BOUNDS ON THE CONVERGENCE TIME OF DISTRIBUTED SELFISH BIN PACKING*

FLÁVIO K. MIYAZAWA

Institute of Computing, University of Campinas, Av. Albert Einstein, 1251
Campinas, SP 13084-971–6176, Brazil
fkm@ic.unicamp.br
http://www.ic.unicamp.br/∼fkm

ANDRÉ L. VIGNATTI

Institute of Computing, University of Campinas, Av. Albert Einstein, 1251
Campinas, SP 13084-971–6176, Brazil
vignatti@ic.unicamp.br
http://www.ic.unicamp.br/∼vignatti

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

We consider a game-theoretic *bin packing problem* with identical items, and we study the convergence time to a *Nash equilibrium*. In the model proposed, users choose their strategy simultaneously. We deal with two bins and multiple bins cases. We consider the case when users know the load of all bins and cases with less information. We consider two approaches, depending if the system can undo movements that lead to infeasible states. In the two bins case, we show an $O(\log \log n)$ and an $O(n)$ bounds when undo movements are allowed and when they are not allowed, resp. In multiple bins case, we show an $O(\log n)$ and an $O(nm)$ bounds when undo movements are allowed and when they are not allowed, resp. In the case with less information, we show an $O(m \log n)$ and an $O(n^3m)$ bounds when undo movements are allowed and when they are not allowed, resp. Also, in the case with less information where the information about completely filled/empty bins is not available, we show an $O(m^2 \log n)$ and an $O(n^3m^3)$ bounds when undo movements are allowed and when they are not allowed, resp.

Keywords: Bin packing; Nash equilibrium; convergence time

1991 Mathematics Subject Classification: 22E46, 53C35, 57S20

1. Introduction

The Internet is formed by several entities, where each entity has itself one goal, and those entities are related one to another in many ways. Their relationships are sometimes cooperative, competitive, or even related in a selfish way. Each entity

*This research was partially supported by CNPq.

(also called users, agents or players) has a set of strategies and preferences over these strategies, modeled in an algorithmic way [13].

We are interested in the case where entities are *selfish*. A selfish strategy of a user may influence the decision of other users, making them change their strategies as well. An important question in this situation is if we can reach a state that nobody wants to change their strategy. That is, if we can reach a *Nash equilibrium* [12] in this system. If the answer is yes, then how many steps we need to reach it? And how worse is an equilibrium solution when compared to the optimal solution?

Our focus is on a game theoretic version of the *bin packing* problem when users are *selfish*. More specifically, we are interested in the analysis of the *convergence time* to reach Nash equilibrium. Another important focus in this research direction is the study of the quality of a Nash equilibrium, which is not addressed in this paper, but some results of this kind can be found on [4, 6].

The model we consider is composed of n items, each one with size 1 and controlled by exactly one user, and m bins, each one with capacity C and cost s. At each step, a user that controls an item, selects a bin to migrate. The cost of a bin is equally paid by all users that have an item in the bin. Thus, if all bins have the same cost, a selfish user prefers to pack its item to a bin that is as full as possible.

We consider a strongly distributed and parallel setting, i.e., there is no centralized control mechanism whatsoever, and all users choose to migrate their items at the same time. This contrasts with the Elementary Step System (ESS) [7], where only one user can migrate in each time step. There are some advantages in considering a parallel setting. First, this model is closer to practical situations of large scalable distributed systems, where it may be too expensive, or impossible, to implement a central control responsible for keeping one migration in each step, like the ESS model. Another drawback is that ESS has convergence time bounded by $\Omega(n)$.

In the protocols we present, a user action is based on a probability distribution over the bins. Probabilistic algorithms have some advantages on deterministic ones. First, in a two bins case, as we suppose that all users do not fit in only one bin, a deterministic action needs to deterministically migrate some of the items to the most filled bin, and deterministically leave other items in the less filled bin, being an unfair strategy to those who are selected to stay in the less filled bin. Also, in a multiple bins case, it is unclear how to design a deterministic protocol, for each user, to select an improvement action based on the other users' actions (although, it would be very simple to design such protocol if we assume the existence of a centralized control).

Besides the bin packing is an important problem in computer science, it has several real-world applications, especially in cutting and packing. Also, the problem is used in many areas of computer science, such as multiprocessor scheduling, networks, parallel and distributed systems. Some examples includes data trading in peer-to-peer systems [5], video-on-demand [15], packet scheduling [11], to name only a few. These applications motivate the study of bin packing problem from a game-theoretic and distributed approach.

Related Work: A game theoretic model for bin packing was first proposed by Bilò in [4]. He proved that bin packing game in the ESS model always converges to Nash equilibrium, showing an exponential upper bound in the number of steps. Also, he proved upper and lower bounds on the price of anarchy for that problem. In [6], Epstein and Kleinman obtained better bounds for the price of anarchy, and also lower and upper bounds for the strong price of anarchy [1]. To the present, these two papers are the only ones to address the bin packing problem under the game theoretic perspective. The bin packing problem is also related with the *load balancing* problem. Bilò [4] observed some similarities between these two problems and used a potential function [10, 14] to prove convergence time in a similar way as done for the load balancing problem [7]. The load balancing papers of [9, 3, 8] are most closely related to our work. Goldberg [9] shows a weakly distributed protocol that simulates the ESS. In this protocol, a task choose machines at random, and migrates if the load is lower. He uses a potential function to show upper bounds in the number of steps to reach Nash equilibrium. Even-dar and Mansour [8] consider the case where all users choose to migrate at the same time, in a real concurrent model. In their model, tasks migrate from overloaded to underloaded machines according to some probabilities computed by considering that they know the load information of all machines. Berenbrink et al. [3] propose a strong distributed protocol that needs very little global information, where a task needs to query the load of only one other machine; migrating if that machine has a smaller load. For non-distributed systems, Even-dar et al. [7] studied convergence time to reach a Nash equilibrium of load balancing problems in ESS, and show lower and upper bounds results to many cases.

Our Results: We consider the two bins case and its extension to multiple bins. As we migrate items simultaneously, this can lead to an *infeasible solution* if the number of items that migrates to a bin exceeds its capacity. To deal with this, we propose two approaches.

In the first approach, if a bin has its capacity exceeded in a given time step, then all items that migrated to that bin in this step undo their migration, returning to the previous bins. Undo actions are only performed in bins that had their capacity exceeded. Note, however, that choosing a high migration probability implies a higher number of infeasible movements. On the other hand, choosing a low migration probability implies a higher number of steps to reach the equilibrium. So, clearly we have a trade-off between the chosen probability, the number of infeasible steps and the number of steps to reach a Nash equilibrium. We show that in this approach, within $O(\log\log n)$ steps it is possible to reach a Nash equilibrium with high probability in the two bins case, and $O(\log n)$ steps in the multiple bins case. Also, we obtain an $O(m \log n)$ bound when users have less global information, i.e., a user knows his own bin load and can inspect only one additional bin to obtain the load information. If users have less global information and the information about which bins are completely filled or empty is not available, then we obtain an $O(m^2 \log n)$ bound.

Since not every system allows undoing migrations, we also consider a second approach for which infeasible migrations does not happen, with high probability. It is more likely that real systems embrace this approach, because in most systems an infeasible migration would cause the whole game to become invalid. We show that in this approach, within $O(n)$ steps it is possible to reach a Nash equilibrium with high probability in the two bins case. For multiple bins case, we show an $O(nm)$ bound when users have the load information of all bins and $O(n^3m)$ when users have less global information. If users have less global information and the information about which bins are completely filled or empty is not available, then we obtain an $O(n^3m^3)$ bound.

Organization: Sections 2 and 3 presents the two bins case, with and without undoing infeasible migrations, resp. In Sections 4 and 5, we consider the multiple bins case with and without undoing infeasible migrations, resp. Section 6 presents the case where users have less global information. Section 7 considers the case with even less information, where the information of which bins are completely filled/empty is not available. We comment an extension to bins with different cost in Section 8.

Notation and Model Description

We deal with a model composed of a set of n items $x_1, \ldots, x_n$, each one with size 1 and controlled by a user, and a set of m bins $b_1, \ldots, b_m$, with costs $s_1, \ldots, s_m$ respectively, and capacity C (i.e., all bins have the same capacity). We have a notion of time t, initially equal to 1, denoting the number of steps that had occurred until then. For a given time step t, when an item x_i is assigned to a bin b_k we say that x_i is *in* b_k. The total number of items assigned to bin b_k at step t is denoted by $n_t(b_k)$ and the available space of bin b_k in step t is denoted by $D_t(b_k) = C - n_t(b_k)$. If a bin b has its capacity exceeded after some migrations in a given step, we call them as *infeasible migrations*. An item pays $s_k/n(b_k)$ when it is in bin b_k. We assume that the users who control the items are *selfish*, and therefore they want to minimize how much they pay, without caring for the system as a whole. To minimize the price paid, an item can migrate to another bin in a given step. Thus, in the case of bins with equal costs, a user wants to be in a most filled bin. Let b_{k_i} be the bin to which i is assigned. A state is in α-approximate *Nash equilibrium* if for each user i and bin $b_k \neq b_{k_i}$, we have $s_{b_{k_i}}/n(b_{k_i}) \leq \alpha \cdot s_{b_k}/(n(b_k)+1)$; if $\alpha = 1$ then we simply say that a state is in Nash equilibrium. In this paper, we consider the case where migrations are done *simultaneously* in each step. That is, in a given step, all items choose their strategy (either migrate or stay in the same machine) based on the probabilities defined in the protocols they use. We measure the running time of an algorithm by the number of required steps to reach a Nash equilibrium.

Throughout this paper, we use some technical tools, as stated in Lemmas 1 and

2.

Lemma 1 (Chernoff bounds [2]) *Let* $X_1, \ldots, X_n$ *be binary independent random variables, such that* $\Pr(X_j = 1) = p_j$*. Let* $X = \sum_{j=1}^{n} X_j$ *and* $\mu = E[X]$*. Then*

$$\begin{array}{ll} \Pr[X > (1+\delta)\mu] < e^{-\mu\delta^2/3} & 0 < \delta \le 1; \\ \Pr[X < (1-\delta)\mu] < e^{-\mu\delta^2/2} & 0 < \delta < 1. \\ \Pr\left(\mu \le X + \sqrt{2\ln(\frac{1}{\delta})\mu}\right) \ge 1-\delta & 0 \le \mu \le n; \\ \Pr\left(\mu \ge X - \sqrt{3\ln(\frac{1}{\delta})\mu}\right) \ge 1-\delta & \frac{\ln(1/\delta)}{3} \le \mu \le n. \end{array}$$

The following lemma can be proved using the Stirling's approximation for factorials.

Lemma 2 (probability of hitting the mean) *Let* $X_1, \ldots, X_n$ *be binary independent random variables, such that* $\Pr(X_j = 1) = p$ *and* $X = \sum_{j=1}^{n} X_j$*. If* pn *is an integer, then* $\Pr(X = pn) \ge \frac{1}{\sqrt{2\pi pn}}$*.*

2. Two bins, with Undo of Infeasible Migrations

This section considers the case with two bins of equal costs and it is allowed to undo infeasible migrations (the feasible migrations are maintained). We denote a step with infeasible migrations as an *infeasible step*. Without loss of generality, consider an initial configuration where bin b_1 is most filled than bin b_2. We assume that users know the load information of both bins, so users in b_1 do not want to migrate, and users in b_2 would like to migrate to b_1. We also consider, w.l.o.g., that $n > C$, otherwise Algorithm 1 assigns migration probability equal to 1, and finishes with only one step. As the migrations occur simultaneously, if all users in b_2 decide to migrate to b_1, the capacity of b_1 is exceeded causing a sequence of infeasible steps. Thus, Algorithm 1 defines a *protocol* that all users must follow to reach a Nash equilibrium.

```
begin
    t = 1
    while D_t(b_1) > 0 do
        forall {j ∈ b_2} in parallel do
            move j to b_1 with probability D_t(b_1)/n_t(b_2)
        t = t + 1
end
```

Algorithm 1: TwoBins-UndoInfeasibleMigrations

To simplify the notation, we denote by $D_t = D_t(b_1)$. Given a step t, let X_j be a binary random variable that is equal to 1 if user j in b_2 migrates, and 0 otherwise. Let $X = \sum_j X_j$. Note that $D_{t+1} = D_t - X$ and $E[X] = \sum_j E[X_j] = D_t$. Thus

$E[D_{t+1}] = D_t - E[X] = 0$.

Lemma 3. *In step t, we have* $\Pr\left(0 \le D_{t+1} \le \sqrt{2\ln 4\, D_t}\right) \ge \frac{1}{4}$.

Proof. Since X is a binomial random variable, we have $\Pr(X > E[X]) \le \frac{1}{2}$. Given $\delta > 0$, we have from Lemma 1 that $\Pr\left(X \le E[X] - \sqrt{2\ln(1/\delta)E[X]}\right) \le \delta$. Hence $\Pr\left(E[X] - \sqrt{2\ln(1/\delta)E[X]} \le X \le E[X]\right) \ge \frac{1}{2} - \delta$. As $D_t = E[X]$ and $D_{t+1} = D_t - X$, we conclude that $\Pr\left(E[X] - \sqrt{2\ln(1/\delta)E[X]} \le X \le E[X]\right) = \Pr\left(\sqrt{2\ln(1/\delta)D_t} \ge D_{t+1} \ge 0\right) \ge \frac{1}{2} - \delta$. Using $\delta = 1/4$, we obtain the desired result. □

Lemma 4. *If the current step is t, a Nash equilibrium is reached in $l+1$ additional steps with probability at least* $\left(\frac{1}{4}\right)^l \frac{1}{\sqrt{4\pi \ln 4 \cdot D_t^{(1/2)^l}}}$.

Proof. Applying l times Lemma 3, we have $D_{t+l} \le 2\ln 4 \cdot D_t^{\frac{1}{2^l}}$, with probability at least $\left(\frac{1}{4}\right)^l$. Applying Lemma 2, the result follows. □

Theorem 5. *A Nash equilibrium is reached after $O(\log^2 n \log\log n)$ steps, with high probability.*

Proof. Since $D_0 \le n$, it suffices to apply Lemma 4 for a certain $l = O(\log\log n)$ to have probability at least $\frac{1}{\log n}$ to reach a Nash equilibrium. Repeating this procedure $\log^2 n$ times, the probability that a Nash equilibrium is not obtained is at most $((1 - \frac{1}{\log n})^{\log n})^{\log n} \le \frac{1}{e^{\log n}} = o(1)$. □

Theorem 5 assumes that we can repeatedly restart the game while we have an infeasible step. However, this model not always occurs in a practical situation. In a more practical framework, it is sufficient to cancel only the infeasible steps, and a new step is done from the previous state, as we show in Theorem 6.

Theorem 6. *When only infeasible steps are canceled, a Nash equilibrium is reached in $O(\log\log n)$ steps, with high probability.*

Proof. After $t = \log\log n$ feasible steps, according to Lemma 3, D_t becomes constant. As we will se, after $16t$ steps, we have at least t feasible steps, with high probability. From Lemma 3, the probability to have a feasible step is at least $1/4$. Let X_i be a random variable such that $X_i = 1$ if the i-th step is feasible or $X_i = 0$, otherwise. Let $X = \sum_{i=0}^{16t} X_i$. Thus, $E[X] \ge 4t$. From Lemma 1, we have $\Pr(X \le t) \le \Pr\left(X \le (1 - \frac{3}{4})E[X]\right) \le e^{-\frac{(9/16)4\log\log n}{2}} = o(1)$. When D_t is constant, we have from Lemma 2 that the probability to hit the mean is constant.

Hence after $O(\log \log n)$ steps the probability that a Nash equilibrium is not reached is $o(1)$. $\square$

3. Two Bins, with no Infeasible Steps

This section considers the case with two bins of equal costs where it is not allowed to undo infeasible steps. That is, if an infeasible step occurs, then the game is over without reaching the Nash equilibrium. Also, we introduce the proof scheme used in cases where the undo actions are not allowed. We extend the analysis using the same scheme in Sections 5 and 6, where we consider the cases with multiples bins with more and less global information, respectively.

Like the previous section, we consider that b_1 have more items than b_2 in the initial configuration. Algorithm 2 defines a *protocol* that all users must follow to reach a Nash equilibrium.

```
begin
    t = 1
    while D_t(b_1) > 0 do
        forall {j ∈ b_2} in parallel do
            move j to b_1 with probability:
                • (2/3) D_t(b_1)/n_t(b_2)      if D_t(b_1) ≥ 36 ln n
                • 1/(n_t(b_2)√n)               if 3 ≤ D_t(b_1) < 36 ln n
                • 1/(n_t(b_2)n)                if 1 ≤ D_t(b_1) < 3
        t = t + 1
end
```

Algorithm 2: TwoBins-NoInfeasibleSteps

To simplify the notation, we denote by $D_t = D_t(b_1)$.

Lemma 7. *If $D_t \geq 36 \ln n$, then the probability that b_1 have its capacity exceeded in a step is less than $1/n^2$.*

Proof. Let X_j be a binary random variable such that $X_j = 1$ iff item x_j migrates to bin b_1 in step t and $X = \sum_j X_j$. We have that $E[X] = \frac{2}{3} D_t$. Thus, from Lemma 1, we have $\Pr(X > D_t) = \Pr\left(X > (1 + \frac{1}{2})E[X]\right) < e^{-\frac{(1/2)^2(2/3)36 \ln n}{3}} = e^{-\ln n^2} = 1/n^2$. $\square$

Lemma 8. *If $3 \leq D_t \leq 36 \ln n$, then the probability that b_1 have its capacity exceeded in a step is at most $\frac{2}{n^2}$.*

Proof. The probability that a bin receives more items than its capacity is at most the probability that this bin receives at least 4 items in a single step, which in turn is bounded by $\sum_{i=4}^{n_t(b_2)} \binom{n_t(b_2)}{i} \left(\frac{1}{n_t(b_2)\sqrt{n}}\right)^i \left(1 - \frac{1}{n_t(b_2)\sqrt{n}}\right)^{n_t(b_2)-i} \leq \sum_{i=4}^{n_t(b_2)} n_t(b_2)^i \cdot \frac{1}{n_t(b_2)^i(\sqrt{n})^i} \cdot \left(1 - \frac{1}{n_t(b_2)\sqrt{n}}\right)^{n_t(b_2)-i} \leq \sum_{i=4}^{n_t(b_2)} \frac{1}{(\sqrt{n})^i} \leq \frac{2}{n^2}$. □

Lemma 9. *If $D_t(b) < 3$, then the probability that b_1 have its capacity exceeded in a step is at most $\frac{2}{n^2}$.*

Proof. The probability that a bin receives more items than its capacity is at most the probability that this bin receives at least 2 items in a single step, which in turn is bounded by $\sum_{i=2}^{n_t(b_2)} \binom{n_t(b_2)}{i} \left(\frac{1}{n_t(b_2)n}\right)^i \left(1 - \frac{1}{n_t(b_2)n}\right)^{n_t(b_2)-i} \leq \sum_{i=2}^{n_t(b_2)} \frac{1}{n^i} \leq \frac{2}{n^2}$. □

Lemma 10. *Let X be the total number of items that migrates to a bin b_1 in a step such that $D_t \geq 36 \ln n$. Then $\Pr(X < \frac{1}{3}D_t(b)) < 1/n^3$.*

Proof. Since $D_t \geq 36 \ln n$ we have $E[X] = \frac{2}{3}D_t$. Therefore $\Pr(X < \frac{1}{3}D_t) = \Pr(X < (1 - \frac{1}{2})E[X]) < e^{-\frac{(1/2)^2(2/3)36 \ln n}{2}} \leq e^{-3 \ln n} = 1/n^3$. □

Theorem 11. *After $O(n)$ steps, Algorithm 2 terminates without infeasible steps, with high probability.*

Proof. The analysis is divided in three phases, depending on the values of D_t. In the first phase, we have $D_t \geq 36 \ln n$. Lemma 10 states that $D_{t+1} > \frac{2}{3}D_t$ with probability less than $1/n^3$. We know that $D_0 \leq C$. Thus, after $T = O(\log C)$ steps, we have $D_T > 36 \ln n$ with probability at most $\frac{1}{n^2}$. That is, after $O(\log C)$ steps, this phase ends with high probability. In the second phase, we have $3 \leq D_t < 36 \ln n$. By Lemma 8, a step is infeasible with probability at most $\frac{2}{n^2}$, hence if the algorithm performs at most $\frac{n}{2}$ steps, there is no infeasible step with high probability (at least $1 - 1/n$). In fact, we show that $\sqrt{n} \log n$ steps are sufficient, with high probability. Bin b_1 needs at most $36 \ln n$ items migrating to it before reaching the third phase. In each step, it is expected that $1/\sqrt{n}$ items migrate to it, therefore, the expected number of steps is $\sqrt{n} 36 \ln n$. The probability that no item migrates in a step is given by $(1 - \frac{1}{n_t(b_2)\sqrt{n}})^{n_t(b_2)} \leq e^{-1/\sqrt{n}}$. So, in $\sqrt{n}$ steps, the probability that at least one item migrates is at least $1 - \frac{1}{e} \geq \frac{1}{2}$. As done before, this phase is finished with $\sqrt{n} 144 \log n$ steps with high probability. In the third phase, we have $1 \leq D_t < 3$. By Lemma 9, a step is infeasible with probability at most $\frac{2}{n^2}$, hence the algorithm terminates with high probability (at least $1 - 1/n$) if this phase performs $O(n)$ steps. Applying the same idea used in the analysis of the second phase, after $O(n)$ steps, b_1 will be completely full, with high probability. □

4. Multiple Bins, with Undo of Infeasible Steps

In this section we extend the two bins case presented in Section 2 for the multiple bins case. We also assume that n/C is integer. Algorithm 3 presents a simple protocol that will be executed in parallel for all users. As in Section 2, this section considers the case where it is possible to perform *undo of infeasible migrations*. That is, whenever a bin has its capacity exceeded, the invalid migrations to that bin are canceled and the corresponding items return to their previous bins. In this case, valid migrations in the same step are maintained.

input: bins $b_1, \ldots, b_m$ sorted in non-increasing order according to their loads, items $x_1, \ldots, x_n$
begin
  $t = 1$; $A = \{b_1, \ldots, b_{n/C}\}$; $B = \{b_{n/C+1}, \ldots, b_m\}$; $S = \sum_{i=1}^{n/C} D_t(b_i)$
  while $\{b_i \in B : n(b_i) > 0\} \neq \emptyset$ **do**
    forall $\{j \in b_i : b_i \in B\}$ **in parallel do**
      move j to $b_l \in A$ with probability $\frac{D_t(b_l)}{S}$
    Update S; $t = t + 1$
end

Algorithm 3: MultBins-UndoInfeasibleMigrations

We show that Algorithm 3 reaches a Nash equilibrium with high probability in few steps. It is also desirable that the probability imposed to users in each step leads to a Nash equilibrium. In this case, users would agree with the probabilities attributed to them in each step. However, the probabilities imposed by Algorithm 3 does not characterize a strategy in Nash equilibrium, as we explain in the following.

Let s be the cost of each bin. Thus, a user using bin b_i has to pay $s/n_t(b_i)$. In a given step t, a user evaluates the expected load of each bin in step $t+1$ *without considering its own action*, and then choose its strategy. The expectation $E[n_{t+1}(b_i)]$ without considering the action of user x_j is given by

$$E[n_{t+1}(b_i)] = n_t(b_i) + \tfrac{D_t(b_i)}{S}(S-1) = C - \tfrac{(C-n_t(b_i))}{S}. \tag{1}$$

That is, the largest expectation (and lower cost for a user) is obtained by the most filled bins. Therefore, if a selfish user can choose its own migration probability, it will use a best response strategy with probability 1 to migrate to the most filled bin. This behavior will lead to invalid migrations for all users.

Although Algorithm 3 does not use the best response strategy (i.e., users do not necessarily migrate to the most filled bin), it is justified by the fact that it is an improvement response strategy. That is, in each step items migrate to more filled bins, diminishing the value paid by the users. Moreover, we prove that the strategy above is a 2-approximate Nash equilibrium and it reaches a Nash equilibrium in few steps, with high probability.

Theorem 12. *In each step, Algorithm 3 is a* 2*-approximation Nash equilibrium strategy.*

Proof. Consider a user j in a bin of set B in step t and a bin $b_i \in A$. From Eq. 1, the expectation $E[n_{t+1}(b_i)]$ without considering the action of user j is given by

$E[n_{t+1}(b_i)] = C - \frac{(C-n_t(b_i))}{S} \le C$. We also have that $E[n_{t+1}(b_i)] = C - \frac{(C-n_t(b_i))}{S} \ge C - \frac{C}{S} \ge C/2$. The last inequality is valid since $S \ge 2$. □

In each step of Algorithm 3 it is expected that the bins in A become completely filled and the bins in B completely empty. The following theorem presents an upper bound in the number of steps to reach a Nash equilibrium.

Theorem 13. *If in each step infeasible migrations can be cancelled then Algorithm 3 reaches a Nash equilibrium in $O(\log n)$ steps, with high probability.*

Proof. Let $l = O(\log \log n)$ be the number of steps necessary to find a Nash equilibrium for the case with two bins with high probability, in Theorem 6. We say that a *round* is a sequence of l steps. Each bin can be viewed in a independent way, hence, after one round (see proof of Theorem 6) the probability that a given bin does not become completely full is at most $\frac{1}{\log n}$. Thus, after $\frac{2 \log n}{\log \log n}$ rounds, a given bin is not completely full with probability $1/n^2$. Using the union bound, the probability that some bin is not completely full in $\frac{2 \log n}{\log \log n} \cdot l = O(\log n)$ steps is at most $1/n$. □

5. Multiple Bins, with no Infeasible Steps

In this section, we are interested in the case of multiple bins *avoiding infeasible steps* (an infeasible step finishes the game without a solution). Since no step can exceeds the bin capacity, we use more "conservatives" migration probabilities in such a way that a bin does not have its capacity exceeded, with high probability.

In this section, we assume that n/C is an integer. Algorithm 4 presents a simple protocol where each step is executed in parallel for all players.

```
input: bins b_1, ..., b_m sorted in non-increasing order according to their
       loads, items x_1, ..., x_n
begin
  t = 1; A = {b_1, ..., b_{n/C}}; B = {b_{n/C+1}, ..., b_m}; S = Σ_{i=1}^{n/C} D_t(b_i)
  while {b_i ∈ B : n(b_i) > 0} ≠ ∅ do
    forall {j ∈ b_i : b_i ∈ B} in parallel do
      move j to b_l ∈ A with probability:
        • (2/3) D_t(b_l)/S      if D_t(b_l) ≥ 54 ln n
        • 1/(S√(nm))            if 3 ≤ D_t(b_l) < 54 ln n
        • 1/(Snm)               if 1 ≤ D_t(b_l) < 3
    Update S; t = t + 1
end
```

Algorithm 4: MultBins-NoInfeasibleSteps

Throughout this section, we use A and B as defined by Algorithm 4.

Lemma 14. *For all bins b such that $D_t(b) \geq 54 \ln n$, the probability that one of these bins have its capacity exceeded in a step is less than $1/n^2$.*

Proof. Let b be a bin such that $D_t(b) \geq 54 \ln n$. Let X_j be a binary random variable such that $X_j = 1$ iff item x_j migrates to bin b in step t and $X = \sum_j X_j$. For simplicity, we denote by D_t the value $D_t(b)$. We have that $E[X] = \frac{2}{3}D_t$. Thus, from Lemma 1, we have $\Pr(X > D_t) = \Pr\left(X > (1+\frac{1}{2})E[X]\right) < e^{-\frac{(1/2)^2(2/3)54 \ln n}{3}} = e^{-\ln n^3} = 1/n^3$. The proof follows by the union bound. □

Lemma 15. *For all bins b such that $3 \leq D_t(b) \leq 54 \ln n$, the probability that one of these bins have its capacity exceeded in a step is at most $\frac{2}{n^2 m}$.*

Proof. The probability that a bin receives more items than its capacity is at most the probability that this bin receives at least 4 items in a single step, which in turn is bounded by $\sum_{i=4}^{S} \binom{S}{i} \left(\frac{1}{S\sqrt{nm}}\right)^i \left(1 - \frac{1}{S\sqrt{nm}}\right)^{S-i} \leq \sum_{i=4}^{S} S^i \cdot \frac{1}{S^i(\sqrt{nm})^i} \cdot \left(1 - \frac{1}{S\sqrt{nm}}\right)^{S-i} \leq \sum_{i=4}^{S} \frac{1}{(\sqrt{nm})^i} \leq \frac{2}{n^2 m^2}$. The proof follows by the union bound. □

Lemma 16. *For all bins b such that $D_t(b) < 3$, the probability that one of these bins have its capacity exceeded in a step is at most $\frac{2}{n^2 m}$.*

Proof. The probability that a bin receives more items than its capacity is at most the probability that this bin receives at least 2 items in a single step, which in turn is bounded by $\sum_{i=2}^{S} \binom{S}{i} \left(\frac{1}{Snm}\right)^i \left(1 - \frac{1}{Snm}\right)^{S-i} \leq \sum_{i=2}^{S} \frac{1}{(nm)^i} \leq \frac{2}{n^2 m^2}$. The proof follows by the union bound. □

Lemma 17. *If X is the total number of items that migrates to a bin $b \in A$ in a given step t and $D_t(b) \geq 54 \ln n$ then $\Pr(X < \frac{1}{3}D_t(b)) < 1/n^4$.*

Proof. Let X_j be a binary random variable such that $X_j = 1$ iff item x_j migrates to bin b in step t and let $X = \sum_j X_j$ be the total number of items that migrates to b in step t. For simplicity, we denote by D_t the value $D_t(b)$. Since $D_t \geq 54 \ln n$ we have $E[X] = \frac{2}{3}D_t$. Therefore $\Pr(X < \frac{1}{3}D_t) = \Pr(X < (1 - \frac{1}{2})E[X]) < e^{-\frac{(1/2)^2(2/3)54 \ln n}{2}} \leq e^{-4 \ln n} = 1/n^4$. □

Lemma 18. *After $T = O(\log C)$ steps, some bin b have $D_T(b) > 54 \ln n$ with probability at most $\frac{1}{n^2}$.*

Proof. In this proof, we only refer to bins $b \in A$ such that $D_t(b) > 54 \ln n$, the other bins do not need to be taken into consideration. Let t be the current step. For a bin b, Lemma 17 states that $D_{t+1}(b) > \frac{2}{3}D_t(b)$ with probability less than $1/n^4$. Applying the union bound, after a step, some bin $b \in A$ will have $D_{t+1}(b) > \frac{2}{3}D_t(b)$ with probability less than $\frac{1}{n^3}$. We know that $D_0(b) \leq C$. Thus, after $O(\log C)$ steps, the result follows. □

Theorem 19. *After $O(nm)$ steps, Algorithm 4 terminates without infeasible steps, with high probability.*

Proof. The analysis is divided in three phases, depending on the values of D_t. In the first phase, we have $D_t \geq 54 \ln n$. By Lemma 18, after $O(\log C)$ steps, this phase ends with high probability. In the second phase, we have $3 \leq D_t < 54 \ln n$. By Lemma 15, a step is infeasible with probability at most $\frac{2}{n^2 m}$, hence if the algorithm performs at most $\frac{nm}{2}$ steps, there is no infeasible step with high probability (at least $1 - 1/n$). In fact, we show that we need at most $\sqrt{nm} \log n$ steps as follows. A bin needs at most $54 \ln n$ items migrating to it before reaching the third phase. In each step, it is expected that $1/\sqrt{nm}$ items migrate to it, therefore, the expected number of steps is $\sqrt{nm} 54 \ln n$. The probability that no item migrates to bin i in a step is given by $(1 - \frac{1}{n'_t \sqrt{nm}})^{n'_t} \leq e^{-1/\sqrt{nm}}$, where $n'_t = n - n_t(b_i)$. So, in $\sqrt{nm}$ steps, the probability that at least one item migrates is at least $1 - \frac{1}{e} \geq \frac{1}{2}$. Therefore, this phase is finished with $\sqrt{nm} 216 \log n$ steps with high probability. In the third phase, we have $1 \leq D_t(b) < 3$. By Lemma 16, a step is infeasible with probability at most $\frac{2}{n^2 m}$, hence the algorithm terminates with high probability (at least $1 - 1/n$) if this phase performs at most $O(nm)$ steps. Applying the same idea used in the analysis of the second phase, after $O(nm)$ steps, all bins will be completely full, with high probability. □

6. Multiples Bins and Less Global Information

In Sections 4 and 5, we assume that users know, in each step, the load information of every bin. This can be a strong assumption if the interval between each step is constant, because knowing the load information of every bin takes $O(m)$ time. In this section, we consider that an item knows his own bin load and can only inspect the load of one additional bin that is not completely full or empty, incurring in a constant time step of the protocol. Once a bin becomes completely full or empty, it is not considered anymore.

Algorithm 5 defines a protocol that users follow to reach a Nash equilibrium when the system can undo infeasible migrations, as in Sections 2 and 4. Later, we present Algorithm 6 for the case where undoing infeasible migrations is not possible.

In Algorithm 5, bins have *labels*, each label is a number in $\{1, \ldots, m\}$ and no two bins have the same label. Let $\ell(b_i)$ be the label of bin b_i. Also, if a bin becomes completely full or empty then it will not be considered by players in subsequent steps. Thus, we define m_t as the number of bins not completely full or empty at time t.

Notice that, in the algorithm, the random choice of bin is done considering only bins different from b_{x_i} and bins not completely full or empty. That is, we choose each bin with probability $1/(m_t - 1)$, never choosing the bin in which the item is assigned or a bin already completely filled or empty.

The algorithm does not incur in a high number of infeasible steps, as we explain

```
input: bins b_1, ..., b_m, items x_1, ..., x_n
begin
    t = 1;
    foreach item x_i in parallel do
        let b_{x_i} be the current bin of item x_i
        choose bin b_j ≠ b_{x_i} uniformly at random
        if (n_t(b_j) > n_t(b_{x_i})) or (n_t(b_j) = n_t(b_{x_i}) and ℓ(b_j) < ℓ(b_{x_i})) then
            move x_i to b_j with probability min( D_t(b_j) / n_t(b_{x_i}), 1 )
        t = t + 1
end
```

Algorithm 5: MultBins-LessInformation

next.

Lemma 20. *After one step, the probability that a bin j receives more items than its capacity is at most* $3/4$.

Proof. Let j be a bin. We compute the expected number of items that migrate to j in one step. Let j' be a bin with less items or equal number of items but greater label than j. If $\frac{D_t(j)}{n_t(j')} \leq 1$ then j' sends an expected number of $\frac{D_t(j)}{m_t-1}$ items to j. Otherwise ($n_t(j) < D_t(j)$), j' sends an expected number of $\frac{n_t(j)}{m_t-1} < \frac{D_t(j)}{m_t-1}$ items to j. Thus, the most filled bin with smallest label receives more items, receiving an expected number of at most $D_t(j)$ items. So, analyzing in the same way as in Section 2, and by Lemma 3, it has probability at most $3/4$ of receiving more items than its capacity. Since the other bins receive less items than the most filled bin, the result follows. □

Lemma 21. *After $O(\log n)$ steps, at least one bin becomes filled or empty, with high probability.*

Proof. Let $b^*(t)$ be the most filled bin (that is not completely full) with the smallest label in time t and $b^o(t)$ the less filled bin (not empty) with the greatest label in time t. We have two cases. (i) If $\frac{D_t(b^*(t))}{n_t(b^o(t))} \geq 1$, then $\frac{D_t(b)}{n_t(b^o(t))} \geq 1$ for any bin b. Therefore $b^o(t)$ do not receive items, and try to migrate with probability 1 all its items. By Lemma 20, it is expected that at least $1/4$ of the items in $b^o(t)$ successfully migrates. Note that $b^o(t+1)$ may be different to $b^o(t)$, but this is not a problem since if $b^o(t+1) \neq b^o(t)$ implies that $b^o(t+1)$ is less filled than $b^o(t)$ after items migrate. Therefore, after $O(\log n)$ steps, $b^o(t)$ becomes empty, with high probability. On the other hand, (ii) if $\frac{D_t(b^*(t))}{n_t(b^o(t))} < 1$ then $\frac{D_t(b^*(t))}{n_t(b)} < 1$ for any bin b. Thus, each bin is expected to send $\frac{D_t(b^*(t))}{m_t-1}$ items to $b^*(t)$. As $m_t - 1$ bins send this amount to b^*, it is expected that j^* receives $D_t(b^*(t))$ items. Following the analysis idea of Section 2, in $O(\ln \ln n)$ steps $b^*(t)$ becomes filled. Again, $b^*(t+1)$ may be different to $b^*(t)$, but as explained above, this is not a problem. Note that, as time progresses, case (i) may turn to case (ii) and vice-versa, but this fact does not affect the analysis, and the result follows. □

The following theorem follows directly from Lemma 21.

Theorem 22. *After $O(m \log n)$ steps, Algorithm 5 terminates with high probability.*

Algorithm 6 is designed for the case where users have little global information and cannot undo infeasible migrations. As before, bins have *labels* in $\{1, \dots, m\}$ and no two bins have the same label. Let $\ell(b_i)$ be the label of bin b_i. Also, if a bin becomes completely full or empty then it will not be considered by players in subsequent steps. Thus, we define m_t as the number of bins not completely full or empty in time t.

input: bins $b_1, \dots, b_m$, items $x_1, \dots, x_n$
begin
  $t = 1$;
  foreach *item* x_i **in parallel do**
    let b_{x_i} be the current bin of item x_i
    choose bin $b_j \neq b_{x_i}$ uniformly at random
    if $\Big(n_t(b_j) > n_t(b_{x_i})\Big)$ *or* $\Big(n_t(b_j) = n_t(b_{x_i})$ *and* $\ell(b_j) < \ell(b_{x_i})\Big)$ **then**
      move x_i to b_j with probability

- $\min\left(\frac{2}{3}\frac{D_t(b_j)}{n_t(b_{x_i})}, 1\right)$ if $D_t(b_j) \geq 126 \ln n$
- $\frac{1}{n^2 n_t(b_{x_i})}$ if $3 \leq D_t(b_j) < 126 \ln n$
- $\frac{1}{n^3 n_t(b_{x_i})}$ if $1 \leq D_t(b_j) < 3$

    $t = t + 1$
end

Algorithm 6: MultBins-LessInformation-NoUndo

In what follows, we denote by b^* the most filled bin (not completely full) with the smallest label.

Lemma 23. *For all bins b such that $D_t(b) \geq 126 \ln n$, the probability that one of these bins has its capacity exceeded in a step is less than $1/n^6$.*

Proof. Let X_j be a binary random variable such that $X_j = 1$ iff item x_j migrates to bin b^* in step t and $X = \sum_j X_j$. For simplicity, we denote by D_t the value $D_t(b^*)$. Note that $E[X] \leq \frac{2}{3}D_t$, however the highest probability of exceeding the bin capacity is when $E[X] = 2/3D_t$. Thus, from Lemma 1, we have $\Pr(X > D_t) \leq \Pr\left(X > (1+\frac{1}{2})E[X]\right) < e^{-\frac{(1/2)^2(2/3)126 \ln n}{3}} = e^{-\ln n^7} = 1/n^7$. All other bins receive less items than b^*, thus they can be bounded this way. The proof follows by the union bound. □

Lemma 24. *For all bins b such that $3 \leq D_t(b) \leq 126 \ln n$, the probability that one of these bins has its capacity exceeded in a step is at most $\frac{2}{n^4(m_t-1)^3}$.*

Proof. Let $n_t(b^o)$ be the load of the less filled bin (not empty). The probability that b^* receives more items than its capacity is at most the probability that b^* receives at least 4 items in a single step, which in turn is bounded by $\sum_{i=4}^{n-n_t(b^*)} \binom{n-n_t(b^*)}{i} \left(\frac{1}{n^2(m_t-1)n_t(b^o)}\right)^i \cdot \left(1 - \frac{1}{n^2(m_t-1)n_t(b^o)}\right)^{n-n_t(b^*)-i} \leq \sum_{i=4}^{\infty} \binom{n}{i} \left(\frac{1}{n^2(m_t-1)}\right)^i \leq \sum_{i=4}^{\infty} n^i \frac{1}{n^{2i}(m_t-1)^i} = \sum_{i=4}^{\infty} \frac{1}{n^i(m_t-1)^i} \leq \frac{2}{n^4(m_t-1)^4}$. All other bins receive less items than b^*, thus they can be bounded this way. By the union bound (the less filled bin with greatest label never have its capacity exceeded), the result follows. $\square$

Lemma 25. *For all bins b such that $D_t(b) < 3$, the probability that one of these bins has its capacity exceeded in a step is at most $\frac{2}{n^4(m_t-1)}$.*

Proof. The probability that b^* receives more items than its capacity is at most the probability that b^* receives at least 2 items in a single step, which in turn is bounded by $\sum_{i=2}^{n-n_t(b^*)} \binom{n-n_t(b^*)}{i} \left(\frac{1}{n^3(m_t-1)n_t(b_i)}\right)^i \cdot \left(1 - \frac{1}{n^3(m_t-1)n_t(b_i)}\right)^{n-n_t(b^*)-i} \leq \sum_{i=2}^{\infty} \binom{n}{i} \left(\frac{1}{n^3(m_t-1)}\right)^i \leq \sum_{i=2}^{\infty} n^i \frac{1}{n^{3i}(m_t-1)^i} = \sum_{i=2}^{\infty} \frac{1}{n^{2i}(m_t-1)^i} \leq \frac{2}{n^4(m_t-1)^2}$. All other bins receive less items than b^*, thus they can be bounded this way. By the union bound (the less filled bin with greatest label never have its capacity exceeded), the result follows. $\square$

Corollary 26 follows from Lemmas 23, 24 and 25.

Corollary 26. *In Algorithm 6, an infeasible step occurs with probability at most $\frac{2}{n^4(m_t-1)}$.*

Theorem 27. *After $O(n^3m)$ steps, Algorithm 6 terminates with high probability.*

Proof. We show that, after $O(n^3)$ steps, at least one bin becomes completely full. This fact, together with Corollary 26, is sufficient to prove the result, because there are m bins. We consider that a bin b goes through 3 phases until it becomes completely full. In the first, second and third phases we have, resp., $D_t(b) \geq 126 \ln n$, $3 \leq D_t(b) < 126 \ln n$ and $D_t(b) < 3$. We focus the analysis at bin b^*. In the second and third phases, it is expected that b^* receives, resp., $1/n^2$ and $1/n^3$ items in each step. Therefore, we need $O(n^2 \log n)$ and $O(n^3)$ expected steps to terminate, resp., second and third phases. It is possible to show that these number of steps is sufficient to terminate the both second and third phases with high probability (see proof of Theorem 19). Let b^o be the less filled bin (not empty) with greatest label. In the first phase, we consider two cases: (i) if $\frac{2D_t(b^*)}{3n(b^o)} > 1$ and (ii) otherwise. In case (i), we have $\frac{2D_t(b)}{3n(b^o)} > 1$ for each bin b that is not completely full or empty.

Therefore, b^o gets empty with high probability in a single step, because as seen in Corollary 26, we have low probability of error. As there are m bins, case (i) occurs at most m times. In case (ii), we have $\frac{2D_t(b^*)}{3n(b)} \leq 1$ for each valid bin b. Thus, b^* receives expected number of $\frac{2}{3}D_t(b^*)$ items, and it is possible to show that after $O(\log n)$ steps in case (ii) the first phase terminates with high probability for bin b^* (see proof of Lemma 18). As noted in Lemma 21, case (i) may lead to case (ii) and vice-versa. Therefore, the first phase terminates in $O(m + \log n)$ steps. Adding the number of steps needed in each phase, b^* becomes completely full in $O(n^3)$ steps, with high probability. □

7. Even Less Global Information

In Section 6, the random choice of bin is done without considering completely full or empty bins. Thus, we need a certain degree of global information regarding the bins that become full or empty. This section discuss the case where such information is not available. In this case, the random choice is done considering all bins. Thus, the expected number of items that migrates to the "correct" bins are smaller than the case considered before, which leads to greater bounds to reach the equilibrium. In the rest of this section, we point some modifications in the proofs or protocols of the Section 6 to deal with this lack of information.

In Algorithm 5, we just need to point some modifications in the proofs. As the expected number of items that migrates to another bin is smaller than the case of Section 6, the probability that a bin have its capacity exceeded is lower. Therefore, Lemma 20 remains valid. The proof of Lemma 21 is divided in two cases. In case (i), it is expected that at least $\frac{1}{4m}$ of the items in $b^o(t)$ successfully migrates, instead of 1/4. Thus, after $O(m \log n)$ steps, $b^o(t)$ becomes empty, with high probability. In case (ii), it is expected that j^* receives $D_t(b^*(t))/m$ items. Thus, after $O(m \log n)$ steps $b^*(t)$ becomes filled. Therefore, after $O(m \log n)$ steps, at least one bin becomes filled or empty, with high probability. As there are m bins, we can bound the number of steps needed to reach the equilibrium, as shown in Theorem 28.

Theorem 28. *If all bins are considered for migration, then after $O(m^2 \log n)$ steps, Algorithm 5 terminates with high probability.*

In Algorithm 6, we modify the probability of the case where $1 \leq D_t(b_j) < 3$ from $\frac{1}{n^3 n_t(b_{x_i})}$ to $\frac{1}{n^3 m n_t(b_{x_i})}$. Thus, the proof of Lemma 25 is modified, stating that the probability that one of the bins has its capacity exceeded in a step is at most $\frac{2}{n^4(m-1)^3}$. We can now rewrite Corollary 26 as follows.

Corollary 29. *In the modified version of Algorithm 6, an infeasible step occurs with probability at most $\frac{2}{n^4(m-1)^3}$.*

Theorem 30 presents the result using the modification proposed.

Theorem 30. *If all bins are considered for migration, then after $O(n^3 m^3)$ steps, the modified version of Algorithm 6 terminates with high probability.*

Proof. This proof is a modification of the proof of Theorem 27. We show that, after $O(n^3m^2)$ steps, at least one bin becomes totally full. This fact, together with Corollary 26, is sufficient to prove the result, because there are m bins. We consider that a bin b goes through 3 phases until it becomes totally full. In the first, second and third phases we have, resp., $D_t(b) \geq 126 \ln n$, $3 \leq D_t(b) < 126 \ln n$ and $D_t(b) < 3$. We focus the analysis at bin b^*. In the second and third phases, it is expected that b^* receives, resp., $\frac{1}{n^2m}$ and $\frac{1}{n^3m^2}$ items in each step. Therefore, we need $O(n^2m \log n)$ and $O(n^3m^2)$ expected steps to terminate, resp., second and third phases. In the first phase, we consider two cases: (i) if $\frac{2D_t(b^*)}{3n(b^o)} > 1$ and (ii) otherwise. In case (i), we have $\frac{2D_t(b)}{3n(b^o)} > 1$ for each bin b that is not totally full or empty. Thus, at least a fraction of $1/m$ of the load of b^o migrates to a valid bin, and after $O(m \log n)$ steps b^o gets empty with high probability. In case (ii), we have $\frac{2D_t(b^*)}{3n(b)} \leq 1$ for each valid bin b. Thus, b^* receives expected number of $\frac{2}{3m}D_t(b^*)$ items, and it is possible to show that after $O(m \log n)$ steps in case (ii) the first phase terminates with high probability for bin b^* (see proof of Lemma 18). Therefore, the first phase terminates in $O(m \log n)$ steps. Adding the number of steps needed in each phase, b^* becomes totally full in $O(n^3m^2)$ steps, with high probability. □

8. Extension to Different Costs

Our protocols also work for bins with different costs. Let $L(b_i) = \frac{s_i}{n_t(b_i)}$. In the two bins case, let $b_{\min} = \operatorname{argmin}(L(b_1), L(b_2))$ and $b_{\max} = \operatorname{argmax}(L(b_1), L(b_2))$. Note that items in $b_{\max}$ want to migrate to $b_{\min}$. It is easy to see that the same protocol presented in Section 2 works for bins with different costs simply doing $b_{\min}$ as the most filled bin and $b_{\max}$ the less filled bin. For multiple bins case, we use the same idea. Let $b_1, \ldots, b_m$ be the bins sorted in non-decreasing order according to $L(b_i)$. Thus, bins $b_1, \ldots, b_{n/C}$ are used on protocols of Sections 4 and 5 as set A, and the other bins are used as set B. All results remain valid except Theorem 12, which is not valid when bins have different costs. In algorithms of Section 6, we compare using $L(b_i)$ instead $n(b_i)$.

9. Closing Remarks

In this paper, we presented protocols for a bin packing game when migration is done simultaneously, motivated by parallel and distributed systems. The simplicity and efficiency of these protocols make them very attractive. Without following protocols like the ones presented in this paper, users know that their selfish strategies will lead to infeasible steps and invalidate attempts of the system to reach Nash equilibrium. Some questions that remain open in our model are related to lower bounds in the number of steps and other protocols that requires even less global information.

References

[1] Nir Andelman, Michal Feldman, and Yishay Mansour. Strong price of anarchy. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete*

algorithms, pages 189–198, 2007.

[2] Dana Angluin and Leslie Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155–193, 1979.

[3] Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul Goldberg, Zengjian Hu, and Russell Martin. Distributed selfish load balancing. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 354–363, 2006.

[4] Vittorio Bilò. On the packing of selfish items. In *20th Internacional Parallel and Distributed Processing Symposium - IPDPS*, pages 9–18. IEEE, 2006.

[5] Brian F. Cooper and Hector Garcia-Molina. Peer-to-peer data trading to preserve information. *ACM Trans. Inf. Syst.*, 20(2):133–170, 2002.

[6] Leah Epstein and Elena Kleiman. Selfish bin packing. In *ESA '08: Proceedings of the sixteenth annual European Symposium on Algorithms*, 2008.

[7] Eyal Even-Dar, Alex Kesselman, and Yishay Mansour. Convergence time to nash equilibrium in load balancing. *ACM Transactions on Algorithms*, 3(3):Article 32, 2007.

[8] Eyal Even-Dar and Yishay Mansour. Fast convergence of selfish rerouting. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 772–781, 2005.

[9] Paul W. Goldberg. Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 131–140, 2004.

[10] Dov Monderer and Lloyd Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[11] Nir Naaman and Raphael Rom. Packet scheduling with fragmentation. In *INFOCOM '02: Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 427–436, 2002.

[12] John Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, September 1951.

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

[14] Robert Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[15] E. C. Xavier and F. K. Miyazawa. The class constrained bin packing problem with applications to video-on-demand. *Theor. Comput. Sci.*, 393(1-3):240–259, 2008.