Decentralization & Mechanism Design for Online Machine Scheduling

Birgit Heydenreich Rudolf Müller Marc Uetz

Maastricht University
Quantitative Economics

Supported by NWO grant
"Local Decisions in Decentralized Planning"

Introduction

Classical optimization

- input is completely known by a central planner
- **goal**: find a good solution

Three directions to depart from this

- data available over time: online optimization
- selfish agents instead of central planner: cost of anarchy
- agents with private data: mechanism design

This talk: all three directions at the same time

Introduction

Classical optimization

- input is completely known by a central planner
- **goal**: find a good solution

Three directions to depart from this

- data available over time: online optimization
- selfish agents instead of central planner: cost of anarchy
- agents with private data: mechanism design

This talk: all three directions at the same time

Introduction

Classical optimization

- input is completely known by a central planner
- **goal**: find a good solution

Three directions to depart from this

- data available over time: online optimization
- selfish agents instead of central planner: cost of anarchy
- agents with private data: mechanism design

This talk: all three directions at the same time

Introduction

Classical optimization

- input is completely known by a central planner
- **goal**: find a good solution

Three directions to depart from this

- data available over time: online optimization
- selfish agents instead of central planner: cost of anarchy
- agents with private data: mechanism design

This talk: all three directions at the same time

Introduction

Classical optimization

- input is completely known by a central planner
- **goal**: find a good solution

Three directions to depart from this

- data available over time: online optimization
- selfish agents instead of central planner: cost of anarchy
- agents with private data: mechanism design

This talk: all three directions at the same time

Parallel Machine Scheduling

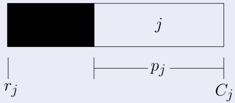
Machines

m parallel identical machines $M = \{1, \ldots, m\}$

Jobs

n jobs $J = \{1, \ldots, n\}$, non-preemptive

- release date $r_j \geq 0$
- processing time $p_j > 0$
- weight $w_j \geq 0$: indifference cost for waiting one time unit



Objective

minimize total weighted completion time $\sum w_j C_j$

Online Setting

Online Scheduling, $\min \sum w_j C_j$

- each job known upon release date only
- **goal:** competitive online algorithm ALG

$$\text{ALG} \leq \alpha \cdot \text{OFFLINE OPT}$$

- no online algorithm can be better than 1.309-competitive [Vestjens 1997]
- the best known algorithm is 2.6-competitive [Correa and Wagner 2005]

Strategic Setting

Jobs = Agents who have private information

- (r_j, p_j, w_j) "type" of a job
- the type is not publicly known

Jobs are selfish

- jobs: minimize own C_j
- valuation: $-w_jC_j$ for schedule
- central planner: design game to maximize social welfare: $\max \sum -w_jC_j \rightarrow \min \sum w_jC_j$
- jobs might pretend other type: $\tilde{r}_j \geq r_j$, $\tilde{p}_j \geq p_j$, $\tilde{w}_j$ arbitrary

Strategic Setting

Jobs = Agents who have private information

- (r_j, p_j, w_j) "type" of a job
- the type is not publicly known

Jobs are selfish

- jobs: minimize own C_j
- valuation: $-w_j C_j$ for schedule
- central planner: design game to maximize social welfare:
  $\max \sum -w_j C_j \rightarrow \min \sum w_j C_j$
- jobs might pretend other type: $\tilde{r}_j \geq r_j$, $\tilde{p}_j \geq p_j$, $\tilde{w}_j$ arbitrary

Mechanism Design

What is a Mechanism?

- actions: jobs report types (to whom and how?)
- (allocation) algorithm: jobs are scheduled in some way
- Ultimate goal: maximize total social welfare in equilibrium
- payment scheme: helps to induce (rational) jobs to report types truthfully
- quasi-linear utilities: $u_j = -w_j C_j - \pi_j$

Mechanism Design

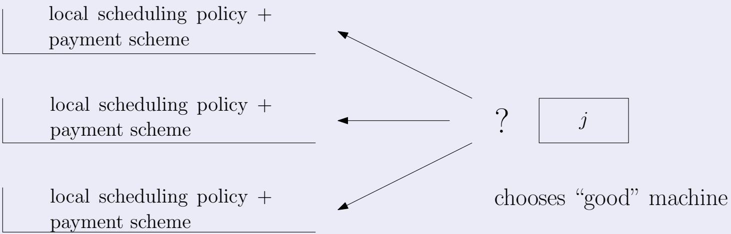
What is a Mechanism?

- actions: jobs report types (to whom and how?)
- (allocation) algorithm: jobs are scheduled in some way
- Ultimate goal: maximize total social welfare in equilibrium
- payment scheme: helps to induce (rational) jobs to report types truthfully
- quasi-linear utilities: $u_j = -w_j C_j - \pi_j$

Decentralized Mechanism

Decentralized decisions, limited communication

- no central coordination collecting data or distributing jobs over machines
- communication only between jobs and machines
- jobs select machines themselves



Summary - Model

Goals:

Mechanism for parallel machine scheduling that is

- online
- decentralized
- in equilibrium competitive for $\min \sum w_j C_j$ (social welfare)

Needed:

- local scheduling policy
- payment scheme

Summary - Model

Goals:

Mechanism for parallel machine scheduling that is

- online
- decentralized
- in equilibrium competitive for $\min \sum w_j C_j$ (social welfare)

Needed:

- local scheduling policy
- payment scheme

Equilibria Concepts

Definition: Dominant Strategy Equilibrium

each job has a strategy that maximizes its ex-post utility, no matter what the other jobs' types and strategies are

Tentative Utility

$\hat{u}_j = -w_j \hat{C}_j - \hat{\pi}_j$ (utility upon arrival $\tilde{r}_j$)

Definition: Myopic Best Response Equilibrium

each job has a strategy that maximizes its tentative utility, no matter what the other jobs' types and strategies are

Equilibria Concepts

Definition: Dominant Strategy Equilibrium

each job has a strategy that maximizes its ex-post utility, no matter what the other jobs' types and strategies are

Tentative Utility

$\hat{u}_j = -w_j \hat{C}_j - \hat{\pi}_j$ (utility upon arrival $\tilde{r}_j$)

Definition: Myopic Best Response Equilibrium

each job has a strategy that maximizes its tentative utility, no matter what the other jobs' types and strategies are

Equilibria Concepts

Definition: Dominant Strategy Equilibrium

each job has a strategy that maximizes its ex-post utility, no matter what the other jobs' types and strategies are

Tentative Utility

$\hat{u}_j = -w_j\hat{C}_j - \hat{\pi}_j$ (utility upon arrival $\tilde{r}_j$)

Definition: Myopic Best Response Equilibrium

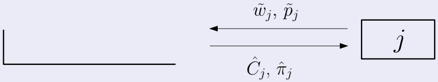
each job has a strategy that maximizes its tentative utility, no matter what the other jobs' types and strategies are

The Decentralized Local Greedy Mechanism

Local Scheduling Policy: "highest $\tilde{w}_j/\tilde{p}_j$ first"
Idea Payment Scheme: compensate displaced jobs for delay

Distribution:

- at time $\tilde{r}_j$:



- j chooses machine, tentative utility $\hat{u}_j$
- each job k displaced by j receives compensation $\tilde{w}_k\tilde{p}_j$

Decentralized Local Greedy Mechanism

By definition

- budget neutral
- online payment scheme

Decentralized Local Greedy Mechanism - Equilibria

Theorem 1

Truth telling (r_j, p_j, w_j) and choosing machine maximizing $\hat{u}_j$ $\forall j \in J$ is myopic best response equilibrium.

Theorem 2

regard restricted strategy space: $\forall j$: $\tilde{w}_j = w_j$.
Truth telling r_j and p_j and choosing machine maximizing $\hat{u}_j$ $\forall j \in J$ is dominant strategy equilibrium.

Decentralized Local Greedy Mechanism - Equilibria

Theorem 1

Truth telling (r_j, p_j, w_j) and choosing machine maximizing $\hat{u}_j$ $\forall j \in J$ is myopic best response equilibrium.

Theorem 2

regard restricted strategy space: $\forall j$: $\tilde{w}_j = w_j$.
Truth telling r_j and p_j and choosing machine maximizing $\hat{u}_j$ $\forall j \in J$ is dominant strategy equilibrium.

Decentralized Local Greedy Mechanism - Performance

Theorem 3

If jobs play the myopic best response equilibrium, that is report (r_j, p_j, w_j) and choosing machine maximizing $\hat{u}_j \Rightarrow$ Decentralized LocalGreedy Mechanism 3.281-competitive.

Discussion

Question 1

Can we make truth telling even a dominant strategy equilibrium if we require decentralization & online mechanism?

Theorem 4

There is no payment scheme for our mechanism that makes truth telling a dominant strategy equilibrium for parallel machines.

Theorem 5

For a single machine, there is one.

Question 2

Is strategic & decentralized setting "harder" than non-strategic? (Competitive ratio in non-strategic setting: 2.6 [Correa & Wagner])

Discussion

Question 1

Can we make truth telling even a dominant strategy equilibrium if we require decentralization & online mechanism?

Theorem 4

There is no payment scheme for our mechanism that makes truth telling a dominant strategy equilibrium for parallel machines.

Theorem 5

For a single machine, there is one.

Question 2

Is strategic & decentralized setting “harder” than non-strategic?
(Competitive ratio in non-strategic setting: 2.6 [Correa & Wagner])

Discussion

Question 1

Can we make truth telling even a dominant strategy equilibrium if we require decentralization & online mechanism?

Theorem 4

There is no payment scheme for our mechanism that makes truth telling a dominant strategy equilibrium for parallel machines.

Theorem 5

For a single machine, there is one.

Question 2

Is strategic & decentralized setting “harder” than non-strategic?
(Competitive ratio in non-strategic setting: 2.6 [Correa & Wagner])

Discussion

Question 1

Can we make truth telling even a dominant strategy equilibrium if we require decentralization & online mechanism?

Theorem 4

There is no payment scheme for our mechanism that makes truth telling a dominant strategy equilibrium for parallel machines.

Theorem 5

For a single machine, there is one.

Question 2

Is strategic & decentralized setting “harder” than non-strategic? (Competitive ratio in non-strategic setting: 2.6 [Correa & Wagner])

Key Lemma

report true w_j instead of $\tilde{w}_j \Rightarrow$
tentative utility $\hat{u}_j$ (at time $\tilde{r}_j$) can only increase

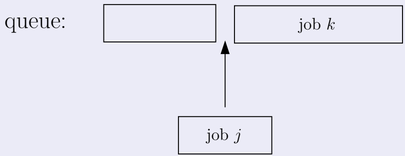
Remark

any false report $\tilde{w}_j \neq w_j$ may yield suboptimal utility
(recall: jobs only get to know $\hat{C}_j(i)$ and $\hat{\pi}_j(i)$)

Proof Idea for Key Lemma

Consider job j

Given arbitrary report on $\tilde{p}_j$:
Choosing $\tilde{w}_j$, job j might be queued anywhere in queue



- Assume job j is inserted in front of k:
- utility gain: $w_j \tilde{p}_k$ payment: $\tilde{w}_k \tilde{p}_j$
- Beneficial if: $w_j \tilde{p}_k > \tilde{w}_k \tilde{p}_j$, or $\dfrac{w_j}{\tilde{p}_j} > \dfrac{\tilde{w}_k}{\tilde{p}_k}$
- Thus true w_j gives optimal position in queue

Properties when truth telling w_j

- tentative = ex-post utility
- greedily choosing the best machine (=maximizing tentative utility $\hat{u}_j(i)$) maximizes ex-post utility

Decentralized Local Greedy Mechanism - Performance

Theorem 3

If jobs play the myopic best response equilibrium, that is report (r_j, p_j, w_j) and choosing machine maximizing $\hat{u}_j \Rightarrow$ DECENTRALIZED LOCALGREEDY Mechanism 3.281-competitive.

Proof Sketch

- $\sum w_j C_j = \sum_j -\hat{u}_j(i_j)$
- $-\hat{u}_j(i_j) \leq \frac{1}{m} \sum_{i=1}^{m} -\hat{u}_j(i)$
- (off line) lower bound from [Eastman et al. '64]

Can we adjust payments to get dominant strategy equilibrium?

Theorem 4

There exists no payment scheme that makes truth-telling in the Local Greedy Mechanism a dominant strategy equilibrium.

Proof idea

- From recent mechanism design literature (e.g., Bikhchandani, Chatterji, Lavi, Mu'alem, Nisan, Sen, 2006) it follows that such a payment scheme only exists if the following monotonicity holds:

$$\text{increase in reported } \tilde{w}_j \Rightarrow \text{decrease in } C_j$$

- Construct an example where higher $\tilde{w}_j$ leads to higher C_j

Equilibria

Definition: Dominant Strategy Equilibrium

strategies $s = (s_1, \ldots, s_n)$ are dominant strategy equilibrium $\Leftrightarrow$ $\forall j \in J$, $\forall$ type vectors t, $\forall$ strategy vectors $\tilde{s}_{-j}$ of the other jobs: playing s_j maximizes j's ex-post utility $u_j((s_j, \tilde{s}_{-j}), t)$.

$\hat{u}_j(s,t) = -w_j\hat{C}_j - \hat{\pi}_j$ (tentative utility at time $\tilde{r}_j$)

Definition: Myopic Best Response Equilibrium

strategies $s = (s_1, \ldots, s_n)$ are myopic best response equilibrium $\Leftrightarrow$ $\forall j \in J$, $\forall$ type vectors t, $\forall$ strategy vectors $\tilde{s}_{-j}$ of the other jobs: playing s_j maximizes j's tentative utility $\hat{u}_j((s_j, \tilde{s}_{-j}), t)$.

Related Work

Mechanism Design and Machine Scheduling

- Nisan & Ronen, 2001
- Archer & Tardos, 2004
- Kovacs, 2005
- Porter, 2004

Online Machine Scheduling

- Megow, Uetz, Vredeveld, 2006
- Correa, Wagner, 2005

Mechanism Design in Scheduling: Related Work

[Archer, Tardos, FOCS'01],[Nisan, Ronen, STOC'99]:

- agents=machines, offline related/unrelated machines
- private information: time needed to do the jobs
- (central) objective: $C_{\max}$

[Porter, EC'04]:

- agents=jobs, online single machine, preemptive
- private information: (r_j, p_j, d_j, w_j)
- (central) objective: $\max \sum_{j \text{early}} w_j$

Results: Truthful mechanisms, performance bounds, lower bounds
Note: All are direct (revelation) mechanisms

Mechanism design notation

Strategies: map types to actions

	strategy	
j: type	$\longmapsto$	actions
(r_j, p_j, w_j)		$\tilde{r}_j, \tilde{p}_j, \tilde{w}_j$ and $m \in M$

Job j's utility for a solution

utility	$=$	valuation	$-$	payment
$u_j = u_j(s,t)$	$=$	$-w_j C_j$	$-$	π_j

Assumption

Jobs are rational: utility maximizers when choosing strategy