Improving the Efficiency of Load Balancing Games through Taxes*

Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos

Research Academic Computer Technology Institute and
Dept. of Computer Engineering and Informatics
University of Patras, 26500 Rio, Greece

Abstract. In load balancing games, there is a set of available servers and a set of clients; each client wishes to run her job on some server. Clients are selfish and each of them selects a server that, given an assignment of the other clients to servers, minimizes the latency she experiences with no regard to the global optimum. In order to mitigate the effect of selfishness on the efficiency, we assign taxes to the servers. In this way, we obtain a new game where each client aims to minimize the sum of the latency she experiences and the tax she pays. Our objective is to find taxes so that the worst equilibrium of the new game is as efficient as possible. We present new results concerning the impact of taxes on the efficiency of equilibria, with respect to the total latency of all clients and the maximum latency (makespan).

1 Introduction

Load balancing games are special cases of the well-known *congestion games* introduced by Rosenthal [18]. A congestion game Π consists of a set E of resources, each resource e having a non-negative and non-decreasing latency function f_e defined over non-negative numbers, and a set of n players. Each player i has a weight (or demand) w_i and can select among a set of permissible strategies $S_i \subseteq 2^E$ (where each strategy of player i is a set of resources). In general, players may follow *mixed strategies*, i.e., use a probability distribution over their permissible strategies. An assignment $A = (A_1, ..., A_n)$ is a vector of strategies, one (possibly mixed) strategy for each player. We mostly refer to pure assignments, i.e., assignments where each player selects a single strategy with probability 1. The cost of a player i at an assignment A is defined as $cost_i(A) = \sum_{e \in A_i} f_e(n_e(A))$, where $n_e(A)$ is the total weight of players using resource e in A. The *social cost* of an assignment can be either the weighted total cost over all players or the maximum latency (makespan) over all resources. A pure (resp., mixed) assignment is a *pure* (resp., *mixed*) *Nash equilibrium* if no player has any incentive to unilaterally deviate to another strategy,

* This work was partially supported by the European Union under IST FET Integrated Project 015964 AEOLUS and by a "Caratheodory" grant from the University of Patras.

C. Papadimitriou and S. Zhang (Eds.): WINE 2008, LNCS 5385, pp. 374–385, 2008.

i.e., $cost_i(A) \leq cost_i(A_{-i}, s)$ for any player i and for any pure (resp., mixed) strategy s, where (A_{-i}, s) is the assignment obtained if just player i deviates from A_i to s. In *linear congestion games*, the latency function of resource e is of the form $f_e(x) = \alpha_e x + \beta_e$ with non-negative constants α_e and β_e. We use the terms *weighted* and *unweighted* to distinguish between the cases where the clients have different or identical weights.

Load balancing games are congestion games where the strategies of players are singleton sets. In load balancing terminology, we use the terms *server* and *client* instead of the terms resource and player. The set of strategies of a client contains the servers that are permissible for the client. A load balancing game is called *symmetric* when all servers are permissible for any client. Usually, the servers of load balancing games have linear latency functions; an important special case is that of *related* servers where the latency function of server j is of the form $f_j(x) = \alpha_j x$, with $\alpha_j > 0$. Motivated by [8], we use the term *graph balancing games* to denote asymmetric load balancing games where each client is unweighted and has at most two permissible servers, and all servers have identical linear latency functions.

Since players act selfishly, load balancing games may reach assignments that do not minimize the social cost. We use the notion of the *price of anarchy* introduced in [15,17] to quantify the degradation of the overall system performance. In particular, the price of anarchy of a game Π is the maximum over all pure (or mixed) Nash equilibria of the ratio of the social cost of a pure (or mixed) Nash equilibrium over the social cost of the optimal assignment.

A vast amount of the literature (see [20,25] and the references therein) studies the complexity of computing equilibria of best and worst social cost and provides bounds on the price of anarchy for various games that can be thought of as special cases of congestion games such as load balancing games, when the social cost is defined as the makespan or the weighed total latency. Awerbuch et al. [1] and, independently, Christodoulou and Koutsoupias [4] prove tight bounds on the price of anarchy of congestion games with respect to the weighted total latency. Among other results concerning polynomial latency functions, they show that the price of anarchy of pure Nash equilibria in unweighted linear congestion games is $5/2$ while for mixed Nash equilibria or pure Nash equilibria of weighted players it is 2.618. These bounds carry over to load balancing games [2] and can be improved for interesting special cases [2,16,22]. The price of anarchy of weighted load balancing games on m related servers is $\Theta(\frac{\log m}{\log \log \log m})$ [7] over mixed Nash equilibria with respect to the makespan. A better tight bound of $\Theta(\frac{\log m}{\log \log m})$ is known for pure Nash equilibria as well as for mixed Nash equilibria at identical servers [7,14].

In order to downscale the effect of selfishness to performance, we assign taxes to the servers. Formally, a *tax function* $\delta : E \times Q^+ \to Q^+$ assigns a tax $\delta_j(w)$ to each client of weight w that wishes to use server $j \in E$. Furthermore, we assume that clients are not equally sensitive to taxes. In particular, client i has a *tax sensitivity* $\gamma_i > 0$. Assuming selfish behavior of the clients, we obtain a new *extended game* (Π, δ) where each client now aims to minimize the expected

latency she experiences plus her disutility due to the taxes she pays at the server she uses. This disutility equals $\gamma_i \delta_j(w_i)$ when client i selects server j. Again, an assignment y is a pure Nash equilibrium for the extended game if no player has an incentive to unilaterally change her strategy, i.e., $cost_i(y) + \gamma_i \delta_s(w_i) \leq cost_i(y_{-i}, s^*) + \gamma_i \delta_{s^*}(w_i)$ for any client i that is on server s under the assignment y, where y_{-i}, s^* is the assignment produced when client i moves from s to s^*.

Like in our previous work [3] on the topic and motivated by [6], we consider both *refundable* and *non-refundable* taxes. In the former case, we assume that the collected taxes can be feasibly returned (directly or indirectly) to the players (e.g., as a "lump-sum refund") and therefore the overall system disutility depends only on the social cost. However, refunding the collected taxes could be logistically or economically infeasible; the latter case models this scenario. We will say that a function $\delta : E \times Q^+ \rightarrow Q^+$ is a ρ-pure-efficient refundable tax for the load balancing game Π if the social cost for any pure Nash equilibrium of the extended game is at most ρ times the social cost of the optimal assignment. Similarly, a function $\delta : E \times Q^+ \rightarrow Q^+$ is a ρ-pure-efficient non-refundable tax for the load balancing game Π if the social cost plus the total disutility due to taxes at any pure Nash equilibrium is at most ρ times the social cost of the optimal assignment. Similar definitions apply to the case of mixed Nash equilibria.

The problem of computing optimal taxes has received significant attention in the economics and transportation science literature; the main underlying model in these studies is that of *non-atomic* congestion games [21]. These games differ from the atomic games that we consider in that each player controls an infinitesimal amount of demand, and, therefore, the actions of a single player cannot affect the overall system performance. The results about taxes in non-atomic congestion games (see for example [5,6,10,13]) do not carry over to the atomic model. In our previous work [3], we presented (among several negative and positive results on the influence of taxes, under the assumption that all clients are equally sensitive to taxes) 2-mixed-efficient refundable taxes with respect to the weighted total latency for linear atomic congestion games and a pure-optimal tax function for symmetric load balancing games; this latter result was extended by Fotakis and Spirakis [12] to also hold for network congestion games on series-parallel graphs. Swamy [23] studied more general (e.g., polynomial) latency functions for the case of atomic congestion games with splittable demands and presented taxes that ensure that the optimal assignment is a pure Nash equilibrium.

In this paper we show the following results concerning non-refundable taxes. For the case of graph balancing games and unweighted clients with different sensitivities, we present an 1.618-pure-efficient tax function. This is the first class of asymmetric load balancing games for which an upper bound better than 2 is achieved, while we note that the lower bound of 11/10 presented in [3] also holds for graph balancing games. Recall that the price of anarchy of these games can be at least 2.012 [22]. Our tax function exploits the structure of graph balancing games and also uses the optimal assignment which can be computed in polynomial time. Then, we consider symmetric load balancing games with unweighted clients and servers with polynomial latency functions of degree p.

We prove a negative result that no non-refundable tax function can be better than $\frac{(p+1)^{1+1/p}}{(p+1)^{1+1/p}-p}$-pure-efficient, i.e., $O\left(\frac{p}{\ln p}\right)$-pure-efficient. Note that this lower bound matches the known upper bound on the price of anarchy of these games which is a corollary of the relation to symmetric non-atomic congestion games in [11] and the upper bounds of [19]. Next, we focus on the makespan as the social cost. For the case of pure Nash equilibria and weighted clients on m related servers, we present a 2-pure-efficient tax function, greatly improving upon the $\Theta(\frac{\log m}{\log \log m})$ bound on the price of anarchy presented in [7]. The tax function is defined using a particular fractional schedule of clients to servers. We also present a lower bound that shows that this tax function is best possible. Finally, for mixed Nash equilibria we observe that the introduction of taxes does not mitigate significantly the impact of selfishness, since no better than $O(\frac{\log m}{\log \log m})$-mixed-efficient taxes exist, even for games with unweighted clients on identical servers.

The rest of the paper is structured as follows. We begin by presenting, in Section 2, our result concerning graph balancing games. We continue in Section 3 with the negative result about non-refundable taxes in symmetric load balancing games with polynomial latency functions. The results concerning the objective of minimizing the makespan are presented in Section 4.

2 Efficient Taxes for Graph Balancing Games

In this section, we present 1.618-pure-efficient tax functions for graph balancing games. This is the first subclass of asymmetric load balancing games which is proved to have better than 2-pure efficient taxes. The tax function is simple and exploits the structure of the game. We will assign very small taxes to the servers so that each server is assigned a different tax. So, although we prove the result ignoring the total taxes paid by the clients, this quantity can become arbitrarily small and our result carries over to non-refundable taxes by adding an ϵ factor to the efficiency.

Consider a graph balancing game with a set of clients $\mathcal{U}$ (with $|\mathcal{U}| = n$) and let $\hat{\delta}$ be such that $0 < \hat{\delta} \leq 1/\max_i \gamma_i$, where γ_i is the tax sensitivity of client i. First, we compute an optimal assignment and denote by o_j the number of clients using server j in this assignment. This computation can be done in polynomial time by a natural reduction to a minimum cost flow problem on a single-source network, similar to the reduction presented in [9] for computing an equilibrium of symmetric congestion games that minimizes Rosenthal's potential. Then, we consider the graph having a node for each server j and an edge between two different nodes j_1 and j_2 for each client that has servers j_1 and j_2 as permissible servers. For each such edge corresponding to a client i, we define the edge's *optimal node* to be the endpoint corresponding to the server that client i uses in the optimal assignment. We compute an orientation of the edges so that the corresponding directed graph is acyclic. Then, either this directed graph or the one in which all edges have opposite directions have the following property: at most half of the edges point to their non-optimal node. We select the orientation

that has this property and assign different taxes from $\left\{\frac{1}{n}\hat{\delta}, \frac{2}{n}\hat{\delta}, ..., \frac{n-1}{n}\hat{\delta}, \hat{\delta}\right\}$ to the nodes/servers so that for any edge directed from j_1 to j_2, it is $\delta_{j_1} > \delta_{j_2}$.

Now, consider any pure Nash equilibrium of the extended game and let n_j denote the number of clients using server j. For a client i, denote by j_1 the server that client i uses in the pure Nash equilibrium and let j_2 be the other permissible server of client i ($j_1 = j_2$ if the client has only one permissible server). Since no client has an incentive to change her strategy, it is $n_{j_1} + \gamma_i \delta_{j_1} \leq n_{j_2} + 1 + \gamma_i \delta_{j_2}$. This means that $n_{j_1} \leq n_{j_2} + 1 + \gamma_i (\delta_{j_2} - \delta_{j_1})$ and $n_{j_1} \leq n_{j_2} + 1$ since n_{j_1} and n_{j_2} are integers and $\gamma_i (\delta_{j_2} - \delta_{j_1}) < 1$ by the definition of the tax function. This inequality holds for any client and we conclude that any pure Nash equilibrium of the extended game is also a pure Nash equilibrium for the original game.

We now show that any pure Nash equilibrium for the extended game is a $\frac{1}{2}$-PNE for the original game, i.e., a pure Nash equilibrium that satisfies the property

$$\sum_j n_j^2 \leq \sum_j \left(n_j o_j + \frac{o_j}{2}\right). \tag{1}$$

For each client i, we denote by j_i and j_i' the servers she uses in the pure Nash equilibrium and in the optimal assignment, respectively. Denote by S the set of clients i such that $j_i = j_i'$. Denote by F the set of clients i such that $j_i \neq j_i'$ and $\delta_{j_i} > \delta_{j_i'}$. Then, the condition $n_{j_i} + \gamma_i \delta_{j_i} \leq n_{j_i'} + 1 + \gamma_i \delta_{j_i'}$, implying that client i has no incentive to use the server she uses in the optimal assignment, implies that $n_{j_i} \leq n_{j_i'}$, since n_{j_i} and $n_{j_i'}$ are integers and $\delta_{j_i} > \delta_{j_i'}$. The condition $n_{j_i} \leq n_{j_i'} + 1$ holds for any client i not belonging in S and F since the pure Nash equilibrium for the extended game is also a pure Nash equilibrium for the original game.

By the definition of the tax function, we have that $|\mathcal{U} \setminus (F \cup S)| \leq |\mathcal{U}|/2 = \frac{1}{2}\sum_j o_j$. By considering the equilibrium conditions for all clients, we have

$$\begin{aligned}
\sum_j n_j^2 &= \sum_{i=1}^{n} \sum_{j:j_i=j} n_j \\
&= \sum_{i\in S} \sum_{j:j_i=j} n_j + \sum_{i\in F} \sum_{j:j_i=j} n_j + \sum_{i\in\mathcal{U}\setminus(F\cup S)} \sum_{j:j_i=j} n_j \\
&\leq \sum_{i\in S} \sum_{j:j_i'=j} n_j + \sum_{i\in F} \sum_{j:j_i'=j} n_j + \sum_{i\in\mathcal{U}\setminus(F\cup S)} \sum_{j:j_i'=j} (n_j + 1) \\
&= \sum_{i=1}^{n} \sum_{j:j_i'=j} n_j + |\mathcal{U} \setminus (F \cup S)| \\
&\leq \sum_j n_j o_j + \frac{1}{2}\sum_j o_j \\
&= \sum_j \left(n_j o_j + \frac{o_j}{2}\right).
\end{aligned}$$

This completes the proof of inequality (1).

In our analysis, we will consider all $\frac{1}{2}$-PNE for the original graph balancing game, and we will show that their price of anarchy is at most $\frac{1+\sqrt{5}}{2}$. We will need the following technical claim.

Lemma 1. *For any non-negative integers x and y,*

$$\frac{3-\sqrt{5}}{4}x^2 + \frac{3+\sqrt{5}}{4}y^2 \geq xy + \frac{3(\sqrt{5}-1)}{4}y - \frac{3\sqrt{5}-5}{4}x.$$

Theorem 1. *For any graph balancing game, the tax function described above is a $\frac{1+\sqrt{5}}{2} \approx 1.618$-pure-efficient tax.*

Proof. We will show that the price of anarchy of any $\frac{1}{2}$-PNE of a graph balancing game is at most $\frac{1+\sqrt{5}}{2}$. Again, we denote by n_j and o_j the number of clients in server j in the $\frac{1}{2}$-PNE and in the optimal assignment, respectively. By inequality (1) and since $\sum_j n_j = \sum_j o_j$, we have that the social cost is

$$\begin{aligned}
\sum_j n_j^2 &\leq \sum_j \left(n_j o_j + \frac{o_j}{2}\right) \\
&= \sum_j \left(n_j o_j + \frac{o_j}{2}\right) + \frac{3\sqrt{5}-5}{4}\sum_j o_j - \frac{3\sqrt{5}-5}{4}\sum_j n_j \\
&= \sum_j \left(n_j o_j + \frac{3(\sqrt{5}-1)}{4}o_j - \frac{3\sqrt{5}-5}{4}n_j\right) \\
&\leq \sum_j \left(\frac{3-\sqrt{5}}{4}n_j^2 + \frac{3+\sqrt{5}}{4}o_j^2\right) \\
&= \frac{3-\sqrt{5}}{4}\sum_j n_j^2 + \frac{3+\sqrt{5}}{4}\sum_j o_j^2
\end{aligned}$$

where the first equality follows since $\sum_j n_j = \sum_j o_j$, the second and third equalities are obvious, and the second inequality follows by Lemma 1. We obtain that the price of anarchy is

$$\frac{\sum_j n_j^2}{\sum_j o_j^2} \leq \frac{\frac{3+\sqrt{5}}{4}}{1-\frac{3-\sqrt{5}}{4}} = \frac{1+\sqrt{5}}{2}.$$

□

Broadening the class of load balancing games that admit better than 2-pure-efficient taxes (or even pure-optimal taxes) is an interesting open problem.

3 Non-refundable Taxes in Symmetric Load Balancing

We now proceed to answer in a negative way a question posed in [3] concerning non-refundable taxes in symmetric load balancing games, i.e., whether taxes can diminish the effect of selfishness. Our following theorem suggests that taxes do not help in the case of symmetric load balancing with polynomial latency

functions of degree p, since for any tax function, the price of anarchy of the extended game in these games is $\frac{(p+1)^{1+1/p}}{(p+1)^{1+1/p}-p} \in O\left(\frac{p}{\ln p}\right)$. Clearly, our result also demonstrates that the known upper bound on the price of anarchy of such games (without taxes) is tight.

Theorem 2. *For any $p \geq 1$ and any $\epsilon > 0$, there exists a symmetric load balancing game with polynomial latency functions of degree p that does not admit better than $(\rho - \epsilon)$-pure-efficient non-refundable taxes where $\rho = \frac{(p+1)^{1+1/p}}{(p+1)^{1+1/p}-p} \in O\left(\frac{p}{\ln p}\right)$.*

Proof. Let $k \geq 2$ be an integer and define $\lambda = \frac{2k^{p+1}-k^p-(k-1)^{p+1}}{k}$. We have $\lambda = k^p + k^p\left(\left(1-\frac{1}{k}\right) - \left(1-\frac{1}{k}\right)^{p+1}\right)$ and, since $k \geq 2$ and $p \geq 1$, it is $k^p < \lambda < 2k^p$. Define $y^* = k - \left\lfloor \left(\frac{\lambda}{p+1}\right)^{1/p} \right\rfloor$. Since $p \geq 1$ and $\lambda < 2k^p$, it is $1 \leq y^* \leq k$.

Consider a game with k clients where each client j has $\gamma_j = 1$, and $k+1$ servers $0, 1, ..., k$. Server 0 has latency function x^p while each of the other k servers has latency function λx^p. The assignment in which server 0 has $k - y^*$ clients, y^* among the other servers have exactly one client and any other server is empty has cost

$$opt = (k - y^*)^{p+1} + y^*\lambda.$$

In the absence of taxes, the assignment where all clients select server 0 is a pure Nash equilibrium since each of them has a latency of k^p and, in case a client decides to choose another server, she would face latency $\lambda > k^p$. The cost of this equilibrium is $cost = k^{p+1}$ and the price of anarchy is

$$PoA \geq \frac{cost}{opt} = \frac{k^{p+1}}{(k - y^*)^{p+1} + y^*\lambda}.$$

Therefore, in order to avoid this assignment as an equilibrium of the extended game, we have to assign taxes in such a way that at least one client has an incentive to change her choice. So, without loss of generality, we assume that there is a tax function δ, for which it holds that $\delta_0(w) = \alpha$ and $\delta_j(w) = 0$, for any $1 \leq j \leq k$. Note that, for any $\alpha \leq \lambda - k^p$, the aforementioned assignment remains a pure Nash equilibrium of the extended game, since any client at server 0 would have a cost of $k^p + \alpha \leq \lambda$. Now, assume that $\alpha = \lambda - k^p + \epsilon$ for any $\epsilon > 0$. Then, any client would have a incentive to leave server 0 and move to another server. Then, assuming that one client moves, the total cost $cost'$ (latency plus taxes) of the resulting assignment would be

$$\begin{aligned} cost' &= (k-1)^{p+1} + \alpha\,(k-1) + \lambda \\ &> (k-1)^{p+1} + (\lambda - k^p)\,(k-1) + \lambda \\ &= (k-1)^{p+1} + \lambda\,(k-1) - k^p\,(k-1) + \lambda \\ &= \lambda k + (k-1)^{p+1} - k^{p+1} + k^p \\ &= k^{p+1} \\ &= cost. \end{aligned}$$

Applying similar reasoning, it is not hard to see that by increasing $\delta_0(w) = \alpha$ even more so that more clients have an incentive to leave server 0, the total cost similarly increases. Therefore, the total cost is minimized by setting $\alpha = 0$. Observe that $\lim_{k\to\infty} \frac{\lambda}{k^p} = 1$ and $\lim_{k\to\infty} \frac{y^*}{k} = 1 - \left(\frac{1}{p+1}\right)^{1/p}$. Hence,

$$\begin{aligned}
\lim_{k\to\infty} \frac{k^{p+1}}{(k-y^*)^{p+1} + y^*\lambda} &= \lim_{k\to\infty} \frac{1}{(1-\frac{y^*}{k})^{p+1} + \frac{y^*\lambda}{k^{p+1}}} \\
&= \frac{1}{\left(\frac{1}{p+1}\right)^{1+1/p} + 1 - \left(\frac{1}{p+1}\right)^{1/p}} \\
&= \frac{(p+1)^{1+1/p}}{(p+1)^{1+1/p} - p} \\
&= \rho.
\end{aligned}$$

Hence, for any $\epsilon > 0$, by setting k to a sufficiently large value, we obtain that the price of anarchy becomes at least $\rho - \epsilon$. □

4 Minimizing the Makespan

In this section we focus on the makespan as the social cost. We consider the well-known case where servers are related, i.e., server j has latency function $\alpha_j x$. Our upper bound uses tax functions that assign to each server a tax of either 0 or ∞. In this setting, there is no difference between refundable and non-refundable taxes since no client is assigned to a server where it has to pay an infinite tax. Furthermore, the tax sensitivity of each client does not affect her behavior.

Denote by n the number of clients and by m the number of servers. We assume that the servers are sorted in non-decreasing order of α_i (i.e., $\alpha_i \leq \alpha_{i+1}$) and clients are sorted in non-increasing order of their weight (i.e., $w_i \geq w_{i+1}$). We define the following procedure that produces fractional schedules of makespan $T \geq \frac{\sum_i w_i}{\sum_i 1/\alpha_i}$. Observe that the quantity $\frac{\sum_i w_i}{\sum_i 1/\alpha_i}$ is a lower bound on the makespan of any fractional schedule; the numerator is the total weight of the clients and the denominator is the "capacity" of all servers.

```
1.  set j = 1, i = 1, and t = 0;
2.  while i ≤ n do
3.      if T − t ≥ α_j w_i then
4.          put the remaining weight of client i at server j;
5.          set t = t + α_j w_i and i = i + 1;
6.      else
7.          put weight (T−t)/α_j of client i at server j;
8.          set w_i = w_i − (T−t)/α_j, j = j + 1, and t = 0;
```

What the above procedure is doing is to consider each client (according to their ordering) and put as much of her weight as possible to the server of smallest

index so that the latency does not exceed T. This will end up with a fractional schedule in which there exists a server j' such that the latency of all servers $j \leq j'$ is exactly T, (if $j' < m$) the latency of server $j'+1$ is at most T, and the latency of all servers $j > j'+1$ (if any) is 0. Each client occupies consecutive servers and, furthermore, at most one client may have non-zero weights in two specific consecutive servers.

Given a value of T, the schedule produced by the procedure above is called 2-*feasible* if for any client i and any two consecutive servers j and $j+1$, it holds that $\alpha_j w_i^j + \alpha_{j+1} w_i^{j+1} \leq T$, where w_i^j denotes the weight of client i assigned to server j. We start with the value $T = \frac{\sum_i w_i}{\sum_i 1/\alpha_i}$ and run the procedure. If the schedule produced is 2-feasible, we stop. Otherwise, we increase T until the schedule produced by the procedure is 2-feasible. Let T^* be the corresponding value of T (i.e., the minimum value for which the schedule produced is 2-feasible). Clearly, if $T^* > \frac{\sum_i w_i}{\sum_i 1/\alpha_i}$, there will be at least one client i and two consecutive servers j and $j+1$ such that $\alpha_j w_i^j + \alpha_{j+1} w_i^{j+1} = T^*$.

We now describe the tax function. We partition the clients into groups according to their weight, so that two clients i_1 and i_2 belong to same group when $w_{i_1} = w_{i_2}$. We denote by w_g^* the weight corresponding to group g. Let S_g denote the set of servers that contain non-zero weights of clients belonging to group g in the fractional schedule of makespan T^*. If $|S_g| = 1$, then we set $\delta_j(w_g^*) = 0$ for the server $j \in S_g$ and $\delta_{j'}(w_g^*) = \infty$ for any other server $j' \notin S_g$. Otherwise, when $|S_g| > 1$, we distinguish between two cases depending on whether the last server of S_g (i.e., the one with the larger index) contains only clients of group g or also clients of different groups. In the first case, we set $\delta_j(w_g^*) = 0$ for any server $j \in S_g$ and $\delta_{j'}(w_g^*) = \infty$ for any other server $j' \notin S_g$, while in the second case, we set $\delta_j(w_g^*) = 0$ for the $|S_g| - 1$ servers of S_g with smallest index and $\delta_{j'}(w_g^*) = \infty$ for any other server j'. In any case, we denote with Δ_g the set of servers j for which $\delta_j(w_g^*) = 0$.

We show the following result.

Theorem 3. *For any symmetric load balancing game on related servers, the above tax function is* 2*-pure-efficient with respect to the makespan.*

Proof. Consider the 2-feasible fractional schedule of makespan T^* produced as above. We first show that the optimal assignment has makespan at least T^*. This clearly holds if $T^* = \frac{\sum_i w_i}{\sum_i 1/\alpha_i}$. Otherwise, there will be a client i and two consecutive servers j and $j+1$ such that $\alpha_j w_i^j + \alpha_{j+1} w_i^{j+1} = T^*$. Then, all clients with smaller index than i are fractionally scheduled at servers $1, ..., j$ which have latency exactly T^*. In any integral schedule, either all of the clients $1, ..., i$ will be scheduled to servers $1, ..., j$ or some of them will be scheduled at some server with larger index than j. In the first case, the makespan will be at least T^* since the total weight of clients assigned to servers $1, ..., j$ does not decrease compared to the fractional schedule. In the second case, a client of weight at least w_i will be assigned to a server j' with $\alpha_{j'} \geq \alpha_{j+1} \geq \alpha_j$. This server will have latency at least $\alpha_{j'} w_i \geq \alpha_j w_i^j + \alpha_{j+1} w_i^{j+1} = T^*$.

Now, we show that there exists an integral schedule with makespan at most $2T^*$ in which each client in group g selects a server from the set Δ_g. The clients that have non-zero weight in the server of S_g with the smallest index are scheduled in this server. Each other client of group g for which the server with largest index containing a non-zero amount of her weight in the fractional schedule is j is scheduled at server $j-1$. In this way, the total weight of any server j may increase by at most the weight of clients in server $j+1$ in the fractional schedule. Since $\alpha_j \leq \alpha_{j+1}$, the latency at server j will not exceed $2T^*$.

Observe that the tax function essentially divides the original game into subgames in the following sense. In any pure Nash equilibrium, the clients of group g with $|\Delta_g| = 1$ are forced to use the server of Δ_g. The clients of group g with $|\Delta_g| > 1$ play a symmetric game with linear latency functions at server j of the form $\alpha_j x + \beta_j$. Here, β_j denotes the latency at server j due to clients not belonging to group g which are forced to use server j. Furthermore, by the definition of the tax function, the sets Δ_g with size more than 1 are disjoint and, hence, the corresponding sets of clients do not interfere. It is not hard to see that any equilibrium in each subgame of clients of group g has the minimum possible integral makespan, i.e., at most $2T^*$. This completes the proof of the theorem. □

The next theorem states that this tax function is best possible. The proof is omitted; it will appear in the final version.

Theorem 4. *For any $\epsilon > 0$, there exists a load balancing game on m identical servers that does not admit better than $(2-\epsilon)$-pure-efficient taxes with respect to the makespan.*

Unfortunately, taxes cannot significantly improve the price of anarchy with respect to the makespan over mixed Nash equilibria. To show this, we use a construction that we have also used in [3] to lower-bound the efficiency of taxes at mixed Nash equilibria with respect to the total latency. The construction applies to symmetric load balancing games with identical clients and identical servers and the proof follows by a standard balls-to-bins argument.

Consider a tax function δ. Without loss of generality, we assume that $\delta_j \leq \delta_{j'}$ for $j < j'$. Let k be equal to m if $1 + \frac{\sum_{j=1}^{m-1} \delta_j}{m-1} > \delta_m$, otherwise k is equal to the largest integer such that $\frac{m-1+\sum_{j=1}^{k} \delta_j}{k} \leq \delta_{k+1}$. Let $D = \sum_{j=1}^{k} \delta_j$. Consider the following assignment y for all clients. Client i uses server j with probability $y_{ij} = \frac{1}{k} + \frac{D}{k(m-1)} - \frac{\delta_j}{m-1}$ if $j \leq k$ and $y_{ij} = 0$ otherwise. Notice that all clients have the same probability distribution. It can be verified that y is a mixed Nash equilibrium of the extended game.

In order to compute the expected makespan, it suffices to observe that it is the expectation of the maximum number of balls at any bin when m balls are thrown independently at m bins according to the probability distribution y. It is well-known (e.g., see [24]) that this expectation is minimized to $\Theta\left(\frac{\log m}{\log \log m}\right)$ when y is the uniform distribution. Thus, we obtain the following statement.

Theorem 5. *There exists a symmetric load balancing game on m servers that does not admit better than $\Omega\left(\frac{\ln m}{\ln \ln m}\right)$-mixed-efficient taxes with respect to the makespan.*

Note that this bound matches the price of anarchy of symmetric load balancing with identical servers [7,14]. The price of anarchy for related servers is slightly higher [7]. We leave as an open problem whether taxes can improve the price of anarchy with respect to the makespan in this particular case and, more importantly, in the more general case of congestion games.

References

1. Awerbuch, B., Azar, Y., Epstein, A.: The price of routing unsplittable flow. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005), pp. 57–66 (2005)
2. Caragiannis, I., Flammini, M., Kaklamanis, C., Kanellopoulos, P., Moscardelli, L.: Tight bounds for selfish and greedy load balancing. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 311–322. Springer, Heidelberg (2006)
3. Caragiannis, I., Kaklamanis, C., Kanellopoulos, P.: Taxes for linear atomic congestion games. In: Azar, Y., Erlebach, T. (eds.) ESA 2006. LNCS, vol. 4168, pp. 184–195. Springer, Heidelberg (2006)
4. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005), pp. 67–73 (2005)
5. Cole, R., Dodis, Y., Roughgarden, T.: Pricing network edges for heterogeneous selfish users. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC 2003), pp. 521–530 (2003)
6. Cole, R., Dodis, Y., Roughgarden, T.: How much can taxes help selfish routing? Journal of Computer and System Sciences 72(3), 444–467 (2006)
7. Czumaj, A., Vöcking, B.: Tight bounds for worst-case equilibria. ACM Transactions on Algorithms 3(1) (2007)
8. Ebenlendr, T., Krcal, M., Sgall, J.: Graph balancing: a special case of scheduling unrelated parallel machines. In: Proceedings ot the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008), pp. 483–490 (2008)
9. Fabrikant, A., Papadimitriou, C., Talwar, K.: On the complexity of pure equilibria. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004), pp. 604–612 (2004)
10. Fleischer, L., Jain, K., Mahdian, M.: Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004), pp. 277–285 (2004)
11. Fotakis, D.: Stackelberg strategies for atomic congestion games. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 299–310. Springer, Heidelberg (2007)
12. Fotakis, D., Spirakis, P.: Cost-balancing tolls for atomic network congestion games. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 179–190. Springer, Heidelberg (2007)

13. Karakostas, G., Kolliopoulos, S.: Edge pricing of multicommodity networks for heterogeneous selfish users. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004), pp. 268–276 (2004)
14. Koutsoupias, E., Mavronicolas, M., Spirakis, P.: Approximate equilibria and ball fusion. Theory of Computing Systems 36(6), 683–693 (2003)
15. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 404–413. Springer, Heidelberg (1999)
16. Lücking, T., Mavronicolas, M., Monien, B., Rode, M.: A new model for selfish routing. In: Diekert, V., Habib, M. (eds.) STACS 2004. LNCS, vol. 2996, pp. 547–558. Springer, Heidelberg (2004)
17. Papadimitriou, C.: Algorithms, games and the internet. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001), pp. 749–753 (2001)
18. Rosenthal, R.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory 2, 65–67 (1973)
19. Roughgarden, T.: The price of anarchy is independent of the network topology. Journal of Computer and System Sciences 67(2), 341–364 (2003)
20. Roughgarden, T.: Routing games. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds.) Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
21. Roughgarden, T., Tardos, E.: How bad is selfish routing? Journal of the ACM 49(2), 236–259 (2002)
22. Suri, S., Tóth, C., Zhou, Y.: Selfish load balancing and atomic congestion games. Algorithmica 47(1), 79–96 (2007)
23. Swamy, C.: The effectiveness of Stackelberg strategies and tolls for network congestion games. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), pp. 1133–1142 (2007)
24. Vöcking, B.: How asymmetry helps load balancing. Journal of the ACM 50(4), 568–589 (2003)
25. Vöcking, B.: Selfish load balancing. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds.) Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)