Convergence of Best-Response Dynamics in Games with Conflicting Congestion Effects

Michal Feldman[1] and Tami Tamir[2]

[1] Hebrew University of Jerusalem and Harvard University.
E-mail: mfeldman@huji.ac.il.
[2] School of Computer science, The Interdisciplinary Center, Herzliya, Israel.
E-mail: tami@idc.ac.il

Abstract. We study the model of resource allocation games with conflicting congestion effects that was introduced by Feldman and Tamir (2012). In this model, an agent's cost consists of its resource's load (which increases with congestion) and its share in the resource's activation cost (which decreases with congestion). The current work studies the convergence rate of best-response dynamics (BRD) in the case of homogeneous agents. Even within this simple setting, interesting phenomena arise. We show that, in contrast to standard congestion games with identical jobs and resources, the convergence rate of BRD under conflicting congestion effects might be super-linear in the number of jobs. Nevertheless, a specific form of BRD is proposed, which is guaranteed to converge in linear time.

1 Introduction

Resource allocation is considered to be a fundamental problem in algorithmic game theory, and has naturally been the subject of intensive research within this field. Most of the game-theoretic literature on resource allocation settings emphasizes either the negative or the positive congestion effects on the individual cost of an agent. The former approach assumes that the cost of a resource is some non-decreasing function of its load. This literature includes job scheduling and routing models [22, 10, 20]. In these cases an individual user will attempt to avoid sharing its resource with others as much as possible. The second approach, in stark contrast, assumes that a resource's cost is a decreasing function of its load. This is the case, for example, in network design and cost sharing connection games, in which each resource has some *activation cost*, which should be covered by its users [2, 6]. In these cases, an individual user wishes to share its resource with as many other users as possible in attempt to decrease its share in the cost.

In reality, most applications have cost functions that exhibit both negative and positive congestion effects. Accordingly, more practical models that integrate the two congestion effects into a unified cost function have been considered [1, 14, 9]. The present paper studies the resource allocation setting that is introduced by Feldman and Tamir [9], in which the individual cost of an agent is composed of two components, one that exhibits positive externalities, and the other that exhibits negative externalities. More specifically, every resource has some activation cost, that is shared among all the agents using it. The individual cost of an agent is the sum of its chosen resource's load (reflecting the negative externalities) and its share in the resource's activation cost (reflecting the positive externalities). This model is applicable to a large set of applications, including job scheduling, network routing, and network design settings. In all of these applications, users gain some benefits from sharing a resource with others, but also incur some costs from using a congestible resource.

The induced game, unlike its two "parent games," is not a potential game[3] when played by heterogeneous agents. Indeed, it has been shown in [9] that best-response dynamics (BRD) do not necessarily converge in this setting. Yet, in the special case where agents are identical, the induced game is a potential game; consequently, any BRD is guaranteed to converge to a Nash equilibrium [9]. The rate of the convergence, however, has been overlooked thus far. It is argued that the convergence rate is crucial for the Nash equilibrium hypothesis to hold; that is, it is more plausible that a Nash equilibrium will be reached if natural dynamics lead to such an outcome within a small number of moves.

In this paper, we study the convergence rate of BRD in a job scheduling game with conflicting congestion effects and identical agents.

1.1 Our results

It is fairly easy to see that for unit-size jobs, convergence to a Nash equilibrium is linear in the number of jobs in both of the "parent" models; namely, if the the cost function equals the resource's load or if it equals the job's share in the resource's activation cost. We find that if the cost function takes both components into consideration, the convergence rate might be super-linear. We then introduce a specific form of BRD, referred as *max-cost*, where the job that incurs the highest cost is the one to perform its best move. The motivation behind this BRD is clear; it seems reasonable that the job that incurs the highest cost has the strongest incentive to improve its state. For this special case, the linear convergence rate is recovered.

1.2 Related work

A lot of research has been conducted in the analysis of job-scheduling applications using a game-theoretic approach. One branch emphasizes the *mechanism design* approach, and seeks for *truthful* mechanisms (see the seminal paper by [17]). The current work belongs to the second branch, which treats the jobs as players who choose the machine to run on. The questions that are commonly analyzed under this approach are Nash equilibrium existence, the convergence of best-response dynamics to a Nash equilibrium, and the inefficiency of Nash equilibria (quantified mainly by the price of anarchy [15, 18] and price of stability [2] measures).

[3] Potential games have been introduced by [16].

It is well known that every congestion game is a potential game [19, 16], and therefore admits a pure Nash equilibrium, and every best-response dynamics converges to a pure Nash equilibrium. However, the convergence time may, in general, be exponentially long [1, 8, 21]. This observation has led to a large amount of work that identified special classes of congestion games, where best-response dynamics converge to a Nash equilibrium in polynomial time or even linear time. This agenda has been the focus of [7, 11] in a setting with negative congestion effects, and was also studied in a setting of positive congestion effects [2]. In particular, it has been shown that it takes at most n steps (where n is the number of users) to converge to a Nash equilibrium if the network is composed of parallel links [7], and this result has been later extended to extension-parallel networks [11]. For resource selection games (i.e., where feasible strategies are composed of singletons), it has been shown in [13] that better-response dynamics converge within at most mn^2 steps for general cost functions (where m and n are the number of resources and users, respectively). In addition to standard better- and best-response dynamics, a few variants have been explored. One example is the study of the convergence rate of α-Nash dynamics to an approximate Nash equilibrium [5] and to an approximate optimal solutions [3]. Also, the robustness of best-response convergence to altruistic agents has been studied in [12], where it has been shown that best-response dynamics may cycle as a result of altruism.

More recently, congestion models with simultaneous negative and positive congestion effects have been considered. The model introduced by Johari and Kumar [14] considers a system with identical users and a single server, where the negative and positive congestion effects are modeled through fairly general increasing and decreasing cost functions, respectively. The model introduced by Feldman and Tamir [9] considers a specific form of negative and positive congestion effects with multiple machines. This model was studied recently also by Chen and Gürel [4]. For the special case of unit-size jobs, a more general model of conflicting congestion effects in routing games has been introduced by Anshelevich *et al.*, where the cost incurred by an agent is the sum of the delay on its chosen path and the setup cost on that path. Our model can be seen as a special case of this model, where the network is composed of parallel links and the setup cost is determined through the cost-sharing rule.

2 Model and Preliminaries

We consider a *job-scheduling* setting with identical machines and identical (unit-size) jobs. There is a set of machines $M = \{M_1, M_2, \ldots\}$ of unlimited size,[4] each associated with an *activation cost*, B. An instance of our problem is given as a tuple (n, B), where n denotes the number of jobs. An assignment method produces an assignment $s = (s_1, \ldots, s_n)$ of jobs into machines, where $s_j \in M$ denotes the machine to which job j is assigned. We use the terms assignment,

[4] Although we assume an unlimited number of machines, the number of machines will clearly be less than n.

schedule, and profile interchangeably. The load of a machine M_i in a schedule s, denoted $L_i(s)$, is the number of jobs assigned to M_i in s.

Given a job-scheduling setting and an activation cost B, a *job-scheduling game* is induced in which the set of players is the set of jobs, and the action space of each player j, is the set of machines. The cost function of job j in a given schedule is the sum of two components: the load on j's machine and j's share in the machine's activation cost. It is assumed that the activation cost B is shared equally between all the jobs that use a particular machine. That is, given a profile s in which $s_j = M_i$, the cost of job j is

$$c_j(s) = L_i(s) + \frac{B}{L_i(s)}.$$

We denote the cost of a job that is assigned to a machine with load x by $c(x)$, where $c(x) = x + \frac{B}{x}$. It can be easily verified that the cost function exhibits the following structure.

Observation 1 *The function $c(x) = x + B/x$ for $x > 0$ attains its minimum at $x = \sqrt{B}$, is decreasing for $x \in (0, \sqrt{B})$, and increasing for $x > \sqrt{B}$.*

Practically, the input to the cost function in our setting is an integral value. If B is a perfect square, then the integral load achieving the minimal cost is exactly $\sqrt{B}$. For example, if $B = 100$, then being assigned to a machine with load 10 is optimal. In general, however, the optimal integral load (i.e., the load that minimizes the cost function) may be either $\left\lfloor \sqrt{B} \right\rfloor$ or $\left\lceil \sqrt{B} \right\rceil$, and for some values of B it may be both. For example, if $B = 12$ then both 3 and 4 are optimal loads, as $c(3) = c(4) = 12$. We denote an optimal load by $\ell^* = \ell^*(B)$. Assuming a unique integral optimal load, it is easy to verify that the cost function is decreasing for $x \in [1, \ell^*]$ and increasing for $x \geq \ell^*$. For two optimal integral loads, $\ell^* - 1$ and ℓ^*, the cost function is decreasing for $x \in [1, \ell^* - 1]$ and increasing for $x \geq \ell^*$.

An assignment $s \in S$ is a *pure Nash equilibrium* (NE) if no job $j \in N$ can benefit from unilaterally deviating from its machine to another machine (possibly a new machine). In our game, this implies that for every job j assigned to M_i and every $i' \neq i$, it holds that $c(L_i(s)) \leq \min(c(1), c(L_{i'}(s) + 1))$.

3 Convergence of Best-Response Dynamics

Best-Response Dynamics (BRD) is a local search method where in each step some player is chosen and plays its best-response strategy, given the strategies of the others. In systems where the agents always reach a Nash equilibrium after repeatedly performing improvement steps, the notion of a pure Nash equilibrium is well justified. In order to better understand the behavior of these dynamics, this section explores the convergence rate of best-response dynamics into a pure NE.

In the general case, in which jobs have arbitrary lengths and the activation cost of a machine is shared by the jobs proportionally to their length, BRD is not guaranteed to converge to a NE [9]. In contrast, if the jobs are identical, then the induced game is equivalent to a congestion game with n resources [19]. One can easily verify that the function $\Phi(s) = \sum_i \left(B \cdot H_{\ell_i} + \frac{1}{2}\ell_i^2\right)$, where ℓ_i denotes the number of jobs on machine i, $H_0 = 0$, and $H_k = 1 + 1/2 + \ldots + 1/k$, is a potential function for the game. Convergence to a NE is guaranteed in potential games, but the convergence time might be exponential.

In this section we study the convergence time of BRD of unit-length jobs. We show that the convergence of BRD in general might take $\Omega(n \log \frac{n}{B})$ moves, and propose a specific form of BRD that ensures convergence within a linear number of moves. Specifically,

Max-cost BRD: At every time step, a job that incurs the highest cost among those who can benefit from migration, is chosen to perform its best-response move (where ties are broken arbitrarily).

The analysis of the convergence rate of BRD and max-cost BRD (MC-BRD hereafter) is quite complicated and requires several preparations and terminology. Recall that all jobs assigned to a machine with load x incur the same cost $c(x) = x + B/x$. We denote by ℓ^* a load achieving minimal cost. By Observation 1, ℓ^* may be either $\left\lfloor \sqrt{B} \right\rfloor$ or $\left\lceil \sqrt{B} \right\rceil$, and for some values of B it may be both. For simplicity, throughout this section we assume a unique optimal load. All the results hold also for the case of two optimal loads, where minor straightforward modifications are required in the proofs.

We denote by ℓ_i^t the load of machine M_i at time t, i.e., *before* the migration of iteration t takes place. A machine that has load at least (respectively, smaller than) ℓ^* is said to be a *high* (*low*) machine.

We observe that if at some iteration a job migrates to a low machine, then in subsequent iterations that machine will attract more jobs up to load at least ℓ^*. Indeed, since $c(\ell + 1) < c(\ell)$ for $\ell < \ell^*$, a low best-response machine continues to be a best response until it is filled up to load at least ℓ^*. Formally,

Observation 2 *If at some iteration t there is a migration to a low machine M_i such that $\ell_i^t = \ell^* - x$ for some $x > 0$, then the following $x - 1$ iterations will involve migrations to M_i.*[5]

Properties of MC-BRD. By the design of the MC-BRD process and as a direct corollary of Observation 1, every migration in the MC-BRD process is from either the lowest or the highest machine into either the lowest-high or the highest-low machine (see Figure 3).

[5] It is possible that the system reaches a NE and the BRD process terminates before $x - 1$ iterations are performed.

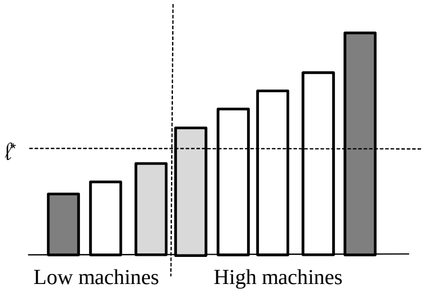

Fig. 1. MC-BRD process. Every migration is from one of the extreme machines into one of the middle grey machines.

Since all jobs on a particular machine share the same cost, the MC-BRD process can be described as if it acts on machines rather than on jobs. Specifically, in every iteration t, one job migrates from machine M_i to machine $M_k, k \neq i$, where (i) $c(\ell_k^t + 1)$ is minimal, (ii) $c(\ell_k^t + 1) < c(\ell_i^t)$, and (iii) $c(\ell_i^t)$ is maximal among all the machines from which a beneficial migration exists. While the MC-BRD process does not specify which job is migrating from M_i, for simplicity we will assume a LIFO (last in first out) job selection rule. Specifically, the job that entered M_i last is the one to migrate. If all jobs on M_i were assigned to it in the initial configuration, then an arbitrary job is selected. Since the BRD-process can be characterized by the load-vector of the machines in every time step, the number of iterations is independent of the job-selection rule. Consequently, our analysis of the convergence rate of MC-BRD applied with a LIFO job-selection rule is valid for any MC-BRD process.

Note that M_i, the machine from which a job is selected to migrate in iteration t, is not necessarily the machine for which $c(\ell_i^t)$ is maximal. For example, suppose that $B = 100$ and there are two active machines, a low one with load 3, and a high one with load 33. It is easy to verify that $c(4) < c(33) < c(3) < c(34)$. In this case, $c(3)$ is the maximal cost, but jobs on the low machine have no beneficial move (since $c(34) > c(3)$). On the other hand, jobs on the high machine wish to migrate to the low one (since $c(4) < c(33)$). Thus, the high machine is the one selected by MC-BRD to perform a migration, although the low machine is the one incurring max-cost. Clearly, such a case can only occur if the machine that incurs the max-cost is itself the best-response machines, as summarized in the following observation.

Observation 3 *If at time t the machine M_i that incurs max-cost is not the one from which a job is selected to migrate in MC-BRD, then $c(\ell_i^t + 1)$ is the best-response, in particular, this implies that M_i is low.*

We next observe that in MC-BRD, if at some iteration a job leaves some low machine, then in the following iterations all the jobs assigned to that machine will leave it one by one until the machine empties out. To see this, note that $c(\ell - 1) > c(\ell)$ for $\ell < \ell^*$; thus, if a low machine incurs the highest cost, it continues to incur the highest cost after its load decreases. It remains to show that if a beneficial move out of M_i exists when it has load $\ell < \ell^*$, then it is also beneficial to leave M_i when it has load $\ell - 1$. This is ensured by Observation 3. Specifically, if it is not beneficial, then $c(\ell)$ is the cost of the best-response machine. But this is impossible since $c(\ell)$ was the max-cost in the previous iteration.

Observation 4 *If at some iteration t there is a migration from a low machine M_i such that $\ell_i^t = \ell^* - x$ for some $x > 0$, then the following $\ell^* - x - 1$ iterations will involve migrations from M_i.*

We are now ready to state the bound on the convergence rate of MC-BRD.

Theorem 1. *For every job scheduling game with identical jobs, every MC-BRD process converges to a NE within at most* $\max\{\frac{3n}{2} - 3, n - 1\}$ *steps.*

Proof. First note that at least one machine that is active in the initial configuration will remain active in the final configuration. Thus, at least one job, located on this machine, does not migrate at all. If every other job migrates at most once then clearly there are at most $n - 1 \leq \max\{\frac{3n}{2} - 3, n - 1\}$ iterations. We show that it is possible for some jobs to migrate twice, but not more than twice. Moreover, we show that a second migration of a job might happen only towards the end of the BRD process and that the number of jobs that migrate twice is limited.

If no job migrates more that once, then the claim is valid. Assume that at least one job migrates more than once. Let j be the first job to migrate for the second time. Assume that j migrates from M_1 to M_2 at time t and from M_2 to M_3 at time $t' > t$. By our LIFO assumption, $\ell_2(t') = \ell_2(t) + 1$.

Claim 2 *At time t', all the machines, except possibly for M_3, have load either $\ell_2(t')$ or $\ell_2(t') - 1$.*

Proof. We first show that there are no low machines at time t'. Assume by way of contradiction that M_0 is low at time t'. The load on M_0 remains the same between time t and t'; that is, $\ell_0(t') = \ell_0(t)$ because M_0 is not the current best response and according to Observation 2, if it were the best response at some time point then it would keep attracting jobs until it was not low anymore. We next show that M_0 is not the max-cost machine at time t'. To see this, note that if it were a max-cost machine, then, by Observation 3, it should be either the machine from where a job migrates, or the machine to where a job migrates. But these two machines are M_2 and M_3, respectively. Thus, at time t', M_2 is the max-cost machine. Therefore $c(\ell_0(t')) \leq c(\ell_2(t'))$. Also $c(\ell_0(t')) = c(\ell_0(t)) > c(\ell_0(t) + 1)$, because M_0 is low. However, at time t, j preferred joining M_2 having cost $c(\ell_2(t'))$ to joining M_0 having cost $c(\ell_0(t) + 1)$, a contradiction.

It remains to consider high machines at time t'. As M_2 is the max-cost machine, there are no machines with load larger than $\ell_2(t')$ at time t'. We show that no high machine (except maybe for M_3) has load less than $\ell_2(t')-1=\ell_2(t)$ at time t'. Assume by way of contradiction that there is a machine, $M_4 \neq M_3$, with load less than $\ell_2(t')-1$. The load of M_4 did not decrease between times t and t' because in order to get below $\ell_2(t')-1$ it must be that the max-cost is achieved by load $\ell_2(t')-1$, while we know that the load on M_2 is higher during this whole time interval. The same argument can be applied to M_3 (we know that $\ell_3(t') < \ell_2(t')-1$ as otherwise the second migration of j would not be beneficial). Therefore, the load on both M_3 and M_4 does not decrease between times t and t'. If M_3 or M_4 were high already at time t, then j should have preferred one of them to M_2. Thus, both M_3 and M_4 were low at time t and by the choice of j at time t, we know that $c(\ell_4(t)+1) \geq c(\ell_2(t))$ as well as $c(\ell_3(t)+1) \geq c(\ell_2(t))$. Assume w.l.o.g. that $\ell_3(t) \leq \ell_4(t)$; that is, M_4 becomes active before M_3. Since at time t' both are high, it must be that M_3 became a best-response between t and t'. However, the load on M_4 during this time interval was always less than $\ell_2(t')-1$, and since $c(\ell_3(t)+1) \geq c(\ell_2(t)) > c(\ell_4(t''))$ for every time t'' between t and t', we get a contradiction to the choice of M_3 as a best-response during this time interval.

□

Denote by N_0 the number of jobs that do not migrate at all, by N_1 the number of jobs migrating once, and by N_2 the number of jobs migrating more than once. Given that at time t', all machines except for M_3 have loads $\ell_2(t')$ or $\ell_2(t')-1$, the only remaining migrations following time t' are to M_3. Therefore, jobs migrating more than once can only migrate twice and their second migration is to M_3. Thus, N_2 is in fact the number of jobs migrating *twice*. The following claim bounds N_2.

Claim 3 $N_2 \leq \frac{n}{2}-2$.

Proof. By the above discussion, the only remaining migrations following time t' are to M_3, and a NE is reached when the load is balanced (all machines have load x or $x-1$ for some x). Note that all machines except, possibly, M_3 must be high at time t', since if a machine $M' \neq M_3$ is low, then by Observation 2, it keeps attracting jobs. Also, a load on a high machine never goes below ℓ^*, therefore, when a NE is reached, there are at least two active machines, M_2 and M_3. Also, $\ell_3(t') > 1$, as if $\ell_3(t')=1$ and M_3 is the best-response at time t', then it should have been a best-response also at time t (by Observations 2 and 4, the load on a low machine could not decrease before it becomes a best-response), contradicting the choice of j at time t. Therefore, the maximal number of migrations following time t' is obtained when the only active machines at time t' are M_2 and M_3 and their loads are $n-2$ and 2, respectively. The number of migrations required to reach a balanced load is $n-2-\frac{n}{2}=\frac{n}{2}-2$ if n is even, and $n-2-\frac{n+1}{2} < \frac{n}{2}-2$ if n is odd.

□

As argued in the beginning of this proof, $N_0 > 0$. Therefore, $N_1+N_2 \leq n-1$. It follows that the total number of migrations is $0 \cdot N_0 + 1 \cdot N_1 + 2 \cdot N_2 \leq$

$(N_1 + N_2) + N_2 \leq n - 1 + n/2 - 2 = \frac{3n}{2} - 3$, as promised, where the last inequality follows by Claim 3. □

The above result is almost tight. Consider an instance with $n = \frac{B}{2} + 2$ jobs. Initially, the jobs are assigned on $m = n$ machines, a single job on each machine. In a possible MC-BRD process, one of the machines attracts $\frac{B}{2}-1$ jobs from $\frac{B}{2}-1$ other machines. After $\frac{B}{2}-1$ moves, there are three active machines and the load vector is $\langle \frac{B}{2}, 1, 1\rangle$. Note that for any $B > 2$ it holds that $c(1) > c(\frac{B}{2}+1) > c(2)$. Thus, at this point, one of the low machines becomes the best response and the other achieves the max-cost. This leads to load vector $\langle \frac{B}{2}, 2\rangle$. The next $\frac{B}{4} - 1$ moves will balance the load on the two machines. The total number of migrations is $\frac{B}{2} - 1 + 1 + \frac{B}{4} - 1 = \frac{3B}{4} - 1 = \frac{3n}{2} - 4$.

In contrast to MC-BRD, the convergence time of arbitrary BRD, in which the moving job is not necessarily the one having maximal cost, might not be linear in n.

Theorem 4. *There exists a job scheduling game with identical jobs and a BRD process such that the convergence time to a NE is $\Omega(n \log \frac{n}{B})$.*

Proof. We describe an instance and a possible BRD convergence of this instance. Given m, let $n = mB/2$. Consider an initial assignment in which one machine, M_0, is assigned $n - (m-1)$ jobs, and $m-1$ machines, denoted $M_1, ..., M_{m-1}$, are assigned a single job each. A possible BRD process can be described as a sequence of $m-1$ phases. Phase k for $k = 1, \ldots, m-1$ begins when machine M_k is the best-response machine for the first time (i.e., its load increases from 1 to 2). Each phase consists of two stages: (i) *balancing* - after which the loads on machines $M_1, \ldots, M_k$ are equal (up to a gap of 1), and (ii) *filling* - after which the loads on machines $M_1, \ldots, M_k$ are all $B/2$. We demonstrate such a slow BRD process with 5 machines and $B = 100$ in Figure 3. A detailed description of the stages is deferred to the full version.

Claim 5 *The number of steps in the above BRD process is $\Omega(n \log \frac{n}{B})$.*

Proof. **Proof of Claim 5**: For simplicity, in the following analysis, we ignore floors and ceilings and the possible gap of 1 between the machines' load after the balancing stage. One can easily verify that in a closer analysis, the total number of migrations might be reduced by at most m. In general, for arbitrary m, B, the balancing stage in the first phase is empty. The filling stage consists of $B/2-1$ migrations from M_0 to M_1. After this stage, M_2 becomes a best-response machine (an added job will incur a cost of $B/2+2$, which is the current cost of a job on M_1). Thus, the second phase begins with a migration from M_0 to M_2. In the balancing stage jobs are migrating from M_1 to M_2 until they both have load $(B/2+2)/2$. In the filling stage, the best-response alternates between the machines M_1, M_2, until both have loads $B/2$, and M_3 becomes the new best-response machine. In general, the balancing stage of phase k can be described in a recursive manner: first, the balancing stage of phase $k-1$ (on machines $M_1, \ldots, M_{k-1}$) is repeated on machines $M_2, \ldots, M_k$; this results in balancing

	M_0	M_1	M_2	M_3	M_4	
Phase 1	246	1	1	1	1	
	197	50	1	1	1	(49)
Phase 2	196	50	2	1	1	(1)
	196	26	26	1	1	(24)
	148	50	50	1	1	(48)
Phase 3	147	50	50	2	1	(1)
	147	50	26	26	1	(24)
	147	34	34	34	1	(16)
	99	50	50	50	1	(48)
Phase 4	98	50	50	50	2	(1)
	98	50	50	26	26	(24)
	98	50	34	34	34	(16)
	98	38	38	38	38	(12)
	50	50	50	50	50	(48)

Fig. 2. A slow BRD process. Each line gives the machines' load vector at some time point. The numbers in parentheses show the number of migrations performed since the previously presented time point.

the loads on $M_2, \ldots, M_k$. Next, there are additional migrations from M_1 to $M_2, \ldots, M_k$. Specifically, in the last part of the balancing stage of phase k, we move from load vector (on $M_1, \ldots, M_k$)

$$\langle \frac{B}{2}, \frac{k-2}{k-1} \cdot \frac{B}{2}, \frac{k-2}{k-1} \cdot \frac{B}{2}, \ldots, \frac{k-2}{k-1} \cdot \frac{B}{2} \rangle$$

to load vector $\langle \frac{k-1}{k} \cdot \frac{B}{2}, \ldots, \frac{k-1}{k} \cdot \frac{B}{2} \rangle$. This last part consists of $\frac{B}{2k}$ migrations.

In the filling stage of phase k, the load on the k machines $M_1, \ldots M_k$ is increased from $\frac{k-1}{k}\frac{B}{2}$ to $\frac{B}{2}$, which takes $\frac{B}{2}$ migrations (in fact, the exact number of migrations is $\frac{B}{2} - 2$, but the overall distortion in the total counting is at most $2(m-1)$, which is negligible).

Denote by $b_k(f_k)$ the number of migrations in the balancing (filling) stage of phase k. According to the above description, for every $k > 1$ it holds that $b(k) = b_{k-1} + \frac{B}{2k}$ and $f(k) = \frac{B}{2}$. The total number of migrations in phase k is $b_k + f_k = b_{k-1} + \frac{B}{2k} + \frac{B}{2}$. Also, for $k = 1$, we have $b(1) = 0, f(1) = \frac{B}{2}$. Summing up all $m-1$ phases, we get that the total number of migrations until convergence is

$$\begin{aligned}\sum_{k=1}^{m-1}(b_k + f_k) &= \sum_{k=1}^{m-1}(b_{k-1} + \frac{B}{2k} + \frac{B}{2}) = \frac{B}{2}(m-1) + \sum_{k=1}^{m-1}(b_{k-1} + \frac{B}{2k}) \\ &= \frac{B}{2}(m-1) + \frac{B}{2}(\frac{1}{2}(m-2) + \frac{1}{3}(m-3) + \ldots + \frac{1}{m-1}(m-(m-1))) \\ &= \frac{B}{2}(m-1+\frac{m}{2}-1+\frac{m}{3}-1+\ldots+\frac{m}{m-1}-1) = \frac{B}{2}(m\log(m-1)-(m-1)).\end{aligned}$$

Since $m = \frac{2n}{B}$, we get that the total number of migrations is

$$\frac{B}{2}(\frac{2n}{B}\log(\frac{2n}{B}-1)-(\frac{2n}{B}-1)) = \Omega(n\log\frac{n}{B}).$$

□

□

While the convergence rate of general BRD is super-linear, the following theorem establishes an upper bound of n^2. Closing the gap remains an open problem.

Theorem 6. *For every job scheduling game with identical jobs, every BRD process converges to a NE within at most* n^2 *steps.*

Proof. Let m_{low} be the number of low machines in the initial configuration. Among these low machines there are m_1 machines that will be emptied along the BRD process, and m_2 machines that will form the best-response during the process and (by Observation 2) end up high. Partition the BRD process into phases. Phase 0 (which might be empty) consists of migrations as long as the best response is a high machine, these migrations might be among high machines or from low machines that are emptied into high machines. Index the m_2 most-loaded low machines in non-increasing order of initial load. For $i > 0$, phase i begins when the i-th low machine form the best-response for the first time.

Therefore, the total number of migrations is bounded by the number of phases, m_2+1, multiplied by the number of migrations per phase. Since $m_2 < n$, showing that the number of migrations in every phase is bounded by n establishes an upper bound of n^2, as desired. We next prove that the number of migrations in every phase is bounded by n.

For $i > 0$, phase i begins with migrations to the i-th low machine till it becomes high. This machine will never become low again, because the job at height ℓ^* will never leave it, thus, jobs migrating to the i-th machine, as long as it is low, will not migrate again (in this, and in later phases).

The following migrations (and all the migrations in phase 0), are migrations to high machines – among high machines, or from low machines that are emptied. During such migrations, the best-response value can only increase, therefore, if job j is migrating to load ℓ, the best-response value will always be at least $c(\ell+1)$ and by the LIFO property, j will not leave the machine as long as the best-response value is not strictly lower.

□

It is interesting to compare our results to those established for the standard model that considers only the negative congestion effects (i.e., where a job's cost is simply the load of its chosen machine). It has been shown by [7] that if the order of the jobs performing their best-response moves is determined according to their lengths (i.e., longer job first), then best-response dynamics reaches a pure Nash equilibrium within at most n improvement steps. In contrast, if the jobs move in an arbitrary order, then convergence to a Nash equilibrium might take an exponential number of steps. These results imply that for the special case of equal-length jobs, convergence occurs within at most n steps. Our results provide evidence that when there are conflicting congestion effects, it might take longer to reach a Nash equilibrium. Nevertheless, for the special case of max-cost BRD, the consideration of positive congestion effects (through activation costs) does not lead to a longer convergence time.

References

1. H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. In *Proc. of the 47th IEEE Symp. on Foundations of Computer Science* (FOCS), pp. 613 622 (2006).
2. E. Anshelevich, A. Dasgupta, J. M. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. of the 45th IEEE Symp. on Foundations of Computer Science* (FOCS), pp. 295–304 (2004).
3. B. Awerbuch, Y. Azar, A. Epstein, V.S. Mirrokni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. In *Proc. of the 9th ACM conference on Electronic Commerce* (ACMEC), pp 264–273 (2008).
4. B. Chen and S. Gürel. Efficiency analysis of load balancing games with and without activation costs. *Journal of Scheduling*, 15(2) (2011).
5. S. Chien and A. Sinclair. Convergence to approximate Nash equilibria in congestion games. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 169–178 (2007).
6. A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost sharing connection games. In *the 8th ACM Conference on Electronic Commerce* (ACMEC) (2007).
7. E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *the 30th International Colloquium on Automata, Languages and Programming* (ICALP), pp. 502–513 (2003).
8. A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In em Proc. of the 36th ACM Symp. on Theory of Computing (STOC), pp. 604612 (2004).
9. M. Feldman and T. Tamir. Conflicting congestion effects in resource allocation games. *Operations Research (forthcoming)* (2012).
10. D. Fotakis, S Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *29th International Colloquium on Automata, Languages and Programming* (ICALP), pp. 510–519 (2002).
11. D. Fotakis. Congestion games with linearly independent paths: convergence time and price of anarchy. In *Theory Comput. Syst.*, 47(1):113–136 (2010).
12. M. Hoefer and A. Skopalik. Stability and convergence in selfish scheduling with altruistic agents. *Proceedings of 5th Workshop on Internet and Network Economics* (WINE) (2009).
13. S. Ieong, R.Mcgrew, E. Nudelman, Y. Shoham and Q. Sun, Fast and Compact: A Simple Class of Congestion Games. *In Proc. of the 20th Nat. Conference on Artificial Intelligence* (AAAI), pages 489–494 (2005).
14. R. Johari and S. Kumar. Strong nash equilibria in games with the lexicographical improvement property. In *Proceedings of the 11th ACM conference on Electronic Commerce* (2010).

15. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer Science Review*,3(2): 65-69 (2009).
16. D. Monderer and L.S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143 (1996).
17. N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166196 (2001).
18. C. Papadimitriou. Algorithms, games, and the Internet. *Proceedings of 33rd ACM Symposium on Theory of Computing* (STOC), 749–753 (2001).
19. R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67 (1973).
20. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236 – 259 (2002).
21. V. Syrgkanis. The complexity of equilibria in cost sharing games. *In Proceedings of the 6th international conference on Internet and network economics* (WINE), pages 366–377 (2010).
22. B. Vöcking. *In Noam Nisan, Tim Roughgarden, Eva Tardos and Vijay Vazirani, eds., Algorithmic Game Theory. Chapter 20: Selfish load balancing.* Cambridge University Press (2007).