Tight Bounds for Selfish and Greedy Load Balancing*

Ioannis Caragiannis[1], Michele Flammini[2], Christos Kaklamanis[1], Panagiotis Kanellopoulos[1], and Luca Moscardelli[2]

[1] Research Academic Computer Technology Institute and Dept. of Computer Engineering and Informatics University of Patras, 26500 Rio, Greece
[2] Dipartimento di Informatica, Università di L' Aquila Via Vetoio, Coppito 67100, L' Aquila, Italy

Abstract. We study the load balancing problem in the context of a set of clients each wishing to run a job on a server selected among a subset of permissible servers for the particular client. We consider two different scenarios. In *selfish load balancing*, each client is selfish in the sense that it selects to run its job to the server among its permissible servers having the smallest latency given the assignments of the jobs of other clients to servers. In *online load balancing*, clients appear online and, when a client appears, it has to make an irrevocable decision and assign its job to one of its permissible servers. Here, we assume that the clients aim to optimize some global criterion but in an online fashion. A natural local optimization criterion that can be used by each client when making its decision is to assign its job to that server that gives the minimum increase of the global objective. This gives rise to *greedy* online solutions. The aim of this paper is to determine how much the quality of load balancing is affected by selfishness and greediness.

We characterize almost completely the impact of selfishness and greediness in load balancing by presenting new and improved, tight or almost tight bounds on the price of anarchy and price of stability of selfish load balancing as well as on the competitiveness of the greedy algorithm for online load balancing when the objective is to minimize the total latency of all clients on servers with linear latency functions.

1 Introduction

We study the load balancing problem in the context of a set of clients each wishing to run a job on a server selected among a subset of permissible servers for the particular client. We consider two different scenarios. In the first, called *selfish load balancing* (or *load balancing games*), each client is selfish in the sense that it selects to run its job to the server among its permissible servers having the smallest latency given the assignments of the jobs of other clients to servers.

* This work was partially supported by the European Union under IST FET Integrated Project 015964 AEOLUS and COST Action 293 GRAAL.

M. Bugliesi et al. (Eds.): ICALP 2006, Part I, LNCS 4051, pp. 311–322, 2006.

In the second scenario, called *online load balancing*, clients appear online and, when a client appears, it has to make an irrevocable decision and assign its job to one of its permissible servers. Here, we assume that the clients are not selfish and aim to optimize some global objective but in an online fashion (i.e., without any knowledge of clients that may arrive in the future). A natural local optimization criterion that can be used by each client when making its decision is to assign its job to that server that gives the minimum increase of the global objective. This gives rise to greedy online solutions. The aim of this paper is to answer the question of how much the quality of load balancing is affected by selfishness and greediness.

Load balancing games are special cases of the well-known *congestion games* introduced by Rosenthal [22] and studied in a sequence of papers [4,7,8,11,13,19,23, 24]. In congestion games there is a set E of resources, each 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 set of strategies $S_i \subseteq 2^E$ (each strategy of player i is a set of resources). An assignment $A = (A_1, ..., A_n)$ is a vector of strategies, one strategy for each player. The cost of a player for an assignment A is defined as $cost(i) = \sum_{e \in A_i} f_e(n_e(A))$, where $n_e(A)$ is the number of players using resource e in A, while the cost of an assignment is the total cost of all players. An assignment is a *pure Nash equilibrium* if no player has any incentive to unilaterally deviate to another strategy, i.e., $cost_i(A) \leq cost_i(A_{-i}, s)$ for any player i and for any $s \in S_i$, where (A_{-i}, s) is the assignment produced if just player i deviates from A_i to s. This inequality is also known as the *Nash condition*. We use the term *social cost* to refer to the cost of a pure Nash equilibrium. In *weighted congestion games*, each player has a weight w_i and the latency of a resource e depends on the total weight of the players that use e. For this case, a natural social cost function is the weighted sum of the costs of all players (or the weighted average of their costs). In *linear congestion games*, the latency function of resource e is of the form $f_e(x) = \alpha_e x + b_e$ with non-negative constants α_e and b_e. Load balancing games are linear 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.

We evaluate the quality of solutions of a load balancing game by comparing the social cost of Nash equilibria to the cost of the optimal assignment (i.e., the minimum cost). We use the notions of *price of anarchy* introduced in a seminal work of Koutsoupias and Papadimitriou [16] (see also [20]) and *price of stability* (or *optimistic price of anarchy*) defined as follows. The price of anarchy/stability of a load balancing game is defined as the ratio of the maximum/minimum social cost over all Nash equilibria over the optimal cost. The price of anarchy/stability for a class of load balancing games is simply the highest price of anarchy/stability among all games belonging to that class.

[10,12,13,14,15,18] study various games which can be thought of as special cases of congestion games with respect to the complexity of computing equilibria of best/worst social cost and the price of anarchy when the social cost is defined

as the maximum latency experienced by any player. The social cost of the total latency has been studied in [4,7,17,26]. The authors in [17] study symmetric load balancing games where all servers are permissible for any client and show tight bounds on the price of anarchy of 4/3 for arbitrary servers and 9/8 for identical servers with weighted clients. In two recent papers, Awerbuch et al. [4] and Christodoulou and Koutsoupias [7] prove tight bounds on the price of anarchy of congestion games with linear latency functions. Among other results, they show that the price of anarchy of pure Nash equilibria is 5/2 while for mixed Nash equilibria or pure Nash equilibria of weighted clients it is $\frac{3+\sqrt{5}}{2} \approx 2.618$.

Does the fact that load balancing games are significantly simpler than congestion games in general have any implications for their price of anarchy? We give a negative answer to this question by showing that the 5/2 upper bound (as well as the $\frac{3+\sqrt{5}}{2}$ upper bound for weighted clients) is tight. This is interesting since the upper bounds for congestion games (as well as an earlier upper bound of 5/2 proved specifically for load balancing [26]) are obtained using only the *Nash inequality* (i.e., the inequality obtained by summing up the Nash condition inequalities over all players' strategies) and the definition of the social cost. So, it is somewhat surprising that load balancing games are as general as congestion games in terms of their price of anarchy and that the Nash inequality provides sufficient information to characterize their price of anarchy.

An important special case of load balancing is when servers have identical linear latency functions. Here, better upper bounds on the price of anarchy can be obtained. Note that this is not the case for congestion games since, as it was observed in [7], any congestion game can be transformed to a congestion game on identical resources (and, hence, the lower bounds of [4,7] hold for congestion games with identical resources as well). Suri et al. [26] prove that the price of anarchy of selfish load balancing on identical servers is between $1+2/\sqrt{3} \approx 2.1547$ and 2.012067. Again, the upper bound is obtained by using the Nash inequality and the definition of the social cost. We improve this result by showing that the lower bound is essentially tight. Besides the Nash inequality, our proof also exploits structural properties of the game with the highest price of anarchy. We argue that this game can be represented as a directed graph (called the *game graph*) and, then, structural properties of the game follow as structural properties of this graph. Furthermore, for weighted clients and identical servers, we prove that the price of anarchy is at least 5/2.

The price of stability of congestion games has been recently studied in [8] where it was shown that it is between $1 + 1/\sqrt{3} \approx 1.577$ and 1.6. The technique used to obtain the upper bound is to consider pure Nash equilibria with potential not larger than the potential of the optimal assignment and bound their social cost in terms of the optimal cost using the Nash inequality. Using the same technique but also tightening the analysis, we show that the lower bound is tight. Does the fact that load balancing games are significantly simpler than congestion games have any implications in their price of stability? We give a positive answer to this question by showing that the price of stability of selfish load balancing is 4/3. The proof of the upper bound makes use of completely

different arguments since the techniques used for congestion games provably cannot be used to obtain this bound.

From the algorithmic point of view, load balancing has been studied extensively, including papers studying online versions of the problem (e.g., [1,2,3,5,6,9, 21,25,26]). In online load balancing, clients appear in online fashion; when a client appears, it has to make an irrevocable decision and assign its job to a server. In our model, servers have linear latency functions and the objective is to minimize the total latency, i.e., the sum of the latencies experienced by all clients. Clients may also own jobs with non-negative weights; in this case, the objective is to minimize the weighted sum of the latencies experienced by all clients. A natural greedy algorithm proposed in [3] for this problem is to assign each client to that server that yields the minimum increase to the total latency (ties are broken arbitrarily). This results to *greedy assignments*. Given an instance of online load balancing, an assignment of clients to servers is called a greedy assignment if the assignment of a client to a server minimizes the increase in the cost of the instance revealed up to the time of its appearance. Following the standard performance measure in competitive analysis, we evaluate the performance of this algorithm in terms of its *competitiveness* (or *competitive ratio*). The competitiveness of the greedy algorithm on an instance is the maximum ratio of the cost of any greedy assignment over the optimal cost and its competitiveness on a class of load balancing instances is simply the maximum competitiveness over all instances in the particular class.

The performance of greedy load balancing with respect to the total latency has been studied in [3,26]. Awerbuch et al. [3] consider a more general model where each client owns a job with a load vector denoting the impact of the job to each server (i.e., how much the assignment of the job to a server will increase its load) and the objective is to minimize the L_p norm of the load of the servers. In the context similar to the one studied in the current paper, their results imply a $3 + 2\sqrt{2} \approx 5.8284$ upper bound. This result applies also in the case of weighted clients where the objective is to minimize the weighted average latency. Suri et al. [26] consider the same model as ours and show upper bounds of $17/3$ and $2+\sqrt{5} \approx 4.2361$ for arbitrary servers and identical servers, respectively. In a way similar to the study of the price of anarchy of congestion games, [26] develops a *greedy inequality* which is used to obtain the upper bounds on competitiveness. They also present a lower bound of 3.0833 for the competitiveness of greedy assignments in the case of identical servers.

The main question left open by the work of [26] is whether arbitrary servers do hurt the competitiveness of greedy load balancing. We give a positive answer to this question as well. By a rather counterintuitive construction, we show that the $17/3$ upper bound of [26] is tight. This is interesting since it indicates that the greedy inequality is powerful enough to characterize the competitiveness of greedy load balancing. We also consider the case of identical servers where we almost close the gap between the upper and lower bounds of [26] by showing that the competitiveness of greedy load balancing is between 4 and $\frac{2}{3}\sqrt{21} + 1 \approx 4.05505$. In the proof of the upper bound, we use the greedy inequality

but, more importantly, we also use arguments for the structure of greedy and optimal assignments of instances that yield the worst competitiveness. In a similar way to the case of selfish load balancing, we argue that such instances can be represented as directed graphs (called *greedy graphs*) that enjoy particular structural properties. In the case of weighted clients, we present a tight lower bound of $3 + 2\sqrt{2}$ on identical servers matching the upper bound of [3].

The rest of the paper is structured as follows. We present the bounds on the price of stability of linear congestion games and selfish load balancing in Section 2. The bounds on the price of anarchy are presented in Section 3 while the bounds on the competitiveness of greedy load balancing are presented in Section 4. We discuss extensions of the results to selfish and greedy load balancing when clients are weighted and conclude with open problems in Section 5. Due to lack of space, many proofs have been omitted from this extended abstract.

2 Bounds on the Price of Stability

We present a tight upper bound on the price of stability of congestion games. Our proof (omitted) uses the main idea in the proof of [8] and bounds the social cost of any Nash equilibrium having a potential smaller than the potential of the optimal assignment. In the proof we also make use of the Nash inequality which together with the inequality on the potentials yields the upper bound. However, the two inequalities may not be equally important in order to achieve the best possible bound and this is taken into account in our analysis. We obtain the following result. A matching lower bound is presented in [8].

Theorem 1. *The price of stability of congestion games with linear latency functions is at most* $1 + 1/\sqrt{3}$.

In the following we show a tight upper bound of 4/3 on the price of stability of load balancing games. We note that the use of the inequality on the potentials does not suffice since load balancing games may have pure Nash equilibria with potential smaller than the potential of an optimal assignment and with cost strictly larger than 4/3 times the optimal cost. So, in order to prove the 4/3 upper bound on the price of stability of load balancing games, we will use entirely different arguments. Starting from any assignment, we let the clients move (one client moves at each step) until they converge to a pure Nash equilibrium. At each step, the moving client is selected arbitrarily among the clients with current strategy at a server of maximum latency which have an incentive to change their strategy. In our proof, we actually show that the social cost of the pure Nash equilibrium at convergence is no more than 4/3 times the cost of the initial assignment. As a corollary, by starting from an optimal solution, we will obtain that the price of stability is at most 4/3.

Theorem 2. *The price of stability of load balancing games is at most* 4/3.

Proof. Consider a load balancing game, an initial assignment with o_j clients at server j for any j, and the moves as defined above. We denote by n_j the number

of clients at server j at the Nash equilibrium. Also, we denote by $f_j(x) = \alpha_j x + b_j$ the latency function of server j.

We define *segments* as follows. For each server j, consider the set of moves $\mu_1, \mu_2, ..., \mu_k$ into server j at steps $t_1, t_2, ..., t_k$ so that $t_1 < t_2 < ... < t_k$, and the set of moves $\mu'_1, \mu'_2, ..., \mu'_{k'}$ out of server j at steps $t'_1, t'_2, ..., t'_{k'}$ so that $t'_1 < t'_2 < ... < t'_{k'}$. For $i = 1, ..., k$, we match move μ_i with the first move (if any) $\mu'_{i'}$ that happens after move μ_i and has not been matched to any of the moves $\mu_1, ..., \mu_{i-1}$. In this way we obtain *passing segments* which are pairs of a move into server j and a move out of server j, *starting segments* which consist of single moves out of server j which were not matched to any incoming move, and *ending segments* which consist of single moves into server j which were not matched to any outgoing move.

We construct *chains* (i.e., sequence of moves) using the segments defined. A chain begins with the move in a starting segment, terminates with a move in an ending segment, while any two consecutive moves in the chain (if any), one into and one out of the same server j, belong to the same passing segment of server j. A chain may consist of a single move if this belongs to both a starting and an ending segment. For each server j, denote by s_j and e_j the number of starting and ending segments defined at server j, respectively. Equivalently, s_j is the number of chains beginning with a move out of server j and e_j is the number of chains terminating with a move into server j.

To obtain the desired bound, we will use the following lemma. The proof is lengthy and hence omitted; it relies on an inductive argument.

Lemma 1. $\sum_j f_j(o_j)s_j \geq \sum_j f_j(n_j)e_j$.

Using Lemma 1, we have

$$\begin{aligned}
\sum_j f_j(o_j)o_j &= \sum_j \left(f_j(o_j)(o_j - s_j) + f_j(o_j)s_j\right) \\
&\geq \sum_j \left(f_j(o_j)(o_j - s_j) + f_j(n_j)e_j\right) \\
&= \sum_j \left(f_j(o_j)(o_j - s_j) + f_j(n_j)(n_j - o_j + s_j)\right) \\
&= \sum_j \left(\alpha_j \left(o_j^2 - s_j(o_j - n_j) + n_j^2 - n_j o_j\right) + b_j n_j\right) \qquad (1)
\end{aligned}$$

We distinguish between two cases to show that $o_j^2 - s_j(o_j - n_j) + n_j^2 - n_j o_j \geq \frac{3}{4}n_j^2$, for any j. If $n_j \leq o_j$, then since $s_j \leq o_j$, it is $o_j^2 - s_j(o_j - n_j) + n_j^2 - n_j o_j \geq o_j^2 - o_j(o_j - n_j) + n_j^2 - n_j o_j = n_j^2$. If $n_j \geq o_j$, it is $o_j^2 - s_j(o_j - n_j) + n_j^2 - n_j o_j \geq o_j^2 + n_j^2 - n_j o_j = (o_j - n_j/2)^2 + \frac{3}{4}n_j^2 \geq \frac{3}{4}n_j^2$. Hence, (1) yields that

$$\sum_j f_j(o_j)o_j \geq \frac{3}{4}\sum_j \left(\alpha_j n_j^2 + b_j n_j\right) = \frac{3}{4}\sum_j f_j(n_j)n_j. \qquad \square$$

To show that the above result is tight, it suffices to consider, for arbitrarily small $\epsilon > 0$, a game with two servers with latency functions $f_1(x) = (2+\epsilon)x$ and $f_2(x) = x$ and two clients having both servers as strategies.

3 Bounds on the Price of Anarchy

For the study of the price of anarchy, we can consider load balancing games in which each client has at most two strategies. This is clearly sufficient when proving lower bounds. In order to prove upper bounds, we can assume that the highest price of anarchy is obtained by such a game. Consider any load balancing game and let O and N be the optimal assignment and the Nash equilibrium that yields the worst social cost, respectively. The game with the same clients and servers in which each client has its strategies in O and N as strategies also has the same optimal assignment and the same Nash equilibrium (and, consequently the same price of anarchy). We represent such games as directed graphs (called *game graphs*) having a node for each server and a directed edge for each client; the direction of each edge is from the strategy of the client in the optimal assignment to the strategy of the client in the Nash equilibrium. A self-loop indicates that the client has just one strategy.

The next theorem states that the upper bound of 5/2 presented in [26] (and also implied by the results in [4,7] for congestion games) is tight. This bound was known to be tight for congestion games in general but the constructions in the lower bounds in [4,7] are not load balancing games.

Theorem 3. *For any $\epsilon > 0$, there is a load balancing game with price of anarchy at least $5/2 - \epsilon$.*

Proof. We construct a game graph G consisting of a complete binary tree with $k+1$ levels and $2^{k+1}-1$ nodes with a line of $k+1$ edges and $k+1$ additional nodes hung at each leaf. So, graph G has $2k+2$ levels $0, ..., 2k+1$, with 2^i nodes at level i for $i = 0, ..., k$ and 2^k nodes at levels $k+1, ..., 2k+1$. The servers corresponding to nodes of level $i = 0, ..., k-1$ have latency functions $f_i(x) = (2/3)^i x$, the servers corresponding to nodes of level $i = k, ..., 2k$ have latency functions $f_i(x) = (2/3)^{k-1}(1/2)^{i-k}x$, and the servers corresponding to nodes of level $2k+1$ have latency functions $f_{2k+1}(x) = (2/3)^{k-1}(1/2)^k x$. The assignment where all clients select servers corresponding to the endpoint of their corresponding edge which is closer to the root of the game graph can be easily verified that it is a Nash equilibrium. Its cost is $\sum_{i=0}^{k-1} 4 \cdot 2^i (2/3)^i + \sum_{i=k}^{2k} 2^k (2/3)^{k-1}(1/2)^{i-k} = 15(4/3)^k - (2/3)^{k-1} - 12$. To compute an upper bound for the cost of the optimal assignment, it suffices to consider the assignment where all clients select the servers corresponding to nodes which are further from the root. We obtain that the cost of the optimal assignment is at most $\sum_{i=1}^{k-1} 2^i (2/3)^i + \sum_{i=k}^{2k} 2^k (2/3)^{k-1}(1/2)^{i-k} + 2^k (2/3)^{k-1}(1/2)^k = 6(4/3)^k - 4$. Hence, for any $\epsilon > 0$ and for sufficiently large k, the price of anarchy of the game is larger than $5/2 - \epsilon$. □

In the case of identical servers we can show a tight bound on the price of anarchy of approximately 2.012067; a matching lower bound has been presented in [26]. Here, we present the main idea in our analysis to obtain a slightly weaker result; the improved analysis will appear in the final version of the paper.

We will consider the game with the highest price of anarchy and upper-bound the ratio of the social cost of the worst Nash equilibrium to the optimal cost of the particular game. We represent the game by a game graph. We say that server j is of type n_j/o_j meaning that it has n_j clients in the Nash equilibrium and o_j clients in the optimal assignment (equivalently, server j has in-degree n_j and out-degree o_j in the game graph). After observing that each server of type 1/1 can be associated with a neighboring server of type 0/1, the idea behind the proof is to account for their contribution in the social cost together. By extending the neighborhood considered together with each server of type 1/1, we can obtain better and better upper bounds which converge to the lower bound of 2.012067.

In the proof, we make use of the following technical lemma.

Lemma 2. *For any integers x, y, define the functions $g(x,y) = xy + \frac{18+7\sqrt{21}}{30}y - \frac{7\sqrt{21}-12}{30}x$ and $h(x,y) = \frac{6-\sqrt{21}}{10}x^2 + \frac{6+\sqrt{21}}{6}y^2$. For any non-negative integers x, y such that either $x \neq 1$ or $y \neq 1$, it holds that $g(x,y) \leq h(x,y)$. Furthermore, $g(0,1) + g(1,1) = h(0,1) + h(1,1)$.*

Theorem 4. *The price of anarchy of selfish load balancing on identical servers is at most $\frac{2}{3}\sqrt{21} - 1$.*

Proof. Consider a load balancing game on servers with latency function $f(x) = x + b$ and clients having at most two strategies which has the highest price of anarchy. Consider a server j of type 1/1. If a client c had server j as its only strategy (this corresponds to a self-loop in the corresponding game graph), then we may construct a new game by excluding server j and client c from the original one; it can be easily seen that the new game has worse price of anarchy since both the cost of the optimal assignment and the social cost of the Nash equilibrium are decreased by $1 + b$. So, let j' and j'' be the servers to which server j is connected corresponding to clients c_1 and c_2 selecting servers j' and j in the optimal assignment and servers j and j'' in the Nash assignment, respectively.

Server j' is of type 0/1. Assume otherwise that it is of type $n_{j'}/o_{j'}$ for $n_{j'} > 0$ or $o_{j'} > 1$. If $n_{j'} > 0$, we can construct a new game by excluding server j and substituting clients c_1 and c_2 by a client selecting server j' in the optimal assignment and server j'' in the Nash assignment. If $o_{j'} > 1$, then we can add a new server j'_1 and change the strategy of client c_1 to $\{j'_1, j\}$. In both cases, we obtain games with higher price of anarchy.

Denote by F the set of servers of type 1/1 and by S the set of servers of type 0/1 which are connected through an edge to a server in F in the game graph. Also, for each server j in F we denote by $S(j)$ the server of S from which the client destined for j originates. By the Nash inequality, we obtain that $\sum_j (n_j^2 + bn_j) \leq \sum_j (o_j n_j + (1+b)o_j)$ and, since $\sum_j n_j = \sum_j o_j$, we have that

$$\begin{aligned}\sum_j n_j^2 &\leq \sum_j (n_j o_j + o_j) = \sum_j \left(n_j o_j + \frac{18+7\sqrt{21}}{30} o_j - \frac{7\sqrt{21}-12}{30} n_j \right) \\ &= \sum_{j \notin F \cup S} g(n_j, o_j) + \sum_{j \in F} \left(g(n_{S(j)}, o_{S(j)}) + g(n_j, o_j) \right) \\ &\leq \sum_{j \notin F \cup S} h(n_j, o_j) + \sum_{j \in F} \left(h(n_{S(j)}, o_{S(j)}) + h(n_j, o_j) \right) \\ &= \frac{6-\sqrt{21}}{10} \sum_j n_j^2 + \frac{6+\sqrt{21}}{6} \sum_j o_j^2\end{aligned}$$

where the first equality follows since $\sum_j n_j = \sum_j o_j$, the second equality follows by the definition of function g, the second inequality follows by Lemma 2, and the last equality follows by the definition of function h. Hence, we obtain that the price of anarchy is

$$\frac{\sum_j \left(n_j^2 + b n_j\right)}{\sum_j \left(o_j^2 + b o_j\right)} \leq \frac{\sum_j n_j^2}{\sum_j o_j^2} \leq \frac{2}{3}\sqrt{21} - 1. \qquad \square$$

4 Greedy Load Balancing

Similarly to the case of selfish load balancing, in the study of the competitiveness of greedy load balancing, we consider load balancing instances in which each client has at most two strategies. This is clearly sufficient when proving lower bounds. In order to prove upper bounds, we can assume that the highest competitiveness is obtained by such an instance. Consider any load balancing instance and let O and N be the optimal assignment and the greedy assignment of the highest cost, respectively. The instance with the same clients and servers in which each client has its strategies in O and N as strategies also has the same optimal assignment and the same greedy assignment (and, consequently the same competitiveness). We represent such instances as directed graphs (called *greedy graphs*) having a node for each server and a directed edge with timing information for each client; the direction of each edge is from the strategy of the client in the optimal assignment to the strategy of the client in the greedy assignment and the timing information denotes the time the client appears. We can show that the upper bound of [26] for arbitrary servers is tight.

Theorem 5. *For any $\epsilon > 0$, greedy load balancing has competitiveness at least $17/3 - \epsilon$.*

We also study the case of identical servers with latency function $f(x) = x + b$. By reasoning about the structure of the load balancing instance that yields the worst competitiveness and using the greedy inequality developed in [26], we can prove the following theorem.

Theorem 6. *Greedy load balancing on identical servers has competitiveness at most $\frac{2}{3}\sqrt{21} + 1$.*

We also present an almost matching lower bound.

Theorem 7. *For any $\epsilon > 0$, greedy load balancing on identical servers has competitiveness at least $4 - \epsilon$.*

Proof. We assume that there are m servers $s_1, s_2, ..., s_m$, and k groups of clients $g_1, ..., g_k$, where group g_j has m/j^2 clients c_i^j, $1 \leq i \leq m/j^2$. We assume that m is such that all groups have integer size. Each client c_i^j has $s_1, s_2, ..., s_i$ as permissible servers. The clients appear in non-increasing order according to index i, i.e., c_m^1, c_{m-1}^1, ..., $c_{m/4+1}^1$, $c_{m/4}^2$, $c_{m/4}^1$, $c_{m/4-1}^2$, $c_{m/4-1}^1$, ..., $c_{m/9+1}^2$, $c_{m/9+1}^1$, $c_{m/9}^3$, $c_{m/9}^2$, $c_{m/9}^1$, ..., etc.

To upper bound the optimal cost *opt*, it suffices to consider the assignment where each client c_i^j chooses server s_i. We obtain that

$$opt \leq \sum_{i=1}^{k-1} i^2(|g_i| - |g_{i+1}|) + k^2|g_k| = m + m\sum_{i=1}^{k-1} i^2\left(\frac{1}{i^2} - \frac{1}{(i+1)^2}\right)$$
$$= m\left(1 + 2\sum_{i=1}^{k-1} 1/\left(i+1\right) - \sum_{i=1}^{k-1} 1/\left(i+1\right)^2\right) \leq m(2H_k + \zeta_1)$$

for some positive constant ζ_1, where H_k is the k-th Harmonic number.

A greedy assignment is obtained by making each client select the server with the smallest index among its permissible servers having the minimum number of clients. In the analysis we make use of sets of clients called *columns*. A client belongs to column col_i if, when it selects its server, it is the i-th client selecting that server. For example, clients $c_m^1, c_{m-1}^1, ...c_{m/2+1}^1$ select servers $s_1, ..., s_{m/2}$, respectively; each of them is the first client in its server, so they belong to col_1. Then, $c_{m/2}^1, ..., c_{m/4+1}^1$ select servers $s_1, ..., s_{m/4}$; they belong to col_2. We can verify that the set of servers selected by clients in col_{i+1} is subset of the set of servers selected by clients in col_i for $i = 1, ..., 2k-3$, that columns col_{2i-1} and col_{2i} contain clients of groups $g_1, ..., g_i$, and that $|col_{2i}| = \frac{m}{(i+1)^2}$ and $|col_{2i-1}| = \frac{m}{i(i+1)}$ for any $i = 1, ..., k-1$. So, for $i = 1, ..., 2k-3$, the number of servers receiving exactly i clients in the greedy assignment is $|col_i| - |col_{i+1}|$. We compute a lower bound on the cost gr of the greedy assignment by considering only the servers with at most $2k-4$ clients. We have that

$$gr \geq m\sum_{i=1}^{k-2}\left((2i-1)^2(|col_{2i-1}| - |col_{2i}|) + (2i)^2(|col_{2i}| - |col_{2i+1}|)\right)$$
$$= m\sum_{i=1}^{k-2}\left((2i-1)^2\left(\frac{1}{i(i+1)} - \frac{1}{(i+1)^2}\right) + (2i)^2\left(\frac{1}{(i+1)^2} - \frac{1}{(i+1)(i+2)}\right)\right)$$
$$\geq m\sum_{i=1}^{k-2}\left(\frac{8}{i+1} - \frac{20}{(i+1)^2}\right) \geq m(8H_k - \zeta_2)$$

for some positive constant ζ_2. We conclude that for any $\epsilon > 0$ and sufficiently large k and m, the competitiveness of the greedy assignment is at least $4 - \epsilon$. □

By slightly modifying the argument in the proof of Theorem 7 we can show that the lower bound holds for any deterministic online algorithm.

5 Extensions and Open Problems

We have also considered clients with non-negative weights. In the case of clients with weights, upper bounds of $\frac{3+\sqrt{5}}{2} \approx 2.618$ and $3 + 2\sqrt{2} \approx 5.8284$ for the price of anarchy of selfish load balancing and the competitiveness of greedy load balancing follow by the analysis of [4,7] for weighted linear congestion games and by the analysis of [3], respectively. We have shown that both bounds are tight. In particular, the second lower bound holds for greedy load balancing on identical servers. For selfish load balancing of weighted clients on identical servers, we can show a lower bound of $5/2$ on the price of anarchy. It is interesting to close the gap between this lower bound and the upper bound of $\frac{3+\sqrt{5}}{2}$ which has been proved for congestion games [4]. We believe that our lower bound is tight. Another interesting open problem is to compute tight bounds for the price of stability of weighted load balancing games. We have considered pure Nash equilibria of load balancing games. Our results hold or can be extended to hold for mixed and correlated equilibria [8] as well. There is also a small gap between 4 and 4.05505 for the competitiveness of greedy load balancing on identical servers. We believe that it can be further narrowed by extending our upper bound technique.

References

1. N. Alon, Y. Azar, G. J. Woeginger and T. Yadid. Approximation schemes for scheduling. In *Proc. of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '97)*, pp. 493-500, 1997.
2. A. Avidor, Y. Azar and J. Sgall. Ancient and new algorithms for load balancing in the L_p norm. *Algorithmica*, 29(3): 422-441, 2001.
3. B. Awerbuch, Y. Azar, E. F. Grove, M.-Y. Kao, P. Krishnan, and J. S. Vitter. Load balancing in the L_p norm. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pp. 383-391, 1995.
4. B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proc. of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 57-66, 2005.
5. Y. Azar and A. Epstein. Convex programming for scheduling unrelated parallel machines. In *Proc. of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 331-337, 2005.
6. A. K. Chandra and C. K. Wong. Worst-case analysis of a placement algorithm related to storage allocation. *SIAM Journal on Computing*, 4(3): 249-263, 1975.
7. G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 67-73, 2005.
8. G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proc. of the 13th Annual European Symposium on Algorithms (ESA '05)*, LNCS 3669, Springer, pp. 59-70, 2005.

9. R. A. Cody and E. G. Coffman. Record allocation for minimizing expected retrieval costs on drum-like storage devices. *Journal of the ACM*, 23(1): 103-115, 1976.
10. A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '02)*, pp. 413-420, 2002.
11. A. Fabrikant, C. Papadimitriou and K. Talwar. On the complexity of pure equilibria. In *Proc. of the 36th Annual ACM Symposium on Theory of Computing (STOC '04)*, pp. 604-612, 2004.
12. D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas and P. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. In *Proc. of the 29th International Colloquium on Automata, Languages and Programming (ICALP '02)*, LNCS 2380, Springer, pp. 123-134, 2002.
13. D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. In *Proc. of the 31st International Colloquium on Automata, Languages, and Programming (ICALP '04)*, LNCS 3142, Springer, pp. 593-605, 2004.
14. M. Gairing, T. Lücking, M. Mavronicolas and B. Monien. Computing Nash equilibria for scheduling on restricted parallel links. In *Proc. of the 36th Annual ACM Symposium on Theory of Computing (STOC '04)*, pp. 613-622, 2004.
15. E. Koutsoupias, M. Mavronicolas and P. Spirakis. Approximate equilibria and ball fusion. *Theory of Computing Systems*, 36(6): 683-693, 2003.
16. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS '99)*, LNCS 1563, Springer, pp. 404-413, 1999.
17. T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. A new model for selfish routing. In *Proc. of the 21st International Symposium on Theoretical Aspects of Computer Science (STACS '04)*, LNCS 2996, Springer, pp. 547-558, 2004.
18. M. Mavronicolas and P. Spirakis. The price of selfish routing. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC '01)*, pp. 510-519, 2001.
19. D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14: 124-143, 1996.
20. C. Papadimitriou. Algorithms, games and the internet. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC '01)*, pp. 749-753, 2001.
21. S. Phillips and J. Westbrook. Online load balancing and network flow. In *Proc. of the 25th Annual ACM Symposium on Theory of Computing (STOC '93)*, pp. 402-411, 1993.
22. R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2: 65-67, 1973.
23. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2): 236-259, 2002.
24. T. Roughgarden and E. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47(2): 389-403, 2004.
25. D. Shmoys, J. Wein and D. Williamson. Scheduling parallel machines on-line. *SIAM Journal on Computing*, 24(6): 1313-1331, 1995.
26. S. Suri, C. Tóth and Y. Zhou. Selfish load balancing and atomic congestion games. In *Proc. of the 16th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '04)*, pp. 188-195, 2004.