
Parsimonious Predictions for Strategyproof Scheduling

Richard Cole

Anupam Gupta

Pranav Jangir

Department of Computer Science
New York University
New York, NY 10012.

Abstract

We consider the problem of scheduling m jobs on n unrelated strategic machines to minimize the maximum load of any machine. As the machines are strategic they may misreport processing times to minimize their own load. The pioneering work of Nisan and Ronen gave an n -approximate deterministic strategyproof mechanism for this setting, and this was recently shown to be best possible by the breakthrough results of Christodoulou et al. This large approximation guarantee begs the question: how can we avoid these large worst-case results. In this work, we use the powerful framework of algorithms with (machine-learned) predictions to bypass these strong impossibility results. We show how we can predict $O(m + n)$ values to obtain a deterministic strategyproof algorithm whose makespan is within a constant factor of the optimal makespan when the predictions are correct, and $O(n)$ times the optimum no matter how poor the predictions are.

1 Introduction

We consider the NP-hard problem of scheduling jobs on unrelated machines to minimize the makespan, i.e., the load of the most loaded machine, in the setting where the machines are agents which can behave strategically. In this game-theoretic setting, introduced by Nisan and Ronen (2001), only the machines know the processing times for the jobs. Since they suffer a cost for processing jobs (i.e., the processing time/size of the job), they have an incentive to misreport sizes (i.e., to overstate or understate them) to the mechanism in order to get a more desirable bundle of jobs. Our goal is to design a mechanism that achieves (a) a good approximation ratio, so that we always find a schedule with makespan comparable to that of the optimal solution on the true processing sizes, and (b) strategyproofness, so that no machine has any incentive to misstate their processing times.

In the non-strategic setting of classical approximation algorithms, the unrelated machines scheduling problem can be approximated to a factor of 2 (due to works of Lenstra et al. (1990) and Shmoys and Tardos (1993)), and it is NP-hard to approximate it better than a factor of $3/2$, leaving a small gap between the constants in the upper and lower bounds. But when the machines can behave strategically, it is now known that the optimal approximation factor of any *strategyproof* mechanism is not a constant, but exactly n . The paper of Nisan and Ronen (2001) gave a mechanism achieving the n -approximation, and now a matching lower bound of n for any strategyproof mechanism is known due to Christodoulou et al. (2022, 2023). This leaves us with a large gap between the results that are achievable in the non-strategic and strategic settings in the worst-case.

Since worst-case instances may be pathological and the analysis pessimistic, we can ask: *how can we give mechanisms that perform better in non-worst-case settings?* An important approach of *algorithms with predictions* or *learning-augmented framework*, has gained momentum, giving a balance between the robustness guarantees of the worst-case model, and better results when given

machine-learned information. Concretely, the model asks: can we perform better if we have some (machine-learned) untrusted predictions about the instance? We seek the following two outcomes:

1. *Robustness*: this requires that the mechanism’s performance is not much worse than the performance without predictions, regardless of whether the predictions are correct or not.
2. *Consistency*: this asks that if the predictions are indeed correct (i.e., the predictions indeed conform to the parameters/features of the true instance they are supposed to be predicting), then the mechanism’s performance is close to that achievable if it truly believed the predictions. We may also ask for an error-tolerant version of consistency, where the algorithm’s performance relates to the “error” in the predictions.

This line of work was initiated in the setting of online algorithms by Mahdian et al. (2012), Purohit et al. (2018) and Lykouris and Vassilvitskii (2021) (see also the survey article by Mitzenmacher and Vassilvitskii (2022)); it was used in strategic settings by Agrawal et al. (2024), Xu and Lu (2022) and Gkatzelis et al. (2022). (See the annotated bibliography compiled by Balkanski et al. (2024)).

The unrelated machine scheduling problem was studied in this learning-augmented setting by Xu and Lu (2022) and Agrawal et al. (2024), who observed that following the predictions blindly gives an excellent consistency of 1, but has arbitrarily poor robustness if the predictions are faulty; to remedy this, they gave algorithms with bounded robustness and consistency. The current best result is due to Balkanski et al. (2023), who gave the SCALED GREEDY algorithm that is $(1 + c)n$ -robust and $(4 + 2/c)$ -consistent, for any $c > 0$. Their algorithm uses the entire matrix $\hat{p} \in \mathbb{R}_+^{n \times m}$ of processing times, where \hat{p}_{ij} is the prediction for the true processing time p_{ij} of the job j on machine i .

Christodoulou et al. (2024) explore whether we can use a smaller set of predictions, namely *output predictions*, where the mechanism is merely given the predicted machine for each job. They give an algorithm that uses just these m many predictions, and that is $O(1)$ -robust and (n^2) -consistent. Their algorithm—and indeed, all the above algorithms in this domain—compute machine “biases”. They use a greedy-like algorithm using the reported processing times weighted by these biases to assign the jobs. They also show that, given just assignment predictions, no such weighted VGC mechanism can obtain much better trade-offs.

In light of these results, we ask: can we get a different set of linearly many predictions, i.e., $O(n + m)$ quantities, and achieve the same guarantees?

1.1 Our Results

Our main result answers this question in the affirmative, and gives nearly optimal robustness/consistency tradeoffs with only a linear number of predictions.

Theorem 1.1 (Informal Main Theorem). *There exists a strategyproof mechanism that takes predictions for the value of the optimal makespan, a “bias” for each machine, and a “good” machine for each job, and produces an assignment which is $(1 + c)n$ -robust, and $(2 + \frac{1}{c-1})$ -consistent for any choice of $c > 1$.*

This achieves essentially the same guarantees as the result of Balkanski et al. (2023) (our results replace a $1/c$ term in the consistency guarantee by $1/(c-1)$), while using far fewer predictions—only linearly many instead of quadratically many. And using these few extra predictions (corresponding to the “biases” for each machines) to avoid the lower bounds of Christodoulou et al. (2024).

Our approach is based on an essentially simple idea: instead of trying to predict the optimal machines for each job, we predict the machines which would be output by an approximation algorithm. This algorithm follows the “relax-and-round” approach: it solves a linear programming (LP) relaxation for the unrelated machine scheduling problem, and then converts the fractional solution into an integral one. Crucially, the assignments given by this rounding algorithm are supported by linear programming duality. We use this in two ways: firstly, we use the machine variables of the dual program as the machine biases. Secondly, we use the structure of the dual solutions (e.g., complementary slackness) along with the output of the rounding process to prove robustness guarantees.

While this approach leads to an $O(n^2)$ -robust algorithm, the dual structure of the classical LP relaxation is not rigid enough. The next idea is to write a new LP relaxation for unrelated machine scheduling problem, which allows us to ensure that the duals are bounded, which gives us more

control over the robustness. The final ingredient is showing a good approximation guarantee for this new relaxation.

Finally, like Balkanski et al. (2023), we can extend our work to an error-tolerant version. In this setting, we are given a parameter $\eta > 1$, and predictions are considered “ η -good” if they correspond to some processing times that are within a factor of η of the true processing times. The goal now is to be consistent when the predictions are “ η -good”, otherwise the algorithm is required to be robust. The formal definitions and details of the following result appear in Appendix B.

Theorem 1.2 (Informal Error-Tolerant Theorem). *There exists a strategyproof mechanism that takes the above predictions for the value of the optimal makespan, the optimal dual solution and the predicted rounded solution, and produces an assignment which is $(1 + c)\eta^2$ n -robust (when the predictions are not η -good), and $\hat{\eta}^2\gamma(2 + \frac{1}{c-1})$ -consistent (when they are $\hat{\eta}$ -good for some $\hat{\eta} \leq \eta$) for any choice of $c > 1$.*

Comparing the Prediction Models. An advantage of parsimonious predictions is that they require only linear space to manipulate and store, compared to the quadratic storage used in the works of Xu and Lu (2022), Agrawal et al. (2024), Balkanski et al. (2023). It is an open problem to prove that learning these smaller number predictions from data has a smaller sample complexity. (We briefly discuss the learnability of the processing times in a PAC-like model in Appendix A.) That said, we can use the predictions given to Balkanski et al. (2023) (i.e., predictions for all the processing times) to obtain our results by first computing the predictions that our mechanism expects (which are all functions of the processing times), and using these $1 + n + m$ values for our mechanism.

1.2 Other Related Work

Strategyproof mechanisms are increasingly important in open and decentralized systems, since the resources belong to different agents who may act in their own interest and may not cooperate unless they are given the right incentives. In such settings, it is often important to ensure notions of non-manipulability and fairness, either for reasons of social welfare, or because of legal requirements. Strategyproof mechanisms can help ensure that no single agent can manipulate the outcome to their advantage. Specifically for settings of scheduling jobs to machines, note that machines may misreport processing times to reduce their workload, which could lead to poor scheduling decisions and delays for jobs. The first work for the unrelated machine scheduling problem in the strategyproof setting was due to Nisan and Ronen (2001). The lower bounds were given by Nisan and Ronen (2001), Christodoulou et al. (2009), Koutsoupias and Vidali (2013), Giannakopoulos et al. (2021), Dobzinski and Shaulker (2020), Ashlagi et al. (2012), Christodoulou et al. (2007), culminating in a matching lower bound of n in the recent works of Christodoulou et al. (2022, 2023).

The work of Christodoulou et al. (2024) considers the *output predictions* model, where the mechanism is merely given a predicted output for each request: this corresponds to a predicted machine for each job in our setting. They give an algorithm that is $(\beta + 1)$ -robust and (n^2/β) -consistent for any fixed $\beta \in [1, n]$; it assigns a bias of $r_{ij} = 1$ for the predicted machine i , and $r_{i'j} = n/\beta$ for every other machine i' , and chooses the weighted minimizer of $r_{ij}p_{ij}$. They also show that, given just assignment predictions, no weighted VGC mechanism can give a significantly better trade-off than β vs. n^2/β . In contrast, our work shows that if we are also given a prediction for OPT and the dual variables (one for each machine), we can achieve much stronger results.

The use of linear programs and duality as predictions is widespread: they have been used, e.g., the works of Lavastida et al. (2021), Lattanzi et al. (2020), Li and Xian (2021) uses linear/convex programs and their duals to perform load balancing in the online setting. Moreover, linear programming duals are used in the work of Dinitz et al. (2021); it uses learned duals as a “warm start” to speed up algorithms for finding maximum weight bipartite matchings; these ideas are further explored by Davies et al. (2023), Polak and Zub (2024), Sakaue and Oki (2022). However, our use in mechanism design, and in particular, our approach of controlling the variation in the duals to make the mechanism more “robust”, seems to not have been used before.

2 Definitions

An instance \mathcal{I} of the unrelated machine scheduling problem consists of (i) a set J of m jobs, (ii) a collection M of n machines, and (iii) a processing time p_{ij} for each job $j \in J$ and machine $i \in M$.

Let \mathbf{p} denote the vector of all processing times. Given an instance \mathcal{I} , an assignment of jobs to machines is a vector $\mathbf{x} \in \{0, 1\}^{n \times m}$, where $x_{ij} = 1$ if and only if job j is assigned to machine i . Such an assignment \mathbf{x} satisfies $\sum_i x_{ij} = 1$, and induces a *load* of $\sum_j p_{ij}x_{ij}$ on machine $i \in M$.

- The *makespan* of assignment \mathbf{x} is $\text{MS}(\mathbf{p}, \mathbf{x}) := \max_{i \in M} \sum_j p_{ij}x_{ij}$.
- Define the optimal (integer) makespan to be $\text{OPT}(\mathbf{p}) := \min_{\text{assignments } \mathbf{x}} \text{MS}(\mathbf{p}, \mathbf{x})$, where the minimum is taken over all possible assignments \mathbf{x} of jobs to machines, and the optimal (integer) assignment be denoted by $\mathbf{x}^* = \mathbf{x}^*(\mathbf{p})$.

Given an assignment \mathbf{x} , we can also associate an allocation function $\varphi : J \rightarrow M$ with it, where $\varphi(j) = i \iff x_{ij} = 1$. We frequently move between allocation functions φ and assignments \mathbf{x} , as needed. (In some cases, we consider $\mathbf{x} \in \{0, 1\}^{n \times m}$ vectors where $\sum_i x_{ij} \geq 1$, but we can always reduce some of the x_{ij} values to zero and get $\sum_i x_{ij} = 1$, without increasing the machine loads and the makespans.)

2.1 Strategyproofness

Consider an instance of the unrelated machine scheduling problem, where the machines report their processing times for all the jobs. Let $\mathbf{p} \in \mathbb{R}_+^{n \times m}$ be the true processing times for all machine-job pairs. (As is standard, we use \mathbf{p}_i to denote the i^{th} row of matrix \mathbf{p} , and \mathbf{p}_{-i} to denote the rest of the rows.) Now suppose some machine $i \in M$ reports $\mathbf{p}'_i \in \mathbb{R}_+^m$ as its vector of processing times (where \mathbf{p}'_i may or may not equal \mathbf{p}_i), and the other machines truthfully report $\mathbf{p}_{-i} \in \mathbb{R}_+^m$:

- let $\mathbf{x}(\mathbf{p}'_i, \mathbf{p}_{-i}) \in \{0, 1\}_+^{n \times m}$ be the assignment of jobs to machines, and
- let $\pi(\mathbf{p}'_i, \mathbf{p}_{-i}) \in \mathbb{R}_+^{n \times m}$ be the payments made by the mechanism to machines.

Define the *utility* of machine i to be

$$U_i(\mathbf{p}'_i, \mathbf{p}_{-i}) := \sum_j \pi_{i,j}(\mathbf{p}'_i, \mathbf{p}_{-i}) - \sum_j p_{ij}x_{ij}(\mathbf{p}'_i, \mathbf{p}_{-i}). \quad (1)$$

Observe that while both the map \mathbf{x} and the payments π are functions of the reports $(\mathbf{p}'_i, \mathbf{p}_{-i})$, the second term in the definition of utility uses the true processing times.

Definition 2.1 (Strategyproofness). An allocation function φ is *strategyproof* if there exists payments π such that for any machine i and for any misreport of the processing times vector \mathbf{p}'_i , we have $U_i(\mathbf{p}_i, \mathbf{p}_{-i}) \geq U_i(\mathbf{p}'_i, \mathbf{p}_{-i})$.

Definition 2.2 (Monotonicity). Let $\mathbf{x} = \mathbf{x}(\mathbf{p}_i, \mathbf{p}_{-i})$ and $\mathbf{x}' = \mathbf{x}(\mathbf{p}'_i, \mathbf{p}_{-i})$ denote the allocations of the mechanism when machine i reports \mathbf{p}_i and \mathbf{p}'_i respectively. An allocation mechanism is *monotone* if for every machine i and for every two reports of the machine \mathbf{p}_i and \mathbf{p}'_i , the associated allocations satisfy $\sum_{j \in J} (x_{ij} - x'_{ij})(p_{ij} - p'_{ij}) \leq 0$. Moreover, call the mechanism *item-wise monotone* if the allocations satisfy $(x_{ij} - x'_{ij})(p_{ij} - p'_{ij}) \leq 0 \quad \forall j \in J$.

Theorem 2.3 (Nisan and Ronen (2001), Saks and Yu (2005)). *For an allocation algorithm, there exists a payment function π such that (φ, π) is strategyproof if and only if the allocation algorithm satisfies the monotonicity property.*

Since any item-wise monotone mechanism is also monotone, the following corollary holds:

Corollary 2.4. *An allocation algorithm is strategyproof if it satisfies the item-wise monotonicity property. In other words, if for every machine i , suppose $\mathbf{x} = \mathbf{x}(\mathbf{p}_i, \mathbf{p}_{-i})$ and $\mathbf{x}' = \mathbf{x}(\mathbf{p}'_i, \mathbf{p}_{-i})$, and suppose these allocations satisfy $(x_{ij} - x'_{ij})(p_{ij} - p'_{ij}) \leq 0$ for every job j . Then the allocation algorithm is monotone, and hence there exists a payment function that gives strategyproofness.*

3 A New Linear Program Relaxation

We now give a new linear programming relaxation for the scheduling problem, such that the dual variables corresponding to the machines lie in some bounded range. It is this control on the dual variables allows us to show strong robustness guarantees without losing much on the consistency.

The standard linear programming relaxation for the unrelated machine scheduling problem takes an instance \mathcal{I} and a target makespan T , and defines the set of *permissible edges* $E(T, \mathbf{p}) := \{(i, j) \mid$

$p_{ij} \leq T$ by removing all the edges in $M \times J$ which correspond to processing times larger than the target value T . The *feasibility* linear program $P(T, \mathbf{p})$ is then defined as follows:

$$\begin{aligned} \sum_{i:(i,j) \in E(T, \mathbf{p})} x_{ij} &\geq 1 \\ \sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} x_{ij} &\leq T \quad \forall \text{ machines } i \in M \\ x &\geq 0. \end{aligned} \quad (2)$$

Again, the first constraints allow the fractional assignment to allocate more than 1 unit of job j , but any such solution can be transformed into one that satisfies these constraints with equality. Using inequalities simplifies the linear programming dual $D(T, \mathbf{p})$:

$$\begin{aligned} \max \sum_j \alpha_j - T \sum_i \beta_i & \quad (D(T, \mathbf{p})) \\ \alpha_j - \beta_i p_{ij} \leq 0 & \quad \forall (i, j) \in E(T, \mathbf{p}) \\ \alpha, \beta \geq 0. & \end{aligned}$$

Our new linear programming relaxation for the unrelated machine scheduling problem takes an instance \mathcal{I} and a target makespan T , and defines the set of *permissible edges* $E(T, \mathbf{p}) := \{(i, j) \mid p_{ij} \leq T\}$ by removing all the edges in $M \times J$ which correspond to processing times larger than the target value T . Then for any a scalar $c \geq 1$, the linear program relaxation $P_c(T, \mathbf{p})$ is the following:

$$\begin{aligned} \min \quad Z - 1/cn \sum_i Y_i & \quad (P_c(T, \mathbf{p})) \\ \sum_{i:(i,j) \in E(T, \mathbf{p})} x_{ij} &\geq 1 \quad \forall \text{ jobs } j \in J \\ \sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} x_{ij} - Z + Y_i &\leq T \quad \forall \text{ machines } i \in M \\ \mathbf{x}, \mathbf{Y}, Z &\geq 0. \end{aligned}$$

In other words, the primal constraint for machine $i \in M$ says:

$$\sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} x_{ij} + Y_i \leq T + Z. \quad (3)$$

The first constraints assigns each job to some machine, and is a natural one. But the second constraint is mysterious due to the variables Y_i and Z : to demystify this, let us write its dual program $D_c(T, \mathbf{p})$:

$$\begin{aligned} \max \sum_{j \in J} \alpha_j - T \sum_{i \in M} \beta_i & \quad (D_c(T, \mathbf{p})) \\ \alpha_j - \beta_i p_{ij} \leq 0 & \quad \forall (i, j) \in E(T, \mathbf{p}) \\ \sum_{i \in M} \beta_i \leq 1 & \quad (4) \\ \beta_i \geq 1/cn & \quad \forall i \in M \quad (5) \\ \alpha, \beta \geq 0. & \end{aligned}$$

Note that the dual program has variables α_j for each job $j \in J$, and variables β_i for each machine $i \in M$; the latter should be thought of as machine “weights”. Observe that this LP is very similar to the dual for the natural relaxation, where constraints (4) and (5) are the new constraints added to control the range of the β_i values. Moreover, complementary slackness says that $x_{ij}(\alpha_j - \beta_i p_{ij}) = 0$.

Proposition 3.1. If $T \geq OPT(\mathbf{p})$, then both the primal and dual linear programs above are feasible.

Proof. Indeed, consider the assignment $\mathbf{x}^*(\mathbf{p})$ that achieves the optimal makespan for input \mathbf{p} ; it is a feasible solution to the primal program. Moreover, the all-zero solution is feasible for the dual program. \square

3.1 Properties of the New Programs

The optimal solution for the new dual program $D_c(T, \mathbf{p})$ has a value no larger than that of the standard dual, since it is a maximization problem with more constraints. Of course, the added constraints could conceivably make the dual infeasible, but we now show that this is not possible.

Lemma 3.2 (LPs are Bounded). *For any $c > 1$ and any $T \geq OPT(\mathbf{p})$, the primal $P_c(T, \mathbf{p})$ is feasible, and any (fractional) solution $(\mathbf{x}, \mathbf{Y}, Z)$ to it has value at least $Z(1 - 1/c) - T/c \geq -T/c$. Moreover, its dual $D_c(T, \mathbf{p})$ is feasible and bounded as well.*

Proof of Lemma 3.2. The original primal $P(T, \mathbf{p})$ is feasible because of Proposition 3.1; moreover, any solution for the original primal LP is also feasible for the new primal $P_c(T, \mathbf{p})$ by just setting $Z = Y_i = 0$. To show the bound on the primal objective value, consider the constraint (3). The non-negativity of the sum $\sum p_{ij}x_{ij}$ implies that $Y_i \leq T + Z$. Consequently, the primal objective value is

$$Z - \frac{1}{cn} \sum_i Y_i \geq Z - \frac{n(T+Z)}{cn} = Z\left(1 - \frac{1}{c}\right) - \frac{T}{c} \geq -\frac{T}{c},$$

where we used the non-negativity of the Z variable in the final step. Since the primal $P_c(T, \mathbf{p})$ is feasible and bounded from below, we can use strong duality to infer that the dual $D_c(T, \mathbf{p})$ is also feasible and bounded, which completes the proof. \square

While the objective function for the primal $P_c(T, \mathbf{p})$ is not meaningful *per se*, we can also show that the optimal fractional solution has small makespan.

Lemma 3.3 (Fractional Makespan). *For $c > 1$ and target $T \geq OPT(\mathbf{p})$, an optimal fractional solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{Y}}, \tilde{Z})$ for the linear program $P_c(T, \mathbf{p})$ has (fractional) makespan*

$$\max_i \sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} \tilde{x}_{ij} \leq T \cdot \frac{c}{c-1}.$$

Proof. Since any solution to the classical primal is also a solution to the new primal (by setting $Z = Y_i = 0$), the optimal solution to the primal $P_c(T, \mathbf{p})$ has objective value at most zero. Moreover, Lemma 3.2 implies that the optimal solution to $P_c(T, \mathbf{p})$ has objective value at least $\tilde{Z}(1 - 1/c) - T/c$. Combining these facts,

$$\tilde{Z}\left(1 - \frac{1}{c}\right) \leq \frac{T}{c} \implies \tilde{Z} \leq \frac{T}{c-1}.$$

Using constraint (3) again, but this time ignoring the non-negative \tilde{Y}_i terms, we get that

$$\sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} \tilde{x}_{ij} \leq T + \tilde{Z} \leq T \left(1 + \frac{1}{c-1}\right)$$

for each machine $i \in M$, as claimed. \square

In other words, any optimal fractional solution $\tilde{\mathbf{x}}$ for $P_c(T, \mathbf{p})$ is also a solution to the following relaxed version of the original linear program $P(T, \mathbf{p})$:

$$\begin{aligned} \sum_{i:(i,j) \in E(T, \mathbf{p})} x_{ij} &\geq 1 && \forall \text{ jobs } j \in J && (Q_\lambda(T, \mathbf{p})) \\ \sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} x_{ij} &\leq \lambda T && \forall \text{ machines } i \in M && (6) \\ x &\geq 0, \end{aligned}$$

as long as we set $\lambda = \frac{c}{c-1}$. (Note that when the relaxation parameter $\lambda = 1$, then $Q_1(T, \mathbf{p}) = P(T, \mathbf{p})$, i.e., we get back the original primal linear program.)

3.2 Rounding Algorithm

Assume that $T \geq OPT(\mathbf{p})$. In this case, the rounding algorithms of Lenstra et al. (1990) and Shmoys and Tardos (1993) take a fractional solution $\tilde{\mathbf{x}} \geq 0$ satisfying the relaxed LP $Q_\lambda(T, \mathbf{p})$ and output an integer assignment $\mathbf{x}^\dagger \in \{0, 1\}^{n \times m}$ with the following properties:

1. $\sum_i x_{ij}^\dagger = 1$ so that each job is assigned to exactly one machine,
2. each job j is assigned to some machine i to which it was originally fractionally assigned in $\tilde{\mathbf{x}}$ (i.e., $x_{ij}^\dagger = 1 \implies \tilde{x}_{ij} > 0$), and moreover
3. \mathbf{x}^\dagger is also a solution to $Q_{\lambda'}(T, \mathbf{p})$ but now with $\lambda' = \lambda + 1$.

In other words, this map satisfies

$$MS(\mathbf{p}, \mathbf{x}^\dagger) \leq \max_i \sum_{j:(i,j) \in E(T, \mathbf{p})} p_{ij} x_{ij}^\dagger \leq (\lambda + 1) \cdot T = (2 + 1/(c-1)) \cdot T. \quad (7)$$

This analysis can be tightened a bit to make the additive loss depend on the largest processing time of any job in the optimal solution, but we stick to the analytical form (7) for simplicity.

4 The Mechanism and the Predictions

Our mechanism takes the following predictions: (i) a predicted threshold \widehat{T} , (ii) predicted dual variables $\widehat{\beta}_i \in [1/cn, 1]$ for each machine $i \in M$, and (iii) a predicted assignment $\widehat{\varphi} : J \rightarrow M$. In the ideal solution, the threshold \widehat{T} should equal $\text{OPT}(\mathbf{p})$, the variables $\widehat{\beta}$ should be an optimal solution to the modified dual program $D_c(\widehat{T}, \mathbf{p})$, and the predicted assignment $\widehat{\varphi}$ should be the assignment \mathbf{x}^\dagger arising from optimally solving the modified primal $P_c(\widehat{T}, \mathbf{p})$ and then rounding it as in Section 3.2. The predicted assignment $\widehat{\varphi}$ corresponds to a vector $\widehat{\mathbf{x}} \in \{0, 1\}^{n \times m}$, and henceforth we will use the two objects interchangeably. Finally, observe that if we are instead given a prediction $\widehat{\mathbf{p}}$ of the processing times, we can compute the values above—as described in Section 3—and use them instead.

4.1 The Mechanism

To begin, the DUAL-PREDICTOR mechanism checks whether the predicted dual variables satisfy $\sum_i \widehat{\beta}_i \leq 1$ and $\min_i \widehat{\beta}_i \geq 1/cn$. If not, it rejects the predictions and just runs the greedy algorithm, which is n -robust and truthful Nisan and Ronen (2001). If the $\widehat{\beta}$ values are in the correct range, the mechanism considers the reported processing times $\mathbf{p} \in \mathbb{R}^{n \times m}$ along with the predictions and does the following:

Mechanism 1: DUAL-PREDICTOR

Input: Reported values p_{ij} for all $(i, j) \in M \times J$, predictions $\widehat{\beta} \in \mathbb{R}_+^n$, map $\widehat{\varphi} : J \rightarrow M$, and $\widehat{T} \in \mathbb{R}_+$.

Output: Assignment $\mathbf{x} \in \{0, 1\}^{n \times m}$.

for each job j do

Let $\text{small}(j) := \{i \mid p_{ij} \leq \widehat{T}\}$.

if $\text{small}(j) = \emptyset$ **then**

| $\varphi(j) \leftarrow \arg \min_i p_{ij}$.

else

| $\varphi(j) \leftarrow \arg \min_i \{\widehat{\beta}_i p_{ij} \mid i \in \text{small}(j)\}$. // breaking ties in favor of $\widehat{\varphi}(j)$.

return the vector \mathbf{x} corresponding to this allocation φ .

In words, the DUAL-PREDICTOR mechanism builds a set of machines $\text{small}(j) := \{j \mid p_{ij} \leq \widehat{T}\}$ on which the job is “small” compared to our prediction for the optimal makespan, based on the reported sizes \mathbf{p} and the prediction \widehat{T} for the optimal makespan. If this set is empty (a sure indicator of the predictions being incorrect), it assigns the job to the machine on which it has the smallest reported size. Else it assigns it to a machine in $\text{small}(j)$ that minimizes $\widehat{\beta}_i p_{ij}$, breaking ties in favor of the predicted machine $\widehat{\varphi}(j)$.

A technical observation: our mechanism uses the reported p_{ij} values to define the set $\text{small}(j)$ (which can be thought of as modifying the biases β_i). This is a departure from previous algorithms, which define the biases r_{ij} based on the predictions and the reported processing times.

5 The Analysis

We now analyze the quality of the assignment φ returned by the DUAL-PREDICTOR mechanism: our proof proceeds in three natural steps.

We first show $(1 + c)n$ -robustness of the mechanism by proving that no matter what the predictions are, the makespan of the assignment produced by the mechanism is at most $(1 + c)n \cdot \text{OPT}(\mathbf{p})$.

Next, we show $(2 + 1/(c-1))$ -consistency. We say that when the predictions are correct (i.e., when $\widehat{T} \approx \text{OPT}(\mathbf{p})$, and the $\widehat{\beta}_i$ values correspond to an optimal solution to the dual $D_c(\widehat{T}, \mathbf{p})$, and also the predicted assignment $\widehat{\varphi}$ corresponds to the integer allocation \mathbf{x}^\dagger obtained taking an optimal solution $\tilde{\mathbf{x}}$ to the primal $P_c(\widehat{T}, \mathbf{p})$ and rounding it as in Section 3.2), then our algorithm achieves a makespan of at most $(2 + 1/(c-1)) \cdot \widehat{T}$; i.e., a nearly optimal solution.

Finally we show that the mechanism is strategyproof, by showing that the mechanism is monotone.

5.1 Robustness

For robustness, we show that no matter what the predictions are, the mechanism's assignment \mathbf{x} has makespan $\text{MS}(\mathbf{p}, \mathbf{x})$ that is at most $O(n)$ times the optimal makespan. Let \mathbf{x}^* be the optimal assignment, so that $\text{OPT}(\mathbf{p}) = \text{MS}(\mathbf{p}, \mathbf{x}^*)$.

Theorem 5.1. *Given any predictions, $\text{MS}(\mathbf{p}, \mathbf{x}) \leq (1 + c)n \cdot \text{MS}(\mathbf{p}, \mathbf{x}^*)$.*

Proof. Let φ and φ^* be maps from J to M that correspond to the assignments \mathbf{x} and \mathbf{x}^* in the natural way. Define $r_j := \max(1, cn \cdot \widehat{\beta}_{\varphi^*(j)})$. To begin, we claim that for any job j ,

$$p_{\varphi(j),j} \leq r_j \cdot p_{\varphi^*(j),j}. \quad (8)$$

Indeed, consider the various cases:

1. If $\text{small}(j) = \emptyset$, then $p_{\varphi(j),j} \leq p_{i,j}$ for all machines i , and hence also for machine $\varphi^*(j)$.
2. Else if $|\text{small}(j)| \geq 1$, we consider two subcases.
 - (a) If $\varphi^*(j) \notin \text{small}(j)$, then $\widehat{T} < p_{\varphi^*(j),j}$. Moreover, we assign job j to some job in $\text{small}(j)$, and therefore $p_{\varphi(j),j} \leq \widehat{T}$. Chaining the two implies $p_{\varphi(j),j} < p_{\varphi^*(j),j}$.
 - (b) Else both $\varphi(j)$ and $\varphi^*(j)$ belong to $\text{small}(j)$. The mechanism's selection criterion ensures that $\widehat{\beta}_{\varphi(j)} p_{\varphi(j),j} \leq \widehat{\beta}_{\varphi^*(j)} p_{\varphi^*(j),j}$. Since the $\widehat{\beta}$ values range between $1/cn$ and 1, the claim immediately follows.

This proves the claim (8). Now consider any machine i , and let $J_i := \{j \mid \varphi(j) = i\}$ be the jobs assigned to machine i by the mechanism. The load of machine i is

$$\begin{aligned} \sum_{j \in J_i} p_{i,j} &\stackrel{(8)}{\leq} \sum_{j \in J_i} (1 + cn \cdot \widehat{\beta}_{\varphi^*(j)}) \cdot p_{\varphi^*(j),j} = \sum_{i' \in M} (1 + cn \cdot \widehat{\beta}_{i'}) \cdot \sum_{j \in J_i: \varphi^*(j)=i'} p_{i',j} \\ &\leq \sum_{i' \in M} (1 + cn \cdot \widehat{\beta}_{i'}) \cdot \text{MS}(\mathbf{p}, \varphi^*) = (n + cn) \cdot \text{MS}(\mathbf{p}, \mathbf{x}^*), \end{aligned}$$

where we use $\sum_i \widehat{\beta}_i \leq 1$ in the final step. \square

5.2 Consistency

We now show the consistency guarantees: when the predictions are correct, the makespan of the mechanism's assignment is at most $(2 + \frac{1}{c-1})$ times the optimal makespan. We consider the following slightly relaxed definition:

Definition 5.2 (γ -Correct Predictions). For $\gamma \geq 1$, the predictions $(\widehat{T}, \widehat{\beta}, \widehat{\varphi})$ are γ -correct for processing times \mathbf{p} if the following conditions hold:

- $\widehat{T} \in [\text{OPT}(\mathbf{p}), \gamma \text{OPT}(\mathbf{p})]$,
- $\widehat{\beta} = \widetilde{\beta}$ for an optimal solution $(\widetilde{\alpha}, \widetilde{\beta})$ to the dual program $D_c(\widehat{T}, \mathbf{p})$, and
- the assignment $\widehat{\mathbf{x}}$ corresponding to the prediction $\widehat{\varphi}$ is the assignment \mathbf{x}^\dagger obtained by rounding an optimal solution $\widetilde{\mathbf{x}}$ to the primal $P_c(\widehat{T}, \mathbf{p})$, as in Section 3.2.

When predictions are 1-correct, we refer to them as *correct* predictions. We show that γ -correct predictions imply a good makespan.

Lemma 5.3 (Consistency). *If the predictions $(\widehat{T}, \widehat{\beta}, \widehat{\varphi})$ are γ -correct for processing times \mathbf{p} , then*

$$\text{MS}(\mathbf{p}, \mathbf{x}) \leq (2 + 1/(c-1)) \cdot \widehat{T} \leq (2 + 1/(c-1)) \cdot \gamma \text{OPT}(\mathbf{p}).$$

Proof. Observe that the set $\text{small}(j)$ is exactly the set of permissible edges from $E(\widehat{T}, \mathbf{p}) := \{(i, j) \mid p_{i,j} \leq \widehat{T}\}$ which are incident to job j . Since $\widehat{T} \geq \text{OPT}(\mathbf{p})$, the set $\text{small}(j)$ is non-empty since it contains the machine that j is assigned to in the optimal integer solution \mathbf{x}^* . The mechanism now

assigns job j to the machine $\varphi(j) \in \text{small}(j)$ minimizing $\widehat{\beta}_i p_{ij} = \widetilde{\beta}_i p_{ij}$. We claim that this machine is in fact $\widehat{\varphi}(j)$. The claim implies that \mathbf{x} (the output of the mechanism) equals $\widehat{\mathbf{x}}$ (the assignment corresponding to the predicted allocation), which in turn equals \mathbf{x}^\dagger (the rounded solution). Hence (7) gives us $\text{MS}(\mathbf{p}, \mathbf{x}) = \text{MS}(\mathbf{p}, \widehat{\mathbf{x}}) \leq (2 + 1/(c-1)) \cdot \widehat{T}$, which completes the proof.

We now prove the claim that $\varphi(j) = \widehat{\varphi}(j)$. By the property of the rounding algorithm, we know that $\widehat{x}_{i,j} = 1$ only if $\widetilde{x}_{i,j} > 0$. Moreover, we have $\widetilde{x}_{i,j} > 0$ only for $(i, j) \in E(\widehat{T}, \mathbf{p})$ and hence $\widehat{\varphi}(j)$ belongs to $\text{small}(j)$. We now claim that

$$\widehat{\beta}_{\widehat{\varphi}(j),j} p_{\widehat{\varphi}(j),j} \leq \widetilde{\beta}_{ij} p_{ij} \quad (9)$$

for all $i \in \text{small}(j)$. Indeed, observe that $\widetilde{\alpha}_j \leq \widetilde{\beta}_i p_{ij}$ for all $(i, j) \in \text{small}(j)$ because of dual feasibility. Moreover, complementary slackness implies that $\widetilde{x}_{ij}(\widetilde{\alpha}_j - \widetilde{\beta}_i p_{ij}) = 0$ for all $i \in \text{small}(j)$. Since $\widetilde{x}_{\widehat{\varphi}(j),j} > 0$ we have that $\widetilde{\alpha}_j = \widetilde{\beta}_{\widehat{\varphi}(j),j} p_{\widehat{\varphi}(j),j}$. Moreover, using that $\widehat{\beta} = \widetilde{\beta}$, we get that $\widehat{\varphi}(j)$ is one of the minimizers of $\widehat{\beta}_i p_{ij}$, proving (9). Finally, our tie-breaking strategy ensures that the mechanism assigns the job j to $\widehat{\varphi}(j)$. \square

As an aside, note that while we used the fact that $\widehat{\varphi}$ is obtained from the rounding algorithm of Section 3.2, any $\widehat{\varphi}$ satisfying the properties above (in particular, that $\widetilde{x}_{\widehat{\varphi}(j),j} > 0$), and having low load, could be used instead.

5.3 Strategyproofness

To show that the mechanism above is strategyproof, recall the results from Section 2.1, and specifically Corollary 2.4 which shows that an item-size monotone mechanism is strategyproof.

Theorem 5.4 (Strategyproofness). *The Dual-Predictor mechanism is item-size monotone and hence strategyproof.*

Proof. Let the true processing times be given by \mathbf{p} , and consider any other report \mathbf{p}_i made by machine i . If $\mathbf{x} = \mathbf{x}(\mathbf{p}_i, \mathbf{p}_{-i})$ and $\mathbf{x}' = \mathbf{x}(\mathbf{p}'_i, \mathbf{p}_{-i})$ are the allocations under the correct reports and the misreport respectively, we want to show that

$$(x_{ij} - x'_{ij})(p_{ij} - p'_{ij}) \leq 0. \quad (10)$$

Let $\text{small}(j)$ and $\text{small}'(j)$ denote the sets corresponding to the runs of the mechanism when given reports $(\mathbf{p}_i, \mathbf{p}_{-i})$ and $(\mathbf{p}'_i, \mathbf{p}_{-i})$ respectively.

1. **Case I:** Suppose $x_{ij} = 1$, i.e., the job j is assigned to machine i under the correct reports. Since $x'_{ij} \in \{0, 1\}$, (10) follows immediately in case $p_{ij} \leq p'_{ij}$, so it suffices to show that $p'_{ij} < p_{ij} \implies x'_{ij} = 1$. We consider two cases based on the size of $\text{small}(j)$.

Suppose $\text{small}(j) = \emptyset$, then for machine i to be assigned job j , it must be that $i = \arg \min_{i'} p_{i',j}$. Since $p'_{ij} \leq \widehat{T}$, it must be the case that $|\text{small}'(j)| \leq 1$. If $\text{small}'(j) = \emptyset$, then $p'_{ij} < p_{ij}$ means we still have $i = \arg \min_{i'} p'_{i',j}$. Else $\text{small}'(j) = \{i\}$, and then the mechanism always allocates job j to the unique machine i in $\text{small}'(j)$. This means $x'_{ij} = 1$, proving the claim for the case when $\text{small}(j) = \emptyset$.

Else $|\text{small}(j)| \geq 1$. Since j is assigned to machine i , we infer that $i \in \text{small}(j)$ and $\widehat{\beta}_i p_{ij} \leq \widehat{\beta}_{i'} p_{i',j}$ for all $i' \in \text{small}(j)$. (This is vacuously true when $|\text{small}(j)| = 1$.) Since $p'_{ij} < p_{ij}$ and the other sizes $p_{i',j}$ remain unchanged for $i' \neq i$, we get $\text{small}'(j) = \text{small}(j)$. Moreover, since $\widehat{\beta}_i \geq 0$, the job j will continue to be assigned to machine i , and hence $x'_{ij} = 1$.

2. **Case II:** Suppose $x_{ij} = 0$. Since $x'_{ij} \in \{0, 1\}$, (10) follows immediately in case $p'_{ij} \leq p_{ij}$, so it suffices to show that $p'_{ij} > p_{ij} \implies x'_{ij} = 0$.

Let i_j be the machine that is assigned the job j in the assignment x : i.e., $x_{i_j,j} = 1$. If $\text{small}(j) = \emptyset$, then $p_{i_j,j} > p_{i',j}$ since it is not assigned the job j . Now, $p'_{i_j,j}$ is even higher, and hence $x'_{i_j,j} = 0$.

Else, $\text{small}(j) \neq \emptyset$, which means that $i_j \in \text{small}(j)$ with $\widehat{\beta}_{i_j} p_{i_j j} \leq \widehat{\beta}_{i'} p_{i' j}$ for all machines $i' \in \text{small}(j)$. In case $p'_{i_j} > \widehat{T}$, we have that $p'_{i_j} \notin \text{small}'(j)$ and therefore $x'_{i_j} = 0$. Else if $p_{i_j} < p'_{i_j} \leq \widehat{T}$, then we have that

$$\widehat{\beta}_{i_j} p_{i_j j} \leq \widehat{\beta}_i p_{i j} < \widehat{\beta}_i p'_{i j}$$

and therefore, $x'_{i_j} = 0$.

This case analysis implies that any machine i misreporting its processing time for job j leads to an allocation satisfying the item-wise monotonicity property (10). Now Corollary 2.4 implies that the mechanism is strategyproof. \square

6 Closing Remarks

This work extends the investigation of learning-augmented algorithms for optimization problems for which the pessimistic lower bounds can be circumvented using machine-learned predictions. We considered the unrelated machine scheduling problem in the strategic settings, where the agents (machines) can lie about the sizes of jobs to get an allocation with lower load. We show how to use only a linear number of predictions (essentially one value for each job and for each machine) to get constant consistency and optimal $O(n)$ robustness, improving on previous work that either achieved suboptimal robustness with linearly many predictions, or used quadratically many predictions for the optimal results. Our approach used a new LP relaxation with small integrality gap that also controlled the range of the dual values.

Several tantalizing directions remain open: can we improve the bounds even further? Since a typical case is when the number of machines is much smaller than the number of jobs, can we use only $O(n)$ predictions and get similar algorithmic guarantees? Can we predict these predictions much faster than predicting job sizes? Finally, what other algorithmic problems can use this “range-controlled dual” idea to get improvements?

References

- Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. 2024. Learning-augmented mechanism design: leveraging predictions for facility location. *Math. Oper. Res.* 49, 4 (2024), 2626–2651.
- Itai Ashlagi, Shahar Dobzinski, and Ron Lavi. 2012. Optimal lower bounds for anonymous scheduling mechanisms. *Math. Oper. Res.* 37, 2 (2012), 244–258. doi:10.1287/moor.1110.0534
- Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. 2023. Strategyproof scheduling with predictions. In *14th Innovations in Theoretical Computer Science Conference*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 251. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Art. No. 11, 22. doi:10.4230/lipics.itcs.2023.11
- Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. 2024. Mechanism Design with Predictions: An Annotated Reading List. *SIGecom Exch.* 21, 1 (Oct. 2024), 54–57. doi:10.1145/3699814.3699821
- Shai Ben-David, Nicolò Cesa-Bianchi, David Haussler, and Philip M. Long. 1995. Characterizations of Learnability for Classes of $\{0, \dots, n\}$ -Valued Functions. *J. Comput. Syst. Sci.* 50, 1 (1995), 74–86. doi:10.1006/JCSS.1995.1008
- George Christodoulou, Elias Koutsoupias, and Annamária Kovács. 2007. Mechanism design for fractional scheduling on unrelated machines. In *Automata, languages and programming*. Lecture Notes in Comput. Sci., Vol. 4596. Springer, Berlin, 40–52. doi:10.1007/978-3-540-73420-8_6
- George Christodoulou, Elias Koutsoupias, and Annamária Kovács. [2022] ©2022. On the Nisan-Ronen conjecture. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science—FOCS 2021*. IEEE Computer Soc., Los Alamitos, CA, 839–850.
- George Christodoulou, Elias Koutsoupias, and Annamária Kovács. [2023] ©2023. A proof of the Nisan-Ronen conjecture. In *STOC'23—Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. ACM, New York, 672–685. doi:10.1145/3564246.3585176
- George Christodoulou, Elias Koutsoupias, and Angelina Vidali. 2009. A lower bound for scheduling mechanisms. *Algorithmica* 55, 4 (2009), 729–740. doi:10.1007/s00453-008-9165-3

- George Christodoulou, Alkmini Sgouritsa, and Ioannis Vlachos. 2024. Mechanism design augmented with output advice. arXiv:2406.14165 [cs.GT] <https://arxiv.org/abs/2406.14165>
- Sami Davies, Benjamin Moseley, Sergei Vassilvitskii, and Yuyan Wang. 2023. Predictive flows for faster ford-fulkerson. In *International Conference on Machine Learning*. PMLR, 7231–7248.
- Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. 2021. Faster matchings via learned duals (*NIPS '21*). Curran Associates Inc., Red Hook, NY, USA, Article 795, 14 pages.
- Shahar Dobzinski and Ariel Shaulker. 2020. Improved Lower Bounds for Truthful Scheduling. arXiv:2007.04362 [cs.GT] <https://arxiv.org/abs/2007.04362>
- Yiannis Giannakopoulos, Alexander Hammerl, and Diogo Poças. 2021. A new lower bound for deterministic truthful scheduling. *Algorithmica* 83, 9 (2021), 2895–2913. doi:10.1007/s00453-021-00847-2
- Vasilis Gkatzelis, Kostas Kollias, Alkmini Sgouritsa, and Xizhi Tan. 2022. Improved Price of Anarchy via Predictions (*EC '22*). Association for Computing Machinery, New York, NY, USA, 529–557. doi:10.1145/3490486.3538296
- Elias Koutsoupias and Angelina Vidali. 2013. A lower bound of $1 + \phi$ for truthful scheduling mechanisms. *Algorithmica* 66, 1 (2013), 211–223. doi:10.1007/s00453-012-9634-6
- Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. 2020. Online Scheduling via Learned Weights. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, Shuchi Chawla (Ed.). SIAM, 1859–1877. doi:10.1137/1.9781611975994.114
- Thomas Lavastida, Benjamin Moseley, R. Ravi, and Chenyang Xu. 2021. Learnable and Instance-Robust Predictions for Online Matching, Flows and Load Balancing. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference) (LIPIcs, Vol. 204)*, Petra Mutzel, Rasmus Pagh, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 59:1–59:17. doi:10.4230/LIPICS.ESA.2021.59
- Jan Karel Lenstra, David B Shmoys, and Éva Tardos. 1990. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming* 46 (1990), 259–271.
- Shi Li and Jiayi Xian. 2021. Online Unrelated Machine Load Balancing with Predictions Revisited. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 6523–6532. <http://proceedings.mlr.press/v139/li21w.html>
- Thodoris Lykouris and Sergei Vassilvitskii. 2021. Competitive caching with machine learned advice. *J. ACM* 68, 4 (2021), Art. 24, 25. doi:10.1145/3447579
- Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. 2012. Online optimization with uncertain information. *ACM TALG* 8, 1 (2012), 2:1–2:29.
- Michael Mitzenmacher and Sergei Vassilvitskii. 2022. Algorithms with predictions. *Commun. ACM* 65, 7 (2022), 33–35. doi:10.1145/3528087
- Noam Nisan and Amir Ronen. 2001. Algorithmic mechanism design. Vol. 35. 166–196. doi:10.1006/game.1999.0790 Economics and artificial intelligence.
- Adam Polak and Maksym Zub. 2024. Learning-augmented maximum flow. *Inform. Process. Lett.* 186 (2024), 106487.
- Manish Purohit, Zoya Svitkina, and Ravi Kumar. 2018. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems* 31 (2018).
- Shinsaku Sakaue and Taihei Oki. 2022. Discrete-convex-analysis-based framework for warm-starting algorithms with predictions. *Advances in Neural Information Processing Systems* 35 (2022), 20988–21000.
- Michael Saks and Lan Yu. 2005. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings of the 6th ACM conference on Electronic commerce*. 286–293.
- David B Shmoys and Éva Tardos. 1993. An approximation algorithm for the generalized assignment problem. *Mathematical programming* 62, 1 (1993), 461–474.
- Chenyang Xu and Pinyan Lu. 2022. Mechanism Design with Predictions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 571–577. doi:10.24963/IJCAI.2022/81

A Learnability of Predictions

Let us show how we can learn predictions of the processing time, say in a PAC setting, where each instance \mathbf{p} is obtained by first drawing a context/feature a from a distribution \mathcal{D} over some set A , and getting the matrix $n \times m$ of processing times using an unknown map $f(a) \in P := \mathbb{R}_+^{n \times m}$. Our goal is to learn some map $\hat{f} : A \rightarrow P$ such that

$$\Pr_{a \sim \mathcal{D}} [\forall i, j, |\log \hat{f}_{ij}(a) - \log f_{ij}(a)| \leq \varepsilon] \geq 1 - \delta.$$

Then, at the time the mechanism is run on the true input, we suppose that the predictor sees the actual contexts/features a , and gives the prediction $\hat{f}(a)$ to the algorithm. The guarantee above ensures that $\Pr [D(f(a), \hat{f}(a)) \geq e^\varepsilon] \leq \delta$, where the error parameter $e^\varepsilon \approx (1 + \varepsilon)$ for small ε .

Let us sketch one way to solve this learning problem. First, we can focus on each coordinate f_{ij} separately and learn it with confidence $1 - \delta' := 1 - \delta/(mn)$; a trivial union bound then gives the claim. The goal now becomes that of learning a real-valued univariate function $g : A \rightarrow \mathbb{R}_+$, where $g(a) = \ln f_{ij}(a)$, to an additive accuracy of ε with high confidence $1 - \delta'$. At this point, we can use standard tools from machine learning theory to give bounds on the number of samples that suffice to learn g to an additive error of ε . Of course, one needs to make some assumptions on the complexity of the map \tilde{g} , which would depend on the applications at hand.

As a concrete example of these results, let us assume that the processing times lie in the range $[1, \dots, e^K]$, and define $\tilde{g}(a) := \lfloor g(a) \rfloor$, then the function \tilde{g} becomes $\{0, 1, \dots, K\}$ -valued. Furthermore, the gap due to this approximation $|\tilde{g} - g| \leq 1$; translating back via the definitions, the true processing time $f_{ij}(a)$ and its estimate $\hat{f}_{ij}(a) = \exp(\tilde{g}(a))$, would differ by at most a constant factor. Finally, given this $\{0, 1, \dots, K\}$ -valued function \tilde{g} , we can use the results of Ben-David et al. (1995) on learning such functions exactly with high confidence; this work reduces the learning of this function to showing the finiteness of VC-dimension of some related binary classification problems. In all these cases, the sample complexity behaves like $O((d_\Psi/\varepsilon^2) \log 1/\delta)$, where the parameter d_Ψ is some notion of the dimension of associated concept classes.

B Error tolerant Dual-Predictor Mechanism

It can be shown that the DUAL-PREDICTOR mechanism is somewhat brittle — even in the presence of minor errors, the mechanism achieves the robustness guarantee. In this section, we design an error tolerant mechanism that builds on the previous mechanism and achieves a constant approximation factor not only when the predictions are accurate, but also when the predictions are approximately accurate.

Specifically, our strategyproof mechanism ERROR-TOLERANT DUAL-PREDICTOR will take as input an error threshold $\eta \geq 1$ in addition to the predictions used in Mechanism 1: it will output an allocation with consistency guarantees if the error of the prediction is at most η , and it will always ensure the robustness guarantees. (Both the guarantees now depend on η .)

B.1 Approximately Correct Predictions

To begin, define the following “distortion” between two vectors \mathbf{p}, \mathbf{q} of processing times:

$$D(\mathbf{p}, \mathbf{q}) := \max_{i,j} \max \left\{ \frac{p_{ij}}{q_{ij}}, \frac{q_{ij}}{p_{ij}} \right\}. \quad (11)$$

Observe that distortion is symmetric, so that $D(\mathbf{p}, \mathbf{q}) = D(\mathbf{q}, \mathbf{p})$.

Definition B.1 ((γ, η) -Approximate Correctness). We now say that predictions $(\hat{T}, \hat{\beta}, \hat{\varphi})$ are (γ, η) -approximately correct with respect to the (true) processing times \mathbf{p} if there exist

- (a) processing times \mathbf{q} such that the distortion $D(\mathbf{q}, \mathbf{p}) \leq \eta$, and
- (b) predictions $(\eta \hat{T}, \hat{\beta}, \hat{\varphi})$ are γ -correct for processing times \mathbf{q} , where γ -correctness is defined as in Definition 5.2.

We say that the processing times \mathbf{q} and estimate \hat{T} certify (γ, η) -approximate correctness.

B.2 Error-Tolerant Mechanism

The error-tolerant mechanism ERROR-TOLERANT DUAL-PREDICTOR is given a parameter $\eta \geq 1$, along with predictions that are claimed to be (η, γ) -approximately correct, and it proceeds very similarly to the original DUAL-PREDICTOR mechanism: building a set of “small” machines for each job and then using a weighted greedy mechanism to assign to one of them (or to the machine with smallest reported processing time) if the set is empty.

Mechanism 2: Error-Tolerant Dual-Predictor

Input: Reported values p_{ij} for all $(i, j) \in M \times J$, predictions $\hat{\beta} \in \mathbb{R}_+^n$, map $\hat{\varphi} : J \rightarrow M$, and $\hat{T} \in \mathbb{R}_+$.

Output: Assignment $\mathbf{x} \in \{0, 1\}^{m \times n}$.

for each job j do

Let $\text{small}(j) := \{i \mid p_{ij} \leq \hat{T}\}$.

if $p_{\hat{\varphi}(j),j} \leq \eta^2 \hat{T}$ **then**

| $\text{small}(j) \leftarrow \text{small}(j) \cup \{\hat{\varphi}(j)\}$

if $\text{small}(j) = \emptyset$ **then**

| $\varphi(j) \leftarrow \arg \min_i p_{ij}$.

else

| **let** $\bar{\beta}_{\hat{\varphi}(j)} \leftarrow \hat{\beta}_{\hat{\varphi}(j)}$ **and** $\bar{\beta}_i \leftarrow \eta^2 \hat{\beta}_i$ for all $i \neq \hat{\varphi}(j)$

| $\varphi(j) \leftarrow \arg \min\{\bar{\beta}_i p_{ij} \mid i \in \text{small}(j)\}$. // breaking ties in favor of $\hat{\varphi}(j)$.

return the vector \mathbf{x} corresponding to this allocation φ .

There are two differences between the DUAL-PREDICTOR mechanism and this error-tolerant version: first, we now add in the predicted machine $\hat{\varphi}(j)$ to the set $\text{small}(j)$ even if the reported size $p_{\hat{\varphi}(j),j}$ is larger than the threshold \hat{T} , as long as it is not too large. Secondly, we modify the multipliers β to give the predicted machine an advantage, before choosing the machine for job j . Observe that for the setting of $\eta = 1$, the two mechanisms coincide.

B.3 Robustness

The proof of robustness is quite similar to that of the original mechanism.

Theorem B.2. *Given any predictions, $MS(\mathbf{p}, \mathbf{x}) \leq (1 + c)n \cdot \eta^2 \cdot MS(\mathbf{p}, \mathbf{x}^*)$.*

Proof. Let φ and φ^* be maps from J to M that correspond to the assignments \mathbf{x} and \mathbf{x}^* in the natural way. Define

$$r_j := \max(1, cn \cdot \hat{\beta}_{\varphi^*(j)}) \cdot \eta^2. \quad (12)$$

To begin, we claim that for any job j ,

$$p_{\varphi(j),j} \leq r_j \cdot p_{\varphi^*(j),j}. \quad (13)$$

Indeed, consider the various cases:

1. If $\text{small}(j) = \emptyset$, then $p_{\varphi(j),j} \leq p_{i,j}$ for all machines i , and hence also for machine $\varphi^*(j)$.
2. Else if $|\text{small}(j)| \geq 1$, we consider two subcases.
 - (a) If $\varphi^*(j) \notin \text{small}(j)$, then $\hat{T} < p_{\varphi^*(j),j}$. Moreover, we assign job j to some job in $\text{small}(j)$, and therefore $p_{\varphi(j),j} \leq \eta^2 \hat{T}$. Chaining the two implies $p_{\varphi(j),j} < \eta^2 \cdot p_{\varphi^*(j),j}$.
 - (b) Else both $\varphi(j)$ and $\varphi^*(j)$ belong to $\text{small}(j)$. The mechanism’s selection criterion ensures that

$$\bar{\beta}_{\varphi(j)} p_{\varphi(j),j} \leq \bar{\beta}_{\varphi^*(j)} p_{\varphi^*(j),j} \implies \hat{\beta}_{\varphi(j)} p_{\varphi(j),j} \leq \eta^2 \hat{\beta}_{\varphi^*(j)} p_{\varphi^*(j),j}.$$

Since the $\hat{\beta}$ values range between $1/cn$ and 1, the claim immediately follows.

This proves the claim (13). Finally a calculation similar to the proof from Theorem 5.1 shows that the load of machine i is at most $\eta^2 \cdot (n + cn) \cdot MS(\mathbf{p}, \mathbf{x}^*)$, completing the proof. \square

B.4 Consistency

Now suppose that the predictions $(\widehat{T}, \widehat{\beta}, \widehat{\varphi})$ are (γ, η) -approximately correct, and let \mathbf{q} be the processing times that certify this (γ, η) -approximate correctness. (Note that \mathbf{q} is not known to the mechanism.) Unwrapping the definition, we get that

1. $D(\mathbf{p}, \mathbf{q}) \leq \eta$, so that $\mathbf{p}/\eta \leq \mathbf{q} \leq \eta\mathbf{p}$,
2. $\text{OPT}(\mathbf{q}) \leq \eta\widehat{T} \leq \gamma\text{OPT}(\mathbf{q})$,
3. $\widehat{\beta} = \beta$ corresponds to an optimal dual solution $(\widetilde{\alpha}, \widetilde{\beta})$ to the dual $D_c(\eta\widehat{T}, \mathbf{q})$, and
4. $\widehat{\varphi}(j)$ corresponds to the solution \mathbf{x}^\dagger obtained by rounding an optimal fractional solution $\widetilde{\mathbf{x}}$ to the primal $P_c(\eta\widehat{T}, \mathbf{q})$.

Lemma B.3 (Consistency). *If the predictions are (γ, η) -approximately correct, then*

$$\text{MS}(\varphi, \mathbf{p}) \leq \widehat{\eta}^2 \cdot (2 + 1/(c-1)) \cdot \widehat{T} \leq \gamma\widehat{\eta}^2 \cdot (2 + 1/(c-1)) \cdot \text{OPT}(\mathbf{p}),$$

where $D(\mathbf{p}, \mathbf{q}) = \widehat{\eta} \leq \eta$.

Proof. We first argue that the set $\text{small}(j)$ is non-empty. Indeed, the machine $\widehat{\varphi}(j)$ lies in the support of some primal solution to the linear program $P_c(\eta\widehat{T}, \mathbf{q})$ and hence $q_{\widehat{\varphi}(j),j} \leq \eta\widehat{T}$. Since $p_{i,j} \leq \eta q_{i,j}$ for all (i, j) , we have $p_{\widehat{\varphi}(j),j} \leq \eta^2\widehat{T}$. By the definition of the mechanism, the machine $\widehat{\varphi}(j)$ is added to the set $\text{small}(j)$ if it satisfies this last inequality, thereby ensuring that $\text{small}(j)$ is non-empty. Hence, the mechanism assigns to the machine in $\text{small}(j)$ minimizing $\widetilde{\beta}_i p_{ij}$.

Now consider the machines in $\text{small}(j) = \{i \mid p_{ij} \leq \widehat{T}\} \cup \{\widehat{\varphi}(j)\}$. Since $\mathbf{p} \geq \widehat{\mathbf{p}}/\eta$, the former set is contained in $\{i \mid q_{ij} \leq \eta\widehat{T}\}$. Moreover, as argued in the previous paragraph, $\widehat{\varphi}(j)$ also belongs to this set. In other words, the pairs $\{(i, j) \mid i \in \text{small}(j)\}$ are contained in the permissible edges $E(\eta\widehat{T}, \mathbf{q})$.

Next, the feasibility for the dual program $D_c(\eta\widehat{T}, \mathbf{q})$ means that $\widetilde{\alpha}_j \leq \widetilde{\beta}_i q_{ij}$ for all $(i, j) \in E(\eta\widehat{T}, \mathbf{q})$; moreover, complementary slackness implies that $\widetilde{\alpha}_j = \widetilde{\beta}_i q_{ij}$ for machines $i \in \text{small}(j)$ with $\widetilde{x}_{ij} > 0$, i.e., in the support of the optimal fractional solution. Hence, the machine $\widehat{\varphi}(j)$ is a minimizer of $\widetilde{\beta}_i q_{i,j} = \widetilde{\beta}_i q_{i,j}$ over machines $i \in \text{small}(j)$. However, we do not have access to the processing times q_{ij} but only to p_{ij} . To fix this issue, we use (i) the fact that $q_{ij}/\eta \leq p_{ij} \leq \eta p_{ij}$ and (ii) the definition of $\widetilde{\beta}$ (as in in the mechanism) to infer that $\widehat{\varphi}(j)$ is the minimizer of $\widetilde{\beta}_i p_{ij}$ over machines $i \in \text{small}(j)$. This means $\mathbf{x} = \widehat{\mathbf{x}} = \mathbf{x}^\dagger$.

Now suppose that $D(\mathbf{p}, \mathbf{q}) = \widehat{\eta} \leq \eta$. Then the resulting makespan is

$$\begin{aligned} \max_i \sum_{j:\varphi(j)=i} p_{ij} &\leq \widehat{\eta} \cdot \max_i \sum_{j:\widehat{\varphi}(j)=i} q_{ij} = \widehat{\eta} \cdot \text{MS}(\widehat{\varphi}, \mathbf{q}) \\ &\leq \widehat{\eta} \cdot (2 + 1/(c-1)) \cdot \eta\widehat{T} \\ &\leq \widehat{\eta} \cdot (2 + 1/(c-1)) \cdot \gamma \text{OPT}(\mathbf{q}) \\ &\leq \widehat{\eta}^2 \cdot (2 + 1/(c-1)) \cdot \gamma \text{OPT}(\mathbf{p}), \end{aligned}$$

which completes the proof. \square

B.5 Strategyproofness

Theorem B.4 (Strategyproofness). *The error-tolerant Dual-Predictor mechanism is strategyproof.*

The proof is essentially identical to the proof of Theorem 5.4, so we only sketch it here.

Proof. (Sketch) For a job j , if a machine i was not in $\text{small}(j)$, the only way for machine i to become part of $\text{small}'(j)$ is to reduce its processing time for the job j . Likewise, the only way to be removed from $\text{small}(j)$ is to increase its processing time. Therefore, if machine i already had the job j ($x_{ij} = 1$), reducing its processing time ensures that it still receives the job, irrespective of whether $\text{small}(j)$ is empty or not.

On the other hand, if machine i did not receive the job, the case when $\text{small}(j) = \emptyset$ is trivial. If $\text{small}(j) \neq \emptyset$, increasing its processing time can only increase its scaled processing time until it eventually does not belong to $\text{small}'(j)$; this shows $x_{ij} = x'_{ij} = 0$ as required. \square

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: As the abstract and introduction claims, we designed a learning-augmented job scheduling mechanism for unrelated machines that requires only $\Theta(n + m)$ predictions and achieves nearly optimal consistency and robustness. Both consistency and robustness are proved thoroughly.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: It is clearly stated in "Our Results" section that we achieve marginally worse consistency than Balkanski et al. (2023) while using far fewer predictions—linear instead of quadratic.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper ensures that each theoretical result has clearly stated assumptions and provides a formal proof. The proofs are well organized and broken up into supporting lemmas wherever required to make them more readable. Due to space limitations, we provide an intuitive proof sketch in the main body and detail the complete proof in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not include experiments requiring code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The paper respects NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper presents theoretical results for a foundational theoretical problem in algorithmic game theory and has no foreseeable societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not contain any result or data that poses such a risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use any existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release any new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No LLMs were used for any of the results in this paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.