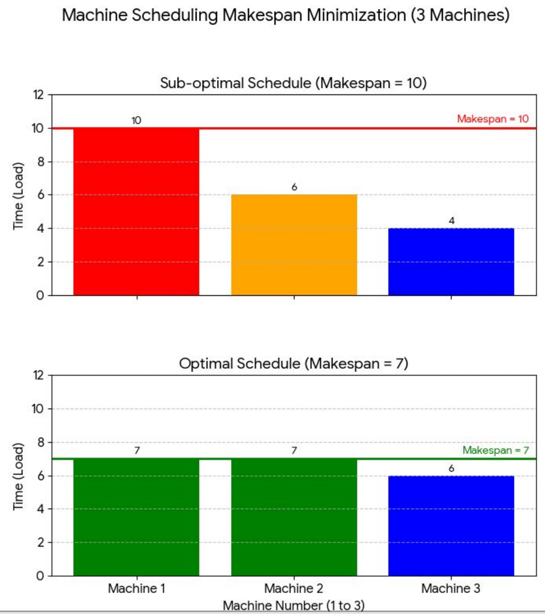
Parsimonious Predictions for Strategyproof Scheduling

Richard Cole, NYU
Anupam Gupta, NYU
Pranav Jangir, NYU

Intro to Scheduling

- Scheduling jobs on unrelated parallel machines with the goal of minimizing the makespan.



Goal : Minimize the **Makespan** (the total time required to complete all jobs, which is the load of the busiest machine).

Strategyproof Scheduling

- N machines and M jobs.
- $p(i, j)$ denotes the processing time of job j on machine i
- Machine i gets paid $\pi(i, j)$ for processing job j

  Machine's payoff for processing a job j is $\pi(i, j) - p(i, j)$
- Machines try to maximize their profit by potentially misreporting their processing times.

NYU

Mechanism design

- **Goal** : Design a payment scheme and scheduling algorithm such that every machine is incentivized to report it's real processing time.

- For the strategic case, the approximation factor is Θ(N).

NYU

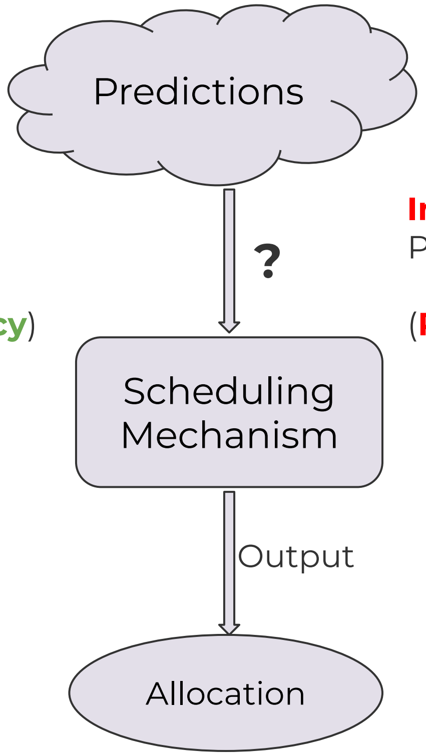
Strategyproof scheduling with predictions

Correct Predictions

(**Consistency**)

Incorrect Predictions

(**Robustness**)



O(1) approximate makespan if predictions are **Correct**

O(N) approximate makespan if predictions are **Incorrect**

NYU

VCG mechanism for strategyproof scheduling

- **Greedy allocation** : Assign every job to the machine with the least processing time.
  - Resulting allocation is **O(N)** approximate in the worst case scenario.
- **Weighted greedy allocation**: Assign every job to the machine with the least *scaled* processing time.
  - Resulting allocation is also **O(N)** approximate in the worst case scenario.
  - **Key Question**: Can we design these weights in a *clever way* ?

NYU

Role of Predictions

- Balkanski et al. designed a mechanism that takes all the processing times as predictions and achieves the best possible **O(1)** consistency and **O(N)** robustness.
- Used the predicted processing times to design weights for the weighted greedy allocation mechanism.
- Required **O(NM)** predictions.
- Can we use **fewer predictions** and achieve the same results?

NYU

Our Results

- A mechanism that achieves best of both worlds results with just **O(N+M)** predictions!

- Resulting mechanism is simpler to understand and implement.

- Introduce new tools and frameworks that can be applied to many problems.

Previous Work

Previous Work	Number of predictions used	Consistency	Robustness
Xu et al.	NM	O(1)	$O(N^3)$
Balkanski et al.	NM	O(1)	O(N)
Christodoulou et al.	M	O(1)	$O(N^2)$
This work	**N + M + 1**	**O(1)**	**O(N)**

NYU

Our weighted greedy allocation

- **Our Predictions**: predicted machine weights, optimal assignment and optimal makespan.

for *each job* j **do**
  Let $\mathsf{small}(j) := \{i \mid p_{ij} \leq \widehat{T}\}$.
  if $\mathsf{small}(j) = \varnothing$ **then**
    $\varphi(j) \leftarrow \arg\min_i p_{ij}$.
  else
    $\varphi(j) \leftarrow \arg\min_i \{\widehat{\beta}_i \, p_{ij} \mid i \in \mathsf{small}(j)\}$.   // breaking ties in favor of $\widehat{\varphi}(j)$.

Predictions from new LPs

$$\min \quad Z - {}^1/_{cn} \sum_i Y_i$$

$$\sum_{i:(i,j)\in E(T,\boldsymbol{p})} x_{ij} \geq 1 \qquad \forall \text{ jobs } j \in J$$

$$\sum_{j:(i,j)\in E(T,\boldsymbol{p})} p_{ij}\, x_{ij} - Z + Y_i \leq T \qquad \forall \text{ machines } i \in M$$

$$\boldsymbol{x}, \boldsymbol{Y}, Z \geq 0.$$

$$\max \sum_{j\in J} \alpha_j - T \sum_{i\in M} \beta_i$$

$$\alpha_j - \beta_i\, p_{ij} \leq 0 \qquad \forall (i,j) \in E(T,\boldsymbol{p})$$

$$\sum_{i\in M} \beta_i \leq 1$$

$$\beta_i \geq {}^1/_{cn} \qquad \forall i \in M$$

$$\alpha, \beta \geq 0.$$

General Framework

- Model the algorithmic version as a Linear Program.
- Modify the linear program to add required constraints.
- Use a weighted VCG mechanism with these predictions—variables from the dual.
- Example of applying this for the error-tolerant version in the paper.

Thanks for listening