Stability and Convergence in Selfish Scheduling with Altruistic Agents

Martin Hoefer * Alexander Skopalik †

October 5, 2009

Abstract

In this paper we consider altruism, a phenomenon widely observed in nature and practical applications, in the prominent model of selfish load balancing with coordination mechanisms. Our model of altruistic behavior follows recent work by assuming that agent incentives are a trade-off between selfish and social objectives. In particular, we assume agents optimize a linear combination of personal delay of a strategy and the resulting social cost. Our results show that even in very simple cases a variety of standard coordination mechanisms are not robust against altruistic behavior, as pure Nash equilibria are absent or better response dynamics cycle. In contrast, we show that a recently introduced TIME-SHARING policy yields a potential game even for partially altruistic agents. In addition, for this policy a Nash equilibrium can be computed in polynomial time. In this way our work provides new insights on the robustness of coordination mechanisms. On a more fundamental level, our results highlight the limitations of stability and convergence when altruistic agents are introduced into games with weighted and lexicographical potential functions.

1 Introduction

One of the most fundamental scenarios in algorithmic game theory are selfish load balancing models [27]. Since the seminal paper by Koutsoupias and Papadimitriou [22] they have attracted a large amount of interest [5,6,11,13,15,17,20,21]. The reasons are central applications in distributed processing, conceptual simplicity, and that they contain in a nutshell many prominent challenges in designing distributed systems for selfish participants. A fundamental assumption in the vast majority of previous work is that all agents are selfish. Their goals are restricted to optimizing their direct personal delay. However, this assumption has been repeatedly questioned by economists and psychologists. In experiments it has been observed that participants' behavior can be quite complex and contradictive to selfishness [23,24]. Various explanations have been given for this phenomenon, e.g. senses of fairness [10], reciprocity among agents [18], or spite and altruism [8,24].

*Dept. of Computer Science, RWTH Aachen University, Germany. `mhoefer@cs.rwth-aachen.de`. Supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD) and by DFG through UMIC Research Centre at RWTH Aachen University.

†Dept. of Computer Science, RWTH Aachen University, Germany. `skopalik@cs.rwth-aachen.de`. Supported in part by the German Israeli Foundation (GIF) under contract 877/05.

In this paper, we consider altruism in non-cooperative load balancing games. It is natural to study the effects of an important phenomenon like altruism in a core scenario of algorithmic game theory. Our model of altruism is similar to the one used recently in [4, 19] and related to the study of coalitional stability concepts [12,14], although we do not require agent cooperation in our model. Instead, each agent i is assumed to be partly selfish and partly altruistic. Her incentive is to optimize a linear combination of personal cost and social cost, given by the sum of cost values of all agents. The strength of altruism of each agent i is captured by her altruism level $\beta_i \in [0, 1]$, where $\beta_i = 0$ results in a purely selfish and $\beta_i = 1$ in a purely altruistic agent.

We consider altruistic agents in various types of scheduling games resulting from coordination mechanisms [5]. In these games agents are tasks, and each task chooses to allocate one out of several machines. For a machine the coordination mechanism is a local scheduling policy that determines the schedule of the tasks which choose to allocate the machine. Quite a number of policies have been proposed [1, 3, 5, 7, 21], mostly with the objective to minimize the price of anarchy [22] for makespan social cost. In addition to modelling a natural phenomenon, altruistic agents yield a measure of robustness for these mechanisms. Our results provide an interesting distinction between the studied policies in terms of stability and convergence properties. In addition, they also shed some light on an interesting and more fundamental aspect. Previously we studied altruists in atomic congestion games [19], which have an exact potential function. For atomic games, there are a number of special cases, in which a potential function argument guarantees existence of pure Nash equilibria and convergence of better response dynamics even for games with altruists. These cases include games with linear delay functions, or β-uniform agents that all have the same altruism level β. In this paper we analyze altruism in arguably the most basic games with weighted and lexicographical potential functions, and we expect our results to hold similarly e.g. for other coordination mechanisms based on lexicographical improvement arguments [3]. After addition of altruists, potential functions are largely absent here, even for identical machines or β-uniform agents. In contrast, the very positive results for the Time-Sharing policy rely on the existence of an exact potential for the original game and the construction is very similar to [19]. It is an interesting open problem to see if there is a connection between these cases, or if a general characterization of the existence of potential functions under altruistic behavior can be derived.

1.1 Our Results

We study altruistic agents with four different coordination mechanisms. At first in Section 3 we consider the classic Makespan policy [22], which is probably the most widely studied policy and yields a weighted potential function. For altruistic agents we show that this favorable property breaks down. There are games without pure Nash equilibria, and deciding this property for a game is NP-hard, even on identical machines. In Section 4 we study simple ordering based policies like Shortest-First and Longest-First that yield a lexicographic potential for non-atruistic users [21]. While for Shortest-First on identical machines existence of a pure Nash equilibrium is guaranteed even for arbitrary altruism levels, the resulting games are no potential games as better response dynamics might cycle. Even if they converge, they can take exponentially long to reach a pure Nash equilibrium. The latter result can be generalized to hold for every policy based on global ordering. For Longest-First we addionally show that there are games without pure Nash equilibria. Finally, in Section 5 we consider the TIME-SHARING policy introduced in [7]. While the policy is somewhat similar to MAKESPAN, the results are completely different. For this policy we show the existence of a potential function, even for arbitrary altruism levels and unrelated machines. Thus, existence of pure Nash equilibria and convergence of better response dynamics is always guaranteed. In addition, we show how to compute a Nash equilibrium in polynomial time.

2 Scheduling with Coordination Mechanisms

We consider scheduling games with coordination mechanisms [5]. A scheduling game G consists if a set N of n agents and a set M of m machines. Each agent $i \in N$ is a *task* and picks as a strategy the machine it wants to be processed on. In the case of identical machines, task i has processing time p_i on every machine. In case of related machines there is a speed factor s_j for machine j, and the processing time of i on j becomes p_i/s_j. For unrelated machines there is a separate processing time p_{ij} for every task i and machine j.

The strategy choices of the tasks result in a schedule $s : N \to M$, an assignment of every task to exactly one machine. On each machine there is a *coordination mechanism*, i.e. a sequencing policy that sequences the tasks and assigns starting and finishing time for each task. We assume here that tasks must be processed non-preemptively, but depending on the coordination mechanism the machine might be able to process multiple tasks simultaneously. For a given sequencing policy SP on the machines, we define the social cost of a schedule as $c^{SP}(s) = \sum_j f_j(s)$, where $f_j(s)$ is finishing time of task j in schedule s. To model altruism we use for each task i the *altruism level* β_i [4, 19]. If $\beta_i > 0$, we call task i an altruist. If $\beta_i = 1$ we call task i a pure altruist, if $\beta_i = 0$ we call him an egoist. The individual cost of a task i incorporates the effect on the social cost: $c_i^{SP}(s) = \beta_i c^{SP}(s) + (1 - \beta_i) f_i(s) = f_i(s) + \beta_i \sum_{j \neq i} f_j(s)$. A pure Nash equilibrium of the game is a schedule, in which no task can decrease his individual cost with a unilateral strategy change. Clearly, if all tasks are pure altruists, then every game on unrelated machines has a pure Nash equilibrium and every sequential better reponse dynamics converges.

3 Makespan and Random Policies

The first and most widely studied policy is the MAKESPAN policy [22], in which all tasks on one machine are processed simultaneously and finish at the same time. In the RANDOM policy [21] tasks are ordered in a random order and then processed consecutively in this order. Obviously, RANDOM and MAKESPAN are equivalent in terms of (expected) finishing times on identical and related machines.

MAKESPAN induces a weighted potential game. Let $\ell_j = \sum_{i\,:\,s_i=j} p_{ij}$ be the load of tasks choosing machine j. For identical machines the weighted potential is $\Phi(s) = \sum_{j=1}^{m} \ell_j^2$. For a task i we have $c_i^{MS}(s) - c_i^{MS}(s_i', s_{-i}) = \frac{1}{p_i}(\Phi(s) - \Phi(s_i', s_{-i}))$. This potential is easily extended to related machines [9]. For the MAKESPAN policy it is shown in [11] that for a population of only egoists best response dynamics can take $O(2^{\sqrt{n}})$ steps to converge to a pure Nash equilibrium. For identical machines there is a scheduling of tasks to reach a Nash equilibrium with better response dynamics in polynomial time. In addition, there are polynomial time algorithms to compute Nash equilibria on related machines and instances with link restrictions [11, 16].

Including altruists provides a quite different set of results. We observe that even if there is only one altruist, existence of a pure Nash equilibrium is not guaranteed.

Proposition 1. *There is a game on two identical machines with the* Makespan *or* Random *policy, one altruist, and appropriately many egoists that has no pure Nash equilibrium.*

Proof. Consider a game with two machines and with one pure altruist with $p_1 = 5$ and four egoists with $p_2 = 10$, $p_3 = p_4 = p_5 = 1$. Assume there is a pure Nash equilibrium. In an equilibrium, task 2 chooses the different machine than task 1. The tasks 3, 4, and 5 choose the machine different than task 2. However, task 1 would choose the machine with only task 2, which leads to a contradiction. The idea can be adjusted to an arbitrary altruist with $\beta_1 > 0$ by adding sufficiently many tasks with small processing time. In particular, instead of 3 we add strictly more than $1 + \frac{1,4}{\beta_1}$ many egoists, which all have equally small processing time, and for which their total processing time adds up to 3. For this game it can be shown that all arguments given above are preserved. □

In addition, we can show that it is NP-hard to decide if a pure Nash equilibrium exists. The reduction is from Partition.

Theorem 2. *It is weakly NP-hard to decide if a game on three identical machines with* Makespan *and one pure altruist has a pure Nash equilibrium.*

Proof. We reduce from Partition. An instance $\mathcal{I}$ is given as $(a_1, \ldots, a_n) \in \mathbb{N}^n$ and $\mathcal{I} \in$Partition if and only if $\exists I \subset \{1, \ldots, n\}$ with $\sum_{i \in I} a_i = \sum_{j \in \{1,\ldots,n\} \setminus I} a_j$. We first reduce a given instance $\mathcal{I} = (a_1, \ldots, a_n)$ to an instance $\mathcal{I}' = (a_1, \ldots, a_n, a_{n+1}, \ldots, a_{n+8})$ with $a_{n+1} = \ldots = a_{n+8} = \sum_{i \in \{1,\ldots,n\}} a_i$. Clearly $\mathcal{I} \in$Partition if and only if $\mathcal{I}' \in$Partition.

In a second step we construct a scheduling game $G_{\mathcal{I}'}$ that has a pure Nash equilibrium if and only if $\mathcal{I}' \in$Partition. The game consists of three machines and $n + 8 + 2$ tasks. The processing time p_i of task $1 \leq i \leq n + 8$ is a_i. Task $n + 9$ has processing time $p_{n+9} = \sum_{1 \leq j \leq n+8} a_j$ and task $n + 10$ has processing time $p_{n+10} = \frac{1}{2} \sum_{1 \leq j \leq n+8} a_j$. All tasks are pure egoists except for task $n + 10$ who is a pure altruist.

If $\mathcal{I} \in$Partition, there is an $I \subset \{1, \ldots, n+8\}$ with $\sum_{i \in I} a_i = \frac{1}{2} \sum_{1 \leq j \leq n+8} a_j$. Scheduling all tasks $i \in I$ on machine one, all tasks $j \in \{1, \ldots, n+8\} \setminus I$ on machine two, and the remaining tasks $n + 9$ and $n + 10$ on machine three is a pure Nash equilibrium. Note that the first two machines have a load of $\frac{1}{2} \sum_{1 \leq j \leq n+8} a_j$ and machine three has a load of $\frac{3}{2} \sum_{1 \leq j \leq n+8} a_j$. Obviously, no task from the first two machines has an incentive to change to machine three. Neither has task $n + 9$ an incentive to change to one of the first two machines because his processing time would not change. The altruistic task cannot improve the social cost by changing to one of the first two machines. Note that at least 4 tasks (half of the tasks $n, \ldots, n + 8$) are scheduled on each of the first two machines. Therefore the social cost increases by at least $4p_{n+10} - (p_{n+10} + p_{n+9}) > 0$.

If $\mathcal{I} \notin$Partition, assume for the sake of contradiction that there is a pure Nash equilibrium. Observe that task $n + 9$ does not choose the machine that task $n + 10$ is scheduled on. Since there is no $I \subset \{1, \ldots, n + 8\}$ with $\sum_{i \in I} a_i = \frac{1}{2} \sum_{1 \leq j \leq n+8} a_j$, there exists a machine that has load of less than $\frac{1}{2} \sum_{1 \leq j \leq n+8} a_j$ (while ignoring task $n+9$). On the other hand, each of the tasks $1, \ldots, n+8$ can always choose a machine that has load less than p_{n+9}. Therefore, in equilibrium they choose the other two machines. Note, that each of these two machines has at least 4 of the tasks $n, \ldots, n+8$. Finally, the altruistic task $n+10$ chooses the machine that only task $n+9$ is scheduled on (changing to one of the other two machines increases the social cost by at least $4p_{n+10} - (p_{n+10} + p_{n+9}) > 0$). This leads to the desired contradiction. $\square$

4 Policies with Global Ordering

Probably the simplest policy with global ordering is the SHORTEST-FIRST policy, in which each machine orders tasks shortest-first depending on their processing time and processes them consecutively in this order. There is a lexicographic potential [5, 21], and every better response dynamics in a game of only egoists converges to a pure Nash equilibrium. In addition, there is a scheduling of better response moves such that a Nash equilibrium is reached in polynomial time [21]. In addition, for identical machines this pure Nash equilibrium is essentially unique and coincides with the social optimum. This implies that for identical machines and SHORTEST-FIRST there always exists a pure Nash equlibrium for any altruistic population of tasks.

Proposition 3. *For a game on identical machines with* SHORTEST-FIRST *policy there is always a pure Nash equilibrium for any altruistic population of tasks.*

Hence, for identical machines and a population of pure egoists, every Nash equilibrium is optimal. However, for a population of pure altruists suboptimal Nash equilibria can evolve, because tasks can get stuck in a local optimum. This means that the presence of altruists actually deteriorates the social cost of stable solutions.

Proposition 4. *The price of anarchy in scheduling games with* SHORTEST-FIRST *and only pure altruists is at least* $9/8$.

Proof. Consider a game with two identical machines and four tasks. Let $p_1 = p_2 = 1$ and $p_3 = p_4 = 2$. We break ties in the order of task ID, and we denote a strategy profile with task i on machine a_i by $(a_i)_{i\in\{1,2,3,4\}}$. The social optimum is the schedule $(1, 2, 1, 2)$ with social cost 8. However, the schedule $(1, 1, 2, 2)$ of cost 9 is a Nash equilibrium, as task 4 is indifferent between both machines. $\square$

Let us further examine convergence properties of best-response dynamics. We use the above game to construct a cycling sequence even for uniform altruists, for any $\beta \in (0, 1)$.

Theorem 5. *Best-reponse dynamics do not converge to a pure Nash equilibrium, even for two identical machines with* SHORTEST-FIRST*, for (1) three egoists and one pure altruist; or (2) four β-uniform altruists, for every $\beta \in (0, 1)$.*

Proof. Consider a game with one altruist with $p_1 = 1$ and three egoists with $p_2 = 4$, $p_3 = 5.9$, and $p_4 = 6$. We again denote a strategy profile with task i on machine a_i by $(a_i)_{i\in\{1,2,3,4\}}$. It is easy to check that the sequence $(1, 1, 2, 2) \to (1, 1, 2, 1) \to (1, 2, 2, 1) \to (1, 2, 1, 1) \to (2, 2, 1, 1)$ is a best response sequence. Note that the first and last strategy profile are symmetric and, therefore, there is cycle of best responses. This proves the first part of the theorem.

For the second part we consider the game more generally. We assume only that $p_1 < p_2 < p_3 < p_4$ and w.l.o.g. set $p_1 = 1$. Task 4 moves $(1, 1, 2, 2) \to (1, 1, 2, 1)$ as long as $1 + p_2 < p_3$. Task 2 always moves $(1, 1, 2, 1) \to (1, 2, 2, 1)$. Task 3 moves $(1, 2, 2, 1) \to (1, 2, 1, 1)$ when $p_2 > \beta p_3 + 1$. Finally, task 1 moves $\to (1, 2, 1, 1) \to (2, 2, 1, 1)$ in any case. This yields two inequalities $1 + p_2 < p_3$ and $p_2 > \beta p_3 + 1$. We obtain $p_2 > (1 + \beta)/(1 - \beta)$, and $1 + p_2 < p_3 < (p_2 - 1)/\beta$. This can obviously be fulfilled for every $\beta \in (0, 1)$. $\square$

Immorlica et al. [21] show that there is a scheduling of better response moves for egoistic selfish tasks such that they reach a Nash equilibirum on unrelated machines in time $O(n^2)$. In a similar manner, the same result can be shown for any deterministic ordering policy on identical and related machines. We show here that in the worst case better response dynamics can require an exponential number of steps to reach a Nash equilibrium, even on identical machines. To the best of our knowledge this has not been shown before. The proof has similarities with the one given for the same result with the MAKESPAN policy in [11,26]. For completeness we provide a proof in the Appendix.

Theorem 6. *For an entirely egoistic population and identical machines with* SHORTEST-FIRST*, better response dynamics can take $O\left(2^{\sqrt{n}}\right)$ steps to reach a Nash equilibrium.*

Note that our main argument can be made essentially without consideration of processing times. Hence, the proof can be adjusted to hold for for any deterministic ordering policy.

Corollary 7. *For an entirely egoistic population, better response dynamics can take $O\left(2^{\sqrt{n}}\right)$ steps to reach a pure Nash equilibrium for any deterministsic ordering policy on identical machines.*

In the remainder of this section we briefly discuss another simple ordering policy, namely LONGEST-FIRST. For entirely egoistic populations this policy yields a potential game for identical and related machines. It has recently been shown that for three unrelated machines LONGEST-FIRST does not guarantee a pure Nash equilibrium [7]. When it comes to heterogeneous populations, it is possible to show that even on identical machines pure Nash equilibria can be absent.

Theorem 8. *There are games that have no pure Nash equilibrium on two identical machines with* LONGEST-FIRST *policy and (1) one altruist, and five egosits; or (2) six β-uniform altruists, for any $\beta \in (0, 1/3)$.*

Proof. Consider a game with the altruistic task with $p_1 = 11$ and five egoists with $p_2 = 10$, $p_3 = 7$, $p_4 = 1.1$, $p_5 = 1.2$, and $p_6 = 1.3$. For the sake of contradiction assume there is an equilibrium. Clearly, task 2 always chooses the machine that task 1 is not on. Now, given that task 1 and 2 are on different machines, task 3 always chooses the machine that task 2 is on. Finally, the tasks 4, 5, and 6 choose the machine with only task 1. However, in such a state the altruistic task 1 could improve the social cost by $4 \cdot 11 - 3 \cdot 11$ when changing to the other machine. This contradicts the assumption that this is a equilibrium.

For the case of β-uniform altruists, we extend the construction as before. We assume that the processing times are $p_1 > ... > p_6$, and for simplicity that p_4, p_5, and p_6 are sufficiently small. This allows to normalize $p_3 = 1$. To make tasks 1 and 2 go to separate machines, we must have $p_1 > 4\beta p_2$. To make task 3 prefer the machine of task 2 it must hold that $p_1 > p_2 + 3\beta$. Also, we require $p_1 < p_2 + 1$. Then, if tasks 4, 5, and 6 are sufficiently small, they will prefer to join task 1 on his machine. Finally, this yields and incentive for task 1 to join task 2, and a contradiction is reached. The set of inequalities only postulates that $0 < \beta < 1/3$. □

Using exponentially increasing task lengths we can adjust the proof of Theorem 6 and create an exponentially long better response sequence also for populations of pure altruists and LONGEST-FIRST. Details are left for a more complete version of the paper.

Corollary 9. *For a population of pure altruists and identical machines with* Longest-First*, better response dynamics can take* $O\left(2^{\sqrt{n}}\right)$ *steps to reach a Nash equilibrium.*

5 Time-Sharing Policy

In contrast to the previous results, we show here that there is a policy closely related to Makespan and Shortest-First, for which stabilization is robust against arbitrary altruistic behavior. The Time-Sharing policy is inspired by generalized processor sharing [25]. It has recently been studied as a coordination mechanism in [7]. All tasks are started simultaneously, and all tasks are processed in equal shares by the machine. When the smallest task is finished, the machine is shared in equal parts by the remaining tasks, and so on. For a population of only egoists the policy yields an exact potential function, even on unrelated machines. The potential function can be rewritten as the sum of completion times $c^{SF}(s)$ for the same assignment and the Shortest-First policy. This turns out to be the sum of completion times $c^{TS}(s)$ for Time-Sharing with a correction term. Using straightforward calculation it is possible to show

$$\Phi(s) = c^{SF}(s) = \frac{1}{2}\left(c^{TS}(s) + \sum_i p_{i,s_i}\right) .$$

This allows us to derive the following result.

Theorem 10. *For any population of tasks on unrelated machines with the* Time-Sharing *policy, a pure Nash equilibrium always exists and any better response dynamics converges.*

Proof. We can construct a weighted potential using Φ and add a set of correction terms. This is essentially the same approach as for the case of linear delays in [19]. In particular, we get

$$\Phi_w(s) = \Phi(s) - \sum_i p_{i,s_i} \cdot \frac{\beta_i}{1+\beta_i} = \frac{1}{2}\left(c^{TS}(s) + \sum_i p_{i,s_i} \cdot \frac{1-\beta_i}{1+\beta_i}\right) .$$

Suppose task i switches from s_i to s_i'. We denote the resulting states by s and $s' = (s_i', s_{-i})$. Then

$$\begin{aligned} c_i^{TS}(s) - c_i^{TS}(s') &= (1-\beta_i)(f_i(s) - f_i(s')) + \beta_i(c^{TS}(s) - c^{TS}(s')) \\ &= (1-\beta_i)(\Phi(s) - \Phi(s')) + \beta_i(c^{TS}(s) - c^{TS}(s')) \\ &= \frac{1+\beta_i}{2} \cdot (c^{TS}(s) - c^{TS}(s')) + \frac{1-\beta_i}{2} \cdot (p_{i,s_i} - p_{i,s_i'}) \\ &= \frac{1+\beta_i}{2} \cdot \left((c^{TS}(s) - c^{TS}(s')) + \frac{1-\beta_i}{1+\beta_i} \cdot (p_{i,s_i} - p_{i,s_i'})\right) \\ &= (1+\beta_i) \cdot (\Phi_w(s) - \Phi_w(s')) \ , \end{aligned}$$

which proves the theorem. □

This implies existence of pure Nash equilibria and convergence of every better response dynamics. In addition, we show that computing a Nash equilibrium can be done in polynomial time.

Theorem 11. *For any population of tasks on unrelated machines with the* Time-Sharing *policy, a pure Nash equilibrium can be computed in polynomial time.*

Proof. Finding a socially optimal schedule for the SHORTEST-FIRST policy can be done with a bipartite matching [2] by setting up a complete bipartite network, in which one partition is the set of tasks and the other partition consists of nm nodes (j, k) for positions $k = 1, \ldots, n$ and machines $j = 1, \ldots, m$. The kth-to-last position on machine j then induces a cost of $k \cdot p_{ij}$ for task i. This cost is attached to the corresponding edge $\{i, (j, k)\}$. Reversing the order of summation yields that a minimum cost matching is an optimal assignment. We can set up this bipartite network and subtract $(2\beta_i p_{ij})/(1 + \beta_i)$ from each edge weight between task i and any position on machine j. The resulting minimization problem can again be solved by matching. Due to the structure the optimal solution respects the shortest-first ordering on each machine. Thus, we can efficiently find a global optimum of Φ_w, which must be a pure Nash equilibrium. □

References

[1] Yossi Azar, Kamal Jain, and Vahab Mirrokni. (Almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proc. 19th Symp. Discrete Algorithms (SODA)*, pages 323–332, 2008.

[2] J. Bruno, E. Coffman Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Comm. ACM*, 17(7):382–387, 1974.

[3] Ioannis Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *Proc. 20th Symp. Discrete Algorithms (SODA)*, pages 815–824, 2009.

[4] Po-An Chen and David Kempe. Altruism, selfishness, and spite in traffic routing. In *Proc. 9th Conf. Electronic Commerce (EC)*, pages 140–149, 2008.

[5] George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. *Theor. Comput. Sci.*, 410(36):3327–3336, 2009.

[6] Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibira. *ACM Trans. Algorithms*, 3(1), 2007.

[7] Christoph Dürr and Nguyen Kim Thang. Non-clairvoyant scheduling games. In *Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT)*, 2009. To appear.

[8] Ilan Eshel, Larry Samuelson, and Avner Shaked. Altruists, egoists and hooligans in a local interaction model. *Amer. Econ. Rev.*, 88(1):157–179, 1998.

[9] Eyal Even-Dar, Alexander Kesselman, and Yishay Mansour. Convergence time to Nash equilibria. In *Proc. 30th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 502–513, 2003.

[10] Ernst Fehr and Klaus Schmidt. A theory of fairness, competition, and cooperation. *The Quarterly Journal of Economics*, 114:817–868, 1999.

[11] Rainer Feldmann, Martin Gairing, Thomas Lücking, Burkhard Monien, and Manuel Rode. Nashification and the coordination ratio for a selfish routing game. In *Proc. 30th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 514–526, 2003.

[12] Amos Fiat, Haim Kaplan, Meital Levy, and Svetlana Olonetsky. Strong price of anarchy for machine load balancing. In *Proc. 34th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 583–594, 2007.

[13] Dimitris Fotakis, Spyros Kontogiannis, Elias Koutsoupias, Marios Mavronicolas, and Paul Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. *Theor. Comput. Sci.*, 410(36):3305–3326, 2009.

[14] Dimitris Fotakis, Spyros Kontogiannis, and Paul Spirakis. Atomic congestion games among coalitions. *ACM Trans. Algorithms*, 4(4), 2008.

[15] Martin Gairing. *Selfish Routing in Networks*. PhD thesis, University of Paderborn, 2006.

[16] Martin Gairing, Thomas Lücking, Marios Mavronicolas, and Burkhard Monien. Computing Nash equilibria for scheduling on restricted parallel links. In *Proc. 36th Symp. Theory of Computing (STOC)*, pages 613–622, 2004.

[17] Martin Gairing, Thomas Lücking, Marios Mavronicolas, Burkhard Monien, and Manuel Rode. Nash equilibria in discrete routing games with convex latency functions. *J. Comput. Syst. Sci.*, 74(7):1199–1225, 2008.

[18] Herbert Gintis, Samuel Bowles, Robert Boyd, and Ernst Fehr. *Moral Sentiments and Material Interests: The Foundations of Cooperation in Economic Life*. MIT Press, 2005.

[19] Martin Hoefer and Alexander Skopalik. Altruism in congestion games. In *Proc. 17th European Symposium on Algorithms (ESA)*, pages 179–189, 2009.

[20] Martin Hoefer and Alexander Souza. Tradeoffs and average-case equilibria in selfish routing. In *Proc. 15th European Symposium on Algorithms (ESA)*, pages 63–74, 2007.

[21] Nicole Immorlica, Li Li, Vahab Mirrokni, and Andreas Schulz. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410(17):1589–1598, 2009.

[22] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Proc. 16th Symp. Theoretical Aspects of Computer Science (STACS)*, pages 404–413, 1999.

[23] John Ledyard. Public goods: A survey of experimental resesearch. In John Kagel and Alvin Roth, editors, *Handbook of Experimental Economics*, pages 111–194. Princeton University Press, 1997.

[24] David Levine. Modeling altruism and spitefulness in experiments. *Review of Economic Dynamics*, 1:593–622, 1998.

[25] Abhay Parekh and Robert Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Trans. Netw.*, 1(3):344–357, 1993.

[26] Manuel Rode. *Nash equilibria in discrete routing games*. PhD thesis, University of Paderborn, 2004.

[27] Berthold Vöcking. Selfish load balancing. In Noam Nisan, Éva Tardos, Tim Roughgarden, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 20. Cambridge University Press, 2007.

A Proof of Theorem 6

Proof. To proof the theorem, we construct a family of games (G_n) of size $O(n^2)$. We show that for each game G_n there is an initial schedule and a sequence of better response steps that has length $2^{\Omega(\sqrt{n})}$. The game G_n has $n+1$ machines $0, \ldots, n$. The tasks of G_n are divided in n groups $g_1, \ldots, g_n$. Each group g_i consists of $i+1$ tasks denoted by $t_i^1, \ldots, t_i^i$.

The processing times of the task are nearly identical. That is, the processing times only determine the priority of execution but the choice of task depends only on the number of tasks with smaller processing times but not on their processing times. That is for each two subset of task I and J with $|I| = |J| + 1$ the sum of the processing times of the tasks in J is larger than that of I. Additionally, the processing times are chosen such that $p_i^j > p_{i'}^{j'}$ for all $i < i'$ and $p_i^j > p_i^{j'}$ for all $j < j'$. We say a task has cost c if c tasks with smaller processing times are scheduled on his machine.

Consider the initial schedule in which all tasks of group g_i are on machine i. The exponential sequence of best responses is described by the recursive algorithm *Go(i)*. Due to its recursive nature, it is easy to see that this algorithm describes an exponentially long sequence. It remains to show that each of the steps is a better response.

Algorithm 1 The recursive algorithm Go

Procedure Go(1):
Task t_1^1 moves to machine 0

Procedure Go(i):
Call Go($i-1$);
Task t_i^2 moves to machine 0, task t_i^1 moves to machine 0.
In decreasing order of their processing times, all task of all groups g_j with $(j < i)$ on machine 0 move to machine 1.
for all $k = 1$ to $i-2$ **do**
    Task t_i^{k+2} moves to machine k
    In decreasing order of their processing times, all tasks of all groups g_j with $(k < j < i)$ on machine k move to machine $k+1$.
end for
Task t_i^1 moves from machine 0 to machine $i-1$.
Call Go($i-1$)

Lemma 12. *The changes of tasks as described by Algorithm 1 are better response moves.*

Proof. To proof this lemma we introduce the following two conditions.

(Condition 1) When Go(i) is executed, the following holds. All tasks of group g_i are on machine i. There is a $k \geq 1$ such that there are exactly k tasks of groups $g_{i'}$ with $i' > i$ on the machines $0, \ldots, i$

(Condition 2) After the execution of Go(i) the following holds. There is a $k \geq 1$ such that there are exactly k tasks of groups $g_{i'}$ with $i' \geq i$ on the machines $0 \ldots i$ For all $i' \geq i$, either each of the machines $0 \ldots i$ has a task of group $g_{i'}$ or none.

We show by induction on the recursive execution of $Go(i)$ that these conditions are true and the described steps are better responses.

1. During the first executions of Go(n), …,Go(1), Condition 1 is obviously satisfied since the initial schedule has not changed.

2. If the Condition 1 holds for Go(1), task p_1^1 can reduce his cost by moving to machine 0. He has cost of $1 + k$ on machine 1 but only cost of k on machine 0. Clearly Condition 2 holds after an execution of Go(1).

3. If the Condition 2 holds for Go($i - 1$), all tasks $t_i^1, \ldots, t_i^{i-1}$ have cost of at least $k + 1$. In steps (3) and (6) and (9) they reduce their costs by at least 1. In step (4) machine 0 has two tasks of group g_i whereas there is exactly one task of each group g_i' with $i' < i$ on both machines (Condition 2 for Go($i - 1$)). Therefore, each task can decrease his cost by 1 if it changes in the described order. The same argument holds for step (7). Therefore, Condition 1 holds for the second execution of $Go(i - 1)$.

4. If Condition 2 holds after the second execution of Go($i-1$), it also holds for Go(i) since the tasks of group g_i remain unchanged during step 10.

By induction Lemma 12 follows.

□

This proves the theorem. □