Dynamic Inefficiency: Anarchy without Stability

Noam Berger[1], Michal Feldman[2], Ofer Neiman[3], and Mishael Rosenthal[4]

[1] Einstein Institute of Mathematics, Hebrew University of Jerusalem. Email: berger@math.huji.ac.il.
[2] School of Business Administration and Center for Rationality, Hebrew University of Jerusalem. Email: mfeldman@huji.ac.il.
[3] Princeton university and Center for Computational Intractability. Email: oneiman@princeton.edu.
[4] School of Engineering and Computer Science, Hebrew University of Jerusalem. Email: mishael@cs.huji.ac.il.

Abstract. The price of anarchy [16] is by now a standard measure for quantifying the inefficiency introduced in games due to selfish behavior, and is defined as the ratio between the optimal outcome and the worst Nash equilibrium. However, this notion is well defined only for games that always possess a Nash equilibrium (NE). We propose the *dynamic inefficiency* measure, which is roughly defined as the average inefficiency in an infinite best-response dynamic. Both the price of anarchy [16] and the price of sinking [9] can be obtained as special cases of the dynamic inefficiency measure. We consider three natural best-response dynamic rules — *Random Walk* (RW), *Round Robin* (RR) and *Best Improvement* (BI) — which are distinguished according to the order in which players apply best-response moves.

In order to make the above concrete, we use the proposed measure to study the job scheduling setting introduced in [3], and in particular the scheduling policy introduced there. While the proposed policy achieves the best possible price of anarchy with respect to a pure NE, the game induced by the proposed policy may admit no pure NE, thus the *dynamic inefficiency* measure reflects the worst case inefficiency better. We show that the dynamic inefficiency may be arbitrarily higher than the price of anarchy, in any of the three dynamic rules. As the dynamic inefficiency of the RW dynamic coincides with the *price of sinking*, this result resolves an open question raised in [3].

We further use the proposed measure to study the inefficiency of the Hotelling game and the facility location game. We find that using different dynamic rules may yield diverse inefficiency outcomes; moreover, it seems that no single dynamic rule is superior to another.

1 Introduction

Best-response dynamics are central in the theory of games. The celebrated Nash equilibrium solution concept is implicitly based on the assumption that players follow best-response dynamics until they reach a state from which no player can improve her utility. Best-response dynamics give rise to many interesting questions which have been extensively studied in the literature. Most of the focus concerning best response dynamics has been devoted to convergence issues, such as whether best-response dynamics converge to a Nash equilibrium and what is the rate of convergence.

Best-response dynamics are essentially a large family of dynamics, which differ from each other in the order in which turns are assigned to players [5]. It is well known that the order of the players' moves is crucial to various aspects, such as convergence rate to a Nash equilibrium [6]. Our main goal is to study the effect of the players' order on the obtained (in)efficiency of the outcome.

The most established measure of inefficiency of games is the Price of Anarchy (PoA) [13,16], which is a worst-case measure, defined as the ratio between the worst Nash equilibrium (NE) and the social optimum (with respect to a well-defined social objective function), usually defined with respect to pure strategies. The PoA essentially measures how much the society suffers from players who maximize their individual welfare rather than the social good. The PoA has been evaluated in many settings, such as selfish routing [19,18], job scheduling [13,5,8], network formation [7,1,2], facility location [20], and more. However, this notion is well defined only in settings that are guaranteed to admit a NE.

One approach that has been taken with respect to this challenge, in cases where agents are assumed to use pure strategies, is the introduction of a *sink equilibrium* [9]. A sink equilibrium is a strongly connected component with no outgoing edges in the *configuration graph* associated with a game. The configuration graph has a vertex set associated with the set of pure strategy profiles, and its edges correspond to best-response moves. Unlike pure strategy Nash equilibria, sink equilibria are guaranteed to exist. The social value associated with a sink equilibrium is the expected social value of the stationary distribution of a random walk on the states of the sink. The *price of sinking* is the equivalence of the price of anarchy measure with respect to sink equilibria.

Indeed, the notion of best response lies at the heart of many of the proposed solution concepts, even if just implicitly. The implicit assumption that underlies the notion of social value associated with a sink equilibrium is that in each turn a player is chosen uniformly at random to perform her best response. However, there could be other natural best-response dynamics that arise in different settings.

In this paper, we focus on the following three natural dynamic rules: (i) **random walk (RW)**, where a player is chosen uniformly at random; (ii) **round robin (RR)**, where players play in a cyclic manner according to a pre-defined order, and (iii) **best improvement (BI)**, where the player with the current highest (multiplicative) improvement factor plays. Our goal is to study the effect of the players' order on the obtained (in)efficiency of the outcome.

To this end, we introduce the concept of *dynamic inefficiency* as an equivalent measure to the price of anarchy in games that may not admit a Nash equilibrium or in games in which best-response dynamics are not guaranteed to converge to a

[5] In fact, best-response dynamics may be asynchronous, but in this paper we restrict attention to synchronous dynamics.

Nash equilibrium. Every dynamic rule D chooses (deterministically or randomly) a player that performs her best response in each time period. Given a dynamic D and an initial configuration u, one can compute the expected social value obtained by following the rules of dynamic D starting from u. The dynamic inefficiency in a particular game is defined as the expected social welfare with respect to the worst initial configuration. Similarly, the dynamic inefficiency of a particular family of games is defined as the worst dynamic inefficiency over all games in the family. Note that the above definition coincides with the original price of anarchy measure for games in which best-response dynamics always converge to a Nash equilibrium (e.g., in congestion [17] and potential [14] games) for every dynamic rule. Similarly, the dynamic inefficiency of the RW dynamic coincides with the definition of the price of sinking. Thus, we find the dynamic inefficiency a natural generalization of well-established inefficiency measures.

1.1 Our Results

We evaluate the dynamic inefficiency with respect to the three dynamic rules specified above, and in three different applications, namely non-preemptive job scheduling on unrelated machines [3], the Hotelling model [10], and facility location [15]. Our contribution is conceptual as well as technical. First, we introduce a measure which allows us to evaluate the inefficiency of a particular dynamic even if it does not lead to a Nash equilibrium. Second, we develop proof techniques for providing lower bounds for the three dynamic rules. In what follows we present our results in the specific models.

Job scheduling (Section 3) We consider job scheduling on unrelated machines, where each of the n players controls a single job and selects a machine among a set of m machines. The assignment of job i on machine j is associated with a processing time that is denoted by $p_{i,j}$. Each machine schedules its jobs sequentially according to some non-preemptive scheduling policy (i.e., jobs are processed without interference, and no delay is introduced between two consecutive jobs), and the cost of each job in a given profile is its completion time on its machine. The social cost of a given profile is the maximal completion time of any job (known as the makespan objective).

Machines' ordering policies may be local or strongly local. A local policy considers only the parameters of the jobs assigned to it, while a strongly local policy considers only the processing time of the jobs assigned to it on itself (without knowing the processing time of its jobs on other machines). Azar et. al. [3] showed that the PoA of any local policy is $\Omega(\log m)$ and that the PoA of any strongly local policy is $\Omega(m)$ (if a Nash equilibrium exists). Ibarra and Kim [11] showed that the shortest-first (strongly local) policy exhibits a matching $O(m)$ bound, and Azar et. al. [3] showed that the inefficiency-based (local) policy (defined in Section 3) exhibits a matching $O(\log m)$ bound.

We claim that there is a fundamental difference between the last two results. The shortest-first policy induces a potential game [12,14]; thus best-response dynamics always converge to a pure Nash equilibrium, and the PoA is an appropriate measure. In contrast, the inefficiency-based policy induces a game that does not necessarily admit a pure Nash equilibrium [3], and even if a Nash equilibrium exists, not every best-response dynamic converges to a Nash equilibrium. Consequently, the realized inefficiency of the last policy may be much higher than the bound provided by the price of anarchy measure.

We study the dynamic inefficiency of the inefficiency based policy with respect to our three dynamic rules. We show a lower bound of $\Omega(\log\log \mathbf{n})$ for the dynamic inefficiency of the RW rule. This bound may be arbitrarily higher[6] than the price of anarchy, which is bounded by $O(\log \mathbf{m})$. This resolves an open question raised in [3]. For the BI and RR rules, we show in the full version even higher lower bounds of $\Omega(\sqrt{n})$ and $\Omega(n)$, respectively.

Hotelling model Hotelling [10] devised a model where customers are distributed evenly along a line, and there are m strategic players, each choosing a location on the line, with the objective of maximizing the number of customers whose location is closer to her than to any other player. It is well known that this game admits a pure Nash equilibrium if and only if the number of players is different than three. This motivates the evaluation of the dynamic inefficiency measure in settings with three players. The social objective function we consider here is the minimal utility over all players, i.e., we wish to maximize the minimal number of customers one attracts.

We show in the full version that the dynamic inefficiency of the BI rule is upper bounded by a universal constant, while the dynamic inefficiency of the RW and RR rules is lower bounded by $\Omega(n)$, where n is the number of possible locations of players. Thus, the BI dynamics and the RW and RR dynamics exhibit the best possible and worst possible inefficiencies, respectively (up to a constant factor). In contrast to the BI dynamics, the RW and RR dynamics are *configuration-oblivious* (i.e., the next move is determined independently of the current configuration).

Facility location In facility location games a central designer decides where to locate a public facility on a line, and each player has a single point representing her ideal location. Suppose that the cost associated with each player is the squared distance of her ideal location to the actual placement of the facility, and that we wish to minimize the average cost of the players. Under this objective function the optimal location is the mean of all the points. However, for any chosen location, there will be a player who can decrease her distance from the chosen location by reporting a false ideal location. Moreover, it is easy to see that if the players know in advance that the mean point of the reported locations is chosen, then every player who is given a turn can actually transfer the location to be exactly at her ideal point. Thus, unless all players are located at exactly the same point, there will be no Nash equilibrium. Our results, that appear in

[6] Note the parameter n; i.e., number of players, versus the parameter m; i.e., number of machines.

the full version, indicate that the dynamic inefficiency of the RR and RW rules is exactly 2, while that of the BI rule is $\Theta(n)$.

2 Preliminaries

In our analysis it will be convenient to use the following graph-theoretic notation: we think of the configuration set (i.e., pure strategy profiles) as the vertex set of a *configuration graph* $G = (V, E)$. The configuration graph is a directed graph in which there is a directed edge $e = (u, v) \in E$ if and only if there is a player whose best response given the configuration u leads to the configuration v. We assume that each player has a unique best response for each vertex. A *sink* is a vertex with no outgoing edges. A *Nash Equilibrium* (NE) is a configuration in which each player is best responding to the actions of the other players. Thus, a NE is a sink in the configuration graph.

A *social value* of $S(v)$ is associated with each vertex $v \in V$. Two examples of social value functions are the social welfare function, defined as the sum of the players' utilities, and the max-min function, defined as the minimum utility of any player.

A best-response dynamic rule is a function $D : V \times \mathbb{N} \to [n]$, mapping each point in time, possibly depending on the current configuration, to a player $i \in [n]$ who is the next player to apply her best-response strategy. The function D may be deterministic or non-deterministic.

Let $P = \langle u_1, \ldots, u_T \rangle$ denote a finite path in the configuration graph (where u_i may equal u_j for some $i \neq j$). The *average social value* associated with a path P is defined as $S(P) = \frac{1}{T} \sum_{t=1}^{T} S(u_t)$. Given a tuple $\langle u, D \rangle$ of a vertex $u \in V$ and a dynamic rule D, let $\mathcal{P}_T(u, D)$ denote the distribution over the paths of length T initiated at vertex u under the dynamic rule D. The social value of the dynamic rule D initiated at vertex u is defined as

$$S(u, D) = \lim_{T \to \infty} \mathbb{E}_{P \sim \mathcal{P}_T(u,D)}[S(P)]. \tag{1}$$

While the expression above is not always well defined, in the full version of the paper we demonstrate that it is always well defined for the dynamic rules considered in this paper.

With this, we are ready to define the notion of *dynamic inefficiency*. Given a finite configuration graph $G = (V, E)$ and a dynamic rule D, the dynamic inefficiency (DI) of G with respect to D is defined as

$$\mathrm{DI}(D, G) = \max_{u \in V} \frac{\mathrm{OPT}}{S(u, D)},$$

where $\mathrm{OPT} = \max_{u \in V} S(u)$. That is, DI measures the ratio between the optimal outcome and the social value obtained by a dynamic rule D under the worst possible initial vertex. Finally, for a family of games $\mathcal{G}$, we define the dynamic inefficiency of a dynamic rule D as the dynamic inefficiency of the worst possible

$G \in \mathcal{G}$. This is given by

$$\mathrm{DI}(D) = \sup_{G \in \mathcal{G}} \{\mathrm{DI}(D, G)\} \ .$$

In some of the settings, social costs are considered rather than social value. In these cases, the necessary obvious adjustments should be made. In particular, $S(u, D)$ will denote the social *cost*, OPT will be defined as $\min_{u \in V} S(u)$, and the dynamic inefficiency of some dynamic D will be defined as $\mathrm{DI}(D) = \max_{u \in V} \frac{S(u,D)}{\mathrm{OPT}}$. We consider both cases in the sequel.

An important observation is that both the price of anarchy and the price of sinking are obtained as special cases of the dynamic inefficiency. In games for which every best-response dynamic converges to a Nash equilibrium (e.g., potential games [14]), the dynamic inefficiency is independent of the dynamic and is equivalent to the price of anarchy. The price of sinking [9] is equivalent to the dynamic inefficiency with respect to the RW dynamic rule.

3 Dynamic Inefficiency in Job Scheduling

Consider a non-preemptive job scheduling setting on unrelated machines, as described in the Introduction. Define the efficiency of a job i on machine j as

$$\mathrm{eff}(i, j) = \frac{p_{ij}}{\min_{k \in [m]} p_{ik}}.$$

The efficiency-based policy of a machine (proposed by [3]) orders its jobs according to their efficiency, from low to high efficiency, where ties are broken arbitrarily in a pre-defined way.

A configuration of a job scheduling game is a mapping $u : [n] \to [m]$ that maps each job to a machine. The processing time of machine j in configuration u is $\mathrm{time}_u(j) = \sum_{i \in u^{-1}(j)} p_{ij}$, and the social value function we are interested in is the *makespan* — the longest processing time on any machine, i.e., $S(u) = \max_{j \in [m]} \mathrm{time}_u(j)$.

The players are the jobs to be processed, their actions are the machines they choose to run on, and the cost of a job is its own completion time.

3.1 Random Walk Dynamic

In this section we consider the RW dynamic, where in each turn a player is chosen uniformly at random to play. The main result of this section is the establishment of a lower bound of $\Omega(\log \log n)$ for the dynamic inefficiency of job scheduling under the efficiency-based policy. This means that the inefficiency may tend to infinity with the number of jobs, even though the number of machines is constant. This result should be contrasted with the $O(\log m)$ upper bound on the price of anarchy, established by [3]. The main result is cast in the following theorem.

Theorem 1. *There exists a family of instances G_n of machine scheduling on a constant number of machines, such that*

$$\mathrm{DI}(\mathsf{RW}, G_n) \geq \Omega(\log\log n) \ ,$$

where n is the number of jobs. In particular the dynamic inefficiency is not bounded with respect to the number of machines.

Remark: The definition of the dynamic inefficiency with respect to the RW dynamic rule coincides with the definition of the *price of sinking*. Thus, the last result can be interpreted as a lower bound on the price of sinking.

The assertion of Theorem 1 is established in the following sections.

The Construction Let us begin with an informal description of the example. As the base for our construction we use an instance, given in [3], with a constant number of machines and jobs, that admits no Nash equilibrium. Then we add to it n jobs indexed by $1, \ldots, n$ and one machine, such that the n additional jobs have an incentive to run on two machines. On the first machine, denoted by W, the total processing time of all the n jobs is smaller than 2, while on the second machine T the processing time of any job is ≈ 1. Each of these jobs has an incentive to move from machine W to machine T if it has the minimal index on T, thus increasing the processing time on T. We show that the expected number of jobs on T, and hence also the expected makespan, is at least $\Omega(\log\log n)$, while the optimum is some universal constant.

Formally, there will be 5 machines, denoted by A, B, C, T, W, and $n+5$ jobs denoted by $0, 1, 2, \ldots, n$ and $\alpha, \beta, \gamma, \delta$. The following table shows the processing time for the jobs on the machines:

	A	B	C	T	W
0	4	24	3.95	25	∞
α	2	12	1.98	∞	∞
β	5	28	4.9	∞	∞
γ	20	∞	∞	∞	∞
δ	∞	∞	50	∞	∞
i	∞	∞	$\frac{1}{50i^3}$	$\sum_{j=1}^{i} \frac{1}{j^2} - \epsilon$	$\frac{1}{i^2}$

In the last row, i stands for any job $1, 2, \ldots, n$, and $\epsilon = \frac{1}{10 \cdot 2^n}$.

Useful Properties

Proposition 1. *The inefficiency policy induces the following order on the machines:*

- *On machine A the order is $(\gamma, \alpha, 0, \beta)$.*
- *On machine B the order is $(\beta, \alpha, 0)$.*
- *On machine C the order is $(\delta, \alpha, \beta, 0, 1, 2, \ldots, n)$.*

- *On machine T the order is $(0, 1, 2, \ldots, n)$.*
- *On machine W the order is $(1, 2, \ldots, n)$.*

Proof. On machine C every job has efficiency 1; hence given any tie-breaking rule between jobs of equal efficiency we let δ be the one that runs first[7] (the rest of the ordering is arbitrary). On the other machines, the order follows from straightforward calculations.

Note that no job except δ has an incentive to move to machine C, and job γ will always be in machine A. The possible configurations for jobs $0, \alpha, \beta$ are denoted by XYZ; for instance, ABB means that job 0 is on machine A and jobs α, β are on machine B. We shall only consider configurations in G that have incoming edges, and in this example there are 8 such configurations among the possible configurations for jobs $0, \alpha, \beta$. The transitions are

$$BAA \rightarrow BBA \rightarrow ABA \rightarrow ABB \rightarrow AAB \rightarrow TAB \rightarrow TAA$$

From state TAA we can go either to TBA or back to BAA. From TBA the only possible transition will take us back to the state ABA. These are all the possible transitions; hence there is no stable state. At any time a job i for $i > 0$ has an incentive to be in machine T only if it has the *minimal* index from all the jobs in T. We want to show that in the single (non-trivial) strongly connected component of G, the expected number of jobs in machine T is at least $\Omega(\log \log n)$.

Dynamic Inefficiency - Lower Bound It can be checked that any configuration for jobs $1, 2, \ldots, n$ is possible among machines T, W. Consider the stationary distribution π over this strongly connected component in G. Let $T(i) \subseteq V$ be the set of configurations in which job i is scheduled on machine T, and let $p_n(i) = \sum_{v \in T(i)} \pi(v)$ denote the probability that job i is on machine T. Let $p_n(\emptyset)$ be the probability that no job is on machine T.

Proposition 2. *For any $n > m \geq i$, $p_n(i) = p_m(i)$. This is because the incentives for jobs $\alpha, \beta, \gamma, \delta, 0, 1, \ldots, i$ are not affected by the presence of any job j for $j > i$.*

Using this proposition we shall omit the subscript and write only $p(i)$. The following claim suggests we should focus our attention on how often machine T is empty.

Claim. For any $n \geq 1$, $p_n(\emptyset) \leq p(n)$.

[7] We can also handle tie-breaking rules that consider the length of the job. For instance, if shorter jobs were scheduled first in case of a tie, we would split job δ into many small jobs.

Proof. Job n has an incentive to move to T if and *only if* the configuration is such that T is empty. The probability of job n to get a turn to move is $1/(n+5)$. Hence the probability that job n is in T at some time t is equal to the probability that for some $i \geq 0$ job n entered T at time $t-i$ (i.e., machine T was empty and n got a turn to play) and stayed there for i rounds. The probability that job n stayed in machine T for i rounds is at least $p_i = \left(1 - \frac{1}{n+5}\right)^i$. We conclude that

$$p(n) \geq \frac{p_n(\emptyset)}{n+5} \sum_{i=0}^{\infty} p_i = p_n(\emptyset).$$

The main technical lemma is the following:

Lemma 1. *There exists a universal constant c such that for any $n > 1$, $p_n(\emptyset) \geq \frac{c}{n \log n}$.*

Let us first show that given this lemma we can easily prove the main theorem:

Proof (Proof of Theorem 1). In the single (non-trivial) strongly connected component of G, the expected number of jobs in machine T (with respect to a RW) is at least

$$\sum_{i=1}^{n} p(i) \geq c \sum_{i=2}^{n} \frac{1}{i \log i} \geq (c/2) \log \log n \ .$$

However, there is a configuration in which every job completes execution in time at most 50;[8] hence $\mathrm{OPT}(G) \leq 50$. We conclude that the dynamic inefficiency for G is at least $\Omega(\log \log n)$.

In what follows, we establish the assertion of Lemma 1.

Proof (Proof of Lemma 1). Let $Y \subseteq V$ be the set of configurations in which T is empty, and let t be the expected number of steps between visits to configurations in Y. We have that $p_n(\emptyset) = \frac{1}{t}$, and need to prove that $t \leq O(n \log n)$. We start with a claim on rapidly decreasing integer random variables.

Claim. Fix some $n \in \mathbb{N}$, $n > 1$. Let $x_1, x_2, \ldots$ be random variables getting non-increasing values in $\mathbb{N}$, such that $\mathbb{E}[x_1] \leq n/2$ and for every $i > 0$, $\mathbb{E}[x_{i+1} \mid x_i = k] \leq k/2$; then if we let s be a random variable which is the minimal index such that $x_s = 0$, then $\mathbb{E}[s] \leq \log n + 2$.

Proof. First we prove by induction on i that $\mathbb{E}[x_i] \leq \frac{n}{2^i}$. This holds for $i = 1$; assume it is true for i and then prove for $i + 1$. By the rule of conditional probability,

$$\begin{aligned}
\mathbb{E}[x_{i+1}] &= \sum_{j \geq 0} \Pr[x_i = j] \cdot \mathbb{E}[x_{i+1} \mid x_i = j] \\
&\leq \sum_{j \geq 0} \Pr[x_i = j] \cdot (j/2) = \mathbb{E}[x_i]/2 \leq \frac{n}{2^{i+1}} \ .
\end{aligned}$$

[8] For instance, if all jobs $1, 2, \ldots, n$ are on machine W, then job i will finish in time $\sum_{j=1}^{i} \frac{1}{j^2} < 2$

Note that if $x_i = 0$ for some i, then it must be that $x_j = 0$ for all $j > i$. Now for any integer $i > 0$,

$$\begin{aligned}\Pr[s = \log n + i] &\leq \Pr[s > \log n + i - 1] \\ &= \Pr[x_{\log n + i - 1} \geq 1] \\ &\leq \mathbb{E}[x_{\log n + i - 1}] \leq 1/2^{i-1}\end{aligned}$$

the second inequality is a Markov inequality. We conclude that

$$\begin{aligned}\mathbb{E}[s] = \sum_{i=1}^{\log n} i \cdot \Pr[s = i] + \sum_{i=\log n+1}^{\infty} i \cdot \Pr[s = i] \\ \leq \log n + \sum_{i=\log n+1}^{\infty} i/2^{i-1} \leq \log n + 2 \ .\end{aligned}$$

Claim. Assume that we are in a configuration u in which job $i \in \{0, 1, \ldots, n\}$ is in machine T. The expected time until we reach a configuration in which job i is not in machine T is at most $O(n)$.

Proof. First note that $p(0) = c'$ for some constant $0 < c' < 1$, this is because jobs $0, \alpha, \beta, \gamma, \delta$ are not affected at all by the location of any job from $1, 2, \ldots, n$, and hence when they get to play they will follow one of the two cycles shown earlier, which implies that in a constant fraction c' of the time job 0 will be in machine T, and in the other $1 - c'$ fraction it will be on another machine.

Note that job i will have incentive to leave T if job 0 is in machine T when i gets its turn to play. Let $q(i)$ be the event that i gets a turn to play (which is independent of the current configuration), $T(i)$ denotes the event that job i is in machine T, then we have that the probability that job i will leave machine T is at least

$$\begin{aligned}\Pr[q(i) \wedge T(0) \mid T(i)] &= \Pr[q(i) \mid T(0) \wedge T(i)] \cdot \Pr[T(0) \mid T(i)] \\ &= Pr[q(i)] \cdot \Pr[T(0)] = \frac{c'}{n+5}\end{aligned}$$

Now the expected time until job i will leave is at most $\frac{1}{\Pr[q(i) \wedge T(0) | T(i)]} \leq O(n)$.

Claim. Let $\ell = \ell(t)$ be a random variable that is the minimal job in T at time t, and let x be the random variable that is the next job that enters T. Then $\mathbb{E}[x \mid \ell = m] \leq m/2$.

Proof. The jobs that have an incentive to enter machine T are $0, 1, \ldots, m - 1$. It is easy to see that for any job $i \in \{1, \ldots, m - 1\}$, $\Pr[x = i \mid \ell = m] \leq \frac{1}{m-1}$ (note that job 0 has also some small probability to enter T, but it does not contribute to the expectation). Now

$$\mathbb{E}[x \mid \ell = m] = \sum_{i=1}^{m-1} i \cdot \Pr[x = i \mid \ell = m] \leq m/2 \ .$$

Let $y \in Y$ be any configuration in which T is empty. We define a series of random variables $x_1, x_2, \dots$ as follows. Let x_1 be the index of the first job to enter T, and let x_i be the maximal index of a job in T when x_{i-1} leaves T (and 0 if T is empty). Note that when $x_{i-1} = k$ is the maximal index of a job in T, no job with index larger than k has an incentive to move to T; hence given that $x_{i-1} = k$ it must be that $x_i \leq k$. Let s be the minimal index such that $x_s = 0$; then either machine T is empty or by Claim 3.1 is expected to become empty in $O(n)$ steps. It is easy to see that $\mathbb{E}[x_1] \leq n/2$, and the tricky part is to bound the expected value of x_i.

Claim. $\mathbb{E}[x_i | x_{i-1} = k] \leq k/2$.

Proof. Fix some k such that $x_{i-1} = k$. Consider the time $\bar{t}$ in which job k became the maximal job in T, and consider the time $t' < \bar{t}$ in which job k moved to machine T and did not leave until time $\bar{t}$. In time t' it must be that no job i, for $i < k$, was in machine T, since job k had an incentive to move to T. In particular, x_i is not in T at time t'.

Consider the time $t'' > t'$ in which x_i enters T, and stays until job k leaves. In time t'' the minimal job in T is at most k; hence by Claim 3.1 we have that $\mathbb{E}[x_i \mid x_{i-1} = k] \leq k/2$.

Consider the random variables $x_1, x_2, \dots, x_s$. By Claim 3.1 they satisfy the conditions of Claim 3.1; hence $\mathbb{E}[s] \leq \log n + 2$. By Claim 3.1 we have that the expected time until the maximal job leaves T is at most $O(n)$. We conclude that in expectation after $O(n \log n)$ steps the maximal job in T will be 0, and within additional $O(n)$ steps machine T is expected to become empty. This concludes the proof.

4 Conclusion

We study the notion of dynamic inefficiency, which generalizes well-studied notions of inefficiency such as the price of anarchy and the price of sinking, and quantify it in three different applications. In games where best-response dynamics are not guaranteed to converge to an equilibrium, dynamic inefficiency reflects better the inefficiency that may arise. It would be of interest to quantify the dynamic inefficiency in additional applications. It is of a particular interest to study whether there exist families of games for which one dynamic rule is always superior to another.

In the job scheduling realm, our work demonstrates that the inefficiency based policy suggested by [3] suffers from an extremely high price of sinking. A natural open question arises: is there a local policy that always admits a Nash equilibrium and exhibits a PoA of $o(m)$? (recall that m is the number of machines). Alternatively, is there a local policy that exhibits a dynamic inefficiency of $o(m)$ for some best response dynamic rule? Recently, Caragiannis [4] found a *preemptive* local policy that always admits a Nash equilibrium and has a PoA of $O(\log \mathbf{m})$. However, we are primarily interested in non-preemptive policies.

References

1. S. Albers, S. Elits, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash equilibria for a network creation game. In *seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006.
2. E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-Optimal Network Design with Selfish Agents. In *STOC'03*, 2003.
3. Y. Azar, K. Jain, and V. Mirrokni. (almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 323–332, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.
4. I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 815–824, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.
5. A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *SODA*, pages 413–420, 2002.
6. E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *ICALP*, pages 502–513, 2003.
7. A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker. On a network creation game. In *ACM Symposium on Principles of Distributed Computing (PODC)*, 2003.
8. M. Feldman and T. Tamir. Conflicting congestion effects in resource allocation games. In *Proceedings of the 4th International Workshop on Internet and Network Economics*, WINE '08, pages 109–117, Berlin, Heidelberg, 2008. Springer-Verlag.
9. M. Goemans, V. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 142–154, Washington, DC, USA, 2005. IEEE Computer Society.
10. H. Hotelling. Stability in competition. *Economic Journal*, 39(53):41–57, 1929.
11. O. H. Ibarra and C. E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM*, 24:280–289, 1977.
12. N. Immorlica, L. Li, V. S. Mirrokni, and A. S. Schulz. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410:1589–1598, April 2009.
13. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.
14. D. Monderer and L. S. Shapley. Potential Games. *Games and Economic Behavior*, 14:124–143, 1996.
15. Herve Moulin. On strategy-proofness and single-peakedness. *Public Choice*, 35:437–455, 1980.
16. C. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of 33rd STOC*, pages 749–753, 2001.
17. R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
18. T. Roughgarden. The price of anarchy is independent of the network topology. In *STOC'02*, pages 428–437, 2002.
19. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236 – 259, 2002.
20. A. R. Vetta. Nash equilibria in competitive societies with applications to facility location, traffic routing and auctions. In *Symposium on the Foundations of Computer Science (FOCS)*, pages 416–425, 2002.