is easy to see that the boy’s best response to the girl playing *football* is to play *football*. The situation is completely symmetric for the best responses of the girl.

The above analysis leads us to the conclusion that whatever one player chooses, the other will benefit from choosing the same strategy. So it is reasonable to propose either profile  $(film, film)$  or  $(football, football)$  (i.e. cells  $(1, 1), (2, 2)$ ) as a stable solution for our game. Here stability has a pretty coherent interpretation: no player has the *incentive* (i.e. utility improvement) to *deviate* (i.e. change his/her strategy) from the above strategy profiles, given that the other players’ strategies will remain unchanged. It is very important to note that the above analysis neither examines nor assumes anything about the *process* through which this steady state is reached. It is just proposed as a reasonable equilibrium for our game’s result.

### 1.1.4 Matching pennies

At the matching pennies game we have two players, each of which possesses a penny<sup>8</sup>. The two players simultaneously show a side of their coins. That is, each announces “heads” or “tails”. If the two announcements coincide, then player 2 gives his penny to player 1. If they differ, player 1 pays player 2. We summarize at the following  $2 \times 2$  matrix:

		Player 2	
		<i>heads</i>	<i>tails</i>
Player 1	<i>heads</i>	1, -1	-1, 1
	<i>tails</i>	-1, 1	1, -1

**Table 1.3:** Matching Pennies

Let us see what happens if we try to locate equilibria similar to those we proposed at the Battle of Sexes game of subsection 1.1.3: Assume that Player 1 intends to play *heads*. Then Player’s 2 best response is to play *tails*, for a utility of 1. But the resulting strategy profile  $(heads, tails)$  could not be a steady result for our game, because Player 1 would be

---

<sup>8</sup>That is \$0.01. The exact amount is irrelevant to our analysis, so we decide here to keep the original formulation of the game, although inflation has made this amount look funny...

better of changing his strategy to *tails*. Then, again Player 2 would have an incentive to change the strategy profile from  $(tails, tails)$  to  $(tails, heads)$ , and so on. This could go on forever, making “circles” around the cells (strategy profiles) of our game matrix. So, in this case, the best-response procedure of the Battle of Sexes deviates in an obvious way: no “convergence” to a simple stable solution is possible.

In order to overcome this obstacle we use an approach which may seem unnatural at a first glance, however computer scientists will feel straight at home: we *randomize*. The term, of course, is very general but here we mean something very simple: instead of asking each player to report his selected strategy, why don't we ask him to report *probabilities* with which he is going to play each strategy. For example, Prisoner 1 of the Prisoner's Dilemma (see table 1.1) could say that “I am going to *confess* with probability  $p = \frac{1}{3}$  and remain silent with probability  $p' = \frac{2}{3}$ ”. Notice that, since these are the only choices of the player,  $p + p' = \frac{1}{3} + \frac{2}{3} = 1$ . These probabilities, one for each strategy of our player, ordered together on a tuple is the **mixed strategy** played by our player. In the above example, all the 2-vector  $(p, p')$  with  $p + p' = 1$ ,  $p, p' \geq 0$  are the possible mixed strategies of Prisoner 1. To distinguish them from the simple strategies, we call the latter **pure strategies**.

Essentially, mixed strategies are probability distributions on the set of pure strategies. Of course, when mixed strategies are used, the utilities-payoffs are conditional upon the probabilities of the pure strategies. Let us demonstrate the notions just introduced, by concentrating on our Matching Pennies game (see table 1.3). Suppose Player 1 plays *heads* with probability  $p_1$  and *tails* with  $p_2$ , i.e. his mixed strategy is  $(p_1, p_2)$ . Suppose Player's 2 mixed strategy is  $(q_1, q_2)$ . Translating this in pure strategies, pure strategy profile  $(heads, heads)$  is played with a probability of<sup>9</sup>  $p_1q_1$ ,  $(heads, tails)$  with  $p_1q_2$ ,  $(tails, heads)$  with  $p_2q_1$  and  $(tails, tails)$  with  $p_2q_2$ . Thus, the expected utilities our players receive are

$$\begin{aligned} u_1 &= p_1q_1 \cdot 1 + p_1q_2 \cdot (-1) + p_2q_1 \cdot (-1) + p_2q_2 \cdot 1 \\ &= (p_1 - p_2)(q_1 - q_2) \\ u_2 &= p_1q_1 \cdot (-1) + p_1q_2 \cdot 1 + p_2q_1 \cdot 1 + p_2q_2 \cdot (-1) \\ &= -(p_1 - p_2)(q_1 - q_2) \end{aligned}$$

We will try to deploy a best-response analysis, similar to what we did for the Battle of Sexes game (see subsection 1.1.3). Suppose Player 2 decides to play *heads* and *tails* with equal probabilities, that is  $q_1 = q_2 = \frac{1}{2}$ . Then, Player's 1 expected utility becomes  $u_1 = (p_1 - p_2)(\frac{1}{2} - \frac{1}{2}) = 0$ , which is independent of his choices (probabilities  $(p_1, p_2)$ ). That means that Player 1 is *indifferent* between his (mixed) strategies. In an analogous way,

<sup>9</sup>Remember, players play independently.

we can see that, if Player 1 plays  $(\frac{1}{2}, \frac{1}{2})$  then Player 2 is indifferent between his strategies. All the above show that if both players play  $(\frac{1}{2}, \frac{1}{2})$  then non of them has an incentive to deviate and change his probabilities (his mixed strategy). So, the *mixed* strategy profile  $((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$  could serve well as an equilibrium<sup>10</sup> for our game.

## 1.2 Definitions and Basic Solution Concepts

In this section we formally introduce the notions we intuitively met at the previous, introductory section 1.1. Here we do not spend much time pointing out all the underlying ideas behind the definitions, as we believe that section 1.1 serves well this purpose.

First of all, we must present a rigorous mathematical definition of what a *game* really is, otherwise we are unable to make even a single solid step towards the rest of this project.

**DEFINITION 1.1 (Game)** A (strategic) game<sup>11</sup>  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  consists of:

- a finite set of players  $\mathcal{N}$ ,  $\mathcal{N} = \{1, 2, \dots, N\}$
- for every player  $i \in \mathcal{N}$ , a set of (pure) strategies  $S_i$  and
- for every player  $i$ , a payoff (or utility) function  $u_i : S \rightarrow \mathbb{R}$  where,  $S = \prod_{i=1}^N S_i$  is the set of all possible strategy profiles.

If  $|S_i| < \infty$  for all  $i \in \mathcal{N}$ , i.e. every player has a finite set of pure strategies, the game is called **finite**. In this thesis we will be primarily interested in finite games, due to the nature of the problems we are going to study and of course Computer Science itself. At section 1.1, we saw that a two player game at which each player has two (pure) strategies can be represented through a  $2 \times 2$  matrix at every entry (cell) (i.e. coordinates of strategies) of which resides an (ordered) pair of the players' utilities for the corresponding strategy profile. If we do not like our matrix consisting of pairs of real values, we can separate this matrix to two matrices, one for the utilities of each player. The first one takes the utilities of the first player (the first coordinate of the ordered pairs) and the other those of the second player. The above concise representation of our simple two-player, two-strategy game can be generalized for arbitrary two-player *finite* games. At the following we assume

<sup>10</sup>This result has also a very strong natural interpretation. It tells us that a game like the Matching Pennies would eventually result to the players *tossing* the (unbiased) coin, instead of choosing what side to report.

<sup>11</sup>These are also called *games in normal form*. This is to make a contrast with the *extensive form* primarily used to model time-*dynamic* games such as multi-stage and repeated games. Extensive form games are very powerful and compose the richest areas within Game Theory, however we are not going to need them explicitly in this thesis and so the interested reader is referred to [Mas-Colell et al., 1995, chapter 9] for a (generous) introduction and to [Fudenberg and Tirole, 1991, parts II, IV] for a more extended analysis on the subject.

we have fixed some ordering upon the (finite) sets of our players' strategies, that is, for every player  $i$  we can see its finite strategy set  $S_i$  as an ordered tuple

$$S_i = (s_{i,1}, s_{i,2}, \dots, s_{i,n_i})$$

where  $n_i = |S_i|$ .

**DEFINITION 1.2 (Utility matrix)** Let  $\mathcal{G} = (\{1, 2\}, \{S_1, S_2\}, \{u_1, u_2\})$  be a two-player game with  $|S_i| = n_i$ . The utility matrix of some player  $i$  is the matrix  $U_i^{\mathcal{G}} \in \mathbb{R}^{n_1 \times n_2}$  for which

$$(U_i^{\mathcal{G}})_{j_1 j_2} = u_i(s_1, s_2).$$

**DEFINITION 1.3 (Game matrix)** Let  $\mathcal{G} = (\{1, 2\}, \{S_1, S_2\}, \{u_1, u_2\})$  be a two-player game with  $|S_i| = n_i$ . The matrix of  $\mathcal{G}$  is the matrix  $U^{\mathcal{G}} \in (\mathbb{R}^2)^{n_1 \times n_2}$  for which

$$U_{j_1 j_2}^{\mathcal{G}} = ((U_1^{\mathcal{G}})_{j_1 j_2}, (U_2^{\mathcal{G}})_{j_1 j_2})$$

Because of Definition 1.2, the expression at Definition 1.3 is equivalent to

$$U_{j_1 j_2}^{\mathcal{G}} = (u_1(s_1, s_2), u_2(s_1, s_2),)$$

which makes the notation a little lighter. The matrix representation of strategic games is the reason why games with only two players are also called *bimatrix games*. They need only two matrices, the utility matrices of the players to be described completely. There are also two other special classes of games that can be easily defined using the matrix notation. *Zero-sum games* are two-player games for which  $U_1^{\mathcal{G}} + U_2^{\mathcal{G}} = \mathbf{0}_{n_1 \times n_2}$ , the zero  $n_1 \times n_2$  matrix. The Matching Pennies game (table 1.3) is a zero-sum game. Such games possess some very "nice". Another interesting class is that of *symmetric games*. These are two-player games with  $U_1^{\mathcal{G}} = [U_2^{\mathcal{G}}]^T$  (obviously,  $n_1 = n_2$ ). Intuitively, this means that we do not care about the identities of the players since they have the same set of strategies and also can switch positions without changing their utilities. Of course, this notion can be generalized for games with more than two players.

The first solution concept to be formally presented is that of *dominant strategies*. It is the "strongest" (and probably the most straightforward) solution concept we are going to need and we informally introduced it during the analysis of the Prisoner's Dilemma game (see table 1.1 and subsection 1.1.2, page 5). It is also probably the most widely used one in



Algorithmic Mechanism Design, due to its worst-case character and its solid and natural interpretation.

**DEFINITION 1.4 (Dominant Strategy Equilibrium<sup>12</sup>)** Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a game. A (pure) strategy profile  $(s_1^*, s_2^*, \dots, s_N^*)$  is a dominant strategy equilibrium for our game  $\mathcal{G}$  if, for every  $i \in \mathcal{N}$ ,  $s_i^*$  is a dominant strategy for player  $i$ , i.e.<sup>13</sup>

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \quad \text{for all } s_i \in S_i, s_{-i} \in S_{-i}.$$

Whatever  $s_{-i}$  the other players choose to play,  $s_i^*$  is the best choice for player  $i$ . Notice, however, that nothing in Definition 1.4 demands our game to necessarily have only a *single* dominant strategy equilibrium. In fact one can easily think of (infinite) games with infinite many dominant strategy equilibria. Consider for example the trivial game at which all players have the same utility for every possible strategy profile.

At a first glance, this may seem as a drawback since we think of our “strongest” solution concept as something very specific. We could overcome this obstacle by demanding the inequality of Definition 1.4 to be a strict one. Then we would have the notion of *strict* dominant strategies and it is trivial to check that no game can have two distinct strict dominant strategy equilibria. In this context one would call our notion of dominant strategies of Definition 1.4 as a *weak* dominant strategy equilibrium. This approach is indeed followed in most Game Theory and Microeconomic Theory textbooks. However, our central subject in this thesis would be Mechanism Design (and in particular Algorithmic Mechanism Design) in the context of which such a notion of strict dominant strategy equilibria would be very restricting. One other reason for which the possible existence of many different dominant strategy equilibrium points is not really a problem for us, is that all these equilibria are in some way (which we are going to explain right away) equivalent to each other. It is trivial to check from the inequality of Definition 1.4 that if  $(s_1^*, s_2^*, \dots, s_N^*)$  and  $(t_1^*, t_2^*, \dots, t_N^*)$  are two dominant strategy equilibria then  $u_i(s_1^*, s_2^*, \dots, s_N^*) = u_i(t_1^*, t_2^*, \dots, t_N^*)$  for every player  $i$ . In an Economic Theory perspective, one would say that every player  $i$  is *indifferent* among the different dominant strategy equilibria points, since they all yield the same utility for him. From a “utilitarian” point of view all these equilibria are equivalent.

<sup>13</sup>Here we use some notation standard in Game Theory: If  $a = (a_1, a_2, \dots, a_n)$  is a  $n$ -vector, then for every  $i = 1, 2, \dots, n$ ,

$$(a_i, a_{-i}) = a \quad \text{and} \quad a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$$

and generally

$$(b, a_{-i}) = (a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n).$$

This utilitarian, indifference approach has a central role in Microeconomic Theory and the interested reader is referred to any introductory textbook in Microeconomics for a first introduction to these ideas (e.g. see [Schotter, 2001, chap. 2])

The next solution concept, was presented informally during the analysis of the Battle of Sexes game at subsection 1.1.3:

**DEFINITION 1.5 (Pure Nash Equilibrium)** *Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a game. A (pure) strategy profile  $(s_1^*, s_2^*, \dots, s_N^*)$  is a pure Nash equilibrium for our game  $\mathcal{G}$  if, for every  $i \in \mathcal{N}$ ,  $s_i^*$  is a best response of player  $i$  to the other players' strategies  $s_{-i}^*$ , i.e.*

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \text{for all } s_i \in S_i.$$

Obviously the pure Nash equilibrium solution concept is weaker than that of dominant strategies and it's easy to see that every dominant strategy equilibrium is also a pure Nash equilibrium. As in the case of dominant strategies, Definition 1.5 allows for a game to have multiple pure Nash equilibria. The example of the trivial game having infinitely many dominant strategy equilibria, which we gave right after Definition 1.4, will suffice since every dominant strategy equilibrium is also a pure Nash equilibrium. Alternatively, think of the two-player, two-strategy game with matrix

		Player 2	
		$s_{2,1}$	$s_{2,2}$
Player 1	$s_{1,1}$	3, 3	1, 1
	$s_{1,2}$	1, 1	2, 2

One can quickly check through Definition 1.5 that matrix entries (1,1) and (2,2) (i.e. strategy profiles  $(s_{1,1}, s_{2,1})$  and  $(s_{1,2}, s_{2,2})$ ) are pure Nash equilibria for that game. Notice, however, that these two equilibrium points *do not* yield the same utilities for the players. At the first one players receive a utility of 3 and at the second one a strictly less utility of 2.

This is a very important difference between dominant strategy and pure Nash equilibria. That means we can not treat the different pure Nash equilibria of a game as being equivalent (in the sense of indifference to the players) and so special care should be taken when using this solution concept to define further game properties. We should always have this kind of “classification” of our game’s pure Nash equilibria at the back of our mind, no matter how subtle (or difficult) this may be when trying to take these solution concepts as “black boxes”. Of course, the above phenomenon is a consequence of weakening the equi-

librium conditions and having a more “versatile” solution concept than that of dominant strategies.

At the opposite side of the possibility of a game to have more than one pure Nash equilibria, there is also the possibility of a game to have no pure Nash equilibria (and thus, also no dominant strategy equilibria) at all. Recall, for example, the Matching Pennies game at subsection 1.1.4. This is certainly an important disadvantage of the solution concepts we have presented so far and which, although being quite natural and strong, they surely cannot be applied to the general class of strategic game. The challenge here is to come up with the right relaxation technique in order to weaken our equilibrium conditions but also to ensure a nontrivial new solution concept.

### 1.3 Mixed Strategies and Nash Equilibria

In this section we proceed with relaxing the equilibrium conditions of our solution concepts even more, in order to get a rather flexible and general solution concepts. Towards doing that we deploy a favorite technique of Computer Science: *randomization*. Of course, when John Nash introduced the notion of mixed equilibria in 1950 he couldn’t possibly know of Computer Science as a well formed scientific discipline. However, looking today at the following definition we can clearly see a simple, yet clear and elegant randomization design technique.

**DEFINITION 1.6 (Mixed Strategies)** *Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a game. A mixed strategy  $\sigma_i$  of player  $i \in \mathcal{N}$  is a probability distribution over  $S_i$ , the set of player’s  $i$  pure strategies. In particular, if  $\mathcal{G}$  is finite and  $|S_i| = n_i \in \mathbb{N}$ , then a mixed strategy can be viewed as a tuple*

$$\sigma_i = (p_1, p_2, \dots, p_{n_i})$$

*where  $\sum_{j=1}^{n_i} p_j = 1$  and  $p_j \geq 0$  for all  $j = 1, 2, \dots, n_i$ .*

It is trivial to see that every pure strategy is also a mixed strategy, in particular  $s_{i,j} = (p_1, p_2, \dots, p_{n_i})$  where  $p_k = 0$  for all  $k \neq j$  and  $p_j = 1$ , i.e. player  $i$  plays its  $j$ -th pure strategy,  $s_{i,j}$  with certainty. We will denote player’s  $i$  set of all possible mixed strategies by  $\Sigma_i$ . Then, naturally enough, the set of all mixed strategy *profiles* (possible outcomes) of our game is  $\Sigma = \prod_{i=1}^N \Sigma_i$ .

Note also that in a mixed strategy setting, rationality implies that each player acts as to maximize his *expected* utility, the expectation taken over her pure strategies following her

mixed strategy distribution. In the case of finite games, this is given by the expression

$$\mathbb{E}_{s_1 \sim \sigma_1, s_2 \sim \sigma_2, \dots, s_N \sim \sigma_N} [u_i(s_1, s_2, \dots, s_N)] = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_N=1}^{n_N} p_{1,j_1} p_{2,j_2} \cdots p_{N,j_N} u_i(s_{1,j_1}, s_{2,j_2}, \dots, s_{N,j_N}). \quad (1.1)$$

We used, implicitly, a simplified form of this expression for two players and two strategies when we analyzed the Matching Pennies game at subsection 1.1.4. For the sake of readability we will slightly abuse notation and denote this expected utility simply by  $u_i(\sigma_1, \sigma_2, \dots, \sigma_N)$ , interpreting it naturally as the utility of the mixed strategy profile.

**DEFINITION 1.7 (Nash Equilibrium)** *Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a game. A mixed strategy profile  $(\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$  is a (mixed) Nash equilibrium for our game  $\mathcal{G}$  if, for every  $i \in \mathcal{N}$ ,  $\sigma_i^*$  is a best response mixed strategy of player  $i$  to the other players' mixed strategies  $\sigma_{-i}^*$ , i.e.*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*) \quad \text{for all } \sigma_i \in \Sigma_i.$$

Again, it is easy to compare Definition 1.7 to Definition 1.5 and see that every pure Nash equilibrium is also a mixed Nash equilibrium, leading to the following hierarchy of the sets of equilibrium points of a game:

$$\{\text{dominant strategy}\} \subseteq \{\text{pure Nash}\} \subseteq \{\text{mixed Nash}\}.$$

The most well-known result in Game Theory, and conceivably the reason for establishing the mixed Nash equilibrium as the predominant solution concept in the field, is the following theorem which ensures us that this solution concept is as general as one will hope to:

**THEOREM 1.8 (NASH, 1950)** *Every finite game (in normal form) has at least one (mixed) Nash equilibrium.*

**PROOF** A formal proof of this result is out of the scope of this thesis, however we believe that its detailed study is of utmost importance<sup>14</sup> to everyone who takes the study of Game Theory seriously. It is an existential, *non-constructive* proof based on fixed-point theorems. Both Brouwer's and its generalization, Kakutani's fixed-point theorem (see [Mas-Colell et al., 1995, p. 952–953]) can be used to arrive to the result<sup>15</sup>. More

<sup>14</sup>In the words of Fudenberg and Tirole [1991, p. 29]:“... this is the archetypal existence proof in game theory ...”

<sup>15</sup>And actually Nash himself used both of them, publishing two versions of the proof in two papers. For more details see the Notes section 1.5.

information about Nash's theorem and its proof are given to the Notes section 1.5 of this chapter.  $\square$

## 1.4 Characterizing Nash Equilibria

When presenting the various solution concepts for our games, we didn't bother at all with *how* a system of players is stabilized at such an equilibrium. We are interested in the final stable state and we don't want to describe possible procedures through which those equilibria are reached nor to justify the arrival at such states. Although this is a common approach in Economics (as a social science), we would like to have an algorithm for finding mixed Nash equilibria in a strategic game (from Nash's Theorem 1.8 we are sure that such an equilibrium always exists). In addition, since computational complexity is a major issue in Computer Science, we would hope for an *efficient* algorithm to do that, not only for its theoretical importance but also for the practical applications of Algorithmic Game Theory in electronic markets, auctions, communication networks, routing and scheduling problems, etc<sup>16</sup>. However, as explained in the Notes section 1.5, all evidence is against the existence of such a computationally efficient algorithm and in this section we provide some characterizations for equilibria points which, in some occasions and especially when our games are "small" (e.g. two-player, two-strategy games), can help us locate Nash equilibria.

**PROPOSITION 1.9** *Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a finite game,  $|S_i| = n_i$ . A mixed strategy profile  $(\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$ , where  $\sigma_i^* = (p_{i,1}, p_{i,2}, \dots, p_{i,n_i})$ , is a Nash equilibrium if and only if, for every player  $i$ ,*

$$p_{i,j} \neq 0 \implies u_i(s_{i,j}, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*) \quad \text{for all } \sigma_i \in \Sigma_i$$

*i.e. pure strategy  $s_{i,j}$  is also a best response mixed strategy to  $\sigma_{-i}^*$ , for all  $j = 1, 2, \dots, n_i$ .*

**PROOF**

( $\implies$ ) Let  $\sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$  be a Nash equilibrium for  $\mathcal{G}$ . Fix some player  $i$  and some pure strategy  $s_{i,j} \in S_i$  for which  $p_{i,j} \neq 0$ . We can also restrict our attention to the case of  $p_{i,j} < 1$  since, if  $p_{i,j} = 1$  then mixed strategy  $\sigma_i^*$  "collapses" to the pure strategy  $s_{i,j}$  and thus, trivially  $u_i(s_{i,j}, \sigma_{-i}^*) = u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*)$  for all possible  $\sigma_i \in \Sigma_i$ , since  $(\sigma_i^*, \sigma_{-i}^*)$  is a Nash equilibrium (recall Definition 1.7). To arrive to a contradiction, assume that  $s_{i,j}$  is *not* a best response strategy to  $\sigma_{-i}^*$ , that is, there exists a mixed strategy

<sup>16</sup>For a better taste of the numerous applications of Algorithmic Game Theory to economic, social, computational and engineering problems the reader is referred to the standard, in that area, reference of Nisan et al. [2007].

$\sigma_i$  for player  $i$  such that  $u_i(s_{i,j}, \sigma_{-i}^*) < u_i(\sigma_i, \sigma_{-i}^*)$ . But  $(\sigma_i^*, \sigma_{-i}^*)$  is a Nash equilibrium, thus

$$u_i(s_{i,j}, \sigma_{-i}^*) < u_i(\sigma_i^*, \sigma_{-i}^*). \quad (1.2)$$

We know, based on (1.1), that

$$u_i(\sigma_i^*, \sigma_{-i}^*) = p_{i,1}u_i(s_{i,1}, \sigma_{-i}^*) + p_{i,2}u_i(s_{i,2}, \sigma_{-i}^*) + \cdots + p_{i,n_i}u_i(s_{i,n_i}, \sigma_{-i}^*) \quad (1.3)$$

so inequality (1.2) gives

$$u_i(s_{i,j}, \sigma_{-i}^*) < p_{i,1}u_i(s_{i,1}, \sigma_{-i}^*) + p_{i,2}u_i(s_{i,2}, \sigma_{-i}^*) + \cdots + p_{i,n_i}u_i(s_{i,n_i}, \sigma_{-i}^*)$$

or, equivalently,

$$u_i(s_{i,j}, \sigma_{-i}^*) < \frac{1}{1 - p_{i,j}} \sum_{\substack{k=1 \\ k \neq j}}^{n_i} p_{i,k} u_i(s_{i,k}, \sigma_{-i}^*).$$

So, from (1.3) we have:

$$\begin{aligned} u_i(\sigma_i^*, \sigma_{-i}^*) &= p_{i,j}u_i(s_{i,j}, \sigma_{-i}^*) + \sum_{\substack{k=1 \\ k \neq j}}^{n_i} p_{i,k} u_i(s_{i,k}, \sigma_{-i}^*) \\ &< \frac{p_{i,j}}{1 - p_{i,j}} \sum_{\substack{k=1 \\ k \neq j}}^{n_i} p_{i,k} u_i(s_{i,k}, \sigma_{-i}^*) + \sum_{\substack{k=1 \\ k \neq j}}^{n_i} p_{i,k} u_i(s_{i,k}, \sigma_{-i}^*) \\ &= \sum_{\substack{k=1 \\ k \neq j}}^{n_i} p_{i,k} \left( \frac{p_{i,j}}{1 - p_{i,j}} + 1 \right) u_i(s_{i,k}, \sigma_{-i}^*) \\ &= \sum_{\substack{k=1 \\ k \neq j}}^{n_i} \frac{p_{i,k}}{1 - p_{i,j}} u_i(s_{i,k}, \sigma_{-i}^*). \end{aligned} \quad (1.4)$$

Recall that  $\sigma_i^* = (p_{i,1}, p_{i,2}, \dots, p_{i,n_i})$  is a mixed strategy, so we know that  $\sum_{k=1}^{n_i} p_{i,k} = 1$ . Also  $p_{i,j} \neq 0$ . Using these it is easy to verify that both following conditions hold:

$$\sum_{\substack{k=1 \\ k \neq j}}^{n_i} \frac{p_{i,k}}{1 - p_{i,j}} = 1 \quad \text{and} \quad 0 \leq \frac{p_{i,k}}{1 - p_{i,j}} \leq 1.$$

That means that  $\sigma_i = (q_1, q_2, \dots, q_{n_i})$  with  $q_k = \frac{p_{i,k}}{1 - p_{i,j}}$  for  $k \neq j$  and  $q_j = 0$ , is a valid

mixed strategy for player  $i$  and

$$\begin{aligned}
 u_i(\sigma_i, \sigma_{-i}^*) &= \sum_{k=1}^{n_i} q_k u_i(s_{i,k}, \sigma_{-i}^*) \\
 &= \sum_{\substack{k=1 \\ k \neq j}}^{n_i} \frac{p_{i,k}}{1 - p_{i,j}} u_i(s_{i,k}, \sigma_{-i}^*) \\
 &> u_i(\sigma_i^*, \sigma_{-i}^*), \qquad \text{from (1.4),}
 \end{aligned}$$

which contradicts the fact that  $(\sigma_i^*, \sigma_{-i}^*)$  is a Nash equilibrium.

( $\Leftarrow$ ) For the opposite direction, fix some player  $i$  and let  $\sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$  be a mixed strategy profile for which

$$p_{i,j} \neq 0 \implies u_i(s_{i,j}, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*) \quad \text{for all } \sigma_i \in \Sigma_i \quad (1.5)$$

for every  $j = 1, 2, \dots, n_i$ . Then, for every possible mixed strategy  $\sigma_i$  for player  $i$ ,

$$\begin{aligned}
 u_i(\sigma_i^*, \sigma_{-i}^*) &= \sum_{j=1}^{n_i} p_{i,j} u_i(s_{i,j}, \sigma_{-i}^*) = \sum_{\substack{j=1 \\ p_{i,j} \neq 0}}^{n_i} p_{i,j} u_i(s_{i,j}, \sigma_{-i}^*) \\
 &\geq \sum_{\substack{j=1 \\ p_{i,j} \neq 0}}^{n_i} p_{i,j} u_i(\sigma_i, \sigma_{-i}^*), \qquad \text{from (1.5),} \\
 &= u_i(\sigma_i, \sigma_{-i}^*) \cdot \sum_{\substack{j=1 \\ p_{i,j} \neq 0}}^{n_i} p_{i,j} = u_i(\sigma_i, \sigma_{-i}^*) \cdot 1 = u_i(\sigma_i, \sigma_{-i}^*). \quad \square
 \end{aligned}$$

The intuition behind the proof is that, if some pure strategy was suboptimal then we could eliminate it (assigning zero probability) and increase the weights of the other strategies in order to arrive to a new mixed strategy with strictly greater utility. Here, the linearity of expression (1.1) plays a major role.

An equivalent formulation of Proposition 1.9 is that of the following corollary, which is somehow more straightforward as far as computations for finding a Nash equilibrium are concerned. In words, it says that  $\sigma^*$  is a Nash equilibrium if and only if every player is indifferent among the pure strategies comprising the support of its mixed strategy  $\sigma_i^*$ . Formally:

**COROLLARY 1.10** *Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a finite game and  $\sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$  a mixed strategy profile, where  $S'_i \subseteq S_i$  is the support of  $\sigma_i^*$ . Then  $\sigma^*$  is a Nash equilibrium*

if and only if, for every player  $i$  there is an  $\alpha_i \in \mathbb{R}$  such that

$$u_i(s_{i,j}, \sigma_{-i}^*) = \alpha_i \quad \text{for all } s_{i,j} \in S'_i$$

and  $u_i(s_{i,j}, \sigma_{-i}^*) \leq \alpha_i$  for all  $s_{i,j} \in S_i \setminus S'_i$ .

Usually the following sufficient (but not necessary) condition is a better choice, when we want to “quickly” check for a Nash equilibrium in “small” games. We essentially used this at subsection 1.1.4 to find/justify  $(\frac{1}{2}, \frac{1}{2})$  being a Nash equilibrium for the Matching Pennies game:

**COROLLARY 1.11** *Let  $\mathcal{G} = (\mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a finite game and  $\sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$  a mixed strategy profile such that for every player  $i$  there is an  $\alpha_i \in \mathbb{R}$  with*

$$u_i(s_{i,j}, \sigma_{-i}^*) = \alpha_i \quad \text{for all } s_{i,j} \in S_i.$$

*Then  $\sigma^*$  is a Nash equilibrium.*

Notice that the condition in Corollary 1.11 generates a set of  $\frac{|S_i| \cdot (|S_i| - 1)}{2}$  equations of utility indifference for every player.

Finally, we must point out that in no way we claim that the characterizations presented in this section are the most efficient way to find a Nash equilibrium, in fact they are far away from achieving this goal and their importance lies more on their elegant characterization conditions. In practice, more efficient procedures (such as the pivotal Lemke-Howson algorithm) are used, even in the simplest case of bimatrix (two-player) games (see von Stengel [2007]). For more on these computational issues see the Notes section 1.5.

## 1.5 Notes

Our exposition in this chapter is based on several standard reference textbooks in Game Theory, namely [Fudenberg and Tirole, 1991], [Mas-Colell et al., 1995, part 2] and [Osborne and Rubinstein, 1994]. Here, we only gave a small glimpse of the beautiful and intriguing field of Game Theory, based only on what we are going to need to properly present the remaining material of this thesis. We recommend the first two of the above books for a further study of the subject, as well as [Tardos and Vazirani, 2007] for a well-balanced, Computer Science oriented introduction. For a more lighter, yet complete treatment, we refer to the undergraduate textbook of Osborne [2004].



Nash's Theorem 1.8 was first presented and proved (obviously by Nash...) in [Nash, 1950], using Kakutani's fixed point theorem. A second, somehow more elegant proof was given one year later again by Nash in his landmark paper [Nash, 1951], this time using Brouwer's fixed point theorem. The proof there is elegant, coherent and rather fundamental and, although it can nowadays be found in every (serious) graduate textbook in Game Theory or Microeconomic Theory, we recommend studying it through the original paper of Nash [1951]. It is a real pleasure. This paper also laid the foundations of what is known today as *noncooperative* Game Theory. The exposition is clear and powerful and thus that paper is, once more, highly recommended. The origins of modern Game Theory go back to the seminal work of von Neumann and Morgenstern [1953] where an extensive study of zero-sum games was carried out, as well as a formulation of what is known today as *cooperative* Game Theory.

The dependence of Nash's proof on Brouwer's fixed point theorem (ironically enough<sup>17</sup>) forces the proof to have a strictly non-constructive character and, although during the years proofs of Brouwer's fixed point theorem with some constructive nature have been discovered, these do not yield computationally efficient algorithms. There are strong indications that finding a fixed point (in Brouwer's theorem) is an intractable problem. This computational hardness is also inherited to the problem of finding a mixed Nash equilibrium of a strategic game. Notice, however, that this intractability is not meant in the sense of NP-completeness, simply because these problems are not *decision* (yes-no) problems. We know that every game does have a Nash equilibrium (trivially from Nash's theorem), the question is *how* to find one efficiently (preferable in polynomial time). It turns out that the right complexity class to study these problems is PPAD, introduced in [Papadimitriou, 1994]. Complexity issues play a leading role in the field of Algorithmic Game Theory, but we have already gone well beyond the scope of our thesis towards that direction. For a rather complete introduction to this area we recommend [Papadimitriou, 2007], as well as the very important paper of Daskalakis, Goldberg, and Papadimitriou [2006].

---

<sup>17</sup>L. E. J. Brouwer (1881–1966) was a *constructivist* mathematician, probably the most notable representative of intuitionistic mathematics. However, his fixed point theorem we cite here, one of his most famous results, utilizes a non-constructive proof in the most absolute way (e.g. see [Hatcher, 2002, Corollary 2.15]).



# Chapter 2

## Mechanism Design

In the previous Chapter 1 we proposed some basic notions from Game Theory as a framework to study a (rather primitive, though powerful) form of strategic interaction between rational players. This framework has two main characteristics which are also its limitations:

1. It is totally **passive**. Players simply “enter” the game and we rely on the “wisdom of economics”<sup>1</sup> that the game will result in a stable state at which the combined choices made by all players serve one and only goal: for each player to **selfishly** maximize her (expected) utility. Such an environment, leaves no space for us (an external “designer”) to have any control whatsoever on the “execution” or the desired stable state reached by the game.
2. We have assumed that players have **full information** of the game’s elements (see Definition 1.1, page 9). In particular, every player’s available choices and utility of every possible game outcome (strategy profile), are common knowledge among all players. Such an assumption, of course, is far from realistic even when we try to model some of the most fundamental examples of strategic behaviour such as auctions, voting systems and electronic markets.

However, these basic notions from Game Theory carry the essence of the behavior of strategic players and thus can be used as important elements in an extended framework which can model more realistically the problems that interest us in this thesis (mainly auctions) by overcoming the above limitations. In general, we would like to be able to model **decision-making** (under uncertainty) problems in which an outcome must be chosen by the decision-maker, given information he receives from a set of participants. The crucial

---

<sup>1</sup>Or God...

point here, which is why we use Game Theory as a basis to study such decision problems, is that we assume that our participants are strategic, selfish players and thus, if this is to maximize their utility, they can (without any compunction) lie about the true information they hold and upon which our decisions must be based.

In the following section we bring down piece by piece the components of the above intuitive description, introducing rigorously the appropriate notions.

## 2.1 Social Choices and Mechanisms

Every player may have various preferences over the set of all possible outcomes (decisions). We model this by assuming that each player has a *valuation function* over the set of outcomes, the value of which on a given outcome quantifies<sup>2</sup> how much our player “values” the specific outcome. The higher the value, the more preferred the outcome.

Before getting into the main goal of this chapter, i.e. how to assure that the “right” decision is always made even if we have to rely on information provided by a set of strategic players, it is essential to fix beforehand what this “right” decision would be. In general, this will depend on the preferences of the players and will choose an outcome based on what “social goal” we need to serve, e.g. in a voting system we usually want to choose the candidate which maximizes the “combined social satisfaction”, i.e. receives the most votes.

**DEFINITION 2.1 (Social choice function)** *Let  $\mathcal{N} = \{1, 2, \dots, N\}$  be a finite set of players<sup>3</sup>, each of whom has a set  $V_i$  of possible valuation functions  $v_i : \mathcal{O} \rightarrow \mathbb{R}$  over a set of outcomes<sup>4</sup>  $\mathcal{O}$ . A social choice function is a function  $f : \prod_{i=1}^N V_i \rightarrow \mathcal{O}$ .*

A social choice function is a rule that aggregates the preferences of the different players (expressed through their valuation functions) to a single outcome.

After fixing a specific social choice function  $f : \prod_{i=1}^N V_i \rightarrow \mathcal{O}$  it may seem trivial what one should do: simply request from the players to state their preferences by reporting their

---

<sup>2</sup>Alternatively, we could have assumed that every player has a *total ordering* (see [Cormen et al., 2001, p. 1077]) over the set of possible outcomes, which is a more “natural” choice if one is considering settings such as voting systems. However, we choose to numerically quantify preferences which is, trivially, a generalization of the total ordering case. Not only we can tell if a player prefers one outcome from another, but also in what extent, i.e. by “how much”, she does so. This opens the way of using “money” as a common measure. Under the light of the famous Arrow and Gibbard–Satterthwaite theorems, such a choice is essentially necessary. We have already gone beyond the scope of this thesis with respect to the current discussion and we refer to [Nisan, 2007, 9.2-9.3]. Arrow’s theorem (from which Gibbard–Satterthwaite theorem can be deduced as a corollary) is a most celebrated result of Social Choice Theory (and Economics, in general) with surprising implications. The reader will certainly benefit from getting in touch with the original work of Arrow [1951]. Some less involved proofs of this classic result can be found in [Geanakoplos, 2005]. The Gibbard–Satterthwaite are, naturally enough, due to Gibbard [1973] and Satterthwaite [1975].

<sup>3</sup>Also called *agents*, especially in some computational settings.

<sup>4</sup>Also called *alternatives*, especially in the context of classic (microeconomic) Social Choice Theory.

valuation functions and then calculate the value of  $f$ . This value is the decision we must make. However, this is far from being the case due to some major implications that have to do with the decision-making model which we have adopted. First, players do not, in general, have to report their valuations but some other kind of information-message to the decision-maker. And, above all, even if their messages are restricted to reports of their valuations, there is no guarantee that they will actually report truthfully to us. Remember that they are selfishly rational and they only care about maximizing their utility rather than helping us make the right decision. Such misreports may have devastating consequences, because they can lead us to compute a value of  $f$  that is far from desired with respect to our fixed “social goal” that  $f$  implements.

All these make necessary the extension of the notion of full information (strategic) games we introduced in Definition 1.1, page 9:

**DEFINITION 2.2 (Strict incomplete information)**

A game  $\Gamma = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  of strict incomplete information consists of:

- a finite set of players  $\mathcal{N}$ ,  $\mathcal{N} = \{1, 2, \dots, N\}$
- for every player  $i \in \mathcal{N}$ , a set of types  $\Theta_i$
- for every player  $i \in \mathcal{N}$ , a set of actions  $A_i$  and,
- for every player  $i$ , a utility function  $u_i : \Theta_i \times \prod_{i=1}^N A_i \longrightarrow \mathbb{R}$ .

The set of strategies of player  $i$  in this incomplete information setting is  $S_i = A_i^{\Theta_i}$ , i.e. a strategy of player  $i$  is a function  $s_i : \Theta_i \longrightarrow A_i$ .

Notice that the strategy sets  $S_i$  need not to be in the description of  $\Gamma$  (as they were in Definition 1.1) since they are completely determined by  $A_i$  and  $\Theta_i$ . In our usual notation,  $\Theta = \prod_{i=1}^N \Theta_i$  will denote the set of all possible *type profiles* and  $A = \prod_{i=1}^N A_i$  the set of *action profiles*.

Type  $\theta_i$  represents any **private information** player  $i$  has. This is only known to player  $i$ , in contrast to the standard, full-information case of Chapter 1. This may include, for example, her utility function  $u_i$ . In classic (economic) Game Theory, we usually suppose that player  $i$  has some (reliable) distributional assumptions about other players’ types (private information), included in her private type  $\theta_i$ . This models are known as *Bayesian games*<sup>5</sup>. Here, as one can easily verify, we have not made any assumptions at all regarding probabilistic information included in the types of Definition 2.2. We make such a

<sup>5</sup>More on Bayesian games can be found in the references proposed in the Notes section 2.4 of our chapter.

choice because we want to study the main problems of this thesis (presented later on in Part C) under a worst-case analysis perspective, standard in Computer Science. We call our model game of *strict* incomplete information<sup>6</sup> to point out the lack of further assumptions. Furthermore, although games of strict incomplete information are generalizations of Bayesian games, the fundamental results in classic Mechanism Design continue to hold in our “strict” framework and sometimes, especially to computer scientist, are much more easy and natural to prove and interpret.

The intuition behind the Definition 2.2 is the following: Initially, each player  $i$  chooses what action  $a_i = s_i(\theta_i) \in A_i$  she will take if her type turns out to be  $\theta_i \in \Theta_i$ , taking into consideration the utility  $u_i(\theta_i, a)$  that each possible type  $\theta_i$ -action profile  $a = (a_1, a_2, \dots, a_N)$  combination will give her. Remember that players are rational and selfish and their only goal is to maximize their expected utility. The crucial detail here, which is the essence of the incomplete information case, is that every player  $i$  does not, in general, have any knowledge at all<sup>7</sup> about the other players’ utility functions and types, thus she can make no reliable predictions of what other players are going to play based on the rationality principle (as was the case in Chapter 1). Next, agent  $i$  **realizes** her type  $\theta_i \in \Theta_i$ , in a way beyond her control. It is useful to think of “nature” determining the “correct” assignment of types to every player and their nothing they can do about it. Obviously, after the realization of all players’ types, the action profile is completely determined by the strategy functions already played by our players.

We must point out here that such an interpretation of incomplete games is strongly intuitive and in no way implies that such a two-step, well defined procedure takes place. In the contrary, as we have argued before in Chapter 1, we do not bother at all with the procedures through which game outcomes and stable states are reached. Instead we view it as a complex phenomenon whom results can only observe and justify.

After a type is fixed for every player, the “layer” of incomplete information can be removed from the general Definition 2.2, inducing a full information (strategic) game where the strategies are essentially the actions available to our players:

**DEFINITION 2.3** *Let  $\Gamma = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a game of strict incomplete information. Fix a type profile  $\theta = (\theta_1, \theta_2, \dots, \theta_N) \in \Theta$ . The full information game defined by  $\Gamma$  and  $t$  is the strategic game  $\Gamma(\theta) = (\mathcal{N}, \{S'_i\}_{i \in \mathcal{N}}, \{u'_i\}_{i \in \mathcal{N}})$  for which*

- $S'_i = A_i$  and
- $u'_i(s'_1, s'_2, \dots, s'_N) = u_i(\theta_i, s'_1, s'_2, \dots, s'_N)$ ,

<sup>6</sup>This is not a standard term in the field and one may also stumble upon the term *pre-Bayesian*.

<sup>7</sup>strict incomplete, bayesian

for all players  $i \in \mathcal{N}$  and strategy profiles  $(s'_1, s'_2, \dots, s'_N) \in \prod_{i=1}^N S'_i (= \prod_{i=1}^N A_i)$ .

Next, we go on by introducing the basic solution concept we are going to use in our incomplete information setting, that of dominant strategy equilibrium. It is a natural adaptation of the dominant strategy equilibrium used in the full information case of Chapter 1 (see Definition 1.4, page 11). However, in order to “conserve” the strength and stability of the classic dominant strategies notion, we demand *every* possible realization of types that may occur to result in a (full information) game possessing the desired dominant strategy equilibrium. This is, as expected, an extremely strong (and thus possibly restricting) solution concept but, once achieved, it serves as the best framework to perform our worst-case analysis.

**DEFINITION 2.4 (Incomplete information dominant strategy equilibrium)** Let  $\Gamma = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  be a strict incomplete information game. A strategy profile  $(s_1^*, s_2^*, \dots, s_N^*)$  is a dominant strategy equilibrium for our game  $\Gamma$  if, for every type profile  $\theta = (\theta_1, \theta_2, \dots, \theta_N)$ ,  $(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N))$  is a dominant strategy equilibrium for the full information game  $\Gamma(t)$ , i.e. for every  $i \in \mathcal{N}$ ,

$$u_i(\theta_i, s_i^*(\theta_i), a_{-i}) \geq u_i(\theta_i, a_i, a_{-i}) \quad \text{for all } a_i \in A_i, a_{-i} \in A_{-i}.$$

It is now time to use all the power that Game Theory gives us, through incomplete information games, to model properly the decision-making environment we use to work in.

**DEFINITION 2.5 (Mechanism environment)** A mechanism environment  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  consists of:

- a finite set of players  $\mathcal{N}$ ,  $\mathcal{N} = \{1, 2, \dots, N\}$
- for every player  $i \in \mathcal{N}$ , a set of types  $\Theta_i$
- for every player  $i \in \mathcal{N}$ , a set of actions  $A_i$
- a set of outcomes  $\mathcal{O}$  and,
- for every player  $i$ , a valuation function  $v_i : \Theta_i \times \mathcal{O} \rightarrow \mathbb{R}$ .

Note that, in giving the above Definition 2.5 we are strongly motivated by Definition 2.1 and the idea of applying some desired *social choice* in our decision-making. The use of a valuation function is a strong evident of that and, although the Social Choice background

of Definition 2.5 may not be so obvious now, it will become apparent at Definition 2.8, page 27, that follows.

Although in the Social Choice framework of Definition 2.1 we have only valuations as the predominant form of information carried by the players, we can extend it slightly to match our incomplete information terminology. We can assume that every player's valuation can be parametrized by its type  $\theta_i$ , therefore viewing social choice functions as  $f : \prod_{i=1}^N \Theta_i \rightarrow \mathcal{O}$ . This is consistent with all our exposition in this chapter and also helps us to see social choice functions as immediate “decision-makers” taking into consideration the players' types.

So, as we stated in the introduction of our Chapter 2 and more solidly at the beginning of our section 2.1 at page 22, our ultimate goal is to be able to reliably make some “socially desirable” decision, implied by a social choice function<sup>8</sup>, in an environment of incomplete information and, more dangerously, of selfish players that are ready to lie to us about their true private information  $\theta_i$ . But selfishness and rationality, are fundamental assumptions within Game Theory and the very reason for which we chose this framework to study our decision-making problems. Thus relaxing them is out of the question. Therefore, the only way through which we may try to “convince” the players to tell us the truth (or, more generally, to give us enough information in order to achieve (or estimate) the desired social goal) is by giving them the right **incentives** in order to “satisfy” their selfishness. Incentives do not need to be only “positive”, i.e. some kind of bonus, but they may very well be negative, i.e. “penalties” subtracted from their initial utility. In such a setting, every strategic player tries to, loosely speaking, minimize the “damage” caused to her by the penalties enforced to her. Since in this thesis we always have the example of auctions in the back of our minds, this is the form of incentives we are going to use: negative ones (primarily in the form of payments, collected from the players). Naturally enough, it is easy to reverse the situation: negative payments correspond to “bonuses”.

All matters regarding “truthfulness” are going to become clear and rigorously defined in the next section 2.2, but for the time being we need to formalize the discussion of the previous paragraph. We introduce the fundamental notion of a *mechanism*, which is essentially the procedure through which we try to give incentives in order to achieve our desired social goal. So, our mechanisms need to comprise of two major elements: the decision they take (corresponding to some social choice) and the (negative) incentives they use, here called *payments*, in order to ensure that players will report as we (the *mechanism*

---

<sup>8</sup>Although we often use terms like “socially desirable” and “social goal”, social choice functions do not need to always compute social “justice” or social “welfare”. For example, in an auction setting we might only be interested in choosing the outcome that maximizes the auctioneer's revenue, not caring about the bidders' “satisfaction”. However, here we choose to use the established terms in the field.



designers – decision-makers) desire:

**DEFINITION 2.6 (Mechanism)** *A mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  over a mechanism environment  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  consists of:*

- a decision function  $x : \prod_{i=1}^N A_i \longrightarrow \mathcal{O}$  and,
- for every player  $i$ , a payment function  $p_i : \prod_{i=1}^N A_i \longrightarrow \mathbb{R}$ .

The vector valued function  $p = (p_1, p_2, \dots, p_N) : A \longrightarrow \mathbb{R}^N$  is usually called *payment vector*. As we have assumed throughout this thesis that we have some way to numerically quantify the preferences of the players by their valuations, we will also assume from now on that, the payments–penalties just introduced are also quantified in a similar way and furthermore that one can subtract payments from valuations to get the quantified total utility that a player receives if we force a penalty to him in order to reduce the initial value a decision has for her. More concisely, we assume that we have a common numerical quantification system between valuations, payments and utilities (see the discussion about “money” at footnote 2 on page 22.)

Of course all these are done in a strategic interaction setting and it’s time to deploy our valuable tools from Game Theory.

**DEFINITION 2.7 (Games of mechanisms)** *The game  $\Gamma(\mathcal{M})$  induced by a mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  over some mechanism environment  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  is the strict incomplete information game  $\Gamma(\mathcal{M}) = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$  where*

$$u_i(\theta_i, a_1, a_2, \dots, a_N) = v_i(\theta_i, x(a_1, a_2, \dots, a_N)) - p_i(a_1, a_2, \dots, a_N),$$

for every  $\theta_i \in \Theta_i, (a_1, a_2, \dots, a_N) \in A$ .

After all this discussion, and having some examples (presumably auctions) in our minds, one may have already felt intuitively the idea behind how a mechanism can simulate a given social choice. However, we must state formally and clearly what qualifications we will demand our mechanism to meet before declaring it acceptable to reliably implement our desired social goal. What better choice exists other than our strong solution concept of dominant strategies?

**DEFINITION 2.8 (Implementation)** *Fix some mechanism environment  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ . We say that a mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  implements (in dominant strategies) the social function  $f : \prod_{i=1}^N \Theta_i \longrightarrow \mathcal{O}$  if the induced*

game  $\Gamma(\mathcal{M})$  has a dominant strategy equilibrium  $(s_1^*, s_2^*, \dots, s_N^*)$  for which

$$f(\theta_1, \theta_2, \dots, \theta_N) = x(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N)) \quad \text{for all } (\theta_1, \theta_2, \dots, \theta_N) \in \Theta.$$

## 2.2 Direct Revelation Mechanisms and Truthfulness

Up to now we have tried to keep our mechanism design model as general as possible. However, the main problems that are going to bother us in this thesis have to do with auction settings. Consider, for example, the following simple auction environment, known as *sealed-bid auction*. We have a single item to sell to one of many bidders. Each bidder writes her bid in a sealed envelope and submits it to us. Next, we open all envelopes and must decide whom to allocate the item to and what payment to collect from her. In such a setting, the private information of each player is her bid (kept secret from all others) and the messages that she can send to the auctioneer is exactly a single report of her bid. Of course, our players are strategic, thus they may very well lie and *misreport* their true bid. Such mechanisms in which a player's action is simply a single, direct claim about her true type are called *direct revelation mechanisms* and are much more simple, natural and easy to interpret and are sufficient to model some of the most important problems in Algorithmic Mechanism Design.

### DEFINITION 2.9 (Direct revelation mechanism environment)

A *direct revelation mechanism environment*  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  is a mechanism environment  $(\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  for which  $A_i = \Theta_i$  for every agent  $i \in \mathcal{N}$ .

If  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  is an (arbitrary) mechanism environment, we will denote by  $\mathcal{E}_d$  the direct revelation “restriction” of  $\mathcal{E}$ , i.e.  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ .

**DEFINITION 2.10 (Direct revelation mechanisms)** A *direct revelation mechanism* is a mechanism over a direct revelation mechanism environment.

Let us return to our sealed-bid example. It is natural to assume that the *true* bid of every player is a measure of how much she values the acquisition of the item<sup>9</sup>. Let us fix

<sup>9</sup>Here we will not dwell on the subject of “social inequalities”. In real life auctions, one may desire an item more than somebody else but not be able to compete with the second player's bid. This is the result, loosely speaking, of a monetary unit having different value between the two players. However, it is extremely difficult and out of the scope of Mechanism Design to try and “weight out” such social inconsistencies. Instead, we assume that our “money” is used as a fair valuation measurement and not as an indication of wealth. It is unnecessary to mention that believing in the existence of such an ideal society is “very optimistic”, at least.

a social objective. Through out Mechanism Design we usually adopt a maximization of social welfare<sup>10</sup> objective. In our simple one item sealed-bid auction, the item will end up on a single bidder, thus the social choice that we wish to implement here is the one that gives the item to the bidder that “needs” it more, i.e. submits the higher bid. Don’t forget, however, that our players act selfishly and they may be lying. By trusting their submitted bids we may end up giving the item to the wrong bidder.

So, it is essential to design a mechanism that gives no incentive to agents to lie. Let us consider the simplest, obvious mechanism that allocates the item to the highest bidding player and collects a payment that equals her bid. Suppose our players (surprisingly enough) have reported truthfully and that the winning player’s bid is  $p$  and  $q < p$  is the second highest submitted bid. The player that wins must pay an amount which equals her valuation, resulting to a total utility of  $p - p = 0$  for her (see Definition 2.7, page 27). But she could have lied, reporting a bid  $p'$  with  $q < p' < p$ , still getting the item and paying only  $p'$ , resulting to a total utility of  $p - p' > 0$ .

The above discussion shows as that the obvious, *first-price* auction cannot guarantee that players will report truthfully. So, what can we try next? The idea is very simple, though extremely brilliant and effective: let’s consider *second-price* auctions, i.e. auctions that still give the item to the highest bidding player but request a payment equal to the *second* highest submitted bid. It is trivial to check that by lying an bidder either loses the item or is requested to pay an amount strictly greater than her true value, resulting to a negative utility. Therefore, under this auction there is no way for a bidder to (strictly) improve her utility by lying. The crucial point here is that the winning player’s payment is independent of her bid and thus she cannot manipulate it. This second-price auction is known as the **Vickrey auction** due to Vickrey [1961] who formalized this brilliant idea of second-price payments.

By now, it is made clear that securing truthfulness for our mechanism is of major importance to the mechanism designer. Knowing that the players report truthfully, we can implement our social choices without worrying of dangerous manipulations of our protocols. It is time to put down formally the notion of truthfulness:

**DEFINITION 2.11 (Truthfulness)** *A direct revelation mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  (over some environment  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ ) is called dominant strategy incentive compatible (DSIC) or truthful<sup>11</sup> if, for every possible type profile  $(\theta_1, \theta_2, \dots, \theta_N)$ , the identity function over  $\Theta_i$  is a dominant strategy for every player  $i$  (on the induced game*

---

<sup>10</sup>Although this is not made clear yet, we rely on the reader’s intuition to interpret it as the decision that maximizes the combined “happiness” of our players. Formal definitions we soon follow at section 2.3.

<sup>11</sup>The term *strategy-proof* is also used.

$\Gamma(\mathcal{M})$ , i.e.

$$u_i(\theta_i, \theta_i, \theta'_{-i}) \geq u_i(\theta_i, \theta'_i, \theta'_{-i}) \quad \text{for all } \theta'_i \in \Theta_i, \theta'_{-i} \in \Theta_{-i}.$$

In words, this describes exactly what we would expect: no player can improve her utility by lying. Put it otherwise, every player is better off telling the truth.

The most “beautiful” property of truthful mechanisms is that, not only they guarantee simple and reliable implementation of our social choice function, but it also turns out that they are as expressive as arbitrary (non-truthful and non-direct revelation<sup>12</sup>) mechanism. This is surprisingly pleasant since one would expect that an arbitrary mechanism which allows for a much richer feedback (message space) and does not apply such strong restrictions as that of truthfulness (in dominant strategies), would also allow us to do much more things. All this collapses by the following famous

**THEOREM 2.12 (REVELATION PRINCIPLE)** *Let  $\mathcal{M}$  be a mechanism, over some mechanism environment  $\mathcal{E}$ , that implements a social choice function  $f$ . Then, there is a truthful (direct revelation) mechanism  $\mathcal{M}_d$ , over  $\mathcal{E}_d$ , that can implement  $f$ . In addition, at the (dominant strategy) equilibria implementing  $f$ , the payment rules of  $\mathcal{M}'$  and  $\mathcal{M}$  are identical.*

**PROOF** Let  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$ ,  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  and the induced game  $\Gamma(\mathcal{M}) = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{A_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}})$ . Mechanism  $\mathcal{M}$  implements  $f$ , so (Definition 2.8) there exists a dominant strategy equilibrium  $(s_1^*, s_2^*, \dots, s_N^*)$  of  $\Gamma(\mathcal{M})$  such that

$$f(\theta_1, \theta_2, \dots, \theta_N) = x(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N)) \quad (2.1)$$

for all possible type profiles  $(\theta_1, \theta_2, \dots, \theta_N) \in \Theta$ . Define a direct revelation mechanism  $\mathcal{M}_d = (x', \{p'_i\}_{i \in \mathcal{N}})$  over  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$  with decision and payment rules

$$x'(\theta_1, \theta_2, \dots, \theta_N) = x(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N)) \quad (2.2)$$

$$p'_i(\theta_1, \theta_2, \dots, \theta_N) = p_i(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N)) \quad \text{for every player } i \in \mathcal{N}. \quad (2.3)$$

We need to show that reporting truthfully is a dominant strategy for every player at the induced game  $\Gamma(\mathcal{M}_d) = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \{\Theta_i\}_{i \in \mathcal{N}}, \{u'_i\}_{i \in \mathcal{N}})$ , i.e. show that  $\mathcal{M}$  is truthful. We will prove that the condition in Definition 2.11 holds. Indeed, fix a player

<sup>12</sup>Recall from Definition 2.11 that truthfulness is defined only for direct revelation mechanisms.

$i$  with (true) type  $\theta_i$ . First notice that, because of the way in which we constructed  $\mathcal{M}_d$ , trivially

$$\begin{aligned} u'_i(\theta_i, \theta'_1, \theta'_2, \dots, \theta'_N) &= v_i(\theta_i, x'(\theta'_1, \theta'_2, \dots, \theta'_N)) - p'_i(\theta'_1, \theta'_2, \dots, \theta'_N) \\ &\stackrel{(2.2)}{=} v_i(\theta_i, x(s_1^*(\theta'_1), s_2^*(\theta'_2), \dots, s_N^*(\theta'_N))) - p_i(s_1^*(\theta'_1), s_2^*(\theta'_2), \dots, s_N^*(\theta'_N)) \\ &= u_i(\theta_i, s_1^*(\theta'_1), s_2^*(\theta'_2), \dots, s_N^*(\theta'_N)), \end{aligned} \tag{2.4}$$

for every type profile  $(\theta'_1, \theta'_2, \dots, \theta'_N) \in \Theta$ . Next, recall that  $(s_1^*, s_2^*, \dots, s_N^*)$  is a dominant strategy equilibrium of  $\Gamma(\mathcal{M})$ , thus (by substituting  $(a_i, a_{-i}) = (s_1^*(\theta'_1), s_2^*(\theta'_2), \dots, s_N^*(\theta'_N)) \in A$  at the equation of Definition 2.4) for all possible types  $\theta'_i \in \Theta_i$ ,  $\theta'_{-i} \in \Theta_{-i}$ ,

$$u_i(\theta_i, s_i^*(\theta_i), s_{-i}^*(\theta'_{-i})) \geq u_i(\theta_i, s_i^*(\theta'_i), s_{-i}^*(\theta'_{-i}))$$

and by (2.4),

$$u'_i(\theta_i, \theta_i, \theta'_{-i}) \geq u'_i(\theta_i, \theta'_i, \theta'_{-i}),$$

establishing truthfulness. Here we slightly abused notation for the sake of readability, writing  $(s_i^*(\theta_i), s_{-i}^*(\theta'_{-i}))$  instead of  $(s_1^*(\theta'_1), \dots, s_{i-1}^*(\theta'_{i-1}), s_i^*(\theta_i), s_{i+1}^*(\theta'_{i+1}), \dots, s_N^*(\theta'_N))$ .

Finally, we must show that  $\mathcal{M}_d$  implements  $f$ , under the same payments as  $\mathcal{M}$  does. Fix some type profile  $(\theta_1, \theta_2, \dots, \theta_N) \in \Theta$ . Since  $\mathcal{M}_s$  is truthful,  $\Gamma(\mathcal{M}_d)$  has a dominant strategy equilibrium, namely that consisting of the identity strategy functions, and the action profile that corresponds to that equilibrium is the truthful report  $(\theta_1, \theta_2, \dots, \theta_N)$ . At that dominant strategy equilibrium,

$$x'(\theta_1, \theta_2, \dots, \theta_N) \stackrel{(2.2)}{=} x(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N)) \stackrel{(2.1)}{=} f(\theta_1, \theta_2, \dots, \theta_N)$$

and the implementation follows immediately from Definition 2.8. For the payment rules at equilibrium, we have

$$p'_i(\theta_1, \theta_2, \dots, \theta_N) \stackrel{(2.3)}{=} p_i(s_1^*(\theta_1), s_2^*(\theta_2), \dots, s_N^*(\theta_N)) \quad \square$$

Simply put, the Revelation Principles allows us to restrict our attention to (direct revelation) truthful auctions, without loss of generality. After establishing such a positive result, we would be very happy if we had a simple way to check whether a given mechanism is truthful or not. Of course, this may very well be done using directly Definition 2.11.

However, many times this requires extensive case analysis and also does not give any design “hints” which could be used to construct truthful mechanisms. The following proposition gives such a characterization. It is essentially a demonstration of the characteristics that we intuitively considered being responsible for the “success” of the Vickrey auction (see page 29).

**PROPOSITION 2.13 (Truthfulness characterization)** *A (direct revelation) mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  is truthful if and only if*

(i) *Every player’s payment  $p_i$  does not depend (directly) on her type  $\theta_i$ , but on the other players’ types  $\theta_{-i}$  and the outcome  $x(\theta_i, \theta_{-i})$  decided by  $\mathcal{M}$ , i.e.*

$$x(\theta_i, \theta_{-i}) = x(\theta'_i, \theta_{-i}) := o \implies p_i(\theta_i, \theta_{-i}) = p_i(\theta'_i, \theta_{-i}) := p_o(\theta_{-i}),$$

and

(ii)  *$x$  decides the optimal, assuming other players’ types  $\theta_{-i}$  fixed, outcome for every player  $i$ , i.e.*

$$x(\theta_i, \theta_{-i}) \in \operatorname{argmax}_{o \in x(\Theta_i, \theta_{-i})} [v_i(\theta_i, o) - p_o(\theta_{-i})].$$

**PROOF**

( $\implies$ ) Assume that  $\mathcal{M}$  is truthful. We will prove that conditions (i) and (ii) hold. For condition (i), to arrive to a contradiction, assume that there exist player  $i$  with type  $\theta_i$  and types  $\theta'_i \in \Theta_i$ ,  $\theta_{-i} \in \Theta_{-i}$  for which

$$x(\theta_i, \theta_{-i}) = x(\theta'_i, \theta_{-i}) \tag{2.5}$$

but  $p_i(\theta_i, \theta_{-i}) \neq p_i(\theta'_i, \theta_{-i})$ . Without loss of generality, let

$$p_i(\theta_i, \theta_{-i}) < p_i(\theta'_i, \theta_{-i}). \tag{2.6}$$

The proof is essentially the same for the case of  $p_i(\theta_i, \theta_{-i}) > p_i(\theta'_i, \theta_{-i})$ . Then

$$\begin{aligned} u_i(\theta'_i, \theta'_i, \theta_{-i}) &= v_i(\theta'_i, x(\theta'_i, \theta_{-i})) - p_i(\theta'_i, \theta_{-i}) \\ &= v_i(\theta'_i, x(\theta_i, \theta_{-i})) - p_i(\theta'_i, \theta_{-i}), && \text{from (2.5),} \\ &< v_i(\theta'_i, x(\theta_i, \theta_{-i})) - p_i(\theta_i, \theta_{-i}), && \text{from (2.6),} \\ &= u_i(\theta'_i, \theta_i, \theta_{-i}), \end{aligned}$$

which contradicts truthfulness (see Definition 2.11).

For condition (ii), again to get to a contradiction, assume that there exists player  $i$ , type profile  $(\theta_i, \theta_{-i})$  and type  $\theta'_i \in \Theta_i$  such that

$$v_i(\theta_i, x(\theta_i, \theta_{-i})) - p_{x(\theta_i, \theta_{-i})}(\theta_{-i}) < v_i(\theta_i, x(\theta'_i, \theta_{-i})) - p_{x(\theta'_i, \theta_{-i})}(\theta_{-i}).$$

Then (by our notational convention of condition (i)), trivially

$$v_i(\theta_i, x(\theta_i, \theta_{-i})) - p_i(\theta_i, \theta_{-i}) < v_i(\theta_i, x(\theta'_i, \theta_{-i})) - p_i(\theta'_i, \theta_{-i})$$

and so,

$$u_i(\theta_i, \theta_i, \theta_{-i}) < u_i(\theta_i, \theta'_i, \theta_{-i}),$$

contradicting truthfulness once again.

( $\Leftarrow$ ) For the opposite direction, suppose that conditions (i) and (ii) do hold, and we need to show that  $\mathcal{M}$  allocates truthfully. To arrive to a contradiction, assume that there exist player  $i$  with type  $\theta_i$  and types  $\theta'_i \in \Theta_i$ ,  $\theta'_{-i} \in \Theta_{-i}$  such that

$$u_i(\theta_i, \theta_i, \theta'_{-i}) < u_i(\theta_i, \theta'_i, \theta'_{-i}).$$

Under condition (i) this gives

$$v_i(\theta_i, x(\theta_i, \theta'_{-i})) - p_{x(\theta_i, \theta'_{-i})}(\theta'_{-i}) < v_i(\theta_i, x(\theta'_i, \theta'_{-i})) - p_{x(\theta'_i, \theta'_{-i})}(\theta'_{-i}),$$

which contradicts condition (ii). □

## 2.3 VCG Mechanisms

In this section we will try to use the fundamental idea behind the Vickrey auction and the mechanism design “techniques” suggested by Proposition 2.13 in order to construct truthful mechanisms for general direct revelation environments (not only simple, one item sealed-bid auctions). Furthermore, the auctions we are going to introduce will also satisfy one other, very important property of the Vickrey auction: the maximization of the “combined happiness” of our players. Let’s start by making this formal, by considering as a measure for that “combined happiness” the sum of the players’ individual “satisfactions”, i.e. valuations.

**DEFINITION 2.14 (Efficiency)** Fix some direct revelation environment

$\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ . Assuming some type profile  $\theta = (\theta_1, \theta_2, \dots, \theta_N) \in \Theta$  fixed, the efficiency (or social welfare) of an outcome  $o \in \mathcal{O}$  is the value  $\sum_{i=1}^N v_i(\theta_i, o)$ . More succinctly, efficiency is the function  $E : \Theta \times \mathcal{O} \rightarrow \mathbb{R}$  with

$$E(\theta, o) = \sum_{i=1}^N v_i(\theta_i, o).$$

If, in addition, a specific (direct revelation) mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  over  $\mathcal{E}_d$  is given then the notion of efficiency can be naturally restricted to the decisions made by  $\mathcal{M}$ , defining the *efficiency of mechanism  $\mathcal{M}$*  to be the function  $E_{\mathcal{M}} : \Theta \rightarrow \mathbb{R}$  with  $E_{\mathcal{M}}(\theta) = \sum_{i=1}^N v_i(\theta_i, x(\theta))$ . Since this expression depends only on the decision rule  $x$  of the mechanism (and not on the payments) we sometimes write  $E_x$  instead of  $E_{\mathcal{M}}$ .

Obviously<sup>13</sup>, we would like to design mechanisms that *maximize* efficiency. Maximizing efficiency is the predominant objective in Mechanism Design and mechanisms that succeed in implementing social choice functions that achieve this maximization are usually called (socially) *efficient*. Alternatively, one could have considered maximizing the happiness of the least happy player as an implementation objective, i.e. try to maximize  $\min_{i \in \mathcal{N}} v_i(\theta_i, x(\theta))$ . To many (presumably those struggling for social equity...) this may seem as a more appropriate<sup>14</sup> measure of “social welfare”.

However “socially sensitive” we may be, we cannot ignore the fact that the motivation behind our thesis lies in the general field of *electronic commerce*, thus we must also take into consideration the “happiness” of the mechanism designer. In an auction setting, for example, we cannot expect the auctioneer to be someone spends time and resources to design a socially efficient auction without caring about his revenues<sup>15</sup>. So, sometimes we will use the maximization of the sum of the payments collected by all players as an objective:

**DEFINITION 2.15 (Revenue)** Let  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  be a direct revelation mechanism (over some environment  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ ). The (total) revenue of  $\mathcal{M}$  is the function  $R_{\mathcal{M}} : \Theta \rightarrow \mathbb{R}$  with

$$R_{\mathcal{M}}(\theta) = \sum_{i=1}^N p_i(\theta).$$

<sup>13</sup>Being socially sensitive...

<sup>14</sup>In fact, this social objective is used in many computational problems of Algorithmic Game Theory, most notably load balancing problems. E.g. see the seminal paper of Koutsoupias and Papadimitriou [1999], or [Vöcking, 2007] for a nice overview of such problems.

<sup>15</sup>It would be very hopeful to know that such people, who give away commodities and also make sure that this is done in a socially optimal way, do exist, however this certainly not the rule.



Now its time to start doing what we promised in the beginning of our section 2.3: designing truthful and socially efficient mechanism.

**DEFINITION 2.16 (Groves mechanisms)** *A Groves mechanism is a direct revelation mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  (over some environment  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ ) for which:*

(i) *the decision rule  $x$  maximizes efficiency  $E$ , i.e.*

$$x(\theta) \in \operatorname{argmax}_{o \in \mathcal{O}} \sum_{i=1}^N v_i(\theta_i, o)$$

(ii) *the payment  $p_i$  collected from every player  $i \in \mathcal{N}$  is of the form*

$$p_i(\theta) = h_i(\theta_{-i}) - \sum_{\substack{j \in \mathcal{N} \\ j \neq i}} v_j(\theta_j, x(\theta))$$

*where  $h_i : \Theta_{-i} \rightarrow \mathbb{R}$  is some function (independent of player's  $i$  type  $\theta_i$ ),*

*for all possible type profiles  $\theta = (\theta_1, \theta_2, \dots, \theta_N) \in \Theta$ .*

Admittedly, the payment expression on the above Definition 2.16 seems to have been conceived in a moment of infinite inspiration, however this choice is going to be sufficiently justified when we study the special case of VCG mechanisms at Definition 2.20, on page 37. The Vickrey auction will, as one expects, makes its appearance once more. A trivial observation to make here is that, although we are given great flexibility in the choice of  $h_i(\theta_{-i})$ , many choices can have significant effect in the payments collected by our mechanism and may make it have many totally undesirable properties.

**THEOREM 2.17** *Every Groves mechanism is truthful.*

**PROOF** We will prove our theorem using the characterization of Proposition 2.13. A proof directly from Definition 2.11 would also be possible (and actually more natural and common), however our aim here is to demonstrate the “higher level” interaction of the conditions in Proposition 2.13.

Let  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  be our Groves mechanism. From property (ii) of Definition 2.16 it is not difficult to see that the players' payments do not directly depend on their types and in particular (using the notation of Proposition 2.13(i))

$$p_o(\theta_{-i}) = h_i(\theta_{-i}) - \sum_{\substack{j \in \mathcal{N} \\ j \neq i}} v_j(\theta_j, o), \tag{2.7}$$

for all type profiles  $\theta = (\theta_i, \theta_{-i})$  and outcomes  $o = x(\theta_i, \theta_{-i})$ , thus satisfying condition (i) of Proposition 2.13.

Next, fix some player  $i$  with type  $\theta_i$  and types  $\theta_{-i}$ . Then, for every possible outcome  $o \in x(\Theta_i, \theta_{-i})$ ,

$$\begin{aligned} v_i(\theta_i, o) - p_o(\theta_{-i}) &= v_i(\theta_i, o) - \left[ h_i(\theta_{-i}) - \sum_{\substack{j \in \mathcal{N} \\ j \neq i}} v_j(\theta_j, o) \right], & \text{from (2.7),} \\ &= v_i(\theta_i, o) + \left[ \sum_{\substack{j \in \mathcal{N} \\ j \neq i}} v_j(\theta_j, o) \right] - h_i(\theta_{-i}) \\ &= \sum_{i=1}^N v_j(\theta_j, o) - h_i(\theta_{-i}), \end{aligned}$$

and due to the fact that  $h_i(\theta_{-i})$  is independent of  $o$ ,

$$\operatorname{argmax}_{o \in x(\Theta_i, \theta_{-i})} [v_i(\theta_i, o) - p_o(\theta_{-i})] = \operatorname{argmax}_{o \in x(\Theta_i, \theta_{-i})} \sum_{i=1}^N v_j(\theta_j, o). \quad (2.8)$$

Now notice that  $x(\Theta_i, \theta_{-i}) \subseteq \mathcal{O}$  and  $x(\theta) = x(\theta_i, \theta_{-i}) \in x(\Theta_i, \theta_{-i})$ , so condition (i) of Definition 2.16 gives  $x(\theta_i, \theta_{-i}) \in \operatorname{argmax}_{o \in x(\Theta_i, \theta_{-i})} \sum_{i=1}^N v_i(\theta_i, o)$  thus, from (2.8),

$$x(\theta_i, \theta_{-i}) \in \operatorname{argmax}_{o \in x(\Theta_i, \theta_{-i})} [v_i(\theta_i, o) - p_o(\theta_{-i})],$$

satisfying condition (ii) of Proposition 2.13, establishing truthfulness for  $\mathcal{M}$ .  $\square$

Next we are going to present two, superficially trivial, properties that we have “silently” expected our mechanisms to satisfy, primarily based on intuitions regarding specific examples of applications, notably auctions. However, up until now we have not stated them and, therefore, neither tested them.

**DEFINITION 2.18 (IR)** *A direct revelation mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  (over some environment  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ ) is called individually rational (IR) if, on the induced game  $\Gamma(\mathcal{M})$ , all players receive nonnegative utility by playing truthfully (i.e.  $s_i(\theta_i) = \theta_i$ ). More succinctly, for every player  $i \in \mathcal{N}$ ,*

$$v_i(\theta_i, x(\theta)) - p_i(\theta) \geq 0$$

for all type profiles  $\theta = (\theta_1, \theta_2, \dots, \theta_N) \in \Theta$ .

The left hand side of the inequality is essentially player's  $i$  utility. Simply put, we wouldn't like our mechanisms to punish players that just choose to "honestly" participate in it. Seeing it from another angle, no rational player would choose to be part of a mechanism that has the ability to force a loss to her.

**DEFINITION 2.19 (No positive transfers)** *We say that a direct revelation mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  (over some environment  $\mathcal{E}_d = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ ) makes no positive transfers if it never "pays" an agent (instead of receiving payment). Formally, for every player  $i \in \mathcal{N}$ ,*

$$p_i(\theta) \geq 0$$

for all type profiles  $\theta \in \Theta$ .

Note that no positive transfers is trivially a sufficient condition (though certainly not a necessary one) for assuring no loss to the mechanism designer (i.e. positive revenue).

In general, Groves mechanisms (see Definition 2.16) may not be IR, neither make no positive transfers. For the right choice of  $h_i(\theta_{-i})$ , however, we can guarantee these desirable properties are satisfied.

**DEFINITION 2.20 (VCG mechanisms)** *A VCG mechanism is a Groves mechanism  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  for which*

$$h_i(\theta_{-i}) = \max_{o \in \mathcal{O}} \sum_{\substack{j \in \mathcal{N} \\ j \neq i}} v_j(\theta_j, o).$$

*That means that a VCG mechanism is a direct revelation mechanism that maximizes efficiency and its payment rule is*

$$p_i(\theta) = \left[ \max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o) \right] - \sum_{j \neq i} v_j(\theta_j, x(\theta)).$$

Note that every VCG mechanism is completely determined by its environment  $\mathcal{E} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ . Therefore we can speak of *the* VCG mechanism (given a fixed direct revelation environment). The intuition behind the rather involved expression of the payment rule is that each player must submit a payment equal to the damage that her presence causes to the other players. In Economic Theory terms, we use such a payment to force each player internalize the externalities she causes. The first term in the expression equals the maximum combined satisfaction (efficiency) all other players could have

achieved if  $i$  was not present, while the second term represent the efficiency they end up having due to player's  $i$  participation. It is trivial to check that in a single item sealed-bid auction setting, the VCG mechanism is our beloved Vickrey auction.

**PROPOSITION 2.21** *Every VCG mechanism is truthful and makes no positive transfers. Furthermore, if we are in an environment for which  $v_i(\theta_i, o) \geq 0$  for every player  $i$ , all types  $\theta_i \in \Theta_i$  and all possible outcomes  $o \in \mathcal{O}$ , then the VCG mechanism is also IR.*

**PROOF** Every VCG mechanism is a Groves mechanism and thus it is truthful (Theorem 2.17). No positive transfers and IR remain to be shown. Let  $\mathcal{M} = (x, \{p_i\}_{i \in \mathcal{N}})$  be our VCG mechanism. Fix some player  $i$  and type profile  $\theta = (\theta_i, \theta_{-i})$ . Since  $x(\theta) \in \mathcal{O}$ ,  $\max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o) \geq \sum_{j \neq i} v_j(\theta_j, x(\theta))$  and so (from Definition 2.20)

$$p_i(\theta) = \left[ \max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o) \right] - \sum_{j \neq i} v_j(\theta_j, x(\theta)) \geq 0,$$

which means that  $\mathcal{M}$  makes no positive transfers (Definition 2.19).

Next, notice that, again because  $\mathcal{M}$  is a Groves mechanism, from property (i) of Definition 2.16 we get that

$$v_j(\theta_j, x(\theta)) \geq v_j(\theta_j, o)$$

for every player  $j \in \mathcal{N}$  and every possible outcome  $o \in \mathcal{O}$ . But we have assumed that  $v_i(\theta_i, o) \geq 0$  for all  $o \in \mathcal{O}$ , so

$$\sum_{j=1}^N v_j(\theta_j, x(\theta)) \geq \sum_{j=1}^N v_j(\theta_j, o) \geq \sum_{j \neq i} v_j(\theta_j, o)$$

for every  $o \in \mathcal{O}$ , thus

$$\sum_{j=1}^N v_j(\theta_j, x(\theta)) \geq \max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o). \quad (2.9)$$

Then,

$$\begin{aligned} v_i(\theta_i, x(\theta)) - p_i(\theta) &= v_i(\theta_i, x(\theta)) + \sum_{j \neq i} v_j(\theta_j, x(\theta)) - \max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o) \\ &= \sum_{j=1}^N v_j(\theta_j, x(\theta)) - \max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o) \\ &\geq 0, \end{aligned} \quad \text{from (2.9),}$$

establishing individual rationality (Definition 2.18). □

Before closing this section, we would like to mention that, generally, every truthful mechanism is essentially simply a variation of the VCG of mechanism. These mechanisms are *affine maximizers* (see [Nisan, 2007, p. 228]) called weighted VCG mechanisms and, informally, result from the payment rule of the standard VCG mechanism by adding appropriate weights to the various components. For more details, we refer to the references proposed in the Notes section 2.4 of this chapter.

## 2.4 Notes

Once again, as in Chapter 1, we refer to the textbooks of Fudenberg and Tirole [1991, chapters 6,7] and Mas-Colell et al. [1995, chapter 23] for a complete introduction in the field of classic Mechanism Design and incomplete information (Bayesian) games. In addition to them we suggest textbooks in Auction Theory, e.g. [Krishna, 2002]. Having as a motivation the most notable applications of Mechanism Design, i.e. auctions, they make extensive use of tools and notions from that field. The brilliant Vickrey auction is of course due to Vickrey [1961], while general Groves mechanisms are due to Groves [1973]. For the inspired choice of  $h_i$ 's in Definition 2.20 responsible is Clarke [1971] who had proposed his *pivotal rule* a couple of years earlier. The initials in "VCG" are a tribute to all previous three, the texts of whom essentially founded the field of Mechanism Design. Modern Social Choice Theory starts with the seminal work of Arrow [1951].

Our exposition in the current Chapter 2 is influenced in some extent by the well-written overview of Nisan [2007]. That text is highly recommended, as it is somehow biased towards a Computer Science audience, without neglecting the roots in Microeconomic Theory, neither compromising in rigor. Moreover, the term *Algorithmic Mechanism Design* was coined in Nisan and Ronen [1999]. This seminal paper marked the beginning of using tools from classic Mechanism Design to study many important computational problems.



# Chapter 3

## Competitive Analysis

### 3.1 Online Optimization Problems

Most of the times in Computer Science, we have to deal with some kind of **optimization problems**. In such problems there is an **objective function** which we try to optimize (maximize or minimize). Let us formalize our discussion, in order to be able to introduce our desired notions with a degree of necessary rigor and clarity.

Let  $P$  be an optimization problem. Problem  $P$  has a set of possible *inputs* (instances)  $\mathcal{X}$  and, for a given input  $x \in \mathcal{X}$  a set of *feasible solutions*  $\mathcal{F}(x)$ . More importantly we have an objective function  $\sigma : \bigcup_{x \in \mathcal{X}} (\{x\} \times \mathcal{F}(x)) \rightarrow \mathbb{R}_{\geq 0}$ . Without loss of generality, let us suppose that our goal is to *maximize* this objective function (the discussion can be readily adopted to the case of minimization problems). For every input  $x \in \mathcal{X}$  we try to find a feasible solution  $y$  that maximizes the value of  $\sigma(x, y)$ , i.e.  $y \in \operatorname{argmax}_{y \in \mathcal{F}(x)} \sigma(x, y)$ . Applying all these in an algorithmic setting, let  $A$  be an algorithm for our maximization problem  $P$ . In every input  $x \in \mathcal{X}$ , algorithm  $A$  computes a feasible solution  $A(x) \in \mathcal{F}(x)$  resulting to a value of  $\sigma(x, A(x))$  of the objective function. Summarizing, our goal is to find an algorithm  $A$  for problem  $P$  which maximizes  $\sigma(x, A(x))$  for all possible inputs  $x \in \mathcal{X}$ . Obviously, an *optimal* algorithm for our maximization  $P$ , denoted by  $\operatorname{OPT}_P$ , would be one satisfying

$$\operatorname{OPT}_P \in \operatorname{argmax}_A \sigma(x, A(x)) \quad \text{for all } x \in \mathcal{X}.$$

The “performance” of an optimal algorithm for a given problem can be used as a “benchmark” to measure the performance of other algorithms for the same problem. We say that an algorithm  $A$  for a maximization problem  $P$  is a  $c$ -approximation algorithm if it performs within a factor of  $c$  with respect to the optimal algorithm for the same problem,

i.e.

$$\sigma(x, \text{OPT}_p(x)) \leq c \cdot \sigma(x, A(x)) \quad \text{for all } x \in \mathcal{X}. \quad (3.1)$$

Note that,  $c \geq 1$ .

In “traditional” algorithmic design, when we try to solve an (optimization) problem we generally assume that every input  $x$  is presented to our algorithm *instantaneously* and *in full*. The efforts of the algorithm designer is how to efficiently (usually with respect to running time) compute an optimal feasible solution (which maximizes the objective function). If that is not possible, we may settle for some other efficient approximation algorithm that performs within a good factor  $c$  with respect to the optimal (see (3.1)). This is the very idea behind the very important field of *Approximation Algorithms*.

On the other hand, many computational problems naturally require to be modeled in a totally different way. In a way that their input is revealed in an *online* (time-dynamic) fashion. An algorithm for such a problem has to make a series of decisions, based only on past events and prior, already received information about our input. No reliable information about the future can be available to us. The input is constructed in a dynamic way. Such problems are called **online problems** and many important problems fall within this framework. Notable examples include the paging problem, many routing and load balancing problems and the  $k$ -server problem. Classic textbook example include the ski-rental problem and the lost-cow problem (also known as the cow-path or bridge problem). We will use the term *offline* to refer to “traditional” optimization problems.

## 3.2 Competitive Analysis

The important question that arises in online computation is how we can measure the performance of various algorithms for some online optimization problem. Towards this direction we adopt a worst-case framework. Let  $P$  be an online maximization problem with an objective function  $\sigma$  and  $A$  be an online algorithm for  $P$ . These notion are extended in the natural way in our online computational setting. We will compare the performance of our algorithm  $A$  to that of an **optimal offline** algorithm  $\text{OPT}_p$  which we assume that has access to the whole input  $x$ , in an offline manner. Of course, the existence of such an algorithm is practically unrealizable, since the input is revealed dynamically and cannot be known in advance. However, we can use it as a theoretical a benchmark to serve our worst-case analysis framework, since obviously it will generally perform (weakly better) than every online algorithm due to the fact that it has access to the entire input and thus



can optimally plan its actions.

In the spirit of Approximation Algorithms and equation (3.1) we give the following definition

**DEFINITION 3.1** *Let  $A$  be an online algorithm for some (online) maximization problem  $P$  with objective function  $\sigma$ . We will say that  $A$  is  $c$ -competitive<sup>1</sup> if*

$$\frac{\sigma(x, \text{OPT}_P(x))}{\sigma(x, A(x))} \leq c \quad \text{for all } x \in \mathcal{X}. \quad (3.2)$$

Note that, if we assure that  $\sigma(x, A(x)) \neq 0$  for all  $x \in \mathcal{X}$ , the condition of the above Definition 3.1 is equivalent to equation (3.1). But it is natural to assume that  $\sigma(x, A(x)) \neq 0$  holds for every  $x$  since, in the opposite case (3.1) would have resulted in  $\sigma(x, \text{OPT}_P(x)) \leq 0$  for some input  $x$  which essentially contradicts the optimality of  $\text{OPT}_P$  (remember that  $\sigma \geq 0$ ).

It easy to see that if an online algorithm is  $c$ -competitive then it is also  $(c+\epsilon)$ -competitive for every  $\epsilon > 0$ . We are, therefore, interested in the “critical” minimum value of  $c$  for which condition (3.2) still holds. This is very important, because it essentially determines the performance of  $A$ . So, what we are looking for is actually the least upper bound<sup>2</sup> of the set  $\left\{ \frac{\sigma(x, \text{OPT}_P(x))}{\sigma(x, A(x))} \leq c \mid x \in \mathcal{X} \right\}$ .

**DEFINITION 3.2** *Let  $A$  be an online algorithm for some (online) maximization problem  $P$  with objective function  $\sigma$ . The competitive ratio of  $A$  is*

$$\text{CR}_P(A) = \sup_{x \in \mathcal{X}} \frac{\sigma(x, \text{OPT}_P(x))}{\sigma(x, A(x))}. \quad (3.3)$$

In many cases this supremum can be replaced by a maximum, for example if the input space  $\mathcal{X}$  is finite. Also, in the case that for a given optimization problem  $P$  we have more than one objective functions, we will enrich our notation to  $\text{CR}_P^\sigma(A)$  in order to make clear with respect to which objective function our competitive ratio is computed. The smaller the competitive ratio of an algorithm, the better its performance. Expression (3.3) can help us develop a very useful interpretation of the notion of competitive ratio. We can think of competing against a malicious, almighty **adversary** which always chooses the

<sup>1</sup>Sometimes this term is used to describe a more relaxed condition than that of our definition, namely  $\sigma(x, \text{OPT}_P(x)) \leq c \cdot \sigma(x, A(x)) + \alpha$  for some constant  $\alpha > 0$ , allowing some flexibility with respect to initialization costs that many problems unavoidably have. In such a case, the term *strictly*  $c$ -competitive is used to describe the condition of our Definition 3.1. However, in this thesis we will only need the strict case of our initial definition.

<sup>2</sup>For a definition, consult any serious book in calculus or analysis, e.g. [Rudin, 1976, Definition 1.8].

worst input  $x$  for us, in order to achieve the supremum of expression (3.3), maximizing our competitive ratio. This could be some kind of “evil” God that has complete information of our environment and the algorithm  $A$  which we have selected to run and always constructs the worst input for us, revealing it to us in the most disastrous online way possible.

We pick the best among all possible algorithms for a given online problem to represent the competitive ratio of that problem.

**DEFINITION 3.3 (Competitive ratio)** *Let  $P$  an online maximization problem with an objective function  $\sigma$ . Problem’s  $P$  competitive ratio is*

$$\text{CR}_P = \inf_A \sup_x \frac{\sigma(x, \text{OPT}_P(x))}{\sigma(x, A(x))}.$$

The adversarial interpretation is once again very useful: we first choose an online algorithm  $A$  in our effort to minimize the competitive ratio for our problem. Next, the adversary, knowing our decision  $A$ , chooses the worst input  $x$  in order to maximize our competitive ratio.

Throughout the years many techniques have been deployed to prove lower and upper bounds of the competitive ratios for specific problems. Some of them were inherited from other fields of Computer Science (especially from Approximation Algorithms) and others were developed with competitive analysis in mind (such Yao’s lemma and the potential function method). Although we are not going to analyse further any of these techniques (we refer to the Notes section 3.3 for appropriate references) there is a simple observation on 3.3 that underlies the whole quest for lower and upper bounds. At first, the competitive ratio of a specific algorithm  $A$  is an upper bound on the competitive ratio of the general problem  $P$ . And secondly, fixing a specific input  $x$ , the infimum of the competitive ratio fraction (see, e.g. expression Definition 3.3) as we run through all possible algorithms  $A$  is also a lower bound of the competitive ratio of the general problem  $P$ . Summarizing, in order to prove upper bounds we need to find specific algorithms that perform well and in order to prove lower bounds we need to find bad inputs that cause all algorithm to run poorly. Obviously, when the upper and lower bounds coincide then we have managed to compute the exact value of the competitive ratio for our problem.

We will refer to the worst-case analysis framework which we presented in this section as **competitive analysis**. In Economic Theory (and other scientific disciplines) the usual framework in which such problems are studied is that of average-case (Bayesian) analysis. In such a framework, one makes some distributional assumption with respect to the input  $x$  and then tries to compute the expected value of the objective function. In competitive

analysis we make no assumptions at all. We want our performance factors to hold even against an almighty, malicious adversary.

Despite all that, we may be able to weaken our adversary's power, without leaving our competitive analysis framework: by allowing randomization. By letting a coin to be tossed at some point during the execution of our algorithm, the adversary will again know the algorithm  $A$  we are using before he decides what's the worst input  $x$  for us, but he is not in position of securely knowing the realization of the random events. This may result in our adversary's choice  $x$  not maximizing  $\sigma(x, \text{OPT}_p(x))$ .

The adversarial model we adopt in our framework of competitive analysis is sometimes referred to as *oblivious* adversary. There are other stronger adversarial models, which for example are not affected by randomization (adaptive offline adversaries) and weaker ones, which for example force the adversary to make some of his decisions before ours (adaptive online adversary). For more details we refer to the text suggestions of the Notes section 3.3 of our chapter.

### 3.3 Notes

Our exposition in this chapter closely resembles that of [Borodin and El-Yaniv, 1998], which is also the standard reference textbook in the area. The collection of papers edited by Fiat and Woeginger [1998] is also cited often. For a study of the power of randomization in competitive analysis and the various adversarial models they occur we refer to [Ben-David et al., 1994]. Finally, it is apparent from our exposition in section 3.1 that competitive analysis has its roots in the field of Approximation Algorithm and therefore it has inherited many notions, techniques and ideas from it. An excellent book on the subject is that of Vazirani [2001].



## **Part B**

# **Online Mechanism Design**



# Chapter 4

## Online Mechanism Design

In this chapter we will extend the framework of classic Mechanism Design introduced in Chapter 2 so as to incorporate time-dynamic environments. Such settings not only are more natural and general but also can model problems of online nature, i.e. situations at which information about our environment and its specific variables is revealed to us in a time-dependent way rather than statically and in full. This is of great importance to us since the very goal of this thesis is to study such *decision-making under uncertainty* problems and in particular dynamic auctions at which either the bidder set or is not fully known by the auctioneer but are revealed in an online fashion.

Without doubt, here we are using motivation and techniques from the field of Online Algorithms and so the reader is advised to familiarize himself with the spirit and context of Chapter 3. On the other hand, the fundamentals of Online Mechanism Design are not presented in the formal way of Chapter 2. Although our exposition will be rigorous and cautious, the notions will be introduced as a means to lay the ground for the powerful results we are going to show at the following chapters rather than for the sake of formality. In this way, this chapter is somehow self-contained as far as the fundamental notions of Mechanism Design are concerned. However, the reader will surely benefit from building a solid background in the spirit of Chapter 2.

We consider a **dynamic environment** consisting of *discrete* (possibly infinite) time periods  $\mathcal{T} = \{1 < 2 < 3 < \dots\}$  which we will usually index by  $t$  and a finite set of participating agents  $\mathcal{N} = \{1, 2, \dots, N\}$  which will be indexed by  $i$ . An (online) mechanism in this environment enforces a sequence of **decisions**  $k = (k^1, k^2, \dots) \in \mathcal{O}$ , decision  $k^t$  made at time period  $t$ ,  $\mathcal{O}$  being the set of all possible **outcomes**. We will use the notation  $k^{[t_1, t_2]} = (k^{t_1}, k^{t_1+1}, \dots, k^{t_2})$  to refer to the decisions made during a discrete time interval  $[t_1, t_2] = \{t \in \mathcal{T} \mid t_1 \leq t \leq t_2\}$ ,  $t_1, t_2 \in \mathcal{T}$ . Agent's  $i$  **type** is a tuple  $\theta_i = (a_i, d_i, w_i)$  where  $a_i, d_i \in \mathcal{T}$ ,  $a_i \leq d_i$  and  $\Theta_i$  will denote the set of all such possible types. We refer to  $a_i$  and

$d_i$  as the **arrival** and **departure time**, respectively, of agent  $i$ .

For every agent  $i$  we define a **valuation function** (or simply valuation)  $v_i : \Theta_i \times \mathcal{O} \rightarrow \mathbb{R}$  where agent's  $i$  **value**<sup>1</sup>  $v_i(\theta_i, k)$  depends only on decisions made within her arrival-departure time interval  $[a_i, d_i]$ , that is

$$v_i(\theta_i, k) = v_i(\theta_i, k^{[a_i, d_i]}).$$

We use the **valuation component**  $w_i$  of agent's  $i$  type in order to parametrize<sup>2</sup> in a rigorous and flexible way her valuation function. Although this is not made explicitly clear through our notation, we don't demand agent's  $i$  valuation component to be constant through time but we allow, for instance, situations in which an agent could gradually discount her valuation component by a factor  $\gamma^{t-a_i}$  at future periods  $t > a_i$ , where  $\gamma \in (0, 1)$ . In this way she can parametrize an equivalent time-dependent discount in her values<sup>3</sup>. So we can consider agent's  $i$  valuation component  $w_i$  to be a sequence  $w_i = (w_i^1, w_i^2, \dots)$ ,  $w_i^t$  being the valuation component at time period  $t$ . However, at this point we decide to keep the notation light and we will simply write  $w_i$ .

Our (online) mechanism also defines a **payment rule**  $p = (p_1, p_2, \dots, p_N) \in \mathbb{R}^N$  where  $p_i$  is the payment "collected" from agent  $i$ . We could assume that  $p_i \geq 0$  for every agent  $i$ , which is called **no-deficit principle**<sup>4</sup>, due to the fact that it is a sufficient condition to guarantee no deficits for our mechanisms. A negative payment  $p_i < 0$  models situations in which the mechanism makes a direct payment *to* agent  $i$  (instead of actually *receiving* one)<sup>5</sup> and this could lead to an overall deficit for the mechanism. If we want to exclude such

---

<sup>1</sup>The terms "valuation" (function) and "value" are traditionally used interchangeably. However, if we want to be accurate, we will use the term "valuation" to refer to the function and "value" to refer to a value of this function at some input  $(\theta_i, k)$ .

<sup>2</sup>Parameter  $w_i$  need not to be always a real number, implementing, for example, an ordering upon agent's preferences. Many times we will need to express more complex parametrizations. Consider, for instance, an online auction in which we want to be able to express preferences such as "Agent  $i$  wants item  $A$  or item  $B$  but not both".

<sup>3</sup>Think, for example, of the price a consumer is willing to pay in order to buy a new laptop computer. When the new model comes out it incorporates many new, "hot" technological features and so (with a little help from advertisements) is highly valued from potential buyers. As time goes by, new models and features appear at the market and gradually prices drop due to the fact that consumers are willing to pay less for older technology. After a year or so, this computer is already considered as a "previous generation" model and possibly many consumers have reached the end point  $d_i$  of their interest to the product. Although we don't want, in any way, to argue that prices in markets are formed through well defined algorithmic mechanisms (like the ones we study in this project), this is still a nice example to understand the time dependency of an agent's value.

<sup>4</sup>This is exactly the same as the *no positive transfers* notion Definition 2.19, page 37, but here we preferred to use the no-deficit term as this is the one more usually found in *online* mechanism design literature.

<sup>5</sup>Many times this is exactly what we want. Consider, for example, an auction at which the auctioneer is the government and auctions the construction of a new airport. In this auction setting the agents are construction companies which report the amount of money they want in order to undertake the project,



---

behavior, it is sufficient to adopt the no-deficit principle. However, what we are actually going to need later on is a weaker condition (Definition 5.12, page 69).

Then, we define agent's  $i$  **utility** (for our mechanism deciding  $k$  and collecting payment  $p$ ) to be the difference

$$u_i = v_i(\theta_i, k) - p_i \quad (4.1)$$

If we think of agent's  $i$  value  $v_i(\theta_i, k)$  as showing how much she “values” decision  $k$  (which is a very natural and useful interpretation) then her utility  $u_i$  shows what she “gains” from participating in our online mechanism setting deciding  $k$ . It represents the *balance* between how much she benefits from the mechanism ( $v_i(\theta_i, k)$ ) and how much she has to give back to it (payment  $p_i$ ).

Based on this interpretation, it is intuition-compatible but also technically useful to assume that our agents are *risk neutral*, that is

$$u_i \geq 0 \iff v_i(\theta_i, k) \geq p_i \quad (4.2)$$

for every agent  $i$ , type  $\theta_i$  and decisions  $k$  and payment rule  $p$  made by our mechanism. This demands nonnegative utility for all agents. We do not want to study mechanisms that can force a deficit to an agent, so we restrict our attention to environments in which agents decide to participate “strategically”, only because they have the chance of gaining something from participating, but nothing to lose. That is why this property is known as *voluntary participation*. The approach here is compatible with one of the cornerstones of Game Theory in general: the primary assumption that players act *rationally*. So, from now on we will refer to the above fundamental property (4.2) as **individual rationality<sup>6</sup> (IR)**.

We could have introduced our notions of dynamic environments and mechanisms in such environments in a more formal way, like we did for classic Mechanism Design in Chapter 2, e.g. defining a dynamic environment to be a tuple  $\mathcal{E}^{\mathcal{T}} = (\mathcal{N}, \{\Theta_i\}_{i \in \mathcal{N}}, \mathcal{O}, \{v_i\}_{i \in \mathcal{N}})$ , etc.. However, as we mentioned at the beginning of this chapter, such a formal exposition was carried out in Chapter 2 and can naturally be extended for our dynamic model, if needed.

The term *Online Mechanism Design* was coined in the important paper of Friedman and Parkes [2003] in a model resembling that presented here (although lacking departure times). Earlier, Lavi and Nisan [2000] have used the term *online auction*, deploying a totally different, economic theoretic model. Our exposition in this Part B of our thesis

---

and are actually going to get payed by the auctioneer.

<sup>6</sup>We have already seen this term in Definition 2.18, page 36.

draws elements from the two basic papers that cover the specific online auction examples we are going to study in Part C. Namely these papers are [Hajiaghayi et al., 2005] and [Hajiaghayi et al., 2004]. However, in spirit we are closer to Parkes [2007].

## 4.1 Direct-Revelation Mechanisms

The messages that agents send to the mechanism are exactly the reports for their types  $\theta_i$ . It is more accurate, though, to speak of *claims* rather than “reports”, because it is obvious that an agent can (and will, actually) lie about her true type if this is to maximize her utility. We call this behavior *misreporting*. These misreports are agents’ *strategies* (in the context of Game Theory). We will consider only **direct-revelation** mechanisms, that is, online mechanisms that restrict the message an agent can send to a *single, direct* claim about her type<sup>7</sup>. This means that agent  $i$  directly makes a report  $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{w}_i)$  about its type (that can be different from its true type  $\theta_i = (a_i, d_i, w_i)$ ) and this claim is made only once (at a single time period  $t \in \mathcal{T}$ ) during the execution of the mechanism. Furthermore, we will usually consider our mechanisms to be *closed*, which means that participating agents get no feedback information about other agents’ types before they “enter” the mechanism, that is, before they report their own types. This is crucial because no agent can condition her strategy upon other agents’ reports.

We define a **mechanism state**  $h^t$  for every time period  $t \in \mathcal{T}$  which captures all information relevant to the decision  $k^t$  made by the mechanism in that period  $t$ . We denote the set of all such possible states in period  $t$  with  $H^t$ . We also define  $\omega \in \Omega$  to be the set of all possible stochastic<sup>8</sup> events that can occur in our dynamic environment,  $\Omega$  denoting the collection of all such possible sets. For example,  $\omega$  may include the realization of uncertainty about supply to the mechanism. We write  $\Omega = \prod_{t \in \mathcal{T}} \Omega^t$  and we let  $\omega^t \in \Omega^t$  denote the information about  $\omega$  that is revealed to the mechanism at time period  $t$ . In a similar way we let  $\theta^t$  denote the set of agent types reported at period  $t$ .

Given this, and the description of the setting of our mechanisms we have made so far, it is convenient to define

$$h^t = (\theta^1, \dots, \theta^t; \omega^1, \dots, \omega^t; k^1, \dots, k^{t-1})$$

<sup>7</sup>We have already introduced direct-revelation mechanisms, in a more formal way, at section 2.2 during our exposition of the fundamentals of classic Mechanism Design. See Definition 2.9 and Definition 2.10, page 28.

<sup>8</sup>We must make clear here that this does not include the types of the agents nor any randomization within the mechanism itself.

letting state  $h^t$  capturing information about all reported types and stochastic events up to the current time period  $t$  and decisions made by the mechanism so far (notice that decision  $k^t$  has not been made yet and that is why the superscripts at the decisions list runs up to  $t-1$ ). In practice and in the mechanism-specific examples that follow only a portion of this information will be used. The state space  $H = \bigcup_{t \in \mathcal{T}} H^t$  may be finite, countably infinite or uncountable. This depends, in part, on whether agent types are discrete or continuous and to be more specific, on the cardinal numbers of the sets  $\mathbb{W}_i$  of the possible valuations of the agents. Let  $K(h^t)$  denote the set of all feasible decisions of the mechanism in the current time period  $t$ . We assume  $K(h^t)$  to be *finite* for all  $h^t \in H^t$ . Finally, we let  $I(h^t)$  denote the set of all *active* agents in state  $h^t$ . By “active” we mean an agent  $i$  for whom  $t \in [a_i, d_i]$ , i.e. current time period belongs to her *reported* arrival-departure interval.

The following definition describes direct-revelation online mechanisms in the spirit of Mechanism Design and establishes the relevant notation we are going to use:

**DEFINITION 4.1 (Direct-revelation online mechanism)** *A direct-revelation online mechanism  $\mathcal{M}^{\mathcal{T}} = (x, \{p_i\}_{i \in \mathcal{N}})$  restricts each agent to making a single claim about its type and defines decision policy  $x = \{x^t\}^{t \in \mathcal{T}}$  and payment policy  $p = \{(p_1^t, p_2^t, \dots, p_N^t)\}^{t \in \mathcal{T}}$ , where decision  $x^t(h^t) \in K(h^t)$  is made in state  $h^t$  and payment  $p_i^t(h^t) \in \mathbb{R}_{\geq 0}$  is collected from each agent  $i \in I(h^t)$ .*

This definition includes a couple of subtle properties of our mechanisms which worth further clarification. Decision policy  $x$  may very well be stochastic (again, this is different from the algorithmic randomization within the mechanism itself), depending on the realization of uncertain events  $\omega \in \Omega$ . Also, the payment policy  $\pi$  has the freedom to collect payments from the same agent across different time periods. To keep the notation convenient and coherent, we let  $x(\theta, \omega) = (x^1(h^1), x^2(h^2), \dots)$  denote the sequence of decisions and  $p_i(\theta, \omega) = \sum_{t \in \mathcal{T}} p_i^t(h^t)$  denote the *total* payment collected from agent  $i$ , given type profile  $\theta = (\theta_1, \theta_2, \dots, \theta_N)$  and the stochastic parameter  $\omega \in \Omega$ . In this way we can look at agent’s  $i$  utility  $u_i$  at expression (4.1) as a function

$$u_i(\theta_i, \hat{\theta}, \omega) = v_i(\theta_i, x(\hat{\theta}, \omega)) - p_i(\hat{\theta}, \omega), \quad (4.3)$$

where  $\theta_i$  is agent’s  $i$  true type and  $\hat{\theta}$  is the reported type profile (from the agents to the mechanism). Sometimes we can lighten the notation further if decision policy  $x$  is given and just write

$$u_i(\theta_i, \hat{\theta}, \omega) = v_i(\theta_i, \hat{\theta}, \omega) - p_i(\hat{\theta}, \omega).$$

Finally, note that the notions of efficiency and revenue introduced for offline mechanism in Chapter 2 (see Definition 2.14 and Definition 2.15, page 33) can be readily transferred to our online setting and in fact they will serve as objective functions in the optimization online problems we are going to study in the remaining of this thesis.

## 4.2 Limited Misreports

As we mentioned earlier (section 4.1, page 52) agents can lie about their true types and the strategy space of every agent  $i$  consists exactly of all these possible. Based on this we give the next definition:

**DEFINITION 4.2 (Limited misreports)** *The set of available misreports to an agent  $i$  is a subset  $C(\theta_i) \subseteq \Theta_i$ , where  $\theta_i$  is her true type.*

We interpret this set of available misreports  $C(\theta_i)$  as the possible claims an agent  $i$  can make about her true type  $\theta_i$ . Of course  $\theta_i \in C(\theta_i)$ , i.e. an agent can claim his true type  $\theta_i$ , although this is not a “mis-report” if we use the exact meaning of the word in english<sup>9</sup>. In the standard treatments of *offline* Mechanism Design it is typical to assume  $C(\theta_i) = \Theta_i$ , which means that we apply no restrictions over the reports of agents. Every agent can declare any possible type  $\hat{\theta}_i \in \Theta_i$  as being her true one.

On the contrary, in our *online* setting, most of the times we are going to assume **no early-arrival** misreports, which means that we limit the strategy space so that no agent can report an earlier arrival  $\hat{a}_i$  than her true one  $a_i$ , demanding  $a_i \leq \hat{a}_i$ . It is a very natural assumption to make because we can think of the real arrival time  $a_i$  as the very first moment that agent  $i$  is able to participate at the mechanism – like she has no knowledge of her type (or even of the mechanism!) before that point in time. We make no restrictions to the valuation claim  $\hat{w}_i$ . So, the no early-arrivals assumption implies

$$C(\theta_i) = \left\{ \hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{w}_i) \mid a_i \leq \hat{a}_i \leq \hat{d}_i \wedge \hat{w}_i \in \mathbb{W}_i \right\},$$

where  $\theta_i = (a_i, d_i, w_i)$  is the true type of agent  $i$ . In addition to no early-arrivals, we will sometimes also assume **no late-departures** misreports, i.e.  $\hat{d}_i \leq d_i$ , which means that no agent can delay (or, to be more precise, report to delay) his departure from the mechanism

<sup>9</sup>From now on we will use the word “misreport” in the context of Definition 4.2, diverging slightly from the use of the word in english.

further than his true departure time. This property, together with no early-arrivals, gives

$$C(\theta_i) = \left\{ \hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{w}_i) \mid a_i \leq \hat{a}_i \leq \hat{d}_i \leq d_i \wedge \hat{w}_i \in \mathbb{W}_i \right\},$$

or, more compactly,

$$[\hat{a}_i, \hat{d}_i] \subseteq [a_i, d_i].$$

Although, as we are going to see later, the no late-departures property is “necessary” in some specific environments in order to assure desirable properties of our mechanisms (see, e.g. Theorem 6.8, page 79), it is, in a way, a less natural

assumption than the no early-arrivals to make. It is obvious that an agent cannot participate in a mechanism before she even knows about it (no early-arrivals) but she might gain something from declaring that she is willing to stay active for longer than she really is. And that, because she might not eventually “reach” her reported departure time  $\hat{d}_i$  but receive the maximized utility (as the result of lying) at a time period  $t \leq d_i \leq \hat{d}_i$  previous to her actual departure time  $d_i$ , so her lie does not get “exposed”. From now on, we will explicitly mention the use of the no-late departures assumption whenever we employ it and we are going to comment on relaxing it, whenever this is feasible.

### 4.3 Truthfulness

For the following, in our usual notation,  $C(\theta_{-i}) = \prod_{j \neq i} C(\theta_j)$  where  $C(\theta) = \prod_i C(\theta_i)$  is the set of all possible type profile misreports  $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N)$ , given true type profile  $\theta = (\theta_1, \theta_2, \dots, \theta_N)$ .

**DEFINITION 4.3 (DSIC, truthful)** *An online mechanism  $\mathcal{M}^{\mathcal{F}} = (x, \{p_i\}_{i \in \mathcal{N}})$  is called dominant-strategy incentive-compatible (DSIC) or just truthful<sup>10</sup>, given limited misreports  $C(\theta)$ , if*

$$u_i(\theta_i, \theta_i, \theta'_{-i}, \omega) \geq u_i(\theta_i, \hat{\theta}_i, \theta'_{-i}, \omega), \quad (4.4)$$

for every agent  $i$  with true type  $\theta_i$ , all possible type profile misreports  $\hat{\theta} \in C(\theta)$ ,  $\theta'_{-i} \in C(\theta_{-i})$  and stochastic parameter  $\omega \in \Omega$ .

The expression (4.4) in Definition 4.3 is trivially equivalent to

$$v_i(\theta_i, x(\theta_i, \theta'_{-i}, \omega)) - p_i(\theta_i, \theta'_{-i}, \omega) \geq v_i(\theta_i, x(\hat{\theta}_i, \theta'_{-i}, \omega)) - p_i(\hat{\theta}_i, \theta'_{-i}, \omega)$$

<sup>10</sup>The term *strategyproof* is also widely used.

if we use equation (4.3). Here, we write  $u_i(\theta_i, \theta_i, \theta'_{-i}, \omega)$ ,  $x(\theta_i, \theta'_{-i}, \omega)$  and  $p_i(\theta_i, \theta'_{-i}, \omega)$  instead of the more accurate  $u_i(\theta_i, (\theta_i, \theta'_{-i}), \omega)$ ,  $x((\theta_i, \theta'_{-i}), \omega)$  and  $p_i((\theta_i, \theta'_{-i}), \omega)$ , for the sake of readability.

It is clear that equation (4.4) is a dominant strategy condition implying that, for every agent  $i$ , the report of her true type  $\theta_i$  is a (weakly) dominant strategy. No matter what other agents' claims  $\theta'_i$  are, she is better off telling the truth  $\theta_i$  than every possible (other) misreport  $\hat{\theta}_i$ . Every agent *maximizes* her utility by being truthful and thus has no incentive to lie.

If our mechanism is randomized, i.e. its decision policy  $x$  is stochastic, then for truthfulness (DSIC) we need equation (4.4) to hold for the *expected* utilities of the agents, for every  $\theta'_{-i} \in C(\theta_{-i})$  and  $\omega \in \Omega$ :

$$\mathbb{E}_x [u_i(\theta_i, \theta_i, \theta'_{-i}, \omega)] \geq \mathbb{E}_x [u_i(\theta_i, \hat{\theta}_i, \theta'_{-i}, \omega)],$$

where the expectation is taken with respect to the randomization of the policy  $x$ . Every agent maximizes her expected utility by telling the truth, regardless of the reports of the other agents and the *external* stochastic events  $\omega$ . But for randomized mechanisms we can define a much stronger notion of truthfulness than DSIC. Instead of maximizing the *expected* utility we can require truthful reporting being a dominant strategy for *every* event of the randomized policy, maximizing utility for all “random coin flips”. We will call this property **strong-truthfulness** and the randomized mechanisms that satisfy it, *strongly truthful*.

It goes without question that truthfulness is such an important and desired property in Mechanism Design that we virtually ignore every mechanism that fails to satisfy it. However, it would be nice to know that such an intuitive restriction to the study of only a special family of mechanisms is without loss of generality. More specifically, we would ideally like the Revelation Principle (Theorem 2.12, page 30) of classic (offline) Mechanism Design to be extended in our online setting. In fact, the Revelation Principle is so deeply rooted in the beliefs of researchers studying the field of Algorithmic Mechanism Design that we many times take for granted that an *online* Revelation Principle<sup>11</sup> *must* hold.

It turns out that this is not (always) the case, since we can give examples of online mechanism design environments in which some non-direct revelation mechanisms cannot be “simulated” by truthful ones. In particular, if in some environment turns out that some

<sup>11</sup>We will not formally state the online Revelation Principle, since this would essentially be a rewriting of Theorem 2.12 of the classic Revelation Principle. We have it in our mind as the adaptation (in the way one would expect) of the idea of this classic result, to our online setting.

agent might *not* to be able to send a message (recall the notion of incomplete information games) in some time period during which she was supposed (base to her report) to be active, then we can exploit this weakness and give a counter example to the online Revelation Principle. Such an example can be found in [Parkes, 2007, p. 416] and is a simple adaptation of an example given originally by Pai and Vohra [2006, part B] in a slightly different model of online mechanism design. However, if we demand no-late departures in addition to our standard, arguably natural, assumption of no-early arrivals then online Revelation Principle do hold. Alternatively, we could demand each player to send to the mechanism an empty, “heartbeat” message at every time point. We omit the formal proof of this online Revelation Principle, since it closely follows that of the classic one, using some simple observations about “legal” message reporting (remember that we have general, non-direct revelation mechanisms) which can be found in a proof originally again given in [Pai and Vohra, 2006, part B].





# Chapter 5

## Single-Valued Online Domains

In this chapter we restrict our attention to a very specific (yet very inclusive) family of mechanism design environments within our general dynamic setting, namely single-valued online domains. In such settings the agents' preferences are expressed through a very simple and clear valuation expression which is of a "yes-or-no" nature. This is particularly useful when trying to model auctions, due to the fact in most such environments an agent is either fully satisfied, if he has received the wanted item(s), or totally unsatisfied otherwise. Of course this is not always the case, e.g. think of an auction setting where agents have different values for various item bundles. However, such combinatorial auctions are not the subject of this thesis and therefore we will not deal with them.

Apart from being able to model all auction problems in which we are interested in this project, single-valued domains are of utmost importance to us for one more reason: we can give a complete characterization of *truthful* mechanisms within them. This is, obviously, something we care very much about, and it is not only restricted to the case of online Mechanism Design. Single-valued domains, along with their very good properties, can be also defined in a classic (offline) Mechanism Design setting. However, we waited until current Chapter 5 to make our exposition rather than in Chapter 2. This is because all the results we are going to obtain in our online setting are essentially a generalization of those in the classic case and thus, one can easily adapt them to case of offline Mechanism Design<sup>1</sup>, if needed.

Again, in this chapter we have the papers of Hajiaghayi et al. [2005] and Hajiaghayi et al. [2004], however we use elements from the more recent exposition in Parkes and Duong [2007].

---

<sup>1</sup>For such an exposition of the results of this chapter in the spirit of classic Mechanism Design of Chapter 2, the reader is referred to [Nisan, 2007, section 9.5.4]

## 5.1 Basic Definitions

We know (see page 53) that  $K(h^t)$  is the set of all feasible decisions at current time period  $t$ , given that the current state of the mechanism is  $h^t$ . So, by  $K^t = \bigcup_{b^t \in H^t} K(h^t)$  we can denote the set of all possible decisions the mechanism may make in time period  $t$ . Then  $K = \bigcup_{t \in T} K^t$  is the set of all possible single-period decision our mechanism can make during its execution. An equivalent way of definition would have been  $K = \bigcup_{b \in H} K(b)$ . For every agent  $i$  we define a finite class  $\mathcal{L}_i = \{L_{i1}, L_{i2}, \dots, L_{im_i}\} \subseteq \mathcal{P}(K)$  of sets  $L_{ij} \subseteq K$  of single-period decisions. Any such set  $L_{ij}$  is called **interesting set** of agent  $i$  and its elements **interesting decisions**.

An agent has *single-valued preferences* if she has the same value  $r_i$ , which we will usually call **reward**, whenever any interesting decision (out of some interesting set) is made in some period  $t \in [a_i, d_i]$ , and has value for at most one such interesting decision. We can incorporate all this information using the valuation component and defining  $w_i = (r_i, L_{ij})$ , with  $w_i \in \mathbb{W}_i = \mathbb{R} \times \mathcal{L}_i$ . Finally, we assume that every agent  $i$  has a partial ordering  $\preceq_{L_i}$  on  $\mathcal{L}_i$ , i.e. upon her interesting sets  $L_{ij}$ . In this way we can make agent  $i$  report only one interesting set  $L_i$  (thus making lighter the notation from  $L_{ij}$  to  $L_i$ ), from now on called *the* interesting set of agent  $i$ , and consider agent  $i$  as being **satisfied** (i.e. receiving reward  $r_i$ ) whenever an interesting decision in  $L_i$  or in some interesting set  $L_{ij}$  of “grater importance” ( $L_i \preceq_{L_i} L_{ij}$ ) is made, while she is active<sup>2</sup>. More formally, whenever a decision

$$k^t \in \mathcal{L}_i(L_i) = \bigcup_{\substack{L_i \preceq_{L_i} L \\ L \in \mathcal{L}_i}} L,$$

for some  $t \in [a_i, d_i]$ . Now we are ready to give the formal definition:

**DEFINITION 5.1 (Single-valued online domains)** *A single-valued (preference) online domain is an online direct-revelation mechanism environment in which each agent  $i$  has (true) type  $\theta_i = (a_i, d_i, (r_i, L_i))$ , with reward  $r_i \in \mathbb{R}$  and interesting set  $L_i \in \mathcal{L}_i$ , and the valua-*

<sup>2</sup>Think of  $L_i$  as being a  $\preceq_{L_i}$ -minimal interesting set. Notice, however, that  $\preceq_{L_i}$  is only a *partial* ordering, thus  $L_i$  need not to be *the* minimal interesting set. Actually, is up to agent  $i$  which interesting set  $L_i \in \mathcal{L}_i$  she is going to report. Perhaps a more coherent way of presenting interesting sets would have been to consider  $\mathcal{L}_i$  as being a *language* for defining interesting sets for agent  $i$  and then define an element  $L_i \in \mathcal{L}_i$  as being the interesting set.

tion is defined by:

$$v_i(\theta_i, k) = \begin{cases} r_i, & \text{if } k^t \in \mathcal{L}_i(L_i) \text{ for some } t \in [a_i, d_i], \\ 0, & \text{otherwise.} \end{cases}$$

For the time being we will assume that for every agent  $i$  the family of sets of interesting decisions  $\mathcal{L}_i$ , its partial ordering  $\preceq_{L_i}$  and the *true* interesting set  $L_i$ , are all known to the mechanism<sup>3</sup>. In this way, the private information of an agent is restricted exactly to her arrival and departure times and her value for an interesting decision. This **known interesting-set assumption** saves us from having to include in our arguments misreports about the interesting set  $L_i$ . At this point, this makes our analysis more solid and clear. This assumption can be relaxed but we will be forced to introduce other conditions about the nature of our interesting sets, in order to assure that our results continue to hold. More details can be found in Parkes [2007, p. 429] and in Parkes and Duong [2007].

## 5.2 Monotonicity

Because in our single-valued domain environments we are primarily interested in *whether* or *not* an agent is satisfied and then in *when* or *how* she gets satisfied, we can think, for simplicity, of the decision policy  $x$  as being a binary function with respect to every agent  $i$  and define, in the spirit of Definition 5.1,

$$x_i(\theta_i, \theta_{-i}, \omega) = \begin{cases} 1, & \text{if } k^t \in \mathcal{L}_i(L_i) \text{ for some } t \in [a_i, d_i], \\ 0, & \text{otherwise,} \end{cases}$$

that is,  $x_i(\theta_i, \theta_{-i}, \omega) = 1$  if an interesting decision is made for agent  $i$  (in some period  $t \in [a_i, d_i]$ ) and  $x_i(\theta_i, \theta_{-i}, \omega) = 0$  if not (given type profile  $\theta$  and external stochastic events  $\omega$ ). Through this, we can express agent's  $i$  valuation simply by

$$v_i(\theta_i, k) = v_i(\theta_i, \pi(\hat{\theta}_i, \hat{\theta}_{-i}, \omega)) = x_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega) \cdot r_i,$$

<sup>3</sup>However, this does not imply that this information is *public*, i.e. known also to the other participating agents.

and so, the utility by

$$u_i(\theta_i, \hat{\theta}_i, \hat{\theta}_{-i}, \omega) = x_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega) \cdot r_i - p_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega).$$

Also, the **efficiency** of an online mechanism  $\mathcal{M}^{\mathcal{T}}$  with decision policy  $x$  in a single-valued domain can be thus expressed as

$$E_{\mathcal{M}^{\mathcal{T}}}(\hat{\theta}) = E_x(\hat{\theta}) = \sum_{i \in \mathcal{N}} x_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega) \cdot r_i. \quad (5.1)$$

Now we are ready to introduce a very important concept in online mechanism design (and especially online auctions):

**DEFINITION 5.2 (Critical value)** *Given an online mechanism with deterministic policy  $x$  in a single-valued domain, for every agent  $i$  with reported type  $\theta_i$ , types  $\theta_{-i}$  and external stochastic events  $\omega$ , we define her critical value*

$$v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) = \begin{cases} \inf \left\{ r'_i \mid x_i(\theta'_i, \theta_{-i}, \omega) = 1 \text{ where } \theta'_i = (a_i, d_i, (r'_i, L_i)) \right\}, & \text{if this exists,} \\ \infty, & \text{otherwise.} \end{cases}$$

In words, an agent's critical value is the smallest reward she can report and still receive an interesting decision, keeping everything else unchanged. However, this is not exactly accurate, and depending on whether the set  $\{r'_i \mid x_i(\theta'_i, \theta_{-i}, \omega) = 1\}$  in Definition 5.2 has a minimum or just an infimum. In the second case, reporting exactly the critical value does not result in an interesting decision, but increasing this reported reward arbitrarily little, an agent can receive an interesting decision. More precisely, immediately from how we define the infimum<sup>4</sup> of a set, we get

**COROLLARY 5.3** *Given an online mechanism on a single-valued online domain, for every agent  $i$  and  $\epsilon > 0$  there exists some  $\zeta$  with  $0 \leq \zeta < \epsilon$  such that agent  $i$  receives an interesting decision by reporting  $\theta_i = (a_i, d_i, (r_i, L_i))$  with*

$$r_i = v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) + \zeta.$$

There is also another, trivial remark we can make immediately from Definition 5.2:

<sup>4</sup>See, for example, [Rudin, 1976, Definition 1.8].

**COROLLARY 5.4** *Given an online mechanism on a single-valued online domain, for every agent  $i$ , type profile  $(\theta_i, \theta_{-i})$  and (external) stochastic events  $\omega$ ,*

$$x_i(\theta_i, \theta_{-i}, \omega) = 1 \implies r_i \geq v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega).$$

In words, if an agents receives an interesting decision then she must have reported at least her critical value (as her reward  $r_i$ ).

There is something important we must point out here. If we look with some attention at Definition 5.2, it is easy to see that the critical value of an agent is *independent* of the the agent's reported reward  $r_i$  as well as of other agents who are going to arrive after her own departure  $d_i$ , due to the online character of the mechanism. These will play major role in establishing truthfulness later at section 5.3.

For the following definition we need to have an ordering upon every agent's possible reported types<sup>5</sup>  $C_i$ . We define a partial<sup>6</sup> ordering

$$\theta_i \preceq_{\theta} \theta'_i \iff (a'_i \leq a_i) \wedge (d_i \leq d'_i) \wedge (r_i \leq r'_i) \wedge (L_i = L'_i),$$

for all types  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,  $\theta'_i = (a'_i, d'_i, (r'_i, L'_i)) \in C_i$ . However, we do not define in the “expected” way the shorthand  $\theta_i \prec_{\theta} \theta'_i \iff (\theta_i \preceq_{\theta} \theta'_i) \wedge (\theta_i \neq \theta'_i)$ , but instead

$$\begin{aligned} \theta_i \prec_{\theta} \theta'_i &\iff (\theta_i \preceq_{\theta} \theta'_i) \wedge (\theta_i \neq \theta'_i) \\ &\iff (a'_i \leq a_i) \wedge (d_i \leq d'_i) \wedge (r_i < r'_i) \wedge (L_i = L'_i). \end{aligned} \quad (5.2)$$

This definition will simplify substantially the presentation of our results in this project. We also say that an arrival-departure time interval  $[a'_i, d'_i]$  is **tighter** than an other interval  $[a_i, d_i]$ , if  $a_i \leq a'_i$  and  $d'_i \leq d_i$  or, equivalently, if  $[a'_i, d'_i] \subseteq [a_i, d_i]$ .

**DEFINITION 5.5 (Monotonic policy)** *A deterministic policy  $x$ , in a single-valued online domain, is called monotonic if, for every agent  $i$  and types  $\theta_i, \theta'_i \in C_i$  with  $\theta_i \prec_{\theta} \theta'_i$ ,*

$$x_i(\theta_i, \theta_{-i}, \omega) = 1 \implies x_i(\theta'_i, \theta_{-i}, \omega) = 1,$$

for every  $\theta_{-i} \in C_{-i}$ ,  $\omega \in \Omega$ .

<sup>5</sup>From now on we shall often write  $C_i$  instead of  $C(\theta_i)$  (and  $C_{-i}$  instead of  $C(\theta_{-i})$ ), if this is to keep our notation lighter.

<sup>6</sup>The fact that  $\preceq_{\theta}$  is a partial ordering on every  $C(\theta_i)$  is trivial, based on the usual ordering  $\leq$  of  $\mathbb{R}$ .

In words, in a monotonic decision policy, if an agent gets allocated<sup>7</sup> by reporting a type  $\theta_i$  then she will also be allocated if she reports a “better” type  $\theta'_i$ , that is, if she reports an earlier arrival time or a later departure time, coupled with a higher reward (see equation (5.2)). We have slightly deviate from the standard, less natural definition given in our references Parkes [2007], Hajiaghayi et al. [2005] and Parkes and Duong [2007]. We have done so by adapting cautiously the definition of the “strict” ordering  $\prec_\theta$  in the way. We feel that our exposition is more simple, thus capturing better the essence of monotonicity, overcoming obscuring technicalities.

**COROLLARY 5.6** *Given a (deterministic) monotonic decision policy  $x$ , for every agent  $i$  with type  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,*

$$r_i > v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \implies x_i(\theta_i, \theta_{-i}, \omega) = 1.$$

**PROOF** Let agent  $i$  have type  $\theta_i = (a_i, d_i, (r_i, L_i))$  with  $r_i > v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega)$ . Then, from Corollary 5.3, there exists some  $\zeta > 0$  such that agent  $i$  gets allocated by reporting type  $\theta'_i = (a_i, d_i, (r'_i, L_i))$  where

$$r_i > r'_i = v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) + \zeta > v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega).$$

Then, it is trivial to see that  $\theta'_i \prec_\theta \theta_i$  and thus, because  $x_i(\theta'_i, \theta_{-i}, \omega) = 1$  (agent  $i$  gets allocated by reporting  $\theta'_i$ ),  $x_i(\theta_i, \theta_{-i}, \omega) = 1$  due to monotonicity (see Definition 5.5).  $\square$

This simply means that monotonic mechanisms have the “nice” property to allocate every agent that places a any bid greater than her critical value. This displays a “canonical” behaviour of such mechanisms which will be the backbone for characterizing truthfulness at section 5.3. Notice here that if the infimum in Definition 5.2 is also a minimum (e.g. if  $\mathbb{W}_i$  is discrete) then it is trivial to show the stronger necessary condition

$$x_i(\theta_i, \theta_{-i}, \omega) = 1 \iff r_i \geq v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega).$$

For the next, we will also need a

<sup>7</sup>Due to the fact that when we study our single-valued domains we usually have in mind auction environments, sometimes we speak of *allocations* instead of interesting decisions. We also say that an agent is *allocated* instead of saying that she is satisfied.

**LEMMA 5.7** *Given a mechanism with a (deterministic) monotonic decision policy, for every agent  $i$  her critical value is a (weakly) increasing function with respect to tighter arrival-departure intervals. That is,*

$$[a'_i, d'_i] \subseteq [a_i, d_i] \implies v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq v_{(a'_i, d'_i, L_i)}^c(\theta_{-i}, \omega),$$

for every agent  $i$ , types  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,  $\theta'_i = (a'_i, d'_i, (r'_i, L'_i)) \in C_i$ ,  $\theta_{-i} \in C_{-i}$  and (external) stochastic events  $\omega$ .

**PROOF** We fix some agent  $i$ , types  $\theta_{-i}$  and stochastic events  $\omega$  and to arrive to a contradiction we assume that there exist types  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,  $\theta'_i = (a'_i, d'_i, (r'_i, L'_i)) \in C_i$  such that

$$[a'_i, d'_i] \subseteq [a_i, d_i] \quad \text{but} \quad v_{(a'_i, d'_i, L_i)}^c(\theta_{-i}, \omega) < v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega).$$

As we've noticed before, the critical values are independent of the rewards, so we are free to choose

$$v_{(a'_i, d'_i, L_i)}^c(\theta_{-i}, \omega) < r'_i < r_i < v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega). \quad (5.3)$$

In addition, from Corollary 5.3, this  $r'_i$  can be chosen to

$$x_i(\theta_i, \theta_{-i}, \omega) = 1$$

But we also know that  $a_i \leq a'_i$ ,  $d'_i \leq d_i$  and  $r'_i < r_i$ , from (5.3), thus  $\theta'_i \prec_\theta \theta_i$  and so, monotonicity of our decision policy (see Definition 5.5), gives  $x_i(\theta_i, \theta_{-i}, \omega) = 1$  which contradicts (5.3) through Definition 5.2.  $\square$

## 5.3 Truthfulness

**THEOREM 5.8 (sufficient condition)** *In any single-valued online domain with no early-arrivals and no late-departures, every (deterministic) monotonic decision policy  $x$  can be truthfully implemented, i.e. there is a payment policy  $p$  such that mechanism  $\mathcal{M}^{\mathcal{T}} = (x, \{p_i\}_{i \in \mathcal{N}})$  is truthful.*

**PROOF** Let  $x$  be the monotonic decision policy. Define a payment policy  $p$ :

$$p_i^t(h^t) = \begin{cases} v_{(\hat{a}_i, \hat{d}_i, L_i)}^c(\hat{\theta}_{-i}, \omega), & \text{if } x_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega) = 1 \wedge t = \hat{d}_i, \\ 0, & \text{otherwise,} \end{cases}$$

for every agent  $i$  with reported type  $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, (\hat{r}_i, L_i))$  and mechanism state  $h^t$  at time period  $t$ . This means that agent's  $i$  payment is

$$p_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega) = \sum_{t=\hat{a}_i}^{\hat{d}_i} p_i^t(h^t) = p_i^{\hat{d}_i}(h^{\hat{d}_i}) = \begin{cases} v_{(\hat{a}_i, \hat{d}_i, L_i)}^c(\hat{\theta}_{-i}, \omega), & \text{if } x_i(\hat{\theta}_i, \hat{\theta}_{-i}, \omega) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Simply, only allocated agents pay and only at their departure time, making critical-value payments. Now, we have to show that mechanism  $\mathcal{M}^{\mathcal{T}}$  is truthful.

Fix some agent  $i$ , types  $\theta_{-i}$  and (external) stochastic events  $\omega$ . Let  $\theta_i = (a_i, d_i, (r_i, L_i))$  be agent's  $i$  true type and  $\theta'_i = (a'_i, d'_i, (r'_i, L_i))$  a misreport of this type. Then, due to our limited misreports assumption,  $[a'_i, d'_i] \subseteq [a_i, d_i]$  and by Lemma 5.7,

$$v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq v_{(a'_i, d'_i, L_i)}^c(\theta_{-i}, \omega). \quad (5.4)$$

Depending on whether agent  $i$  receives an interesting decision or not by reporting her true type, there are two possible cases to analyze:

Case 1. If  $x_i(\theta_i, \theta_{-i}, \omega) = 1$ , agent  $i$  gets allocated,  $v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq r_i$  from Definition 5.2 and has utility

$$u_i(\theta_i, \theta_i, \theta_{-i}, \omega) = x_i(\theta_i, \theta_{-i}, \omega) \cdot r_i - p_i(\theta_i, \theta_{-i}, \omega) = r_i - v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \geq 0.$$

Then, by misreporting  $\theta'_i$ , either  $x_i(\theta'_i, \theta_{-i}, \omega) = 0$  and

$$u_i(\theta_i, \theta'_i, \theta_{-i}, \omega) = 0 - 0 = 0 \leq u_i(\theta_i, \theta_i, \theta_{-i}, \omega),$$

or  $x_i(\theta'_i, \theta_{-i}, \omega) = 1$  and

$$\begin{aligned} u_i(\theta_i, \theta'_i, \theta_{-i}, \omega) &= r_i - v_{(a'_i, d'_i, L_i)}^c(\theta_{-i}, \omega) \\ &\leq r_i - v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega), && \text{from (5.4),} \\ &= u_i(\theta_i, \theta_i, \theta_{-i}, \omega). \end{aligned}$$

Case 2. If  $x_i(\theta_i, \theta_{-i}, \omega) = 0$ , agent  $i$  is not allocated,  $v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq r_i$  from



Corollary 5.6 and has utility  $u_i(\theta_i, \theta_i, \theta_{-i}, \omega) = 0$ . Then, by misreporting  $\theta'_i$ , either  $x_i(\theta'_i, \theta_{-i}, \omega) = 0$  and

$$u_i(\theta_i, \theta'_i, \theta_{-i}, \omega) = 0 = u_i(\theta_i, \theta_i, \theta_{-i}, \omega),$$

or  $x_i(\theta'_i, \theta_{-i}, \omega) = 1$  and

$$\begin{aligned} u_i(\theta_i, \theta'_i, \theta_{-i}, \omega) &= r_i - v_{(a'_i, d'_i, L_i)}^c(\theta_{-i}, \omega) \\ &\leq r_i - v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega), && \text{from (5.4),} \\ &\leq 0 \\ &= u_i(\theta_i, \theta_i, \theta_{-i}, \omega). \end{aligned}$$

In any case, we showed that

$$u_i(\theta_i, \theta_i, \theta_{-i}, \omega) \geq u_i(\theta_i, \theta'_i, \theta_{-i}, \omega),$$

establishing truthfulness (from Definition 4.3). □

From the choice of payments in the constructive proof of the above theorem, we immediately get that

**COROLLARY 5.9** . *Every monotonic (online) mechanism that collects critical-value payments, is truthful.*

**LEMMA 5.10** *Given a truthful, IR online mechanism in a single-valued domain, every allocated agent's payment must be independent of her reported reward, i.e. for every agent  $i$  and types  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,  $\theta'_i = (a_i, d_i, (r'_i, L_i))$ ,*

$$x_i(\theta_i, \theta_{-i}, \omega) = x_i(\theta'_i, \theta_{-i}, \omega) = 1 \implies p_i(\theta_i, \theta_{-i}, \omega) = p_i(\theta'_i, \theta_{-i}, \omega).$$

**PROOF** To arrive to a contradiction, fix some agent  $i$ , types  $\theta_{-i}$  and stochastic events  $\omega$  and suppose that exist types  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,  $\theta'_i = (a_i, d_i, (r'_i, L_i))$  with  $r_i \neq r'_i$ , such that  $x_i(\theta_i, \theta_{-i}, \omega) = x_i(\theta'_i, \theta_{-i}, \omega) = 1$  but  $p_i(\theta_i, \theta_{-i}, \omega) \neq p_i(\theta'_i, \theta_{-i}, \omega)$ . Without loss of generality assume that

$$p_i(\theta'_i, \theta_{-i}, \omega) < p_i(\theta_i, \theta_{-i}, \omega).$$

Then, if agent's  $i$  true type were  $\theta_i$ ,

$$u_i(\theta_i, \theta_i, \theta_{-i}, \omega) = r_i - p_i(\theta_i, \theta_{-i}, \omega) < r_i - p_i(\theta'_i, \theta_{-i}, \omega) = u_i(\theta_i, \theta'_i, \theta_{-i}, \omega)$$

and she would be better off misreporting  $\theta'_i$ , contradicting truthfulness.  $\square$

**PROPOSITION 5.11 (Critical-value payment)** *In any single-valued online domain, every truthful, IR mechanism must collect, from each allocated agent, payment equal to her critical value, i.e.*

$$x(\theta_i, \theta_{-i}, \omega) = 1 \implies p_i(\theta_i, \theta_{-i}, \omega) = v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega),$$

for every agent  $i$ .

**PROOF** Fix some allocated agent  $i$  with (true) type  $\theta_i$ , types  $\theta_{-i}$  and stochastic events  $\omega$ . To get to a contradiction, suppose that  $p_i(\theta_i, \theta_{-i}, \omega) \neq v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega)$  and analyze the following cases:

Case 1.  $p_i(\theta_i, \theta_{-i}, \omega) < v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega)$ . If agent  $i$  had true type  $\theta'_i = (a_i, d_i, (r'_i, L_i))$  with

$$p_i(\theta_i, \theta_{-i}, \omega) < r'_i < v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega),$$

she wouldn't get allocated by reporting. But then, she could lie and misreport  $\theta'_i$  and get allocated with positive utility

$$u_i(\theta'_i, \theta_i, \theta_{-i}, \omega) = r'_i - p_i(\theta_i, \theta_{-i}, \omega) > 0,$$

contradicting truthfulness.

Case 2.  $v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) < p_i(\theta_i, \theta_{-i}, \omega)$ . From Corollary 5.3 we know that there is some  $r'_i$  with

$$v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq r'_i < p_i(\theta_i, \theta_{-i}, \omega),$$

such that agent  $i$  gets allocated by reporting  $\theta'_i = (a_i, d_i, (r'_i, L_i))$ . But then, if  $\theta'_i$  were her true type she would receive negative utility

$$\begin{aligned} u_i(\theta'_i, \theta'_i, \theta_{-i}, \omega) &= r'_i - p_i(\theta'_i, \theta_{-i}, \omega) \\ &= r'_i - p_i(\theta_i, \theta_{-i}, \omega), && \text{from Lemma 5.10,} \\ &< 0, \end{aligned}$$

contradicting IR. □

**DEFINITION 5.12 (No allocation - no payment)** *We say that a mechanism  $\mathcal{M}^{\mathcal{T}} = (x, \{p_i\}_{i \in \mathcal{N}})$  does not pay unallocated agents, if for every type profile  $\theta \in C$  and agent  $i$ ,*

$$x_i(\theta_i, \theta_{-i}, \omega) = 0 \implies p_i(\theta_i, \theta_{-i}, \omega) \geq 0.$$

Remember (page 50) that negative payments  $p_i < 0$  express payments made *from* the mechanism *to* agent  $i$ , which justifies the expression “the mechanism does not pay”. This assumption, together with IR, is sufficient to ensure us a desirable, canonical property of no-allocations at our single-valued environments:

**COROLLARY 5.13 (No allocation - no utility)** *Given an IR mechanism  $\mathcal{M}^{\mathcal{T}} = (x, \{p_i\}_{i \in \mathcal{N}})$ , in a single-valued online domain, that does not pay unallocated agents, for every agent  $i$ , type profile  $\theta \in C$  and (external) stochastic events  $\omega \in \Omega$ ,*

$$x_i(\theta_i, \theta_{-i}, \omega) = 0 \implies u_i(\theta_i, \theta_i, \theta_{-i}, \omega) = 0,$$

*i.e. non-allocated agents have zero utility.*

**PROOF** If  $\pi_i(\theta_i, \theta_{-i}, \omega) = 0$ , from Definition 5.1 we have  $u_i(\theta_i, \theta_i, \theta_{-i}, \omega) = 0 - p_i(\theta_i, \theta_{-i}, \omega) = -p_i(\theta_i, \theta_{-i}, \omega)$ , thus  $u_i(\theta_i, \theta_i, \theta_{-i}, \omega) \leq 0$  by our no allocation - no payment assumption (Definition 5.12). This, together with IR (equation (4.2)), gives us the desired  $u_i(\theta_i, \theta_i, \theta_{-i}, \omega) = 0$ . □

**THEOREM 5.14 (necessary condition)** *In any single-valued online domain with no early-arrivals and no late-departures, every truthful, IR mechanism that does not pay unallocated agents must have a monotonic decision policy.*

**PROOF** Fix some agent  $i$ , types  $\theta_{-i}$  and stochastic events  $\omega$ . To get to a contradiction, suppose that there exist types  $\theta_i = (a_i, d_i, (r_i, L_i))$ ,  $\theta'_i = (a'_i, d'_i, (r'_i, L'_i))$  such that  $\theta_i \prec_{\theta} \theta'_i$  and  $x_i(\theta_i, \theta_{-i}, \omega) > 0$  but  $x_i(\theta'_i, \theta_{-i}, \omega) = 0$ . Then

$$a'_i \leq a_i, \quad d_i \leq d'_i, \quad L_i = L'_i, \quad r_i < r'_i$$

and  $v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq r_i$ , thus  $v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) \leq r_i < r'_i$ . Choose a reward  $\widehat{r}_i$  such that

$$v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega) < \widehat{r}_i < r'_i \quad \text{and} \quad x_i(\widehat{\theta}_i, \theta_{-i}, \omega) = 1, \quad (5.5)$$

where  $\hat{\theta}_i$  is a new type  $\hat{\theta}_i = (a_i, d_i, (\hat{r}_i, L_i))$ . We can do that, by Corollary 5.3 . If  $\theta'_i$  was agent's  $i$  true type, she would not get allocated by reporting it (we have assumed  $x(\theta'_i, \theta_{-i}, \omega) = 0$ ), thus having zero utility  $u_i(\theta_i, \theta'_i, \theta_{-i}, \omega) = 0$  (Corollary 5.13). But then she could do better by misreporting  $\hat{\theta}_i$ , getting allocated (from (5.5)) and achieving utility

$$\begin{aligned} u_i(\theta_i, \hat{\theta}_i, \theta_{-i}, \omega) &= r'_i - p_i(\hat{\theta}_i, \theta_{-i}, \omega) \\ &= r'_i - v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega), && \text{from Proposition 5.11,} \\ &> 0, && \text{from (5.5),} \end{aligned}$$

contradicting truthfulness. □

Single-valued domains as well as monotonicity and all the characterization results that deploy it in this Chapter 5 can be easily applied in classic (offline) Mechanism Design, providing an important addition to the only characterization tool Proposition 2.13 (page 32) we had so far for the offline setting. For an exposition in the limits of offline Mechanism Design, see [Nisan, 2007, section 9.5.4].

## **Part C**

# **Specific Online Auctions**



# Chapter 6

## Expiring Items Auctions

Consider the following problem setting: We have a dynamic, direct-revelation mechanism environment (in the way we have already defined it in Chapter 4) where agents  $\mathcal{N}$  arrive over time and each makes a single report  $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{w}_i)$  about her type, upon her arrival time  $\hat{a}_i$ . Also, we have a *single, re-usable* item to allocate at each time period  $t \in \mathcal{T}$  (to some agent  $i \in \mathcal{N}$ ). This makes our environment essentially an auction environment and that is why the valuation component  $w_i$  of player's  $i$  type is called her **bid** and our agents are also referred to as bidders. The very important fact about this auction setting is that we further assume that every agent is interested in being allocated with *one* instance of the item (at some time period while she is active) and does not value more the allocation of more instances. Furthermore, naturally enough, every agent has zero value for the receipt of no items. From the above, it is easy to see that the auction environment we just described is a single-valued online domain (see Chapter 5), with the interesting set of each bidder being the set of decisions that allocate at least one instance to her.

Due to the fact that interesting sets are here defined in such a clear and easy way, it is common practice to remove the  $L_i$  component from the formal expression  $\theta_i = (a_i, d_i, (L_i, r_i))$  of agents' types in single-valued domains, and leaving bid  $w_i$  to essentially represent only bidder's  $i$  reward  $r_i$  (value for getting allocated). From now on we will refer only to bids  $w_i$  and by that we will mean exactly this value  $r_i$ . For a further comment on notation, since in this environment we do not allow for any *external* stochastic events, parameter  $\omega$  will be removed from all our valuation expressions. Taking all these into consideration, we will denote agent's  $i$  critical value (recall Definition 5.2, page 62) simply by  $v_{(a_i, d_i)}^c(\theta_{-i})$  (instead of the more involved  $v_{(a_i, d_i, L_i)}^c(\theta_{-i}, \omega)$ ) and refer to it as **critical bid**, due to the auction interpretation of our problem setting.

A mechanism  $\mathcal{A} = (x, \{p_i\}_{i \in \mathcal{N}})$  in our problem setting, from now on called *auction*, receives its input *online* as a type profile  $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N)$  and dynamically defines:

- Which agent  $i$  gets allocated at every time period  $t$ , shown by the *decision policy*

$$x^t(\hat{\theta}) = i.$$

If no agent gets allocated at time  $t$  then we define  $x^t(\hat{\theta}) = 0$ . Due to the online character of our auctions, it is obvious that each decision  $x^t(\hat{\theta})$  must be made before the end of period  $t$  and taking into consideration *only* types  $\hat{\theta}_i$  with  $(\hat{a}_i \leq t)$ , i.e. of agents that have arrived until current time period  $t$ . No knowledge of future types is possible.

- What each agent  $i$  has to pay, shown by the *payment policy*

$$p_i(\hat{\theta}_i, \hat{\theta}_{-i}) \in \mathbb{R}_{\geq 0},$$

here adopting the no-deficit principle (see page 50). Payment  $p_i(\hat{\theta}_i, \hat{\theta}_{-i})$  must be collected from agent  $i$  before  $d_i$  when she leaves the auction.

It is easy to see that auction  $A$  is essentially an online algorithm with input  $\hat{\theta}$ . By determining an objective function we can define an optimization problem and apply our competitive analysis techniques and notions from Chapter 3 to study our auction problems. Here, the goal of the “mechanism designer” (a.k.a. auctioneer) in our problem setting would be to design an auction  $\mathcal{A} = (x, \{p_i\}_{i \in \mathcal{N}})$  with the maximum possible *efficiency* (see Definition 2.14, page 33 and (5.1), page 62), i.e. to *maximize*

$$E_{\mathcal{A}}(\hat{\theta}) = E_x(\hat{\theta}) = \sum_{i \in \mathcal{N}} x_i(\hat{\theta}_i, \hat{\theta}_{-i}) \cdot w_i, \quad (6.1)$$

for all possible type profile reports  $\hat{\theta}$ .

Finally note a fine point: since  $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N)$  is revealed online, our auction does not know the size  $N$  of the bidders’ space  $\mathcal{N}$  nor for how long it is going to last, until our adversary chooses to end it.

**DEFINITION 6.1 (CEI)** *We will refer to the above optimization (maximization) problem setting, coupled with the no early-arrivals and no late-departures assumptions<sup>1</sup>, as the Canonical Expiring-Items (CEI) problem.*

---

<sup>1</sup>See section 4.2



## 6.1 The Greedy Auction

Now, we are ready to construct a specific online auction for our CEI problem:

**DEFINITION 6.2 (GREEDY AUCTION)** *The online Greedy Auction is an auction for the CEI problem defined by:*

- *Decision policy: at every time point  $t$ , allocate an item to the unassigned (unallocated so far) agent with the highest bid. Break ties randomly.*
- *Payment policy: Every agent  $i$  that gets allocated, pays her critical bid  $v_{(a_i, d_i)}^c(\theta_{-i})$  upon her departure  $d_i$ . Non-allocated agents make no payments.*

*The name “greedy” is easy to justify, since this auction allocates “myopically” to maximize efficiency. Note also that if all agents are impatient, i.e.  $a_i = d_i$ , then Greedy Auction is a sequence of Vickrey (second-price) auctions (see page 29).*

**THEOREM 6.3** *The Greedy Auction is IR and strongly truthful in the CEI environment*

**PROOF** For IR, if an agent  $i$  with type  $\theta_i = (a_i, d_i, w_i)$  is not allocated by the Greedy Auction then makes no payment,  $p_i(\theta_i, \theta_{-i}) = 0$  and has utility  $u_i(\theta_i, \theta_i, \theta_{-i}) = 0 \cdot w_i - 0 = 0$ . If she gets allocated, then pays  $p_i(\theta_i, \theta_{-i}) = v_{(a_i, d_i)}^c(\theta_{-i})$  and has utility  $u_i(\theta_i, \theta_i, \theta_{-i}) = w_i - v_{(a_i, d_i)}^c(\theta_{-i}) \geq 0$ , because  $w_i \geq v_{(a_i, d_i)}^c(\theta_{-i})$ , immediately from Corollary 5.4. In any case,  $u_i(\theta_i, \theta_i, \theta_{-i}) \geq 0$ .

For truthfulness, let  $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{w}_i)$  be a misreport of the true type  $\theta_i = (a_i, d_i, w_i)$  of agent  $i$ , which gets agent  $i$  allocated. Then, its not difficult to see that

$$v_{(a_i, d_i)}^c(\theta_{-i}) \leq v_{(\hat{a}_i, \hat{d}_i)}^c(\theta_{-i}). \quad (6.2)$$

Due to no early-arrivals and no late-departures,  $[\hat{a}_i, \hat{d}_i] \subseteq [a_i, d_i]$  and from the definition of Greedy Auction, if a bid gets agent  $i$  allocated some time in the tighter time interval  $[\hat{a}_i, \hat{d}_i]$ , so will in the wider  $[a_i, d_i]$ , which proves equation (6.2). If agent  $i$  is not allocated by reporting true type  $\theta_i = (a_i, d_i, w_i)$ , this means that  $w_i < v_{(a_i, d_i)}^c(\theta_{-i})$  and by (6.2),  $w_i < v_{(\hat{a}_i, \hat{d}_i)}^c(\theta_{-i})$  and thus

$$u_i(\theta_i, \hat{\theta}_i, \theta_{-i}) = w_i - v_{(\hat{a}_i, \hat{d}_i)}^c(\theta_{-i}) < 0,$$

which is not acceptable, because of IR. If, on the other hand, telling the truth gets agent  $i$  allocated,

$$u_i(\theta_i, \theta_i, \theta_{-i}) = w_i - v_{(a_i, d_i)}^c(\theta_{-i}) \stackrel{(6.2)}{\geq} w_i - v_{(\hat{a}_i, \hat{d}_i)}^c(\theta_{-i}) = u_i(\theta_i, \hat{\theta}_i, \theta_{-i}),$$

which proves truthfulness.

Alternatively, one could much more easily prove that the Greedy Auction is monotonic and use Corollary 5.9 (page 67).  $\square$

## 6.2 Upper Bound

We now turn to study the performance of the Greedy Auction, in terms of competitive analysis.

**THEOREM 6.4** *The Greedy Auction is 2-competitive for the CEI problem.*

**PROOF** Let  $x$  be the Greedy Auction allocation,  $x^*$  an optimal offline allocation and some adversary chooses input types  $\theta$ . Define

$$A = \{\text{all agents allocated by } x\} = \{x^t(\theta) \mid t \in \mathcal{T}\}$$

$$B = \{\text{all agents allocated by both } x \text{ and } x^*\} = \{x^t(\theta) \mid t \in \mathcal{T}\} \cup \{(x^*)^t(\theta) \mid t \in \mathcal{T}\}$$

$$C = \{\text{all agents allocated by } x^* \text{ only}\} = \{(x^*)^t(\theta) \mid t \in \mathcal{T}\} \setminus \{x^t(\theta) \mid t \in \mathcal{T}\}$$

Using this, we can express the efficiencies of our auctions as

$$E_x(\theta) = \sum_{i \in A} w_i,$$

$$E_{x^*}(\theta) = \sum_{i \in B} w_i + \sum_{i \in C} w_i.$$

Now, notice that  $B \subseteq A$  which (using here that  $w_i > 0$  for every agent  $i$ ) gives

$$\sum_{i \in B} w_i \leq \sum_{i \in A} w_i. \tag{6.3}$$

Next, consider an agent  $i \in C$  allocated only by  $x^*$  at some time period  $t_i$ . At this period  $t_i$ , online  $x$  allocates to some other agent  $j_i \in A$  for which  $w_i \leq w_{j_i}$  (otherwise  $\pi$

would have allocated  $i$  instead of  $j_i$ ). This gives us

$$\sum_{i \in C} w_i \leq \sum_{i \in C} w_{j_i} \leq \sum_{j \in A} w_j. \quad (6.4)$$

By using equations (6.3) and (6.4) we take

$$E_{x^*}(\theta) = \sum_{i \in B} w_i + \sum_{i \in C} w_i \leq V(\pi^*(\theta)) = \sum_{i \in A} w_i + \sum_{i \in A} w_i = 2 \cdot E_x(\theta)$$

The above analysis shows that

$$\text{CR}_x^{\text{CEI}} = \max_{\theta} \frac{E_{x^*}(\theta)}{E_x(\theta)} \leq 2$$

which means that Greedy Auction is indeed 2-competitive.  $\square$

This results gives us immediately an upper bound for the competitive ratio of our general problem.

**COROLLARY 6.5** *For the CEI problem,  $\text{CR}^{\text{CEI}} \leq 2$*

## 6.3 Lower Bound

In this section, we prove a lower bound for our general CEI problem, showing also that the upper bound of the previous section is tight. This also proves optimality of the Greedy Auction.

**THEOREM 6.6** *No truthful, IR online auction can be  $(2 - \epsilon)$ -competitive for the CEI problem, for every  $0 < \epsilon < 1$ .*

**PROOF** Let  $\mathcal{A}$  be an online auction and  $\mathcal{A}^*$  be an optimal offline. Fix random  $0 < \epsilon < 1$ . In a first scenario, assume that online  $\mathcal{A}$  is given input only 2 agents  $i = 1, 2$  with types

$$\theta_1 = (1, 1, w(1 + \delta)) \quad \text{and} \quad \theta_2 = (1, 2, w) \quad (\text{scenario I})$$

for some  $w \in \mathbb{R}_{>0}$  and  $0 < \delta < \frac{\epsilon}{1 - \epsilon}$ . Offline  $\mathcal{A}^*$  obviously allocates both agents, agent 1 at  $t = 1$  and agent 2 at  $t = 2$ , having efficiency  $E_{x^*}(\theta) = w(1 + \delta) + w = w(2 + \delta)$ . Online  $\mathcal{A}$  has to also allocate both, otherwise its efficiency would be  $E_x(\theta) \leq w(1 + \delta)$ , giving

$$\text{CR}_{\mathcal{A}}^{\text{CEI}} \geq \frac{w(2 + \delta)}{w(1 + \delta)} > 2 - \epsilon,$$

and finishing the proof of our theorem. In addition, agent 1 has positive utility. For this we need to show that  $v_{(1,1)}^c(\theta_{-1}) < w(1 + \delta)$ . It is enough to show that  $v_{(1,1)}^c(\theta_{-1}) \leq w$ . To arrive to a contradiction, suppose that  $v_{(1,1)}^c(\theta_{-1}) > w$ . Then, because  $v_{(1,1)}^c(\theta_{-1}) = v_{(1,1)}^c(\theta_2)$  is independent of  $\delta$ , so we can choose  $\delta$  small enough to  $v_{(1,1)}^c(\theta_{-1}) > w(1 + \delta) > w$ , which contradicts the definition of critical bid and the fact that agent 1 is allocated. So, in this scenario

agent 1 is allocated at  $t = 1$  (with positive utility)  
agent 2 is allocated at  $t = 2$

In a second scenario, we “reverse” the types of our two agents and give input

$$\theta'_1 = (1, 2, w(1 + \delta)) \quad \text{and} \quad \theta'_2 = (1, 1, w) \quad (\text{scenario II})$$

Like before, at scenario I, both agents must be allocated by online  $\mathcal{A}$ . In addition, we will show that agent’s 2 utility is positive. For this we have to show that  $w > v_{(1,1)}^c(\theta'_{-2})$ . We already know that  $w \geq v_{(1,1)}^c(\theta'_{-2})$ , because of the definition of critical bid. If  $w = v_{(1,1)}^c(\theta'_{-2})$ , we choose some  $\alpha > 1$  and replace *only* agent’s 2 bid with  $w' = \alpha w > w$  in both scenarios I and II, i.e. we redefine

$$\theta_1 = (1, 1, w(1 + \delta)) \quad \text{and} \quad \theta_2 = (1, 2, w') \quad (\text{scenario I})$$

$$\theta'_1 = (1, 2, w(1 + \delta)) \quad \text{and} \quad \theta'_2 = (1, 1, w') \quad (\text{scenario II})$$

If we choose  $\alpha$  arbitrarily close to 1 so that  $\alpha w < w + \delta$ , nothing changes in our results so far and we can repeat it for the new input. That means, w.l.o.g. we can assume that  $w > v_{(1,1)}^c(\theta'_{-2})$ . So, in this scenario we have

agent 1 is allocated at  $t = 2$   
agent 2 is allocated at  $t = 1$  (with positive utility)

Finally, in a third scenario, an adversary chooses as combination of scenarios I and II, together with a third agent, giving as input

$$\theta'_1 = (1, 2, w(1 + \delta)) \quad \text{and} \quad \theta_2 = (1, 2, w') \quad \text{and} \quad \theta_3 = (2, 2, M) \quad (\text{scenario III})$$

with  $M$  arbitrarily large. Online auction  $\mathcal{A}$  must decide who to allocate at  $t = 1$  without knowing the existence of agent 3 who’s going to arrive later at  $t = 2$ . This means

that only agent 1 or 2 can be allocated at  $t = 1$ . Furthermore, at  $t = 3$  auction  $\mathcal{A}$  must allocate the new agent with arbitrarily large bid  $M$ , otherwise its competitive ratio is unbounded and our proof is complete. We now can conclude that the allocations in this final scenario will be

$$\begin{aligned} \text{agent 1 is allocated at } t = 1 & & (*) \\ \text{agent 3 is allocated at } t = 2, & \end{aligned}$$

because if agent 2 is allocated at  $t = 1$  instead of agent 1, then agent 1 would have been better to misreport type  $\theta_1 = (1, 1, w(1 + \delta))$  instead of  $\theta'_1 = (1, 2, w(1 + \delta))$ . Then, at  $t = 1$  we would have been exactly like in scenario I, thus agent  $i$  would have been allocated with positive utility. But this is not acceptable since it contradicts truthfulness. However, allocation policy  $(*)$ , is also not acceptable, because in a similar way agent 2 (which is now unallocated) will be better misreporting  $\theta'_2 = (1, 1, w)$ , leading us to scenario II and getting allocated with positive utility, again giving us a contradiction to truthfulness.  $\square$

As an immediate consequence, taking into consideration Corollary 6.5, we have an exact competitive ratio for our general problem

**THEOREM 6.7** *For our CEI problem,*

$$\text{CR}^{\text{CEI}} = 2.$$

By this, Theorem 6.4 shows that Greedy Auction is optimal for our CEI problem.

## 6.4 An Impossibility Result

As we saw in page 55, the no late-departures is not as natural as the no early-arrivals. However, relaxing it would lead us to the disastrous:

**THEOREM 6.8 (IMPOSSIBILITY THEOREM)** *No truthful, IR online auction has constant competitive ratio in the CEI environment, if we relax the no late-departures assumption and allow arbitrary misreports of departure.*

**PROOF** Let  $\mathcal{A}$  be an online auction,  $\mathcal{A}^*$  be an optimal offline and  $M$  some arbitrarily large integer. An adversary chooses time horizon  $t = 1, 2, \dots, M$  and  $M$  agents with

types

$$\theta_1 = (1, M, w_1), \theta_2 = (1, M, w_2), \dots, \theta_M = (1, M, w_m)$$

where  $w_1, w_2, \dots, w_M \in (q, q + \delta)$  for some  $q \in \mathbb{R}_{>0}$  and  $\delta > 0$  arbitrarily small. Obviously, the optimal  $\mathcal{A}^*$  allocates all agents (we do not care in what order) and has efficiency

$$E_{\mathcal{A}^*}(\theta) = \sum_{i=1}^M w_i > \sum_{i=1}^M q = M \cdot q.$$

First of all, we can assume that every allocated (by  $\mathcal{A}$ ) agent  $i = 1, 2, \dots, M$  has positive utility, i.e.

$$w_i > v_{(1,M)}^c(\theta_{-i}). \quad (6.5)$$

Otherwise, for every allocated agent  $i$  with  $w_i = v_{(1,M)}^c(\theta_{-i})$ , replace  $w_i$  with  $w'_i = v_{(1,M)}^c(\theta_{-i}) + \zeta$ , for some small  $\zeta$  such that still  $w'_i \in (q, q + \delta)$ , keeping everything else  $\theta_{-i}$  fixed. Then, agent  $i$  still gets allocated with her new type  $(1, M, w'_i)$  (Corollary 5.3).

Next will show that for every agent  $i$

$$v_{(1,M)}^c(\theta_{-i}) = v_{(1,1)}^c(\theta_{-i}). \quad (6.6)$$

Suppose that  $v_{(1,M)}^c(\theta_{-i}) < v_{(1,1)}^c(\theta_{-i})$ . Fix  $\theta_{-i}$  and replace agent's  $i$  type with

$$\theta'_i = (1, 1, w'_i) \quad \text{where } w'_i = v_{(1,1)}^c(\theta_{-i}) + \epsilon > v_{(1,M)}^c(\theta_{-i})$$

In this scenario, with proper selection of  $\epsilon$  (Corollary 5.3) agent  $i$  gets allocated with payment  $p_i(\theta'_i, \theta_{-i}) = v_{(1,1)}^c(\theta_{-i})$  (Proposition 5.11) and utility

$$u_i(\theta'_i, \theta'_i, \theta_{-i}) = w'_i - v_{(1,1)}^c(\theta_{-i})$$

But if he misreports (arbitrary departure misreport,  $1 < M$ ) type  $\theta_i = (1, M, w_i)$  then he can do better

$$u_i(\theta'_i, \theta_i, \theta_{-i}) = w_i - v_{(1,M)}^c(\theta_{-i}) > w_i - v_{(1,M)}^c(\theta_{-i}) = u_i(\theta'_i, \theta'_i, \theta_{-i}),$$

which is absurd since our auction is truthful. With a similar (actually even easier) argument we can show that  $v_{(1,1)}^c(\theta_{-i}) < v_{(1,M)}^c(\theta_{-i})$  is also impossible, which proves our case for  $v_{(1,M)}^c(\theta_{-i}) = v_{(1,1)}^c(\theta_{-i})$ .

For the main step of our proof, to arrive to contradiction, suppose that some agent

$k$  with type  $\theta_k = (1, M, w_k)$  gets allocated at some time period  $t_k > 1$ . Then, consider a scenario in which our auction is supplied dynamically with  $M - 1$  additional types

$$(t, t, \beta^{2t-3}), \quad t = 2, 3, \dots, M$$

where  $\beta > q + \delta$  is arbitrarily large. Both auction  $\mathcal{A}$  and  $\mathcal{A}^*$  must allocate to one of the initial agents with types  $(1, M, w_i)$  at time period  $t = 1$ , because none of the new agents is “active” yet. To stay constant competitive in this scenario, online  $A$  has to allocate all new agents with types  $(t, t, \beta^{t-1})$ , at consecutive time periods  $t = 2, 3, \dots, M$ . Indeed, if  $t^* > 1$  is the first time period at which  $A$  does not allocate new agent  $(t^*, t^*, \beta^{t^*-1})$  (allocating some with type  $(1, M, w_i)$  instead), we stop the supply of the next new agents with types  $(t, t, \beta^{t-1})$ ,  $t = t^*, \dots, M$  and after time period  $t^*$  only agents with types  $(1, M, w_i)$  can be allocated. Then, the efficiency of  $\mathcal{A}$  would be

$$\begin{aligned} E_{\mathcal{A}}(\theta) &< (q + \delta) + \sum_{t=2}^{t^*-1} \beta^{t-1} + (M - t^* + 1)(q + \delta) \\ &= (M + 2 - t^*)(q + \delta) + \beta \frac{\beta^{t^*-2} - 1}{\beta - 1} \\ &= (M + 2 - t^*)(q + \delta) + \frac{\beta^{t^*-1} - \beta}{\beta - 1}, \end{aligned}$$

while the optimal is

$$\begin{aligned} E_{\mathcal{A}^*}(\theta) &> q + \sum_{t=2}^{t^*} \beta^{t-1} + (M - t^*)q \\ &= (M + 1 - t^*)q + \beta \frac{\beta^{t^*-1} - 1}{\beta - 1} \\ &= (M + 1 - t^*)q + \frac{\beta^{t^*} - \beta}{\beta - 1}, \end{aligned}$$

thus the competitive ratio would be

$$\begin{aligned}
 \text{CR}_{\mathcal{A}} &> \frac{E_{\mathcal{A}^*}(\theta)}{E_{\mathcal{A}}(\pi(\theta))} = \frac{(M+1-t^*)q + \frac{\beta^{t^*}-\beta}{\beta-1}}{(M+2-t^*)(q+\delta) + \frac{\beta^{t^*-1}-\beta}{\beta-1}} \\
 &= \frac{(M+1-t^*)q(\beta-1) + \beta^{t^*} - \beta}{(M+2-t^*)(q+\delta)(\beta-1) + \beta^{t^*-1} - \beta} \\
 &= \frac{\beta^{t^*} + [(M+1-t^*)q-1] \cdot \beta - (M+1-t^*)q}{\beta^{t^*-1} + [(M+2-t^*)(q+\delta)-1] \cdot \beta - (M+2-t^*)(q+\delta)} \rightarrow \infty,
 \end{aligned}$$

as  $\beta \rightarrow \infty$ . So, in this scenario, agent  $j$  is not allocated (0 utility) and would be better to misreport type

$$\theta'_j = (1, 1, w'_j) \quad \text{where } w'_j = v_{(1,1)}^c(\theta_{-j}) + \epsilon,$$

keeping everything else  $\theta_{-j}$  fixed, and get allocated (Corollary 5.3) with positive utility

$$\begin{aligned}
 u_i(\theta_j, \theta'_j, \theta_{-j}) &= w_i - v_{(1,1)}^c(\theta_{-j}) \\
 &> v_{(1,M)}^c(\theta_{-j}) - v_{(1,1)}^c(\theta_{-j}), && \text{from (6.5)} \\
 &= v_{(1,1)}^c(\theta_{-j}) - v_{(1,1)}^c(\theta_{-j}), && \text{from (6.6)} \\
 &= 0.
 \end{aligned}$$

But this is not possible since our auction is truthful, proving that the online  $\mathcal{A}$  does not allocate agents at  $t > 1$ . Thus,  $\mathcal{A}$  allocates to at most one agent, at  $t = 1$ , having efficiency

$$E_{\mathcal{A}}(\theta) \leq \max_{i=1,2,\dots,M} w_i < q + \delta.$$

This would be sufficient to complete the proof of our theorem, because

$$\text{CR}_{\mathcal{A}} \geq \frac{E_{\mathcal{A}^*}(\theta)}{E_{\mathcal{A}}(\theta)} > \frac{Mq}{q+\delta} \rightarrow M, \quad \text{as } \delta \rightarrow 0,$$

which shows that the competitive ratio is unbounded ( $M$  is arbitrarily large).  $\square$

This proof, taken again from Hajiaghayi et al. [2005] is an adaptation of a similar result of Lavi and Nisan [2005].



## 6.5 Extensions – Open Problems

The most natural extension to make to our CEI model is that of allowing for  $k > 1$  reusable (identical) items. Fortunately enough, the competitive ratio of 2 continues to hold for this general scenario and is achieved by the greedy auction which allocates the  $k$ -highest bidding, unallocated agents. For more see [Hajiaghayi et al., 2005, 4.3]. Also, one can consider a continuous-time (asynchronous) case. Under certain assumptions about that “tricky” model, [Hajiaghayi et al., 2005, 4.2] show a competitive ratio of 5. Finally, if we let aside our social sensibilities for a while and relax the demand for truthful mechanism designing, we can show a lower bound of  $\phi \approx 1.618$ , the golden ratio, for our problem (see [Hajek, 2001]).

As far as open problems are concerned, the main challenge is to come up with non-trivial randomization ideas, possibly improving our lower bounds. Also, there are some questions with respect to the competitive ratio with respect to *revenue*. Generally, in this case we use the ratio  $h = \frac{a}{b}$  (where  $a$  and  $b$  are the highest and lower, respectively, bids submitted) as our competitive ratio’s parameter.



# Chapter 7

## Adaptive, Limited-Supply Auctions

Instead of having a reusable good to be allocated at every time period  $t \in \mathcal{T}$ , now consider having only *one* instance of a *single, indivisible* item to allocate to only *one* agent during the auction's execution. As we did in the case of the Canonical Expiring Items (CEI) problem (Chapter 6), we can easily see that this auction setting is a single-valued online domain and thus, the notational conventions and the discussion carried out during the introduction of the CEI environment at pages 73–74, can (and will) be adopted also here, in the natural way one would expect.

However, there is a very important point at which we are going to deviate substantially from our exposition in Chapter 6. To analyse the performance of an online auction for the CIE problem we used the standard competitive analysis approach of competing against an optimal, offline auction algorithm that knows in advance the input type profile  $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N)$  of our players' reports. That means, we let our adversary choose the number  $N$  of agents, their bids  $w_i$  as well as the order in which they are going to arrive (by selecting the  $a_i$ 's) and for how long they are going to stay active (by selecting the  $d_i$ 's). On one hand, such a powerful adversarial model guarantees the competitiveness of our algorithms in the worst possible scenario, but on the other hand it may be trivially inappropriate to model our problem settings, as it's the case for our current, one item limited-supply environment. If, in our setting, the adversary could determine the size of our bidders set  $\mathcal{N}$  and the order in which they arrive, then he would wait until we allocate the item to some agent and then he would right after insert a new agent in the auction, one with a bid arbitrarily many times that of the agent's we just allocated. We can not give the item to the new agent, since our item is already sold. Remember that the bid equals the value for an allocation and thus, the efficiency of our auction equals the bid of the one allocated agent. An optimal offline algorithm would have allocated the agent with the huge bid, achieving an efficiency arbitrarily many times that of our auction's. The above

analysis shows us that the competitive analysis framework of Chapter 6 trivially produces infinite competitive ratios for every possible auction one may construct in our current setting.

That is why, we need to define a weaker adversarial model, yet natural enough for our competitiveness results to make sense. Towards this, we adopt the **random-ordering hypothesis**, i.e. we assume that our adversary can choose agents' bids  $w_i$  but has no control on the order on which they are going to arrive, nor can he change online the number  $N$  of the arriving agents, which is known in advance to the online auction. To be more specific, we suppose that every online algorithm knows  $N$  and the time frame  $T$ ,  $\mathcal{T} = \{1, 2, \dots, T\}$ , and that the adversary chooses a set of  $N$  bids  $W_N = \{w_j\}_{j=1}^N$  and a set of arrival-departure intervals  $I_N = \{[a_k, d_k]\}_{k=1}^N$  but he has no control over which bid  $w_{j_i}$  will be matched to arrival-departure  $[a_{k_i}, d_{k_i}]$  in order to construct a type  $\theta_i = (a_{k_i}, d_{k_i}, w_{j_i})$ ,  $i = 1, 2, \dots, N$ . Every such type  $\theta_i$  is constructed by choosing randomly  $w_{j_i}$  and  $(a_{k_i}, d_{k_i})$ . Formally,  $w_i$  and  $(a_{k_i}, d_{k_i})$  are picked *uniformly at random* (and without replacement) from  $W_N$  and  $I_N$ , respectively, for every  $i = 1, 2, \dots, N$ .

In addition to our usual objective function when auction problems are concerned, i.e. that of efficiency (see expression (6.1), page 74), here we will also consider the auction's **revenue** (recall Definition 2.15, page 34)

$$R_{\mathcal{A}}(\theta) = \sum_{i=1}^N p_i(\theta_i, \theta_{-i}),$$

as an alternative measure of performance for our online auctions  $\mathcal{A}$  on an input (type profile)  $\theta = (\theta_1, \theta_2, \dots, \theta_N)$ . It goes without saying that the goal is to **maximize** revenue. However, this carries a small implication. Instead of comparing our auction's revenue  $R_{\mathcal{A}}$  to that of an offline optimal auction  $\mathcal{A}^*$ , which will greedily allocated the highest bidding agent, i.e.<sup>1</sup>  $R_{\mathcal{A}^*}(\theta) = w_{(1)}$ , we will use (offline) Vickrey (second-price) auction's revenue as a benchmark, i.e.  $R_{\mathcal{V}}(\theta) = w_{(2)}$ . The main reason for doing this is that truthfulness is a major priority in Mechanism Design and so we would like to compete against optimal algorithms that respect this property. As we know (page 29), the Vickrey auction is optimal among truthful auctions as far as revenue is concerned. Also it is the optimal offline auction with respect to efficiency (even among non-truthful auctions) and so it is essentially the benchmark we have been already using in the calculation of efficiency's competitive ratios so far (in the standard framework of competitive analysis).

<sup>1</sup>Generally, if  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$  then by  $x_{(k)}$  we denote the  $k$ -th highest component of  $x$ ,  $k = 1, 2, \dots, n$ . Notice that it is well defined if  $a_i \neq a_j$  for all  $i \neq j$ .

**DEFINITION 7.1 (CLS)** *We will refer to the above optimization (maximization of two possible objective functions) problem setting, coupled with the no-early arrivals assumption, as the Canonical Limited-Supply (CLS) problem.*

It is worth pointing out the absence of the no-late departures assumption from the definition of the CLS problem, the reason for such a relaxation primarily being that in this setting (in contrast to the CEI environment, see Theorem 6.8, page 79) such a restriction is not necessary in order to guarantee “desirable” properties for our auctions. Also, note the adoption of revenue as an additional, important, performance criterion. This is, mainly, due to the fact that we are able to provide simple and solid results for the revenue competitiveness of the auctions we will present for the CLS problem, but also due to nature and interpretation of the CLS problem as a product-selling procedure, in which, obviously, the auctioneer’s (seller’s) revenue is of major importance.

Based on our discussion preceding Definition 7.1, we have the following expressions for the competitive ratios of an online auction  $\mathcal{A} = (x, \{p_i\}_{i=1}^N)$  for the CLS problem with respect to efficiency and revenue, respectively:

$$\text{CR}_{\text{CLS}}^E(\mathcal{A}) = \max_{W_N, I_N} \frac{w_{(1)}}{\mathbb{E}_\theta \left[ \sum_{i=1}^N x_i(\theta) w_i \right]} \quad (7.1)$$

$$\text{CR}_{\text{CLS}}^R(\mathcal{A}) = \max_{W_N, I_N} \frac{w_{(2)}}{\mathbb{E}_\theta \left[ \sum_{i=1}^N p_i(\theta) \right]} \quad (7.2)$$

At these expressions, the adversary chooses the bids and arrival-departures sets  $W_N, I_N$  so as to maximize these ratios (minimize our auction’s  $\mathcal{A}$  performance/competitiveness) and the expectations are taken with respect to the random-order hypothesis deciding randomly the input  $\theta = (\theta_1, \theta_2, \dots, \theta_N)$  (that is, quite informally, deciding the *order* in which the bids arrive). Finally, note that equations (7.1) and (7.2) may seem more involved than they actually are, since, eventually only one of the summation terms will be nonnegative (only one allocation can be made), at each expression.

## 7.1 The Classical Secretary Problem

We now turn to study a well-known to computer scientists and probabilists, classic problem in the field of optimal stopping theory and which, not only will it help us prove some of the competitiveness results for the auctions we are going to present for the CIE problem, but it will also help as a justification for our choice of these particular auctions. Due to the fact that the problem is simple to state, fundamental and interesting enough to be applied

to various problem settings and has a clear and powerful solution, it has been extended in many directions and can be recognized under many variations. For an interesting review of the area of “secretary problems” we refer to [Freeman, 1983]. Here we will study the simplest form of the problem, known as the *classical secretary problem*: We know that  $N$  applicants for a particular job opening (e.g. a secretary position) are going to be interviewed, one after the other. However, we do not know the specific *order* in which they are going to arrive. In particular, we assume that all  $N!$  possible permutations are equally likely to occur. Immediately, after we interview an applicant we must decide whether to hire him (in which case the interview process is finished) or not and this decision is irrevocable, i.e. in case we reject him and move on to the next applicant we cannot regret that and hire him at some following stage of the process. The important assumption here, is that we do not know the qualifications of the applicants, until we actually interview them and, after interviewing an applicant we can only determine his relative rank with respect to those interviewed before him. We have no idea about the “quality” of a future arriving applicant. Furthermore, we assume that no two applicants have the same qualifications (can be equally ranked). The problem asks for the best time to stop the process, in order to maximize the *probability* that the applicant we actually hired is the best one. We are particularly interested in the case of  $N \rightarrow \infty$ . Note that the way in which our applicants arrive (random permutation) is essentially a simplified version of the random-order hypothesis we made for the CLS problem (see page 86).

Let  $M(j)$ ,  $j = 1, 2, \dots, N$ , denote the highest ranked applicant among the first  $j$  arriving applicants. We will restrict our attention to **threshold, learning algorithms** for our problem, i.e. algorithms which, for some  $k \in \{1, 2, \dots, N - 1\}$ , they just observe the first  $k$  arriving applicants in order to learn  $M(k)$ , without making any selection, and then hire the first applicant to arrive and be more qualified than  $M(k)$ . Let’s see what is the best choice for  $k$ . Define the following probability (with respect to the random permutation of applicants) events

$A_i$  : “applicant  $i$  is the best”

$B_i$  : “applicant  $i$  is hired”

$C_i$  : “applicant  $i$  is the best and hired”.

Also, let  $S$  be the event of success, i.e. the event that we hire the best applicant. Our goal is to maximize  $\mathbb{P}[S]$ . A small note on notation: here we use  $i$  as an index of the order in which the applicants arrive, i.e. by “applicant  $i$ ” we mean “the  $i$ -th arriving applicant”, and not explicitly of the applicants themselves. Obviously, no permutation can result to

two different best applicants (remember that we allow no ties), thus events  $A_i$  are mutually disjoint and so it easy to see that  $\mathbb{P}[S] = \sum_{i=1}^N \mathbb{P}[C_i]$ . But  $\mathbb{P}[C_i] = \mathbb{P}[A_i \cap B_i]$  and from the definition of conditional probability  $\mathbb{P}[C_i] = \mathbb{P}[A_i] \cdot \mathbb{P}[B_i | A_i]$ . Finally,

$$\mathbb{P}[S] = \sum_{i=1}^N \mathbb{P}[A_i] \mathbb{P}[B_i | A_i]. \quad (7.3)$$

Since we have a random permutation, each applicant  $i$  is equally likely to be the best one, so  $\mathbb{P}[A_i] = \frac{1}{N}$  for every  $i = 1, 2, \dots, N$ . Let's compute  $\mathbb{P}[B_i | A_i]$ . Assume that applicant  $i$  is the best. First of all, trivially  $\mathbb{P}[B_i | A_i] = 0$  for every  $i = 1, 2, \dots, k$  since we make no selection at this "learning" phase of our algorithm. For  $i \geq k + 1$ , the probability of applicant  $i$  being hired equals the probability that no applicant better than the threshold  $M(k)$  is among applicants  $k + 1, k + 2, \dots, i - 1$ . But for applicant  $M(k)$  being the threshold means that he is the best among applicants  $1, 2, \dots, k$ . Combining the above, we get that  $\mathbb{P}[B_i | A_i]$  equals the probability that the most highly ranked candidate among applicants  $1, 2, \dots, i - 1$  actually arrives among the first  $k$  and, because it is equally likely to appear at any of these positions  $1, 2, \dots, k$  we conclude that, for  $\mathbb{P}[B_i | A_i] = \frac{k}{i-1}$ . Combining all these in (7.3) we get

$$\mathbb{P}[S] = \sum_{i=k+1}^N \frac{1}{N} \frac{k}{i-1} = \frac{k}{N} \sum_{i=k+1}^N \frac{1}{i-1} = \frac{k}{N} \sum_{i=k}^{N-1} \frac{1}{i}$$

and approximating by integrals (see, e.g. [Cormen et al., 2001, p. 1067]), since  $\frac{1}{x}$  is a decreasing function of  $x$ ,

$$\frac{k}{N} \int_k^N \frac{1}{x} dx \leq \mathbb{P}[S] \leq \frac{k}{N} \int_{k-1}^{N-1} \frac{1}{x} dx$$

and evaluating the integrals,

$$\frac{k}{N}(\ln N - \ln k) \leq \mathbb{P}[S] \leq \frac{k}{N}(\ln(N-1) - \ln(k-1)). \quad (7.4)$$

At first, note that expression (7.4) provides a rather tight lower bound for  $\mathbb{P}[S]$ , since

$$\begin{aligned}
 & \lim_{N \rightarrow \infty} \left[ \frac{k}{N} (\ln(N-1) - \ln(k-1)) - \frac{k}{N} (\ln N - \ln k) \right] \\
 &= k \cdot \lim_{N \rightarrow \infty} \left[ \frac{1}{N} (\ln(N-1) - \ln N) - \frac{1}{N} (\ln k - \ln(k-1)) \right] \\
 &= k \cdot \lim_{N \rightarrow \infty} \left[ \frac{1}{N} \ln \frac{N-1}{N} - 0 \right], \quad \text{since } \ln \frac{k}{k-1} \leq \ln N, \\
 &= k \cdot \lim_{N \rightarrow \infty} \ln \sqrt[N]{1 - \frac{1}{N}} \\
 &= k \cdot \ln 1 = 0
 \end{aligned}$$

and by differentiating the lower bound  $L(k) = \frac{k}{N} (\ln N - \ln k)$  with respect to  $k$ , it is not difficult to see that it achieves its maximum value for  $k = \frac{N}{e}$ . So, selecting  $k = \frac{N}{e}$  results at (7.4) to the best lower bound for our probability of success

$$\mathbb{P}[S] \geq L\left(\frac{N}{e}\right) = \frac{\frac{N}{e}}{N} \left( \ln N - \ln \frac{N}{e} \right) = \frac{1}{e} \ln \frac{N}{\frac{N}{e}} = \frac{1}{e} \ln e = \frac{1}{e}.$$

Note, however, that we can not use  $k = \frac{N}{e}$ , as it is, to optimize our threshold mechanism, because  $\frac{N}{e} \notin \mathbb{N}$ . Instead, we use  $k = \lfloor \frac{N}{e} \rfloor$  which has an insignificant effect on our derived optimal lower bound, since

$$L\left(\left\lfloor \frac{N}{e} \right\rfloor\right) = \frac{\lfloor \frac{N}{e} \rfloor}{N} \left( \ln N - \ln \left\lfloor \frac{N}{e} \right\rfloor \right) \geq \frac{\frac{N}{e} - 1}{N} \left( \ln N - \ln \frac{N}{e} \right) \rightarrow L\left(\frac{N}{e}\right),$$

as  $N \rightarrow \infty$ .

Summarizing, the optimal solution to the classical secretary problem is to interview the first  $\lfloor \frac{N}{e} \rfloor$  applicants without hiring any of them and then hire the first applicant ranking higher than all these first  $\lfloor \frac{N}{e} \rfloor$  candidates. This results to a probability of at least  $\frac{1}{e} \approx 36.8\%$  that we will actually hire the best candidate, which is quite satisfying if one thinks about the unfavourable (for the employer) conditions under which the interviewing process is taking place.

It seems like the first scientific paper in which the classical secretary problem was solved is that of Lindley [1961]. However, the origins and motivation of the problem go way before that publication and for a rather entertaining (though retaining a high scientific standard) exposition of the history and the ideas of the problem we strongly recommend [Ferguson, 1989]. For a more thorough analysis of the classical secretary problem (e.g.



why it suffices to consider only threshold algorithms?) as well as an introduction to a reach mathematical theory underpinning this classic problem we refer to a text in *optimal stopping theory*, e.g. [Ferguson, 2007, chapter 2]. The problem has been extended and generalized in many interesting and powerful ways, applying in many mathematical fields. For the classic paper that set the foundations of that development, analyzing some standard variations of the secretary problem we refer to Gilbert and Mosteller [1966].

## 7.2 Adaptive Threshold Auctions for the CLS Problem

We now return to the study of our CLS problem (Definition 7.1) and define a specific family of auctions, parametrized by some  $k = 1, 2, \dots, N$ .

**DEFINITION 7.2 (ADAPTIVE AUCTION)** *For every  $k = 1, 2, \dots, N$  define  $\mathcal{A}(k)$  to be the following auction for the CLS problem:*

- (i) *Learning Phase: Make no allocation until you receive the  $k$ 'th bid at time period  $\tau$ . Let  $p \geq q$  be the two top bids received so far.*
- (ii) *Transition Phase: If some agent  $i$  with bid  $w_i = p$  is still active at time period  $\tau$ , then allocate  $i$  for a payment of  $q$  (breaking ties randomly).*
- (iii) *Accepting Phase: If no agent got allocated during the transition phase (i.e. at  $\tau$ ), allocate the first agent to arrive after  $\tau$  bidding at least  $p$  (no ties possible), for a payment of  $p$ . If no such agent arrives, allocate the last bidder to arrive.*

The reader would immediately recognize the essence of the threshold algorithm for the classical secretary problem we presented in section 7.1 underlying the design of this family of auctions  $\mathcal{A}(k)$  (especially for the case of  $k = \lfloor \frac{N}{e} \rfloor$ ). Moreover, the term *adaptive* comes exactly from this procedure of observing in order to learn and set the proper threshold value  $p$  and, depending on whether the maximum bidding so far agent is still “alive”, adapt appropriately. Apart from that, another familiar, classic auction makes its subtle appearance, in particular the Vickrey auction. This is apparent at the transition phase, but also consists the essence of the accepting phase (if we eventually reach it). These second-price payments are the main reason for us being able to establish the following

**THEOREM 7.3** *For every  $k = 1, 2, \dots, N$ , the adaptive auction  $\mathcal{A}(k)$  is IR and strongly truthful for the CLS problem.*

**PROOF** Proving IR is trivial, since every adaptive auction allocates for a payment not exceeding the bid of the allocated agent. For truthfulness, fix some  $k = 1, 2, \dots, N$ , an agent  $i$  with true type  $\theta_i = (a_i, d_i, w_i)$  and types  $\theta_{-i}$ . We must prove that (recall Definition 4.3, page 55), for every  $\theta'_i \in C(\theta_i)$  and every possible random tie-breaking (at transition and accepting phases, see Definition 7.2) of  $\mathcal{A}(k)$ ,

$$u_i(\theta_i, \theta_i, \theta_{-i}) \geq u_i(\theta'_i, \theta_i, \theta_{-i}).$$

Let  $\tau$ ,  $p$  and  $q$  be as in Definition 7.2 when  $\mathcal{A}(k)$  is run on input  $(\theta_i, \theta_{-i})$  and  $\tau'$ ,  $p'$  and  $q'$  be their respective values when agent  $i$  misreports  $\theta'_i$ , i.e. on input  $(\theta'_i, \theta_{-i})$ . Note that, due to no early-arrivals, for every possible misreport  $\theta'_i = (a'_i, d'_i, w'_i)$  we have that  $a'_i \geq a_i$ . Thus, since all other agents' arrival times are fixed, agent  $i$  can only delay (but not accelerate) the reach of the transition phase, i.e.  $\tau' \geq \tau$ . Also, it is not difficult to see that at most one agent (namely  $i$ ) can be within the first  $k$  to arrive in the first auction scenario (where  $i$  reports truthfully) but arriving after the transition phase in the new auction (where  $i$  misreports  $\theta'_i$ ). That means that the maximum bid  $p$  (up to time  $\tau$ ) can “fall”, at the new auction, no lower than the second highest bid, i.e.

$$p' \geq q. \tag{7.5}$$

Now we have the following different cases to consider:

Case 1,  $a_i \leq d_i < \tau$ . Then, from the Definition 7.2 of  $\mathcal{A}(k)$ , agent  $i$  is not allocated, thus  $u_i(\theta_i, \theta_i, \theta_{-i}) = 0$ . MENTION THAT CANNOT RECEIVE AFTER TRUE DEPARTURE. Even if, when agent  $i$  misreports  $\theta'_i$ ,  $\mathcal{A}(k)$  decides to allocate her, this would occur at some time point  $t \geq \tau' \geq \tau > d_i$ , resulting to a zero utility for our bidder.

Case 2,  $a_i \leq \tau \leq d_i$ . Here agent  $i$  is active at the transition phase so, if  $w_i < q$  then she is not allocated and  $u_i(\theta_i, \theta_i, \theta_{-i}) = 0 > w_i - q$ , if  $w_i = q$  then either she is not allocated and again  $u_i(\theta_i, \theta_i, \theta_{-i}) > w_i - q$  or she receives the item for a payment of  $q$  in which case  $u_i(\theta_i, \theta_i, \theta_{-i}) = w_i - q$ , and, finally, if  $w_i > q$  then  $w_i = p$  (i.e. she has the single maximum bid received so far) and gets allocated with  $u_i = w_i - q$ . So, in any case we know that

$$u_i(\theta_i, \theta_i, \theta_{-i}) \geq w_i - q. \tag{7.6}$$

Now, when agent  $i$  misreports  $\theta'_i$  there are two possible cases to consider:

Case 2a,  $a'_i \leq \tau'$ , i.e. our agent is active at the transition phase of the new auction. Notice that, since all other agents' types  $\theta_{-i}$  (and in particular their arrival times) are

kept fixed, the fact that both  $a_i \leq \tau$  and  $a'_i \leq \tau'$  hold means that the  $k$  first agents of the old auction (the one where  $i$  reports truthfully) are exactly the same as the first  $k$  of the new one (their order of arrival, though, is not necessarily the same). With a little more thought one can see that, whatever agent's  $i$  new bid  $w'_i$  may be,

$$q' \geq \min \{w'_i, q\}. \quad (7.7)$$

But then, even if  $i$  receives the item in the new auction, this would be in the transition phase and would require a bid  $w'_i = p'$ , thus, from (7.5),

$$w_i \geq q. \quad (7.8)$$

Bidder's  $i$  utility of this allocation would be

$$\begin{aligned} u_i(\theta_i, \theta'_i, \theta_{-i}) &= w_i - q', && \text{transition phase allocation,} \\ &\leq w_i - q, && \text{from (7.7) and (7.8),} \\ &\leq u_i(\theta_i, \theta_i, \theta_{-i}), && \text{from (7.6).} \end{aligned}$$

Case 2b,  $a'_i \leq \tau'$ . Then even if  $i$  receives the item in the new auction, this would be an accepting phase allocation, resulting to a payment of  $p'$ , thus

$$u_i(\theta_i, \theta'_i, \theta_{-i}) = w_i - p' \stackrel{(7.5)}{\leq} w_i - q \stackrel{(7.6)}{\leq} u_i(\theta_i, \theta_i, \theta_{-i}).$$

Case 3,  $\tau < a_i \leq d_i$ . In this case, due to that  $a'_i \geq a_i > \tau$ , agents  $i$  misreporting has no effect at all at the first two phases of our auction, meaning that

$$\tau = \tau' \quad \text{and} \quad p = p' \quad \text{and} \quad q = q'.$$

Also, if the auction allocates at some times prior to  $i$ 's arrival, there is nothing she can do to change that and receive the item. So, we will restrict our attention to the case where, the old auction, makes an allocation to some agent at a time period (of the accepting phase) with  $a_i \leq t^*$ . If agent  $i$  is not the one allocated, then she must have reported  $w_i < p$  since she is the first to arrive in the interval  $[a_i, T]$  (remember that we allow no ties at the arrival time of different agents). But then, even if she receives the item in the new auction by misreporting  $\theta_i$ , this would be for a utility of  $u_i(\theta_i, \theta'_i, \theta_{-i}) = w_i - p' = w_i - p < 0 = u_i(\theta_i, \theta_i, \theta_{-i})$ . On the other hand, if she is allocated at the first auction, this is done for a payment  $p' = p$  which is independent of  $i$ 's report  $\theta_i$  and thus, agent

$i$  can do nothing to improve her utility, i.e. reduce her payment.  $\square$

One could argue that, since we did all this effort to formally prove our truthfulness characterization results of section 5.3, it would be more easy to use Corollary 5.9 (page 67) in our proof, like we did for the case of the Greedy Auction for the CEI problem (see Theorem 6.3). However, a more careful look brings up the matter of limited misreports. We would need to have a generalization of Corollary 5.9 for arbitrary misreports of departure (we have assumed no restriction in departure misreports for our CLS model). Actually, though, this is possible, by introducing the notion of *monotonic-late* mechanisms (see [Parkes, 2007, p. 420]).

### 7.3 Upper Bounds

**THEOREM 7.4** *The adaptive auction  $\mathcal{A}(k)$  is  $\alpha(k)$ -competitive for efficiency and  $\beta(k)$ -competitive for revenue (for the CLS problem), where*

$$\alpha(k) = \begin{cases} \frac{N}{k}, & \text{if } k \leq \frac{N}{e}, \\ \frac{N}{k} \frac{1}{\ln \frac{N}{k}}, & \text{if } k > \frac{N}{e} \end{cases} \quad \text{and} \quad \beta(k) = \begin{cases} \left(\frac{N}{k}\right)^2, & \text{if } k \leq \frac{N}{2}, \\ \frac{N^2}{k(N-k)}, & \text{if } k > \frac{N}{2}, \end{cases}$$

for every  $k = 1, 2, \dots, N$ .

**PROOF** We fix a  $k = 1, 2, \dots, N$  for our adaptive auction  $\mathcal{A}(k)$  and let our adversary select bids and arrival-departure interval sets  $W_N$  and  $I_N$ , respectively (see (7.1) and (7.2), page 87). Then, a type profile  $\theta = (\theta_1, \theta_2, \dots, \theta_N)$  is generated randomly (uniformly, by “matching”  $W_N$  and  $I_N$ ), according to the random-ordering hypothesis (see page 86) and then passed as an input to our auction algorithm. It is without loss of generality to assume that the components of  $\theta$  are ordered so as to  $w_i = w_{(i)}$  for all  $i = 1, 2, \dots, N$ , i.e.  $w_1 \geq w_2 \geq \dots \geq w_N$ . This is because the indexes (identities)  $i$  of our agents carry no information at all about our input<sup>2</sup>, which is completely determined by our agents’ types, i.e. by the set  $\{\theta_1, \theta_2, \dots, \theta_N\}$ . The input  $(\theta_1, \theta_2, \dots, \theta_N)$  is exactly the same as  $(\theta_{\pi(1)}, \theta_{\pi(2)}, \dots, \theta_{\pi(N)})$  to our adaptive auction, for every permutation  $\pi$  on  $\{1, 2, \dots, N\}$ . Finally, let  $\tau(\theta)$  be the time period of our auctions transition phase (on input  $\theta$  and  $t^*(\theta)$  the time period at which it makes its (one and only) allocation.

First, let us consider the case of efficiency. Adapting equation (7.1) (page 87) to our analysis (and removing the maximum with respect to  $W_N, I_N$  since we have “allowed”

<sup>2</sup>Note that the ordering  $(\theta_1, \theta_2, \dots, \theta_N)$  in no way determines the order of arrival for our agents, since this is completely defined by their arrival times  $a_i$ .

for our adversary to make this selection), we take that

$$\text{CR}_{\text{CLS}}^E(\mathcal{A}(k)) = \frac{w_1}{\mathbb{E}_\theta \left[ \sum_{i=1}^N x_i(\theta) w_i \right]} = \frac{w_1}{\sum_{i=1}^N \mathbb{P} [x_i(\theta) = 1] w_i},$$

and since, obviously,  $\sum_{i=1}^N \mathbb{P} [x_i(\theta) = 1] w_i \geq \mathbb{P} [x_1(\theta) = 1] w_1$ ,

$$\text{CR}_{\text{CLS}}^E(\mathcal{A}) \leq \frac{w_1}{\mathbb{P} [x_1(\theta) = 1] w_1} = \frac{1}{\mathbb{P} [x_1(\theta) = 1]}. \quad (7.9)$$

It remains to calculate  $\mathbb{P} [x_1 = 1]$ , i.e. the probability that the agent with the highest bid is the one who gets the item. There are two possible cases to consider, depending on whether the allocation is made at the transition phase or during the accepting phase.

Case 1,  $t^*(\theta) = \tau(\theta)$ . Condition on selling on the transition phase, the agent with the highest bid seen during time interval  $[1, \tau(\theta)]$  is still active at  $\tau(\theta)$  and the one allocated. So, in this case, the probability of selling to the highest bidding agent  $\theta_1$  equals the probability that  $w_1$  is among these first  $k$  bids to arrive. Thus, based on the random-ordering hypothesis,

$$\mathbb{P} [x_1(\theta) = 1 \mid t^*(\theta) = \tau(\theta)] = \frac{k}{N}.$$

Case 2,  $t^*(\theta) > \tau(\theta)$ . Condition on selling the item at the transition phase, the probability of selling to the highest bidding agent equals the probability that the first bid to arrive after  $\tau(\theta)$  being at least  $p$  is  $w_1$ . With a little thought, one can see that in this case our analysis “collapses” to that of the classical secretary problem (see section 7.1), thus we can obtain a lower bound of

$$\mathbb{P} [x_1(\theta) = 1 \mid t^*(\theta) > \tau(\theta)] \geq L(k) = \frac{k}{N} (\ln N - \ln k) = \frac{k}{N} \ln \frac{N}{k}.$$

So, combining both cases

$$\mathbb{P} [x_1(\theta) = 1] \geq \min_{k \in [1, N]} \left\{ \frac{k}{N}, \frac{k}{N} \ln \frac{N}{k} \right\} = \begin{cases} \frac{k}{N}, & \text{if } k \leq \frac{N}{e}, \\ \frac{k}{N} \ln \frac{N}{k}, & \text{if } k > \frac{N}{e} \end{cases},$$

with the use of some basic calculus. Thus (7.9) results to the desired competitive ratio

of

$$\text{CR}_{\text{CLS}}^E(\mathcal{A}(k)) \leq \begin{cases} \frac{N}{k}, & \text{if } k \leq \frac{N}{e}, \\ \frac{N}{k} \frac{1}{\ln \frac{N}{k}}, & \text{if } k > \frac{N}{e}. \end{cases}$$

Now, let us consider the case of revenue. In the same spirit of our proof for the case of efficiency, we can easily see that for the expected revenue of our adaptive auction  $\mathcal{A}$

$$\begin{aligned} R^{\mathcal{A}}(\theta) &= \mathbb{E}_{\theta} \left[ \sum_{i=1}^N p_i(\theta) \right] \\ &= \sum_{i=1}^N \sum_{j=1}^N \mathbb{P} [x_i(\theta) = 1 \wedge p_i(\theta) = w_j] w_j \\ &\geq \mathbb{P} [x_1(\theta) = 1 \wedge p_1(\theta) = w_2] w_2, \end{aligned}$$

and, from equation (7.2) (page 87), for our competitive ratio,

$$\text{CR}_{\text{CLS}}^R(\mathcal{A}) \leq \frac{w_2}{\mathbb{P} [x_1(\theta) = 1 \wedge p_1(\theta) = w_2] w_2} = \frac{1}{\mathbb{P} [x_1(\theta) = 1 \wedge p_1(\theta) = w_2]}. \quad (7.10)$$

We have to compute  $\mathbb{P} [x_1(\theta) = 1 \wedge p_1(\theta) = w_2]$ , i.e. the probability of the highest bidding agent receiving the item for a second-price payment of  $w_2$ . again we proceed by case analysis:

Case 1,  $t^*(\theta) = \tau(\theta)$ . Condition on selling on the transition phase, the probability of selling to the highest bidding agent for a payment of  $w_2$  equals the probability that both the two highest bids  $w_1, w_2$  arrive during  $[1, \tau(\theta)]$ , i.e. both  $w_1$  and  $w_2$  are among the first  $k$  bids to arrive. Notice that, these two events are independent, due to our random-ordering hypothesis, so  $\mathbb{P} [a_1 \leq \tau(\theta) \wedge a_2 \leq \tau(\theta)] = \mathbb{P} [a_1 \leq \tau(\theta)] \cdot \mathbb{P} [a_2 \leq \tau(\theta)] = \frac{k}{N} \cdot \frac{k}{N}$ .

Thus,

$$\mathbb{P} [x_1(\theta) = 1 \wedge p_1(\theta) = w_2 \mid t^*(\theta) = \tau(\theta)] = \left( \frac{k}{N} \right)^2$$

Case 2,  $t^*(\theta) > \tau(\theta)$ . Condition on selling on the accepting phase, the probability of selling to the highest bidding agent for a payment of  $w_2$  equals the probability that  $w_2$  arrives during  $[1, \tau(\theta)]$  (setting the threshold value  $p = w_2$ ) and  $w_1$  arrives after  $\tau(\theta)$ .

Thus,

$$\begin{aligned} \mathbb{P}[x_1(\theta) = 1 \wedge p_1(\theta) = w_2 \mid t^*(\theta) > \tau(\theta)] &= \mathbb{P}[a_2 \leq \tau(\theta)] \cdot \mathbb{P}[a_1 > \tau(\theta)] \\ &= \frac{k}{N} \cdot \frac{N-k}{N} \\ &= \frac{k(N-k)}{N^2}. \end{aligned}$$

Combining both cases,

$$\mathbb{P}[x_1(\theta) = 1 \wedge p_1(\theta) = w_2] \geq \min_{k \in [1, N]} \left\{ \left( \frac{k}{N} \right)^2, \frac{k(N-k)}{N^2} \right\} = \begin{cases} \left( \frac{k}{N} \right)^2, & \text{if } k \leq \frac{N}{2}, \\ \frac{k(N-k)}{N^2}, & \text{if } k > \frac{N}{2}, \end{cases}$$

hence (7.10) gives the desired

$$\text{CR}_{\text{CLS}}^R(\mathcal{A}(k)) \leq \begin{cases} \left( \frac{N}{k} \right)^2, & \text{if } k \leq \frac{N}{2}, \\ \frac{N^2}{k(N-k)}, & \text{if } k > \frac{N}{2}. \end{cases} \quad \square$$

The natural question that immediately arises here is what would be the best choice of  $k = 1, 2, \dots, N$  in order to optimize (i.e. minimize) these upper bounds on the performance of our adaptive auctions, with respect to efficiency or revenue. It turns out that, as far as efficiency is concerned, the good old trick of the classical secretary problem, i.e. setting the stopping rule at  $k = \lfloor \frac{N}{e} \rfloor$ , does the trick:

**COROLLARY 7.5** *As  $N \rightarrow \infty$ , the adaptive auction  $\mathcal{A}(\lfloor \frac{N}{e} \rfloor)$  is  $e$ -competitive for efficiency and  $e^2$ -competitive for revenue. Furthermore, this choice of  $k = \lfloor \frac{N}{e} \rfloor$  minimizes  $\alpha(k)$ , the upper bound for efficiency given in Theorem 7.4.*

**PROOF** From Theorem 7.4, page 94,

$$\text{CR}_{\text{CLS}}^E(\mathcal{A}(k)) \leq \alpha(k) = \begin{cases} \frac{N}{k}, & \text{if } k \leq \frac{N}{e}, \\ \frac{N}{k} \frac{1}{\ln \frac{N}{k}}, & \text{if } k > \frac{N}{e}. \end{cases}$$

Using basic calculus one can see that function  $\frac{N}{k}$  is strictly decreasing with respect to  $k$  in the interval  $[1, \frac{N}{e}]$  and that  $\frac{N}{k} \frac{1}{\ln \frac{N}{k}}$  is strictly increasing in  $[\frac{N}{e}, N]$ . Thus, if we want to optimize (i.e. minimize) our competitive ratio with respect to efficiency we must

choose  $k = \frac{N}{e}$  in order to minimize  $\alpha(k)$ ,

$$\alpha\left(\frac{N}{e}\right) = \frac{N}{\frac{N}{e}} = e.$$

However,  $\frac{N}{e}$  is not a valid choice, as it is, for our adaptive auctions  $\mathcal{A}(k)$ , because  $\frac{N}{e} \notin \mathbb{N}$ . Instead, we choose  $k = \lfloor \frac{N}{e} \rfloor$  for an (asymptotically) negligible loss in performance, since

$$\alpha(\lfloor \frac{N}{e} \rfloor) = \frac{N}{\lfloor \frac{N}{e} \rfloor} \leq \frac{N}{\frac{N}{e} - 1} = e \cdot \frac{N}{N - e} \rightarrow e = \alpha\left(\frac{N}{e}\right),$$

as  $N \rightarrow \infty$ .

As far as revenue is concerned, for the choice of  $k = \lfloor \frac{N}{e} \rfloor$ ,

$$\beta\left(\frac{N}{e}\right) = \left(\frac{N}{\frac{N}{e}}\right)^2 = e^2$$

and again it is trivial to check that  $e^2 \cdot \left(\frac{N}{N+e}\right)^2 \leq \beta(\lfloor \frac{N}{e} \rfloor) \leq e^2 \cdot \left(\frac{N}{N-e}\right)^2$ , thus

$$\text{CR}_{\text{CLS}}^R(\mathcal{A}(\lfloor \frac{N}{e} \rfloor)) \leq \beta(\lfloor \frac{N}{e} \rfloor) \rightarrow \beta\left(\frac{N}{e}\right) = e^2,$$

as  $N \rightarrow \infty$ . □

So, we have found an adaptive auction that is guaranteed to perform within a  $e \approx 2.718$  factor of the optimal, offline (Vickrey) auction with respect to efficiency. This factor becomes  $e^2 \approx 7.389$  when revenue is concerned. Although the above Corollary 7.5 assures us that this 2.718 efficiency upper bound is the best we can do with the tools we have from Theorem 7.4, this is not the case with revenue and we would like to know if with some other choice of  $k$  we can do better with respect to revenue. Ideally, though, we wouldn't like to see our upper bound of  $e \approx 2.718$  getting a lot "worse". It seem like, for a choice of  $k = \lfloor \frac{N}{2} \rfloor$  all this is possible. For a "small" compromise on efficiency, from 2.718 to  $\frac{2}{\ln 2} \approx 2.885$  we can considerably improve our revenue upper bound from 7.389 to 4:

**COROLLARY 7.6** *As  $N \rightarrow \infty$ , the adaptive auction  $\mathcal{A}(\lfloor \frac{N}{2} \rfloor)$  is  $\frac{2}{\ln 2}$ -competitive for efficiency and 4-competitive for revenue. Furthermore, this choice of  $k = \lfloor \frac{N}{2} \rfloor$  minimizes  $\beta(k)$ , the upper bound for revenue given in Theorem 7.4.*



PROOF From Theorem 7.4, page 94,

$$\text{CR}_{\text{CLS}}^R(\mathcal{A}(k)) \leq \beta(k) = \begin{cases} \left(\frac{N}{k}\right)^2, & \text{if } k \leq \frac{N}{2}, \\ \frac{N^2}{k(N-k)}, & \text{if } k > \frac{N}{2}, \end{cases}$$

as  $N \rightarrow \infty$ . Again, using basic calculus one can see that  $\beta(k)$  has a single minimum at  $k = \frac{N}{2}$  and

$$\beta\left(\frac{N}{2}\right) = \left(\frac{N}{\frac{N}{2}}\right)^2 = 4.$$

Also,

$$\alpha\left(\frac{N}{2}\right) = \frac{N}{\frac{N}{2}} \frac{1}{\ln \frac{N}{\frac{N}{2}}} = \frac{2}{\ln 2}.$$

Like in the proof of Corollary 7.5, it is again trivial to check that,

$$\beta(\lfloor \frac{N}{2} \rfloor) \rightarrow \beta\left(\frac{N}{2}\right) \quad \text{and} \quad \alpha(\lfloor \frac{N}{2} \rfloor) \rightarrow \alpha\left(\frac{N}{2}\right),$$

hence proving that

$$\text{CR}_{\text{CLS}}^R(\mathcal{A}(\lfloor \frac{N}{2} \rfloor)) \leq 4 \quad \text{and} \quad \text{CR}_{\text{CLS}}^E(\mathcal{A}(\lfloor \frac{N}{2} \rfloor)) \leq \frac{2}{\ln 2} \approx 2.885. \quad \square$$

As an immediate corollary from the upper bounds derived for our two adaptive auctions  $\mathcal{A}(\lfloor \frac{N}{e} \rfloor)$  and  $\mathcal{A}(\lfloor \frac{N}{2} \rfloor)$  in the preceding Corollaries 7.5 and 7.6, we get the following upper bounds on the competitive ratios for the CLS problem in general:

**COROLLARY 7.7 (Upper bounds)** *As  $N \rightarrow \infty$ , we have the following upper bounds for the CLS problem:*

$$\text{CR}_{\text{CLS}}^E \leq e \approx 2.718 \quad \text{and} \quad \text{CR}_{\text{CLS}}^R \leq 4.$$

## 7.4 Extensions – Open Problems

In our analysis of the CLS problem we didn't mention anything about *lower bounds*. In fact, such results do exist, namely 2 and  $\frac{3}{2}$  for efficiency and revenue, respectively, and can be found in [Hajiaghayi et al., 2004, 5.2]. As far as the multi-item  $k > 1$  items case is concerned, constant competitive ratios do exist, but are very large (see, again, [Hajiaghayi et al., 2004, sec. 6]). The challenging area of *Matroid Secretary Problems* provides probably

the the most important extension to our CLS problem, as well as an asymptotically close to 1 competitive ratio as  $k \rightarrow \infty$  (for the multi-item case). For these, we refer to the work of Babaioff et al. [2007] and Dimitrov and Plaxton [2008].

The most important open problem is that of analysing the case where bids are drawn independently from an unknown distribution (instead of the random-ordering hypothesis). Obviously, our upper bounds would continue to hold, though it is challenging to come up with the right lower bound. Closing the gap between upper and lower bounds is also a major problem.

# Bibliography

- K. Arrow. *Social Choice and Individual Values*. Yale University Press, 1951.
- M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443, 2007.
- S. Ben-David, A. Borodin, R. Karp, G. Tardos, and A. Wigderson. On the power of randomization in on-line algorithms. *Algorithmica*, 11(1):2–14, 1994.
- A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
- E. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
- C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a nash equilibrium. In *Proceedings of the 38th annual ACM symposium on Theory of computing (STOC)*, pages 71–78, 2006.
- N. B. Dimitrov and C. G. Plaxton. Competitive weighted matching in transversal matroids. Technical report, University of Texas, Austin, Department of Computer Science, 2008.
- T. S. Ferguson. Who solved the secretary problem? *Statistical Science*, 4(3):282–296, 1989.
- T. S. Ferguson. Optimal stopping and applications. Electronic text, 2007. <http://www.math.ucla.edu/~tom/Stopping/Contents.html>.
- A. Fiat and G. Woeginger, editors. *Online Algorithms: The State of the Art*. Springer, 1998.
- P. R. Freeman. The secretary problem and its extensions: A review. *International Statistical Review*, 51(2):189–206, 1983.

- E. J. Friedman and D. C. Parkes. Pricing wifi at starbucks - issues in online mechanism design. In *Proceedings of the 4th ACM Conference on Electronic Commerce (EC '03)*, pages 240–241, 2003.
- D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.
- J. Geanakoplos. Three brief proofs of arrow’s impossibility theorem. *Economic Theory*, 26(1):211–215, 2005.
- A. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41(4): 587–601, 1973.
- J. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966.
- T. Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973.
- B. Hajek. On the competitiveness of on-line scheduling of unit-length packets with hard deadlines in slotted time. In *Proceedings of the 2001 Conference on Information Sciences and Systems*, 2001.
- M. T. Hajiaghayi, R. D. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC '04)*, pages 71–80, 2004.
- M. T. Hajiaghayi, R. D. Kleinberg, M. Mahdian, and D. C. Parkes. Online auctions with re-usable goods. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC' 05)*, pages 165–174, 2005.
- A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Stacs 99: 16th Annual Symposium on Theoretical Aspects of Computer Science*, 1999.
- V. Krishna. *Auction Theory*. Academic Press, 2002.
- R. Lavi and N. Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 233–241, 2000.
- R. Lavi and N. Nisan. Online ascending auctions for gradually expiring items. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05)*, pages 1146–1155, 2005.

- D. V. Lindley. Dynamic programming and decision theory. *Applied Statistics*, 10(1):39–51, 1961.
- A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic theory*. Oxford University Press, New York, 1995.
- J. Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951.
- J. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.
- N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.
- N. Nisan and A. Ronen. Algorithmic mechanism design (extended abstract). In *The Thirty First Annual ACM symposium on Theory of Computing (STCO99)*, pages 129–140, may 1999.
- N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- M. J. Osborne. *An introduction to Game Theory*. Oxford University Press, 2004.
- M. Pai and R. Vohra. Optimal dynamic auctions. Technical report, Tech. rep., Kellogg School of Management, Northwestern University, 2006.
- C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.
- C. H. Papadimitriou. The complexity of finding nash equilibria. In N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 2. Cambridge University Press, 2007.
- D. Parkes and Q. Duong. An ironing-based approach to adaptive online mechanism design in single-valued domains. *Proc. 22nd National Conference on Artificial Intelligence (AAAI'07)*, 2007.
- D. C. Parkes. Online mechanism design. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 16. Cambridge University Press, 2007.

- W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill International, 3rd (international students) edition, 1976.
- M. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.
- A. Schotter. *Microeconomics: A Modern Approach*. Addison Wesley, 3rd edition, 2001.
- É. Tardos and V. V. Vazirani. Basic solution concepts and computational issues. In N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 1. Cambridge University Press, 2007.
- V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, March 1961.
- B. Vöcking. Selfish load balancing. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 20. Cambridge University Press, 2007.
- J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 3rd edition, 1953.
- B. von Stengel. Equilibrium computation for two-player games in strategic and extensive form. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 3. Cambridge University Press, 2007.

# Index

- Adaptive Auction, 91
- adversary, 43
  - oblivious, 45
- affine maximizer, 39
- approximation algorithm, 42
- arrival time, 50
- Arrow's theorem, 22
- auction
  - first-price, 29
  - sealed-bid, 28
  - second-price, 29
  - Vickrey, 29
- battle of sexes game, 6
- best response strategy, 6
- bimatrix game, 10
- Brouwer's theorem, 14, 19
- Brouwer, Luitzen E. J., 19
- Canonical Expiring Items, 74
- Canonical Limited Supply, 86
- CEI, *see* Canonical Expiring Items
- CLS, *see* Canonical Limited Supply
- competitive analysis, 44
- competitive ratio, 43, 44
- cooperation, 3
- critical bid, 73
- critical value, 62, 68
- decision, 22
- decision rule, 27
- decision-making, 21
- departure time, 50
- direct revelation mechanism, 52
- direct-revelation mechanism, 28
- dominant strategy, 11
  - equilibrium, 11, 25
- DSIC, *see* truthfulness
- early-arrivals, 54
- efficiency, 33, 62
- environment
  - Mechanism Design, 25
  - direct-revelation, 28
- equilibrium, 5
  - in dominant strategies, 11, 25
  - mixed Nash, *see* Nash equilibrium
  - Nash, 14
  - pure Nash, 12
- game
  - (strict) incomplete information, 23
  - bimatrix, 10
  - definition of, 9
  - finite, 9
  - full information, 21
  - in normal form, *see* game, strategic matrix, 10
  - strategic, 9
  - symmetric, 10

- zero-sum, 10
- Game Theory, 5, 18, 19
  - cooperative, 19
  - noncooperative, 19
- Greedy Auction, 75
- Groves mechanism, 35
- implementation, 27
- Impossibility result, 79
- incentive compatibility, *see* truthfulness
- indifference, 11
- individual rationality, 36, 51
- interesting decision, 60
- interesting set, 60
- IR, *see* individual rationality
- Kakutani's theorem, 14, 19
- late-departures, 54
- limited misreporting, 54
- matching pennies game, 7
- matrix
  - game, 10
  - utility, 10
- mechanism, 27
  - direct revelation, 52
  - direct-revelation, 28
  - efficiency of, 33
  - Groves, 35
  - revenue of, 34
  - state, 52
  - truthful, *see* truthfulness
  - VCG, 37
- mechanism environment, 25
- mixed Nash equilibrium, *see* Nash equilibrium
- mixed strategy, 13
- monotonicity, 63
- Morgenstern, O., 19
- Nash equilibrium, 14
- Nash's Theorem, 14
- Nash, John F., 14, 19
- Neumann, J. von, 19
- no-deficit principle, 50
- NP-complete, 19
- objective function, 41
- online problem, 42
- optimization problem, 41
- outcome, 22, 25
- payment rule, 27, 50
- payoff, 4
- player
  - selfish, 21
- positive transfers, 37
- PPAD, 19
- prisoner's dilemma game, 3, 4
- pure Nash equilibrium, 12
- random-ordering hypothesis, 86
- rationality, 5
- Revelation Principle, 30, 56
- revenue, 34, 86
- reward, 60
- Secretary Problem, 87
- selfish player, 21
- single-valued domain, 60
- Social Choice
  - function, 22
- social welfare, 33
- strategy
  - mixed, 13



strong truthfulness, 56  
symmetric game, 10  
  
truthfulness, 29, 55  
type, 23, 49  
  
utility, 9, 51  
utility function, 9, 23  
utility matrix, 10  
  
valuation component, 50  
valuation function, 50  
value, 50  
VCG mechanism, 37  
    weighted, 39  
Vickrey auction, 29  
  
zero-sum game, 10