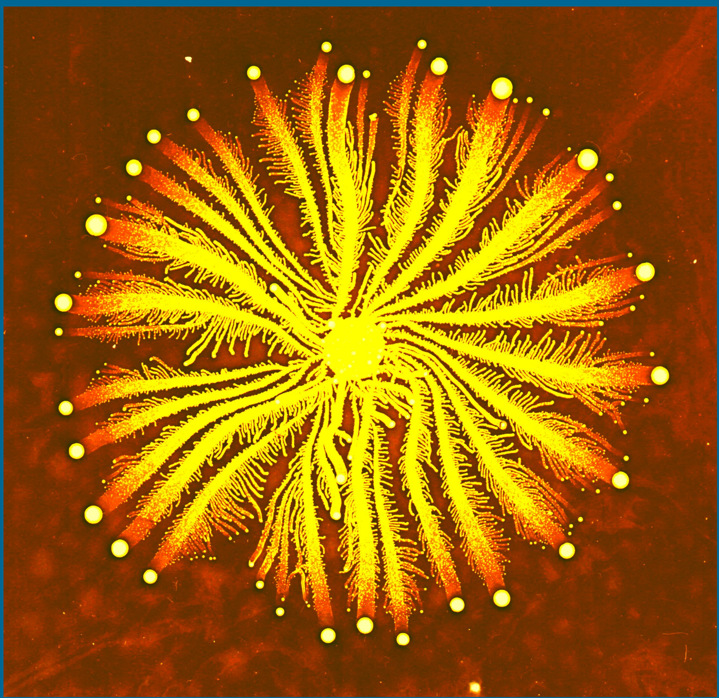
Kiril Kolev

Master Thesis

Sequential Price of Anarchy for Atomic Congestion Games with Limited Number of Players



August 2016

University of Twente

Master Applied Mathematics

Master Thesis

Sequential Price of Anarchy for Atomic Congestion Games with Limited Number of Players

Author:
Kiril Kolev

Supervisors:
Marc Uetz
Judith Timmer
Jasper de Jong, MSc

A thesis submitted in fulfillment of the requirements for the degree of Master of Science in the

Discrete Mathematics and Mathematical Programming
group of the
University of Twente

August 2016

"It is just as foolish to complain that people are selfish and treacherous as it is to complain that the magnetic field does not increase unless the electric field has a curl. Both are laws of nature."

John von Neumann

UNIVERSITY OF TWENTE
Faculty of Electrical Engineering, Mathematics, and Computer Science

Master Applied Mathematics

Abstract

Sequential Price of Anarchy for Atomic Congestion Games with Limited Number of Players

by Kiril Kolev

We consider sequentially played atomic congestion games with linear latency functions. To measure the quality of sub-game perfect equilibria of sequentially played games, we analyze the Sequential Price of Anarchy (SPoA), which is defined to be the ratio of the total cost given by the worst sub-game perfect equilibrium to the total cost of the social optimum. In this project, we want to improve lower and upper bounds on the value of SPoA using a linear programming approach. We limit the number of players to 4, 5 and 6 players and consider different combinations of their possible actions and resources they use. Additionaly we consider two player sequentially played non-atomic congestion games with split flow and give new results for the SPoA in all the possible cases.

Contents

Chapter 1

Introduction

1.1 Background

This thesis covers a topic in a relatively new field of Game Theory called Algorithmic Game Theory. We will first go through the basics of Game Theory needed to understand this thesis. More specifically, we cover the topic of Sequential Price of Anarchy in congestion games. We will explain what congestion games are, followed by the definitions of concepts like Nash equilibrium, subgame perfect equilibrium, extensive form games and price of anarchy.

Game Theory was formally developed during the 1950s by well known mathematicians such as John von Neumann, Oskar Morgenstern and John F. Nash. One of the most notable works is the book "Theory of Games and Economic Behavior" [1] by von Neumann and Morgenstern published in 1944. On the other hand, the development of Algorithmic Game Theory started around 1999, when Noam Nisan and Amir Ronen published the article "Algorithmic Mechanism Design" [2]. Another big cornerstone of the rise of Algorithmic Game Theory are the article on "Worst Case Subgame Equilibria" by Koutsopias and Papadimitriou [3], and the paper "How bad is selfish routing?" by Tardos and Roughgarden [4]. A very important resource we used while preparing for this thesis was the book by Tim Roughgarden titled "Algorithmic Game Theory" [5]. All the above mentioned authors were awarded with the Gödel prize in 2012 [6].

1.2 What is Game Theory?

Game Theory is the study of mathematical models of cooperation or conflict of interests between rational decision makers [7]. Usually, these rational decision makers are called players, and their decisions are called actions. Each player will have a set of admissible actions.The precise definition will follow later, for now the intuitive meaning will suffice. A player's strategy determines the action the player will take at any stage of the game. Game Theory has applications ranging from economics, politics, computer science, psychology, biology, poker and even basketball [7, 8, 9, 10, 11]. The earliest forms of strategic modeling considered throughout history appear in the Bible and the Talmud [12, 13]. An even earlier form of strategic modeling is considered in the work of the Chinese warrior-philosopher Sun Tzu [14].

Let us consider the common example of a game called the prisoner's dilemma, where two prisoners are being interrogated in separate rooms about a crime they commited together. They have two actions available. The first one is to confess their crime, while the second is to plead not guilty. If both of them confess their crimes, they are locked for ten years. If both prisoners plead not guilty, they are locked for one year for minor charges. If one prisoner confesses, and the other pleads not guilty, then the confessing prisoner is released, and the prisoner who pleads not guilty is locked for eleven years.

		prisoner B	
		confess	plead not guilty
prisoner A	confess	(10,10)	(0,11)
	plead not guilty	(11,0)	(1,1)

Table 1.1: Prisoner's dilemma example

If we look at Table 1.1 we can see the actions and the corresponding outcomes of each action for both prisoners. The first number in each entry is the cost for prisoner A, while the second is the cost for prisoner B. If both prisoners behave rationally, they would both confess, since they will both save one year of prison regardless of what the other prisoner chooses. If both of them cooperated, they would have ended up with only one year of prison sentence, but this outcome is not achievable since, if one prisoner denies, it is optimal for the other prisoner to confess and not be convicted, while if he denies he will have to spend one year in jail. The outcome where both players confess is an equilibrium, more specifically, it is a *Nash Equilibrium* and we will define this concept in section 1.3.2.

1.3 Congestion Games

Congestion games are a class of games in game theory first proposed by Rosenthal in 1973 [13]. We consider *sequentially played atomic congestion games with linear latency functions*. Congestion games are a special case of potential games. The elements of a congestion game are the players and the resources they compete for, where the utility of each player depends on the resources she chooses and the number of players choosing the same resources. In general, congestion games are played simultaneously, meaning that all players decide which resources to use at the same time, without knowing what the other players will choose. Sequentially played congestion games means that players act in the order by their number, i.e., Player 1 chooses first, Player 2 chooses next, and Player n chooses last. Player i chooses her action A_i to minimize her cost, observing the actions of players preceeding her, but without knowing the actions of players succeeding her.

1.3.1 Notation

In this section we describe the input of an instance $I \in \mathcal{I}$ of atomic congestion games with linear latency functions. Formally, such a game consists of a finite set of *players* $\mathcal{N} = \{1, 2, ..., n\}$ and a finite set of *resources* $\mathcal{R}$. Each player $i \in \mathcal{N}$ has her own *collection of possible actions* $\mathcal{A}_i = \{A \subset \mathcal{R} | A \text{ is a possible action of player } i\}$. By choosing one action for each player i, say A_i, from this collection, an *action profile* $\boldsymbol{A} = (A_1, ..., A_n)$ is formed. In sequentially played congestion games players act in the order by their number, i.e., Player 1 chooses first, Player 2 chooses next, and Player n chooses last. Player i chooses her action A_i to minimize her cost, observing the actions of players preceeding her, but without knowing the actions of players succeeding her. A strategy S_i then specifies for player i the actions she chooses, one for each potential profile of actions chosen by her predecessors 1,...,i-1. We denote by S a strategy profile $(S_i)_{i \in N}$. The outcome $A = (A_i)_{i \in N}$ of a game is then the set of actions chosen by each player resulting from a given strategy profile S. If S is a strategy profile of a subgame perfect equilibrium, the resulting outcome A is not necessarily a Nash equilibrium of the strategic form (simultaneous game) as we will mention later in section 1.3.2.

Each resource $r \in \mathcal{R}$ has a *latency function* $f_r = d_r + w_r n_r$, where d_r is the non negative *constant activation cost* of resource r, w_r is the non negative *weight* of resource r, and n_r is the number of players choosing resource r in the action profile $\boldsymbol{A}$. It is easily seen that f_r is a nondecreasing function of n_r, indicating that the resource gets more congested the more players choose it. Given an action profile $\boldsymbol{A}$, the cost of player i is $\text{cost}_i = \sum_{r \in A_i} f_r$, and the total cost of all players is $\text{cost}_{\text{total}} = \sum_{i \in \mathcal{N}} \text{cost}_i$.

1.3.2 Nash Equilibria, Subgame Perfect Equilibria and Extensive Form Games

Nash equilibrium is one of the most important concepts in Game Theory. A Nash Equilibrium is a stable state of a system involving the interaction of different participants, in which no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged [13]. In atomic games, a *pure Nash equilibrium* is an action profile $\boldsymbol{A}$ in which no player i can decrease her own cost by unilaterally deviating from choosing her original action A_i. However, the total cost of a Nash equilibrium is not always the optimal cost, i.e. the minimal cost. By optimal cost we mean the concept of social cost of an equilibrium. In our case we define this social cost as the sum of the costs of all players. The following example illustrates this point.

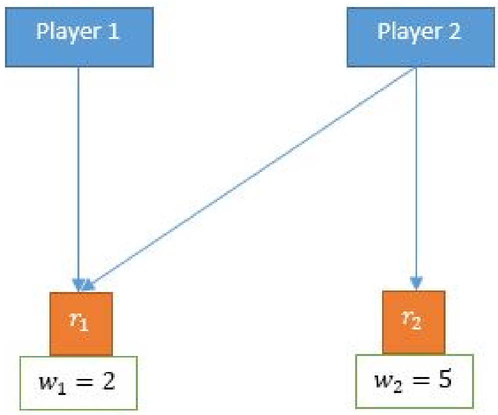


FIGURE 1.1: Illustrative example where Nash Equilibrium is not socially optimal.

In the game shown in **Figure 1.1**, where we have two resources $\mathcal{R} = \{r_1, r_2\}$, two players $\mathcal{N} = \{1, 2\}$, player one can choose only one resource $\mathcal{A}_1 = \{\{r_1\}\}$, player two can choose between two resources $\mathcal{A}_2 = \{\{r_1\}, \{r_2\}\}$, and the costs incured by each resource are $f_{r_1} = 2n_{r_1}$, $f_{r_2} = 5n_{r_2}$ where n_{r_1} and n_{r_2} are the congestion indexes of resource one and two respectively. We see that the optimal solution is when Player 1 chooses r_1, and Player 2 chooses r_2, giving a total cost of $2 + 5 = 7$, while in the only Nash equilibrium, when no one can decrease her cost by unilaterally deviating from her original choice, Player 1 and Player 2 both choose r_1, giving a total cost of $4 + 4 = 8$. Therefore, Nash equilibrium doesn't always give an optimal cost.

Our goal is to compare the quality of an optimal outcome (from the social cost point of view) to the quality of subgame perfect equilibria of an extensive form game as introduced in [15, 16]. Since players act in order, we build a *game tree* to describe the game in *extensive form*. Such a game tree is characterized by *nodes* and *edges*, where each node is either a *decision node* of a player, or an *end note*, and each edge corresponds to an action. A *Subgame* can be described by a subtree of a game tree. It starts at a single decision node of a player. A subgame perfect equilibrium is a refinement of a Nash equilibrium used in dynamic games. A strategy profile is a subgame perfect equilibrium if it represents a Nash equilibrium of every subgame of the original game. It is important to mention that all games considered here have only trivial information sets, meaning that we consider games with perfect information. This means that whenever the game reaches a decision node of the game tree, the players are assumed to play a Nash equilibrium strategy in the corresponding subgame. The example above can be described in the game tree in **Figure 1.2**. "Player 1" and "Player 2" are decision nodes, and nodes with costs are end nodes. The numbers are the costs of Player 1 and Player 2, respectively. Texts on the edges show the collections of actions of players. In the subgame starting from decision node "Player 2", the only subgame perfect action is that Player 2 chooses r_1, since $\text{cost}_2(\{r_1\}, \{r_1\}) = 4 < 5 = \text{cost}_2(\{r_1\}, \{r_2\})$.

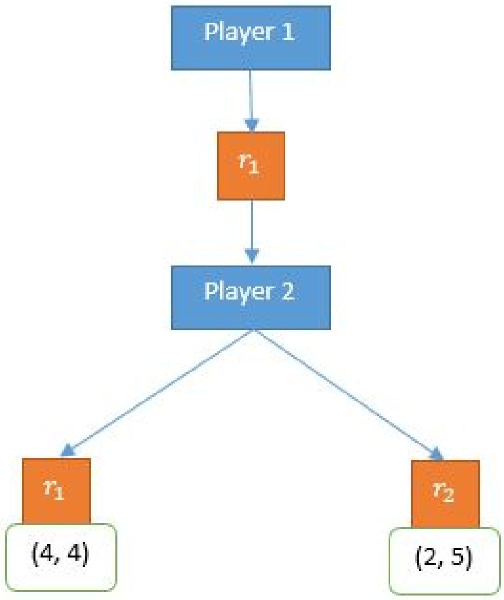


FIGURE 1.2: Extensive form game tree from **Figure 1.1**.

Let us see a second example now. In this case there are two players $\mathcal{N} = \{1, 2\}$, and four resources $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$ with zero constant costs $d_r = 0, \forall r \in R$ and weights $w_1 = 7, w_2 = 4, w_3 = 1, w_4 = 19$. Player 1 can choose either r_1 or resources

$\{r_1, r_2, r_3\}$. Player 2 can choose either resources $\{r_1, r_2\}$ or resources $\{r_3, r_4\}$. This example is shown in **Figure 1.3**.

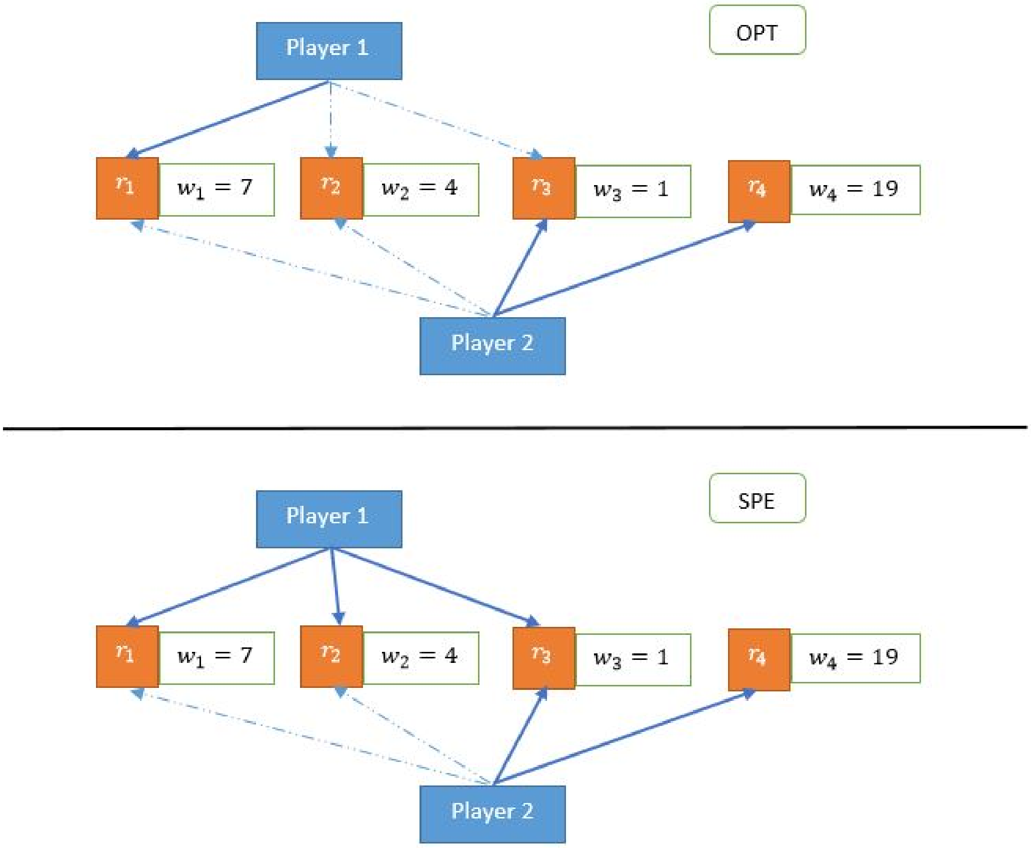


FIGURE 1.3: In this example the SPE is bad for both players. Fat lines correspond to chosen resources. Note that lines do not represent actions, only resources that can be chosen.

In the social optimum, player 1 chooses r_1, and player 2 chooses resources $\{r_3, r_4\}$, which yields a total cost of $7 + (1 + 19) = 27$. However, if player 1 were to choose r_1, then it is subgame perfect for player 2 to choose $\{r_1, r_2\}$, since this would yield him a cost of $7{\cdot}2+4 = 18 \leq 19+1 = 20$. This yields player 1 a cost of $7{\cdot}2 = 14$. Therefore it is subgame perfect for player 1 to choose $\{r_1, r_2, r_3\}$, since then it is subgame perfect for player 2 to choose $\{r_3, r_4\}$, which yields him a cost of $19+2{\cdot}1 = 21 \leq 2{\cdot}7+2{\cdot}4 = 22$. In this equilibrium, player 1 has cost of $7 + 4 + 2 \cdot 1 = 13$. Note that both players have higher cost in SPE than in OPT. Also, the subgame perfect equilibrium is not a Nash equilibrium of the corresponding strategic form game, since player 1 plays an action that is strictly dominated in the strategic form game. This example shows that the SPE can be bad for both players.

1.3.3 Price of Anarchy and Sequential Price of Anarchy

Since our goal is to evaluate the quality of subgame perfect equilibria by comparing the cost induced by a worst subgame perfect equilibrium to the optimal cost, we use the concept of *sequential price of anarchy* which was recently introduced by Paes Leme, Syrgkanis, and Tardos [17]. Generally, it is also known as the price of decentralization, and measures the quality of any Nash or subgame perfect equilibrium relative to the quality of a globally optimal allocation, OPT. By OPT we denote the outcome where the total cost over all players is minimized. This is a utilitarian global objective, meaning that the global objective is to minimize the sum of the costs of all players. The *sequential price of anarchy* of an instance I of a game is defined by [17] as

$$SPoA(I) = \max_{SPE \in SPE(I)} \frac{\mathrm{cost}_{\mathrm{total}}(SPE)}{\mathrm{cost}_{\mathrm{total}}(OPT)},$$

where SPE denotes "subgame perfect equilibrium", and OPT denotes the "globally optimal allocation", i.e. the action profile that gives the minimum cost over all players. For a class of games $\mathcal{I}$, the sequential price of anarchy is

$$SPoA = \sup_{I \in \mathcal{I}} SPoA(I).$$

When the class of games is clear from the context, we write $SPoA$ for simplicity.

In contrast, the *price of anarchy* is the same concept, but in the case where the players act simultaneously. It is defined by [3] as

$$PoA(I) = \max_{NE \in NE(I)} \frac{\mathrm{cost}_{\mathrm{total}}(NE)}{\mathrm{cost}_{\mathrm{total}}(OPT)},$$

where $NE(I)$ denotes the set of all Nash equilibria for instance I. The price of anarchy of a class of instances $\mathcal{I}$ is

$$PoA = \sup_{I \in \mathcal{I}} PoA(I).$$

Chapter 2

Previous Results and Research

2.1 Related Work

Game theory modeling and the analysis of equilibrium in congestion games provide insight in the performance of Internet congestion control and road transportation networks, to give some examples. Game theoretic frameworks provide a sound model for analyzing the performance of large networks formed out of independent, autonomous or non-engineered agents. An example of applications of these concepts is the bandwidth sharing model of the Internet [18] under selfish behaviors of agents, users, computers or network nodes. The Internet is a great place to see this game theoretical point of view, where interactions can be viewed as agents trying to maximize their own payoffs. The main reason to study this topic is to see the implications of selfish behavior and how certain worst case scenarios can be avoided, in other words, how can we set up the rules of a game such that we achieve a (close to) socially optimal result.

Congestion games play an important role in recent research quantifying the inefficiency of game-theoretic equilibria. By researching this topic, we try to understand how the parameters of a congestion game influence the inefficiency of its equilibria, and also we try to establish useful sufficient conditions that guarantee near-optimal or optimal equilibria. Most of the game theoretical frameworks consider selfish behavior of the users, but there is also research done in the case where players don't act selfishly, but altruistically [19].

As mentioned earlier, the sequential price of anarchy was introduced by Paes Leme et al. [17] as an alternative way to measure the costs of decentralization. Compared to the classical price of anarchy, in which players act simultaneously, introduced by Papadimitriou and Koutsopias [3], it avoids the "curse of simultaneity" inherent to certain games. Specifically for machine cost sharing games, generic unrelated machine scheduling games and generic consensus games, the *SPoA* is smaller than the *PoA* [17]. In some cases, myopic behavior leads to better equilibria compared to the farsighted behavior of subgame perfect equilibria [20]. For the class of congestion games, the price of anarchy is equal to 2 in the case of two players, and 2.5 in the case of three or more players [21, 22]. In the sequential version of these congestion games, the sequential price of anarchy is equal to 1.5 when we have two players, and $2\frac{63}{488} \approx 2.13$ in the case of three players [23]. The value for the three player case was reached using a linear program, by using simple combinatorial arguments to show that the worst case must be attained by some instance that is moderate in size, and then computing this worst case instance with a standard LP solver. In the case where there are more than 3 players, an exact value of the SPoA has not been found yet, but we know that a lower bound for the case of four players is 2.465521027, which was found by using the same linear programming approach with a limited number of resources and actions for each player [24]. Our main contribution corresponds to a higher lower bound for the case of four players along with some insights for the case of five or more players in the form of Theorem 3.1, Theorem 3.2 and Conjecture 1.

2.2 Theoretical background

In the previous section we mentioned that to compute the worst case instance of the SPoA certain combinatorial arguments are used. In this section we briefly discuss these combinatorial arguments in the form of Lemma 2.1 below. This lemma is explained in more detail and proved in [23]. Before showing the lemma, we have to use the following notation: Define the series

$$x_1 = 2, \quad \text{and} \quad x_i = 1 + \prod_{j<i} x_j, \ i \geq 2.$$

Note that $x_2 = 3,\ x_3 = 7,\ x_4 = 43$. Our analysis is based on the following Lemma 2.1.

Lemma 2.1. *For any instance I of an atomic congestion game, there exists an instance I' with $|\mathcal{A}_i| \leq x_i$ for any $i \in \mathcal{N}$, any two resources are not in the exact same sets of actions, $|\mathcal{R}| \leq 2^{\sum_{i\in\mathcal{N}}|\mathcal{A}_i|}$, and $d_r + w_r \leq n \cdot cost_{total}(OPT)$ for any $r \in \mathcal{R}$, such that $SPoA(I') = SPoA(I)$.* $\square$

Let us give some intuition on this lemma before we continue. First, we reduce the set of actions for each player to $|\mathcal{A}_i| \leq x_i$ by eliminating all actions that are not played either in *OPT* or in a fixed worst case *SPE* from *I* in the order of the players

$1, 2, ..., n$. For the first player, we reduce $|\mathcal{A}_1|$ to only two actions, one in OPT and one in a worst case SPE. For the second player, we restrict $\mathcal{A}_2$ to $x_2 \leq 3$ actions, the SPE actions in two information sets, one for each possible action of the first player, along with the action in OPT. More generally, for the kth player, we reduce $\mathcal{A}_k$ to at most $1 + \prod_{i<k} x_i$ actions, namely the subgame perfect actions of the fixed SPE in each of the at most $\prod_{i<k} x_i$ information sets, plus the action in OPT. Secondly, we reduce the number of resources to be considered by the fact that any two resources are not part of the exact same set of actions. Meaning that, if two resources r and r' are part of the same set of actions, we can merge them and create a combined resource r'' where the weight is $w_{r''} = w_r + w_{r'}$. Therefore, due to the fact that no two resources are part of the same set of actions, and since there are in total no more than $\sum_{i\in\mathcal{N}} |\mathcal{A}_i|$ actions, by the pigeonhole principle we may assume w.l.o.g. that there are no more than $2^{\sum_{i\in\mathcal{N}} |\mathcal{A}_i|}$ resources in a worst-case instance for the $SPoA$. Finally, we observe that resources r with $d_r + w_r > n \cdot \text{cost}_{\text{total}}(OPT)$ can be safely eliminated, as it cannot be subgame perfect for any player i to choose resource r: choosing OPT_i instead, the action that player i chooses in the optimal allocation, yields a cost of at most $n \cdot \text{cost}_{\text{total}}(OPT_i) \leq n \cdot \text{cost}_{\text{total}}(OPT) < d_r + w_r$

To conclude, in order to find an instance with worst $SPoA$ for four players, we only need to consider games with $|\mathcal{A}_1| \leq 2$, $|\mathcal{A}_2| \leq 3$, $|\mathcal{A}_3| \leq 7$, $|\mathcal{A}_4| \leq 43$, $|\mathcal{R}| \leq 2^{2+3+7+43} = 2^{55}$, with any two resources not in the exact same sets of actions, and $d_r + w_r \leq 4 \cdot \text{cost}_{\text{total}}(OPT)$ for any $r \in \mathcal{R}$.

2.3 Previously Known Values of PoA and SPoA

Previous results concerning $SPoA$ and PoA in general atomic congestion games are sumarized in Table 2.1. Note that for the case of four players we only have a lower bound instance.

Number of players	$SPoA$	PoA
n=2	1.5 ([23])	2 ([21, 22])
n=3	$2\frac{63}{488}$ ([23])	2.5 ([21, 22])
n=4	≥2.465521027 ([24])	2.5 ([21, 22])
$n \to \infty$	$\Omega(\sqrt{n})$ ([25])	2.5 (([21, 22])

Table 2.1: $SPoA$ and PoA values in general atomic congestion games

So far, in the general case, we know that $SPoA \leq PoA$ when $n = 2, 3$. When $n = 4$, although the lower bound on the $SPoA$ is 2.465521027, which is close to the value of the PoA of 2.5, we still don't know if the $SPoA$ will exceed the PoA. Therefore, given the recent result in [25], an interesting question to ask at this point would be:

For how many players does the $SPoA$ exceed the PoA? Given that the $SPoA$ increases monotonically with the number of players n (See Theorem 3.1), we therefore focus on the case with $n = 4$ players first.

2.4 ILP and LP Approaches

Using the AIMMS optimization modeling framework, we build a linear program (LP) for the case of three players, and an integer linear program (ILP) for the case of four or more players. This is explained next.

Let us first give an introduction to the LP formulation in the case of three players. In the formulation, we denote by $\mathcal{A}$ the set of all actions $\bigcup_{i \in N} \mathcal{A}_i$. The LP uses $2^{2+3+7} = 4096$ resources R, one for every combination of actions. We use binary parameters δ_{rA} to specify whether resource r is chosen in action A. For each resource r, we have decision variables d_r and w_r, the constant cost and weight of r, respectively. We use binary parameters, $x_{A_1}, x_{A_1 A_2}, x_{A_1 A_2 A_3}$ to determine which actions are subgame perfect for players 1, 2 and 3 respectively. For example, $x_{A_1 A_2} = 0$ whenever action A_2 is subgame perfect for player 2, anticipating a subgame perfect action of player 3, and knowing that player 1 has chosen action A_1, and $x_{A_1 A_2} = 1$ otherwise. We define the outcome where each player chooses her first action $\{A_{11}, A_{21}, A_{31}\}$ as the social optimum with total costs normalized to 1. For each player w.l.o.g., we define the $a+1-th$ action to be subgame perfect in the $a-th$ branch of the game tree. We denote by $v_A = \sum_{r \in \mathcal{R}} (d_r + w_r)\delta_{rA}$ the cost of a player that chooses action $A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ without taking any other players' actions into consideration (i.e. as if other players were absent). Next we denote by $o_{AA'} = \sum_{r \in \mathcal{R}} w_r \delta_{rA} \delta_{rA'}$ the additional costs that two players with actions $A, A' \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ incur due to overlap in resources. We use these auxiliary variables to determine the total cost of player i for $i \in \{1, 2, 3\}$, when players 1, 2 and 3 choose actions $(A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3$. This we denote by $cost_i(A_1, A_2, A_3)$. Finally, $cost_1(A_1), A_1 \in \mathcal{A}_1$ and $cost_2(A_1, A_2), (A_1, A_2) \in \mathcal{A}_1 \times \mathcal{A}_2$ determine the cost of actions of players 1 and 2 respectively, when successors play subgame perfect actions. For instance, $cost_2(A_1, A_2)$ denotes the cost of action A_2 for player 2, when player 1 chooses action A_1, and player 3 plays a subgame perfect action, given actions A_1 and A_2 of players 1 and 2 respectively. Finally, $cost_{SPE}$ determines the sum of costs for all players in the outcome corresponding to the subgame perfect equilibrium. The constraints that we need are: the definitions of v_A and $o_{AA'}$ for all actions A, A', the definition of the cost in each outcome for a player, the normalization of the costs such that the optimal solution is the outcome $\{A_{11}, A_{21}, A_{31}\}$ and has a total cost equial to 1, the constraints such that no player can improve from any subgame perfect action, the definition of

$cost_1(A_1)$ and $cost_2(A_1, A_2)$ and the final definition of $cost_{SPE}$. The objective is to maximize $cost_{SPE}$, since, due to normalization, this value equals the sequential price of anarchy. See Appendix A for the formulation.

Now that we know the formulation of the LP, to discuss the formulation of the ILP we will just discuss the differences. We will use the ILP for the cases where we have more than three players. Therefore we use the same setup as before but with variables, parameters and constraints for four players or more. In the ILP, instead of predetermining which actions are subgame perfect for players three and above, we use binary variables to let the ILP decide which actions are subgame perfect. In the case of four players this would mean that we have binary variables $x_{A_1A_2A_3}$ and $x_{A_1A_2A_3A_4}$. They have the same interpretation as in the LP model, except that they are variables instead of parameters. We introduce constraints to make sure that there exists at least one subgame perfect action for each player. We also introduce "bigM" constraints to bound the costs. Setting $M = n$, where n is the number of players, suffices as it is mentioned in [23].

The main difference between both formulations is that in the LP we give as input binary parameters that determine the subgame perfect actions for each player, while in the ILP we let the program decide which actions are subgame perfect for players three and above. This is necessary because we need to limit the actions and resources, and therefore cannot simply fix one path to be *SPE*. Therefore, the parameters in the LP become variables in the ILP. The formulations of the linear programs are given in the appendix, where we see the set up in a more detailed way.

Chapter 3

Our Results

3.1 LP model

The linear program for calculating the *SPoA* requires that we consider all the possible actions for each player as defined in section 2.2 along with all the possible resources $\mathcal{R} = 2^{|A|}$. For three players this is $|\mathcal{A}_1| = 2,\ |\mathcal{A}_2| = 3,\ |\mathcal{A}_3| = 7$ actions and $|\mathcal{R}| = 2^{2+3+7} = 2^{12} = 4096$ resources, while for four players it would mean we have to consider $|\mathcal{A}_1| = 2,\ |\mathcal{A}_2| = 3,\ |\mathcal{A}_3| = 7,\ |\mathcal{A}_4| = 43$ actions and $|\mathcal{R}| = 2^{2+3+7+43} = 2^{55}$ resources.

3.1.1 Testing Three Player Result

The only case where we can practically use the linear program without running into problems with memory or too long computational times is the case with three players. The full model of the three player case includes the actions $|\mathcal{A}_1| = 2,\ |\mathcal{A}_2| = 3,\ |\mathcal{A}_3| = 7$ for players one, two and three respectively, along with the resources $|\mathcal{R}| = 2^{2+3+7} = 2^{12} = 4096$. We have implemented this LP in order to test our model and make sure it calculates correctly the *SPoA*. After inputting LP model to our AIMMS LP and using the solver CPLEX 12.6.3 we obtain the same value of $2\frac{63}{488} \approx 2.13$ as found in [23].

3.2 Limiting the number of resources and actions available

When we test instances with $n > 3$ players, the input for the LP becomes exponentially large. Specifically, for the case of four players we would need $|\mathcal{A}_1| = 2,\ |\mathcal{A}_2| = 3,\ |\mathcal{A}_3| = 7,\ |\mathcal{A}_4| = 43,\ |\mathcal{R}| = 2^{2+3+7+43} = 2^{55}$. This would mean that we need a matrix with 2^{55} rows just to show which resources are part of which actions. Due to memory restrictions, we cannot even set up this LP. This is also limited by the number of rows that we can store in an Excel file which is 2^{20}. Even if we were able to represent this matrix in a comma separated value file, we would need a harddrive with a capacity in the order of petabytes. If we compare the cases where we have the action profiles $|\mathcal{A}_1| = 2,\ |\mathcal{A}_2| = 3,\ |\mathcal{A}_3| = 4,\ |\mathcal{A}_4| = 5$, when we increase the number of resources from 30 to $2^{14} = 16384$ the number of variables increases from 927 to 33633 and the number of non-zeros increases from 8722 to 1097382. At this stage we are still able to run the program, but now if we increased the number of resources to 2^{55} this would cause the variables and the non-zeros to increase exponentially causing the program to crash or not even start. That means that we cannot easily use the LP in order to compute the true value of the *SPoA* for $n = 4$ players. Instead, however, we can use the mentioned ILP to compute lower bounds.

3.2.1 Testing for Four Player Results

Before we mention experimental results for the $SPoA$ of four players, let us start by stating Theorem 3.1. The main idea of this theorem is that, if we have an instance I' of a game with n players with a certain *SPoA(I')*$= \alpha$, then we can create a new instance I'' where we add a new player $n+1$ to the game and give this player only one action that corresponds to a new additional resource that is only available to him, then the *SPoA* of this new instance is such that $SPoA(I'') \geq SPoA(I')$.

Theorem 3.1. *For any instance I' of an atomic congestion game with a set of players $|\mathcal{N}| = n$, a set of actions $|\mathcal{A}_i| \leq x_i$ for any $i \in \mathcal{N}$ where $x_i = 1 + \prod_{j<i} x_j$, any two resources are not in the exact same sets of actions, $|\mathcal{R}| \leq 2^{\sum_{i\in\mathcal{N}} |\mathcal{A}_i|}$, and $d_r + w_r \leq n \cdot cost_{total}(OPT)$ for any $r \in \mathcal{R}$, there exists an instance I'' where its set of resources is $|\mathcal{R}'| = |\mathcal{R}|+1$, its set of players is $\mathcal{N}' = \mathcal{N} \cup \{n+1\}$, its set of actions is $\mathcal{A}' = \mathcal{A} \cup \mathcal{A}_{n+1}$ where $|\mathcal{A}_{n+1}| = 1$ and $\delta_{r_{|\mathcal{R}'|}A} = 1$ for $A \in \mathcal{A}_{n+1}$ and 0 otherwise, such that SPoA$(I'') \geq$ SPoA(I').* □

The conclusion of Theorem 3.1 is that $SPoA$ of an atomic congestion game with $n+1$ players is at least the $SPoA$ of the atomic congestion game with n players.

Meaning that the $SPoA$ is monotonically increasing with respect to the number of players. Theorem 3.1 is used later in this section.

Now we dive into the experimental results. For the case where we have four players we first tried to reproduce the value reached in [24] of 2.465521027, but we ran into multiple problems which we describe later in section 3.2.1.1. After solving this problem we decided to plug in the full set of resources needed in the case where we have the set of actions $|\mathcal{A}_1| = 2,\ |\mathcal{A}_2| = 3,\ |\mathcal{A}_3| = 4,\ |\mathcal{A}_4| = 5$. The number of resources needed for this case is $|\mathcal{R}| = 2^{2+3+4+5} = 2^{14} = 16384$. After the first run on AIMMS we reached a value of the *SPoA*=2.550915006 which is higher than the one in [24] and also interestingly higher than the value of the *PoA* of 2.5 [21, 22]. In the so computed instance only 26 resources are used, while the instance presented in [24] requires 30 resources. We can see the values of the weights of the resources and which resources are part of which action in table 3.1. In this game there are 26 resources, along with their constant activation costs d_r (all 0s) and their weights w_r. The actions available to each player are $\mathcal{A}_1 = \{1.1, 1.2\}$, $\mathcal{A}_2 = \{2.1, 2.2, 2.3\}$, $\mathcal{A}_3 = \{3.1, 3.2, 3.3, 3.4\}$, $\mathcal{A}_4 = \{4.1, 4.2, 4.3, 4.4, 4.5\}$, each of which corresponds to a subset of resources that a player is able to choose. We set the action profile $(1.1, 2.1, 3.1, 4.1)$ as optimal, giving a total cost of 1, while the profile $(1.2, 2.3,\ 3.4, 4.3)$ is subgame perfect for each player, and gives a total cost of 2.550915006, so the value of *SPoA* of this particular instance of the game is 2.550915006. Therefore, the value of *SPoA* for the class of all atomic congestion games for four or more players is no less than 2.550915006.

If we combine this result where the value of the $SPoA$ for the class of all atomic congestion games for four players is no less than 2.550915006 with Theorem 3.1 we can state Theorem 3.2. The combination of the previous result with Theorem 3.1 means that the $SPoA$ for the class of all atomic congestion games with four or more players is greater than 2.550915006. We also know from [21, 22] that the PoA for the class of all atomic congestion games with three or more players is 2.5. Therefore by stating Theorem 3.2 we answer our initial question of: is the $SPoA$ always less than the PoA?

Theorem 3.2. *The SPoA for the class of all affine atomic congestion games is greater than 2.550915006, hence it exceeds the PoA for the same class of games when the number of players is* $n \geq 4$. □

We next propose a conjecture to further reduce the size of any worst-case instance. Specifically this conjecture reduces the number of resources to be considered. We reached this conjecture by observing the structure of the resources used in the optimal profile of actions.

r	w_r	1.1	1.2	2.1	2.2	2.3	3.1	3.2	3.3	3.4	4.1	4.2	4.3	4.4	4.5
1	0.05205691856					1	1								
2	0.1099709641		1						1		1				
3	0.02611302439		1						1	1	1				
4	0.0131944271		1	1				1				1			
5	0.008522346756	1								1		1			
6	0.03787835792	1			1					1		1			
7	0.8975899512												1		
8	0.0006288465773	1								1		1	1		
9	0.05118402503		1	1										1	
10	0.1768568568				1	1	1							1	
11	0.04051292995					1			1	1	1			1	
12	0.09764943052		1	1								1		1	
13	0.001420012759					1	1								1
14	0.005540433473	1			1				1	1					1
15	0.05001600494				1			1		1	1				1
16	0.04352095058							1	1	1	1				1
17	0.00995598074					1		1	1	1	1				1
18	0.02753303715		1	1				1				1			1
19	0.02594389417					1	1	1				1			1
20	0.08093732193	1							1	1		1			1
21	0.02731850285		1					1		1	1	1			1
22	0.05874948069					1		1		1	1	1			1
23	0.007512105864	1								1		1	1		1
24	0.02199260367				1	1	1			1				1	1
25	0.006302086999					1			1	1	1			1	1
26	0.01868945649					1	1			1		1		1	1

Table 3.1: Instance corresponding to *SPoA*= 2.550915006

Conjecture 1. For any instance I' of an atomic congestion game with $|\mathcal{A}_i| \leq x_i$ for any $i \in \mathcal{N}$ where $x_i = 1 + \prod_{j<i} x_j$, any two resources are not in the exact same sets of actions, $|\mathcal{R}| \leq 2^{\sum_{i \in \mathcal{N}} |\mathcal{A}_i|}$, and $d_r + w_r \leq n \cdot \text{cost}_{\text{total}}(OPT)$ for any $r \in \mathcal{R}$,, there exists an instance I'' where the resources $r_i \in A_i$ such that $A_i \in OPT$ appear in the optimal action of exactly one player and $SPoA(I') = SPoA(I'')$. $\square$

Meaning that $\sum_{A \in \mathcal{A}_{OPT}} \delta_{rA} = 1, \forall r \in \mathcal{R}$ where $\mathcal{A}_{OPT}$ is the optimal profile of actions. The main idea of the conjecture is that in the profile of actions in OPT, the players would not choose an action that contains a resource that has already been used by another player, meaning that resources in the socially optimal outcome are not congested at all. Choosing an action that contains a resource that has already been used by another player would only increase her cost. However a formal proof of the conjecture is still missing and left for future research.

3.2.1.1 Technical difficulties with ILP solving

Here we briefly discuss some of the technical difficulties that we encountered while doing the computational experiments.

If we give as input the data for 4 players and 30 resources from [24] to the LP solver, we were only able to reproduce the result when we explicitly gave the values of the weights (first four decimals) of the resources as an extra constraint with the CPLEX solver. Trying with the Gurobi solver, we were able to reach the same value after a really long computational time, but the solver was not able to prove that it was optimal. Yet another problem we had while trying to reproduce the result in [24] was when we added an extra constraint stating that the value of the $SPoA$ should be upper bounded by the number of players. After adding this constraint both solvers conclude that the solution is infeasible even though the bound holds. Therefore, adding the upper bound harms the performance of the branch and bound algorithm used by CPLEX and Gurobi. It is interesting to note also that when we keep this upper bound constrain, and add the constraint for the weight of the resources, where the weight has to be higher than the first four decimal numbers of the weights in [24] then the problem becomes feasible and the ILP solver finds the value of 2.465521027. Unfortunately, we still do not know why this happens, and leave it open to explain that seemingly inconsistent behaviour of the ILP solvers.

The results are shown in Table 3.2. We used two different solvers, and while using CPLEX we tried with two different feasibility tolerances. In case 1 we set which action uses which resource as a variable. In case 2 we fixed which action uses which resource as a parameter. In case 3 we add the upper and lower bound constrains of $SPoA \leq 4$ and $SPoA \geq 2.4$. In case 4 we add the constraints for lower bound of the weights such that the resource value has to be higher than the first 4 decimals from the weights reported in [24]. We see from Table 3.2 that the only cases where we reach the desired value of 2.465521027 are 1+3+4 and 2+3+4, that is, when we explicitly add the weight of the resources. In the Gurobi tests, we see that it also reaches the desired value, but the program keeps running without being able to prove optimality of this value. The different objective values observed in the table are caused by early terminations of the program related to memory problems. It is important to note that the moment CPLEX runs out of memory, it is not deterministic anymore. In order to solve the memory problem, we used CPLEX node files by setting the CPLEX option 'Node File' to 'On disk and compressed'. However it might take CPLEX a very long time to solve the problem to optimality. Our advice is to try looking for a tighter formulation of the problem such that you get a better LP relaxation objective at the root node.

	CPLEX 10^{-9} **tolerance**		**CPLEX** 10^{-6} **tolerance**		**GUROBI**	
case	**SPoA**	**Time (sec)**	**SPoA**	**Time (sec)**	**SPoA**	**Time (sec)**
1	2.426478491	79344.62 Then crashed	2.4070603323	71913	2.465521027	2300 continues until crash at 24216
1+3	Intermediate infeasible	7790	Infeasible	8894	Infeasible unbounded	11432
1+3+4	2.465521027	0.05	2.465521027	0.26	2.465521027	0.11
2	2.42647849112	88612.12	2.4070603323	66449.89	2.465521027	33164 and running
2+3	Intermediate infeasible	7725.11	Infeasible	9186	Still infeasible	218693
2+3+4	2.465521027	0.42	2.465521027	≤ 1	2.465521027	0.03

Table 3.2: Results and infeasibilities while testing data for 4 player case with weights as reported in [24]

Chapter 4

Non-Atomic Sequential Price of Anarchy for Two Players

Until now we have only considered sequential atomic and affine congestion games with pure strategies. That means that we have (implicitly) assumed that players are not allowed to randomize on their set of available actions, and neither to split their (unit) demand. In this chapter we briefly discuss sequential congestion games, where players are allowed to split their unit demand over their available actions. More specifically, if a player has two actions available, she can distribute her unit demand between both actions, where she splits the demand such that p_1 is assigned to action one and $(1 - p_1)$ is assigned to action two where $p_1 \in [0, 1]$. The main reason to study this type of games is to compare the results with the counterpart were splitting is not allowed. We first give an example and then continue with the general case for two players and discuss our results. In this section we denote the sequential price of anarchy for congestion games with splittable demand by $SPoA_N$. In this chapter splittable demand should not be mistaken for mixed strategies, where players assign a probability value to each action and maximize their expected utility.

4.1 Setup and Example

Let us start first with a simple example of a non-atomic congestion game with two players. In this case there are two players $\mathcal{N} = \{1, 2\}$, three resources $\mathcal{R} = \{r_1, r_2, r_3\}$ with zero constant costs $d_r = 0, \forall r \in R$ and weights $w_1 = 2, w_2 = 1, w_3 = 2$. Player 1 can choose either r_1 or r_2. Player 2 can choose either r_2 or r_3. This example is shown in **Figure 4.1**. In the optimal outcome OPT, both players split their demand between both available resources for them such that $p_1 = p_2 = \frac{1}{2}$. Player 1 has a cost of $cost_1(OPT) = 1.5$, player 2 has a cost of $cost_2(OPT) = 1.5$ and the social cost is $cost(OPT) = 3$. While in the SPE player 1 chooses r_2 with full load of 1 and player 2 splits her demand between both resources available with p_2. Therefore player 1 has a cost of $cost_1(SPE) = 1 + p_2$, player 2 has a cost of $cost_2(SPE) = (1 + p_2)p_2 + 2(1 - p_2) = 2 + {p_2}^2 - p_2$ and the social cost is $cost(SPE) = 3 + p_2^2$. Player 2 aims to minimize $cost_2(SPE)$ so let us find the value of p_2. To minimize $cost_2(SPE) = 2 + {p_2}^2 - p_2$ we take the first derivative and set it to zero such that $2p_2 - 1 = 0$. Therefore $p_2 = \frac{1}{2}$. Then $cost_1(SPE|p_2 = \frac{1}{2}) = 1 + 0.5 = 1.5$, $cost_2(SPE|p_2 = \frac{1}{2}) = 2 + 0.25 - 0.5 = 1.75$ and the social cost is $cost(SPE|p_2 = \frac{1}{2}) = 1.5 + 1.75 = 3.25$. Therefore the $SPoA_N = \frac{3.25}{3} = 1.0833$. In the atomic version of this game, the OPT outcome would mean that player 1 chooses r_2 and player 2 chooses r_3 with a social cost of $cost(OPT) = 3$, while in the SPE outcome both players would choose r_2 with a social cost of $cost(SPE) = 4$. Therefore the $SPoA = \frac{4}{3} = 1.3333$. Meaning that $SPoA_N < SPoA$ for this instance of a congestion game. In this case we have that the non-atomic version is more efficient. It is important to notice in the SPE that player 1 would not split her demand over both actions.

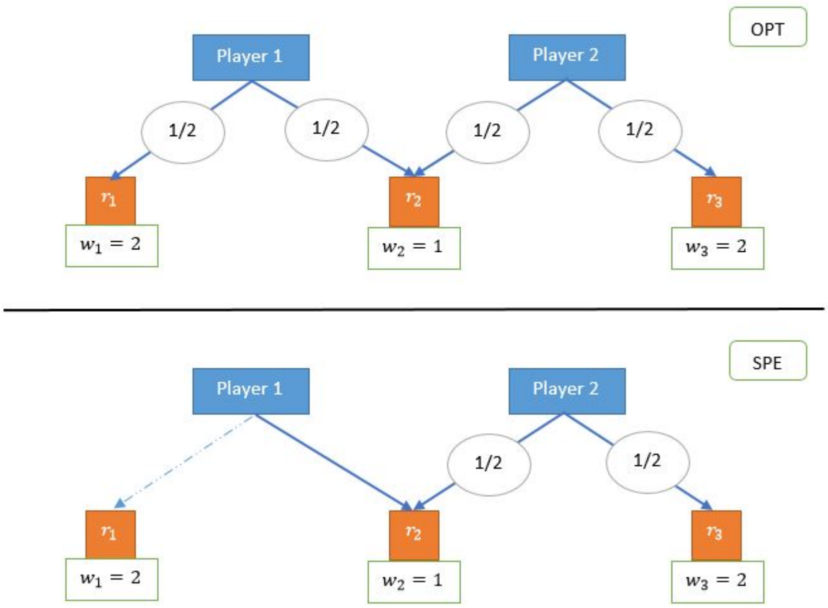


FIGURE 4.1: Non-atomic congestion game example with two players

Now let us define the general setting for two players $\mathcal{N} = \{1, 2\}$, and three resources $\mathcal{R} = \{r_1, r_2, r_3\}$ with zero constant costs $d_r = 0, \forall r \in R$ and weights $w_1 = a, w_2 = 1, w_3 = b$. Player 1 can choose either r_1 with $1 - p_1$ or r_2 with p_1 where $0 \leq p_1 \leq 1$. Player 2 can choose either r_2 with p_2 or r_3 with $1 - p_2$ where $0 \leq p_2 \leq 1$.

The cost for player 1 given (p_1, p_2) is $cost_1(p_1, p_2) = (1-p_1)a + p_1(p_1+p_2)$. The cost for player 2 given (p_1, p_2) is $cost_2(p_1, p_2) = p_2(p_1+p_2) + (1-p_2)b$. The social cost would be $cost(p_1, p_2) = (1-p_1)a + (p_1+p_2)^2 + (1-p_2)b$. We can see the decision tree in **Figure 4.2**.

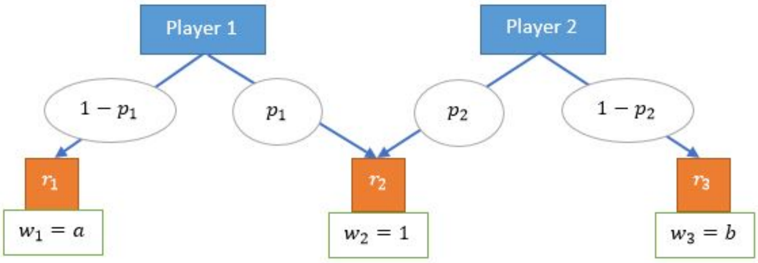


FIGURE 4.2: General non-atomic congestion game with two players and three resources

In order to study this general case, we set different values for the weights of resources $(r_1, r_3) = (a, b)$. We see most of the possibilities in Figure 4.3[1]. From Figure 4.3 we see that in some cases the OPT outcome will be the same as the SPE outcome. Case 0 is the general case where we do not assign values to a or b. The solution for these cases is marked with X. In this situation we will have a $SPoA_N = 1$. These cases are 1,2 and 4. We notice that in this situation the players would play atomic actions. While in the rest of the cases, the OPT outcome is different than the SPE outcome, where at least one of them splits his actions making it non-atomic and the $SPoA_N > 1$. In each of these cases we give an example with specific values assigned to a and b so that we can see the actual valye of the $SPoA_N$.

4.1.1 Interpretation

The two player non-atomic congestion game where each player is given the option of choosing her own resource or sharing a resource with the other player can be interpreted as follows: How do we set the weight of each resource in order to make the players choose their own resource or make them share the common resource? Depending on the value of the weights of each resource they will be willing to share the middle resource or they split their demand minimizing their own cost. Even though we exhaust all the possible cases for the values of the weights, this simple two player model is not enough to reach a conclusion about the $SPoA_N$. It is however valid as a lower bound for two players non-atomic congestion games. In order to reach a conclusion on this type of

[1] We do not include the following cases in the table: $3 < a, b < 4$, $2 < a < 3 \& 3 < b < 4$ and $2 < b < 3 \& 3 < a < 4$.

Case	a, b	Sol.	p_1	p_2	$Cost(p_1, p_2)$	$Cost_1(p_1, p_2)$	$Cost_2(p_1, p_2)$	SPoA
0	a,b	OPT	$\min(\frac{a}{2} - p_2, 1)$	$1 \geq p_2 \geq 1 - \frac{a}{2}$	$(1-p_1)a + (p_1+p_2)^2 + (1-p_2)b$	$(1-p_1)a + p_1(p_1+p_2)$	$(1-p_2)b + p_2(p_1+p_2)$	SPoA
			$1 \geq p_1 \geq 1 - \frac{b}{2}$	$\min(\frac{b}{2} - p_1, 1)$				
		SPE	1	$(b-1)/2$				
1	a,b>=4	X	1	1	4	2	2	1
2	a,b<1	X	0	0	a+b	a	b	1
3	a,b=2	OPT	½	½	3	1.5	1.5	1.08333
		SPE	1	½	3.25	1.5	1.75	
4	a,b=1	X	0	1	2	1	1	1
			1	0				
			0	0				
5	a,b=3	OPT	1	½	3.75	1.5	2.25	1.06666
			½	1		2.25	1.5	
			¾	¾		1.875	1.875	
			$\min(1.5 - p_2, 1)$	$p_2 \geq 0.5$		$(1-p_1)3 + p_1(p_1+p_2)$	$(1-p_2)3 + p_2(p_1+p_2)$	
			$p_1 \geq 0.5$	$\min(1.5 - p_1, 1)$				
		SPE	1	1	4	2	2	
6	2<a<3 2<b<3	OPT	$\min(\frac{a}{2} - p_2, 1)$	$1 \geq p_2 \geq 1 - \frac{a}{2}$	$(1-p_1)a + (p_1+p_2)^2 + (1-p_2)b$	$(1-p_1)a + p_1(p_1+p_2)$	$(1-p_2)b + p_2(p_1+p_2)$	>1
		SPE	1	$(b-1)/2$	$(8b - b^2 + 1)/4$	$1 + (b-1)/2$	$b - (b-1)(b-2)/2$	
	a,b=2.5	OPT	$1.25 - p_2$	$p_2 \geq 0.25$	3.4375	$(1-p_1)2.5 + p_1(p_1+p_2)$	$(1-p_2)2.5 + p_2(p_1+p_2)$	1.0727
			$p_1 \geq 0.25$	$1.25 - p_1$	3.4375	$(1-p_1)2.5 + p_1(p_1+p_2)$	$(1-p_2)2.5 + p_2(p_1+p_2)$	
		SPE	1	¾	3.6875	1.75	1.9375	
7	1<a,b<2	OPT	$\min(\frac{a}{2} - p_2, 1)$	$1 \geq p_2 \geq 1 - \frac{a}{2}$	$(1-p_1)a + (p_1+p_2)^2 + (1-p_2)b$	$(1-p_1)a + p_1(p_1+p_2)$	$(1-p_2)b + p_2(p_1+p_2)$	>1
		SPE	1	$(b-1)/2$	$(1 + 4b - b^2)/4$	$1 + (b-1)/2$	$b - (b-1)(b-2)/2$	
	a,b=1.5	OPT	1	0	2.5	1	1.5	1.075
			0	1		1.5	1	
			$\max(0, 0.75 - p_2)$	p_2		$(1-p_1)a + p_1(p_1+p_2)$	$(1-p_2)b + p_2(p_1+p_2)$	
			1/2	1/2		1.25	1.25	
		SPE	1	1/4	2.6875	1.25	1.4375	

FIGURE 4.3: Table with values for two player non-atomic congestion games

games, we would need to develop a worst-case scenario instance where we consider the necessary number of resources and actions. We leave this for further research.

Chapter 5

Discussion and Further Research

In this section we want to discuss the results we have reached. We would also like to mention some advice for further research.

First of all, it is interesting to discuss the fact that the current lower bound for the *SPoA* of four players that we have achieved (2.550915006) is higher than the result of the *PoA* for the simultanous congestion games (2.5). In general, we knew that $SPoA \leq PoA$ when $n = 2, 3$. But for $n \geq 4$ now we have that $SPoA > PoA$. This contradicts the conjecture from the conclusions in [23], where it is conjectured that for atomic congestion games, the sequential price of anarchy is lower than the price of anarchy. The question had been answered asymptotically for $n \to \infty$ in [25], even showing that $SPoA \in \Omega(\sqrt{n})$, however the precise "tipping point" for the comparison between *PoA* and *SPoA* was open so far. It would be interesting to know why this happens after we increase the number of players from three to four. The interpretation of this result, means that if we have a congestion game with four players, letting the players choose simultaneously would be better than letting them play sequentially from the social cost point of view.

It is also of high importance to prove or disprove Conjecture 1. This would shed a lot of light on the structure of congestion games and would decrease the size of the worst-case instances in order to find better lower bounds for the *SPoA*. It would also be useful to derive more combinatorial results to improve our ILP model in order to find the exact value of the *SPoA* for all the cases.

Other approaches that should be considered could be the dual approach [26] in order to find a lower upper bound of the *SPoA*. The LP approach is used to find the exact value of the *SPoA* for the three player atomic congestion game. The ILP is used to find lower bounds of the *SPoA* in the case where we give do not give as input the whole instance of a congestion game as described in Lemma 2.1. The ILP could, in principle, also be used to find the exact value of the *SPoA* in the case where we give the full model of the problem, but the huge size of the model leads to large memory usage and long computation times, therefore, practically, we can only compute lower bounds. We know from [23] that the theoretical upper bound of *SPoA* is the number of players n.

We also noticed that when we give as input to the ILP a big number of resources, most of these resources would not actually be necessary in the worst-case instance. Meaning that some of these resources are not used at all and should not be considered. A good way to approach this problem could be by using column generation. Column generation is a very efficient algorithm for solving linear programs with a large number of variables. The main idea is that these linear programs are too large to consider all the variables explicitly. Since most of the variables will be non-basic and assume a value of zero in the optimal solution, only a subset of these variables is needed to be considered when solving the problem. Column generation uses this idea to generate only the varaibles which have the potential to improve the objective function, and proving optimality by use of duality.

Another important topic to discuss is the problems we had while trying to find new lower bound instances and reproduce previous results. This could be due to the solver configuration, the different solvers we used or the actual structure of the linear program. The fact that we reached different results with the same input on the same computer without changing anything in the linear program makes us think that there is a problem which we are not aware of yet. This could be due to the computational power of the computer or just the way the solver tries to find an optimal solution. We advice to try to use different configurations of the solver or use a completely different solver. Another approach could be to make a completely new model for the *SPoA*.

Finally, with respect to the model for the two player non-atomic congestion game, further research is needed in order to reach a conclusion on the value of the $SPoA_N$ for two players. As we mentioned earlier, it is needed to develop a worst-case instance with the necessary number of resources and actions, since our example can only be used as a lower bound. It would also be interesting to consider mixed strategies, instead of splittable demand, where players assign probabilities to each action and they minimize their expected costs.

Chapter 6

Conclusions

In this thesis, we conclude that in the case of atomic congestion games with linear latency functions we have the following results represented in table 6.1.

Number of players	***SPoA***	***PoA***
n=2	1.5 ([23])	2 ([21,22])
n=3	$2\frac{63}{488}$ ([23])	2.5 ([21,22])
$n \geq 4$	$\geq$2.5509150067	2.5 ([21,22])
$n \to \infty$	$\Omega(\sqrt{n})$ ([25])	2.5 (([21, 22])

TABLE 6.1: *SPoA* and *PoA* values in general atomic congestion games

We thereby answer our initial question of: For how many players does the *SPoA* exceed the *PoA*? The main insight of this thesis is that the $SPoA$ for atomic congestion games with four or more players is higher than the PoA.

Appendices

Appendix A

LP Formulation for 3 players

- **Objective**

$$\text{maximize:} \quad \text{cost}_{SPE}.$$

- **Parameters**

$$\delta_{rA} = \begin{cases} 1, & \text{if } r \in A, \\ 0, & \text{if } r \notin A. \end{cases} \quad r \in \mathcal{R}, A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3.$$

$$x_{A_{11}} = 1, \quad x_{A_{12}} = 0.$$

$$x_{A_{11}A_{21}} = 1, \quad x_{A_{11}A_{22}} = 0, \quad x_{A_{11}A_{23}} = 1,$$

$$x_{A_{12}A_{21}} = 1, \quad x_{A_{12}A_{22}} = 1, \quad x_{A_{12}A_{23}} = 0.$$

$$x_{A_1A_2A_3} = \begin{cases} 0, & \text{if it is subgame perfect for Player 3 to choose } A_3 \\ & \text{and knowing that Player 1 chooses } A_1 \\ & \text{and Player 2 chooses } A_2, \\ 1, & \text{otherwise.} \end{cases}$$

$$((A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3).$$

- **Variables**

$$d_r, \quad r \in \mathcal{R}.$$

$$w_r, \quad r \in \mathcal{R}.$$

$$v_A, \quad A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3.$$

$$o_{AA'}, \quad A, A' \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3.$$

$$\mathrm{cost}_1(A_1), \quad A_1 \in \mathcal{A}_1.$$

$$\mathrm{cost}_2(A_1, A_2), \quad (A_1, A_2) \in \mathcal{A}_1 \times \mathcal{A}_2.$$

$$\mathrm{cost}_3(A_1, A_2, A_3) \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\mathrm{cost}_{total}(A_1, A_2, A_3), \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

cost_{SPE} : the total cost in a subgame perfect equilibrium.

- **Constraints**

$$\mathrm{cost}_{OPT} = 1.$$

$$v_A = \sum_{r \in \mathcal{R}} (d_r + w_r)\delta_{rA}, \quad A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4.$$

$$o_{AA'} = \sum_{r \in \mathcal{R}} w_r \delta_{rA} \delta r A', \quad A, A' \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$$

$$\mathrm{cost}_1(A_1, A_2, A_3) = v_{A_1} + o_{A_1A_2} + o_{A_1A_3}, \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\mathrm{cost}_2(A_1, A_2, A_3) = v_{A_2} + o_{A_2A_1} + o_{A_2A_3}, \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\mathrm{cost}_3(A_1, A_2, A_3) = v_{A_3} + o_{A_3A_1} + o_{A_3A_2}, \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\mathrm{cost}_{\mathrm{total}}(A_1, A_2, A_3) = \sum_{i=1}^{3} \mathrm{cost}_i(A_1, A_2, A_3), \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\mathrm{cost}_{\mathrm{total}}(A_{11}, A_{21}, A_{31}) = \mathrm{cost}_{OPT}.$$

$$\mathrm{cost}_{\mathrm{total}}(A_1, A_2, A_3) \geq \mathrm{cost}_{OPT}, \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3$$

$$\mathrm{cost}_1(A_1) \leq \mathrm{cost}_1(A'), \quad A_1, A' \in \mathcal{A}_1 and A_1 is SPE$$

$$\mathrm{cost}_2(A_1, A_2) \leq \mathrm{cost}_2(A_1, A'), \quad A_1 \in \mathcal{A}_1,\ A_2, A' \in \mathcal{A}_2 and A_1 A_2 are SPE.$$

$$\mathrm{cost}_3(A_1, A_2, A_3) \leq \mathrm{cost}_3(A_1, A_2, A'), \quad A_1 \in \mathcal{A}_1,\ A_2 \in \mathcal{A}_2,\ A_3, A' \in \mathcal{A}_3 and A_1 A_2 A_3 are SPE.$$

$$\mathrm{cost}_1(A_1) = \mathrm{cost}_1(A_1, A_2, A_3), \quad A_1 \in \mathcal{A}_1, A_2 \in \mathcal{A}_2, A_3 \in \mathcal{A}_3 \text{where} A_2 \text{and} A_3 \text{are SPE}$$

$$\mathrm{cost}_2(A_1, A_2) = \mathrm{cost}_3(A_1, A_2, A_3), \quad A_1 \in \mathcal{A}_1,\ A_2 \in \mathcal{A}_2 \text{where} A_3 \text{is SPE}$$

$$\mathrm{cost}_{SPE} = \mathrm{cost}_1(A_{12}, A_{23}, A_{37}) + \mathrm{cost}_2(A_{12}, A_{23}, A_{37}) + \mathrm{cost}_3(A_{12}, A_{23}, A_{37})$$

Appendix B

ILP Formulation for 4 players

Here we present the MIP model we use in **Section 2** for games with 4 players. We denote the collections of possible actions by $\mathcal{A}_1 = \{A_{11}, A_{12}\}$, $\mathcal{A}_2 = \{A_{21}, A_{22}, A_{23}\}$, $\mathcal{A}_3 = \{A_{31}, A_{32}, A_{33}, A_{34}\}$, $\mathcal{A}_4 = \{A_{41}, A_{42}, A_{43}, A_{44}, A_{45}\}$. Each action is a subset of the set $\mathcal{R}$ of resources. We denote an action profile by $\boldsymbol{A} = (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4$, and a resource by $r \in \mathcal{R}$. Due to the argument of **Lemma 1**, we only need to consider at most $2^{2+3+4+5} = 2^{14} = 16384$ resources.

Let $A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4$. We use binary parameters δ_{rA} to specify whether a resource r is part of an action A. For each resource r, we use variables d_r and w_r to denote the constant activation cost and weight, respectively. $v_A = \sum_{r \in \mathcal{R}} (d_r + w_r)\delta_{rA}$ is the cost of a player that chooses action A without taking other players' choices into consideration. $o_{AA'} = \sum_{r \in \mathcal{R}} w_r \delta_{rA} \delta_{rA'}$ is the additional cost that two players incur due to overlap in resources. Given an action profile $\boldsymbol{A} = (A_1, A_2, A_3, A_4)$, we have $\text{cost}_1 = v_{A_1} + o_{A_1A_2} + o_{A_1A_3} + o_{A_1A_4}, ..., \text{cost}_4 = v_{A_4} + o_{A_4A_1} + o_{A_4A_2} + o_{A_4A_3}$, and $\text{cost}_{\text{total}} = \sum_{i=1}^{4} \text{cost}_i$.

We use binary parameters, $x_{A_1}, x_{A_1A_2}$, and binary variables, $x_{A_1A_2A_3}, x_{A_1A_2A_3A_4}$, to determine which actions are subgame perfect.[1]For example, $x_{A_{12}A_{23}} = 0$ iff. it is subgame perfect for Player 2 to choose A_{23}, anticipating subgame perfect actions of Player 3 and Player 4, and knowing that Player 1 has chosen action A_{12}. $\text{cost}_1(A_1)$, $\text{cost}_2(A_1, A_2)$, $\text{cost}_3(A_1, A_2, A_3)$ determine the cost of Player 1, Player 2, and Player 3, respectively, when successors play subgame perfect. For example, $\text{cost}_2(A_{12}, A_{23})$ is the cost of Player 2, when Player 1 chooses A_{12}, Player 2 chooses A_{23}, and Player 3 and 4 plays subgame perfect, knowing the actions of Player 1 and 2.

[1]We use variables for $x_{A_1A_2A_3}$, because we cannot set manually which action of Player 3 is subgame perfect, since $|\mathcal{A}_3| = 4 < 7 = x_3$ in **Lemma 2.1**. It is shown clearly in [23]. Same reasoning also applies for $x_{A_1A_2A_3A_4}$.

At last, due to the symmetry of different actions, we simply set $(A_{11}, A_{21}, A_{31}, A_{41})$ to be the action profile which yields the optimal total cost, which we set to be 1. The variable cost_{SPE} is the total cost in a subgame perfect equilibrium. It is the solver's work to find the largest value of cost_{SPE} over all possible games, as well as the certain game which gives the largest cost_{SPE}. This largest value is exactly what we look for.

- **Objective**

$$\text{maximize:} \quad \text{cost}_{SPE}.$$

- **Parameters** [2]

$$\text{cost}_{OPT} = 1.$$

$$\delta_{rA} = \begin{cases} 1, & \text{if } r \in A, \\ 0, & \text{if } r \notin A. \end{cases} \quad r \in \mathcal{R}, A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4.$$

$$x_{A_{11}} = 1, \quad x_{A_{12}} = 0.$$

$$x_{A_{11}A_{21}} = 1, \quad x_{A_{11}A_{22}} = 0, \quad x_{A_{11}A_{23}} = 1,$$

$$x_{A_{12}A_{21}} = 1, \quad x_{A_{12}A_{22}} = 1, \quad x_{A_{12}A_{23}} = 0.$$

- **Variables**

$$x_{A_1A_2A_3} = \begin{cases} 0, & \text{if it is subgame perfect for Player 3 to choose } A_3\text{, given that Player 4} \\ & \text{ chooses subgame perfect action, and knowing that Player 1 chooses } A_1 \\ & \text{and Player 2 chooses } A_2, \\ 1, & \text{otherwise.} \end{cases}$$

$$((A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3).$$

$$x_{A_1A_2A_3A_4} = \begin{cases} 0, & \text{if it is subgame perfect for Player 4 to choose } A_4\text{, knowing that Player 1} \\ & \text{chooses } A_1\text{, Player 2 chooses } A_2\text{, and Player 3 chooses } A_3, \\ 1, & \text{otherwise.} \end{cases}$$

$$((A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4).$$

$$d_r, \quad r \in \mathcal{R}.$$

$$w_r, \quad r \in \mathcal{R}.$$

[2] The initial value we set for δ_{rA} is shown in **Table B.1**.

$$v_A, \quad A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4.$$

$$o_{AA'}, \quad A, A' \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4.$$

$$\text{cost}_1(A_1), \quad A_1 \in \mathcal{A}_1.$$

$$\text{cost}_2(A_1, A_2), \quad (A_1, A_2) \in \mathcal{A}_1 \times \mathcal{A}_2.$$

$$\text{cost}_3(A_1, A_2, A_3), \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\text{cost}_1(A_1, A_2, A_3, A_4), \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_2(A_1, A_2, A_3, A_4), \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_3(A_1, A_2, A_3, A_4), \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_4(A_1, A_2, A_3, A_4), \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_{\text{total}}(A_1, A_2, A_3, A_4), \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_{SPE} : \text{the total cost in a subgame perfect equilibrium.}$$

- **Constraints**

$$v_A = \sum_{r \in \mathcal{R}} (d_r + w_r)\delta_{rA}, \quad A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4.$$

$$o_{AA'} = \sum_{r \in \mathcal{R}} w_r \delta_{rA} \delta r A', \quad A, A' \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4.$$

$$\text{cost}_1(A_1, A_2, A_3, A_4) = v_{A_1} + o_{A_1 A_2} + o_{A_1 A_3} + o_{A_1 A_4}, \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_2(A_1, A_2, A_3, A_4) = v_{A_2} + o_{A_2 A_1} + o_{A_2 A_3} + o_{A_2 A_4}, \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_3(A_1, A_2, A_3, A_4) = v_{A_3} + o_{A_3 A_1} + o_{A_3 A_2} + o_{A_3 A_4}, \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_4(A_1, A_2, A_3, A_4) = v_{A_4} + o_{A_4 A_1} + o_{A_4 A_2} + o_{A_4 A_3}, \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_{\text{total}}(A_1, A_2, A_3, A_4) = \sum_{i=1}^{4} \text{cost}_i(A_1, A_2, A_3, A_4), \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\text{cost}_{\text{total}}(A_{11}, A_{21}, A_{31}, A_{41}) = \text{cost}_{OPT}.$$

$$\text{cost}_{\text{total}}(A_1, A_2, A_3, A_4) \geq \text{cost}_{OPT}, \quad (A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4.$$

$$\sum_{A_3 \in \mathcal{A}_3} x_{A_1 A_2 A_3} \leq |\mathcal{A}_3| - 1, \quad (A_1, A_2) \in \mathcal{A}_1 \times \mathcal{A}_2.$$

$$\sum_{A_4 \in \mathcal{A}_4} x_{A_1 A_2 A_3 A_4} \leq |\mathcal{A}_4| - 1, \quad (A_1, A_2, A_3) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3.$$

$$\text{cost}_1(A_1) \leq \text{cost}_1(A') + 4 \cdot \text{cost}_{\text{OPT}} \cdot x_{A_1}, \quad A_1, A' \in \mathcal{A}_1.$$

$$\mathrm{cost}_2(A_1, A_2) \le \mathrm{cost}_2(A_1, A') + 4 \cdot \mathrm{cost}_{\mathrm{OPT}} \cdot x_{A_1A_2}, \quad A_1 \in \mathcal{A}_1,\ A_2, A' \in \mathcal{A}_2.$$

$$\mathrm{cost}_3(A_1, A_2, A_3) \le \mathrm{cost}_3(A_1, A_2, A') + 4 \cdot \mathrm{cost}_{\mathrm{OPT}} \cdot x_{A_1A_2A_3}, \quad A_1 \in \mathcal{A}_1,\ A_2 \in \mathcal{A}_2,\ A_3, A' \in \mathcal{A}_3.$$

$$\mathrm{cost}_4(A_1, A_2, A_3, A_4) \le \mathrm{cost}_4(A_1, A_2, A_3, A') + 4 \cdot \mathrm{cost}_{\mathrm{OPT}} \cdot x_{A_1A_2A_3A_4},$$

$$(A_1 \in \mathcal{A}_1,\ A_2 \in \mathcal{A}_2,\ A_3 \in \mathcal{A}_3,\ A_4, A' \in \mathcal{A}_4).$$

$$|\mathrm{cost}_1(A_1) - \mathrm{cost}_1(A_1, A_2, A_3, A_4)| \le 4 \cdot \mathrm{cost}_{\mathrm{OPT}} \cdot (x_{A_1A_2A_3} + x_{A_1A_2A_3A_4}),$$

$$((A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4,\ x_{A_1A_2} = 0).$$

$$|\mathrm{cost}_2(A_1, A_2) - \mathrm{cost}_2(A_1, A_2, A_3, A_4)| \le 4 \cdot \mathrm{cost}_{\mathrm{OPT}} \cdot (x_{A_1A_2A_3} + x_{A_1A_2A_3A_4}),$$

$$((A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4).$$

$$|\mathrm{cost}_3(A_1, A_2, A_3) - \mathrm{cost}_3(A_1, A_2, A_3, A_4)| \le 4 \cdot \mathrm{cost}_{\mathrm{OPT}} \cdot x_{A_1A_2A_3A_4},$$

$$((A_1, A_2, A_3, A_4) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3 \times \mathcal{A}_4).$$

	A_{11}	A_{12}	A_{21}	A_{22}	A_{23}	A_{31}	A_{32}	A_{33}	A_{34}	A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
r_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r_2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
r_3	0	1	0	0	0	0	0	0	0	0	0	0	0	0
r_4	1	1	0	0	0	0	0	0	0	0	0	0	0	0
r_5	0	0	1	0	0	0	0	0	0	0	0	0	0	0
r_6	1	0	1	0	0	0	0	0	0	0	0	0	0	0
r_7	0	1	1	0	0	0	0	0	0	0	0	0	0	0
r_8	1	1	1	0	0	0	0	0	0	0	0	0	0	0
r_9	0	0	0	1	0	0	0	0	0	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{16384}	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE B.1: The initial value we set for δ_{rA}

Appendix C

Costs Data of the Game Yielding the $SPoA$ for Games with 4 players

Here we present the costs data of the game yielding the $SPoA$ of 2.550915006, highlighting the subgame perfect actions. We use "1.1" for A_{11}, ..., "4.5" for A_{45} in Table C.1:

				cost1	cost2	cost3	cost4	cost total
			4.1	0.141019413	0.18956092	0.296959743	0.372459925	1
			4.2	0.276498392	0.327937815	0.341593093	0.723046433	1.669075732
		3.1	4.3	0.149160365	0.18956092	0.296959743	0.913871856	1.549552883
			4.4	0.141019413	0.338394375	0.514498659	0.779559762	1.773472209
			4.5	0.235009274	0.217093957	0.36500571	0.575000738	1.392109678
			4.1	0.141019413	0.230288384	0.486520662	0.562020845	1.419849304
			4.2	0.276498392	0.368665279	0.449699084	0.831152424	1.926015179
		3.2	4.3	0.149160365	0.230288384	0.296959743	0.913871856	1.590280348
			4.4	0.141019413	0.37912184	0.296959743	0.562020845	1.37912184
			4.5	0.235009274	0.257821421	0.539997594	0.749992622	1.78282091
	2.1		**4.1**	0.227497168	0.18956092	0.645707384	0.608835862	1.671601334
			4.2	0.362976147	0.327937815	0.49026877	0.759350404	1.940533135
		3.3	4.3	0.23563812	0.18956092	0.409331448	0.913871856	1.748402344
			4.4	0.227497168	0.338394375	0.456146465	0.608835862	1.63087387

Continued on next page

				cost1	cost2	cost3	cost4	cost total
			4.5	0.321487029	0.217093957	0.555588221	0.653211544	1.747380752
		3.4	**4.1**	0.282038825	0.18956092	0.847698808	0.634948886	1.954247439
			4.2	0.417517804	0.327937815	0.825446265	0.918649501	2.489551385
			4.3	0.290179778	0.18956092	0.593350799	0.922012809	1.995104305
			4.4	0.282038825	0.338394375	0.672706923	0.649517922	1.942658046
			4.5	0.376028686	0.217093957	0.915744775	0.837489699	2.346357117
	2.2	3.1	**4.1**	0.184438204	0.584568514	0.495809203	0.42247593	1.687291851
			4.2	0.319917183	0.572430867	0.540442554	0.622547896	2.055338499
			4.3	0.192579156	0.534552509	0.495809203	0.913871856	2.136812724
			4.4	0.184438204	0.733401969	0.71334812	0.829575767	2.46076406
			4.5	0.278428065	0.612101551	0.56385517	0.625016743	2.079401529
		3.2	**4.1**	0.184438204	0.435735058	0.495809203	0.61203685	1.728019315
			4.2	0.319917183	0.423597411	0.458987625	0.730653887	1.933156106
			4.3	0.192579156	0.385719053	0.306248283	0.913871856	1.798418349
			4.4	0.184438204	0.584568514	0.306248283	0.61203685	1.687291851
			4.5	0.278428065	0.463268095	0.549286134	0.800008627	2.090990922
		3.3	4.1	0.270915959	0.391259487	0.651247818	0.658851867	1.972275131
			4.2	0.406394938	0.37912184	0.495809203	0.658851867	1.940177848
			4.3	0.279056912	0.341243482	0.414871881	0.913871856	1.949044131
			4.4	0.270915959	0.540092942	0.461686898	0.658851867	1.931547666
			4.5	0.364905821	0.418792524	0.561128655	0.703227549	2.048054549
		3.4	**4.1**	0.325457616	0.501146453	0.963126208	0.684964891	2.474695168
			4.2	0.460936596	0.489008806	0.940873665	0.818150964	2.708970031
			4.3	0.333598569	0.451130448	0.708778199	0.922012809	2.415520024
			4.4	0.325457616	0.649979909	0.788134323	0.699533927	2.463105776
			4.5	0.419447478	0.52867949	1.031172175	0.887505704	2.866804847
	2.3	3.1	**4.1**	0.141019413	0.824960442	0.593919485	0.487980404	2.047879743
			4.2	0.276498392	0.812822795	0.638552836	0.688052369	2.415926391
			4.3	0.149160365	0.709439963	0.593919485	0.913871856	2.366391669
			4.4	0.141019413	0.973793897	0.811458402	0.89508024	2.821351952
			4.5	0.235009274	0.852493479	0.661965452	0.690521216	2.439989421
		3.2	**4.1**	0.141019413	0.622650055	0.540442554	0.677541323	1.981653344
			4.2	0.276498392	0.610512408	0.503620976	0.79615836	2.186790136

Continued on next page

				cost1	cost2	cost3	cost4	cost total
			4.3	0.149160365	0.507129576	0.350881634	0.913871856	1.921043431
			4.4	0.141019413	0.77148351	0.350881634	0.677541323	1.94092588
			4.5	0.235009274	0.650183092	0.593919485	0.8655131	2.344624951
		3.3	**4.1**	0.227497168	0.584771697	0.702478382	0.72435634	2.239103587
			4.2	0.362976147	0.57263405	0.547039767	0.72435634	2.207006304
			4.3	0.23563812	0.469251219	0.466102445	0.913871856	2.08486364
			4.4	0.227497168	0.733605153	0.512917462	0.72435634	2.198376123
			4.5	0.321487029	0.612304734	0.612359219	0.768732023	2.314883005
		3.4	**4.1**	0.282038825	0.684203238	1.003901346	0.750469365	2.720612774
			4.2	0.417517804	0.672065591	0.981648804	0.883655438	2.954887636
			4.3	0.290179778	0.568682759	0.749553337	0.922012809	2.530428683
			4.4	0.282038825	0.833036693	0.828909462	0.765038401	2.709023381
			4.5	0.376028686	0.711736275	1.071947313	0.953010177	3.112722452
	2.1	3.1	4.1	0.705926822	0.37912184	0.296959743	0.535862417	1.917870821
			4.2	0.708219729	0.517498734	0.341593093	0.753262851	2.320574407
			4.3	0.542524331	0.37912184	0.296959743	0.905730904	2.124336817
			4.4	0.691357787	0.527955295	0.514498659	0.928393218	2.662204959
			4.5	0.597375871	0.406654877	0.36500571	0.535862417	1.904898874
		3.2	4.1	0.773972789	0.419849304	0.554566629	0.725423336	2.473812059
			4.2	0.776265696	0.558226199	0.517745052	0.861368842	2.713605788
			4.3	0.610570298	0.419849304	0.36500571	0.905730904	2.301156215
			4.4	0.759403754	0.568682759	0.36500571	0.710854301	2.403946523
			4.5	0.665421838	0.447382341	0.608043561	0.710854301	2.43170204
		3.3	4.1	0.842010811	0.37912184	0.695313617	0.772238353	2.688684621
			4.2	0.844303717	0.517498734	0.539875003	0.789566822	2.691244276
			4.3	0.678608319	0.37912184	0.458937681	0.905730904	2.422398743
			4.4	0.827441775	0.527955295	0.505752698	0.757669318	2.618819085
			4.5	0.733459859	0.406654877	0.605194454	0.614073223	2.359382414
		3.4	4.1	0.75935835	0.37912184	0.760110922	0.798351378	2.696942489
			4.2	0.761651256	0.517498734	0.73785838	0.94886592	2.96587429
			4.3	0.595955858	0.37912184	0.505762914	0.913871856	2.394712467
			4.4	0.744789314	0.527955295	0.585119038	0.798351378	2.656215025
			4.5	0.650807398	0.406654877	0.828156889	0.798351378	2.683970542

Continued on next page

				cost1	cost2	cost3	cost4	cost total
			4.1	0.516365902	0.541149722	0.495809203	0.585878422	2.139203249
			4.2	0.518658809	0.529012075	0.540442554	0.652764314	2.240877752
		3.1	4.3	0.352963411	0.491133717	0.495809203	0.905730904	2.245637235
			4.4	0.501796867	0.689983178	0.71334812	0.978409223	2.883537387
			4.5	0.407814951	0.568682759	0.56385517	0.585878422	2.126231302
			4.1	0.58441187	0.392316267	0.56385517	0.775439341	2.316022648
			4.2	0.586704776	0.38017862	0.527033592	0.760870306	2.254787293
		3.2	4.3	0.421009378	0.342300262	0.37429425	0.905730904	2.043334794
			4.4	0.569842834	0.541149722	0.37429425	0.760870306	2.246157112
			4.5	0.475860918	0.419849304	0.617332101	0.760870306	2.273912629
	2.2		4.1	0.652449891	0.347840695	0.700854051	0.822254358	2.523398995
			4.2	0.654742797	0.335703048	0.545415436	0.689068286	2.224929567
		3.3	4.3	0.4890474	0.29782469	0.464478114	0.905730904	2.157081108
			4.4	0.637880855	0.496674151	0.511293131	0.807685322	2.453533459
			4.5	0.54389894	0.375373732	0.610734888	0.664089228	2.194096788
			4.1	0.56979743	0.457727662	0.875538322	0.848367383	2.751430796
			4.2	0.572090336	0.445590015	0.85328578	0.848367383	2.719333513
		3.4	4.3	0.406394938	0.407711657	0.621190314	0.913871856	2.349168765
			4.4	0.555228394	0.606561117	0.700546438	0.848367383	2.710703332
			4.5	0.461246478	0.485260699	0.943584289	0.848367383	2.738458849
			4.1	0.516365902	0.824960442	0.593919485	0.651382895	2.586628724
			4.2	0.518658809	0.812822795	0.638552836	0.718268788	2.688303227
		3.1	4.3	0.352963411	0.709439963	0.593919485	0.905730904	2.562053763
			4.4	0.501796867	0.973793897	0.811458402	1.043913696	3.330962862
			4.5	0.407814951	0.852493479	0.661965452	0.651382895	2.573656777
			4.1	0.58441187	0.622650055	0.608488521	0.840943815	2.65649426
			4.2	0.586704776	0.610512408	0.571666943	0.826374779	2.595258905
		3.2	4.3	0.421009378	0.507129576	0.418927601	0.905730904	2.252797459
			4.4	0.569842834	0.77148351	0.418927601	0.826374779	2.586628724
			4.5	0.475860918	0.650183092	0.661965452	0.826374779	2.614384241
	2.3		4.1	0.652449891	0.584771697	0.752084615	0.887758832	2.877065035
			4.2	0.654742797	0.57263405	0.596646	0.754572759	2.578595606
		3.3	4.3	0.4890474	0.469251219	0.515708678	0.905730904	2.3797382

Continued on next page

				cost1	cost2	cost3	cost4	cost total
			4.4	0.637880855	0.733605153	0.562523695	0.873189796	2.807199499
			4.5	0.54389894	0.612304734	0.661965452	0.729593702	2.547762827
			4.1	0.56979743	0.684203238	0.916313461	0.913871856	3.084185984
			4.2	0.572090336	0.672065591	0.894060919	0.913871856	3.052088702
		3.4	**4.3**	0.406394938	0.568682759	0.661965452	0.913871856	2.550915006
			4.4	0.555228394	0.833036693	0.741321577	0.913871856	3.04345852
			4.5	0.461246478	0.711736275	0.984359428	0.913871856	3.071214037

TABLE C.1: Costs data of the game yielding the *SPoA* of 2.550915006

Acknowledgements

In this section of my master thesis I want to express my gratitude to all those who helped me complete this work, finalize my studies and stood by me in all the bad and good times.

I want to thank my parents for their true, sincere and selfless support they provided me with through all the years I have been a student. I owe you more than words can say and I would have never made it without your help.

Professor Uetz and Jasper. Thank you for your help and guidance through my project. I learned from you more than I could imagine.

Lillian Spijker, Renate van Luijk and Linda Stegeman thank you for your support and advice during my master program.

Dimitris, Dan, Aykan, Laura, Antonia, Kostakos, Razvan, Moustafa, Gabi, Christos, Vassilis. I'm glad I met you.

Bibliography

[1] J Von Neumann and O Morgenstern. Theory of Games and Economic Behavior. *Princeton University Press*, 1944. doi: 10.1177/1468795X06065810.

[2] Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35(1-2):166–196, 2001. doi: 10.1006/game.1999.0790.

[3] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009. doi: 10.1016/j.cosrev.2009.04.003.

[4] T. Roughgarden and E. Tardos. How bad is selfish routing? *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 49(2):1–26, 2000. doi: 10.1109/SFCS.2000.892069.

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

[6] Association for Computing Machinery. Three papers cited for laying foundation of growth in algorithmic game theory. 2012. URL http://www.acm.org/press-room/news-releases/2012/goedel-prize-2012/.

[7] J. Maynard Smith. The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology*, 47(1):209–221, 1974. doi: 10.1016/0022-5193(74)90110-6.

[8] Nolan McCarty and Adam Meirowitz. *Political Game Theory: An Introduction*. Princeton University, 2007.

[9] Peter Hammerstein and Reinhard Selten. Chapter 28 Game theory and evolutionary biology, 1994.

[10] Tatiana Kornienko. Nature's Measuring Tape: A Cognitive Basis for Adaptive Utility. 2013, University of Edinburgh.

[11] Brian Skinner. The price of anarchy in basketball. *Journal of Quantitative Analysis in Sports*, 6(1):9, 2009. doi: 10.2202/1559-0410.1217.

[12] Robert J. Aumann and Michael Maschler. Game theoretic analysis of a bankruptcy problem from the Talmud. *Journal of Economic Theory*, 36(2):195–213, 1985. doi: 10.1016/0022-0531(85)90102-4.

[13] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973. doi: 10.1007/BF01737559.

[14] Lionel Giles. Sun Tzu on the Art of War. 1910.

[15] Harold W. Kuhn. Extensive games and the problem of information. *Contribution to the Theory of Games, Vol. II, Annals of Math. Studies, 28, 193-216, 1953.*

[16] R. Selten. A simple model of imperfect competition, where 4 are few and 6 are many. *International Journal of Game Theory*, 2(1):141–201, 1973. doi: 10.1007/BF01737566.

[17] Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. The curse of simultaneity. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 60–67, New York, NY, USA, 2012. ACM. doi: 10.1145/2090236.2090242.

[18] Frank P. Kelly, Aman K. Maulloo, and David KH H Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998. doi: 10.1057/palgrave.jors.2600523.

[19] Martin Hoefer and Alexander Skopalik. Altruism in atomic congestion games. In *Lecture Notes in Computer Science*, volume 5757 LNCS, pages 179–189, 2009. doi: 10.1007/978-3-642-04128-0_16.

[20] G. Monaco V. Bilo, M. Flammini and L. Moscardelli. Some anomalies of farsighted strategic behavior. In *Proceedings 10th WAOA, Lecture Notes in Computer Science, Vol. 7846, 229-241, 2013.*

[21] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 67–73, New York, NY, USA, 2005. ACM. doi: 10.1145/1060590.1060600. URL http://doi.acm.org/10.1145/1060590.1060600.

[22] Y. Azar B. Awerbuch and A. Epstein. The price of routing unsplittable flow. In *Proceedings 37th STOC*, pages 56–77, 2005.

[23] J. de Jong and M. Uetz. The sequential price of anarchy for atomic congestion games. In *Proceedings of the 10th WINE*, pages 429–434, 2014.

[24] Zhu Yuzixuan. Improvements of lower bounds on the sequential price of anarchy of certain classes of games. Internship Report, University of Twente, 2014.

[25] J. Correa, J. de Jong, B. de Keijzer, and M. J. Uetz. The curse of sequentiality in routing games. In E. Markakis and G. Schäfer, editors, *11th International Conference on Web and Internet Economics, WINE 2015, Amsterdam, The Netherlands*, volume 9470 of *Lecture Notes in Computer Science*, pages 258–271, Heidelberg, December 2015. Springer Verlag.

[26] Vittorio Bilò. A unifying tool for bounding the quality of non-cooperative solutions in weighted congestion games. *CoRR*, abs/1110.5439, 2011. URL http://arxiv.org/abs/1110.5439.