Setting Lower Bounds on Truthfulness

Ahuva Mu'alem* Michael Schapira†

Abstract

We present and discuss general techniques for proving inapproximability results for truthful mechanisms. We make use of these techniques to prove lower bounds on the approximability of several non-utilitarian multi-parameter problems.

In particular, we demonstrate the strength of our techniques by exhibiting a lower bound of $2 - \frac{1}{m}$ for the scheduling problem with unrelated machines (formulated as a mechanism design problem in the seminal paper of Nisan and Ronen on Algorithmic Mechanism Design). Our lower bound applies to truthful randomized mechanisms (disregarding any computational assumptions on the running time of these mechanisms). Moreover, it holds even for the weaker notion of truthfulness for randomized mechanisms – i.e., truthfulness in expectation. This lower bound nearly matches the known $\frac{7}{4}$ (randomized) truthful upper bound for the case of two machines (a non-truthful FPTAS exists). No lower bound for truthful randomized mechanisms in multi-parameter settings was previously known.

We show an application of our techniques to the workload-minimization problem in networks. We prove our lower bounds for this problem in the inter-domain routing setting presented by Feigenbaum, Papadimitriou, Sami, and Shenker.

Finally, we discuss several notions of non-utilitarian "fairness" (Max-Min fairness, Min-Max fairness, and envy minimization). We show how our techniques can be used to prove lower bounds for these notions.

1 Introduction

1.1 Inapproximability Issues in Algorithmic Mechanism Design

The field of *Algorithmic Mechanism Design* [33] deals with designing protocols for achieving global goals that require interaction with selfish agents. Algorithmic Mechanism Design combines an economic perspective that takes into account the strategic behavior of the agents, with a theoretical computer-science perspective that focuses on computational aspects such as efficiency and approximability. In most of the works in this field the very robust notion of equilibrium in dominant strategies is used. It is well known ([31], see [33]) that, without loss of generality, we can limit ourselves to only considering "incentive compatible" mechanisms, also known as "truthful" mechanisms or "strategy-proof" mechanisms. In such mechanisms participants are always rationally motivated to correctly report their private information.

Let us now describe, more formally, the nature of the problems that Algorithmic Mechanism Design attempts to solve: There is a finite set of *alternatives* $A = \{a, b, c, ...\}$, and a set of *strategic agents* $N = \{1, ..., n\}$. Each agent i has a *valuation function* $v_i : A \rightarrow R$ that is his private information. The agents are self-interested and only wish to maximize their own gain. The global goal is expressed by a *social choice function* f that assigns every possible n-tuple of agents' valuations $(v_1, ..., v_n)$ an alternative $a \in A$. Mechanisms are said to *truthfully implement* a social choice function if their outcome for every n-tuple of agents' valuations matches that of the social choice function, and if they enforce payments of the different agents in a way that motivates truthful behavior.

*Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel. ahumu@yahoo.com.

†The School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel. mikesch@cs.huji.ac.il.
Supported by grants from the Israel Science Foundation and the USA-Israel Bi-national Science Foundation.

A very common social choice function is the *utilitarian* function; A utilitarian function aims to maximize the *social welfare*, i.e. to find the alternative a for which the expression $\Sigma_i v_i(a)$ is maximized. Another well known example of a social choice function is the *Max-Min* function (based on the philosophical work of John Rawls); For every n-tuple of v_i valuations the Max-Min function assigns the alternative a that maximizes the expression $min_i\, v_i(a)$. Intuitively, the Max-Min function chooses the alternative $a \in A$ in which the least satisfied agent has the highest value compared to the least satisfied agent in all other alternatives $b \in A$.

While in many computational and economic settings the social choice function we wish to implement in a truthful manner is utilitarian (e.g. combinatorial auctions), often this is not the case. Problems in which the social choice function is non-utilitarian include revenue maximization in auctions (e.g. [19]), minimizing the makespan in scheduling (e.g. [33, 3, 1]), fair allocation of resources (e.g. [9, 8, 30]), etc. A classic result of mechanism design – a subfield of economic theory and game-theory (see [31, 35]) – states that for every utilitarian problem there exists a mechanism that truthfully implements it – namely, a member of the celebrated family of *VCG mechanisms* [39, 11, 21]. No general technique is known for truthfully implementing non-utilitarian social-choice functions. In fact, some non-utilitarian social-choice functions cannot be truthfully implemented [9, 33]. Hence, from a computational point of view it is natural to ask how well these social choice functions can be *approximated* in a truthful manner.

1.2 Our Results

In this paper we present and discuss several general techniques for setting lower bounds on the approximability of truthful mechanisms. Our techniques are powerful in the following sense: Firstly, due to their generality and simplicity they can easily be applied to a variety of problems (as we shall demonstrate)[1]. Secondly, they apply to *multi-parameter settings* in which the agents have complex multi-dimensional preferences (and not only to single-parameter settings in which the private information of each agent consists of a single numerical parameter). Thirdly, they do not impose any computational assumptions on the mechanism (such as polynomial running-time). Finally, our techniques apply to both deterministic and randomized mechanisms. In particular, we show how to derive lower bounds for both notions of truthfulness for randomized mechanisms - *universal truthfulness* and *truthfulness in expectation*.

In Section 2 we present our techniques and demonstrate their use on a scheduling problem. The single-parameter version of the scheduling problem has received much attention in recent years [3, 1] (and references therein). We deal with the multi-parameter version of the problem presented by Lenstra, Shmoys, and Tardos [29]. This optimization problem was formulated as a mechanism design problem by Nisan and Ronen in their seminal paper on Algorithmic Mechanism Design [33]: There are n tasks $1, ..., n$ that are to be scheduled on m machines $1, ..., m$[2]. Every machine i is a strategic agent with a valuation function $v_i : 2^{[n]} \rightarrow R_{\geq 0}$ such that for every task $j \in [n]$, $v_i(\{j\})$ (we shall sometimes simply denote $v_i(j)$) specifies the *cost* of task j on machine i. One can think of the cost of task j on machine i as the time it takes i to complete j. For every $S \subseteq [n]$, $v_i(S) = \Sigma_{j \in S} v_i(j)$. That is, the total cost of a set of tasks on machine i is the additive sum of the costs of the individual tasks on that machine. The global goal is minimizing the makespan of the chosen schedule. I.e., assigning the jobs to the machines in a way that minimizes the finishing time of the schedule. Obviously, the makespan-minimization social choice function is non-utilitarian and hence cannot necessarily be truthfully implemented by any mechanism. Nisan and Ronen prove that not only is it impossible to minimize the makespan in a truthful manner, but that any approximation better than 2 cannot be achieved by a truthful *deterministic* mechanism.

Section 2 illustrates our techniques by proving several lower bounds for this problem. In particular, we prove that no randomized truthful mechanism can achieve an approximation ratio better than $2 - \frac{1}{m}$. This nearly matches the known truthful upper bound of $\frac{7}{4}$ for the case in which there are only 2 machines [33] (a non-truthful FPTAS exists for this case [23]). Hence, randomness cannot help in obtaining approximation ratios that are considerably better than the known lower bound for truthful deterministic mechanisms. Somewhat surprisingly, this lower bound applies even for the substantially weaker notion of truthfulness for

[1]In particular, our techniques do not involve payments (either made by the agents or distributed by the mechanism).

[2]We chose m to be the number of machines and n to be the number of tasks in order to be consistent with the formulation of Nisan and Ronen. Recall, that we used n previously to denote the number of agents, whereas here the agents are the machines.

randomized mechanisms - truthfulness in expectation. This is the first lower bound for truthful *randomized* mechanisms for this problem. In fact, to the best of our knowledge this is the first lower bound for truthful randomized mechanisms in multi-parameter settings in general.

In addition, we show how to prove lower bounds for two important classes of deterministic mechanisms: *strongly-monotone* mechanisms (a somewhat similar lower bound was proved by Lavi and Swamy [27]) and *affine maximizers* ("weighted-VCG"). This is another step towards proving the conjecture of Ronen and Nisan that no truthful deterministic mechanism can obtain an approximation ratio better than m.

In Section 3 we show an application of our techniques to another multi-parameter non-utilitarian problem – minimizing the workload in communication networks. This problem arises naturally in the design of routing mechanisms. We study the approximability of this problem in the inter-domain routing setting presented by Feigenbaum, Papadimitriou, Sami, and Shenker [16].

Finally, in Section 4 we discuss three notions of non-utilitarian fairness – Max-Min fairness, Min-Max fairness, and envy-minimization. We highlight the connections between these notions and the problems studied in this paper and prove several general results using our techniques.

1.3 Related Work

In a seminal paper Nisan and Ronen [33] introduced the field of Algorithmic Mechanism Design. The main problem presented in [33] to illustrate the novelty of this new area of research was *scheduling with unrelated machines*. Nisan and Ronen explored the approximability of this non-utilitarian multi-parameter problem and exhibited a lower bound of $2 - \epsilon$ for truthful deterministic mechanisms. For this NP-hard scheduling problem there exist an FPTAS (for any fixed numbers of machines) [23] and a polynomial-time 2-approximation algorithm [29], that are both non-truthful.

In recent years Algorithmic Mechanism Design has been the subject of extensive study. The vast majority of this research has focused on single parameter settings (see e.g. [28, 4, 32, 2, 19, 24]). Truthful mechanisms with constant approximation ratios were designed for the single-parameter problem of *minimum makespan for scheduling jobs on related machines* [3, 5, 1]. The exploration of truthful mechanisms for multi-parameter settings has mainly revolved around the problem of welfare maximization in combinatorial auctions [36], that has gained the status of the paradigmatic problem of this field. As this is a utilitarian problem, it can be optimally and truthfully implemented by a VCG mechanism. However, it has been shown that the social welfare in combinatorial auctions cannot be maximized (or even closely approximated) in polynomial time [28, 34]. As algorithmic mechanism design seeks time-efficient implementations, the main challenge faced by researchers was devising truthful polynomial-time mechanisms that approximately maximize the social welfare in combinatorial auctions ([26, 13, 14, 7, 22]).

There are but a few inapproximability results for truthful mechanisms. This is particularly true in multi-parameter settings. Other than Nisan and Ronen's $2 - \epsilon$ lower bound discussed previously, the following inapproximabily results are known: Lavi, Mu'alem and Nisan [25] proved several lower bounds for *polynomial-time* truthful *deterministic* mechanisms. Their work necessitated making several rather restrictive assumptions on the mechanisms beside assuming that they are truthful. Recently, Dobzinski and Nisan [12] proved inapproximability results for polynomial-time VCG mechanisms. We contribute to this ongoing research by presenting methods for deriving the first lower bounds for multi-parameter settings that apply to *general* truthful *randomized* mechanisms. Our lower bounds do not require any assumptions on the running-time of the mechanisms.

Our techniques greatly rely on the work of Bikhchandani et al. [9]. They characterize truthfulness in multi-parameter settings by showing that any truthful deterministic mechanism must maintain a certain *weak monotonicity* property. Using this characterization [9] manages to show that while welfare maximization can be truthfully implemented in combinatorial auctions, one cannot truthfully implement the Max-Min social choice function, even in a very restricted type of combinatorial auctions. Saks and Yu [37] proved that not only is weak monotonicity necessary for truthfulness, but in convex domains it is also sufficient. The weak monotonicity property (and several of its extensions) will play a major role in our inapproximability proofs.

Independently of our work, Christodoulou, Koutsoupias, and Vidali [10], proved a lower bound of $1 + \sqrt{2}$ that applies to *deterministic* mechanisms for the multi-parameter version of the scheduling problem. This improves over the previously known lower bound of 2 proven by Nisan and Ronen.

1.4 Open Questions

- We prove lower bounds for the scheduling problem with unrelated machines (see Section 2) and for the workload-minimization problem in inter-domain routing (see Section 3). In both problems, there are very large gaps between the known upper and lower bounds for truthful mechanisms (deterministic and randomized). Narrowing these gaps is an interesting open question.
- This paper did not make any computational assumptions on mechanisms. Proving (possibly stronger) lower bounds for *polynomial time* truthful mechanisms is a big open question.

1.5 The Organization of the Paper

In Section 2 we present our techniques for setting lower bounds on truthfulness and demonstrate their application to the *scheduling problem with unrelated machines*. In Section 3 we present an applications of our techniques to the problem of *workload-minimization in networks*. In Section 4 we discuss several notions of non-utilitarian fairness.

2 A Presentation of Our Techniques Via the Scheduling Problem

In this section we present our techniques. To illustrate the use of these techniques we show how they can be used to derive lower bounds for the scheduling problem with unrelated machines. Nisan and Ronen [33] exhibited a truthful m-approximation deterministic mechanism for this problem. This mechanism is basically a VCG mechanism, and can easily be shown to be strongly-monotone (see Subsection 2.1 for a formal definition of strong monotonicity). They also proved a lower bound of $2 - \epsilon$ for truthful deterministic mechanisms that applies even when there are only two machines and is tight for this case. However, [33] conjectures that their lower bound is not tight in general, and that any truthful deterministic mechanism cannot obtain an approximation factor better than m. [33] shows that this is indeed the case for two very restricted types of truthful deterministic mechanisms.

For the case of two machines Nisan and Ronen show that randomness helps get an approximation ratio better than 2; They present a truthful randomized mechanism that has an approximation ratio of $\frac{7}{4}$. We generalize their result to m machines by designing a truthful randomized mechanism that obtains an approximation ratio of $\frac{7m}{8}$. Thus, we prove that randomness achieves better performances than the known truthful deterministic m upper bound for any number of machines (see Appendix A.1).

In Subsection 2.1 we show ways of proving lower bounds for truthful deterministic mechanisms. Using these methods we provide a simple and shorter proof for Nisan and Ronen's $2 - \epsilon$ lower bound. Our proof (unlike the original) relies on exploiting the *weak monotonicity* property defined in [9]. The techniques of Subsection 2.1 also aid us in deriving stronger lower bounds for two important classes of deterministic mechanisms – *strongly-monotone* mechanisms (a somewhat similar result was independently proved by Lavi and Swamy [27]), and affine maximizers ("weighted VCG"). We note, that the mechanism in [33], which is the best currently known deterministic mechanism for the scheduling problem, is contained in both classes. We prove that no approximation ratio better than m is possible for these classes of mechanisms (thus making another step towards proving the conjecture of [33]).

After discussing lower bounds for truthful deterministic mechanisms we turn our attention to truthful randomized mechanisms. There are two possible definitions for the truthfulness of a randomized mechanism [3, 2, 14, 33]. The first and stronger one is that of *universal truthfulness* that defines a truthful randomized mechanism as a probability distribution over truthful deterministic mechanisms. Thus, this definition requires that for *any* fixed outcome of the random choices made by the mechanism, the agents still maximize their utility by reporting their true valuations. A considerably weaker definition of truthfulness is that of *truthfulness in expectation*. This definition only requires that players maximize their *expected* utility, where the expectation is over the random choices of the mechanism (but still for every behavior of the other players). Unlike universally truthful mechanisms, randomized mechanisms that are truthful in expectation only motivate *risk-neutral* bidders to act truthfully. Risk-averse bidders may benefit from strategic behavior. In addition, truthful in expectation randomized mechanisms induce truthful behavior only as long as players have no information about the outcomes of the random coin flips before they need to act.

In Subsection 2.2 we prove the first lower bound on the approximability of truthful randomized mechanisms in multi-parameter settings. Namely, we show that any universally truthful mechanism for the scheduling problem cannot achieve an approximation ratio better than $2 - \frac{1}{m}$. This lower bound nearly matches the universally truthful randomized $\frac{7}{4}$ upper bound for the case of two machines. To prove this lower bound, we make use of a general technique that is based on Yao's powerful principle [40]. Our proof for the $2 - \epsilon$ lower bound for deterministic mechanisms (in Subsection 2.1) functions as a "building block" in the proof of this lower bound.

In Subsection 2.3 we strengthen this result by proving that the same lower bound holds even when one is willing to settle for truthfulness in expectation. Our proof relies on some of the ideas that appear in the proof of the previous lower bound but takes a different approach. In particular, we generalize the weak monotonicity requirement to fit the class of truthful randomized mechanisms, and explore the implications of this extended monotonicity on the probability distributions over allocations generated by such mechanisms.

2.1 Lower Bounds for Truthful Deterministic Mechanisms

Bikhchandani et al. [9] formally define the weak monotonicity property for deterministic mechanisms: Consider an Algorithmic Mechanism Design setting with n strategic agents that wish to maximize their personal gain. Before we present the formal definition for monotonicity we will require the following notation: For every n-tuple of agents' valuations $v = (v_1, ..., v_n)$ we shall denote by v_{-i} the $(n-1)$-tuple of agents' valuations $(v_1, ..., v_{i-1}, v_{i+1}, ..., v_n)$. Let v'_i be a valuation function. We denote by (v'_i, v_{-i}) the n-tuple of valuations $(v_1, ..., v_{i-1}, v'_i, v_{i+1}, ..., v_n)$. I.e., (v'_i, v_{-i}) is the n-tuple of valuations we get by altering the i'th coordinate in v from v_i to v'_i.

Definition 1 *Let M be a truthful deterministic mechanism. Let $i \in [n]$ and let $v = (v_1, ..., v_n)$ be an n-tuple of agents' valuations. Let v'_i be a valuation function. Denote by a the alternative M outputs for v and by b the alternative that M outputs for (v'_i, v_{-i}). M is said to be weakly monotone if for all such i, v, and v'_i it holds that: $v_i(a) + v'_i(b) \geq v'_i(a) + v_i(b)$.*

Remark 2.1 *This definition of weak monotonicity is for cases in which each agent wishes to maximize his value. In problems in which agents wish to minimize costs (such as the scheduling and workload minimization problems considered in this paper) the inequality is in the other direction.*

[9] proves that any truthful deterministic mechanism must be weakly monotone. For completeness, we present this simple proof.

Lemma 1 *Any truthful deterministic mechanism must be weakly monotone.*

Proof: Let M be a truthful deterministic mechanism. Let $i \in [n]$ and let $v = (v_1, ..., v_n)$ be an n-tuple of agents' valuations. Let v'_i be a valuation function. Denote by a the alternative M outputs for v and by b the alternative that M outputs for (v'_i, v_{-i}). Consider agent i. It is well known that the price an agent is charged by the mechanism to ensure his truthfulness cannot depend on the agent himself. Hence, the payment of agent i in a and b is a function of v_{-i} and of a and b respectively. We denote by $p_i(v_{-i}, a)$ and by $p_i(v_{-i}, b)$ i's payment in a and b respectively. It must hold that $v_i(a) - p_i(v_{-i}, a) \geq v_i(b) - p_i(v_{-i}, b)$ (for otherwise, if i's valuation function is v_i, he would have an incentive to declare his valuation to be v'_i). Similarly, $v'_i(b) - p_i(v_{-i}, b) \geq v'_i(a) - p_i(v_{-i}, a)$. By adding these two inequalities we reach the weak monotonicity requirement. □

Relying on the weak monotonicity property we provide an alternative proof for the $2-\epsilon$ lower bound of [33] for the scheduling problem with unrelated machines. Our proof shows that any deterministic mechanism that achieves an approximation ratio better than 2 violates the weak monotonicity property.

Theorem 2.2 *Any weakly-monotone mechanism cannot achieve an approximation ratio better than* 2.

Proof: Let $\epsilon > 0$. Consider the scheduling problem with two machines and three tasks. For every machine $i = 1, 2$ we define two possible valuation functions v_i and v'_i:

$$v_i(t) = \begin{cases} 1 & t = i \text{ or } t = 3 \\ 100 & \text{otherwise} \end{cases}$$

$$v'_i(t) = \begin{cases} 0 & t = i \\ 1+\epsilon & t = 3 \\ 100 & \text{otherwise} \end{cases}$$

Let M be a deterministic, weakly-monotone, mechanism that achieves an approximation factor better than 2. Then, when agents 1 and 2 have the valuations v_1 and v_2 respectively, M must assign task 1 to agent 1, task 2 to agent 2, and can choose to which agent to assign task 3 (because the optimal makespan is 2 and any other assignment results in a makespan of at least 100). W.l.o.g. assume that M assigns task 3 to agent 2. Now, consider the instance with agents' valuations (v'_1, v_2). Notice that the only task-allocation that guarantees an approximation ratio better than 2 is assigning tasks 1 and 2 to agents 1 and 2 respectively, and task 3 to agent 1. However, this turns out to be a violation of the weak monotonicity requirement. Weak monotonicity, in this case, dictates that for every agent $i = 1, 2$ it must hold $v_i(a) + v'_i(b) \leq v'_i(a) + v_i(b)$. However, if we look at agent 1 we find that $1 + (1+\epsilon) = v_1(1) + v'_1(\{1,3\}) > v'_1(1) + v_1(\{1,3\}) = 2$. A contradiction. □

[25] presents another property – strong monotonicity.

Definition 2 *Let M be a deterministic mechanism. Let $i \in [n]$ and let $v = (v_1, ..., v_n)$ be an n-tuple of agents' valuations. Let v'_i be a valuation function. Denote by a the alternative M outputs for v and by b the alternative that M outputs for (v'_i, v_{-i}). M is said to be strongly-monotone if for all such i, v, and v'_i it holds that: If $a \neq b$, then $v_i(a) + v'_i(b) > v'_i(a) + v_i(b)$.*

Remark 2.3 *As in the case of weak monotonicity, this definition of strong monotonicity is for cases in which each agent wishes to maximize his value. In problems in which agents wish to minimize costs (such as the scheduling and workload minimization problems considered in this paper) the inequality is reversed.*

We prove that no member of the class of strongly-monotone mechanisms can obtain an approximation better than m for the scheduling problem (even for the case of zero/one valuations). A somewhat similar lower bound was independently proved by Lavi and Swamy [27]. The idea at the heart of our proof of Theorem 2.4 is an iterative use of the strong monotonicity property to construct an instance of the problem for which the allocation generated by the mechanisms is "very far" from optimal.

Theorem 2.4 *Any strongly-monotone mechanism cannot obtain an approximation ratio better than m.*

Proof: Consider an instance of the scheduling problem with m machines and $n = m^2$ tasks. Let M be a deterministic mechanism for which the strong monotonicity property holds. Let I be the instance of the scheduling problem in which every machine i has a valuation function v_i such that $v_i(j) = 1$ for all $j \in [n]$. Denote by $S = (S_1, ..., S_m)$ the allocation of tasks produced by M for instance I. It must be that there is some machine r such that $|S_r| \geq m$. Without loss of generality let $r = m$.

We will now create a new instance I' by altering the valuation function of machine 1 to v'_1 while leaving all the other valuation functions unchanged (in case $S_1 = \emptyset$ we skip this part). That is, machine 1 will have the valuation function $u_1 = v'_1$:

$$v'_1(t) = \begin{cases} 0 & t \in S_1 \\ 1 & t \notin S_1 \end{cases}$$

and every other machine $i \neq 1$ will have a valuation function $u_i = v_i$. Denote by $T = (T_1, ..., T_m)$ the allocation M generates for I'. The first step of the proof is showing that $S_1 = T_1$. This is guaranteed by the strong monotonicity of M. Assume, by contradiction that $S_1 \neq T_1$. The strong monotonicity property ensures that $v_1(S_1) + v'_1(T_1) < v_1(T_1) + v'_1(S_1)$. By assigning values we get:

$$|S_1| + |T_1 \setminus S_1| < |T_1| + 0$$

Observe, that $|S_1| + |T_1 \setminus S_1| - |T_1| = |S_1 \setminus T_1|$, therefore:

$$|S_1 \setminus T_1| < 0$$

A contradiction.

We shall now prove that not only does S_1 equal T_1, but in fact $S_i = T_i$ for every $i \in [m]$; Since $S_1 = T_1$ it must be that $v_1(S_1) + v'_1(T_1) = v'_1(S_1) + v_1(T_1)$. However, the strong monotonicity property dictates that if this is true then $S = T$.

In an analogous manner we shall now turn the valuation function of machine 2 into v'_2 while keeping all the other valuation functions in I' unchanged (in case $S_2 = \emptyset$ we skip this part). That is, the valuation function of machine 2 is changed into:

$$v'_2(t) = \begin{cases} 0 & t \in S_2 \\ 1 & t \notin S_2 \end{cases}$$

Similar arguments show that the allocation produced by the M for this new instance will remain S. We can now iteratively continue to change the valuation functions of machines $3, ..., m-1$ into $v'_3, ..., v'_{m-1}$ respectively, without changing the allocation the mechanism generates for these new instances. After performing this, we are left with an instance in which every machine $i \in [m-1]$ has the valuation function v'_i, and machine m has the valuation function v_m. We have shown that the allocation generated by M for this instance is S. Remember that $|S_m| \geq m$. Let $R \subseteq S_m$ such that $|R| = m$. We will now create a new instance INS from the previous one by only altering the valuation function v_m into the following valuation function v'_m:

$$v'_m(t) = \begin{cases} 0 & t \in S_m \setminus R \\ 1 & \text{otherwise} \end{cases}$$

By applying similar arguments to the ones used before, one can show that the allocation generated by M when given instance INS remains S. Observe that the finishing time of S is m because all the tasks in R are assigned to machine m. Also notice that the finishing time of the optimal allocation of tasks for INS is precisely 1. The theorem follows. $\square$

A lower bound of m can also be proven for any affine maximizer (see [25] for a formal definition). Affine maximizers, also sometimes called "weighted VCG", contain the celebrated class of VCG mechanisms.

Proposition 2.5 *Any affine maximizer cannot obtain an approximation ratio better than m.*

Proof: We define an instance in which machine 1 has a cost of 1 for every task, and machines $2, ..., m$ have a cost of $1 + \epsilon$. Observe, that while the value of the optimal makespan is $1 + \epsilon$, any VCG mechanism (which minimizes the cost), will allocate all the tasks to machine 1 and hence reach a makespan value of m. Since this is true for any ϵ, no VCG mechanism can obtain an approximation ratio better than m. This can easily be generalized to any affine maximizer. $\square$

2.2 Lower Bounds for Universally Truthful Mechanisms

We now present a technique for deriving lower bounds for universally truthful mechanisms, based on Yao's powerful principle [40]. Consider a zero-sum game with two players. Let the "row player"'s strategies be the various different instances of a specific problem, and let the "column player"'s strategies be all the deterministic truthful mechanisms for solving that problem. Let entry g_{ij} in the matrix G depicting the game be the approximation ratio obtained by the algorithm of column j when given the instance of row i.

Recall that every randomized mechanism that is truthful in the universal sense is a probability distribution over deterministic truthful mechanisms. The "natural" approach for proving a lower bound for such randomized mechanisms is to find an instance of the problem on which every such randomized mechanism cannot achieve (in expectation) a certain approximation factor. By applying the well known Minimax Theorem to the game described above we see that an alternate and just as powerful way for setting lower bounds is to show that there is a probability distribution over instances on which any deterministic mechanism cannot obtain (in expectation) a certain approximation ratio.

We demonstrate this technique by proving a $2-\frac{1}{m}$ lower bound for universally truthful mechanisms for the scheduling problem. Our proof is based on finding a probability distribution over instances of the scheduling problem for which no deterministic truthful mechanism can provide an approximation ratio better than $2-\frac{1}{m}$. To show this, we shall exploit the weak monotonicity property of truthful deterministic mechanisms (as discussed in Subsection 2.1).

Theorem 2.6 *Any randomized mechanism that is truthful in the universal sense cannot achieve an approximation ratio better than* $2-\frac{1}{m}$.

Proof: Let $\epsilon > 0$ be a real number. Consider the scheduling problem with m machines and $n = m+1$ tasks. For every machine $i \in [m]$ we define two possible valuation functions:

$$v_i(t) = \begin{cases} 1 & t = i \text{ or } t = m+1 \\ \frac{4}{\epsilon} & \text{otherwise} \end{cases}$$

and

$$v'_i(t) = \begin{cases} 0 & t = i \\ 1+\epsilon & t = m+1 \\ \frac{4}{\epsilon} & \text{otherwise} \end{cases}$$

Let I be the instance in which the valuation function of every machine i is v_i. For every $j \in [m]$, let I^j be the instance in which every machine $i \neq j$ has the valuation function v_i, and machine j has the valuation function v'_j. We are now ready to define the probability distribution P over instances: instance I is assigned the probability ϵ, and for every $j \in [m]$ instance I^j is picked with probability $\frac{1-\epsilon}{m}$.

We now need to show that any deterministic truthful mechanism M cannot achieve an approximation ratio better than $2-\frac{1}{m}$ on P. Let T^j be the allocation of the $m+1$ tasks to the m machines in which every machine i gets task i, and machine j is also assigned task $m+1$. Observe, that T^j is the optimal allocation of tasks for instance I^j. Also observe, that while the finishing time of the allocation T^j for instance I^j is $1+\epsilon$, the finishing time of any other allocation of tasks is at least 2. We shall denote the allocation M outputs for instance I by $M(I)$. Similarly, we shall denote the allocation M outputs for instance I^j by $M(I^j)$ (for every $j \in [m]$). We will now examine two distinct cases: The case in which $M(I) \neq T^r$ for any $r \in [m]$, and the case that $M(I) = T^r$ for some $r \in [m]$.

Observe, that in the first case the finishing time is at least $\frac{4}{\epsilon}$ while the optimal finishing time is 2. Thus, M obtains a $\frac{2}{\epsilon}$-approximation. Since instance I appears in P with probability ϵ we have that A's expected approximation ratio is at least $\frac{2}{\epsilon} \times \epsilon = 2$.

We are left with the case in which $M(I) = T^r$ for some $r \in [m]$. Consider an instance I^j such that $j \neq r$. The following lemma states that M will not output the optimal allocation for I^j (that is, T^j).

Lemma 2 *If* $M(I) = T^r$ *for some* $r \in [m]$*, then for every* $j \neq r$ $M(I^j) \neq T^j$.

Proof: Let $j \neq r$. Let us assume by contradiction that $M(I^j) = T^j$. The weak monotonicity property dictates that $v_j(j)+v'_j(\{j,m+1\}) \leq v'_j(j)+v_j(\{j,m+1\})$. By assigning values we get that $1+(1+\epsilon) \leq (1+1)$, and reach a contradiction. $\square$

From lemma 2 we learn that if $M(I) = T^r$ for some $r \in [m]$ we have that for every $j \neq r$ $M(I^j)$ is an allocation that is not the optimal one (i.e. not T^j). In fact (as mentioned before), any allocation M outputs given I^j will have a finishing time of at least 2, while the optimal allocation (T^j) has a finishing time of $1+\epsilon$. Thus, for every $j \neq r$ the approximation ratio of M for instance I^j is at least $\frac{2}{1+\epsilon}$. The expected approximation ratio of M for P is therefore at least $\frac{(m-1)(1-\epsilon)}{m} \times \frac{2}{1+\epsilon} + \frac{1-\epsilon}{m} \times 1$. Since this is true for any value of ϵ, the approximation ratio cannot be better than $2 - \frac{1}{m}$. □

2.3 Lower Bounds for Mechanisms that are Truthful in Expectation

After handling the case of universally truthful mechanisms we now turn to the weaker notion of truthfulness in expectation. We start by generalizing the weak monotonicity definition to the case of randomized mechanisms. Any randomized mechanism can be regarded as a mechanism that for every instance of a problem produces a probability distribution over possible alternatives.

Definition 3 *A randomized mechanism is a function from n-tuples of agents' valuations to probability distributions over the set of alternatives A.*

The valuation function of each of the agents in such mechanisms can therefore be viewed as assigning values to probability distributions over possible alternatives rather than only to the alternatives themselves.

Definition 4 *Let v be a valuation function. We define the* extended valuation function *V_v as follows. For every probability distribution P over the set of alternatives A, $V_v(P) = \Sigma_{a \in A}\ Pr_P[a] \times v(a)$.*

Arguments similar to those of lemma 1 show that randomized mechanisms that are truthful in expectations must be weakly monotone (given the new definition of the valuation functions). This *extended weak monotonicity* is equivalent to the *monotonicity in expectation* property defined by Lavi and Swamy [26].

Definition 5 *Let M be a randomized mechanism. Let $i \in [n]$ and let $v = (v_1, ..., v_n)$ be an n-tuple of agents' valuations. Let v_i' be a valuation function. Denote by P the distribution over alternatives M outputs for v and by Q the distribution over alternatives M outputs for (v_i', v_{-i}). M is said to be weakly monotone in the extended sense if for all such i, v, and v_i' it holds that: $V_{v_i}(P) + V_{v_i'}(Q) \geq V_{v_i'}(P) + V_{v_i}(Q)$.*

Remark 2.7 *As before, if the agents wish to minimize costs rather than maximize values, the inequality is reversed.*

Lemma 3 *Any truthful randomized mechanism must be weakly monotone in the extended sense.*

We can exploit this extended definition of weak monotonicity to prove inapproximability results. We show how this is done by strengthening our $2 - \frac{1}{m}$ lower bound for universally truthful mechanisms by showing that it applies even for the case of truthfulness in expectation. To do this, we show that the extended weak monotonicity of truthful randomized mechanisms implies non-trivial connections between the probability distributions over allocations they produce for different instances of the scheduling problem.

A key element in the proof of Theorem 2.8 is the observation that instead of regarding a randomized mechanism for the scheduling problem as generating probability distributions over allocations of tasks, it can be regarded as generating, for each task, a probability distribution over the machines it is assigned to by the mechanism. This is true due to the linearity (additivity) of the valuation functions. This different view of a randomized mechanism for this specific problem, enables us to analyze the contribution of each task to the expected makespan.

The main lemma in the proof of Theorem 2.8, namely Lemma 4, makes use of this fact together with the extended weak monotonicity of truthful randomized mechanisms. Lemma 4 essentially proves that for two carefully chosen instances of the problem, the probability that a specific task is assigned to a specific machine by M in one of the instances, cannot be considerably higher than the probability it is assigned to the same machine in the other. Thus, we show that even though allocating this task to that machine in one of the instances leads to a good approximation, any truthful randomized mechanism will fail to do so.

Theorem 2.8 *Any randomized mechanism that is weakly monotone in the extended sense cannot achieve an approximation ratio better than* $2 - \frac{1}{m}$.

Proof: Let $\epsilon > 0$ be a real number. Consider the scheduling problem with m machines and $n = m + 1$ tasks. For every machine $i \in [m]$ we define two possible valuation functions:

$$v_i(t) = \begin{cases} 1 & t = i \text{ } or \text{ } t = m+1 \\ \frac{4}{\epsilon^2} & \text{otherwise} \end{cases}$$

and

$$v'_i(t) = \begin{cases} 0 & t = i \\ 1+\epsilon & t = m+1 \\ \frac{4}{\epsilon^2} & \text{otherwise} \end{cases}$$

Let I be the instance in which the valuation function of every machine i is v_i. For every $j \in [m]$ let I^j be the instance in which every machine $i \neq j$ has the valuation function v_i, and machine j has the valuation function v'_j. Let T^j be the allocation of the $m+1$ tasks to the m machines in which every machine i gets task i, and machine j is also assigned task $m+1$.

Let M be a randomized mechanisms that is weakly monotone in the extended sense. We shall denote by P the distribution over all possible allocations produced by M when given instance I, and by P^j the distribution over all possible allocations M produces when given instance I^j. Let R be some distribution over the possible allocations. Fix a machine i and a task j, we define $p_{i,j}(R)$ to be the probability that machine i gets item j given R. Formally, $p_{i,t}(R) = \Sigma_{a|t\in a_i} Pr_R[a]$. Observe that $V_{v_i}(R) = \Sigma_{t\in[n]} p_{i,t}(R) v_i(t)$ and $V_{v'_i}(R) = \Sigma_{t\in[n]} p_{i,t}(R) v'_i(t)$.

We are now ready to prove the theorem. In order to do so, we prove that for every mechanism M, as defined above, one can find an instance of the scheduling problem for which M fails to give an approximation ratio better than $2 - \frac{1}{m}$. Consider instance I.

If for some $i \in [m]$ $p_{i,i}(P) < 1 - \epsilon^2$ then machine i does not get task i with probability of at least ϵ^2. However, when machine i does not get task i, the finishing time of a schedule for I cannot be less than $\frac{4}{\epsilon^2}$, while the optimal finish time is 2. Therefore, with probability of at least ϵ^2 the approximation ratio obtained by the algorithm is at least $\frac{2}{\epsilon^2}$. If this is the case then, in expectation, the approximation ratio is at least 2 (and the theorem follows). Hence, from now on we will only deal with the case in which for every $i \in [m]$, $p_{i,i}(P) \geq 1 - \epsilon^2$.

Let r be some machine such that $p_{r,m+1}(P) \leq \frac{1}{m}$. Intuitively, r is a machine that is hardly assigned task $m+1$ in P. We will show that in this case we can choose the instance I^r to prove our lower bound. The main idea of the proof is showing that machine r will not be assigned task $m+1$ in P^r with probability that is significantly higher than the probability it was assigned the task in P. Thus, even though assigning task $m+1$ to machine r is a smart step approximation-wise, the extended weak monotonicity of the mechanism will prevent it from doing so.

Lemma 4 *Let* r *be some machine such that* $p_{r,m+1}(P) \leq \frac{1}{m}$. *It holds that* $p_{r,m+1}(P^r) \leq \frac{1}{m} + \epsilon$.

Proof: As M is weakly monotone in the extended sense we have that $V_{v_r}(P) + V_{v'_r}(P^r) \leq V_{v'_r}(P) + V_v v_r(P^r)$. That is:

$$\Sigma_{t\in[n]} p_{r,t}(P) v_r(t) + \Sigma_{t\in[n]} p_{r,t}(P^r) v'_r(t) \leq \Sigma_{t\in[n]} p_{r,j}(P) v'_r(t) + \Sigma_{t\in[n]} p_{r,t}(P^r) v_r(t)$$

After subtracting identical summands from both sides of the equation we get:

$$p_{r,r}(P) v_r(r) + p_{r,m+1}(P) v_r(m+1) + p_{r,r}(P^r) v'_r(r) + p_{r,m+1}(P^r) v'_r(m+1) \leq$$

$$p_{r,r}(P) v'_r(r) + p_{r,m+1}(P) v'_r(m+1) + p_{r,r}(P^r) v_r(r) + p_{r,m+1}(P^r) v_r(m+1)$$

By assigning values we reach the following inequality:

$$p_{r,r}(P) + p_{r,m+1}(P) + p_{r,m+1}(P^r) \times (1+\epsilon) \leq p_{r,m+1}(P) \times (1+\epsilon) + p_{r,r}(P^r) + p_{r,m+1}(P^r)$$

Therefore:

$$p_{r,r}(P) + p_{r,m+1}(P^r) \times \epsilon \leq p_{r,m+1}(P) \times \epsilon + p_{r,r}(P^r)$$

Because $p_{r,r}(P) \geq 1 - \epsilon^2$ and $p_{r,r}(P^r) \leq 1$ we get:

$$(1-\epsilon^2) + p_{r,m+1}(P^r) \times \epsilon \leq p_{r,m+1}(P) \times \epsilon + 1$$

$$p_{r,m+1}(P^r) \times \epsilon - \epsilon^2 < p_{r,m+1}(P) \times \epsilon$$

$$p_{r,m+1}(P^r) \leq p_{r,m+1}(P) + \epsilon$$

Since $p_{r,m+1}(P) \leq \frac{1}{m}$ we have that:

$$p_{r,m+1}(P^r) \leq p_{r,m+1}(P) + \epsilon \leq \frac{1}{m} + \epsilon$$

This concludes the proof of the lemma. □

From Lemma 4 we learn that if r is a machine such that $p_{r,m+1}(P) \leq \frac{1}{m}$ then $p_{r,m+1}(P^r) \leq \frac{1}{m} + \epsilon$. Relying on this fact, we can choose I^r as our instance and show that M fails to provide an approximation ratio better than $2 - \frac{1}{m}$ for I^r. The optimal allocation for I^r is T^r, which has a finishing time of $1+\epsilon$. Any other allocation has a finishing time of at least 2. Hence, when T^r is not reached by A, the approximation factor obtained is a at least $\frac{2}{1+\epsilon}$. However, we know that with high probability T^r is not reached by M; Since $p_{r,m+1} \leq \frac{1}{m} + \epsilon$, and since machine r gets task $m+1$ in T^r, we know the probability that M outputs T^r is at most $\frac{1}{m} + \epsilon$. The expected approximation ratio of M is therefore at least $(1 - (\frac{1}{m} + \epsilon)) \times \frac{2}{1+\epsilon} + (\frac{1}{m} + \epsilon) \times 1$. Since this is true for any value of ϵ the theorem follows.

□

3 Application: Workload Minimization in Inter-Domain Routing

In this section, we show an application of our techniques to another non-utilitarian multi-parameter problem – *workload minimization in inter-domain routing*. Feigenbaum, Papadimitriou, Sami, and Shenker formulated the inter-domain routing problem as a distributed mechanism design problem [16] (inspired by the extensive literature on the real-life problem of inter-domain routing in the Internet). In recent years several works that study their model and its extensions have been published [15, 17, 18]. All these works deal with the realization of utilitarian social-choice functions (cost minimization, welfare maximization), and focus on the efficient and distributed design of VCG mechanisms.

Workload minimization is a problem that arises naturally in the design of routing protocols, as we wish that no single Autonomous System (AS) will be overloaded with work. It can easily be shown that any such VCG mechanism performs very poorly with regards to workload minimization. Thus, while optimally minimizing the total cost, or maximizing the social welfare, the known truthful mechanisms for this problem can result in workloads that are very far from optimal (in which one AS is burdened by the traffic sent by all other ASes). We initiate the study of truthful workload minimization in inter-domain routing by presenting constant lower bounds that apply to any truthful mechanism (deterministic and randomized).

Formal Statement of the Problem

We are given an undirected graph $G =< N, L >$ (called the *AS graph*) in which the set of nodes N corresponds to the Autonomous Systems (ASes) of which the Internet is comprised. N consists of a *destination* node d, and n *source* nodes. The set of edges L corresponds to communication links between the ASes. Each source node i is a strategic agent. The number of packets (intensity of traffic) originating in source node i and destined for d is denoted by t_i.

Let $neighbours(i)$ be all the ASes that are linked to i in the AS graph. Each source node i has a *cost function* $c_i : neighbours(i) \rightarrow R_{\geq 0}$ that specifies the per-packet cost incurred by this node for carrying traffic, where $L_i \subseteq L$ is the set of links node i participates in. This cost function represents the additional internal load imposed on the internal AS network when sending a packet to an adjacent AS[3]. The cost function of each node is its private information[4]. In the single parameter version of this problem an AS i incurs the same per-packet cost c_i for sending traffic to each of its neighbors (i.e., $c_i(l_1) = c_i(l_2)$ for every l_1, l_2).

The goal is to assign all source nodes routes to d. This *route allocation* should form a confluent tree to the destination d. I.e., no node is allowed to transfer traffic to two adjacent nodes. We seek truthful mechanisms that output routing trees in which the workload imposed on the busiest node is minimized. Formally, let N_i^T be the set of all nodes whose paths in the routing tree T go through node i. Let $Next^T(i)$ be the node i transfers traffic to in T. We wish to minimize the expression $max_i\ \Sigma_{j \in N_i^T}\ t_j \times c_i((i, Next^T(i))$ over all possible routing trees T. The problem of load minimization arises naturally in inter-domain routing as we require that no single AS will be overloaded with work.

Hardness of the Single-Parameter Case

It is easy to show (via a simple reduction from Partition) that even the single-parameter version of the workload-minimization problem is NP-hard. However, is it at all possible to optimally solve this problem in a truthful manner? The answer to this question is yes (we note that this is also the case in the single-parameter version of machine-scheduling [3]).

Lemma 5 *There exists a truthful, deterministic, exponential-time mechanism that always finds a workload-minimizing route allocation in the single-parameter case.*

Proof: The mechanism M simply goes over all possible route allocations and outputs the optimal one with regards to workload-minimization. As in [3], our truthful mechanism outputs the lexicographically-minimal optimal route allocation; That is, let a and b be two optimal route allocations (if two such allocations exist). Let $a_1, ..., a_n$ be a decreasing order of the workloads of the different nodes in a. Similarly, let $b_1, ..., b_n$ be a decreasing order of the workloads of the different nodes in b. Let $j \in [n]$ be the first index such that $a_j \neq b_j$ or $j = n$ if no such index exists. The mechanism will choose a if $a_j < b_j$, b if $b_j < a_j$, and otherwise according to a predefined determinstic tie breaking rule.

Obviously, the mechanism always outputs an optimal solution. We are left with proving the truthfulness of the mechanism. It is well known that a mechanism is truthful in a single-parameter setting such as ours *iff* it is weakly monotone [3]. Let a be the route allocation M outputs when the per-packet cost of i is c_i, and the per-packet costs of the other nodes are $c_{-i} = c_1, ..., c_{i-1}, c_{i+1}, ..., c_n$. Let b be the route allocation M outputs when the per-packet cost of i is c'_i, and the per-packet costs of the other nodes are c_{-i}. Weak monotonicity states that if $c_i < c'_i$ then $k_i \geq k'_i$, where k and k' are the number of packets that go through i in a and b respectively (and this is true for every node i, for every vector of costs per-packet c_{-i} of the other nodes, and for every two costs per-packet $c_i \neq c'_i$).

Fix a node i. Assume, by contradiction, that there are $c_i < c'_i$, and c_{-i} such that $k < k'$. Let $a_1, ..., a_n$ and $b_1, ..., b_n$ be defined as before. Let $j \in [n]$ be the first index such that $a_j \neq b_j$ (as before). If the two allocations are identical then no such j exists. However, note that in this case if node i declares c'_i then the allocation b will not be chosen (because $c'_i k_i < c'_i k'_i$ and so a comes before b in the lexicographic order). This contradicts the definition of b. We now turn to a sketch of a case by case analysis.

[3] In the formulation of the problem in [16], a node does not incur a cost for packets that originate in that node. However, as we are interested in workload minimization, this is not the case in our formulation.

[4] As we are interested in proving lower bounds we can restrict our attention to the model in which the t_is are common knowledge.

Case 1: $a_j < b_j$. It is not hard to verify that if $a_j < b_j$ and i declares c_i' then a comes before b in the lexicographic order (a contradiction to the definition of b).

Case 2: $b_j < a_j$. The are four sub-cases to consider:

- $c_i'k_i' \leq b_j < a_j$: In this case it can be shown that if i declares c_i then a will not be chosen (simply by showing that b comes before a in the lexicographic order). This contradicts the definition of a.
- $b_j < c_i'k_i' < a_j$: The same analysis as in the previous subcase applies to this sub-case as well.
- $b_j < a_j < c_i'k_i'$: If i declares c_i' then a is chosen and not b (it can easily be shown that a comes before b in the lexicographic order). This contradicts the definition of b.
- $b_j < a_j = c_i'k_i'$: Clearly, here $c_i'k_i < a_j = c_i'k_i'$. If i declares c_i' and b is chosen, then $a_{j+1} \geq b_j$. However, if this is the case then a will not be chosen if i decreases its cost value and declares $c_i < c_i'$. A contradiction.

$\square$

Hardness of the Multi-Parameter Case

[16] presents a truthful polynomial-time VCG mechanism that always outputs the cost-minimizing tree (a tree that minimizes the total sum of costs incurred for the packets sent to to d). We begin our discussion on the multi-parameter version of the workload minimization problem by showing that this VCG mechanism obtains an n-approximation for the multi-parameter version of our problem (and hence also for the single-parameter version) in polynomial time.

Theorem 3.1 *There is a truthful polynomial-time deterministic n-approximation mechanism for the workload minimization problem in inter-domain routing.*

Proof: We prove that any mechanism that minimizes the total cost provides an n-approximation to the minimal workload. Hence, the mechanism of [16] obtained the required approximation ratio.

Denote by T the cost-minimizing routing-tree and by T' the workload-minimizing routing-tree. Let $C(T)$ and $C(T')$ be the total costs of T and T' respectively. Let $W_i(T')$ be the workload on node i in T', and let $W(T') = max_i W_i(T')$. Notice, that (by the definition of T') $W(T')$ is the value of optimal solution for the workload-minimization problem. By contradiction, assume that $W(T') < \frac{C(T)}{n}$. Observe, that $C(T') = \Sigma_i W_i(T')$ (simple summation arguments). However, if this is the case then $C(T') = \Sigma_i W_i(T') \leq n \times W(T') < n \times \frac{C(T)}{n} = C(T)$. This contradicts the optimality of T for the cost-minimization problem. $\square$

Unfortunately, it can be shown that any mechanism that minimizes the total cost (and in particular the mechanism in [16]) cannot obtain a good approximation ratio.

Claim 3.2 *Any mechanism that minimizes the total cost of the routing tree cannot achieve an approximation ratio better than n for the workload minimization problem in inter-domain routing.*

Proof: Consider the routing instance in figure 1. Each source node has a single packet it wishes to send to the destination. The number beside every directed link (u, v) in these figures represents the cost u incurs for transferring a packet to v. Assume that all the other values assigned by the source nodes to links are very large (say $100,000$). Observe, that any total-cost minimizing mechanism would choose the routing tree in which both II and III send packets through I, and I forwards packets directly to d. This means that the workload on I is 3. However, if all nodes chose to send their packets directly to d we would reach a workload of $1 + \epsilon$. The example in figure 1 can easily be generalized to n source nodes. $\square$

Therefore, there is a tradeoff between the goal of minimizing the total-cost and the goal of minimizing the workload. It would be interesting to construct a truthful mechanism that optimizes (or at least closely approximates) the minimal workload. We present two negative results for this problem:

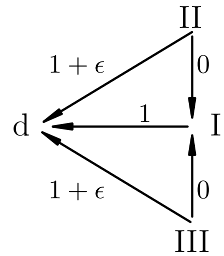

Figure 1:

Theorem 3.3 *No truthful deterministic mechanism for minimizing the workload in inter-domain routing can obtain an approximation ratio better than* $\frac{1+\sqrt{5}}{2} \approx 1.618$.

Proof:

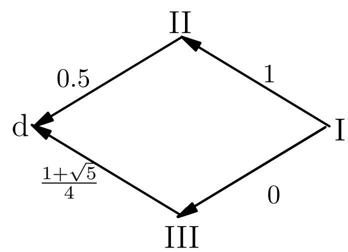

Figure 2:

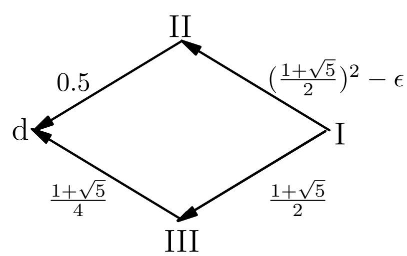

Figure 3:

This proof is similar to the proof of Theorem 2.2. To prove the lower bound consider the instances of the workload-minimization problem with 3 source nodes I, II, III depicted in figures 2 and 3. Each source node has a single packet it wishes do send to the destination. The number beside every directed link (u, v) in these figures represents the cost u incurs for transferring a packet to v. Assume that all the other values assigned by the source nodes to links are very large (say $100,000$). Denote the instance in figure 2 by INS and the instance in figure 3 by INS'. Observe that only the cost function of node I is different in INS and INS'. We denote the cost function of I in INS by c_I and his cost function in INS' by c'_I.

Assume, by contradiction, that M is a truthful deterministic mechanism that obtains an approximation ratio better than $\frac{1+\sqrt{5}}{2}$. Observe, that for instance INS M must direct the traffic originating in node I through node II (otherwise this contradicts the fact that M obtains an approximation ratio better than

$\frac{1+\sqrt{5}}{2}$). Similarly, for instance INS' M must direct the traffic originating in node I through node III. However, this violates the monotonicity of M as $1 + 1.618 = c_I((I, II)) + c'_I((I, III)) > c_I((I, II)) + c'_I((I, III)) = (1.618)^2 - \epsilon$.

□

Theorem 3.4 *No universally truthful randomized mechanism for minimizing the workload in inter-domain routing can obtain an approximation ratio better than* $\frac{3+\sqrt{5}}{4}$.

Proof: This proof is similar to the proof of Theorem 2.6. We define INS and INS' as in the proof of Theorem 3.3. Consider the uniform distribution over INS and INS'. Let M be a truthful deterministic mechanism. As shown in the proof of Theorem 3.3, M cannot achieve an approximation better than $\frac{1+\sqrt{5}}{2}$ on both INS and INS' due to its monotonicity. Therefore, the expected approximation of M is at least $\frac{1}{2} \times 1 + \frac{1}{2} \times \frac{1+\sqrt{5}}{2} \approx 1.309$. □

4 On Non-Utilitarian Fairness

In many scenarios, we desire to implement a utilitarian social choice function. A well studied example of such a scenario is social-welfare maximization in combinatorial auctions. In a combinatorial auction we wish to allocate m indivisible items $1, ..., m$ to n agents $1, ..., n$. Each agent i is defined by a valuation function $v_i : 2^{[m]} \rightarrow R_{\geq 0}$. We assume that for every i $v_i(\emptyset) = 0$ (free disposal) and for every two bundles of items $S, T \subseteq [m]$ such that $S \subseteq T$ $v_i(S) \leq v_i(T)$ (monotonicity). The goal is to partition the items into disjoint sets $S_1, ..., S_n$ such that the expression $\Sigma_i v_i(S_i)$ is maximized.

Utilitarian functions represent the "overall content" of the agents, as they maximize the sum of agents' values. This notion of *fairness* is but one of several that have been considered (explicitly and implicitly) in mathematical, economic and computational literature. A well known example of non-utilitarian fairness is the cake-cutting problem, presented by the Polish school of mathematicians in the 1950's (Steinhaus, Banach, Knaster [38]). In recent years, fair allocations of indivisible items (other than social-welfare maximization) have also been studied [25, 30] (these can be regarded as discrete versions of the cake-cutting problem).

In this section, we discuss three general notions of non-utilitarian fairness – Max-Min fairness, Min-Max fairness, and envy-minimization. We prove several general results; In particular, we show that Max-Min fairness is inapproximable within *any ratio*, even for extremely restricted special cases. In contrast, we show that Min-Max fairness (which is a generalization of both the scheduling and workload-minimization problems considered in this paper) can always be truthfully approximated to within a ratio of n from optimum via a simple VCG mechanism. This can be shown to be the best approximation ratio possible for Max-Min fairness (in the general case). Finally, we make use of our techniques to prove a lower bound for the envy-minimization problem.

Max-Min Fairness

While utilitarian functions maximize the "overall content" of the agents, the Max-Min social choice function is concerned with maximizing the content of the least satisfied agent. Formally, for every n-tuple of v_i valuations the Max-Min function assigns the alternative a that maximizes the expression $min_i\ v_i(a)$.

[25] proved that Max-Min fairness in allocations of indivisible items cannot be *optimally* implemented in a truthful manner. In recent years, non-truthful algorithms for this problem were designed [6], as well as algorithms that settle for restricted notions of truthfulness [8, 20]. We prove that no truthful deterministic mechanism can obtain *any* approximation ratio to the Max-Min fairness value. We prove this lower bound even for the case of 2 agents and 2 items.

Theorem 4.1 *No truthful deterministic mechanism can obtain* any *approximation to the Max-Min fairness value in the allocation of indivisible items. This holds even for the case of* 2 *agents and* 2 *items.*

Proof: Let $c > 1$. Consider an instance with two agents $1, 2$ and two goods a, b. Each agent $i = 1, 2$ has an additive valuation function. $v_1(a) = 2$ $v_1(b) = \frac{1}{c}$ $v_2(a) = 4 - \epsilon$ $v_2(b) = 1 + \epsilon$. Note, that the optimal allocation assigns a to agent 1 and b to agent 2, thus obtaining a Max-Min value of $1 + \epsilon$. Also note, that this allocation will also be chosen by any c-approximation mechanism.

We alter the valuation of agent 2 into v_2' such that $v_2'(a) = \frac{1}{c}$ $v_2'(b) = \frac{1}{c^2} - \epsilon$. The optimal Max-Min value is now $\frac{1}{c}$. Observe that any c-approximation mechanism must assign item b to agent 1 and item a to player 2. However, if this happens we have that:

$$(1 + \epsilon) + \frac{1}{c} = v_2(b) + v_2'(a) < v_2(a) + v_2'(b) = (4 - \epsilon) + \frac{1}{c^2}$$

This violates weak monotonicity, and so no truthful c-approximation mechanism exists. Since this is true for any $c > 0$ the theorem follows □

Min-Max Fairness

Min-Max fairness can be thought of as the dual notion of Max-Min fairness. It is relevant in settings in which each agent incurs a cost for every chosen alternative. While utilitarian functions minimize the "overall discontent", the Min-Max social choice function is concerned with minimizing the cost incurred by the least satisfied agent. Formally, for every n-tuple of v_i valuations the Max-Min function assigns the alternative a that minimizes the expression $max_i\, v_i(a)$.

Observe, that both the scheduling problem and the workload-minimization problem discussed in this paper, are in fact special cases of this notion of fairness. Studying Max-Min fairness in this more abstract setting enables us to state this simple observation – any Min-Max social-choice function can be truthfully approximated within a factor of n (recall that n is the number of agents) by a simple VCG mechanism. Since the best currently known approximation-mechanisms for both scheduling and workload-minimization are VCG-based, this result can be viewed as a generalization of both.

Theorem 4.2 *Let f be a Min-Max social choice function. Then, there exists a truthful deterministic mechanism that for every n-tuple of valuations $v_1, ..., v_n$ outputs an alternative a such that $max_i\, v_i(a)$ is an n-approximation to the value of the solution f outputs for these valuations.*

Proof: Let $v_1, ..., v_n$ be the valuation function of the agents, and let A be the set of alternatives. Let b be the allocation f outputs for $v_1, ..., v_n$. Consider the VCG mechanism that minimizes the total cost the agents incur. The truthfulness of this mechanism is guaranteed by the VCG technique. Let a be the allocation this mechanism outputs.

Assume, by contradiction, that $max_i\, v_i(a) > n \times max_i\, v_i(b)$. If this is the case, then

$$\Sigma_i v_i(b) \leq n \times max_i\, v_i(b) < max_i\, v_i(a) \leq \Sigma_i v_i(a)$$

However, this contradicts the optimality of a with regards to cost-minimization. □

This turns out to be the best approximation ratio that can be achieved in a truthful manner in the general case.

Theorem 4.3 *No truthful deterministic mechanism can obtain an approximation ratio better than n to the Max-Min fairness value.*

Proof: A careful look at the proof of Theorem 2.4 shows that if we do not insist that the valuations be additive (linear) then only weak monotonicity is required (rather than strong monotonicity). Thus, a similar proof shows that in the general case a lower bound that equals the number of agents can be obtained. □

Envy-Minimization

Lipton, Markakis, Mossel, and Saberi [30] presented the problem of finding envy-minimizing allocations of indivisible items. An envy-minimizing allocation of items is a partition of the m items into disjoint sets

$S_1, ..., S_n$ (agent i is assigned S_i) that minimizes the expression $max_{i,j}\ v_i(S_j) - v_i(S_i)$ (over all possible allocations). Intuitively, we wish to minimize the maximal envy an agent might feel by comparing his value for a bundle of items given to another agent to the value he assigns the items allocated to him. [30] proves several approximability results for this problem. The parameter considered in [30] is the *maximal marginal utility.*

Definition 6 *The maximal marginal utility α is defined as follows:*
$\alpha = max_{i\in[n],j\in[m],S\subseteq[m]}\ v_i(S \bigcup \{j\}) - v_i(S)$.

That is, α is the maximal value by which the value of an agent increases when one good is added to his bundle. [30] proves that there always exists an allocation of items with an envy value of at most α. [30] also exhibits a universally truthful randomized mechanism that obtains an approximation of $O(\sqrt{\alpha} n^{\frac{1}{2}+\epsilon})$ w.h.p. for large values of n.

Lipton et al. are interested in the question of whether there are truthful mechanisms that produce allocations with minimal or bounded envy. [30] shows that no truthful mechanism can guarantee a perfect solution (minimum envy). We strengthen this lower bound by showing that no truthful deterministic mechanism can guarantee an allocation that has an envy value within α from optimal.

Theorem 4.4 *No truthful deterministic mechanism can guarantee an allocation that has an envy within α from optimal.*

Proof: Let M be a truthful deterministic mechanism for this problem Consider an instance with 2 agents and 3 goods. Each agent $i = 1, 2$ has the same additive valuation function v_i that assigns any of the single items a value of 1. Observe, that in this case $\alpha = 1$. Notice, that the minimal envy for this instance is 1 (simply assign two items to one of the agents and one item to the other). Hence, if M assigns all items to one of the agents the envy of the other is precisely 3, which is a 2α distance from optimal.

We are left with the case in which one of the agents receives two items and the other is given one item. Assume w.l.o.g. that agent 1 is given items $1, 2$ and agent 2 is given item 3. We now change the valuation function of agent 1 into the following additive valuation:

$$v_1(j) = \begin{cases} 1+\epsilon, & j = 1, 2 \\ \epsilon, & j = 3 \end{cases}$$

Observe, that now $\alpha = 1 + \epsilon$. Also observe that the minimal envy for this new instance is ϵ (e.g., assign item 1 to agent 1 and items $2, 3$ to agent 2). However, the reader is encouraged to verify that the monotonicity of M dictates that the allocation remain the same even after the alteration of the valuation of agent 1. Therefore, we end up with an allocation in which the envy of agent 2 is 1. As $\alpha = 1 + \epsilon$ this is arbitrarily close to α. $\square$

Remark 4.5 *[30] also considers the social-choice function that aims to minimize the envy-ratio (defined therein) of the chosen allocation (over all possible allocations of goods). Using similar arguments to those in the proof of Theorem 4.4 one can easily show that no truthful deterministic mechanism for the envy-ratio minimization problem has an approximation factor better than 2. This result too can easily be extended to a weaker lower bound for truthful randomized mechanisms.*

Acknowledgements

We thank Ron Lavi, Noam Nisan, Chaitanya Swamy, and Amir Ronen for helpful discussions.

References

[1] Nir Andelman, Yossi Azar, and Motti Sorani. Truthful approximation mechanisms for scheduling selfish related machines. In *STACS*, 2005.

[2] Aaron Archer, Christos Papadimitriou, Kunal Talwar, and Eva Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agent. In *SODA*, 2003.

[3] Aaron Archer and Eva Tardos. Truthful mechanisms for one-parameter agents. In *IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2001.

[4] Aaron Archer and Eva Tardos. Frugal path mechanisms. In *Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02)*, 2002.

[5] Vincenzo Auletta, Roberto De Prisco, Paolo Penna, Giuseppe Persiano, and Carmine Ventre. New constructions of mechanisms with verification. In *ICALP*, 2006.

[6] Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *STOC*, 2006.

[7] Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi-unit combinatorial auctions, 2003. Working paper. Presented in TARK-03.

[8] Ivona Bezakova and Varsha Dani. Allocating indivisible goods, 2005. in ACM SIGecom Exchanges, Vol. 5.3.

[9] S. Bikhchandani, S. Chatterji, R. Lavi, A. Mu'alem, N. Nisan, and A. Sen. Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica*, To appear.

[10] G. Christodoulou, E. Koutsoupias, and A. Vidali. A lower bound for scheduling mechanisms. In *SODA*, 2007.

[11] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[12] Shahar Dobzinski and Noam Nisan. Approximations by computationally-efficient vcg-based mechanisms, 2006. Working paper.

[13] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC*, 2005.

[14] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *STOC*, 2006.

[15] Joan Feigenbaum, David Karger, Vahab Mirrokni, and Rahul Sami. Subjective-cost policy routing, 2005. in Proceedings of the Workshop on Internet and Network Economics, Lecture Notes in Computer Science, vol. 3828, Springer, Berlin.

[16] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A bgp-based mechanism for lowest-cost routing. *Distributed Computing*, 18:61–72, 2005.

[17] Joan Feigenbaum, Vijay Ramachandran, and Michael Schapira. Incentive-compatible inter-domain routing. In *EC*, 2006.

[18] Joan Feigenbaum, Rahul Sami, and Scott Shenker. Mechanism design for policy routing. *Distributed Computing*, 18:293–305, 2006.

[19] Amos Fiat, Andrew Goldberg, Jason Hartline, and Anna Karlin. Competitive generalized auctions. In *Proc. of the 34th ACM Symposium on Theory of Computing (STOC'02)*, 2002.

[20] Daniel Golovin. Max-min fair allocation of indivisible, 2005. Technical Report CMU-CS-05-144, Carnegie Mellon University.

[21] T. Groves. Incentives in teams. *Econometrica*, pages 617–631, 1973.

[22] Ron Holzman, Noa Kfir-Dahav, Dov Monderer, and Moshe Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 47:104–123, 2004.

[23] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *J. ACM*, 23:317–327, 1976.

[24] Anna Karlin, David Kempe, and Tami Tamir. Beyond vcg: Frugality of truthful mechanisms. In *FOCS*, 2005.

[25] Ron Lavi, Ahuva Mu'alem, and Noam Nisan. Towards a characterization of truthful combinatorial auctions, 2003. Working paper. Preliminary version presented in FOCS-03.

[26] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. In *FOCS*, 2005.

[27] Ron Lavi and Chaitanya Swamy, 2006. private communication.

[28] Daniel Lehmann, Liadan O'Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002.

[29] Jan Karel Lenstra, David B. Shmoys, and Eva Tardos:. Approximation algorithms for scheduling unrelated parallel machines. In *FOCS*, 1987.

[30] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *EC*, 2004.

[31] A. Mas-Collel, W. Whinston, and J. Green. *Microeconomic Theory*. Oxford university press, 1995.

[32] Ahuva Mu'alem and Noam Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI-02*, 2002.

[33] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[34] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting lindahl prices. *JET*, 2006.

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

[36] Y. Shoham P. Cramton and R. Steinberg (eds.). *Combinatorial Auctions*. MIT Press, 2006.

[37] Michael E. Saks and Lan Yu. Weak monotonicity suffices for truthfulness on convex domains. In *EC*, 2005.

[38] H. Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[39] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.

[40] A. C-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, 1977.

A Appendix

A.1 A Truthful $\frac{7m}{8}$-Approximation Mechanism for the Scheduling Problem with Unrelated Machines

Nisan and Ronen [33] present a truthful deterministic mechanism that obtains an m-approximation. For the case of 2 machines, they exhibit a universally-truthful randomized mechanism that obtains an approximation of $\frac{7}{4}$. We generalize this result by presenting a universally-truthful randomized mechanism that obtains an approximation-ratio of $\frac{7m}{8}$. This mechanism is based on that of [33]. We now turn to the description of our mechanism:

Input: m valuations v_i.

Output: An allocation $T = T_1,...,T_m$ of tasks, and payments $p_1, ..., p_m$ such that T has a makespan value which is a $\frac{7m}{8}$-approximation to the optimal makespan value, and the payments induce truthfulness.

The Mechanism:

1. For every machine i let $T_i = \emptyset$ and $p_i = 0$.
2. Partition the set of machines into two sets $S_1 = \{1, ..., \frac{m}{2}\}$ and $S_2 = \{\frac{m}{2} + 1, ..., m\}$.
3. For each task $j = 1, ..., n$ perform the following actions:
   - Let $v^1 = min_{i \in S_1} \ v_i(j)$ and let $I = argmin_{i \in S_1} \ v_i(j)$.
   - Let $v'^1 = min_{i \in S_1 - \{I\}} \ v_i(j)$.
   - Let $v^2 = min_{i \in S_2} \ v_i(j)$ and let $II = argmin_{i \in S_2} \ v_i(j)$.
   - Let $v'^2 = min_{i \in S_2 - \{II\}} \ v_i(j)$.
   - Randomly and uniformly choose a value $R \in \{0, 1\}$.
   - If $R = 0$ and $v^1 \leq \frac{4}{3}v^2$ set $T_I = T_I \bigcup \{j\}$ and set $p_I = p_I + min\{v'^1, \frac{4}{3}v^2\}$.
   - If $R = 0$ and $v^1 > \frac{4}{3}v^2$ set $T_{II} = T_{II} \bigcup \{j\}$ and set $p_{II} = p_{II} + min\{v'^2, \frac{3}{4}v^1\}$.
   - If $R = 1$ and $v^2 \leq \frac{4}{3}v^1$ set $T_{II} = T_{II} \bigcup \{j\}$ and set $p_{II} = p_{II} + min\{v'^2, \frac{4}{3}v^1\}$.
   - If $R = 1$ and $v^2 > \frac{4}{3}v^1$ set $T_I = T_I \bigcup \{j\}$ and set $p_I = p_I + min\{v'^1, \frac{3}{4}v^2\}$.
4. Allocate each machine i the tasks in T_i, and pay it a sum of p_i.

Remark A.1 *If m cannot be divided by 2 simply add the extra machine to either S_1 or S_2.*

Theorem A.2 *There exists a universally truthful randomized mechanism for the scheduling problem that obtains an approximation ratio of $\frac{7m}{8}$.*

Proof: We prove the theorem for the case that m can be divided by 2. The proof for the other case is similar. Our proof relies on the proof of Nisan and Ronen [33]. Observe, that the utility of each machine after the algorithm finishes is the sum of its utilities for the different tasks. Hence, it is sufficient to prove that for each individual task a machine has no incentive to lie. As in [33], this is guaranteed because the allocation of each task is in fact a weighted VCG mechanism (see [33] for further explanations), which is known to be truthful. Hence, this mechanism is universally truthful.

We now need to prove that the approximation ratio guaranteed by the mechanism is indeed $\frac{7m}{8}$. Let A be an instance of the scheduling problem with n tasks, and with m machines that have the valuation functions $v_1, ..., v_m$. We define an instance B of scheduling problem with n tasks, and with 2 machines that have the valuation function v'_1, v'_2, in the following way: For all $j \in [n]$ $v'_1(j) = min_{i \in S_1} \ v_i(j)$. Similarly, for all $j \in [n]$ $v'_2(j) = min_{i \in S_2} \ v_i(j)$. We denote by $M(A)$ and by $M(B)$ the makespan values our mechanism generates for A and B respectively. We denote by $O(A)$ and by $O(B)$ the optimal makespan values for A and B respectively.

First, notice that $M(A) \leq M(B)$. This is because applying our mechanism to B results in the same makespan value as applying it to A in the worst-case scenario in which tasks are always assigned to the same machines in S_1 and in S_2. It also holds that $M(B) \leq \frac{7}{4}O(B)$ because in the case that there are only two machines our mechanism is precisely that of [33], which guarantees a $\frac{7}{4}$ approximation ratio. We now have that $M(A) \leq \frac{7}{4}O(B)$. All that is left to show is that $O(B) \leq \frac{m}{2}O(A)$. Consider the optimal allocation of tasks for A. By giving all tasks assigned to machines in S_1 to machine 1 in B, and allocating all tasks assigned to machines in S_2 to machine 2 in B, we end up with a makespan value for B that is at most $\frac{m}{2}O(A)$. The theorem follows. $\square$