The Pareto Frontier of Inefficiency in Mechanism Design*

Aris Filos-Ratsikas[1,2][0000-0001-7868-8114], Yiannis Giannakopoulos[3][0000-0003-2382-1779], and Philip Lazos[4][0000-0001-9684-7609]

[1] University of Liverpool, United Kingdom, Aris.Filos-Ratsikas@liverpool.ac.uk
[2] École Polytechnique Fédérale de Lausanne, Switzerland
[3] TU Munich, Germany, yiannis.giannakopoulos@tum.de
[4] Sapienza University of Rome, Italy, lazos@diag.uniroma1.it

Abstract. We study the trade-off between the Price of Anarchy (PoA) and the Price of Stability (PoS) in mechanism design, in the prototypical problem of unrelated machine scheduling. We give bounds on the space of feasible mechanisms with respect to the above metrics, and observe that two fundamental mechanisms, namely the First-Price (FP) and the Second-Price (SP), lie on the two opposite extrema of this boundary. Furthermore, for the natural class of anonymous task-independent mechanisms, we completely characterize the PoA/PoS Pareto frontier; we design a class of optimal mechanisms $\mathcal{SP}_\alpha$ that lie *exactly* on this frontier. In particular, these mechanisms range smoothly, with respect to parameter $\alpha \geq 1$ across the frontier, between the First-Price ($\mathcal{SP}_1$) and Second-Price ($\mathcal{SP}_\infty$) mechanisms.
En route to these results, we also provide a definitive answer to an important question related to the scheduling problem, namely whether non-truthful mechanisms can provide better makespan guarantees in the equilibrium, compared to truthful ones. We answer this question in the negative, by proving that the Price of Anarchy of *all* scheduling mechanisms is at least n, where n is the number of machines.

Keywords: Mechanism design · Price of Anarchy · Price of Stability · Pareto Frontier.

1 Introduction

The field of *algorithmic mechanism design* was established in the seminal paper of Nisan and Ronen [15] and has ever since been at the centre of research in the intersection of economics and computer science. The research agenda put forward in [15] advocates the study of approximate solutions to interesting optimization problems, in settings where rational agents are in control of the

* Supported by ERC Advanced Grant 321171 (ALGAME), the Swiss National Science Foundation under contract No. 200021_165522 and the Alexander von Humboldt Foundation with funds from the German Federal Ministry of Education and Research (BMBF). A full version of this paper is available at [8].

input parameters. More concretely, the authors of [15] proposed a framework in which, not unlike classical approaches in approximation algorithms, algorithms that operate under certain limitations are evaluated in terms of their approximation ratio. In particular, in algorithmic mechanism design, this constraint comes from the requirement that agents should have the right incentives to always report their inputs *truthfully*. The corresponding algorithms, paired with appropriately chosen payment functions, are called *mechanisms*.

Another pioneering line of work, initiated by Koutsoupias [12] and popularized further by Roughgarden [18], studies the *inefficiency* of games through the notion of the *Price of Anarchy (PoA)*, which measures the deterioration of some objective at the worst-case Nash equilibrium. A more optimistic version of the same principle, where the inefficiency is measured at the *best* equilibrium, was introduced in [1], under the name of *Price of Stability (PoS)*.

Given the straightforward observation that mechanisms induce games between the agents that control their inputs, as well as the fact that truthfulness is typically a very demanding property, an alternative approach to the framework of Nisan and Ronen [15] is to design mechanisms that perform well *in the equilibrium*, i.e., they provide good PoA or PoS guarantees. This approach has been adopted, among others, by central papers in the field (e.g., see [17] and references therein) and is by now as much a part of algorithmic mechanism design as the original framework of [15]. An interesting question that has arisen in many settings is whether non-truthful mechanisms (evaluated at the worst-case equilibrium, in terms of their PoA) can actually outperform truthful ones (evaluated at the truth-telling, dominant strategy equilibrium), for a given objective.

While the literature that studies the concepts of PoA and PoS is long and extensive, there seems to be a lack of a *systematic approach* investigating the trade-off between the two notions *simultaneously*. More concretely, given a problem in algorithmic mechanism design, it seems quite natural to explore not only the best mechanisms in terms of the two notions independently, but also the mechanisms that achieve the best trade-off between the two. In a sense, this approach concerns a "tighter" optimality notion, as among a set of mechanisms with an "acceptable" Price of Anarchy guarantee, we would like to identify the ones that provide the best possible Price of Stability. Our main contribution in the current paper is the proposal of such a research agenda and its application on the canonical problem in the field, introduced in the seminal work of Nisan and Ronen [15], that of scheduling on unrelated machines.

1.1 Our Contributions

PoA/PoS trade-off: We propose the *research agenda of studying systematically the trade-off between the Price of Anarchy and the Price of Stability in algorithmic mechanism design*. Specifically, given a problem at hand and an objective function, we are interested in the trade-off between the PoA and the PoS of mechanisms for the given objective. We apply this approach on the prototypical problem of algorithmic mechanism design studied in [15], that of unrelated machine scheduling, where the machines are self-interested agents.

First, in Section 3, for the class of *all* possible mechanisms, we prove that PoA guarantees imply corresponding PoS lower bounds and vice-versa (Theorem 2), which allows us to quantify the possible trade-off between the two inefficiency notions in terms of a feasible region (see Fig. 1); we refer to the boundary of this region as the *inefficiency boundary*. Interestingly, two well-known mechanisms, namely the First-Price and the Second-Price mechanisms, turn out to lie on the extreme points of this boundary.

Next, in Section 4, for the well-studied class of task-independent and anonymous mechanisms, we are able to show a tighter feasibility region (Theorem 5). As a matter of fact, its inefficiency boundary turns out to *completely characterize* the achievable trade-off between the PoA and the PoS: we design a class of mechanisms (Section 4.2) called $\mathcal{SP}_\alpha$, parameterized by a quantity α, which are *optimal* in the sense that for any possible trade-off between the two inefficiency notions, there exists a mechanism in the class (i.e., an appropriate choice of α) that exactly achieves this trade-off (Theorem 6). In other words, we obtain an exact description of the *Pareto frontier of inefficiency* (see Fig. 2).

Our $\mathcal{SP}_\alpha$ mechanisms are simple and intuitive and are based on the idea of setting reserve prices *relatively* to the declarations of the fastest machines. While this is clearly not truthful, we prove that it induces the equilibria which are desirable for our results. More precisely, the choice of α enables us to "control" the set of possible equilibria in a way that allows us to achieve any trade-off on the boundary.

The Price of Anarchy of scheduling: Our results also offer insights in an other interesting direction. The inefficiency boundary result for general mechanisms is based on a novel monotonicity lemma (Lemma 1), which is quite different from the well-known *weak monotonicity* property [19]. Interestingly, we also use this lemma to prove a *general lower bound of* n on the PoA of *any* mechanism for the scheduling problem (Theorem 1), where n is the number of machines. This result contributes to the intriguing debate [11,10,4] of whether general mechanisms (that may be non-truthful, evaluated at the worst-case equilibrium) can outperform truthful ones (evaluated at the truth-telling equilibrium). Given that the best known truthful mechanism achieves an n-approximation, our results here provide a definitive, negative answer to the aforementioned question. As a matter of fact, in Theorem 4, we actually show that when evaluated at their worst-case equilibrium, truthful mechanisms are bound to perform even more poorly, as their PoA is unbounded.

Due to space constraints, all omitted proofs can be found in the full version of the paper [8].

1.2 Related Work

The (Selfish) Scheduling Problem: The scheduling problem on unrelated selfish machines is the prototypical problem studied by Nisan and Ronen [15] in 1999, when they introduced the field of algorithmic mechanism design. The authors consider the worst-case performance of truthful mechanisms on dominant strategy, truth-telling equilibria, and discover that the well-known Second-Price auction[5] has an approximation ratio of n for the problem, where n is the number of machines. Despite several attempts over the years, this is still the best-known truthful mechanism. On the other hand, the succession of the best proven lower bounds started with 2 in [15], improved to 2.41 in [5] and finally to 2.61 in [13]. Interestingly, Ashlagi et al. [2] showed a matching lower bound of n for *anonymous* mechanisms (i.e., mechanisms that do not take the identities of the machines into account) and whether there is a better mechanism that is not anonymous is still the most prominent open problem in the area.

The Truthful Setting vs the Strategic Setting: As we mentioned earlier, given that truthfulness is a very demanding requirement which imposes strict constraints on the allocation and payment functions, it is an interesting direction to consider whether *non-truthful* mechanisms could perform better, when evaluated in the worst-case equilibrium. In other words, for a given problem, one could ask the following question:

> *"Do there exist (non-truthful) mechanisms whose Price of Anarchy outperforms the approximation ratio guarantee of all truthful mechanisms?"*.

To differentiate, we will refer to the traditional approach of Nisan and Ronen [15] as the *truthful setting* and to the setting where all mechanisms are explored (with respect to their Nash equilibria) as the *strategic setting*.

Koutsoupias [11] studied the truthful setting for the problem of unrelated machine scheduling *without money* but he explicitly advocated the strategic setting as a future direction. This was later pursued in Giannakopoulos, Koutsoupias and Kyropoulou [10] for the same problem, where the authors answered the aforementioned question in the affirmative. The same approach was taken in [4] following the results of [7] on the limitations of truthful mechanisms for indivisible item allocation. In the literature of auctions, the strategic setting was studied even in domains for which an optimal truthful mechanism (the VCG mechanism) exists, motivated by the fact that non-truthful mechanisms are being employed in practice, with the Generalized Second-Price auction used by Google for the Adwords allocation being a prominent example [3]. We refer the reader to the survey of Roughgarden [17] for more details.

Somewhat surprisingly, although the exploration of different solution concepts besides dominant strategy equilibria was already explicitly mentioned as a future direction in [15], the strategic setting for the scheduling problem was not studied before our paper. As we mentioned earlier, the answer to the highlighted question above here is negative, but the setting proved out to be quite rich in terms of the achievable trade-off between the two different inefficiency notions.

To the best of our knowledge, ours is the first paper that proposes the systematic study of the trade-off between the Price of Anarchy and the Price of Stability. While preparing our manuscript, we became aware that a trade-off

[5] In the related literature, this mechanism is often referred to as the Vickrey-Clarke-Groves (VCG) mechanism.

between the two notions was very recently considered also in [16], though in a fundamentally different setting: the authors of [16] study a special case of covering games, originally introduced by Gairing [9], which is not inherently a mechanism design setup. One the contrary, our interest is in explicitly studying this trade-off in the area of algorithmic mechanism design, thus choosing the prototypical scheduling problem as the starting point.

2 Model and Notation

Let $\mathbb{R}_{\geq 0} = [0, \infty)$ denote the nonnegative reals and $\mathbb{N} = \{1, 2, \dots\}$ the positive integers. For any $n \in \mathbb{N}$, let $[n] = \{1, 2, \dots, n\}$. In the *strategic scheduling* problem (on unrelated machines), there is a set $N = \{1, \dots, n\}$ of *machines* (or agents) and a set $J = \{1, \dots, m\}$ of *tasks*. Each machine i has a *processing time* (or *cost*) $t_{i,j} \geq 0$ for task j. The induced matrix $\mathbf{t} \in \mathbb{R}_{\geq 0}^{n \times m}$ is the *profile* of processing times. For convenience, we will denote by $\mathbf{t}_i = (t_{i,1}, \dots, t_{i,m})$ the vector of processing times of machine i for the tasks and by $\mathbf{t}^j = (t_{1,j}, \dots, t_{n,j})^{\mathrm{T}}$ the vector of processing times of the machines for task j, so that $\mathbf{t} = (\mathbf{t}_1, \dots, \mathbf{t}_n) = (\mathbf{t}^1, \dots, \mathbf{t}^m)^{\mathrm{T}}$. The machines are *strategic* and therefore, when asked, they do not necessarily report their true processing times $\mathbf{t}$ but they rather use *strategies* $\mathbf{s} \in \mathbb{R}_{\geq 0}^{n \times m}$. To emphasize the distinction, we will often refer to $\mathbf{t}$ as the profile of *true* processing times. Adopting standard game-theoretic notation, we use $\mathbf{t}_{-i}$ and $\mathbf{s}_{-i}$ to denote the profile of true or reported processing times respectively, without the coordinates of the i'th machine.

A (deterministic, direct revelation) *mechanism* $\mathcal{M} = (\mathbf{x}, \mathbf{p})$ gets as input a strategy profile $\mathbf{s} \in \mathbb{R}^{n \times m}$ reported by the machines and outputs *allocation* $\mathbf{x} = \mathbf{x}(\mathbf{s}) \in \{0, 1\}^{n \times m}$ and *payment* $\mathbf{p} = \mathbf{p}(\mathbf{s}) \in \mathbb{R}_{\geq 0}^{n}$: $x_{i,j}$ is an indicator variable denoting whether or not task j is allocated to machine j, and p_i is the payment with which $\mathcal{M}$ compensates machine i for taking part in the mechanism. Thus, the allocation rule needs to satisfy $\sum_{i \in N} x_{i,j}(\mathbf{s}) = 1$ for all tasks j.

The *utility* of machine i under a mechanism $\mathcal{M} = (\mathbf{x}, \mathbf{p})$, given true running times $\mathbf{t}_i$ and a reported profile $\mathbf{s}$ by the machines, is

$$u_i^{\mathcal{M}}(\mathbf{s}|\mathbf{t}_i) = p_i(\mathbf{s}) - \sum_{j=1}^{m} x_{i,j}(\mathbf{s}) t_{i,j},$$

that is, the payment she receives from $\mathcal{M}$ minus the total workload she has to execute. This is exactly the reason why machines may lie about their true processing times; they will change their report $\mathbf{s}_i$ and deviate to another $\mathbf{s}_i'$ if this improves the above quantity. A stable solution with respect to such best-response selfish behaviour is captured by the well-known notion of an equilibrium. Given a mechanism $\mathcal{M}$ and a strategy profile $\mathbf{s}$, we will say that $\mathbf{s}$ is a *(pure Nash) equilibrium*[6] of $\mathcal{M}$ (with respect to a true profile $\mathbf{t}$) if, for every machine i and

[6] We will be interested in pure Nash equilibria in this paper; we provide a discussion on different solution concepts in the full version.

every possible deviation $\mathbf{s}'_i \in \mathbb{R}^m_{\geq 0}$,

$$u_i^{\mathcal{M}}(\mathbf{s}|\mathbf{t}) \geq u_i^{\mathcal{M}}(\mathbf{s}'_i, \mathbf{s}_{-i}|\mathbf{t}).$$

Let $\mathcal{Q}_{\mathbf{t}}^{\mathcal{M}}$ denote the set of pure Nash equilibria of mechanism $\mathcal{M}$ with respect to true profile $\mathbf{t}$. As is standard in the literature, we focus on the case where $\mathcal{Q}_{\mathbf{t}}^{\mathcal{M}} \neq \emptyset$ for all $\mathbf{t} \in \mathbb{R}^{n\times m}_{\geq 0}$ (see, e.g.,[15,4,10]).

Our objective is to design mechanisms that minimize the *makespan*

$$C^{\mathcal{M}}(\mathbf{s}|\mathbf{t}) = \max_{i\in N} \sum_{j=1}^{m} x_{i,j}(\mathbf{s}) t_{i,j},$$

that is, the total completion time if our machines run in parallel. For a matrix $\mathbf{t}$ of running times, let $\mathrm{OPT}(\mathbf{t})$ denote the optimum makespan, i.e., $\mathrm{OPT}(\mathbf{t}) = \min_{\mathbf{y}} \max_{i\in N} \sum_{j=1}^{m} y_{i,j} t_{i,j}$ where $\mathbf{y}$ ranges over all feasible allocation of tasks to machines. It is a well-known phenomenon that equilibria can result in suboptimal solutions, and the following, extensively studied, notions where introduced to quantify exactly this discrepancy: the *Price of Anarchy* (PoA) and the *Price of Stability* (PoS) of a scheduling mechanism $\mathcal{M}$ on n machines are, respectively,

$$\mathrm{PoA}(\mathcal{M}) = \sup_{m\in\mathbb{N}, \mathbf{t}\in\mathbb{R}^{n\times m}_{\geq 0}} \frac{\sup_{\mathbf{s}\in\mathcal{Q}_{\mathbf{t}}^{\mathcal{M}}} C^{\mathcal{M}}(\mathbf{s}|\mathbf{t})}{\mathrm{OPT}(\mathbf{t})}$$

$$\mathrm{PoS}(\mathcal{M}) = \sup_{m\in\mathbb{N}, \mathbf{t}\in\mathbb{R}^{n\times m}_{\geq 0}} \frac{\inf_{\mathbf{s}\in\mathcal{Q}_{\mathbf{t}}^{\mathcal{M}}} C^{\mathcal{M}}(\mathbf{s}|\mathbf{t})}{\mathrm{OPT}(\mathbf{t})}.$$

For simplicity, we will sometimes drop the $\mathcal{M}$, $\mathbf{t}$ and $\mathbf{s}$ in the notation introduced in this section, whenever it is clear which mechanism and which true or reported profile we are referring to.

2.1 Task-Independent Mechanisms

For a significant part of this paper, we will focus on the class of anonymous, task-independent mechanisms. This is a rather natural class of mechanisms; as a matter of fact, two of the arguably most well-studied and used mechanisms in practice, namely the First-Price and Second-Price, lie within this class.

Definition 1 (Task-independence). *A mechanism $\mathcal{M} = (\mathbf{x}, \mathbf{p})$ is called* task-independent *if each one of its tasks is allocated independently of the others. Formally, there exists a collection of single-task mechanisms $\{\mathcal{A}_j\}_{j=1,\dots,m}$, $\mathcal{A}_j = (\mathbf{y}^j, \mathbf{q}^j)$, such that, for any task j, any machine i, and for any strategy profile $\mathbf{s}$,*

$$\mathbf{x}^j(\mathbf{s}) = \mathbf{y}^j(\mathbf{s}^j) \qquad \textit{and} \qquad p_i(\mathbf{s}) = \sum_{j=1}^{m} q_i^j(\mathbf{s}^j).$$

We will refer to the single-task mechanisms $\mathcal{A}_j$ of the above definition as the *components* of $\mathcal{M}$. It is important to notice here that the definition does not require the mechanism to necessarily use the same component for all the tasks.

Another standard property in the literature of the problem is anonymity. The property can be defined generally (e.g., see [11,2]), but here we will define it for task-independent mechanisms. Since we are dealing with potentially non-truthful mechanisms, we need to handle the notion of anonymity in a more delicate way, in order to appropriately deal with ties.[7]

Definition 2 (Anonymity). *A single-task mechanism* $\mathcal{A} = (\mathbf{x}, \mathbf{p})$ *is* anonymous *if, for any permutation of the reports:*

- *The winning agent is permuted in the same way and receives the same payment. If there are multiple agents with the same bid, the winner is chosen to be the one with the largest index.*[8]
- *The payments of the agents that did not receive the task are permuted the same way. Additionally, losing agents with the same report receive the same payment.*

Formally, for any inputs $\mathbf{s}, \tilde{\mathbf{s}}$ *such that* $\tilde{\mathbf{s}} = \pi(\mathbf{s})$ *for some permutation* π*, if* $s_{i^\star}$ *is the report of the winner in* $\mathbf{s}$*, then the winner in* $\tilde{\mathbf{s}}$ *has index* $\max\{i \in N | \tilde{s}_i = s_{i^\star}\}$*. Additionally, let* π' *be any permutation such that* $\tilde{\mathbf{s}} = \pi'(\mathbf{s}) = \pi(\mathbf{s})$*. For any* $i \neq i^\star$ *we have* $p_{\pi(i)}(\tilde{\mathbf{s}}) = p_{\pi'(i)}(\tilde{\mathbf{s}}) = p_i(\mathbf{s})$*. In particular, if all entries in* $\mathbf{s}$ *are distinct:*

$$\mathbf{x}(\tilde{\mathbf{s}}) = \pi(\mathbf{x}(\mathbf{s})) \qquad \textit{and} \qquad \mathbf{p}(\tilde{\mathbf{s}}) = \pi(\mathbf{p}(\mathbf{s})).$$

A task-independent mechanism $\mathcal{M}$ *is anonymous, if all its components are anonymous (single-task) mechanisms.*

Perhaps the simplest and most natural mechanism that one can think of is the following, which assigns the task to the fastest machine (according to the declared processing times) and pays her her declaration.

Definition 3 (First-Price (FP) mechanism). *Assign each task* j *to the fastest machine* $\iota(j)$ *for it, i.e.* $\iota(j) \in \arg\min_{i \in N} s_{i,j}$ *(breaking ties arbitrarily), paying her her declared running time* $s_{\iota(i),j}$*; pay the remaining* $N \setminus \{\iota(j)\}$ *machines* 0 *for task* j*.*

Second-Price mechanisms have also been extensively studied and applied in auction theory, but also in strategic scheduling.

Definition 4 (Second-Price (SP) mechanism). *Assign each task* j *to the fastest machine* $\iota(j)$ *for it, i.e.,* $\iota(j) \in \arg\min_{i \in N} s_{i,j}$ *(breaking ties arbitrarily), paying her the declared processing time of the second-fastest machine, i.e.* $\min_{i \in N \setminus \{\iota(j)\}} s_{i,j}$*; pay the remaining* $N \setminus \{\iota(j)\}$ *machines* 0 *for task* j*.*

[7] For a more detailed discussion of anonymity and tie-breaking, see Remark 1 of the full version.

[8] This is without loss of generality for our results; the tie-breaking could be any fixed total order on the machines that does not depend on the reports.

Notice that both FP and SP mechanisms are task-independent and anonymous. Furthermore, SP is truthful. For a more detailed discussion on the connection between the different solution concepts (truthfulness vs Nash equilibria) and inefficiency notions (Price of Anarchy vs Price of Stability vs approximation ratio), we refer the reader to Section 2.2 of the full version.

3 The Inefficiency of *All* Mechanisms

We start with a lower bound of n for the Price of Anarchy of the scheduling problem, which applies to *all* mechanisms. The lower bound will be based on the following monotonicity lemma. We note that this monotonicity property is different from the weak monotonicity (WMON) used in the literature of truthful machine scheduling, in the sense that (a) it is global, whereas WMON is local and (b) it applies to the relation between the true processing times and the equilibria of the mechanism, rather than the actual allocations.

Lemma 1 (Equilibrium Monotonicity). *Let $\mathcal{M}$ be any mechanism for the scheduling problem. Let $\mathbf{t}$ be a profile of true processing times and let $\mathbf{s} \in \mathcal{Q}_{\mathbf{t}}$ be an equilibrium under $\mathbf{t}$. Denote by S_i the set of tasks assigned to machine i by $\mathcal{M}$ on input $\mathbf{s}$. Consider any profile $\hat{\mathbf{t}}$ such that for every machine i, $\hat{t}_{i,j} \leq t_{i,j}$ if $j \in S_i$ and $\hat{t}_{i,j} \geq t_{i,j}$ if $j \notin S_i$. Then $\mathbf{s} \in \mathcal{Q}_{\hat{\mathbf{t}}}$, i.e., $\mathbf{s}$ is an equilibrium under $\hat{\mathbf{t}}$ as well.*

Proof. Assume by contradiction that $\mathbf{s} \notin \mathcal{Q}_{\hat{\mathbf{t}}}$, which means that for the profile of processing times $\hat{\mathbf{t}}$, there exists some machine i that has a beneficial deviation $\mathbf{s}'_i$, i.e., $u_i(\mathbf{s}'_i, \mathbf{s}_{-i}|\hat{\mathbf{t}}) > u_i(\mathbf{s}|\hat{\mathbf{t}})$. Let S'_i be the set of tasks assigned to machine i under report $\mathbf{s}' = (\mathbf{s}'_i, \mathbf{s}_{-i})$ (and underlying true reports $\hat{\mathbf{t}}$). The difference in utility for machine i between profiles $\mathbf{s}'$ and $\mathbf{s}$ is

$$\Delta u_i(\hat{\mathbf{t}}) \equiv u_i(\mathbf{s}'|\hat{\mathbf{t}}) - u_i(\mathbf{s}|\hat{\mathbf{t}}) = p_i(\mathbf{s}') - p_i(\mathbf{s}) + \sum_{j \in S_i \setminus S'_i} \hat{t}_{i,j} - \sum_{j \in S'_i \setminus S_i} \hat{t}_{i,j}.$$

By the fact that s'_i is a beneficial deviation, it holds that $\Delta u_i(\hat{\mathbf{t}}) > 0$. Now consider the profile of processing times $\mathbf{t}$ and the same deviation $\mathbf{s}'_i$ of machine i. The increase in utility now is

$$\begin{aligned}\Delta u_i(\mathbf{t}) = p_i(\mathbf{s}') - p_i(\mathbf{s}) + \sum_{j \in S_i \setminus S'_i} t_{i,j} - \sum_{j \in S'_i \setminus S_i} t_{i,j} &\geq p_i(\mathbf{s}') - p_i(\mathbf{s}) \\ + \sum_{j \in S_i \setminus S'_i} \hat{t}_{i,j} - \sum_{j \in S'_i \setminus S_i} \hat{t}_{i,j} &= \Delta u_i(\hat{\mathbf{t}}),\end{aligned}$$

which holds because $t_{i,j} \geq \hat{t}_{i,j}$, if $j \in S_i$ and $t_{i,j} \leq \hat{t}_{i,j}$, if $j \notin S_i$. This implies that $\Delta u_i(\mathbf{t}) > 0$, which contradicts the fact that $\mathbf{s} \in \mathcal{Q}_{\mathbf{t}}$. □

Using this lemma, we can prove our first lower bound:

Theorem 1. *For any scheduling mechanism $\mathcal{M}$ for n machines, it must be that* $\mathrm{PoA}(\mathcal{M}) \geq n$.

Proof. Let $\mathcal{M}$ be any mechanism and consider a profile of true processing times $\mathbf{t}$ with n machines and n^2 tasks, where $t_{i,j} = 1$ for all machines i and all tasks j. Let $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ be a pure Nash equilibrium of $\mathcal{M}$ under $\mathbf{t}$. For each machine i, let S_i be the set of tasks assigned to that machine and note that there exists some machine k for which $|S_k| \geq n$. Let $T_k \subseteq S_k$ be any subset of S_k such that $|T_k| = n$.

Now consider the following profile $\hat{t}$ of processing times:

- For all $i \neq k$, $\hat{t}_{i,j} = 0$, for all $j \in S_i$ and $\hat{t}_{i,j} = t_{i,j}$, for all $j \notin S_i$.
- $\hat{t}_{kj} = 0$, for all $j \in S_k \backslash T_k$ and $\hat{t}_{kj} = t_{k,j}$, for all $j \notin S_k \backslash T_k$.

By Lemma 1, the profile $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ is a pure Nash equilibrium under $\hat{\mathbf{t}}$ and the allocation is the same as before, for a makespan of at least n, since machine k is assigned all the tasks in T_k. The optimal allocation will assign one task from T_k to each machine, the tasks from S_i to machine i for each $i \neq k$ and the tasks from $S_k \backslash T_k$ to machine k, for a total makespan of 1 and the Price of Anarchy bound follows. □

3.1 PoA/PoS Trade-off

In this section, we prove our main theorem regarding the trade-off between the Price of Anarchy and the Price of Stability. The theorem informally says that if the Price of Anarchy of a mechanism is small, then its Price of Stability has to be high.

Theorem 2. *For any scheduling mechanism $\mathcal{M}$ for $n \geq 2$ machines, and any positive real α,*

$$\mathrm{PoA}(\mathcal{M}) < \alpha \quad \Longrightarrow \quad \mathrm{PoS}(\mathcal{M}) \geq \frac{n-1}{\alpha} + 1.$$

By allowing α in Theorem 2 to grow arbitrarily large, we get the following:

Corollary 1. *Even for just two machines, if a scheduling mechanism has an optimal Price of Stability of* 1*, then its Price of Anarchy has to be unboundedly large.*

From the results of the section, as well as the trivial fact that $\mathrm{PoA}(\mathcal{M}) \geq \mathrm{PoS}(\mathcal{M})$ for any mechanism $\mathcal{M}$, we obtain a feasibility trade-off between the PoA and the PoS of scheduling mechanisms, which is illustrated in Fig. 1. We refer to the boundary of the shaded feasible region as the *inefficiency boundary*; the shape of the boundary follows from Theorem 2, as well as Theorem 1, since for $\mathrm{PoS}(\mathcal{M}) > 2 - \frac{1}{n}$ (or, in the language of Theorem 2, for $\alpha < n$), the best (i.e. largest) lower bound on the PoA is now given by Theorem 1.

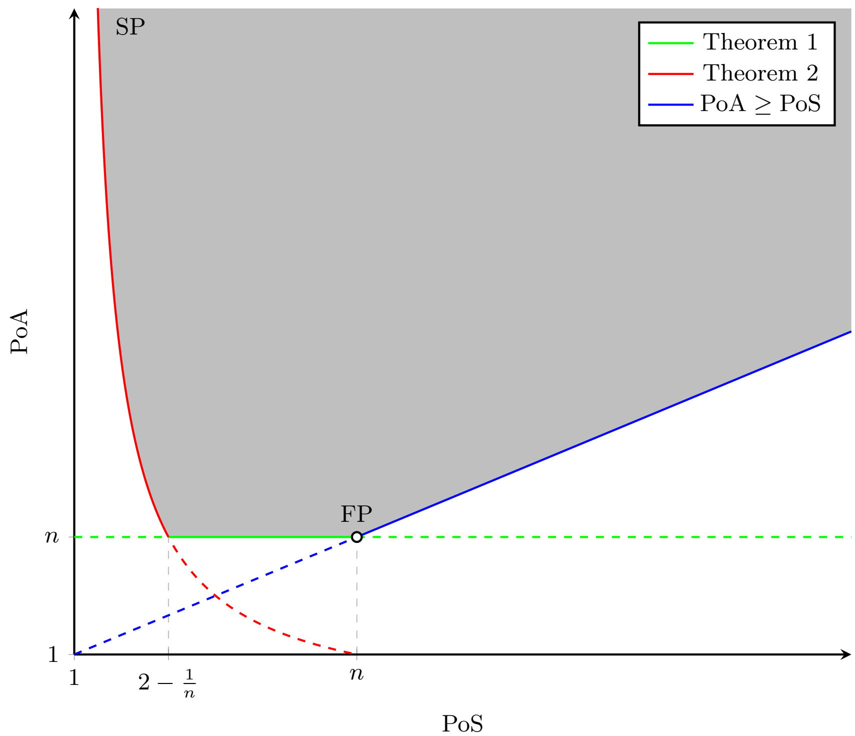

Fig. 1. The inefficiency boundary for general mechanisms, given by Theorem 2 (red line). Combined with the global PoA lower bound of Theorem 1 (green line) and the trivial fact that the PoS is at most the PoA (blue line), we finally get the grey feasible region.

Mechanisms on the Extrema of the Inefficiency Boundary: When looking for mechanisms on the Pareto frontier, the first ones that come to mind are perhaps the First-Price (FP) and Second-Price (SP) mechanisms, defined in Section 2, which are straightforward adaptations of the well-known First-Price auction and Second-Price auction mechanisms from the auction literature.

It follows from known results in the literature for the First-Price auction (see, e.g., [6]) that in every pure Nash equilibrium of the FP, each task is allocated to the machine with the smallest *true* processing time for the task. For the Second-Price mechanism, again it follows from known observations in the literature that while the mechanism is truthful, it has several other pure Nash equilibria as well. More precisely, for a task $j \in J$ and any machine $i \in N$, there exists an equilibrium for which task j is allocated to machine i. Therefore, we have the following.

Theorem 3. *For the First-Price mechanism, the* PoA *and the* PoS *are both* n*. For the Second-Price mechanism, the* PoA *of the mechanism is unbounded and the* PoS *is* 1*.*

Both the First-Price mechanism and the Second-Price mechanism will be obtained as corner-case mechanisms in the class that we will define in Section 4.2. Interestingly, it turns out that the bad PoA bound for the Second-Price Mechanism is a inherent characteristic of all truthful mechanisms. In other words, if one is interested in the set of *all* equilibria, they would have to reach out beyond truthful mechanisms.

Theorem 4. *The Price of Anarchy of any truthful mechanism is unbounded.*

From Theorem 2 and Theorem 3, it is clear that both FP and SP lie on the boundary of the PoA/PoS feasibility space (see Fig. 1).

4 The Pareto Frontier of Task-Independent Mechanisms

As we noted in the previous section, both the SP and FP mechanisms, which lie on the inefficiency boundary (see Fig. 1), are anonymous task-independent mechanisms. In this section, we will construct a tighter boundary on the PoA/PoS trade-off for the class of anonymous task-independent mechanisms. Furthermore, we will show that this boundary is actually tight, by designing a class of optimal mechanisms that lie exactly on it, meaning that for each point on the boundary, there is a mechanism in our class that achieves the corresponding PoA/PoS trade-off. Thus, this results in a *complete characterization of the Pareto frontier* between the PoA and the PoS. For an illustration, see Fig. 2.

4.1 PoA/PoS Trade-off

We start with the theorem that gives us the improved boundary on the space of feasible task-independent and anonymous mechanisms. This is the red line in Fig. 2.

Theorem 5. *For any task-independent anonymous scheduling mechanism $\mathcal{M}$ for n machines, and any real $\alpha > 1$,*

$$\mathrm{PoA}(\mathcal{M}) < (n-1)\alpha + 1 \quad \Longrightarrow \quad \mathrm{PoS}(\mathcal{M}) \geq \frac{(n-1)}{\alpha} + 1.$$

4.2 Optimal Mechanisms on the Pareto Frontier

Next, we will design a class of mechanisms, parameterized by a quantity α that will populate, in a smooth way, the boundary given by Theorem 5. Thus, these mechanisms achieve trade-offs that lie on the Pareto frontier of inefficiency for the class of task-independent and anonymous mechanisms.

Definition 5 (Second-Price mechanism with α-relative reserve price ($\mathcal{SP}_\alpha$)). *For $\alpha \geq 1$, $\mathcal{SP}_\alpha$ is the task-independent mechanism that, for each task j: finds a machine $k \in \arg\min_{i\in N} s_{i,j}$ and sets a* reserve price *at $r = \alpha \cdot s_{k,j}$; assigns the task to the fastest machine $\iota(j) \in \mathrm{argmin}_{i\in N} s_{i,j}$ (breaking ties-arbitrarily); pays machine $\iota(j)$ the amount $\min\{\min_{i\in N\setminus\{\iota(j)\}} s_{i,j}, r\}$; pays nothing to the remaining machines $N \setminus \iota(j)$.*

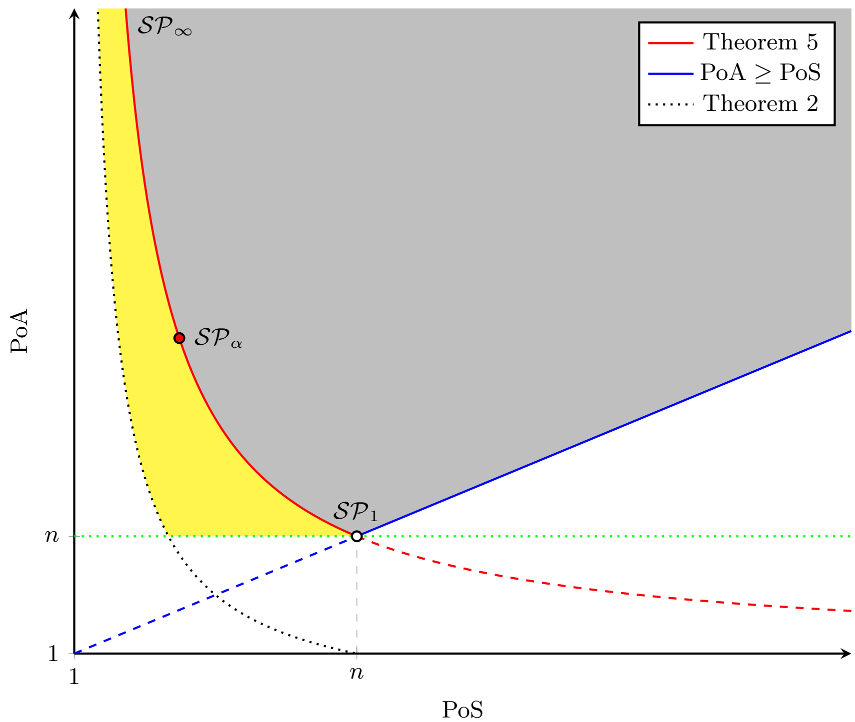

Fig. 2. The inefficiency boundary, for anonymous task-independent mechanisms, given by Theorem 5 (red line). Combined with the global PoA lower bound of Theorem 1 (green line) and the trivial fact that the PoS is at most the PoA (blue line), we finally get the grey feasible region. The family of mechanisms $\mathcal{SP}_\alpha$ described in Section 4.2 lies exactly on this boundary (red line), thus completely characterizing the *Pareto frontier* in a smooth way with respect to parameter $\alpha \geq 1$: on its one end ($\alpha = 1$) is the First-Price mechanism FP $= \mathcal{SP}_1$ and at the other ($\alpha \to \infty$) the Second-Price mechanism SP $= \mathcal{SP}_\infty$.

Informally, for each task j, the mechanism sets a reserve price which is α times larger than the smallest declared processing time, allocates the task to the fastest machine (according to the declarations) and pays the machine the minimum of the second-smallest declared processing time and the reserve price. What this mechanism achieves in terms of the equilibria that it induces is the following: assume that we create a *bucket* of tasks with *true* processing times at most α times larger than the smallest *true* processing time. Then, in every equilibrium of the mechanism, task j is allocated to some machine in the bucket and moreover, for any machine in the bucket, there exists some equilibrium under which $\mathcal{SP}_\alpha$ allocates the task to that machine (see the full version for a formal handling of this intuition). Referencing our discussion in Section 3.1, we remark that in the case of FP $= \mathcal{SP}_1$, the bucket contains only the fastest machine(s) for the task, and in the case of SP $= \mathcal{SP}_\infty$, the bucket contains the whole set of machines. We have the following theorem.

Theorem 6. *For $\mathcal{SP}_\alpha$ on n machines,*

- *the Price of Anarchy is at most $(n-1)\alpha + 1$,*
- *the Price of Stability is at most $\frac{n-1}{\alpha} + 1$.*

5 Discussion and Future Directions

On a general level, one could follow our agenda of studying the inefficiency trade-off between the Price of Anarchy and the Price of Stability for many other problems in algorithmic mechanism design, such as auctions [14,20], machine scheduling without money [11,10], or resource allocation [4], to name a few, for which the two inefficiency notions have already been studied separately.

In terms of the strategic scheduling setting, our work gives rise to a plethora of intriguing questions for future work, both on a technical and a conceptual level. The major open question is whether there exists a mechanism that achieves a better trade-off than that of Theorem 5, or in other words,

"Is the yellow region of Fig. 2 empty or not?"

If such a mechanism exists, it will most probably *not* be task-independent. Another question is whether we can remove anonymity from the statement for task-independent mechanisms. In that regard, we have come close, as captured by the following theorem.

Theorem 7. *For any task-independent scheduling mechanism $\mathcal{M}$ for n machines, and real $\alpha > 1$,*

$$\mathrm{PoA}(\mathcal{M}) < (n-1)\frac{\alpha}{\sqrt{2}} + 1 \quad \Longrightarrow \quad \mathrm{PoS}(\mathcal{M}) \geq \frac{(n-1)}{\alpha\sqrt{2}} + 1.$$

Another natural direction would be to consider different equilibrium notions, beyond pure Nash equilibria, or randomized scheduling mechanisms. We refer the reader to the discussion of the full version for a more insightful discussion of these avenues for future work.

References

1. Anshelevich, E., Dasgupta, A., Kleinberg, J., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. SIAM Journal on Computing **38**(4), 1602–1623 (2008). https://doi.org/10.1137/07068009
2. Ashlagi, I., Dobzinski, S., Lavi, R.: Optimal lower bounds for anonymous scheduling mechanisms. Math. Oper. Res. **37**(2), 244–258 (2012). https://doi.org/10.1287/moor.1110.0534

3. Caragiannis, I., Kaklamanis, C., Kanellopoulos, P., Kyropoulou, M., Lucier, B., Leme, R.P., Tardos, É.: Bounding the inefficiency of outcomes in generalized second price auctions. Journal of Economic Theory **156**, 343–388 (2015). https://doi.org/10.1016/j.jet.2014.04.010
4. Christodoulou, G., Filos-Ratsikas, A., Frederiksen, S.K.S., Goldberg, P.W., Zhang, J., Zhang, J.: Social welfare in one-sided matching mechanisms (extended abstract). In: Proceedings of AAMAS. pp. 1297–1298 (2016). https://doi.org/10.1007/978-3-319-46882-2_3
5. Christodoulou, G., Koutsoupias, E., Vidali, A.: A lower bound for scheduling mechanisms. Algorithmica **55**(4), 729–740 (2009). https://doi.org/10.1007/s00453-008-9165-3
6. Feldman, M., Lucier, B., Nisan, N.: Correlated and coarse equilibria of single-item auctions. In: Proceedings of WINE. pp. 131–144 (2016). https://doi.org/10.1007/978-3-662-54110-4_10
7. Filos-Ratsikas, A., Frederiksen, S.K.S., Zhang, J.: Social welfare in one-sided matchings: Random priority and beyond. In: Proceedings of SAGT. pp. 1–12 (2014). https://doi.org/10.1007/978-3-662-44803-8_1
8. Filos-Ratsikas, A., Giannakopoulos, Y., Lazos, P.: The Pareto Frontier of Inefficiency in Mechanism Design. CoRR **abs/1809.03454** (Sep 2018), https://arxiv.org/abs/1809.03454
9. Gairing, M.: Covering games: Approximation through non-cooperation. In: Proceedings of WINE. pp. 184–195 (2009). https://doi.org/10.1007/978-3-642-10841-9_18
10. Giannakopoulos, Y., Koutsoupias, E., Kyropoulou, M.: The anarchy of scheduling without money. Theoretical Computer Science **778**, 19 – 32 (2019). https://doi.org/10.1016/j.tcs.2019.01.022
11. Koutsoupias, E.: Scheduling without payments. Theory of Computing Systems **54**(3), 375–387 (2014). https://doi.org/10.1007/s00224-013-9473-0
12. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. Computer Science Review **3**(2), 65–69 (2009). https://doi.org/10.1016/j.cosrev.2009.04.003
13. Koutsoupias, E., Vidali, A.: A lower bound of $1+\varphi$ for truthful scheduling mechanisms. Algorithmica **66**(1), 211–223 (2013)
14. Lucier, B., Singer, Y., Syrgkanis, V., Tardos, E.: Equilibrium in combinatorial public projects. In: Proceedings of WINE. pp. 347–360 (2013). https://doi.org/10.1007/978-3-642-45046-4_28
15. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior **35**(1/2), 166–196 (2001). https://doi.org/10.1006/game.1999.0790
16. Ramaswamy, V., Paccagnan, D., Marden, J.R.: Multiagent coverage problems: The trade-off between anarchy and stability. CoRR **abs/1710.01409** (Jul 2018), http://arxiv.org/abs/1710.01409
17. Roughgarden, T., Syrgkanis, V., Tardos, E.: The price of anarchy in auctions. Journal of Artificial Intelligence Research **59**, 59–101 (2017). https://doi.org/10.1613/jair.5272
18. Roughgarden, T., Tardos, É.: How bad is selfish routing? J. ACM **49**(2), 236–259 (2002). https://doi.org/10.1145/506147.506153
19. Saks, M., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings of the 6th ACM Conference on Electronic Commerce. pp. 286–293 (2005). https://doi.org/10.1145/1064009.1064040
20. Syrgkanis, V., Tardos, E.: Composable and efficient mechanisms. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing. pp. 211–220 (2013). https://doi.org/10.1145/2488608.2488635