

Achieving Good Nash Equilibrium by Temporal Addition of Dummy Players

Ofek Dadush and Tami Tamir
 School of Computer Science
 The Interdisciplinary Center
 Herzliya, Israel

Emails: ofek.dadush@post.idc.ac.il, tami@idc.ac.il

Abstract—We consider cost-sharing games in which resources’ costs are fairly shared by their users. The total players’ cost in a Nash Equilibrium profile may be significantly higher than the social optimum. We compare and analyze several methods to lead the players to a good Nash Equilibrium by temporal addition of dummy players. The dummy players create artificial load on some resources, that encourage other players to change their strategies.

We show that it is NP-hard to calculate an optimal strategy for the dummy players. We then focus on symmetric singleton games for which we suggest several heuristics for the problem. We analyze their performance distinguishing between several classes of instances and several performance measures.

I. INTRODUCTION

IN resource allocation applications, a centralized authority is assigning the clients to different resources. For example, in job-scheduling applications, jobs are assigned to servers to be processed; in communication or transportation networks, traffic is assigned to network links to be routed. The centralized utility is aware of all clients’ requests and determine the assignment. Classical computational optimization problems study how to utilize the system in the best possible way. In practice, many resource-allocation services lack a central authority, and are often managed by multiple strategic users, whose individual payoff is affected by the assignment of other users. As a result, game theory has become an essential tool in the analysis of resource-allocation services. In the corresponding game, every client corresponds to a selfish player who aims at maximizing its own utilization. Naturally, suboptimal players will keep changing their strategy, and the dynamic continues as long as the profile is not stable. Pure Nash equilibrium (NE) is the most popular solution concept in games. A strategy profile is a NE if no player has a beneficial deviation.

It is well known that decentralized decision-making may lead to sub-optimal solutions from the point of view of the society as a whole. On the other hand, the system cannot control the decisions made by the players. In this work we propose to analyze *the power of adding dummy players* controlled by the system. The goal of the dummy players is to direct the players to a high quality solution, while still keeping their freedom to act selfishly and select their own strategy.

The addition of dummy players is temporal, that is, the final configuration consists of the initial set of players. Since

the final configuration must be stable, the goal is to lead the players to a good Nash Equilibrium.

Many real life applications can benefit from adapting this approach. For example, navigation apps users receive information about the current status of the traffic and act accordingly, the provider can adjust the information presented to the users in favor of improving the balancing done on cars and roads (and by that avoid creation of traffic jams). Similarly, in communication networks, the delay of using a link can be artificially increased in order to encourage users to use alternative links.

A. Notation and Problem Statement

For an integer $n \in \mathbb{N}$, let $[n] = \{1, \dots, n\}$. A *network-formation game* (NFG, for short) [1] is $\mathcal{N} = \langle N, G, \langle s_j, t_j \rangle_{j \in [n]} \rangle$, where N is a set of n players, $G = \langle V, E, c \rangle$ is a weighted graph, and for each $j \in [n]$, the pair $\langle s_j, t_j \rangle$ describes the objective of Player j , namely forming a path from its source vertex $s_j \in V$ to its target vertex $t_j \in V$.

A pure *strategy* of a player $j \in N$ is a path from s_j to t_j . A *profile* in \mathcal{N} is a tuple $p = \langle p_1, \dots, p_n \rangle$ of strategies for the players, that is, p_j is a path from s_j to t_j . Consider a profile p . Recall that c maps each edge to a cost, intuitively standing for the cost of its formation. The cost of an edge is shared equally by the players that uses it. The players aim at fulfilling their objective with minimal cost.

For a profile p , let $n_e(p)$ denote the load on edge e in p , that is, the number of players that include e in their path. The cost of player j in profile p is defined to be

$$\text{cost}_j(p) = \sum_{e \in p_j} c_e / n_e(p).$$

The cost of a profile p is the total players’ cost, that is $\text{cost}(p) = \sum_{j \in N} \text{cost}_j(p)$.

For a profile p and a strategy p_j of player $j \in [n]$, let $[p_{-j}, p'_j]$ denote the profile obtained from p by replacing the strategy for Player j by p'_j . Given a strategy profile p , the *best response* (BR) of player j is $BR_j(p) = \arg \min_{p'_j \in P_j} \text{cost}_j(p'_j, p_{-j})$; i.e., the set of strategies that minimize player j ’s cost, fixing the strategies of all other players. Player j is said to be *suboptimal* in p if it can reduce its cost by a unilateral deviation, i.e., if $p_j \notin BR_j(p)$. If no player is suboptimal in p , then p is a *Nash equilibrium* (NE).

Given an initial strategy profile p^0 , a BR-sequence from p^0 is a sequence $\langle p^0, p^1, \dots \rangle$ in which for every $T = 0, 1, \dots$ there exists a player $j \in N$ such that $p^{T+1} = (p_j^T, p_{-j}^T)$, where $p_j^T \in BR_j(p_{-j}^T)$. We restrict attention to games in which such best-response dynamics (BRD) are guaranteed to converge to a NE.

A game \mathcal{N} with k dummy players is an extension of \mathcal{N} into $\mathcal{N}' = \langle N, G, \langle s_j, t_j \rangle_{j \in [n]}, k \rangle$. The dummies have no reachability objective of their own, and are controlled by the system. Every dummy player is assigned on a single edge and increases the load on it, thus, making it more attractive for the other players. Practically, a profile of the game \mathcal{N}' is given by the strategies of N and the location of the k dummies. The dummy players are added to a given initial profile p^0 . Due to their addition, some of the players will become suboptimal, and a BR-sequence will be initiated. The system can control which suboptimal player is selected to perform its BR. After a finite number of BR-steps, the dummy players leave the network, and the players may continue the BR-sequence until convergence to a NE.

It is well known that NE profiles may be sub-optimal. Let $OPT(G)$ denote the social optimum of a game \mathcal{N} , that is, the minimal possible social cost of a feasible assignment of N , i.e., $OPT(\mathcal{N}) = \min_p cost(p)$. The inefficiency incurred due to self-interested behavior is quantified according to the *price of anarchy* (PoA) [11], [15] and *price of stability* (PoS) [1] measures. The PoA is the worst-case inefficiency of a pure Nash equilibrium, while the PoS measures the best-case inefficiency of a pure Nash equilibrium. Formally, $PoA(\mathcal{N}) = \max_{p \in NE(\mathcal{N})} cost(p)/OPT(\mathcal{N})$, and $PoS(\mathcal{N}) = \min_{p \in NE(\mathcal{N})} cost(p)/OPT(\mathcal{N})$.

The goal of the dummy addition is to initiate a BR-sequence in which the players converge to a NE whose cost is as close as possible to the cost of the best NE.

Some of our results refer to symmetric singleton games. These games fit several practical environments such as scheduling on parallel machines, or routing on parallel links [11]. A network formation game that corresponds to a symmetric singleton game is given by m parallel $(s-t)$ -links (e_1, \dots, e_m) and a vector of positive link costs (c_1, \dots, c_m) , where c_i is the activation cost of link i . All the players have the same objective – a path from s to t , and thus, the symmetric strategy space is simply the set of edges. A profile p of the game is given by a vector of loads (n_1^p, \dots, n_m^p) , where n_i^p is the number of players on e_i in profile p . Let $n = \sum_i n_i^p$. We assume, w.l.o.g., that $c_1 \leq c_2 \leq \dots \leq c_m$. Clearly, the social optimum profile of such a game is simply assigning all the players on the cheapest link e_1 . On the other hand, it is well known that the price of anarchy is n even for a simple network with only two parallel links having costs $c_1 = 1$ and $c_2 = n$. Indeed, if all the players are assigned on e_2 , then each of them pays $n/n = 1$ and would not benefit from deviating to e_1 . Note that for this network, a single dummy assigned on e_1 is sufficient to encourage the players to deviate to e_1 .

B. Related Work

Many modern systems provide service to multiple strategic users, whose individual payoff is affected by the decisions made by other users of the system. As a result, non-cooperative game theory has become an essential tool in the analysis of this kind of systems, in particular, routing in networks and job scheduling systems [11], [19], [3], [8], [2], [1].

The addition of dummy players will make some of the players suboptimal, and will cause them to change their strategy. Other player will act in response. Thus, our work is closely related to the study of best-response dynamics. The analysis of BR dynamics consists of three main directions: The first studies whether BR dynamics converge to a NE, if one exists (e.g., [13], [8] and references therein). It is well known that BR dynamics does not always converge to a NE, even if one exists. However, for the class of finite *potential games* [16], [14], a pure NE always exists, and BR dynamics is guaranteed to converge to one. The second direction explores how fast it takes until BR dynamics converges to a NE, e.g., [1], [4], [9]. For some games, such as network formation games, the convergence time may be exponential, while for some games, such as singleton congestion games, fast convergence is guaranteed. The third direction studies how the quality of the resulting NE is affected by the choice of the deviating player. Specifically, the order in which players are chosen to perform their best response moves is crucial to the quality of the equilibrium reached [5].

Other related work deal with games in which some of the players are not selfish. In Stackelberg games [17], [10], [6], [7], a centralized authority selects a fraction of players, denoted *leaders*, and assigns them to appropriately selected strategies, this is called the *Stackelberg strategy*. Each of the remaining players, denoted *followers*, selects its strategy selfishly trying to minimize its cost. The behavior of selfish players leads to a *Stackelberg Nash equilibrium* in which none of the selfish players has a beneficial migration.

The goal is to design Stackelberg strategies that will lead the players to a high quality NE. In [17], it is shown that finding an optimal Stackelberg strategy in job scheduling games is NP-hard, and approximation algorithms are presented. In congestion games on parallel links network the usage of a centrally controlled player can lead to the network optimum if its weight is above certain threshold [10]. In parallel networks, under some constraints, there are even optimal Stackelberg strategies [12].

Our model differs from Stackelberg games as we do not assume that some players obey the system. That is, all the players act selfishly. The added dummy players are temporal, and the system should reach a NE after they vanish. The idea of adding a temporal dummy player in order to change the final equilibrium was first presented in [18]. The paper analyzes the potential *damage* a single dummy player can cause to the social optimum in job scheduling games with weighted players.

C. Our Results

Let p^* be the cheapest NE profile. By assigning a sufficiently large number of dummies on the paths in p^* , these paths would become attractive enough, so that the BR of every player would be to join its path in p^* . Since p^* is a NE, the players will remain on these paths after the dummy players depart. Thus, if the number of dummy players is not limited, then it is possible to guarantee convergence to the best NE. We consider two problems:

- 1) What is the minimal number of dummy players required to reach the best NE?
- 2) Given a budget of k dummy players, what is the minimal cost NE that can be reached?

A solution for each of these problems involves also an algorithm for utilizing the dummy players. Specifically, for every profile on the BR-sequence, the algorithm should decide (i) on which links the dummy players are assigned, and (ii) which suboptimal player is activated next to perform its best-response.

In section II we prove NP-hardness of both problems for general networks. Specifically, we present a game with two NE profiles, p^* and p such that $\text{cost}(p)/\text{cost}(p^*) = \Theta(n)$, it is NP-hard to utilize two dummy players in a way that leads the players to p^* , while it is straightforward to do it with three dummy players.

In Section III we define formally the game on m parallel links and provide several basic observations and properties of BR-sequences. In Section IV we present our heuristics for convergence into the social optimum. In section V we presents our heuristics for a given number of dummies, and in Section VI we presents our experimental results.

The addition of dummy players is one temporal perturbation of a game. We conclude in Section VII where we introduce additional perturbation and suggest some directions for future work. Due to space constraints, some of the proofs and experimental results are omitted from this manuscript.

II. HARDNESS PROOF FOR GENERAL NETWORKS

Let p^* be a min-cost NE profile. By assigning a large enough number of dummy players on the edges of p^* , it is clearly possible to attract the players to p^* . We show that calculating the minimal number of dummies required for this task is NP-hard. Our hardness proof is based on the hardness proof in [5] that considers a problem of determining the order according to which players perform BRD (Best response dynamics).

Theorem 2.1: The problem of leading the players to the lowest cost NE using the minimal number of dummies is NP-hard.

Proof: Given a game, an initial NE strategy profile, and a value k , the associated decision problem is whether k dummies are sufficient to lead the players to the lowest cost NE. We show a reduction from the *Partition* problem: Given a set of numbers $\{a_1, a_2, \dots, a_n\}$ such that $\sum_{i \in [n]} a_i = 2$, where $\forall_{i \in [n]} a_i < 1$, the goal is to find a subset $I \subseteq [n]$ such that

$\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i = 1$. Given an instance of *Partition*, consider the network depicted in Figure 1, with the following initial strategy profile, p^0 of $4n + 2$ players:

- $3n$ partition players, i_1, i_2, i_3 for all $i \in [n]$. The objective of every triplet i_ℓ , is a $\langle v_{i-1}, v_i \rangle$ -path. For all $i \in [n]$, the three corresponding partition players has two strategies: an upper edge of cost $420a_i$ and a lower edge of cost $300a_i$. In p^0 , all the partition players use the upper edges.
- Players $1', 2', \dots, n'$: n players whose objective is an $\langle s', t' \rangle$ -path. These players have two strategies: the edge $\langle s', t' \rangle$ of cost $300n$, and the path through (u_1, u_2) , whose cost is $1200 - \epsilon$. In p^0 , they all use the edge $\langle s', t' \rangle$.
- Player a whose objective is an $\langle s_a, t_{a,b} \rangle$ -path. Player a has two strategies: The upper path, and the path through v_0, v_1, \dots, v_n . In p^0 , Player a uses the upper path.
- Player b whose objective is an $\langle s_b, t_{a,b} \rangle$ -path. Player b has three strategies: The upper path, the path through $s_a, v_0, v_1, \dots, v_n$, and the path through the edge (u_1, u_2) . In p^0 , Player b uses the upper path.

Observe that p^0 is a NE. Specifically, each of the partition player has cost $\frac{420a_i}{3}$ and a deviation to the lower edge will lead to cost $300a_i$, Player a 's current cost is $248 + \frac{\epsilon}{2}$. Deviating to the path through v_0 , would result in cost $\frac{420}{4} \cdot 2 + 204 = 414$. Player b 's current cost is $248 + \frac{\epsilon}{2}$. Its alternative would cost $154 + 414 = 568$ (through v_0) or $1200 - \epsilon$ (through u_1). Finally, every player on the lower edge has current cost 300, while its alternative through u_1 costs $1200 - \epsilon$.

The following additional observations limit the possible BRD sequences of the game:

- 1) Not only that the initial profile is a NE, but it is also stable in the presence of a single dummy.
- 2) In order to initiate a deviation of a partition player i_ℓ , two dummies should be placed on the lower (v_{i-1}, v_i) -edge, as $100a_i = \frac{300a_i}{3} \leq \frac{420a_i}{3} = 140a_i$.
- 3) The n players currently on $\langle s', t' \rangle$ would benefit from a deviation only after the edge (u_1, u_2) is utilized by three other player.

The following profile p^* is the minimal cost NE of this game and also its social optimum:

- For every $i \in [n], \ell \in [3]$, the partition player i_ℓ uses the lower (v_{i-1}, v_i) edge. of cost $300a_i$.
- Players $1', 2', \dots, n'$ are on the path through u_1 . use the $1200 - \epsilon$ edge.
- Player a is on the path through v_0 and use the lower edges uses the lower (v_{i-1}, v_i) edges.
- Player b is on the path through u_1 .

The social optimum cost is $300 \cdot 2 + 204 + 1200 - \epsilon = 2004 - \epsilon$.

The main claim of the reduction is based on the properties presented earlier.

Claim 2.2: Two dummies can guarantee convergence to p^* if and only if a partition of exists. ■

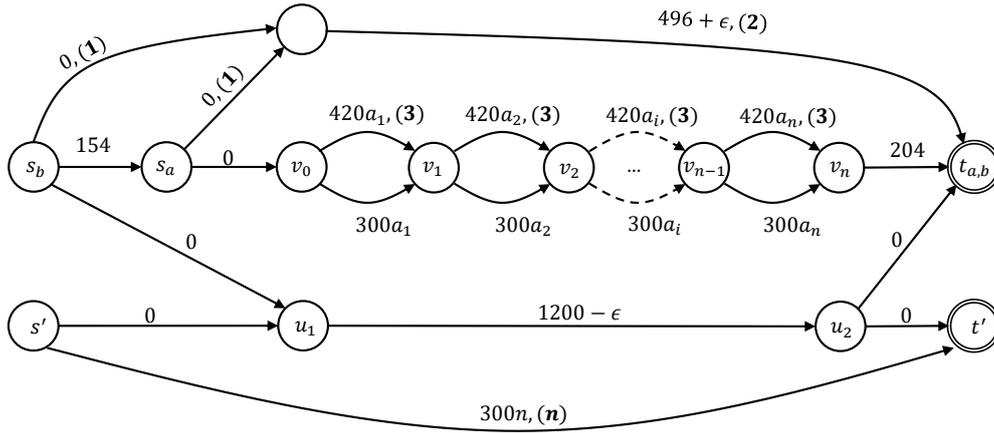


Fig. 1. The network constructed for a given *Partition* instance. Every edge is labeled by its cost, and (in brackets) the number of players using it in p^0 .

III. COST-SHARING GAMES ON PARALLEL LINKS

In light of the hardness result for general networks, we consider a network of parallel links. Recall that the network is given by m parallel links (e_1, \dots, e_m) and a vector of positive link costs (c_1, \dots, c_m) . A profile p of the game is given by a vector of loads (n_1^p, \dots, n_m^p) , where n_i^p is the number of players on e_i in profile p . With fair cost-sharing, the cost of a player assigned on e_i in profile p is c_i/n_i^p . We assume, w.l.o.g., that $c_1 \leq c_2 \leq \dots \leq c_m$. Denote by p_{a+} the profile obtained from p by adding k dummy players on e_a . Given a profile p , the best response of a player on e_i is denoted $BR_i(p)$. A link $e_j \in BR_i(p)$ if and only if $\frac{c_j}{n_j+1} < \frac{c_i}{n_i}$ and $\forall e_l \neq e_i, \frac{c_j}{n_j+1} \leq \frac{c_l}{n_l+1}$. In particular, $e_a \in BR_i(p_{a+})$ if it is possible to attract a player from e_i to migrate to e_a by adding the dummy players on e_a .

Let n_i^0 be the load on e_i in the initial profile p^0 . Let e_1 be the cheapest link, were ties are broken in favor of highly loaded links in p^0 . That is, for every $i > 1$, either $c_1 < c_i$, or $c_1 = c_i$ and $n_1^0 \geq n_i^0$. Since the game is symmetric, the social optimum cost is c_1 .

We present heuristics for solving the following problems: Given a network of parallel links and an initial configuration p^0 , (i) what is the minimal number of dummies required to reach the social optimum, and (ii) what is the social lowest cost we can achieve with a given number of dummies. For both problems we assume that the algorithm can move the dummy players among the links, and can select the deviating suboptimal player in each step. Players that get the right to deviate select their best response move.

Performance Measures: Assume that some heuristic is performed on an initial profile p^0 . The quality of a solution will be measured by 4 parameters.

- 1) The Social cost of the final profile.
- 2) Number of dummies used.
- 3) Length of BR-sequence till convergence.
- 4) Number of times the dummy players move.

In our experiments some of these measures are fixed. For example, we tested the social cost achieved by various heuristics with a given number of dummies, or the numbers of dummies required to converge to the social optimum, e_1 .

A. Preliminaries and Observations

We start by introducing some notation and stating few important observations and claims.

For a profile p , let $E_{min}^p = \arg \min_{i \in E} \frac{c_i}{n_i^p+1}$, be the set of all most attractive links. Let e_{min}^p be a link in E_{min}^p with a highest cost, breaking ties arbitrarily. Let $price_{min}^p = \frac{c_{e_{min}^p}}{n_{e_{min}^p}^p+1}$ be the cost to be paid by a player that joins the most attractive link.

Observation 3.1: In every NE profile, all the players are assigned on the same link.

Claim 3.2: If $e_a \in BR_{e_b}(p)$ for some link e_b , then we can guarantee convergence of BRD to e_a .

Proof: We show that e_a is the BR as long as the dummy players do not change their location. $e_a \in BR_{e_b}(p)$ if and only if $\frac{c_a}{n_a^p+1} < \frac{c_b}{n_b^p}$ and $\frac{c_a}{n_a^p+1} \leq \frac{c_i}{n_i^p+1}$, for every $i \neq b$. After one player moves from e_b to e_a , the load on e_a is $n_a^p + 1$. Now, $e_a \in BR_{e_b}(p^{+1})$ for every $i \neq a$, since

$$\frac{c_a}{n_a^{p+1}+1} = \frac{c_a}{n_a^p+2} < \frac{c_a}{n_a^p+1} \leq \frac{c_i}{n_i^p+1} = \frac{c_i}{n_i^{p+1}+1}$$

$$\frac{c_a}{n_a^{p+1}+1} = \frac{c_a}{n_a^p+2} < \frac{c_a}{n_a^p+1} < \frac{c_b}{n_b^p} < \frac{c_b}{n_b^p-1} = \frac{c_b}{n_b^{p+1}}$$

Thus, independent of the order the players are activated, as long as dummy players are not changing their location, every BR sequence converges to e_a . ■

Observation 3.3: For any $a, b, x, y, k > 0$, if $a \leq x$ and $\frac{a}{b} \leq \frac{x}{y}$ then $\frac{a}{b+k} \leq \frac{x}{y+k}$.

Observation 3.3 implies that if a link is less attractive than e_1 , then it will never get players during a sequence that converges to e_1 , since it requires at least the same number of dummies as making e_1 the BR of some link directly.

Our next claim states that if a link, e_b , is a best-response of some player, then it is also a best-response of the players on e_{min}^p . Note that if the link e_{min}^p is empty then it must be that $e_{min}^p = e_1$ and convergence to e_1 is possible even without dummy players.

Claim 3.4: For a given profile p and a link $e_b \neq e_{min}^p$, if $\exists e_a$ s.t $e_b \in BR_a(p)$ then $e_b \in BR_{e_{min}^p}(p)$.

In addition, if e_a is the BR of some link in E_{min}^p it is the BR of every link in E_{min}^p .

Claim 3.5: If $e_a \in BR_{e_{min}^p}(p)$ then for every $e_i \in E_{min}^p$, it holds that $e_a \in BR_{e_i}(p)$.

Proof: If $|E_{min}^p| = 1$ then it is clearly true. Else $|E_{min}^p| > 1$ and let $e_u, e_v \in E_{min}^p$, assume $e_a \in BR_{e_u}(p)$. It must be that $\frac{c_a}{n_a+1} < \frac{c_u}{n_u}$ and $\frac{c_a}{n_a+1} \leq \frac{c_v}{n_v+1} < \frac{c_v}{n_v}$. Since $\frac{c_u}{n_u+1} = \frac{c_v}{n_v+1}$, we conclude $e_a \in BR_{e_v}(p)$ for any other link in E_{min}^p . ■

B. The Naive Solution

Before presenting the more complicated heuristics, we present a naive solution that is based on directly making e_1 the best-response of some player. By Claim 3.2, once e_1 is the BR of some link, we can guarantee convergences to e_1 , and based on Claim 3.4, we can calculate the number of dummies required to directly make e_1 the BR of e_{min}^p .

Recall that $e_1 \in BR_{e_{min}^p}(p_{1+})$ if and only if $\frac{c_1}{n_1+k+1} < \frac{c_{e_{min}^p}}{n_{e_{min}^p}^p}$, and $\frac{c_1}{n_1+k+1} \leq \frac{c_i}{n_i+1}$, for every $i \neq e_{min}^p$. Therefore, the minimal integer k satisfying

$$\frac{c_1}{n_1+k+1} < \frac{c_{e_{min}^p}}{n_{e_{min}^p}^p} \text{ and } \frac{c_1}{n_1+k+1} \leq \min_{i \neq e_{min}^p} \frac{c_i}{n_i+1}$$

is the minimal number of dummies required to directly make e_1 a BR of some player. We get that k is the minimal integer satisfying

$$k > c_1 \cdot \frac{n_{e_{min}^p}^p}{c_{e_{min}^p}} - (n_1+1) \text{ and } k \geq c_1 \cdot \max_{i \neq e_{min}^p} \left(\frac{n_i+1}{c_i} \right) - (n_1+1) \quad (1)$$

Denote by $k_{naive}(p)$ the minimal k satisfying 1. Specifically, $k_{naive}(p^0)$ is the number of dummies required by the naive solution.

Table I presents an instance demonstrating that the naive solution is suboptimal. Moreover, the number of dummies it needs is higher by factor of about 1.5 from the optimum. The network consists of 8 links whose costs are listed in the first row. The loads in the initial profile p^0 are listed in the second row. The additional rows specify for each link the cost per player in p^0 – given by $\frac{c_e}{n_e^0}$, and cost per player if one player joins e , given by $\frac{c_e}{n_e^0+1}$.

Links $e_2 - e_7$ all have the same cost and initial load.

We can easily see that without dummies, regardless of the activation order, the players will converge into e_8 , which is the most expensive link. Using the naive solution, the required number of dummies needed to converge into e_1 is $k_{naive}(p^0) = 300$. We show that convergence to e_1 can be achieved using $k = 220$ dummies.

Link	e_1	$e_2 - e_7$	e_8
Cost	3000	3100	6000
Load	0	200	600
$\frac{c_e}{n_e^0}$	-	15.5	10
$\frac{c_e}{n_e^0+1}$	3000	15.42	9.98

TABLE I
INITIAL PROFILE p^0 OF INSTANCE I_1

By assigning 220 dummies on e_2 and activating a player on e_8 , a migration from e_8 to e_2 will be performed, then we assign the dummies on e_3 and activate a player on e_8 which creates a migration from e_8 to e_3 , we then continue in a round robin fashion on links $e_2 - e_7$ until 159 players leave e_8 and the following profile is reached:

Link	e_1	$e_2 - e_4$	$e_5 - e_7$	e_8
Cost	3000	3100	3100	6000
Load	0	227	226	441
$\frac{c_e}{n_e^p}$	-	13.65	13.71	13.60
$\frac{c_e}{n_e^p+1}$	3000	13.59	13.65	13.57

TABLE II
THE PROFILE ACHIEVED AFTER PHASE I

The BR-sequence proceed after the 220 dummies are moved to e_1 and a player on e_8 is activated. Since the cost of a player who would join e_1 is $\frac{3000}{0+220+1} = 13.57$, a player on e_8 will choose e_1 as its BR and by Claim 3.2 convergence to e_1 is guaranteed. We conclude that convergence to e_1 can be achieved with only 220 dummy players, while the naive solution requires 300 dummies.

IV. CONVERGENCE TO THE SOCIAL OPTIMUM

In this section we present our heuristics for convergence into the best NE. In a network of parallel links, the best NE is also the social optimum and is simply e_1 , the cheapest edge in the network.

In Observation 3.3 we showed that making links that are less attractive than e_1 the BR of some link is at least as demanding as making e_1 the BR of the same link, therefore, in all our heuristics we do not use such links as BR of any link. Furthermore, for a given instance I , let I' be an instance with the same set of links and load vector in which $n'_i \geq n_i$ for links fulfilling $e_i \in BR_{e_{min}^p}(p_{i+})$. Intuitively, I' is more challenging than I since the links that are more attractive than e_1 become even more attractive. Such links will also not be used in the heuristics we present.

The heuristics we present consists of two phases. In the first phase, players are encouraged to migrate such that the players' cost on the links that are more attractive than e_1 is more balanced compared to p^0 , and then apply the naive solution on the more balanced profile. The goal is to use fewer than $k_{naive}(p^0)$ dummies for the balancing phase, as well as to reach a profile p where $k_{naive}(p) < k_{naive}(p^0)$.

A. Max Cost-reduction Heuristic

The first heuristic we present balances the players costs on links that are more attractive than e_1 by migrating players out of the most attractive link, e_{min}^p , into a link that will gain a maximal cost-reduction by an addition of one player. Formally, a link for which $\frac{c_i}{n_i} - \frac{c_i}{n_i+1}$ is maximal. Intuitively, we want the migration to be as significant as possible.

The algorithm gets as input a profile, p^0 , and the number, k , of dummies, and returns a binary indicator stating whether the max cost-reduction heuristic can be used to lead the players to e_1 . The minimal number of required dummies, can therefore be computed by binary search in the range $[0, k_{naive}(p^0)]$.

Let e_{mcr}^p be a link for which $\max(\frac{c_i}{n_i} - \frac{c_i}{n_i+1})$ is maximal out of the links that can attract players from e_{min}^p . In every iteration the algorithm moves a player from the current e_{min}^p profile to the current e_{mcr}^p until the profile is balanced enough to enable a migration to e_1 (step 3), or identifying that k dummies are not sufficient as a naive solution from the most balanced priced profile (this is detected in step 11 - by having a loop in the balanced pricing phase).

Algorithm 1 Max Cost-reduction Heuristic (decision version)

- 1: **repeat**
 - 2: Calculate E_{min}^p . Let $e_{min}^p \in E_{min}^p$ be a link with max cost.
 - 3: **if** $e_1 \in E_{min}^p$ or $e_1 \in BR_{e_{min}^p}(p_{1+})$ **then**
 - 4: **return true**
 - 5: **else**
 - 6: Let e_{mcr}^p be a link such that $e_{mcr}^p \in BR_{e_{min}^p}(p_{e_{mcr}^p+})$ for which $(\frac{c_i}{n_i} - \frac{c_i}{n_i+1})$ is maximal.
 - 7: place k dummies on e_{mcr}^p .
 - 8: activate a player from e_{min}^p (creates a migration from e_{min}^p to e_{mcr}^p).
 - 9: remove k dummies from e_{mcr}^p .
 - 10: **end if**
 - 11: **until** loop has been detected (profile $p = p^{-2}$)
 - 12: **return false**
-

When the max cost-reduction heuristic is applied on the instance I_1 presented in Table I and $k = 220$, it is able to reach the social optimum. In fact, the BR-sequence performed is exactly the one described in Table II.

B. balanced pricing Heuristic

The second heuristic we present calculates a target load vector in which the marginal cost on the links are balanced. The dummy players are used to achieve this load vector. The naive algorithm is then performed on the balanced profile. The load vector is a one that maximizes the marginal cost on the most attractive link and the second most attractive link. This way the attractiveness of the competitors of e_1 is as low as possible.

For a profile p , let $c_{min1}^p = \min_{i \in E_{min}^p} \frac{c_i}{n_i}$, and let e_{min1}^p be a link determining c_{min1}^p . Also, let $c_{min2}^p = \min_{i \in E \setminus \{e_{min1}^p\}} \frac{c_i}{n_i+1}$.

The idea is to balance the load on the links such that the minimal among these two values are maximal. Intuitively, this way, by activating a player on e_{min1}^p , the attractiveness of the competitors of e_1 is as low as possible. Calculating the exact load vector achieving maximal $\min\{c_{min1}^p, c_{min2}^p\}$ is computationally hard. In order to simplify the calculations, we calculate instead p_{bal} - a load vector that approximates the optimal one. p_{bal} is defined in the following way: Let $n_{\bar{1}} = \sum_{i>1} n_i^0$ be the number of players that are not assigned on e_1 in p^0 and let $c_{\bar{1}} = \sum_{i>1} c_i$ be the total cost of edges except e_1 . In p_{bal} we determine the assignment of the $n_{\bar{1}}$ players that are not on e_1 . We first determine load $\lfloor \frac{c_i}{c_{\bar{1}}} \cdot n_{\bar{1}} \rfloor$ on every link e_i for $i > 1$, we then add the remaining players iteratively, each time adding a player on a link with maximal $\frac{c_i}{n_i+1}$.

For example, the profile p_{pal} of the instance I_1 introduced in Table I is the following.

Link	e_1	$e_2 - e_5$	$e_6 - e_7$	e_8
Cost	3000	3100	3100	6000
Load	0	227	226	440
$\frac{c_i}{n_i}$	-	13.65	13.71	13.63
$\frac{c_i}{n_i+1}$	3000	13.59	13.65	13.60

TABLE III
PROFILE p_{bal} OF INSTANCE I_1

Once p_{bal} is calculated, we would like to reach this profile from p^0 using the lowest possible number of dummies. Given p and p_{bal} , let $E_{drop}^p = \{e_i | n_i^p > n_i^{p_{bal}}\}$ be the set of edges whose load is higher than their load in p_{bal} and let $E_{gain}^p = \{e_i | n_i^p < n_i^{p_{bal}}\}$ be the set of edges whose load is lower than their load in p_{bal} . Given p , E_{drop}^p and E_{gain}^p , let $k_{migration}$ be the minimal number of dummies required to achieve a migration from a link $e_a \in E_{drop}^p$ to a link $e_b \in E_{gain}^p$, and let the source and target links be e_{drop} and e_{gain} , respectively. The algorithm iteratively calculates e_{drop} and e_{gain} and perform the corresponding migrations. The number of dummies required may increase during the algorithm, and k is updated accordingly. When k is large enough to enable a migration to e_1 , that is, when $k_{migration} \geq k_{naive}(p)$, convergence to e_1 is guaranteed.

C. Exhaustive Heuristic

The third heuristic we consider balances the links that are more attractive than e_1 by migrating players outside of the most attractive link e_{min}^p into the least attractive possible link while making sure $price_{min}$ is not getting lower. Formally for a profile p , let $E_t^p = \{e_a | e_a \in BR_{e_{min}^p}(p_{a+}) \text{ and } \frac{c_a}{n_a+2} > price_{min}^p\}$ be the group of target links, meaning they are BR of e_{min}^p if the dummy player are added on them and do not lower $price_{min}$ if a migration from e_{min}^p to that link occurs. Let e_t^p be the link in E_t^p with maximal $\frac{c_a}{n_a+2}$. Our algorithm is based on moving players out of e_{min}^p into e_t^p .

The algorithm gets as input a profile, p^0 , and the number, k , of dummies, and returns a binary indicator stating whether the exhaustive heuristic leads the players to e_1 . The minimal

Algorithm 2 Balancing Heuristic (min k version)

```

1: Calculate  $p_{bal}$ 
2: set  $k = 0$ 
3: repeat
4:   Calculate  $k_{migration}, e_{gain}, e_{drop}$ .
5:   if  $k_{migration} \geq k_{naive}(p)$  then
6:     return  $\max\{k, k_{naive}(p)\}$ .
7:   end if
8:    $k = \max\{k, k_{migration}\}$ .
9:   place  $k$  dummies on  $e_{gain}$ 
10:  while  $n_i^{pb} - n_i^p > 0$  do
11:    activate a player from some  $e \in E_{drop}^p$  (creates a
      migration into  $e_{gain}$ ).
12:  end while
13: until  $E_{gain}^p = \emptyset$ 
14: return  $\max\{k, k_{naive}(p)\}$ 
    
```

number of required dummies, can therefore be computed by binary search in the range $[0, k_{naive}(p^0)]$.

Algorithm 3 Exhaustive Heuristic (decision version)

```

1: repeat
2:   Calculate  $E_{min}^p$ , let  $e_{min}^p \in E_{min}^p$  be a link with max
      cost and let  $price_{min}^p = \frac{c_{e_{min}^p}}{n_{e_{min}^p} + 1}$ .
3:   if  $e_1 \in E_{min}^p$  or  $e_1 \in BR_{e_{min}^p}(p_{1+})$  then
4:     return true
5:   else if  $E_t^p \neq \emptyset$  then
6:     place  $k$  dummies on  $e_t^p$ .
7:     activate a player from  $e_{min}^p$  (creates a migration from
       $e_{min}^p$  to  $e_t^p$ ).
8:     remove  $k$  dummies from  $e_t^p$ .
9:   end if
10: until no player has being activated
11: return false
    
```

D. Performance Measure Comparison

Recall that in Section III the performance measures according to which the quality of our heuristics is evaluated is listed.

Table IV describe an initial profile of instance I_2 with 8 links. It is easy to see that BRD would converge into e_6 if no dummies are used. Table V summarizes the strengths and weaknesses of the different heuristics when applied on I_2 .

Clearly, the naive solution is most efficient in terms of BR-steps and dummy moves, on the other hand, the number of dummies required to reach e_1 is significantly higher. The max cost-reduction needs more dummies than the other algorithms but dominates the number of steps. The balancing algorithm moves the dummies only once for every link in E_{gain} , so the number of dummy moves is very low. The number of dummies is lower than the naive solution. Finally, the exhaustive heuristic achieves the social optimum using the least number of dummies.

V. EXPLOIT A GIVEN NUMBER OF DUMMY PLAYERS

In this section we present our heuristics for finding the lowest achievable social cost, using a limited budget of k dummies. Notice that if we know that e_1 cannot become the BR of any link using a given K , then, as explained in the previous section, migrating players out of e_1 cannot be helpful.

We modify the algorithms presented in Section IV for the new goal. As we elaborate below, the Naive approach and the balancing heuristic are slightly modified, only their destination link may be more expensive than e_1 . The two other heuristics, specifically, max cost-reduction and exhaustive, have a different version for the new goal.

Recall that in the naive solution (see Section III-B) the algorithm locates the dummies on the target link. When the number of dummies is limited, we simply calculate the minimal ℓ such that

$$\frac{c_\ell}{n_\ell + k + 1} < \frac{c_{e_{min}^p}}{n_{e_{min}^p}^p} \text{ and } \frac{c_\ell}{n_\ell + k + 1} \leq \min_{i \neq e_{min}^p} \frac{c_i}{n_i + 1}.$$

The corresponding link e_ℓ , is the solution of the naive algorithm with a budget of k dummies.

Next, we describe how the balancing heuristic is tuned for the budgeted problem: recall that the idea is to calculate a target load vector and lead the players on $\{e_2, \dots, e_n\}$ to the corresponding configuration. With a given number of dummies, we calculate the minimal ℓ such that it is possible to balance the players on $\{e_{\ell+1}, \dots, e_n\}$, thus leading the players to e_ℓ . That is, the original algorithm is applied only on subset of the links. Recall that once e_ℓ is a BR of some link, then all other players would benefit from joining it, in particular, those on $e_1, \dots, e_{\ell-1}$.

A. Max Cost-reduction Heuristic

Recall that in this heuristic, players are migrated out of e_{min}^p into a link in $BR_{e_{min}^p}$. With a given budget of k dummies we run the same algorithm, and keep track of the lowest-cost link that was a target of some migration during the run.

Therefore, the algorithm differs from the decision version, only in the return statements. In line 4, instead of True it returns e_1 and in line 12, instead of False it returns the minimal i s.t. $e_i \in E_{min}^p$ or $e_i \in BR_{e_{min}^p}(p_{i+})$ in any seen p .

B. Exhaustive Heuristic

In its decision version (Algorithm 3), the Exhaustive heuristic is used to decide whether convergence to e_1 is possible with k dummies. We now show that without changing the algorithm we can answer the optimization question, namely, what is the lowest-cost link we can converge to using k dummies.

The algorithm differs from the decision version, only in the return statements. In line 4, instead of True it returns e_1 and in line 11, instead of False it returns the minimal i s.t. $e_i \in E_{min}^p$ or $e_i \in BR_{e_{min}^p}(p_{i+})$.

Based on Claim 3.2, when the algorithm returns i such that $e_i \in E_{min}^p$ or $e_i \in BR_{e_{min}^p}(p_{i+})$, then we can converge to e_i . We show that links that were the BR of some link, will always be able to attract a player from some link, and that

Link	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
Cost	3000	3100	3100	3100	3100	6000	6000	10000
Load	0	188	180	179	175	700	620	600
$\frac{c_e}{n_e^p}$	-	16.48	17.22	17.127	17.71	8.57	9.67	16.66
$\frac{c_e}{n_e^{p+1}}$	3000	16.40	17.127	17.22	17.61	8.55	9.66	16.63

TABLE IV
INITIAL PROFILE OF INSTANCE I_2 .

algorithm	Naive	Max CR	Balancing	Exhaustive
# of dummies	350	251	310	240
# of steps	2,642	2,956	2,874	3,043
# of dummy moves	1	312	5	312

TABLE V
RESULTS FOR PROFILE I_2

links that were not the BR of any link until we exhausted our effort to make e_1 the BR of some link, will not be able to be the BR of any link. This implies that the returned link is indeed the best achievable link.

The following claims and observations will be used in the analysis of the Exhaustive Heuristic. The first claim shows that $price_{min}$ is monotonically increasing during the algorithm.

Claim 5.1: $price_{min}^{p+1} \geq price_{min}$

The next claim shows that the load on any link $e_m \in E_{min}^p$ monotonically decreases.

Claim 5.2: For a profile p and $e_m \in E_{min}^p$, n_a^p monotonically decreases after profile p .

We now combine Claims 5.1 and 5.2 to show that, $price_{min}$ strictly increases every $|E_{min}^p|$ iterations and at the worst case after $n - 1$ iterations.

Observation 5.3: For a given profile p , $price_{min}^{p+|E_{min}^p|} > price_{min}^p$

The next claim shows that if e_a can attract players in some profile p , then it can attract additional players in any profile p' that succeeds p .

Claim 5.4: For a profile p and profile p^{+i} (where $i > 0$) reached in a later stage of our algorithm. If $e_a \in BR_{e_{min}^p}(p_{a+})$, then either $e_a \in BR_{e_{min}^{p+i}}(p^{+i}_{a+})$ or $e_a = e_{min}^{p+i}$.

We turn to show that if a link was not the BR of any link at no point of the algorithm it cannot become the BR using k dummies.

Claim 5.5: For a profile p^{+i} ($i \geq 0$) where not $e_1 \in E_{min}^p$ or $e_1 \in BR_{e_{min}^p}(p_{1+})$ or $E_t^p \neq \emptyset$. If $e_a \notin E_{min}^p$ and $e_a \in BR_{e_{min}^p}(p_{a+})$ then we cannot converge into e_a

Proof: Assume by contradiction that for some profile p , $e_a \in BR_{e_{min}^p}(p_{a+})$ but a profile p^{+i} can be reached where $e_a \notin BR_{e_{min}^{p+i}}(p^{+i}_{a+})$ and $e_a \neq e_{min}^{p+i}$. Using Claims 5.1 and 5.2, $n_a^{p+i} < n_a^p$ and $price_{min}^{p+i} \geq price_{min}^p$. Furthermore, if $n_a^{p+i} \geq n_a^p$, meaning he only gained play-

ers, then $e_a \in BR_{e_{min}^p}(p^{+i}_{a+})$ and using Claim 5.3 $e_a \in BR_{e_{min}^{p+i}}(p^{+i}_{a+})$ or $e_a = e_{min}^{p+i}$.

Else $n_a^{p+i} < n_a^p$, then for some profile p^{+j} , where $0 < j < i$ reached between p and p^{+i} $e_{min}^{p+j} = e_a$. If e_a is still the minimum then clearly $e_a = e_{min}^{p+i}$, else it is not the minimum and as seen in Claim 5.2 $\frac{c_a}{n_a^{p+i+k+1}} < price_{min}^{p+i}$ meaning $e_a \in BR_{e_{min}^{p+i}}(p^{+i}_{a+})$. ■

VI. EXPERIMENTAL RESULTS

In this section we present some experimental results, achieved by simulating the heuristics presented in Sections IV and V. The heuristics were performed on random instances in random initial profiles.

We created a test-base consisting of four classes. The first class, denoted *random* includes instances with a random number of links (between 6 and 20), for each link there was a randomly generated cost between 2,000 – 100,000 and load between 0 and 10,000. All values were drawn assuming uniform distribution in their range. Figure 2 shows that in the majority of the random profiles the heuristics are redundant, but there exist a significant amount of profiles in which they are helpful, in those cases the needed number of dummies decrease significantly.

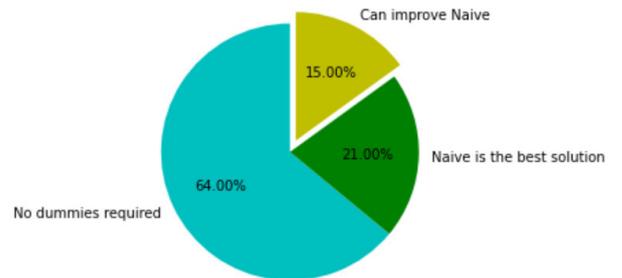


Fig. 2. Percentage of random profiles in which the addition of dummy players is beneficial, and the heuristics are more efficient than the naive solution.

In order to emphasize the differences between the different heuristics, we included in our test-base only instances for which, in at least two heuristics, the required number of dummies is lower than the number of dummies required by the naive solution.

The second class of instances is the most challenging one. It is denoted *Beat The Competitor*. In the initial profiles of this class, e_2 has the highest initial load, therefore, the social optimum, e_1 , has an attractive competitor.

The third class of instances is denoted *"Beat The Giants"*. In the initial profiles of this class, the initial load on e_1 is very low. There are a few heavily loaded links which cost several times the cost of e_1 . Without the use of dummies, any BRD will converge to one of them. In addition there is a larger number of contender links with a lower cost than the heavy links and less attractive than them, but they can attract players from the heavy links with less dummies than e_1 .

The last class of instances is denoted *"Beat The Median"*, it is a generalization of the above class. It is similar to *Beat The Giants* with an addition of few links that are even heavier than the heavy links of *Beat The Giants* and the players on those links pay around the same cost as the players on the contender links.

We first present our results for the problem considered in Section IV: what is the number of dummies required to converge to the social optimum. Thus, we fix the social cost of the final profile, and measure the 3 other parameters characterizing the quality of a solution.

Figure 3 shows the number of dummies required to converge to the social optimum scaled compared to the naive solution, and averaged on all instances in the test-base. As shown in the figure, the heuristics can achieve the social optimum with significantly less dummies than the naive solution. In particular, the exhaustive heuristic is always the solution that uses the least number of dummies.

Only In 1% out of the random profiles the balancing algorithm was better than the naive solution. On the other hand, for the classes *Beat The Giants / Median*, the balancing phase is essential.

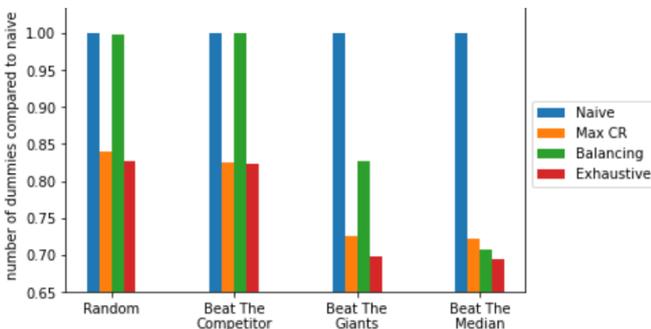


Fig. 3. Number of dummies required in order to converge to the social optimum, compared to the naive solution.

Figure 4 compares the length of the BR-sequence till convergence to e_1 , scaled by n_1 , which is the number of players that need to migrate to e_1 . That is, how many times each player is activated on average. Recall that in the naive solution, every player, except for the players that are assigned on e_1 in p^0 , migrates exactly once. Clearly, this is our lower bound. We can see that all the heuristics perform well with respect to this measure. Specifically, even in the longest sequences, players migrate on average at most 1.125 times, but we can clearly see that in the more specific profiles *Beat The Giants / Median* the length of BR-sequence is larger than the naive solution while

Random and *Beat The Competitor* lengths on all heuristics are close to the optimal solution.

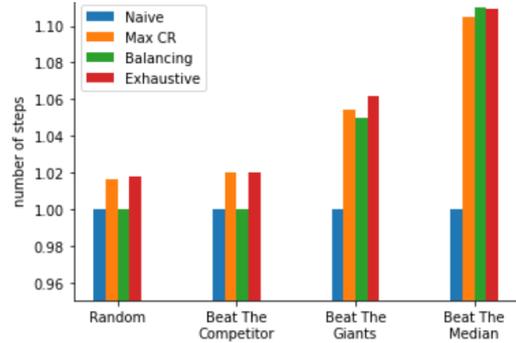


Fig. 4. Length of BR-sequence, measured by the average number of migrations performed by each player that is not on e_1 in p^0 .

The next measure we consider is the number of times the dummy players are moved. Again, the naive heuristic provides the lower bound, as the dummy players are assigned exactly once – on e_1 . Figure 5 presents the results for this measure, scaled by the number of links in the network. The naive solution assigns the dummy players only once, and the balancing heuristic move the dummies at most $|E_{gain}|$ times. In contrast, the exhaustive and max cost-reduction heuristics may migrate the dummies many times.

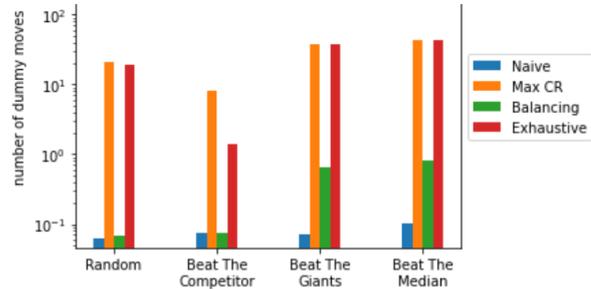


Fig. 5. Number of dummy moves scaled by the number of links

We turn to present our results for the problem considered in Section V: what is the lowest cost achievable link, given a limited budget k of dummy players. Thus, the interesting measure of a heuristic is the resulting social cost.

Figure 6 presents the social costs achieved by each Heuristic where the given k is a parameter which is a percentage of $k_{naive}(p^0)$. (naive is the minimal i we can achieve using one assignment of dummies on some link)

In the chart, the Social cost is divided by the Social Optimum showing how far is the result from the Social optimum.

We can see that in the *Random* and *"Beat The Competitor"*, the separation from the naive solution is shown in the higher percentage of dummies but in *Beat The Giants / Median* there is a difference even in the lower percentages.

VII. CONCLUSIONS AND OPEN PROBLEMS

In this work we demonstrated the power of a temporal addition of dummy players to a game. The dummy players

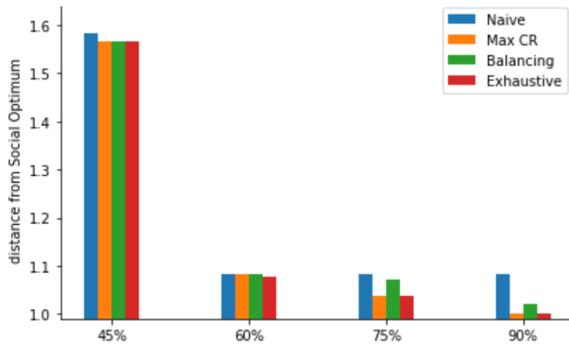


Fig. 6. "Beat The Giants" social cost by algorithm and percentage of dummies from the naive solution, divided by social optimum

initiate a dynamic in which the players are encouraged to reach a Nash Equilibrium profile of better quality. We suggested several heuristics for operating the dummies, and analyzed their quality distinguishing between the number of dummy players, the value of the final solution, and the convergence time. Our main message is that the use of dummy players may significantly improve the equilibrium inefficiency. In general, finding an optimal algorithm for exploiting the dummies is NP-hard. However, as we show, even simple heuristics may need significantly less dummies than a naive solution, and the quality of the final solution can improve further by extending the length of the BR-sequence by a factor of 1.15, and if the dummy players can be migrated intensively. Practically, in real-world application such as routing, the possibility of creating a controlled fake load or by temporarily disable the use of some resources, can help migrate the players to routes that improve the global system's performance.

The addition of dummy player is only one possible perturbation of a stable solution. It would be interesting to study the power of additional temporal perturbation in resource allocation games, that refer not only to the set of participating clients, but to set of resources, e.g., temporal closure or addition of resources or temporal change in resources'

activation cost.

REFERENCES

- [1] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.
- [2] V. Bilò and C. Vinci. On the impact of singleton strategies in congestion games. In *Proc. 25th Annual European Symposium on Algorithms*, pages 17:1–17:14, 2017.
- [3] I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. *Algorithmica*, 61(3):606–637, 2011.
- [4] E. Even-Dar and Y. Mansour. Fast Convergence of Selfish Rerouting. In *Proc. of SODA*, pp. 772–781, 2005.
- [5] M. Feldman, Y. Snappir, and T. Tamir. The efficiency of best-response dynamics. In *The 10th International Symposium on Algorithmic Game Theory (SAGT)*, 2017.
- [6] A. Fanelli, M. Flammini, and L. Moscardelli. Stackelberg strategies for network design games. In *Proc. of International Workshop on Internet and Network Economics (WINE)*, pp. 222–233, 2010.
- [7] D. Fotakis. Stackelberg strategies for atomic congestion games. *Theory of Computing Systems*, 47(1):218–249, 2010.
- [8] T. Harks and M. Klimm. On the existence of pure nash equilibria in weighted congestion games. *Math. Oper. Res.*, 37(3):419–436, 2012.
- [9] S. Jeong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, AAAI'05*, 2005.
- [10] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using stackelberg routing strategies. *IEEE/ACM Trans. Netw.*, 5(1):161–173, 1997.
- [11] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.
- [12] W. Krichene, J. D. Reilly, S. Amin, and A. M. Bayen. Stackelberg routing on parallel networks with horizontal queues. *IEEE Transactions on Automatic Control*, 59(3):714–727, 2014.
- [13] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1):111 – 124, 1996.
- [14] D. Monderer and L. S. Shapley. Potential Games. *Games and Economic Behavior*, 14: 124–143, 1996.
- [15] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–753, 2001.
- [16] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- [17] T. Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2):332–350, 2004.
- [18] T. Tamir. The power of one evil secret agent. *Theoretical Computer Science*, 839:1–12, 2020.
- [19] B. Vöcking. *Algorithmic Game Theory*, chapter 20: Selfish Load Balancing. Cambridge University Press, 2007.