Tradeoffs and Average-Case Equilibria in Selfish Routing

Martin Hoefer *   Alexander Souza †

June 23, 2007

Abstract

We consider the price of selfish routing in terms of tradeoffs and from an average-case perspective. Each player in a network game seeks to send a message with a certain length by choosing one of several parallel links that have transmission speeds. A player desires to minimize his own transmission time (latency). We study the quality of Nash equilibria of the game, in which no player can decrease his latency by unilaterally changing his link. In this paper we treat two important aspects of network-traffic management: the influence of the total traffic upon network performance and fluctuations in the lengths of the messages. We introduce a probabilistic model where message lengths are random variables and evaluate the *expected price of anarchy* of the game for various social cost functions.

For total latency social cost, which was only scarcely considered in previous work so far, we show that the price of anarchy is $\Theta\left(\frac{n}{t}\right)$, where n is the number of players and t the total message-length. The bound states that the relative quality of Nash equilibria in comparison with the social optimum increase with increasing traffic. This result also transfers to the situation when fluctuations are present, as the expected price of anarchy is $O\left(\frac{n}{\mathbb{E}[T]}\right)$, where $\mathbb{E}[T]$ is the expected traffic. For maximum latency the expected price of anarchy is even $1 + o(1)$ for sufficiently large traffic.

Our results also have algorithmic implications. For the special case of identical links, we give an algorithm for computing the social optimum for total latency cost in polynomial time. Furthermore, our analyses of the expected prices are average-case analyses of a local search algorithm that computes Nash equilibria in polynomial time.

1 Introduction

Large-scale networks, e.g., the Internet, usually lack a central authority to coordinate the network-traffic. Instead, users that seek to send messages behave selfishly in order to maximize their own welfare. This selfish behaviour of network users motivates the use of game theory for the analysis of network-traffic. In standard non-cooperative games, each user, referred to as a player, is aware of the behavior of other players, and seeks to minimize his own cost. A player is considered to be satisfied with his behaviour (also referred to as his strategy) if he can not decrease his cost by unilaterally changing his strategy. If all players are satisfied, then the system is said to be in a *Nash equilibrium*.

In order to relate selfishly obtained solutions with those of an (imaginary) central authority, it is necessary to distinguish between the cost of the individual players and the cost of the whole system. The latter is also referred to as *social cost*. Depending on the choice

*Supported by DFG Research Training Group "Explorative Analysis and Visualization of Large Information Spaces". Department of Computer & Information Science, Konstanz University, Germany. hoefer@inf.uni-konstanz.de

†Department of Computer Science, University of Freiburg, souza@informatik.uni-freiburg.de

of a social cost function selfish behaviour of the players might not optimize the social cost. Consequently, the question arises how bad the social cost of a Nash equilibrium can be in comparison to the optimum. In a seminal work, Koutsoupias and Papadimitriou [17] formulated the concept of the *price of anarchy* (originally referred to as *coordination ratio*) as the maximum ratio of the social cost of a Nash equilibrium and the optimum social cost, taken with respect to all Nash equilibria of the game. In other words, the worst selfish solution in comparison with the optimum.

Specifically, in [17], the KP-model for selfish routing was introduced: each of n players seeks to send a message with respective length t_j across a network consisting of m parallel links having respective transmission speed s_i. The cost of a player j, called his *latency* ℓ_j, is the total length of messages on his chosen link i scaled with the speed, i.e., $\ell_j = \frac{1}{s_i} \sum_{k \text{ on } i} t_k$. The latency corresponds to the duration of the transmission when the channel is shared by a certain set of players. The social cost of an assignment was assumed to be the maximum duration on any channel, i.e., the social cost is $\ell_{\max} = \max_j \ell_j$. Koutsoupias and Papadimitriou [17] proved initial bounds on the price of anarchy for special cases of this game, but Czumaj and Vöcking [8] were the first ones to give tight bounds for the general case: $\Theta\left(\frac{\log m}{\log\log\log m}\right)$ for mixed and $\Theta\left(\frac{\log m}{\log\log m}\right)$ for pure Nash equilibria.

In this paper, we mainly concentrate on total latency social cost $\sum_j \ell_j$, but also apply our techniques to maximum latency and polynomial load. The KP-model is usually associated with maximum latency. Other latency functions were considered in [7, 11, 12], specifically total latency was included in [12], but there was no treatment of the price of anarchy. The price of anarchy in more general atomic and non-atomic congestion games was considered e.g. by [1, 3, 5, 20, 21]. For a recent survey on results related to atomic congestion games we defer the reader to [16].

We consider two important aspects of network-management: the influence of the *total traffic* $t = \sum_j t_j$ upon the overall system performance and *fluctuations* in the length of the respective message-lengths t_j. We model the first aspect as follows: an adversary is allowed to specify the task-lengths t_j subject to the constraint that $\sum_j t_j = t$, where t is a parameter specified in advance. We consider the price of anarchy and stability of the game with a malicous adversary explicitly restricted in this way. This model is closely related to the work of Awerbuch et al. [4], which treats the KP-model with link restrictions and unrelated machines, i.e., each task j has an arbitrary length $t_{i,j}$ on machine i. They proved tight bounds on the price of anarchy in both models that depend on the tradeoff-ratio of the longest task with the optimum value. However, their results are only for maximum latency.

For the second aspect – fluctuations – consider a sequence of such network games all with the same set of players and the same network topology, but where the message-lengths of the players might differ from game to game. This corresponds to the natural situation that users often want to transmit messages with different lengths. The main question addressed in this respect is how the price of anarchy behaves in a "typical" game. To formalize the notion of fluctuation, we consider a probabilistic model in which the message-lengths are random variables T_j. We evaluate the quality of Nash equilibria in this probabilistic KP-model with the *expected value of the price of anarchy* of the game. The notion of an expected price of anarchy was, to our knowledge, considered before only by Mavronicolas et al. [18] in the context of a cost sharing mechanism, who referred to it as *diffuse price of anarchy.* Different forms of randomization in the KP-model were either subject to machine assignment in mixed Nash equilibria [17] or subject to incomplete information [13].

This new measure is interesting by itself, however, it also captures the performance of a polynomial time algorithm proposed by Feldmann et al. [9]. This algorithm calculates for any given initial assignment a pure Nash equilibrium – and for maximum latency this equilibrium is of smaller or equal cost. This shows that for maximum latency the price of stability is 1; and using a PTAS for makespan scheduling [14], a Nash equilibrium which is a $(1+\varepsilon)$-approximation to the optimum social cost can be found in polynomial time. This is an interesting fact as iterative myopic best-response of players can take $\Omega(2^{\sqrt{n}})$ steps to converge to a pure Nash equilibrium. Our results deliver an average-case analysis of this algorithm for total and maximum latency and polynomial load.

An interesting adjustment, which we will also analyze in terms of traffic and fluctuations, concerns coordination mechanisms, see [6,15]: instead of processing all tasks in parallel each machine has a deterministic scheduling rule to sequence the tasks assigned to it. A player then tries to minimize the completion time of his job by switching to the machine where it is processed earliest.

1.1 Model and Notation

We consider the following network model: m parallel links with speeds $s_1 \geq \cdots \geq s_m \geq 1$ connect a source s with a target t. We note that the assumption of $s_m \geq 1$ is without loss of generality, as speeds can be normalized without changing the results. There are n players in the game, and each player seeks to send a message from s to t, where each message j has *length* t_j.

This model can naturally be described with scheduling terminology, and we refer to it as *selfish scheduling*. Each of the m links is a machine with speed s_i and each message j is a task with task-length t_j. The strategy of a player is to choose of one of the machines to execute its task. The total length on machine i is its *load* $w_i = \sum_{k \text{ on } i} t_k$. We assume that each machine executes its tasks in parallel, i.e., the resource is shared among the players that have chosen it. Hence, the duration of a task j is proportional to the total length on the chosen link i and its speed s_i, i.e., its *latency* is $\ell_j = \frac{1}{s_i} \sum_{k \text{ on } i} t_k = \frac{w_i}{s_i}$.

A *schedule* is any function $s : J \longrightarrow M$ that maps any element of the set $J = \{1, 2, \ldots, n\}$ of tasks to an element in the set of machines $M = \{1, 2, \ldots, m\}$. Each machine i executes the tasks assigned to it in parallel, which yields that each task j is finished at time ℓ_j. The *finishing time* of a machine i is hence given by $f_i = \frac{1}{s_i} \sum_{k \text{ on } i} t_k = \frac{w_i}{s_i}$. The disutility of each player is the latency of its task, i.e., the selfish incentive of every player is to minimize the individual latency.

A schedule is said to be in a *(pure) Nash equilibrium* if no player can decrease his latency by unilaterally changing the machine his task is processed on. More formally, the schedule s has the property that for each task j

$$f_i + \frac{t_j}{s_i} \geq f_{s(j)} \quad \text{holds for every } i. \tag{1}$$

In this paper, we restrict our attention to pure Nash equilibria, i.e., the strategy of each player is to choose one machine rather than a probability distribution over machines. It is known that any selfish scheduling game admits at least one pure Nash equilibrium, see, e.g., [10,24].

Schedules are valued with a certain *(social) cost* function cost : $\Sigma \longrightarrow \mathbb{R}_+$, where Σ denotes the set of all schedules. Notice that each Nash equilibrium is simply a schedule that satisfies the stability criterion (1). In contrast, an *optimum* schedule s^* is one which minimizes the cost function over all schedules, regardless if it is a Nash equilibrium or not. These differences in mind, it is natural to ask how much worse Nash equilibria can be compared to the optimum. The *price of anarchy* [17] relates the Nash equilibrium with highest social cost to the optimum, i.e, it denotes the fraction of the of the cost of the worst Nash equilibrium over the cost of the best possible solution. In contrast, the *price of* *stability* [2] relates the Nash equilibrium with lowest social cost to the optimum, i.e, it denotes the fraction of the cost of the best Nash equilibrium over the cost of the best possible solution. One might wonder about a reasonable choice for a social cost function. Arguably, it depends on the application at hand which social cost function is advisable. Clearly, depending on the game, the prices of anarchy and stability behave quite differently.

1.2 Results and Contributions

We characterize two important aspects of network-traffic management in the KP-setting: the influence of total traffic and fluctuations in message-lengths on the performance of Nash equilibria. Throughout the paper upper-case letters denote random variables and lower-case letters their outcomes, respectively constants.

Traffic and Fluctuations Model

We first concentrate on the influence of the system load upon Nash equilibrium performance. In the *traffic model*, an adversary is free to specify task-lengths $\mathbf{t} = (t_1, \ldots, t_n)$ subject to the constraints that $t_j \in [0, 1]$ and $\sum_j t_j = t$, where $0 < t \leq n$ is a parameter specified in advance. We evaluate the quality of Nash equilibria with prices of anarchy and stability on the induced set of possible inputs.

Then we formulate an extension to message fluctuations. The standard KP-game is deterministic, i.e., the task-lengths are fixed in advance. What happens if the task-lengths are subject to (random) fluctuations? What are prices of anarchy and stability in a "typical" instance? We capture these notions with the *fluctuations model*. Let the task-length T_j of a task j be a random variable with finite expectation $\mathbb{E}[T_j] = \mu_j$. As before, a schedule is a *Nash equilibrium* if (1) holds, i.e., if the concrete *realisations* t_j of the random variables T_j satisfy the stability criterion. Consequently, the set of schedules that are Nash equilibria is a random variable itself.

We define the *expected price of anarchy* by

$$\mathrm{EPoA}(\Sigma) = \mathbb{E}\left[\max\left\{\frac{\mathrm{cost}(S)}{\mathrm{cost}(S^*)} : \ S \in \Sigma \text{ is a Nash equilibrium}\right\}\right]$$

and the *expected price of stability*

$$\mathrm{EPoS}(\Sigma) = \mathbb{E}\left[\min\left\{\frac{\mathrm{cost}(S)}{\mathrm{cost}(S^*)} : \ S \in \Sigma \text{ is a Nash equilibrium}\right\}\right]$$

in straightforward manner. Notice that each expected value is taken with respect to the random task-lengths T_j. This means that the expectation is accumulated by evaluating the price of anarchy (respectively stability) for each outcome t_j of the random variables T_j which is then weighted with the respective probability.

Social Cost Functions

We consider three different social cost functions: total latency $\sum_{j \in J} \ell_j$, maximum latency $\max_{j \in J} \ell_j$, and total polynomial load $\sum_{i \in M} w_i^d$, where $d \in \mathbb{N}$ is constant. The machines have possibly different speeds in the first two cases; identical machines in the third.

Our main results for total latency $\sum_j \ell_j$ in Section 2 are as follows. Theorem 2.1 for the traffic model states that the (worst-case) prices of stability, respectively anarchy are essentially $\Theta\left(\frac{n}{t}\right)$. This means that in the worst-case, for small values for t, a Nash equilibrium judged with total latency social cost can be up to a factor $\Theta(n)$ larger than the optimum solution. However, as the total traffic t grows, the performance loss of Nash equilibria becomes less severe. For highly congested networks, i.e., for t being linear in n, Nash equilibria approximate the optimum solution within a constant factor.

It turns out that this behavior is stable also under the presence of fluctuations in message-lengths. Theorem 2.4 states an analogous result for the stochastic setting: the expected price of anarchy of this game is $O\left(\frac{n}{\mathbb{E}[T]}\right)$, where $T = \sum_j T_j$ is the total random traffic and $\mathbb{E}[T] = \omega\left(\sqrt{n \log n}\right)$. The result holds under relatively weak assumptions on the distributions of the T_j; even limited dependence among them is allowed. The assumption $\mathbb{E}[T] = \omega\left(\sqrt{n \log n}\right)$ is already satisfied if, for example, there is a constant lower bound on the task length of every player. Intuitively, it requires that on average there are not too many too small tasks in the game. In our opinion, it is a reasonable assumption when analyzing practical systems.

The results for maximum latency and total polynomial load fall into a similar regime. For maximum latency $\max_j \ell_j$, it is already known (see [8]) that the price of stability is 1 and that the price of anarchy is $\Theta\left(\min\left\{\frac{\log m}{\log\log m}, \log\frac{s_1}{s_m}\right\}\right)$. However, in Theorem 3.2, we establish that the expected price of anarchy is $1 + \frac{m^2}{\mathbb{E}[T]}$, i.e., Nash equilibria are almost optimal solutions for sufficiently large traffic. This result is related to Awerbuch et al. [4] since their bounds also depend on a tradeoff between the largest task and the optimum solution. However, our results translate to selfish scheduling with coordination mechanisms; see Corollary 3.5.

The treatment of total polynomial load $\sum_i w_i^d$ is deferred to Appendix 4 due to space limitations. It is known that the price of anarchy is at most $\frac{(2^d-1)^d}{(d-1)(2^d-2)^{d-1}}\left(\frac{d-1}{d}\right)^d$ for identical machines, see [11]. Theorem 4.2 shows that the expected price of anarchy is even $(1 + \frac{m^2}{\mathbb{E}[T]}(1 + o(1)))^d$ for identical machines.

Algorithmic Perspective

Our analyses of the expected prices of anarchy of the social cost functions provide average-case analyses of the algorithm by Feldmann et al. [9] which computes pure Nash equilibria for the KP-model in polynomial time (see e.g. Observation 2.5). Note that this algorithm can be used for classical scheduling, instead of only computing Nash equilibria. Our analysis hence gives bounds on the (expected) performance of that algorithm, which is of independent interest. Remarkably, the analysis holds for machines with (possibly) different speeds, which is a novelty over previous average-case analyses, e.g., [22, 23], where only identical machines were considered.

In addition, we show that – for the case of total latency and identical machines – the social optimum solution can be computed in polynomial time; see Theorem 2.6.

2 Total Latency Cost

In this section, we consider the social cost function *total latency* $\text{cost}(s) = \sum_j \ell_j$. Let $p_i = p_i(s)$ be the number of players assigned to machine i and let $f_i = f_i(s) = \frac{1}{s_i}\sum_{k \text{ on } i \text{ in } s} t_k$ denote the finishing time of machine i in the schedule s. Observe that we can rewrite the social cost to $\text{cost}(s) = \sum_j \ell_j = \sum_i p_i f_i$.

2.1 Traffic Model

In this model, an adversary is free to specify task-lengths $\mathbf{t} = (t_1, \ldots, t_n)$ subject to the constraints that $t_j \in [0, 1]$ and $\sum_j t_j = t$, where $0 < t \leq n$ is a parameter specified in advance.

Theorem 2.1. *Consider the selfish scheduling game on m machines with speeds $s_1 \geq \cdots \geq s_m \geq 1$, total task length $t = \sum_j t_j > 0$, where $t_j \in [0,1]$, and* $\mathrm{cost}(s) = \sum_j \ell_j$. *Then we have the bounds*

$$\frac{n}{2t} \leq \mathrm{PoS}(\Sigma) \leq \mathrm{PoA}(\Sigma) \leq \frac{n}{t} + \frac{m^2+m}{t^2}, \quad \text{for } t \geq 2 \text{ and} \tag{2}$$

$$\mathrm{PoA}(\Sigma) \leq n \quad \text{for general } t \geq 0. \tag{3}$$

The proof of the theorem relies on Lemma 2.2 and Lemma 2.3, in which a lower bound on the optimum cost and an upper bound on the cost of any Nash equilibrium are derived.

Lemma 2.2. *Let s^* be an optimum schedule for the instance $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ with speeds $s_1 \geq \cdots \geq s_m \geq 1$. Let $t = \sum_j t_j$, where $t_j \in [0,1]$. Then we have that* $\mathrm{cost}(s^*) \geq \frac{t^2}{\sum_k s_k}$.

Proof. First observe that $t_j \leq 1$ implies $\mathrm{cost}(s^*) = \sum_i p_i^* f_i^* \geq \sum_i (f_i^*)^2$. In order prove a lower bound we seek to minimize the function $\sum_i \frac{x_i^2}{s_i}$ subject to the constraint that $\sum_i x_i = t$. It is easily observed that this function is minimized for the choice $x_i = \frac{ts_i}{\sum_{k=1}^m s_k}$. Hence, $\mathrm{cost}(s^*) = \sum_i p_i^* f_i^* \geq \sum_i (f_i^*)^2 \geq \sum_i x_i^2 = \sum_i \left(\frac{s_i}{\sum_k s_k} t\right)^2 = \frac{t^2}{\sum_k s_k}$ as claimed. ∎

Lemma 2.3. *For every Nash equilibrium s for the selfish scheduling game on m machines with speeds $s_1 \geq \cdots \geq s_m \geq 1$, $t_j \in [0,1]$, and $t = \sum_j t_j$ we have that* $\mathrm{cost}(s) \leq \frac{n(t+m^2+m)}{\sum_i s_i}$.

Proof. Recall that the cost of any schedule s can be written as $\mathrm{cost}(s) = \sum_j \ell_j = \sum_i p_i f_i$, where ℓ_j denotes the latency of task j, p_i the number of players on machine i, and f_i its finishing time.

It turns out that it is useful to distinguish between fast and slow machines. A machine is *fast* if $s_i \geq \frac{1}{m} \sum_k s_k$; otherwise *slow*. Notice that this definition implies that there is always at least one fast machine.

Recall that the *load* of a machine i is defined by $w_i = \sum_{j \text{ on } i} t_j$. We rewrite the load w_i in terms of deviation from the respective ideally balanced load. Let $\overline{w}_i = \frac{s_i}{\sum_k s_k} \sum_j t_j = \frac{s_i}{\sum_k s_k} t$ and define the variables y_i by $w_i = \overline{w}_i + y_i$. Observe that $f_i = \frac{1}{s_i}(\overline{w}_i + y_i) = \frac{t}{\sum_k s_k} + \frac{y_i}{s_i}$. We use the notation $x_i = \frac{y_i}{s_i}$ as a shorthand.

Let s be any schedule for which the conditions (1) of a Nash equilibrium hold. Let us assume for the moment that we can prove the upper bound $|x_i| \leq \frac{m}{s_i}$ for those schedules. Further notice that the Nash conditions (1) imply that $f_i \leq f_1 + \frac{t_j}{s_1} \leq f_1 + \frac{1}{s_1}$, where t_j is the task length of any task j on machine i. Notice that machine 1 is fast, i.e., we have $\frac{1}{s_1} \leq \frac{m}{\sum_k s_k}$. Now we calculate and find

$$\begin{aligned}\mathrm{cost}(s) &= \sum_i p_i f_i \leq \sum_i p_i \left(f_1 + \frac{1}{s_1}\right) = n\left(f_1 + \frac{1}{s_1}\right) \leq n\left(\frac{\sum_j t_j}{\sum_k s_k} + |x_1| + \frac{m}{\sum_k s_k}\right) \\ &\leq n\left(\frac{t}{\sum_k s_k} + \frac{m}{s_1} + \frac{m}{\sum_k s_k}\right) \leq \frac{n(t+m^2+m)}{\sum_k s_k}.\end{aligned}$$

It only remains to prove that the upper bound $|x_i| \leq \frac{m}{s_i}$ holds.

A machine i is *overloaded* if $y_i > 0$ (and hence also $x_i > 0$), *underloaded* if $y_i < 0$ (and hence also $x_i < 0$), and *balanced* otherwise, i.e., $y_i = x_i = 0$. Notice that $\sum_i w_i = \sum_j t_j = t$ and that $\sum_i w_i = \sum_i(\overline{w}_i + y_i) = \sum_i \frac{s_i}{\sum_k s_k} t + \sum_i y_i = t + \sum_i y_i$. This implies that $\sum_i y_i = 0$. Hence if there is an overloaded machine, there must also be an underloaded machine.

If all machines are balanced, then there is nothing to prove because $|x_i| = 0$ and the claimed bound holds. So let k be an underloaded machine and let i be an overloaded machine. Suppose that k receives an arbitrary task j from machine i, then its resulting finishing time equals $f_k + \frac{t_j}{s_k}$. The Nash conditions (1) state that $f_k + \frac{t_j}{s_k} \geq f_i$. The simple but important observation is that moving one task to an underloaded machine k turns it into an overloaded one. As $\frac{t_j}{s_k} \leq \frac{1}{s_k}$ we have $|x_k| \leq \frac{1}{s_k}$ for any underloaded machine k.

Finally, to prove $|x_i| \leq \frac{m}{s_i}$ we show a bound on the number of tasks whose removal is sufficient to turn an overloaded machine into an underloaded or balanced one. Let i be an overloaded machine and let there be u underloaded machines. Migrating (at most) u tasks from i to underloaded machines suffices to turn i into an underloaded or balanced machine. Suppose that there are at least u tasks on i, because otherwise moving the tasks on it to underloaded machines yields that i executes no task at all, and is hence clearly underloaded. Move u arbitrary tasks to the u underloaded machines by assigning one task to one underloaded machine, each. Now assume that i is still overloaded. This is a contradiction to $\sum_i y_i = 0$, because there are no underloaded machines in the system any more. Therefore i must be underloaded or balanced. As x_i equals the difference of f_i and $\frac{\sum_j t_j}{\sum_k s_k}$, we have that $|x_i| \leq \frac{u}{s_i} \leq \frac{m}{s_i}$ as each task contributes at most $\frac{1}{s_i}$ to the finishing time of machine i. The proof of the upper bound therefore proves the lemma. ∎

Proof of Theorem 2.1. The upper bound $\mathrm{PoA}(\Sigma) \leq \frac{n}{t} + \frac{m^2+m}{t^2}$ stated in (2) follows from Lemma 2.2 and Lemma 2.3. For the lower bound $\mathrm{PoS}(\Sigma) \geq \frac{n}{2t}$ we give the following instance. Let there be two unit-speed machines and an even number n of tasks. Let t be an even positive integer and define the task-lengths as follows: $t_1 = t_2 = \cdots = t_t = 1$ and $t_{t+1} = \cdots = t_n = 0$. In the (unique) optimum schedule s^* the t 1-tasks are scheduled on machine 1. All the other 0-tasks are assigned to machine 2. The value of that solution is $\mathrm{cost}(s^*) = t^2$, but we still have to prove that it is the unique optimum.

Assume that x 1-tasks and y 0-tasks are assigned to machine 1. The cost of such an assignment is given by $f(x, y) = (x + y)x + (n - x - y)(t - x)$. We are interested in the extrema of f subject to the constraints that $0 \leq x \leq t$ and $0 \leq y \leq n - t$. It is standard to prove that f is maximized for $x = \frac{t}{2}$ and $y = \frac{n-t}{2}$. Considering the gradient of the function $f(\frac{t}{2} + a, \frac{n-t}{2} + b) = \frac{nt}{2} + 2a^2 + 2ab$ in the variables a and b yields that the global minimum is attained at the boundary of the feasible region; in specific for $x = t$ and $y = 0$. The value of the minimum is $\mathrm{cost}(s^*) = f(t, 0) = t^2$.

For every Nash equilibrium of the game assigning $\frac{t}{2}$ 1-tasks to each machine is necessary: If there are $x > \frac{t}{2}$ 1-tasks e.g. on machine 1, then there is a task that can improve its latency by changing the machine. Hence $x = \frac{t}{2}$ for every Nash equilibrium. It turns out that each feasible value for y gives $f(\frac{t}{2}, y) = \frac{nt}{2}$. Thus, $\mathrm{cost}(s) = \frac{nt}{2}$ and the lower bound $\mathrm{PoS}(\Sigma) \geq \frac{n}{2t}$.

Now we prove the upper bound $\mathrm{PoA}(\Sigma) \leq n$ stated in (3). Consider an arbitrary Nash equilibrium s for an instance of the game with task-lengths $t_1, \ldots, t_n$. In the sequel we will convert s into the solution s' in which all tasks are assigned to machine 1 and we will prove that $\mathrm{cost}(s) \leq \mathrm{cost}(s')$. (The solution s' need not be a Nash equilibrium.) Notice that $\mathrm{cost}(s') = \frac{nt}{s_1}$. Further we use the trivial lower bound $\mathrm{cost}(s) \geq \frac{t}{s_1}$ which holds because each task contributes to the total cost possibly on the fastest machine 1. Both bounds imply $\mathrm{PoS}(\Sigma) \leq n$.

It remains to prove that $\mathrm{cost}(s) \leq \mathrm{cost}(s')$. We construct a series of solutions $s = \sigma_1, \sigma_2, \ldots, \sigma_m = s'$, where σ_{i+1} is obtained from σ_i by moving all tasks on machine $i + 1$ to machine 1. Recall that $w_i = \sum_{j \text{ on } i} t_j$ and notice that w_1 is monotone increasing. Therefore, the Nash conditions $\frac{w_i}{s_i} \leq \frac{w_1}{s_1} + \frac{t_j}{s_1}$ for all tasks j on machine i continue to hold until i loses all its tasks. Since $t_j \leq w_i$, this condition implies $\frac{w_1+w_i}{s_1} - \frac{w_i}{s_i} \geq 0$, which will be important for the argument below. Now we consider the change of cost when obtaining σ_i from σ_{i-1} for $i \geq 2$. For ease of notation let p_1 and p_i, w_1 and w_i denote the number of players, respectively the load of machines 1 and i of the solution σ_{i-1}. We find

$$\mathrm{cost}(\sigma_i) - \mathrm{cost}(\sigma_{i-1}) = (p_1 + p_i)\left(\frac{w_1}{s_1} + \frac{w_i}{s_1}\right) - p_1\frac{w_1}{s_1} - p_i\frac{w_i}{s_i} = p_1\frac{w_i}{s_1} + p_i\left(\frac{w_1 + w_i}{s_1} - \frac{w_i}{s_i}\right) \geq 0$$

since $\frac{w_1+w_i}{s_1} - \frac{w_i}{s_i} \geq 0$ as shown above. This completes the proof of the theorem. ∎

2.2 Fluctuations Model

Suppose that the T_j are random variables that take values in the interval $[0, 1]$ and that the T_j have respective expectations $\mathbb{E}[T_j]$. Notice that the T_j need *not* be identically distributed; even the following limited dependence is allowed.

We say that we deal with *martingale* T_j if the sequence $S_i = T_1 + \cdots + T_i + \mathbb{E}[T_{i+1}] + \cdots + \mathbb{E}[T_n]$ satisfies $\mathbb{E}[S_i \mid T_1, \ldots, T_{i-1}] = S_{i-1}$ for $i = 1, \ldots, n$.

Theorem 2.4. *Let $T = \sum_j T_j$ with $\mathbb{E}[T] = \omega\left(\sqrt{n \log n}\right)$ with martingale $T_j \in [0, 1]$. Then the expected price of anarchy of the selfish scheduling game with m machines and speeds $s_1 \geq \cdots \geq s_m \geq 1$ is bounded by:*

$$\mathrm{EPoA}(\Sigma) \leq \left(\frac{n}{\mathbb{E}[T]} + \frac{m^2}{\mathbb{E}[T]^2}\right)(1 + o(1)).$$

The asymptotics in $o(1)$ is in n. This bound does not only hold in expectation but also with probability $1 - o(1)$.

Proof. First notice that for every outcome $t = \sum_j t_j$ of the random variable $T = \sum_j T_j$ we have

$$\mathrm{PoA}(\Sigma) \leq \min\left\{n, \frac{n}{t} + \frac{m^2 + m}{t^2}\right\},$$

by Theorem 2.1.

How large is the probability that T deviates "much" from its expected value? The differences of the martingale are bounded by one: $|S_i - S_{i-1}| \leq 1$. Therefore we may apply the following Azuma-Hoeffding inequality (e.g. [19] for an introduction):

$$\Pr[|S_n - S_0| \geq \lambda] \leq 2e^{-\frac{\lambda^2}{2n}}. \tag{4}$$

With the choice $\lambda = \sqrt{4n \log n}$ we have $\Pr\left[|T - \mathbb{E}[T]| \geq \sqrt{4n \log n}\right] \leq \frac{2}{n^2}$.

Now we remember the assumption $\mathbb{E}[T] = \omega\left(\sqrt{n \log n}\right)$, and find

$$\begin{aligned}
\mathrm{EPoA}(\Sigma) &\leq \mathbb{E}\left[\min\left\{n, \frac{n}{T} + \frac{m^2 + m}{T^2}\right\}\right] \\
&= \mathbb{E}\left[\min\left\{n, \frac{n}{T} + \frac{m^2 + m}{T^2}\right\} \;\middle|\; |T - \mathbb{E}[T]| < \sqrt{4n \log n}\right] \Pr\left[|T - \mathbb{E}[T]| < \sqrt{4n \log n}\right] \\
&+ \mathbb{E}\left[\min\left\{n, \frac{n}{T} + \frac{m^2 + m}{T^2}\right\} \;\middle|\; |T - \mathbb{E}[T]| \geq \sqrt{4n \log n}\right] \Pr\left[|T - \mathbb{E}[T]| \geq \sqrt{4n \log n}\right] \\
&\leq \frac{n}{\mathbb{E}[T] - \sqrt{4n \log n}} + \frac{m^2 + m}{(\mathbb{E}[T] - \sqrt{4n \log n})^2} + n\frac{2}{n^2} \\
&= \left(\frac{n}{\mathbb{E}[T]} + \frac{m^2}{\mathbb{E}[T]^2}\right)(1 + o(1))
\end{aligned}$$

and the proof is complete ∎

2.3 Algorithmic Perspectives

2.3.1 The Average-Case of a Generic Scheduling Algorithm

In this short section, we point out that Theorem 2.1 and Theorem 2.4 also have algorithmic implications. In specific, by proving upper bounds on the expected price of anarchy of selfish scheduling, we obtain an (average-case) analysis for a generic algorithm for classical scheduling.

We consider the natural scheduling algorithm NASHIFY due to Feldmann et al. [9] introduced for maximum latency social cost and related machines. The algorithm works as follows: starting with an arbitrary schedule, it performs an initial sorting of tasks followed by greedly changing machines until a Nash equilibrium is obtained. It has running time $O\left(nm^2\right)$. It is remarkable that the algorithm also performs well for total latency minimization for classical scheduling, see Observation 2.5 below.

In the classical scheduling problem, we are given m related machines with speeds $s_1 \geq \cdots \geq s_m \geq 1$ and n tasks with respective task-length t_j. The objective is to minimize the objective function $\sum_j \ell_j$, regardless if it is a Nash equilibrium or not. Let $\text{cost}(s)$ and $\text{cost}(s^*)$ denote the objective values of a schedule obtained by NASHIFY and by an (not necessarily polynomial time) optimum algorithm OPT. The quantity $\frac{\text{cost}(s)}{\text{cost}(s^*)}$ is called the *performance ratio* of the schedule s. For random task-lengths T_j the quantity $\mathbb{E}\left[\frac{\text{cost}(S)}{\text{cost}(S^*)}\right]$ is called the *expected performance ratio*. The result below follows directly from [9], Theorem 2.1 and Theorem 2.4.

Observation 2.5. *Under the respective assumptions of Theorem 2.1 and Theorem 2.4, the bounds stated therein are upper bounds for the (expected) performance ratio of the algorithm* NASHIFY *with the objective to minimize total latency.*

2.3.2 Computing the Social Optimum

Here we consider the complexity of finding Nash equilibria and solutions with optimum social cost for identical machines. Gairing et al. [12] showed that computing the best and the worst Nash equilibrium is NP-hard. We show that, for identical machines, the social optimum can be computed in polynomial time.

Theorem 2.6. *For identical machines there is an algorithm to compute the social optimum solution for total latency scheduling in polynomial time.*

Proof. Consider any solution for the total latency scheduling problem, and suppose on machine i a number of p_i players are scheduled. It is possible to improve the solution by exchanging tasks such that tasks with small length reside on machines with large number of players.

Suppose there are tasks j and k on machines i and h, respectively, with $p_h \leq p_i$ and $t_k \leq t_j$. Then exchanging j and k results in a difference in the total latency of $(p_i - p_h)(t_k - t_j) \leq 0$. But notice that the p_i, i.e., the number of players on each machine did not change.

Now consider the tasks numbered in non-increasing order of their length. The previous property shows that in at least one optimum solution tasks are scheduled consecutively, i.e., there is an optimum solution in which each machine i gets assigned tasks j from an interval $j \in \{j_i^l, \ldots, j_i^r\}$.

The remaining open problem is to find the respective values of the p_i. The above property allows us to precompute for each possible interval the corresponding latency cost and to find the best combination of intervals by a shortest path computation.

For an instance of the total latency scheduling problem we assume w.l.o.g. that there are $m \leq n$ machines. We consider them increasingly with the intervals they get assigned. A machine i gets assigned tasks from the interval $\{j_{i-1}+1, \ldots, j_i\}$. This allows us to create a graph G, which contains nodes (i,j) for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Node (i,j) represents the fact that the interval for machine i goes up to task j. For node (i,j) we include a directed edge to each node $(i+1, v)$ for $v \in \{i+1, \ldots, n-m+i\}$. The cost $(v-i)\sum_{j=i+1}^{v} t_j$ of such an edge corresponds to the assigned interval of tasks that is implicit in the incident nodes. Finally, we add two special nodes s and t, where s is connected with directed edges to all nodes $(1, v)$ for $v \in \{1, \ldots, n-m+1\}$ of cost $v \sum_{j=1}^{v} t_j$. Node (n, m) is connected to t with an edge of cost 0. Now it can be verified that each feasible s-t-path corresponds to a feasible split of the ordered task sequence into m intervals and hence a feasible schedule. In particular, as we argued above at least one optimum solution can be represented as a shortest s-t-path in G. Hence, with a shortest path calculation in G we can find a schedule of minimum total latency in polynomial time. ∎

3 Maximum Latency Cost

This section is concerned with the social cost function $\text{cost}(s) = \ell_{\max} = \max_j \ell_j$. Both, the traffic and the fluctuations model, are considered in Section 3.1. Section 3.2 outlines how the results extend to a KP-model with coordination mechanisms.

3.1 Traffic and Fluctuations Model

Worst-case prices of stability and anarchy are already known [8] and only summarized here; see below for a further discussion.

Theorem 3.1. *In the selfish scheduling game on m machines with speeds $s_1 \geq \cdots \geq s_m \geq 1$ and $\text{cost}(s) = \ell_{\max}$ we have that $\text{PoS}(\Sigma) = 1$ and $\text{PoA}(\Sigma) = \Theta\left(\min\left\{\frac{\log m}{\log\log m}, \log \frac{s_1}{s_m}\right\}\right)$.*

Algorithm Nashify considered in Section 2.3.1 transforms any non-equilibrium schedule into a Nash equilibrium without increasing the social cost. This proves $\text{PoS}(\Sigma) = 1$. The tight bounds on the price of anarchy for pure Nash equilibria are due to Czumaj and Vöcking [8]. For the special case of identical machines different bounds for the price of anarchy hold. An upper bound $\text{PoA}(\Sigma) \leq 2 - \frac{2}{m+1}$ follows from Finn and Horowitz [10] and Nashify. Vredeveld [24] gave a schedule which is a Nash equilibrium where $2 - \frac{2}{m+1}$ is tight.

Theorem 3.2. *Consider the selfish scheduling game on m machines with speeds $s_1 \geq \cdots \geq s_m \geq 1$, total task length $t = \sum_j t_j > 0$, where $t_j \in [0,1]$, and $\text{cost}(s) = \max_j \ell_j$. Then we have the bound*

$$\text{PoA}(\Sigma) \leq 1 + \frac{m(m+1)}{t}.$$

Let $T = \sum_j T_j$ and $\mathbb{E}[T] = \omega\left(\sqrt{n \log n}\right)$ with martingale $T_j \in [0,1]$. Then the expected price of anarchy is bounded by

$$\text{EPoA}(\Sigma) \leq 1 + \frac{m^2}{\mathbb{E}[T]}(1 + o(1)).$$

The asymptotics in $o(1)$ is in n. This bound not only holds in expectation but also with probability $1 - o(1)$. Furthermore, these are bounds on the (expected) performance of algorithm Nashify.

Proof. The result for the traffic model follows from the following Lemma 3.3.

Lemma 3.3. *Let s^* be an optimum schedule and s be a Nash equilibrium of the instance $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ with speeds $s_1 \geq \cdots \geq s_m \geq 1$. Let $t_j \in [0,1]$ and $t = \sum_j t_j$. Then it holds that $\frac{t}{\sum_i s_i} \leq \text{cost}(s^*) \leq \text{cost}(s) \leq \frac{t}{\sum_i s_i} + \frac{m+1}{s_1} t_{\max}$.*

Proof. We first prove the bound for $\text{cost}(s^*)$. If each machine i receives load $\frac{s_i}{\sum_k s_k} t$, then all finishing times are equal and the schedule hence optimal. Then, the finishing time of machine 1, say, with speed s_1 is $\frac{1}{s_1} \frac{s_1}{\sum_i s_i} t = \frac{t}{\sum_i s_i}$, as claimed.

Now we prove the upper bound for $\text{cost}(s)$. Let k be the machine where the maximum is attained. The Nash conditions (1) imply that $f_k \leq f_1 + \frac{t_{\max}}{s_1}$.

Define the variable x_i by $f_i = \frac{\sum_j t_j}{\sum_i s_i} + x_i$. Analogously to the proof of Lemma 2.3 we find $|x_i| \leq \frac{m}{s_i} t_{\max}$. Therefore we have

$$f_k \leq f_1 + \frac{t_{\max}}{s_1} \leq \frac{\sum_j t_j}{\sum_i s_i} + |x_1| + \frac{t_{\max}}{s_1} \leq \frac{\sum_j t_j}{\sum_i s_i} + \frac{m+1}{s_1} t_{\max}$$

and the proof is complete. ▪

For the fluctuations model we use Lemma 3.3 and $\text{cost}(s^*) \geq \frac{t_{\max}}{s_1}$. This immediately yields

$$\frac{\text{cost}(s^*)}{\text{cost}(s)} \leq \min \left\{ 1 + \frac{\frac{m+1}{s_1} t_{\max}}{\frac{t_{\max}}{s_1}}, 1 + \frac{\frac{m+1}{s_1} t_{\max}}{\frac{t}{\sum_i s_i}} \right\} \leq \min \left\{ m+2, 1 + \frac{m(m+1)}{t} \right\}$$

where we have used $s_1 \geq \frac{1}{m} \sum_i s_i$, i.e., machine 1 is fast and $t_{\max} \leq 1$.

As in the proof of Theorem 2.4 we use the martingale and apply the Azuma-Hoeffding inequality (4). With the choice $\lambda = \sqrt{4n \log n}$ we have $\Pr\left[|T - \mathbb{E}[T]| \geq \sqrt{4n \log n} \right] \leq \frac{2}{n^2}$. We deduce

$$\text{EPoA}(\Sigma) \leq 1 + \frac{m(m+1)}{\mathbb{E}[T] - \sqrt{4n \log n}} + 2\frac{m+2}{n^2} = 1 + \frac{m^2}{\mathbb{E}[T]}(1 + o(1))$$

and the proof is complete. ▪

3.2 Coordination Mechanisms

We observe that the results of Theorem 3.2 translate to selfish scheduling with coordination mechanisms as considered by Immorlica et al. [15]. In this scenario, the machines do not process the tasks in parallel, but instead, every machine i has a (possibly different) local sequencing policy. For instance, with the Shortest-First policy, a machine considers the tasks in order of non-decreasing length. Machine i uses its policy to order the tasks that have chosen i. This yields a completion time c_j, which is different for every task processed on i. The cost for a player is now the completion time of its task on the chosen machine. Naturally, a schedule is in a Nash equilibrium if no player can reduce his completion time by switching machines. Immorlica et al. [15] showed the following worst-case bounds on the price of anarchy.

Theorem 3.4. *The price of anarchy of a deterministic policy for scheduling on machines with speeds $s_1 \geq \cdots \geq s_m \geq 1$ and social cost makespan $\text{cost}(s) = c_{\max} = \max_j c_j$ is $O(\log m)$. The price of anarchy of the* Shortest-First *policy is $\Theta(\log m)$.*

The proof of Theorem 3.2 can be adjusted to deliver the following direct corollary.

Corollary 3.5. *Under the respective assumptions of Theorem 3.2 the bounds stated therein are upper bounds for the (expected) price of anarchy for selfish scheduling with coordination mechanisms, arbitrary deterministic and social cost makespan $\text{cost}(s) = c_{\max} = \max_j c_j$.*

4 Total Polynomial Load Cost

In this section, we consider the social cost function *total polynomial load* $\text{cost}(s) = \sum_i w_i^d$ with constant $d \in \mathbb{N}$ for the special case of identical machines. Recall that the *load* of a machine i is defined by $w_i = \sum_{j \text{ on } i} t_j$. Gairing et al. [11] proved the following worst-case result.

Theorem 4.1. *In the selfish scheduling game on m identical machines and $\text{cost}(s) = \sum_i w_i^d$ we have that $\text{PoA}(\Sigma) = \frac{(2^d-1)^d}{(d-1)(2^d-2)^{d-1}} \left(\frac{d-1}{d}\right)^d$.*

However, on average, the Nash equilibria of the game can yield almost optimal solutions for this social cost function.

Theorem 4.2. *Consider the selfish scheduling game on m identical machines, total task length $t = \sum_j t_j > 0$, where $t_j \in [0,1]$, and $\text{cost}(s) = \sum_i w_i^d$. Then we have the bound*

$$\text{PoA}(\Sigma) \leq \left(1 + \frac{m(m+1)}{t}\right)^d.$$

Let $T = \sum_j T_j$ and $\mathbb{E}[T] = \omega\left(\sqrt{n \log n}\right)$ with martingale $T_j \in [0,1]$. Then the expected price of anarchy is bounded by

$$\text{EPoA}(\Sigma) = (1 + \frac{m^2}{\mathbb{E}[T]}(1 + o(1)))^d.$$

The asymptotics in $o(1)$ is in n. This bound not only holds in expectation but also with probability $1 - o(1)$. Furthermore, these are bounds on the (expected) performance of algorithm Nashify.

Proof. First we consider the traffic model and prove an upper bound on the cost of any Nash equilibrium. For machine i define x_i by $f_i = \frac{\sum_j t_j}{m} + x_i$. Without loss of generality $s_1 = \cdots = s_m = 1$ holds, i.e., the identical machines have unit speed. With the same load balancing argument as in the proof of Lemma 2.3 we find $x_1 \leq m$ and

$$\begin{aligned}\text{cost}(s) = \sum_i w_i^d \leq \sum_i (f_1 + 1)^d \leq m\left(\frac{t}{m} + |x_1| + 1\right)^d \\ \leq m\left(\frac{t}{m} + m + 1\right)^d.\end{aligned}$$

For a lower bound we use $w_i = \frac{t}{m}$ as the minimizer of $\sum_i w_i^d$ subject to. Thus,

$$\text{cost}(s^*) = \sum_i (w_i^*)^d \geq m\left(\frac{t}{m}\right)^d$$

The upper bound for the traffic model follows.

For the fluctuations model we proceed as in the proof of Theorem 2.4. We use the martingale and apply the Azuma-Hoeffding inequality (4). Then we calculate

$$\begin{aligned}\mathbb{E}\left[\frac{\text{cost}(s)}{\text{cost}(s^*)}\right] &= \mathbb{E}\left[\min\left\{\frac{(2^d-1)^d}{(d-1)(2^d-2)^{d-1}}\left(\frac{d-1}{d}\right)^d, \left(\frac{T+m^2+m}{T}\right)^d\right\}\right] \\ &\leq \mathbb{E}\left[\left(\frac{T+m^2+m}{T}\right)^d \,\middle|\, |T - \mathbb{E}[T]| \geq \sqrt{4n \log n}\right] + O\left(\frac{1}{n^2}\right) \\ &= \left(1 + \frac{m^2}{\mathbb{E}[T]}(1 + o(1))\right)^d\end{aligned}$$

and the proof is complete. ∎

5 Conclusion

In this paper we provided an initial systematic study of tradeoffs and average-case performance of Nash equilibria. Our approach is to consider total traffic as a parameter which restricts an adversary in his ability to construct (worst-case) instances. This allows to prove matching upper and lower bounds on the price of anarchy and stability of the game depending on this parameter. Then, with the expected price of anarchy the results of fluctuation are characterized. We identified conditions under which selfish behaviour is capable of producing small expected social cost. Furthermore, as a byproduct this yields an average-case analysis on the expected performance of a generic local search algorithm for various scheduling problems. Most notably, the analysis holds for machines with different speeds and relatively weak probabilistic assumptions.

Naturally, a lot of open problems remain. An immediate question is the applicability of the obtained results to the average-case performance of Nash equilibria in more general (network) congestion games. Furthermore, the characterization of average-case performance for mixed or correlated equilibria – especially for total latency – represents an interesting direction. The total latency social cost function has not been explored in a similar way as polynomial load or maximum latency, although in our opinion it has an appealing intuitive motivation as a social cost function. This might be due to the tight linear worst-case bounds, which can be obtained quite directly. However, our approach to a more detailed description offers an interesting perspective to study total latency in more general settings. In particular, the obvious similarities to polynomial load suggest that some techniques developed for the characterization of the price of anarchy in general congestion games might be applicable.

References

[1] Aland, S., Dumrauf, D., Gairing, M., Monien, B., and Schoppmann, F. Exact price of anarchy for polynomial congestion games. *Proc 23th STACS* (2006), 218–229.

[2] Anshelevich, E., Dasgupta, A., Tardos, E., and Wexler, T. Near-optimal network design with selfish agents. *Proc 35th STOC* (2003), 511 – 520.

[3] Awerbuch, B., Azar, Y., and Epstein, A. The price of routing unsplittable flow. *Proc 37th STOC* (2005), 57 – 66.

[4] Awerbuch, B., Azar, Y., Richter, Y., and Tsur, D. Tradeoffs in worst-case equilibria. *Proc 1st WAOA* (2003), 41 – 52.

[5] Christodoulou, G., and Koutsoupias, E. The price of anarchy of finite congestion games. *Proc 37th STOC* (2005), 67 – 73.

[6] Christodoulou, G., Koutsoupias, E., and Nanavati, A. Coordination mechanisms. *Proc 31st ICALP* (2004), 345 – 357.

[7] Czumaj, A., Krysta, P., and Vöcking, B. Selfish traffic allocation for server farms. *Proc 34th STOC* (2002), 287 – 296.

[8] Czumaj, A., and Vöcking, B. Tight bounds for worst-case equilibria. *Proc 13th SODA* (2002), 413 – 420.

[9] Feldmann, R., Gairing, M., Lücking, T., Monien, B., and Rode, M. Nashification and the coordination ratio for a selfish routing game. *Proc 30th ICALP* (2003), 514 – 526.

[10] Finn, G., and Horowitz, E. A linear time approximation algorithm for multiprocessor scheduling. *BIT 19* (1979), 312 – 320.

[11] Gairing, M., Lücking, T., Mavronicolas, M., and Monien, B. The price of anarchy for polynomial social cost. *Proc 29th MFCS* (2004), 574 – 585.

[12] Gairing, M., Lücking, T., Mavronicolas, M., Monien, B., and Rode, M. Nash equilibria in discrete routing games with convex latency functions. *Proc 31st ICALP* (2004), 645 – 657.

[13] Gairing, M., Monien, B., and Tiemann, K. Selfish routing with incomplete information. *Proc 17th SPAA* (2005), 203 – 212.

[14] Hochbaum, D., and Shmoys, D. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM Journal on Computing 17*, 3 (1988), 539 – 551.

[15] Immorlica, N., Li, L., Mirrokni, V., and Schulz, A. Coordination mechanisms for selfish scheduling. *Proc 1st WINE* (2005), 55 – 69.

[16] Kontogiannis, S., and Spirakis, P. Atomic selfish routing in networks: A survey. *Proc 1st WINE* (2005), 989 – 1002.

[17] Koutsoupias, E., and Papadimitriou, C. Worst-case equilibria. *Proc 17th STACS* (1999), 404 – 413.

[18] Mavronicolas, M., Panagopoulou, P., and Spirakis, P. A cost mechanism for fair pricing of resource usage. *Proc 1st WINE* (2005), 210–224.

[19] Mitzenmacher, M., and Upfal, E. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[20] Roughgarden, T. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences 67*, 2 (2003), 341–364.

[21] Roughgarden, T., and Tardos, E. How bad is selfish routing? *Journal of the ACM 49*, 2 (2002), 236 – 259.

[22] Scharbrodt, M., Schickinger, T., and Steger, A. A new average case analysis for completion time scheduling. *Proc 34th STOC* (2002), 170 – 178.

[23] Souza, A., and Steger, A. The expected competitive ratio for weighted completion time scheduling. In *Proc 21st STACS* (2004), Springer Verlag, pp. 620 – 631.

[24] Vredeveld, T. *Combinatorial approximation algorithms. Guaranteed versus experimental performance*. PhD thesis, Technische Universiteit Eindhoven, 2002.