Selfish Bin Packing

Leah Epstein[1] and Elena Kleiman[1]

Department of Mathematics, University of Haifa, 31905 Haifa, Israel.
lea@math.haifa.ac.il, elena.kleiman@gmail.com.

Abstract. Following recent interest in the study of computer science problems in a game theoretic setting, we consider the well known bin packing problem where the items are controlled by selfish agents. Each agent is charged with a cost according to the fraction of the used bin space its item requires. That is, the cost of the bin is split among the agents, proportionally to their sizes. Thus, the selfish agents prefer their items to be packed in a bin that is as full as possible. The social goal is to minimize the number of the bins used. The social cost in this case is therefore the number of bins used in the packing.
A pure Nash equilibrium is a packing where no agent can obtain a smaller cost by unilaterally moving his item to a different bin, while other items remain in their original positions. A Strong Nash equilibrium is a packing where there exists no subset of agents, all agents in which can profit from jointly moving their items to different bins. We say that all agents in a subset profit from moving their items to different bins if all of them have a strictly smaller cost as a result of moving, while the other items remain in their positions.
We measure the quality of the equilibria using the standard measures PoA and PoS that are defined as the worst case worst/best asymptotic ratio between the social cost of a (pure) Nash equilibrium and the cost of an optimal packing, respectively. We also consider the recently introduced measures $SPoA$ and $SPoS$, that are defined similarly to the PoA and the PoS, but consider only Strong Nash equilibria.
We give nearly tight lower and upper bounds of 1.6416 and 1.6428, respectively, on the PoA of the bin packing game, improving upon previous result by Bilò, and establish the fact that $PoS = 1$. We show that the bin packing game admits a Strong Nash equilibrium, and that $SPoA$=$SPoS$. We prove that this value is equal to the approximation ratio of a natural greedy algorithm for bin packing.

1 Introduction

Motivation and framework. In the last few decades, we have witnessed the tremendous development of the Internet and its penetration into almost any aspect of our lives, influencing the society on a scope not known before. The emergence of the Internet created a major shift in our view of computational networking systems. Traditional system design assumes that all participants behave according to some protocol that serves the intentions of the system designers and the users often have to sacrifice some of their own performance for the sake of the entire network. Unlike any other distributed system, the Internet is

built, operated and used by various autonomous and self-interested entities, in different levels of competition and cooperation relationship with one another. These entities (or *agents*) have diverse sets of interests and aim at achieving their individual goals as opposed to obtaining a global optimum of the system. Hence, selfishness is an inherent characteristic of the Internet. Also, there exists no central authority that can enforce a certain policy or regulation on the participants of the system. Under these assumptions, network optimization problems that model situations where rational agents compete each other over network resources while seeking to satisfy their individual requirements at minimum cost, can be viewed as non-cooperative strategic games considered by the classical Game Theory. It becomes natural to revisit various aspects concerning networking under a Game-Theoretic perspective, using the tools that the Game Theory provides us with. In particular, we are interested in quantifying the loss to the system performance caused by the lack of cooperation and the selfishness of the players. This is achieved by analyzing the *Nash equilibrium* (the main concept of stability in Game Theory) of the game and comparing it to the social optimum.

In reality, in many networking applications the multitude of entities that use and operate the system are not completely autonomous. They interact with each other and form *coalitions*, members of which agree to coordinate their actions in a mutually beneficial way. Of course, these entities still remain selfish. Thus, each agent will agree to participate, if at all, in a coalition that ensures him a benefit from participation in that coalition. This scenario evokes the Coalitional Game Theory and the concept of *Strong Nash equilibrium*. Considering the possibility of players to gather in coalitions allows us to separate the effect incurred to the system performance due to selfishness from that of lack of coordination (which disappears if we let the participants of the game to cooperate).

In this paper, we consider the well-known Bin Packing problem (see e.g. [16], [17], [7] for surveys). The basic Bin Packing problem consists of packing a set of objects with sizes in (0,1] into a set of unit-capacity bins while using as few bins as possible. Among other important real-life applications, such as multiprocessor scheduling and stock cutting, the Bin Packing problem can be met in a great variety of network problems. For example, the packet scheduling problem (the problem of packing a given set of packets into a minimum number of time slots for fairness provisioning), the bandwidth allocation problem (signals have usually a small size and several of them can be transmitted in the same frame so as to minimize bandwidth consumption) and the problem of packing the data for Internet phone calls into ATM packets (filling fixed-size frames to maximize the amount of data that they carry), to mention only a few. Therefore, the study of this problem from a Game-Theoretic standpoint is clearly well motivated.

Definitions and notations. To establish notation, we will briefly introduce the basic concepts from Game Theory. A non-cooperative strategic game is a tuple $G = \langle N, (S_i)_{i \in N}, (c_i)_{i \in N} \rangle$, where N is a non-empty, finite set of players, each player $i \in N$ has a non-empty, finite set S_i of *strategies* (actions) and a cost function c_i. Each player chooses a strategy independently of the choices of the other players. The choices of all players can thus be thought to be made simultaneously. It is assumed that each player has a full knowledge over all strategy sets of all the players. In a setting of *pure* strategies, each player chooses

exactly one strategy (with probability one); in a setting of *mixed* strategies, each player uses a probability distribution over the strategies. A combination of strategies chosen by the players $s = (x_j)_{j \in N} \in \times_{j \in N} S_j$, is called a *strategy profile* or a *configuration*. $X = \times_{j \in N} S_j$ denotes the set of the strategy profiles. Let $i \in N$. $X_{-i} = \times_{j \in N \setminus \{i\}} S_j$ denotes the strategy profiles of all players except player i. Let $A \subseteq N$. $X_A = \times_{j \in A} S_j$ denotes the set of strategy profiles of players in A. Strategy profiles $s = (x_j)_{j \in N} \in X$ will be denoted by (x_i, x_{-i}) or $(x_A, x_{N \setminus A})$ if the strategy choice of player i or of the set A of players needs stressing. The cost function $c_i : X \to \mathbb{R}$ specifies for each strategy profile $s \in X$ the cost charged from player i, $c_i(x) \in \mathbb{R}$. The cost charged from each player depends not only on his own strategy but also on the strategies chosen by all other players. Each player $i \in N$ would prefer to chose a strategy that minimizes his cost. The accepted concept of rationality in a game is the *Nash equilibrium* [24]. Throughout the paper, the Nash equilibrium is considered only in the setting of pure strategies.

Definition 1. *A strategy profile $s \in X$ is called a pure Nash equilibrium (NE) if for every i and for all $x'_i \in S_i$, $x'_i \neq x_i$, $c_i(x_i, x_{-i}) \leq c_i(x'_i, x_{-i})$ holds. That is, no player can reduce his cost by unilaterally changing his strategy, while the strategies of all other players remain unchanged.*

Nash equilibrium (perhaps only in mixed strategies) exists in every finite game [24]. A game can have several Nash equilibria, with different social cost values. If only pure strategies are allowed, there may exist no Nash equilibrium at all. The set of pure Nash equilibria of a game G is denoted by $NE(G)$.

Games as defined above assume that players can not negotiate and cooperate with each other. *Coalitional Game Theory* considers cooperative games, where the notion of players is replaced by the set of possible coalitions (i.e., groups of players) rather than individuals. A participation in a coalition is voluntary. Each coalition can achieve a particular value (the smallest possible sum of costs among players in the coalition, against worst-case behavior of players outside the coalition). Aumann [4] introduced the concept of Strong Nash equilibrium. Since each player can either participate or decline to participate in a coalition, given the strategy he will be obligated to choose in case he does, and the cost he will be charged with as a result, the Strong Nash equilibrium is studied only for settings that involve no randomization, that is, only pure strategies are considered.

Definition 2. *A strategy profile $s \in X$ is called a Strong Nash equilibrium (SNE) if for every $S \subseteq N$ and for all strategy profiles $y_S \in X_S$, there is at least one player $i \in S$ such that $c_i(x_S, x_{-S}) \leq c_i(y_S, x_{-S})$. That is, no subset of players can deviate by changing strategies jointly in a manner that reduces the costs charged from all its members, given that nonmembers stick to their original strategies.*

The set of Strong Nash equilibria of a game G is denoted by $SNE(G)$. Every Strong Nash equilibrium is a Nash equilibrium (by definition). Hence, $SNE(G) \subseteq NE(G)$. The opposite does not usually hold. A game can have no Strong Nash equilibrium at all. Several specific classes of congestion games were shown in [15, 27] to possess Strong Nash equilibria. For any other game, the existence of

Strong equilibria should be checked specifically in each case. Other variants of Strong equilibria studied consider static predefined coalitions [14, 12] and coalitions that are not subject to deviations by subsets of their own members [29].

The *social cost* of a game G, is an objective function $SC(s) : X \to \mathbb{R}$ that numerically expresses the 'social cost" of an outcome of the game for a strategy profile $s \in X$. The *social optimum* of a game G, is the game outcome that optimizes the social cost function. It is denoted by $OPT(G)$, and defined by $OPT(G) = \min_{s \in X} SC(s)$.

2 The bin packing game

The model. The bin packing problem consists of packing a set N of items, each item $i \in N$ having a size $a_i \in (0, 1]$, into a set of unit-capacity bins while using as few bins as possible. The induced bin packing game BP is defined by a tuple $BP = \langle N, (B_i)_{i \in N}, (c_i)_{i \in N} \rangle$, where N is the set of selfish players. Each player $i \in N$ controls a single item with size $a_i \in (0, 1]$ and selects the bin to which this item is packed. We identify the player with the item he wishes to pack. Thus, the set of players corresponds to the set of items. The set of strategies B_i for each item $i \in N$ is the set of all possible open bins. Each item can be assigned to one bin only. Splitting items among several bins is not allowed. The outcome of the game is a particular assignment $(b_j)_{j \in N} \in \times_{j \in N} B_j$ of items to bins of equal capacity. Let X denote the set of all possible assignments. All the bins have the same fixed cost which equals their capacity and the cost of a bin is shared among all the items it contains. The cost function of item i is c_i. If we scale the cost and the size of each bin to one, the cost paid by item i for choosing to be packed in bin $\mathcal{B}_j$ such that $j \in B_i$ is defined by $c_i(j, b_{-i}) = \dfrac{a_i}{\sum_{k:b_k=j} a_k}$, when $b_{-i} \in X_{-i}$; i.e, an item is charged with a cost which is proportional to the portion of the bin it occupies in a given packing. We consider the cost charged from an item for being packed in a bin in which it does not fit to be ∞. The items are interested in being packed in a bin so as to minimize their cost. Thus, item i packed into $\mathcal{B}_j$ in a particular assignment $(b_j)_{j \in N}$ will migrate from $\mathcal{B}_j$ each time it will detect another bin $\mathcal{B}_{j'}$ such that $c_i(j', b_{-i}) < c_i(j, b_{-i})$. This inequality holds for each j' such that $\sum_{k:b_k=j'} a_k + a_i > \sum_{k:b_k=j} a_k$, thus an item will perform an improving step each time it will detect a strictly more loaded bin in which it fits. At a Nash equilibrium, no item can unilaterally reduce its cost by moving to a different bin. The social cost function that we want to minimize is the number of used bins.

In the cooperative version of the game, we consider all possible (non-empty) groups of items $A \subseteq N$. A group can contain a single item. The cost functions of the players are defined the same as in the non-cooperative case. Each group of items is interested to be packed in a way so as to minimize the costs of all group members. Thus, given a particular assignment, all members of group A will perform a joint improving step if there is a configuration in which, for each member, the new bin will admit a strictly greater load than the bin of origin. The costs of the non-members may be enlarged as a result of this step.

At a Strong Nash equilibrium, no group of items can reduce the costs of its members by jointly moving to a different bin. The social cost function remains the same one we consider in the non-cooperative setting.

Measuring the inefficiency of the equilibria. It is well-known that Nash equilibrium does not always optimize the social cost function. Even in very simple settings, selfish behavior can lead to highly inefficient outcome. Our bin packing game is no exception: an equilibrium packing does not necessarily have minimum cost. Note also that not every optimal solution is an equilibrium.

The quality of an equilibrium is measured with respect to the social optimum. In the bin packing game, the social optimum is the number of bins used in a coordinated optimal packing. In the computer science literature, the Price of Anarchy *(PoA)* [20, 25] (also referred to as the Coordination Ratio *(CR)*) and the Price of Stability (*PoS*) [3, 2] (also called optimistic price of anarchy) have become prevalent measures of the quality of the equilibria reached with uncoordinated selfish players. The Price of Anarchy/ Price of Stability of a game G are defined to be the ratio between the cost of the worst/best Nash equilibrium and the social optimum, respectively. Formally,

$$PoA(G) = \sup_{s \in NE(G)} \frac{SC(s)}{OPT(G)}, \quad PoS(G) = \inf_{s \in NE(G)} \frac{SC(s)}{OPT(G)}.$$

The former quantifies the worst possible loss to performance incurred by selfish uncoordinated agents, and the latter measures the minimum penalty in performance required to ensure a stable equilibrium outcome.

The bin packing problem is usually studied via asymptotic measures. The asymptotic *PoA* and *PoS* of the bin packing game *BP* are defined by

$$PoA(BP) = \limsup_{OPT(G) \to \infty} \sup_{G \in BP} PoA(G), \; PoS(BP) = \limsup_{OPT(G) \to \infty} \sup_{G \in BP} PoS(G).$$

Recent research by Andelman et al. [1] initiated a study of measures that separate the effect of the lack of coordination between players from the effect of their selfishness. The measures considered are the Strong Price of Anarchy (*SPoA*) and the Strong Price of Stability (*SPoS*). These measures are defined similarly to the *PoA* and the *PoS*, but only Strong equilibria are considered. We define the Strong Price of Anarchy/ Strong Price of Stability of a game G as the ratio between the cost of the worst/best Strong Nash equilibrium and the social optimum, respectively. Formally,

$$SPoA(G) = \sup_{s \in SNE(G)} \frac{SC(s)}{OPT(G)}, \quad SPoS(G) = \inf_{s \in SNE(G)} \frac{SC(s)}{OPT(G)},$$

As before, we define the asymptotic *SPoA* and *SPoS* of the bin packing game *BP* by

$$SPoA(BP) = \limsup_{OPT(G) \to \infty} \sup_{G \in BP} SPoA(G)$$

$$SPoS(BP) = \limsup_{OPT(G) \to \infty} \sup_{G \in BP} SPoS(G).$$

3 Related work and our contributions

Related work. The application of concepts and techniques borrowed from Game Theory to various problems in computer science, specifically, to network problems, was initiated in [20, 25]. Since then, issues like *routing* [28, 23, 8], *bandwidth allocation* [30], and *congestion control* [18], to name only a few, have been analyzed from a Game-Theoretic perspective. The studied models are simplification of problems arising in real networks, that seem appropriate for describing basic network problems. The bin packing problem discussed in this paper belongs to a class of problems induced by selfish flow routing in non-cooperative networks. The first model studied in that context is the KP model introduced by Koutsoupias and Papadimitriou in [20]. This model features a network consisting of two nodes, a source and a destination, connected by a set of parallel links, all with the same bandwidth capacity, and a set of selfish users, each wishing to route a certain amount of flow from the source to the destination. The delay suffered by each user for utilizing a link equals to total amount of flow routed through this link. Hence, the more flow routed on a specific link, the longer the delay. For such a reason, users, which are assumed to be selfish, want to route their flow on the least loaded link. The goal is minimize the greatest delay. The resulting problem can be viewed as a selfish job scheduling problem. The bounds on the PoA for the aforementioned model were initially analyzed both in pure and mixed strategies setting in [20]. They were later improved by [23], and definitively characterized in [9, 19]. The existence of pure Nash equilibrium in this setting was proved in [11]. The cooperative version of the job scheduling problem was first studied in [1]. The authors proved that job scheduling games admit Strong equilibria, established the fact that $SPoS = 1$ as for every instance of the scheduling game there exists an optimal solution which is a Strong equilibrium, and gave non-tight bounds on the $SPoA$ that were later definitively characterized in [10]. Since then, many variants and generalizations of this basic model have been studied, with different network topology, different social costs, different nature of the flow, etc.. See for example [28, 26, 22].

The selfish bin packing problem defined above can also be interpreted as a routing problem. Consider a network consisting of two nodes, a source and a destination, connected by a potentially infinite number of parallel links having the same bandwidth capacity, and a set of users wishing to send a certain amount of unsplittable flow between the two nodes. To establish a link, one has to pay a fixed cost which equals the capacity of the link. The cost of each link is shared among the users routing their flow on that link according to the normalized fraction of its utilized bandwidth. For such a reason, users, who are assumed to be selfish, want to route their traffic on the most loaded link. The goal is to minimize the number of links used. This model resembles the KP model with different cost and social functions. It was suggested by Bilò in [5].

Bilò [5] was the first to study the bin packing problem under game theoretic perspective. He proved that the bin packing game admits pure Nash equilibria and provided non-tight bounds on the Price of Anarchy. He also proved that the bin packing game converges to a pure Nash equilibrium in a finite sequence of selfish improving steps, starting from any initial configuration of the items.

The *Subset Sum*[1] algorithm for bin packing we refer to in the sequel of this paper, is a greedy algorithm that repeatedly solves a one-dimensional knapsack problem for packing each bin in turn. It was originally suggested by Prim and first mentioned by Graham [13], who also gave a lower bound of $\sum_{k=1}^{\infty} \frac{1}{2^k-1} = 1.6067$ on its asymptotic worst-case performance. An upper bound of 1.6210 was proved only recently by Caprara and Pferschy in [6].

Our results and organization of the paper. In this paper we consider the pin packing game, in a variant originally proposed and analyzed by Bilò in [5]. We establish that for every instance of bin packing game there exists an optimal *NE* packing where the social cost is equal to the social optimum; in other words, $PoS = 1$. We also give improved (and nearly tight) lower and upper bounds on the *PoA* of the bin packing game. We extend the results in [5] and show that bin packing game admits Strong Nash equilibria as well. Moreover, we show that the aforementioned *Subset Sum* algorithm in fact produces an assignment that admits Strong equilibrium. Therefore, we provide an exponential time deterministic algorithm with guaranteed (asymptotic) worst-case performance ratio [6] that actually calculates the Strong Nash assignment for each bin. Interestingly, the *SPoA* equals the *SPoS*, and we prove this value is equal to the approximation ratio of the *Subset Sum* algorithm. Thus, we provide bounds on the *SPoA* and the *SPoS* of the game.

Our results for the *PoA* improve upon previous results of Bilò [5]. The other concepts were not addressed to the bin packing framework prior to this paper, to the best of our knowledge. Our contributions can therefore be summarized in Table 1. Some of the proofs were omitted due to space constraints.

Table 1. Summary of the results.

		Lower Bound	Upper Bound
PoA	Bilò [5]	1.6	1.6667
	Our paper	1.6416	1.6428
SPoA=*SPoS*	Our paper	1.6067	1.6210
PoS	Our paper	1	1

4 The Price of Stability

In our first result, we establish that for every instance of the bin packing game there always exists a packing, among the optimal ones, which is a *NE*. We do it by introducing an order relation similar to the one used by Fotakis et al. in [11] between the different configurations and showing that an optimal packing which is the "highest" among all optimal packings according to this order is always a *NE*. Specifically, in this section we prove the following theorem.

Theorem 1. *For every instance of the bin packing game there is a NE packing which is optimal.*

[1] Also called *fill bin* or *minimum bin slack* in the literature.

Definition 3. *For a configuration b, the load vector is an n-tuple $L(b) = (L_1(b), L_2(b), \ldots, L_n(b))$, where each component $L_i(b)$ is the load of bin $\mathcal{B}_i$ in a packing defined by b.*

Definition 4. *A vector $u = (u_1, u_2, \ldots, u_n)$ is greater than $v = (v_1, v_2, \ldots, v_n)$ lexicographically, if there is some $k \geq 1$ such that $u_i = v_i$ for $i = 1, \ldots k-1$, and $u_k > v_k$.*

We define a sorted lexicographic order on the configurations via the lexicographic order on the vectors.

Definition 5. *Let b and b' be two configurations with the corresponding load vectors $L(b) = (L_1(b), L_2(b), \ldots, L_n(b))$ and $L(b') = (L_1(b'), L_2(b'), \ldots, L_n(b'))$. A configuration b' is greater than b lexicographically, if and only if the load vector $L(b')$ sorted in non-increasing order is greater lexicographically than $L(b)$, sorted in non-increasing order. We denote this relation by $b' \succ_L b$.*

The sorted lexicographic order $\succ_L$ defines a total order on the configurations. Next, we show that when an item migrates, we move to a "higher" configuration in the order.

Lemma 1. *The sorted lexicographic order of the load vector always increases when an item migrates.*

Lemma 2. *For any instance of the bin packing game, the lexicographically maximal optimal packing b^* is a NE.*

Theorem 1 now follows from Lemmas 1 and 2. An immediate conclusion from Theorem 1 is that the upper bound on the Price of Stability (PoS) of the bin packing game is 1. Combined with the fact that $PoS(G) \geq 1$ for any $G \in BP$ as no equilibrium point can be better than the social optimum, we conclude that $PoS(BP) = 1$.

5 The Price of Anarchy

We now provide a lower bound for the Price of Anarchy of the bin packing game and also prove a very close upper bound.

5.1 A lower bound: construction

In this section, we present our main technical contribution, which is a lower bound on the PoA. We start with presenting a set of items. The set of items consists of multiple levels. Such constructions are sometimes used to design lower bounds on specific bin packing algorithms (see e.g., [21]). Our construction differs from these constructions since the notion of order (in which packed bins are created) does not exist here, and each bin must be stable with respect to all other bins. The resulting lower bound on the PoA is different from any bounds known on the asymptotic approximation ratio of well known algorithms for bin packing. Since we prove an almost matching upper bound, we conclude that the PoA is probably not related directly to any natural algorithm. We prove the following theorem.

Theorem 2. *The Price of Anarchy of the bin packing game is at least the sum of the following series:* $\sum_{j=1}^{\infty} 2^{-j(j-1)/2}$*, which is equal to approximately* 1.64163.

Proof. Let $s > 2$ be an integer. We define a construction with s phases of indices $1 \leq j \leq s$, where the items of phase j have sizes which are close to $\frac{1}{2^j}$, but can be slightly smaller or slightly larger than this value.

We let $OPT = n$, and assume that n is a large enough integer, such that $n > 2^{s^3}$. We use a sequence of small values, δ_i such that $\delta_j = \frac{1}{(4n)^{3s-2j}}$. Note that this implies $\delta_{j+1} = (4n)^2\delta_j$ for $1 \leq j \leq s-1$. We use two sequences of positive integers $r_j \leq n$ and $d_j \leq n$, for $2 \leq j \leq s$, and in addition, $r_1 = n$ and $d_1 = 0$. We define $r_{j+1} = \frac{r_j - 1}{2^j}$, for $1 \leq j \leq s-1$, and $d_{j+1} = r_j - r_{j+1} = \frac{(2^j-1)r_j+1}{2^j} = (2^j-1)r_{j+1} + 1$.

Proposition 1. *For each* $1 \leq j \leq s$, $\frac{n}{2^{j(j-1)/2}} - 1 < r_j \leq \frac{n}{2^{j(j-1)/2}}$.

Phase 1 simply consists of r_1 items of size $\sigma_1 = \frac{1}{2} + 2(d_1+1)\delta_1$. For $j \geq 2$, phase j consists of the following $2d_j + r_j$ items. There are r_j items of size $\sigma_j = \frac{1}{2^j} + 2(d_j+1)\delta_j$, and for $1 \leq i \leq d_j$, there are two items of sizes $\pi_j^i = \frac{1}{2^j} + (2i-1)\delta_j$ and $\theta_j^i = \frac{1}{2^j} - 2i\delta_j$. Note that $\pi_j^i + \theta_j^i = \frac{1}{2^{j-1}} - \delta_j$.

The packing will contain d_j bins of level j, for $2 \leq j \leq s$, and the remaining bins are of level $s+1$, where a bin of level j, contains only items of phases $1, \ldots, j$.

To show that we can allocate these numbers of bins, and to calculate the number of level $s+1$ bins, note that $\sum_{j=2}^{s} d_j = r_1 - r_s = n - r_s$. Thus, the number of level $s+1$ bins is (at most) r_s.

The packing of a bin of a given level is defined as follows. For $2 \leq j \leq s$, a level j bin contains one item of each size σ_k for $1 \leq k \leq j-1$, and in addition, one pair of items of sizes π_j^i and θ_j^i for a given value of i such that $1 \leq i \leq d_j$. A bin of level $s+1$ contains one item of each size σ_k for $1 \leq k \leq j-1$.

Proposition 2. *This set of items can be packed into* n *bins, i.e.,* $OPT \leq n$

We next define an alternative packing, which is a *NE*. In the sequel, we apply a modification to the input by removing a small number of items. Clearly, $OPT \leq n$ would still hold for the modified input.

Our modification to the input is the removal of items π_j^1 and $\theta_j^{d_j}$ for all $2 \leq j \leq s$. We construct r_j bins for phase j items. A bin of phase j consists of $2^j - 1$ items, as follows. One item of size $\sigma_j = \frac{1}{2^j} + 2(d_i+1)\delta_i$, and $2^{j-1} - 1$ pairs of items of phase j. A pair of items of phase j is defined to be the items of sizes π_j^{i+1} and θ_j^i, for some $1 \leq i \leq d_j - 1$. The sum of sizes of this pair of items is $\frac{1}{2^j} + (2i+1)\delta_j + \frac{1}{2^j} - 2i\delta_j = \frac{1}{2^{j-1}} + \delta_j$. Using $d_j = (2^{j-1}-1)r_j + 1$ we get that all phase j items are packed. The sum of items in every such bin is $1 - \frac{1}{2^{j-1}} + (2^{j-1}-1)\delta_j + \frac{1}{2^j} + 2(d_j+1)\delta_j = 1 - \frac{1}{2^j} + \delta_j(2^{j-1} + 1 + 2d_j)$.

Proposition 3. *The loads of the bins in the packing defined above are monotonically increasing as a function of the phase.*

Proposition 4. *The packing as defined above is a valid NE packing.*

Finally, we bound the *PoA* as follows. The cost of the resulting *NE* is $\sum_{j=1}^{s} r_j$. Using Proposition 1 we get that $\sum_{j=1}^{s} r_j \geq \sum_{j=1}^{s} (\frac{n}{2^{j(j-1)/2}} - 1)$ and since $OPT = n >> s$, we get a ratio of at least $\sum_{j=1}^{s} 2^{-j(j-1)/2}$. Letting s tend to infinity as well results in the claimed lower bound. □

5.2 An upper bound

To bound the *PoA* from above, we prove the following theorem.

Theorem 3. *For any instance of the bin packing game $G \in BP$: Any NE packing uses at most $1.64286 \cdot OPT(G) + 2$ bins, where $OPT(G)$ is the number of bins used in a coordinated optimal packing.*

6 Bounding the SPoA and the SPoS

The *SPoA* and the *SPoS* measures are well defined only when a Strong equilibrium exists. Our Bin Packing game does not belong to the set of games that were already shown to admit a Strong equilibrium. Thus, in order to analyze the *SPoA* and the *SPoS* of the Bin Packing game, we must first prove that a Strong equilibrium exists in our specific setting.

Theorem 4. *For each instance of the bin packing game, the set of Strong Nash equilibria is non-empty.*

Proof. We give a constructive proof to this theorem, by providing a deterministic algorithm that, for each instance of the bin packing game, produces a packing which admits *SNE*. This is the well-known *Subset Sum* algorithm, that proceeds by filling one bin at a time with a set of items that fills the bin as much as possible. We will show a stronger result; For the Bin Packing game introduced above, the set of *SNE* and the set of outcomes of *Subset Sum* algorithm coincide. A proof of this result is given in two parts.

Proposition 5. *The output of the Subset Sum algorithm is always a SNE.*

Proposition 6. *Any SNE is an output of some execution of the Subset Sum algorithm.*

As the *Subset Sum* algorithm is deterministic, Proposition 5 shows that an *SNE* always exists. □

Now, we would like to show that for the bin packing game, $SPoA$ equals $SPoS$, and that this value is equal to the approximation ratio of the *Subset Sum* algorithm. For shortening notation, from now on, we refer to the *Subset Sum* algorithm as to algorithm C, and denote its approximation ratio by R_C.

Theorem 5. *For the bin packing game introduced above,* $SPoA = SPoS = R_C$.

Theorem 5 implies that the problem of bounding the *SPoA* and the *SPoS* of the bin packing game is equivalent to the problem of bounding the approximation worst-case ratio R_C of the well known *Subset Sum* algorithm for bin packing. The latter was tackled by Caprara and Pferschy, who used a novel and non-trivial method to show $1.6067 \leq R_C \leq 1.6210$, thus determining the exact value of R_C within a relative error smaller than 1% (see [6]). This exact value is yet to be found. We conclude that $1.6067 \leq SPoS(BP) = SPoA(BP) \leq 1.6210$.

7 Summary and conclusions

We have studied the Bin Packing problem, where the items are controlled by selfish agents, and the cost charged from each bin is shared among all the items packed into it, both in non-cooperative and cooperative versions. We proved a tight bound on the *PoS* and provided improved and almost tight upper and lower bounds on the *PoA* of the induced game. We have also provided a simple deterministic algorithm that computes the *SNE* assignment for any instance of the Bin Packing game, and proved that the asymptotic worst-case performance of this algorithm equals the *SPoA* and the *SPoS* values of the game. Two open problems in that context are closing of the small gaps between upper and lower bounds for the *PoA* and the *SPoA/SPoS* of the bin packing game. The latter, if achieved, will result in giving a tight bound on the worst-case performance of the *Subset Sum* algorithm for bin packing, as we proved these two problems are equivalent. This probably would not be an easy task, as finding tight bound on the approximation ratio of the *Subset Sum* algorithm, though very nearly approximated by Caprara and Pferschy in [6], remains open problem since 70's.

References

1. N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. In *SODA*, pages 189–198, 2007.
2. E. Anshelevich, A. Dasgupta, J. M. Kleinberg, É.Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS*, pages 295–304, 2004.
3. E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *STOC*, pages 511–520, 2003.
4. R. J. Aumann. Acceptable points in general cooperative n-person games. In A. W. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV, Annals of Mathematics Study 40*, pages 287–324. Princeton University Press, 1959.
5. V. Bilò. On the packing of selfish items. In *IPDPS*. IEEE, 2006.
6. A. Caprara and U. Pferschy. Worst-case analysis of the subset sum algorithm for bin packing. *Oper. Res. Lett.*, 32(2):159–166, 2004.
7. E. Coffman Jr. and J. Csirik. Performance guarantees for one-dimensional bin packing. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, chapter 32, pages (32–1)–(32–18). Chapman & Hall/Crc, 2007.
8. A. Czumaj. Selfish routing on the internet. In J. Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 42. CRC Press, 2004.

9. A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms*, 3(1), 2007.
10. A. Fiat, H. Kaplan, M. Levy, and S. Olonetsky. Strong price of anarchy for machine load balancing. In *ICALP2007*, pages 583–594, 2007.
11. D. Fotakis, S. C. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *ICALP*, pages 123–134, 2002.
12. D. Fotakis, S. C. Kontogiannis, and P. G. Spirakis. Atomic congestion games among coalitions. In *ICALP Part I*, pages 572–583, 2006.
13. R. L. Graham. Bounds on multiprocessing anomalies and related packing algorithms. In *Proceedings of the 1972 Spring Joint Computer Conference*, pages 205–217, 1972.
14. A. Hayrapetyan, É. Tardos, and T. Wexler. The effect of collusion in congestion games. In *STOC*, pages 89–98, 2006.
15. R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21(1-2):85–101, 1997.
16. E. C. Jr., J. Galambos, S. Martello, and D. Vigo. Bin packing approximation algorithms: Combinatorial analysis. In D.-Z. Du and P. M. P. (Eds.)., editors, *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, Amsterdam, 1998.
17. E. C. Jr., M. Garey, and D. Johnson. Approximation algorithms for bin-packing: An updated survey. In M. L. G. Ausiello and P. S. (Eds.)., editors, *Algorithm Design for Computer Systems Design*. Springer-Verlag, New York, 1984.
18. R. M. Karp, E. Koutsoupias, C. H. Papadimitriou, and S. Shenker. Optimization problems in congestion control. In *FOCS*, pages 66–74, 2000.
19. E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis. Approximate equilibria and ball fusion. *Theory of Computing Systems*, 36(6):683–693, 2003.
20. E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.
21. C. C. Lee and D. T. Lee. A simple online bin packing algorithm. *J. ACM*, 32:562–572, 1985.
22. T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. A new model for selfish routing. In *STACS*, pages 547–558, 2004.
23. M. Mavronicolas and P. G. Spirakis. The price of selfish routing. In *STOC*, pages 510–519, 2001.
24. J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
25. C. H. Papadimitriou. Algorithms, games, and the internet. In *STOC*, pages 749–753, 2001.
26. T. Roughgarden. Designing networks for selfish users is hard. In *FOCS*, pages 472–481, 2001.
27. M. Tennenholtz and O. Rozenfeld. Strong and correlated strong equilibria in monotone congestion games. In *WINE*, page 7486, 2006.
28. T.Roughgarden and É. Tardos. How bad is selfish routing? In *FOCS*, pages 93–102, 2000.
29. M. Whinston, B. Bernheim, and B. Peleg. Coalition-proof nash equilibria: I concepts. *Journal of Economic Theory*, 42:1–12, 1987.
30. H. Yaïche, R. Mazumdar, and C. Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Trans. Netw.*, 8(5):667–678, 2000.