How to Route and Tax Selfish Unsplittable Traffic*

Vincenzo Auletta
auletta@dia.unisa.it

Roberto De Prisco
robdep@dia.unisa.it

Paolo Penna
penna@dia.unisa.it

Pino Persiano
giuper@dia.unisa.it

Dipartimento di Informatica ed Applicazioni "Renato M. Capocelli"
Università di Salerno
via S. Allende 2, I-84081 Baronissi (SA), Italy

ABSTRACT

We study the problem of assigning unsplittable traffic to a set of m links so to minimize the maximum link congestion (i.e., the makespan). We consider the case of *selfish agents* owning pieces of the traffic. In particular, we introduce a variant of the model by Koutsopias and Papadimitriou [1999] in which owners of the traffic cannot directly choose which link to use; instead, the assignment is performed by a *scheduler*. The agents can manipulate the scheduler by reporting false information regarding the size of each piece of unsplittable traffic.

We provide upper and lower bounds on the approximation achievable by *mechanisms* that induce a Nash equilibrium when all agents report their true values.

For the case of each agent owning one job, our positive results for m identical links show the effectiveness of introducing such a scheduler since, in this case, $(1+\epsilon)$-approximate solutions are guaranteed in polynomial time. In contrast, the result by Koutsopias and Papadimitriou [1999] shows that, without payments and allowing selfish routing, Nash equilibria yield (in the worst case) $\Omega(\frac{\log m}{\log \log m})$-approximate solutions, even for unitary weighted traffic. When links have different speeds we prove lower and upper bounds on the approximation achievable by a mechanism inducing a Nash equilibrium.

For the case of agents owning more than one job, we give mechanisms that achieve constant approximation and prove lower bounds on the approximation ratio that can be achieved by a mechanism.

*Work supported by the European Project IST-2001-33135, Critical Resource Sharing for Cooperation in Complex Systems (CRESCCO).

SPAA'04, June 27–30, 2004, Barcelona, Spain.

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Nonnumerical Algorithms and Problems—*sequencing and scheduling*

General Terms

Algorithms, Economics, Theory

Keywords

Scheduling, Selfish Routing, Nash Equilibrium, Algorithmic Mechanism Design

1. INTRODUCTION

Selfish routing games have been the subject of several studies because of their applications to situations, typical in the Internet, where different entities compete for shared resources and may act *selfishly* trying to increase their own benefits. For instance, allowing *source routing* gives the possibility to the users of choosing the best path for their own traffic (e.g., minimizing the latency of their own traffic), regardless of the overall network performances (e.g., the overall latency in the network).

A natural approach to study this kind of problems comes from micro-economics and game theory. In particular, it is common to consider so called *selfish agents*, that is, a set of "players" whose goal is to perform the "best strategy" w.r.t. their own benefits. A powerful tool for studying these problems is the well-known concept of *Nash equilibrium* (see e.g. [15]). Informally speaking, Nash equilibria correspond to those "stable" situations for which no player has an incentive to *unilaterally* change her strategy.

The seminal paper by Koutsopias and Papadimitriou [11] introduces the concept of *coordination ratio* as the measure of the loss of performance due to the lack of cooperation of selfish agents: given a global optimization function (the maximum link congestion), how bad is the *worst-case* Nash equilibria? In other words, the coordination ratio measures the "price of the anarchy." A number of papers for (variants of) the model introduced in [11] have studied the problem of characterizing, computing, and bounding the cost of Nash equilibria for the corresponding routing problem [12, 4, 8, 7, 5, 6] (see also [18, 17] for a different model considering splittable traffic on arbitrary networks).

In the attempt to cope with the negative effects of selfishly acting agents, the elegant theory of *mechanism design* (see the milestone papers by Vickrey [19], Clarke [2] and Groves [9]) has been recently applied to a number of optimization problems arising in the context of network optimization [13, 14, 1]. Intuitively, with mechanism design we aim at providing payments to the agents so to incentivate each of them to cooperate. In particular, mechanisms are applied to those problems for which (part of) the input is privately known to the agents and the strategy of an agent is to report (a possibly false) piece of information; based on the reported values, a solution is computed and each agent is *rewarded* with some amount specified by a function $p^i(\cdot)$; moreover, each agent valuates the computed solution X some amount $v^i(X)$. In order to induce cooperation, the mechanism should guarantee that the *net profit*, or *utility*, of every agent i, given by the sum of the valuation and the payment, is maximized when reporting the correct information. So, mechanism design can be seen as the "reverse" problem of designing a game (payments + algorithm) such that the designers' goal matches with the agents' preferences [16].

In this paper we investigate the routing problem introduced in [11] from a different prospective: instead of allowing selfish routing, we consider a *scheduler* which allocates the traffic (i.e., jobs) on the links (i.e., parallel related machines). In this model, agents cannot directly choose the link to be put on, nor refuse the allocation chosen by the scheduler. However, they may still manipulate the system by reporting false information on the size of their own job(s). A detailed description of the model is provided in Sect. 1.1. We investigate the benefits of replacing the "anarchic" policy of having agents choosing their own route with a scheduling algorithm which, combined with a suitable payment function, yields a mechanism inducing the agents' to report the correct information. We measure such benefits by considering how good the makespan of the solution computed by the mechanism can be w.r.t. the optimal one. So, our problem can be seen as the problem of scheduling *selfish jobs* on parallel related machines, as opposed to the problem of scheduling (non-selfish) jobs on selfish machines investigated in [1].

1.0.0.1 Roadmap.

In Section 1.1 we formally define our model and the basic concepts/definitions used throughout the paper. We discuss our results in Section 1.2. Section 2 provides a necessary condition for the existence of a truthful mechanism and related consequences on the existence of truthful approximation mechanisms. Sections 3-4 provide a deterministic and a randomized mechanism for identical and non-identical machine speeds, respectively. Finally, in Sect. 5 we investigate the case of agents owning more than one job.

We remark that due to space limitation we have moved some of the proofs to the Appendix.

1.1 The Model

We consider (selfish) routing over a network consisting of m parallel links through which n (selfish) users want to route their traffic trying to minimize their *own* latency. The goal is to minimize the maximum latency. The case in which each user owns one unsplittable piece of traffic can be formulated as a scheduling problem as follows (the generalization to more than one piece of traffic per agent is given in Sect. 5).

We have to schedule a set of jobs with weights $t = (t_1, t_2, \ldots, t_n)$ on a set of m related machines with speeds $(s_1, s_2, \ldots, s_m)$ and we wish to minimize the *makespan*. Each job represents a certain amount of traffic owned by a *selfish agent*. The strategy of each agent is to declare a value b_i representing the weight of her job. Based on the reported values $b = (b_1, \ldots, b_n)$, the scheduler runs an *allocation* algorithm A to compute an assignment $A(b)$.

Naturally, agents would prefer to be assigned to faster and less loaded machines, in order to have their *own* jobs to be processed faster. In particular, a valuation function $v^i(A(b)|t)$ expresses how much agent i "likes" the assignment $A(b)$, given the job weights t.

To stress that agent i can only change the i-th value b_i of b, we distinguish the declared values of the other agents by introducing the notation $b_{-i} := (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$. For any value x, we denote by

$$(x, b_{-i}) := (b_1, b_2, \ldots, b_{i-1}, x, b_{i+1}, \ldots, b_n).$$

We remark that the valuation function of agent i depends on *both* its job weight t_i and the weights of the other jobs t_{-i}.

We proceed as in the KP model [11] by assuming that machines process jobs in round-robin fashion, i.e., all jobs assigned to the same machine finish roughly at the same time. We let $A^i(b)$ denote the machine to which job i is allocated according to $A(b)$ and by $s^i_A(b)$ its speed.[1] Also, we let $w^i_A(b|t_{-i})$ denote the sum of the true weights of the jobs assigned to machine $A^i(b)$, excluding job i.

We thus define agent's i valuation of solution $A(b)$ as

$$v^i(A(b)|t) := -\frac{t_i + w^i(A(b)|t_{-i})}{s^i_A(b)}.$$

Hence, agent i may report $b_i \neq t_i$ so to induce algorithm A to compute a solution that i likes better, i.e., with higher utility. In this case, there is no guarantee that a c-approximation algorithm A returns a solution $A(b)$ which is a c-approximate solution for the *true input* t. In order to incentivate the agents to report their *true* values t, we define suitable *payment functions* $p^i_A(b)$ which determine the amount of money that each agent i receives according to the declared values b. These functions are *known to the agents*, as well as the allocation algorithm A. A pair (A, p_A) is termed in the literature *mechanism*. So, each agents is now willing to maximize her *utility* (or net profit) $u^i(\cdot)$ which is defined as follows:

$$u^i(A(b)|t) := p^i_A(b) + v^i(A(b)|t).$$

We assume that agents are selfish but rational, i.e., they declare a false value only if they can obtain a strictly larger utility. Nash equilibria guarantee that no agent has an incentive to *unilaterally* change her strategy (see, for example, [15]). In particular, we would like to guarantee that truthtelling is a Nash equilibrium:

DEFINITION 1. *A mechanism $M = (A, p_A)$ is* equilibria-truthful *if, for every agent i, it holds that for all t and for all b_i*

$$u^i(A(t_i, t_{-i})|t) \geq u^i(A(b_i, t_{-i})|t).$$

[1] A more appropriate notation would be $s^i_{A^i(b)}$ but for the sake of readability we use $s^i_A(b)$.

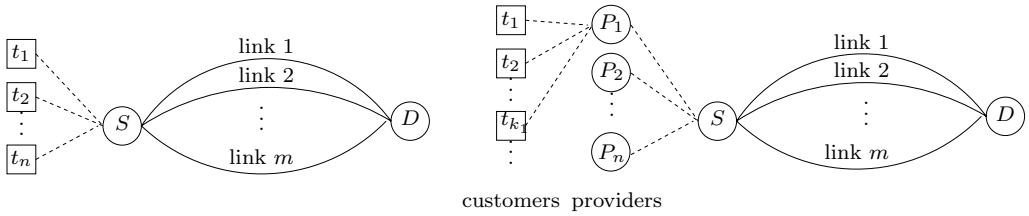

Figure 1: The model. On the left, a set of n users wants to route their traffic from S to D choosing one of the m available links (i.e., disjoint paths). A scheduler resides on node S and, based on the declared traffic weights, assigns a link to every user. On the right, a set of n providers connects the users to node S towards the destination node D; each provider P_i has a pull of customers; the faster traffic t_{i_j} is processed, the more money provider P_i receives from the corresponding customer.

Equilibria-truthful mechanisms guarantee that, if all agents report their job size correctly, then none has an incentive to report a different value. A stronger notion is that of requiring that truth-telling is the best strategy for a player i regardless of the strategy adopted by other players:

DEFINITION 2. *A mechanism $M = (A, p_A)$ is* truthful with dominant strategies *(in short,* truthful*) if, for every agent i, for every reported values $b_{-i} := (b_1, \ldots, b_{i-1}, b_{i+1})$ of the other agents, and for every b_i*

$$u^i(A(t_i, b_{-i})|t) \geq u^i(A(b_i, b_{-i})|t). \quad (1)$$

It is worth observing that the fact that a mechanism $M = (A, p_A)$ is (equilibria-)truthful does *not* imply that the mechanism $M' = (A', p_A)$ is (equilibria-)truthful as well.

According to our terminology, the cost of solution $A(b)$, that is the makespan, with respect to the input t is

$$\mathsf{cost}(A(b)|t) := \max_i \left\{ \frac{t_i + w^i(A(b)|t_{-i})}{s^i_A(b)} \right\} = \max_i \left\{ v^i(A(b)|t) \right\}.$$

Observe that, in our model, we assume that the mechanism is not able to *verify* whether the agent is reporting the real weight b_i or a different one. It can only compute the allocation and the payments but it cannot verify the real cost of the solution. Indeed, in practice, it may be too expensive to keep track of the actual amount of traffic sent by each user, thus preventing from the possibility of checking, for instance, whether $b_i < t_i$. Moreover, a user may report $b_i > t_i$ and actually send an amount of traffic equal to b_i, by just padding the original traffic with some "fake" traffic up to the declared value b_i.

In the sequel, for the sake of readability, we will sometimes omit b_{-i}, t_{-i} and A in the definitions above and simply use $s^i(b_i)$, $w^i(b_i)$, $p^i(b_i)$, and $u^i(b_i|t)$. We also denote $\mathsf{cost}(A(b)|t)$ simply as $\mathsf{cost}(A, b)$.

1.2 Our Results

In this work we characterize the existence of approximation mechanisms for our problem depending on (the combination of) the following factors:

– The ratio r between the largest and the smallest machine speeds;

– Whether we consider truthful or equilibria-truthful mechanisms;

– Whether each agent owns a single job or more than one;

– The number m of machines.

Since the problem is NP-hard even for 2 machines with identical speeds, we focus on both exponential-time and polynomial-time mechanisms. In particular, the negative results on exponential-time mechanisms show that the inapproximability of the problem does not arise because of its computational intractability but it comes from the "lack of cooperation" due to the selfish agents.

We first consider the case in which each agent owns one job. For this problem version, we provide a *necessary condition* for an algorithm A to admit a payment function $p = (p^1, \ldots, p^n)$ such that $M = (A, p)$ is equilibria-truthful. Roughly speaking, this condition states that A should assign jobs "monotonically", that is, if we increase the size of one job then this job should not be moved to a slower machine. This condition can be considered as the "dual" of the monotonicity requirement for scheduling problems involving selfish machines [1]. Clearly, the same condition must hold also for truthful mechanisms.

Based on this result, we show that, even for two machines, no equilibria-truthful mechanism can guarantee exact solutions. Notice that our result also holds for exponential-time mechanisms. Moreover, our negative results can be strengthened to prove inapproximability results for equilibria-truthful mechanisms depending on r. Our results for agents owning one job are summarized in Table 1.

We then consider the case of m machines with *identical speeds* (for which the necessary condition above always holds) and we show that, for every algorithm A there exists a payment p_A such that $M = (A, p_A)$ is equilibria-truthful. This implies the existence of deterministic polynomial-time $(1 + \epsilon)$-approximation equilibria-truthful mechanisms, for any m and for any $\epsilon > 0$. Moreover, it is possible to achieve exact solutions if we use equilibria-truthful mechanisms running in non-polynomial time. In Sect. 6 we briefly discuss an application of this result to a stronger model in which the scheduler, based on the reported values, can only suggest a link to each of the agents, but cannot avoid them to perform selfish routing after they received the payment.

It is then natural to ask whether a similar result does apply to (the stronger notion of) truthful mechanisms. We give a negative answer and show that no (exponential-time) truthful exact mechanism exists even for two machines with identical speeds.

Finally, we provide a randomized mechanism for $m = 2$ which is equilibria-truthful w.r.t. the expected utility of the agents and whose approximation ratio is r. This mechanism exploits the payment schemes for identical machines suitably combined with a randomized allocation.

We then turn our attention to the case of agents owning more than one job. This is a natural extension of the

KP model [11] motivated by the scenario in which users are first gathered together into n disjoint sets corresponding to n (selfishly acting) providers (see Fig. 1), and each user pays an amount of money to her provider depending on the experienced latency (see Sect. 5 for a more detailed discussion of the model). We first show that no 7/6-approximate solution can be achieved even when considering exponential-time truthful mechanisms for the case of two identical machines and at most two jobs per agent. This result contrasts with the $(1+\epsilon)$-approximate mechanism for identical speeds, when each agent owns one job. Motivated by this negative result, we turn our attention to truthful approximate mechanisms and give upper and lower bounds on the approximation ratio of such mechanisms (see Table 2). Some of our positive results are obtained via new polynomial-time approximation algorithms which can be combined with suitable payment functions so to obtain equilibria-truthful mechanisms achieving the same approximation ratio.

1.2.0.2 Related work.

In our work we consider the KP model [11] where traffic is unsplittable and each agent can choose her link. In the KP model, the congestion for the case of m links of equal speed can be $\Omega(\frac{\log m}{\log\log m})$ times worse than the optimal. In contrast one of our results shows that $(1+\epsilon)$ approximation is possible if the allocation is computed by a scheduler.

Our work is similar in spirit to the works [6, 5] in that both of them consider the issue of designing a scheduler for selfish unsplittable traffic in the KP model [11]. However, both [6, 5] assume that the scheduler is provided with the correct traffic weights and the scheduler must compute an assignment which is a Nash equilibrium. In [6] the authors show how to obtain pure Nash equilibria whose cost is $(1+\epsilon)$-times the optimum. We prove instead that, in our setting, $(1+\epsilon)$-approximate solutions cannot be obtained at all, even when considering exponential-time algorithms. One interpretation of the two results is that, from the network designer/manager point of view, it is better to have jobs of *known weight* which are allowed to move from a precomputed assignment, than having agents that can lie about the weights of their jobs but cannot refuse the assignment of the scheduler. The work in [5] focuses on particular scheduling policies which, at every time step, allow exactly one job, if any, to move to a better machine until a configuration is reached corresponding to any Nash equilibrium; the work reported in [5] focuses on the overall number of steps, rather than the quality of the corresponding solution (this issue is also tackled in [6]). An approach similar to ours for reducing the cost of selfish routing by imposing payments on the agents has been adopted by [3] for the so-called *flow model* where each agent controls a negligible portion of the entire traffic.

2. A NECESSARY CONDITION FOR TRUTHFULNESS

In this section we provide a necessary condition for an algorithm A to admit a payment function $p = (p^1, p^2, \ldots, p^n)$ such that $M = (A, p)$ is (equilibria-) truthful and we use this condition to derive lower bounds on the approximation of any deterministic truthful mechanism.

THEOREM 3. *A deterministic mechanism $M = (A, p)$ is (equilibria-) truthful only if A satisfies $s^i_A(x) \geq s^i_A(y)$, for any agent i and for any $x > y$.*

PROOF. Let M be an (equilibria-) truthful, mechanism and let x and y be two positive integers, with $x > y$. Observe that if $t_i = x$ the utility of agent i when reporting the weight x must be greater than or equal to the utility obtained reporting y. From the definition of $u^i(\cdot)$ we obtain

$$p^i(x) - \frac{x + w^i(x)}{s^i(x)} \geq p^i(y) - \frac{x + w^i(y)}{s^i(y)}. \tag{2}$$

Similarly when $t_i = y$ we obtain

$$p^i(y) - \frac{y + w^i(y)}{s^i(y)} \geq p^i(x) - \frac{y + w^i(x)}{s^i(x)}. \tag{3}$$

By Eq. 2 we obtain

$$p^i(x) \geq p^i(y) + \frac{x + w^i(x)}{s^i(x)} - \frac{x + w^i(y)}{s^i(y)},$$

and by plugging in the right side in Eq. 3 we have

$$\begin{aligned} p^i(y) &\geq p^i(x) + \frac{y + w^i(y)}{s^i(y)} - \frac{y + w^i(x)}{s^i(x|y)} \\ &\geq p^i(y) + \frac{x + w^i(x)}{s^i(x)} - \frac{x + w^i(y)}{s^i(y)} \\ &+ \frac{y + w^i(y)}{s^i(y)} - \frac{y + w^i(x)}{s^i(x)} \\ &= p^i(y) + \frac{x - y}{s^i(x)} + \frac{y - x}{s^i(y)} \end{aligned}$$

Hence we have $\frac{x-y}{s^i(x)} + \frac{y-x}{s^i(y)} \leq 0$, thus implying $s^i(x) \geq s^i(y)$. □

We use the theorem above to prove lower bounds on the approximation ratio achievable by a mechanism. Given a set of m machines, we construct two instances t and t' where t' differs from t only for the weight of job j which is one of the jobs allocated to the fastest machine on input t. By the theorem above job j has to be allocated to the same machine also for t'. By selecting appropriately the weight t'_j we obtain that any optimal algorithm has to allocate this job to a different machine. Therefore the optimal allocation cannot be used in an equilibria-truthful mechanism.

We present our lower bounds as a function of the ratio between the largest and the smallest machine speed. We define $r := s_{\max}/s_{\min}$, where $s_{\max} := \max_{1\leq i\leq m}\{s_i\}$ and $s_{\min} := \min_{1\leq i\leq m}\{s_i\}$.

In the proof of the lower bounds we use the notation "$\mathsf{opt}(x \to s)$" to denote the minimum cost of all allocations that assign the job of weight x to the machine of speed s.

THEOREM 4. *For any two machines such that r is an integer and $r \geq \phi$, where ϕ is the golden ratio, no deterministic (equilibria-) truthful mechanism can guarantee c-approximate solutions, for $c < 1 + \frac{r-1}{2r^2-r}$.*

PROOF. Consider two machines of speed 1 and r and a set of $2r$ unitary weight jobs. Clearly, the optimum allocates these jobs with a cost bounded above by 2. Consider a (equlibria-) truthful mechanism $M = (A, p)$.

Speed ratio $r = s_{\max}/s_{\min}$	**Lower bound**	**Upper bound** (for any m and any r)
$r = 1$	no exact deterministic and truthful even exp. time [Thm 9]	exact deterministic and equilibria-truthful (non poly-time) [Thm 7] $1+\epsilon$ (deterministic, poly-time) [Cor 8]
$1 < r < 2$	$\min\left\{r, \frac{1}{2}+\frac{1}{r}\right\}$ [Thm 5] (deterministic, equilibria-truthful)	r [Thm 10] (equilibria-truth. in expect.)
$r \geq \phi \approx 1.61$	$1+\frac{r-1}{2r^2-r}$ [Thm 4] (deterministic, equilibria-truthful)	$1+\frac{m-1}{r}$ [trivial] (deterministic, truthful)

Table 1: Our Results: (in-)approximability via (equilibria-)truthful mechanisms when each agent owns a single job.

k **versus** m		**Lower bound**	**Upper bound**	
$k = 1$		1	$1+\epsilon$	[Thm 7]
$k \leq m$	$k \leq 2$	7/6 [Thm 11]	$3/2+\epsilon$	[Thm 21]
	$k > 2$	7/6	2	[Thm 17]
$k > m$	$m = 2$, k_i even for all i	4/3 [Thm 12]	$3/2+\epsilon$	[Thm 21]
	$m = 3$, k_i even for all i	4/3	$9/4+\epsilon$	[Cor 22]
	$m = 4$, k_i even for all i	4/3	$3+\epsilon$	[Cor 23]
	$m > 2$, m even	4/3	4	[Thm 19]
	$m > 2$, m odd	4/3	$4(1+\frac{1}{m})$	[Cor 20]

Table 2: Our Results: (in-)approximability via deterministic equilibria-truthful mechanisms for identical machines. k_i denotes the number of jobs owned by agent i and $k = \max_i k_i$; lower bounds apply to exponential-time mechanisms as well, while upper bounds are provided via polynomial-time mechanisms.

The case when A assigns no jobs to the faster is easy to handle: the cost of the solution is $2r$ and the approximation ratio is at least r. The theorem follows from the hypothesis that $r \geq \phi$.

So we need to consider the case when A assigns at least one job to the faster machine. Let j be the index of one such a job. Consider a new set of jobs $t' = (1, \ldots, x, 1, \ldots, 1)$, where the j^{th} job has weight $x = 2 - 1/r$, and show that A has to compute a non-optimal allocation on t'. From Theorem 3, the algorithm A must allocate job j to the faster machine. Observe that $\mathsf{opt}(t') = \mathsf{opt}(x \to 1) = \max\{x, (2r-1)/r\} = 2 - 1/r$ and $\mathsf{cost}(A, t') \geq \mathsf{opt}(x \to r)$. Moreover, if $\mathsf{opt}(x \to r)$ assigns two or more jobs to machine 1, then $\mathsf{opt}(x \to r) \geq 2$. Otherwise, the work of machine of speed r is at least $x + 2r - 2 = 2r - 1/r$, thus implying $\mathsf{opt}(x \to r) \geq 2 - 1/r^2$. Putting things together

$$\begin{aligned}\frac{\mathsf{cost}(A,t')}{\mathsf{opt}(t')} &\geq \frac{\mathsf{opt}(x \to r)}{\mathsf{opt}(x \to 1)} \geq \frac{2-1/r^2}{2-1/r} = \frac{2r^2-1}{2r^2-r} \\ &= \frac{2r^2-r+r-1}{2r^2-r} = 1 + \frac{r-1}{2r^2-r}.\end{aligned}$$

The theorem thus follows from the observation that, for the instance of all unitary jobs, at least one of them must be assigned to machine r (unless it computes a solution of cost r' the optimum). Because of Theorem 3, this job cannot be reallocated when increasing its weight to x; in this case, the ratio between the best solution satisfying this constraint and the optimum is the bound above. □

THEOREM 5. *For any $m \geq 2$ machines with $r < 2$, no deterministic (equilibria-) truthful mechanism can guarantee c-approximate solutions, for any $c < \min\left\{r, \frac{1}{2}+\frac{1}{r}\right\}$.*

PROOF. Consider m machines with speed $(1, 1, \ldots, 1, r)$ and a job sequence t consisting of $m+1$ jobs of weight 1. Any (equilibria-) truthful mechanism $M = (A, p)$ assigning no job to the fastest machine incurs in a cost of at least 2, while the optimum is $2/r$. In this case the approximation ratio is r.

Let us thus assume that A assigns at least one job to the fastest machine, and let j be the index of such a job. Let us consider a new job sequence $t' = (1, 1, \ldots, x, 1, \ldots, 1)$, where $x > 1$ is the weight of the j^{th} job. From Theorem 3, the algorithm A must allocate job j to the fastest machine. Hence, $\mathsf{cost}(A, t') \geq \mathsf{opt}(x \to r)$.

For $x = 2/r > 1$, any optimal solution that assigns x to the fastest machine must allocate at most two jobs to such a machine. We thus have to consider only two kind of solutions as shown in Fig. 2. Thus, $\mathsf{opt}(x \to r) = \min\{2, (1+2/r)/r\}$.

On the other hand, any optimal solution must assign job of weight x to a machine of speed 1, and two jobs of weight 1 to the fastest machine: this yields $\mathsf{opt}(t') = \mathsf{opt}(x \to 1) = 2/r$. Hence

$$\begin{aligned}\frac{\mathsf{cost}(A,t')}{\mathsf{opt}(t')} &\geq \frac{\mathsf{opt}(x \to r)}{\mathsf{opt}(x \to 1)} \geq \frac{\min\{2, (1+2/r)/r\}}{2/r} \\ &= \min\left\{r, \frac{1}{2}+\frac{1}{r}\right\}.\end{aligned}$$

□

Since, for $r \leq \frac{1+\sqrt{17}}{4}$, $r \leq \frac{1}{2}+\frac{1}{r}$, we obtain the following result.

COROLLARY 6. *For $m \geq 2$ machines and for any $r \leq \frac{1+\sqrt{17}}{4}$, no deterministic (equilibria-) truthful mechanism can guarantee c-approximate solutions, for any $c < r$.*

solutions	speed 1	$\cdots$	speed 1	$\cdots$	speed 1	speed r	cost
case 1	1	$\ldots$	1		1, x	1	$(1+2/r)/r$
case 2	1	$\ldots$	1, 1	$\ldots$	1	x	2

Figure 2: Assignments of jobs used in the proof of Theorem 5.

3. MECHANISMS FOR IDENTICAL MACHINE SPEEDS

In this section we consider the case in which all machines have the same speed. We start by proving that, in this particular case, any scheduling algorithm A can be used to obtain an equilibria-truthful mechanism.

THEOREM 7. *For any $m \geq 2$ machines with $r = 1$ and for any algorithm A there exists a payment function $p_A = (p_A^1, \ldots, p_A^n)$ such that $M = (A, p_A)$ is equilibria-truthful.*

PROOF. Assume that the speed of each machine is s. The mechanism pays agent i for the additional time she has to wait because of the other jobs assigned to the machine $A^i(b)$, computed w.r.t. the reported values b. Formally, we set $p^i(b_i, b_{-i}) := z^i(b_i, b_{-i})/s$, where

$$z^i(b_i, b_{-i}) := \sum_{j:A^j(b_i,b_{-i})=A^i(b_i,b_{-i}),j\neq i} b_j.$$

It is easy to see that, if $b_{-i} = t_{-i}$, then $z^i(b_i) = w^i(b_i)$ and

$$u^i(b_i|t) = p^i(b) - \frac{t_i + w^i(b_i)}{s} = -\frac{t_i}{s}.$$

Thus, if all the agents other than i are truth-telling then agent i cannot improve her utility by reporting $b_i \neq t_i$, and the mechanism is equilibria-truthful. □

Using the approximation scheme of [10], we obtain the following corollary.

COROLLARY 8. *For any $m \geq 2$, there exists a $(1+\epsilon)$-approximate polynomial-time equilibria-truthful mechanism.*

It is natural to ask whether the above result can be extended to truthful mechanisms. In the sequel we will provide a negative answer to this question. Intuitively, this is due to the fact that, in our problem, the valuation of a solution of agent i depends on the types of all agents. Instead, in the "dual" problem of scheduling with selfish machines (see [1]), the valuation of agent i depends only on her own type t_i and on some other public input (i.e., the size of the jobs). In the latter case, agent i can compute the valuation of a given solution, while in our problem this does not hold.

THEOREM 9. *No exact truthful mechanism exists for the problem of scheduling selfish jobs on related machines, even for the case of two machines with identical speeds.*

PROOF. Consider two machines of equal speed. By contradiction, let $M = (A, p_A)$ be a truthful exact mechanism for the problem. Consider a set of three jobs of size $t = (t_1, t_2, 3)$ and let $b' = (2, 1, 3)$ and $b'' = (4, 1, 3)$ be two vectors of values reported to the mechanism. Since M is truthful and $b'_{-1} = b''_{-1}$, we have that, if $t_1 = 2$, then $u^1(b'|2) \geq u^1(b''|2)$, while if $t_1 = 4$ then $u^1(b''|4) \geq u(b'|4)$. This is equivalent to

$$p^1(b') - p^1(b'') = w^1(b') - w^1(b''). \quad (4)$$

Consider now the allocations computed by the optimal algorithm A on input b' and b''. It is easy to see that both instances admit only one optimal allocation. The following table shows these allocations.

instance	machine 1	machine 2
b'	1, **2**	3
b''	**4**	1, 3

Thus, $w^1(b') = t_2$ and $w^1(b'') = 0$ and by Eq. 4 we have that $p^1(b') - p^1(b'') = t_2$. Since t_2 is not known to the mechanism, M cannot guarantee that truth-telling maximizes the utility of agent 1 on any instance, contradicting the hypothesis that is an exact truthful mechanism. □

Algorithm `rand-shift`(b, m)

1. solve the problem on m identical machines using some algorithm `alg`;
2. let W_j denote the set of jobs assigned to the j^{th} identical machine;
3. pick a number $q \in [0, m-1]$ at random with uniform distribution;
4. assign the jobs in W_j to the machine $(j+q) \mod m$, $j = 1, 2, \ldots, m$;

Figure 3: A randomized algorithm for small values of $r = s_{\max}/s_{\min}$.

4. RANDOMIZED MECHANISMS FOR "ALMOST IDENTICAL" MACHINES

In this section we provide a randomized mechanism for the case in which the ratio $r = s_{\max}/s_{\min}$ is small. The mechanism is truthful in expectation and is based on the following idea: we first solve the problem on m identical machines; then, we assign the work of the identical machines to the machines of our instance *uniformly at random*. Intuitively speaking, the random shift guarantees that, no matter what the reported value b_i is, the probability of being assigned to a certain machine is always $1/m$. So, from the agent point of view, in expectation, the speed of its machine is constant, and we can define payments similar to the case of identical speeds. Algorithm `rand-shift` is given in Fig. 3.

We can thus prove the following result:

THEOREM 10. *For any $\epsilon > 0$, there exists a polynomial-time randomized mechanism which is equilibria-truthful in expectation and whose approximation ratio is $r + \epsilon$.*

PROOF. Consider the *expected utility* of a mechanism $M = (A, p)$, when $A =$ `rand-shift`. Let $A^i(b, q)$ denote the machine where job i is allocated on input b and when the algorithm selects the random value $q \in [0, m-1]$ and let $s^i_A(b, q)$ be its speed. Observe that, the quantity $w^i(b)$, that is the sum of the real weights of the jobs assigned to the same machine as job i, does not depend on q. Let $p^i(b)$ be a payment function that does not depend on the random number q. For each fixed $q \in [0, m-1]$, the corresponding utility $u^i(b|t, q)$ of agent i satisfies

$$u^i(b|t, q) = p^i(b) - \frac{t_i + w^i(b|t_{-i})}{s^i(b, q)}.$$

Observe that, for every b, the probability that job i is assigned to the j^{th} machine is $1/m$. Thus, the expected utility $E(b_i, b_{-i})$ is equal to

$$E^i(b_i, b_{-i}|t) := \mathrm{E}[u^i(b_i, b_{-i}|t)] \quad (5)$$

$$= \sum_{j=1}^{m} \Pr[A^i(b, q) = j] \cdot \left(p^i(b) - \frac{t_i + w^i(b)}{s_j}\right) \quad (6)$$

$$= p^i(b) - \frac{t_i + w^i(b)}{m} \sum_{j=1}^{m} \frac{1}{s_j}. \quad (7)$$

The expected utility $E^i(\cdot)$ equals to the utility in the case of m machines of identical speed $s := m/(\sum_{j=1}^{m} 1/s_j)$. We can thus define the payments as in the proof of Theorem 7, that is, $p^i(b) := \frac{z^i(b_i)}{s}$, where $z^i(x) = \sum_{j:A^j(x,b_{-i})=A^i(x,b_{-i}), j\neq i} b_j$. When $b_{-i} = t_{-i}$ it holds that $z^i(b_i) = w^i(b_i)$, thus implying

$$E^i(b_i, t_{-i}) = \frac{w^i(b_i)}{s} - \frac{t_i + w^i(b_i)}{m} \sum_{j}^{m} \frac{1}{s_j} = -\frac{t_i}{s}.$$

So, the mechanism is equilibria-truthful in expectation. By using a polynomial-time $(1+\epsilon)$-approximate algorithm `alg` in the definition of `rand-shift`, the resulting mechanism runs in polynomial time, is equilibria-truthful and guarantees an approximation ratio equal to $r(1+\epsilon)$. □

5. AGENTS OWNING MORE THAN ONE JOB

In this section we investigate the case in which an agent may own more than one job, and machine speeds are identical.

We have m machines of speed s, l jobs of weight $(t_1, \cdots, t_l)$, and $n < l$ agents. We denote by X^i the set of the k_i jobs owned by agent i and by k the maximum of the k_i's. For any job j, we denote by own_j the set of jobs with the same owner as j. If A is a scheduling algorithm then, for $1 \leq j \leq l$, we denote by $A^j(b)$ the set of jobs that algorithm A, on input the reported weights b, assigns to the same machine as job j (including job j itself) and by $w^j_A(b|t)$ the sum of the real weights of the jobs of $A^j(b)$. Hence,

$$w^j_A(b|t) = \sum_{h \in A^j(b)} t_h.$$

Since all machines have speed s, the quantity $-w^j_A(b|t)/s$ is the finish time of job j according to solution $A(b)$. We define the valuation of agent i as minus the sum of the finish times of her jobs, that is,

$$v^i_A(b|t) := -\sum_{j \in X^i} \frac{w^j_A(b|t)}{s}.$$

Our valuation function models the case in which each customer pays the agent controlling her piece of traffic a fixed amount minus the experienced latency of her traffic. This is motivated by the scenario in Fig. 1, where each of the n providers (the selfish agents) wants to maximize the amount of money received from the customers (i.e., the jobs owned by that agent).

We rewrite the above valuation function as follows. For any algorithm A, we define $o^j_A(b|t)$ as the *sum* of the real weights of those jobs in $A^j(b)$ that do not belong to own_j; that is,

$$o^j_A(b|t) = \sum_{h \in A^j(b), h \notin own_j} t_h. \quad (8)$$

Moreover, we define $m^j_A(b)$ as the *number* of jobs in $A^j(b)$ that belong to own_j; that is,

$$m^j_A(b) = |A^j(b) \cap own_j|. \quad (9)$$

Using the above definitions we can write $v^i_A(b|t)$ as

$$v^i_A(b|t) = -\sum_{j \in X^i} \frac{m^j_A(b) \cdot t_j}{s} - \sum_{j \in X^i} \frac{o^j_A(b|t)}{s}. \quad (10)$$

When algorithm A is clear from the context or immaterial, we will drop the subscript "A" and simply write $m^j(b), w^j(b|t)$, $v^i(b|t)$, and $o^j(b|t)$.

5.1 Lower bounds

In this section we prove lower bounds on the approximation ratio obtained by truthful mechanisms in the case where agents can own more than one job. Our proofs adopt the following strategy: given two possible declarations b' and b'', any truthful mechanism must guarantee, for all agents i, that

$$p^i(b') + v^i(b'|b') \geq p^i(b'') + v^i(b''|b') \quad (11)$$

and

$$p^i(b'') + v^i(b''|b'') \geq p^i(b') + v^i(b'|b''). \quad (12)$$

From the above equations, we then derive necessary conditions on the payment function and the allocation algorithm of a truthful mechanism which in turn imply a lower bound on the approximation ratio.

THEOREM 11. *For any $m \geq 2$ and for any $c < 7/6$, no truthful mechanism can guarantee c-approximate solutions on m machines of equal speeds when each agent owns at most 2 jobs.*

PROOF. We prove the theorem for $m = 2$ machines of unitary speed. Let $M = (A, p)$ be a truthful mechanism for this problem. Let us consider an instance with 2 agents and 3 jobs such that $t = (t_1, t_2, 1)$ and $X^1 = \{1, 2\}$, $X^2 = \{3\}$.

Consider two vectors of declared weights $b' = (x', y', 1)$ and $b'' = (x'', y'', 1)$, with $x' \leq y' < 1$ and $x'' \leq 1 < y''$. Observe that if M is an exact mechanism then its allocation algorithm A is optimal. In particular, on input b' or b'', it must allocate the two smallest jobs on the same machine (say 1) and the largest one on the other machine (say 2). Thus, A should produce the following two allocations

instance	machine 1	machine 2
b'	x', y'	1
b''	$x'', 1$	y''

and $v^1(b'|b') = -2(x' + y')$, $v^1(b'|b'') = -2(x'' + y'')$, $v^1(b''|b') = -(x' + y' + 1)$ and $v^2(b''|b'') = -(x'' + y'' + 1)$. Then, by Eq. 11-12 we have that

$$p^1(x', y') - 2(x' + y') \geq p^1(x'', y'') - x' - y' - 1 \quad (13)$$

$$p^1(x'', y'') - x'' - y'' - 1 \geq p^1(x', y') - 2(x'' + y'') \quad (14)$$

It is easy to verify that these two inequalities both hold only when $x'' + y'' \geq x' + y'$. Thus, if M is truthful and $x'' + y'' < x' + y'$ then A cannot give an optimal allocation on both inputs b' and b''.

We now give a lower bound on the approximation ratio of the solution given by A. Consider vectors $b' = (3/4, 3/4, 1)$ and $b'' = (1/4 - \varepsilon, 5/4, 1)$, for some sufficiently small $\varepsilon > 0$. Then since $1/4 - \varepsilon + 5/4 < 3/4 + 3/4$ A cannot be optimal on both vectors. Observe that any sub-optimal allocation on b' must allocate the two jobs of agent 1 on different machines, while on b'' it must allocate them on the same machine. If A gives a sub-optimal allocation on input b', then the cost of $A(b')$ is at least $7/4$ and the approximation ratio is at least $\frac{7/4}{3/2} = 7/6$. If, instead, A gives a sub-optimal allocation on b'' then the cost of $A(b'')$ is at least $3/2 - \varepsilon$ and the approximation ratio is at least $\frac{3/2-\varepsilon}{5/4} \simeq 6/5$. Then, the theorem follows.

□

The above theorem also applies to the case in which agents may own more than two jobs. However, in this case, we can obtain a better lower bound:

THEOREM 12. *For any $m \geq 2$ and for any $c < 4/3$, no truthful mechanism can guarantee c-approximate solutions on m machines of equal speeds when each agent owns $k \geq m$ jobs.*

PROOF. We prove the theorem for $m = 2$ machines of unitary speed. Let $M = (A, p)$ be a truthful mechanism for this problem. Let us consider an instance with 1 agent and 4 jobs and consider two vectors of declared weights $b' = (1, 1, 1, 1)$ and $b'' = (x, x, x, 3x - 3)$ for some $x > 0$. Observe that if M is an exact mechanism then, for sufficiently large x, its allocation algorithm computes the following allocations

instance	machine 1	machine 2
b'	1, 1	1, 1
b''	x, x, x	$3x - 3$

and $v(b'|b') = -8$, $v(b''|b'') = -12x + 3$, $v(b''|b') = -10$ and $v(b'|b'') = -12x + 6$. Since M is truthful, by Eq. 11-12 we have that $p(b') - p(b'') \geq -2$ and $p(b') - p(b'') \leq -3$, proving that A cannot compute an optimal allocation on both b' and b''. Thus, if A computes a sub-optimal solution on b' (putting at least 3 jobs on the same machine) then its solution costs at least 3 while the optimum is 2. If, instead, A computes a sub-optimal solution on b'' (putting an even number of jobs on each machine) then its solution costs at least $4x - 3$ while the optimum is $3x$. Thus, for any $\epsilon > 0$, there exists a sufficiently large x such that the approximation ratio of A is at least $(4/3 - \epsilon)$. This completes the proof. □

5.2 Upper bounds

Observe that the valuation by agent i of an allocation depends on two terms: the allocation of i's jobs *i.e.*, the first summation of Eq. 10, and the overall load due to the other agents' jobs w.r.t. the machines used by agent i, i.e., the second summation of Eq. 10. In particular, if two jobs of agent i are assigned to the same machine we count their weights twice, while if we assign them to distinct machines we count their weights once. With this in mind, we consider algorithms that keep the first part constant and payment functions that counterbalance the influence of the jobs of other agents. In this way the utility of each agent is independent from the values she declares and thus there is no incentive to lie.

DEFINITION 13 (INDEPENDENT ALGORITHMS). *An algorithm A is* independent *if, for every b, b', it assigns jobs in such a way that, for any j, $m_A^j(b) = m_A^j(b')$. In this case, we will just write m_A^j instead of $m_A^j(b)$.*

THEOREM 14. *For any independent algorithm A, there exists a payment function p_A such that $M = (A, p_A)$ is equilibria-truthful.*

PROOF. For every agent i we define the payment function

$$p_A^i(b) := \sum_{j \in X^i} \frac{z_A^j(b)}{s}, \quad (15)$$

where $z_A^j(b) := \sum_{h \in A(b)_j, h \notin own_j} b_h$. Then

$$u^i(b|t) = \sum_{j \in X^i} \frac{z_A^j(b)}{s} - \sum_{j \in X^i} \frac{m_A^j \cdot t_j}{s} - \sum_{j \in X^i} \frac{o_A^j(b)}{s}.$$

When all agents other than i are truth-telling (that is, $b_h = t_h$ for all $h \notin X^i$) we have that $u^i(b|t) = -\sum_{j \in X^i} \left(m_A^j \cdot t_j\right)/s$. So the utility does not depend on the declarations of the agents, and (A, p_A) is an equilibrium-truthful mechanism. □

We start by considering the case in which, for each agent i, $k_i \leq m$ or k_i is a multiple of m and present algorithm `spread`, given in Fig. 4, that is independent and guarantees a 2-approximation.

Algorithm `spread`(b, m)

for $i = 1$ to n do
  set $c_i = \lceil k_i/m \rceil$;
  for $j \in X^i$ in nonincreasing order by declared weight do
    assign job j to the least loaded machine that has less than c_i jobs from agent i;

Figure 4: An independent algorithm.

LEMMA 15. *Algorithm* `spread` *is independent if, for any agent i, $k_i \leq m$ or k_i is a multiple of m.*

We now study the approximation guaranteed by algorithm `spread`. We start with the following technical lemma.

Algorithm $\texttt{split}(b, m)$

01. partition the machines in two sets S_1 and S_2 of cardinality $\lfloor m/2 \rfloor$ and $\lceil m/2 \rceil$;
02. for $i = 1$ to n
03. write k_i as $k_i = c_i \cdot \lceil m/2 \rceil + q_i$ for some $0 \leq q_i < \lceil m/2 \rceil$;
04. partition X^i in X_1^i containing the q_i heaviest jobs and X_2^i containing the $c_i \cdot \lceil m/2 \rceil$ remaining jobs;
05. let $b^1 = \cup_{i=1}^n X_1^i$ and $b^2 = \cup_{i=1}^n X_2^i$;
06. schedule jobs in b^1 on machines of S_1 using spread;
07. schedule jobs in b^2 on machines of S_2 using spread;

Figure 5: An algorithm for the case $k > m$.

LEMMA 16. *Fix an input vector b. Let L_h^i denote the load of machine h after algorithm* spread *has assigned all jobs of agent i and let $L_{\max}^i := \max_{1 \leq h \leq m} L_h^i$, $L_{\min}^i := \min_{1 \leq h \leq m} L_h^i$, and $b_{\max} := \max_{1 \leq j \leq l} b_j$. If $k \leq m$ then it holds that $L_{\max}^i - L_{\min}^i \leq b_{\max}$.*

Let $p_{\texttt{spread}}$ be defined as in Eq. 15. We now prove that (spread, $p_{\texttt{spread}}$) is an equilibria-truthful 2-approximate mechanism for the case when for each agent i, $k_i \leq m$ or k_i is a multiple of m.

THEOREM 17. *$M = (\texttt{spread}, p_{\texttt{spread}})$ is a polynomial-time equilibria-truthful 2-approximation mechanism for allocating selfish jobs to m identical machines if, for each agent i, $k_i \leq m$ or k_i is a multiple of m.*

PROOF. The truthfulness follows from Lemma 15 and Theorem 14.

We next show that spread is a 2-approximation algorithm and consider first the case in which $k_i \leq m$ for all agents i. Let $L_{\max}^i$ and $L_{\min}^i$ be defined as in Lemma 16. Observe that $\mathsf{opt}(b) \geq \sum_{i=1}^n b_i/m \geq L_{\min}^n$ and $\mathsf{opt}(b_1, \ldots, b_n) \geq b_{\max}$. Thus, we have that

$$\mathsf{cost}(A, b) = L_{\max}^n \leq b_{\max} + L_{\min}^n \leq 2\mathsf{opt}(b),$$

where the first inequality follows from Lemma 16. Thus spread is 2-approximated with respect to the declared weights. However, since the mechanism is equilibrium-truthful, the declared weights coincides with the real weights and thus the mechanism is 2-approximated.

Now let us consider the case in which some agent i has $c_i \cdot m$ jobs. Then it is easy to observe that if in place of agent i, we consider c_i agents each with m jobs then the allocation computed by spread does not change. □

We now show how algorithm spread can be used as a building block to design algorithm split (see Fig. 5) that can be used to construct an equilibria-truthful mechanism for the case $k > m$.

THEOREM 18. *$M = (\texttt{split}, p_{\texttt{split}})$ is an equilibria-truthful mechanism.*

PROOF. Let $p_{\texttt{split}}$ be defined as in Eq. 15. The utility function of agent i can be rewritten as

$$\begin{aligned} u^i(b_{X^i}, b_{-X^i}|t) &= -\sum_{j \in X^i} m^j_{\texttt{split}}(b_{X^i}, b_{-X^i}|t) \cdot \frac{t_j}{s} \\ &= -\sum_{h \in X_1^i} \frac{t_h}{s} - \sum_{h \in X_2^i} c_i \cdot \frac{t_h}{s}. \end{aligned}$$

By Eq. 10, if every agent other than i reports her true weights then $z_A^j(b) = o_A^j(b)$ and thus we have that, for any agent i and for any pair of vectors b_{X^i},

$$u^i(t_{X^i}, t_{-X^i}|t) \geq u^i(b_{X^i}, t_{-X^i}|t).$$

□

THEOREM 19. split *is a 4-approximate algorithm for m even.*

PROOF. Observe that the algorithm partitions machines in two sets S_1 and S_2, both of size $m/2$, and jobs in two sets b^1 and b^2 and then assigns jobs in b^1 to machines in S_1 and jobs in b^2 to machines in S_2. Denote by cost_i the cost of the allocation of b^i on machine S_i, for $i = 1, 2$. The cost of the allocation computed by algorithm split is obviously equal to $\max\{\mathsf{cost}_1, \mathsf{cost}_2\}$. We prove the theorem by showing that $\mathsf{cost}_i \leq 4\mathsf{opt}_m(b)$, for $i = 1, 2$, where $\mathsf{opt}_m(b)$ is the cost of an optimal allocation of jobs b on m identical machines. We have that

$$\mathsf{cost}_i \leq 2\mathsf{opt}_{\frac{m}{2}}(\beta_i) \leq 2\mathsf{opt}_{\frac{m}{2}}(b) \leq 4\mathsf{opt}_m(b), \qquad (16)$$

where the first inequality follows from the fact that $|X_2^i| = c_i \lceil m/2 \rceil$ and from Theorem 17. □

COROLLARY 20. *There exists a polynomial-time deterministic truthful $4(1 + 1/m)$-approximation mechanism for any odd m, when agents own up to $k > m$ jobs.*

PROOF. The case m even follows from Theorems 18-19. The case m odd easily follows by modifying split so to use $m - 1$ out of the m machines. □

The above results can be improved when considering small values of m. In particular we prove the following theorem.

THEOREM 21. *For every $\epsilon > 0$, there exists a polynomial-time deterministic truthful $(\frac{3}{2} + \epsilon)$-approximation mechanism for allocating jobs on two identical machines, when each agent owns either a single job or an even number of jobs.*

PROOF. Consider the following algorithm: first run PTAS for 2 machines to get a solution X; then transform X into a new solution X' as follows: For every i such that $k_i > 1$ and X allocates $c > k_i/2$ jobs on machine j (where j is either 1 or 2), move the $c - k_i/2$ lightest jobs of i from machine j to the other machine.

First observe that this algorithm is independent. Next we show that it is $(3/2 + \epsilon)$-approximate.

Since $c \leq k_i/2$, and since we move the lightest jobs, after performing this step for all agents, we have moved at most $1/2$ of the load of each machine to the other. It follows that, the maximum load of X' is at most $3/2$ times the maximum

load of X. Since solution X is $(1+\epsilon)$-approximated we have that the solution computed by the algorithm is $(3/2+\epsilon)$-approximate. Thus, by Theorem 14 there exists a polytime equilibria-truthful mechanism using this algorithm.

□

When we have more than 2 machines, we can use the same algorithm as in Theorem 21, considering only to the two fastest machines. Clearly we lose a factor of $1/m$ for each ignored machine. For the cases of $m = 3, 4$, the approximation ratio is still better than the approximation of `split`. Thus we get the following corollaries.

COROLLARY 22. *For every $\epsilon > 0$, there exists a polynomial-time deterministic truthful $(\frac{9}{4}+\epsilon)$-approximation mechanism for allocating jobs on three identical machines, when each agent owns either a single job or an even number of jobs.*

COROLLARY 23. *For every $\epsilon > 0$, there exists a polynomial-time deterministic truthful $(3+\epsilon)$-approximation mechanism for allocating jobs on four identical machines, when each agent owns either a single job or an even number of jobs.*

6. EXTENSIONS AND OPEN PROBLEMS

We first observe that our results on m identical machines and $k = 1$ can be combined with the results in [6] so to obtain Nash equilibria w.r.t. the following generalization of our (and also of the KP model) selfish routing game. Initially agents declare the weight of their jobs and a mechanism schedules them and provide some payment. Then, each agent can change machine if a better one exists. Equivalently, the scheduler (i.e., mechanism) now suggests a scheduling which the agents may not accept, unless it corresponds to a Nash equilibria in the KP model. Theorem 7 combined with the algorithm computing Nash equilibria of cost $(1+\epsilon)$ the optimal cost [6], yield a solution for our extended model with the same approximation.

One of the main problem left open is to generalize this result to the case of agents owning more than one job. This may be of particular interests since providers (i.e., agents) may not be forced to choose a certain link. We believe that the situation in this case is much more intricate since solution at Nash equilibrium for the KP model are not at Nash equilibria when considering an agent with more than one job: it may be the case that one of her jobs can improve its finish time, but this may also worsen other jobs from the same agent.

7. REFERENCES

[1] A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the IEEE Symposium on Foundations of Computer Science (**FOCS**)*, pages 482–491, 2001.

[2] E.H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, pages 17–33, 1971.

[3] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Pricing network edges for heterogeneous selfish users. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 521–530. ACM Press, 2003.

[4] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *Proc. of the 13th Annual ACM Symposium on Discrete Algorithms (**SODA**)*, pages 413–420, 2002.

[5] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibria. In *Proc. of the 30th International Colloquium on Automata, Languages, and Programming (**ICALP**)*, volume 2719 of *LNCS*, 2003.

[6] R. Feldmann, M. Gairing, T. Luecking, B. Monien, and M. Rode. Nashification and the coordination ratio for a selfish routing game. In *Proc. of the 30th International Colloquium on Automata, Languages, and Programming (**ICALP**)*, volume 2719 of *LNCS*, 2003.

[7] A. Ferrante and M. Parente. Existence of Nash Equilibria in Selfish Routing problems. Technical Report, Università di Salerno, 2002.

[8] D. Fotakis, S. Kontongiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *Proc. of the 29th International Colloquium on Automata, Languages, and Programming (**ICALP**)*, volume 2382 of *LNCS*, pages 123–134, 2002.

[9] T. Groves. Incentive in Teams. *Econometrica*, 41:617–631, 1973.

[10] D. S. Hochbaum and D. B. Shmoys. Using dual approximaiton algorithms for scheduling problems: theoretical and practical results. *Journal of the ACM*, 34:144–162, 1987.

[11] E. Koutzoupias and C.H. Papadimitriou. Worst-case equilibria. In *Proc. of Annual Symposium on Theoretical Aspects of Computer Science (**STACS**)*, volume 1563 of *LNCS*, pages 404–413, 1999.

[12] M. Mavronicolas and P.G. Spirakis. The price of selfish routing. In *Proc. of the Annual ACM Symposium on Theory of Computing (**STOC**)*, pages 510–519, 2001.

[13] N. Nisan and A. Ronen. Algorithmic Mechanism Design. In *Proc. of the 31st Annual ACM Symposium on Theory of Computing (**STOC**)*, pages 129–140, 1999.

[14] N. Nisan and A. Ronen. Computationally Feasible VCG Mechanisms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (**EC**)*, pages 242–252, 2000.

[15] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[16] C. H. Papadimitriou. Algorithms, Games, and the Internet. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing (**STOC**)*, pages 749–753, 2001.

[17] T. Roughgarden. Designing networks for selfish users is hard. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science (**FOCS**)*, pages 472–481, 2001.

[18] T. Roughgarden and E. Tardos. How bad is selfish routing? In *Proc. of the 41st IEEE Symposium on Foundations of Computer Science (**FOCS**)*, pages 93–102, 2000.

[19] W. Vickrey. Counterspeculation, Auctions and Competitive Sealed Tenders. *Journal of Finance*, pages 8–37, 1961.