Mechanism Design for Scheduling

Auction Protocols for Decentralized Scheduling

Wellman *et al.*

Elodie Fourquet

Electronic Market Design Presentation
School of Computer Science
University of Waterloo

November 22, 2004

Outline

1. Introduction
   The Factory Scheduling Problem
2. Formal Model
   Optimal Allocation & Equilibrium Solution Discussion
3. Ascending Auction (MM)
4. Combinatorial Auction (MM)
5. Generalized Vickery Auction (DRM)
6. Conclusions

Scheduling Problem Motivation

- Basic scheduling = hard problem
- Resource allocation problem

Scheduling Problem Motivation

- Basic scheduling = hard problem
- Resource allocation problem
- Essential to :
  1. computer science
  2. manufacturing & service industries

Scheduling Problem Motivation

- Basic scheduling = hard problem
- Resource allocation problem
- Essential to :
  1. computer science
  2. manufacturing & service industries
- In the Internet no time delivery guarantee

Scheduling Approaches

1. Distributed scheduling heuristics :
   First-come first-served, priority-first, shortest-job-first

Scheduling Approaches

1. Distributed scheduling heuristics :
   First-come first-served, priority-first, shortest-job-first
2. Market mechanism : price system
3. Direct revelation mechanism : GVA

An Application : Scheduling

- Goals
  1. Agents make effective decision
  2. Pareto optimal solution = resources are not wasted
  3. Reasonable communication, closure and computation

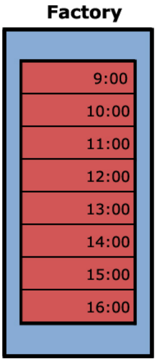
An Application : Scheduling

- Goals
  1. Agents make effective decision
  2. Pareto optimal solution = resources are not wasted
  3. Reasonable communication, closure and computation
- Problems
  1. Equilibrium solution
  2. Sometimes hard problem = NP-complete
     Discreteness & complementarity issues
  3. Combinatorial and Generalized Vickery Auction

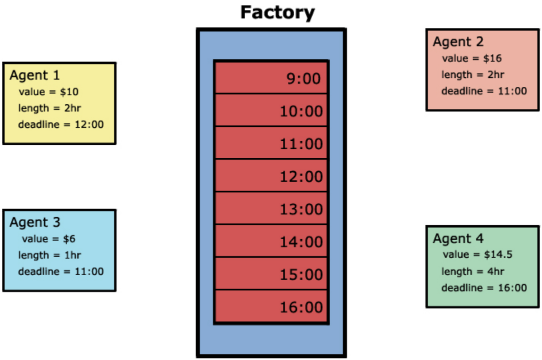
An Application : Scheduling

- Goals
  1. Agents make effective decision
  2. Pareto optimal solution = resources are not wasted
  3. Reasonable communication, closure and computation
- Problems
  1. Equilibrium solution
  2. Sometimes hard problem = NP-complete
     Discreteness & complementarity issues
  3. Combinatorial and Generalized Vickery Auction
- Practical application of course theory

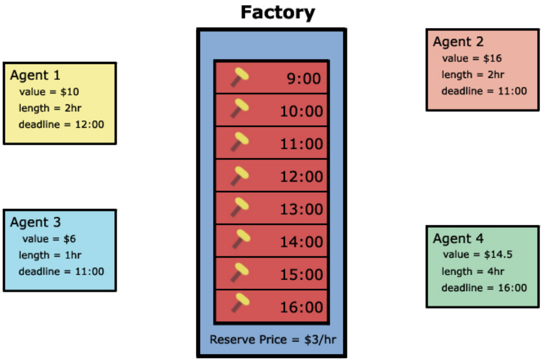
Factory Scheduling Example



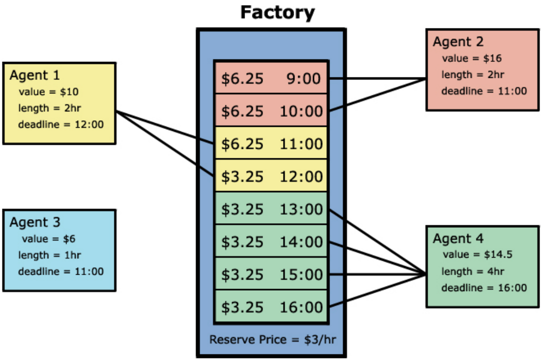
Agents' Jobs


Factory
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
Agent 1
value = $10
length = 2hr
deadline = 12:00
Agent 2
value = $16
length = 2hr
deadline = 11:00
Agent 3
value = $6
length = 1hr
deadline = 11:00
Agent 4
value = $14.5
length = 4hr
deadline = 16:00

Allocation with an Auction



Results



- Equilibrium solution
- Globally optimal allocation. Solution global value = $40.5

Decentralized vs Centralized Scheduling

Decentralized	Each agent is self-interested
Decentralized	Each agent knows only private info
Decentralized	Each agent communicates relevant private info
Decentralized	*Market Mechanisms (MM) : AA and CA*

Decentralized vs Centralized Scheduling

Decentralized	Each agent is self-interested
Decentralized	Each agent knows only private info
Decentralized	Each agent communicates relevant private info
Decentralized	*Market Mechanisms (MM) : AA and CA*

Centralized	Every agent info is known
Centralized	Decision-maker controls resources
Centralized	*Direct Revelation Mechanism (DRM): GVA*

Decentralized vs Centralized Scheduling

Decentralized	Each agent is self-interested
Decentralized	Each agent knows only private info
Decentralized	Each agent communicates relevant private info
Decentralized	*Market Mechanisms (MM) : AA and CA*

Centralized	Every agent info is known
Centralized	Decision-maker controls resources
Centralized	*Direct Revelation Mechanism (DRM): GVA*

Decentralized vs Centralized information & computation

General Discrete Resource Allocation Problem

Definition

- G, a set of n discrete goods
- A, a set of m agents
- $\perp$, the seller
- $p =< p_1, ..., p_n >$, set of prices

General Discrete Resource Allocation Problem

Definition

- G, a set of n discrete goods
- A, a set of m agents
- $\perp$, the seller
- $p =< p_1, ..., p_n >$, set of prices

Valuations

- Agent j has utility $v_j(X)$ for holding set of goods X, $X \subseteq G$
- Seller has utility q_i = reserve price, if good i is unallocated

Allocation Solution

A mapping, f, assigns discrete good to agents :

$$f : G \rightarrow A \cup \perp$$

Allocation Solution

A mapping, f, assigns discrete good to agents :

$$f : G \to A \cup \perp$$

	Allocated to agent j	Unallocated
Set of goods	$F_j \equiv \{i \| f(i) = j\}$	$F_\perp \equiv \{i \| f(i) = \perp\}$

Values Achievable

Maximum surplus value of agent j for holding set X at p

$$H_j(p) \equiv \max_{X \subseteq G}[v_j(X) - \sum_{i \in X} p_i]$$

Values Achievable

Maximum surplus value of agent j for holding set X at p

$$H_j(p) \equiv \max_{X \subseteq G} [v_j(X) - \sum_{i \in X} p_i]$$

Global value of solution f

Sum of agent values achieved + reserve value of goods not sold

$$v(f) \equiv \sum_{j=1}^{m} v_j(F_j) + \sum_{i \in F_\perp} q_i$$

Simple Scheduling

Definition

Each agent j has a job of :

- Length λ_j
- Deadlines $d_j^1 < \dots < d_j^{K_j}$
- Values $v_j^1 > \dots > v_j^{K_j}$

where $1 \leq K_j \leq n$, n total number of slots available

Several deadlines : higher values for earlier deadlines

Different Problems

1. Lengths of job :

Single-unit	$\lambda_j = 1$ for all j
Multiple-unit	$\lambda_j > 1$ for some j

2. Deadlines of job :

Fixed-deadline	$K_j = 1$ for all j
Variable-deadline	$K_j > 1$ for some j

Price Equilibrium

Definition

A solution f is in *equilibrium* at prices p iff :

1. All agents j get goods in allocation f that max his surplus at p

$$v_j(F_j) - \sum_{i \in F_j} p_i = H_j(p)$$

2. For all i, $p_i \geq q_i$
3. For all $i \in F_\perp$, $p_i = q_i$

Optimality of Equilibrium

Theorem

For the general discrete resource allocation problem, if there exists a p such that f is in equilibrium at p, then f is an optimal solution.

Optimality of Equilibrium

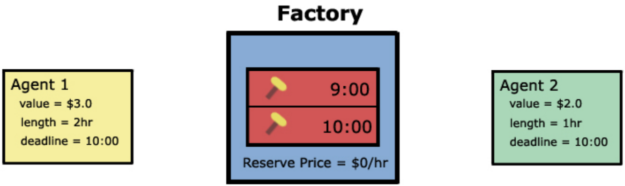
Theorem

For the general discrete resource allocation problem, if there exists a p such that f is in equilibrium at p, then f is an optimal solution.

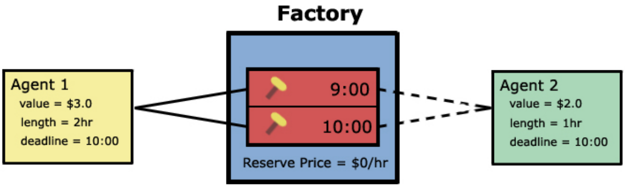
Proof (Main Idea).

Price forms a boundary between equilibrium and alternate solution. □

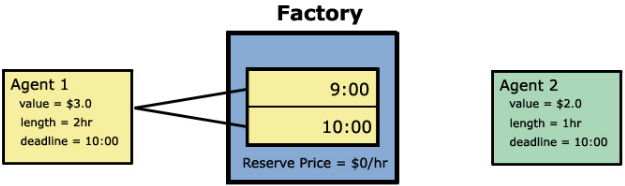
Agents' Jobs


Factory
Agent 1
value = $3.0
length = 2hr
deadline = 10:00
9:00
10:00
Reserve Price = $0/hr
Agent 2
value = $2.0
length = 1hr
deadline = 10:00

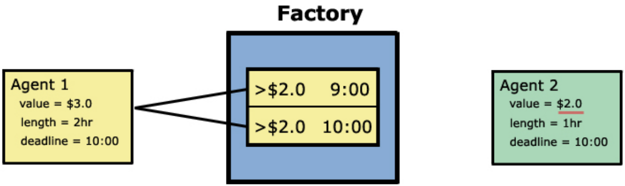
Agents' Interests



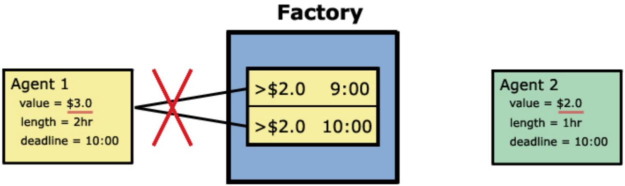
Optimal Solution



Price Equilibria Requirements



No Equilibrium Exists



Problem of complementarities in Agent1 preferences.

Equilibrium

- Single-unit scheduling problem always has at least one price equilibrium.
- But in general case, equilibrium may not exist.
- Single complementarity is sufficient to prevent a price equilibrium.

Market Mechansim Advantages

Considering decentralized scheduling :

- Markets are naturally decentralized
- Communication = exchange of bids & prices
- Mechanism can elicit info for Pareto & global optima
- Price is a common scale of value

Market Mechansim Advantages

Considering decentralized scheduling :

- Markets are naturally decentralized
- Communication = exchange of bids & prices
- Mechanism can elicit info for Pareto & global optima
- Price is a common scale of value
- Price system significantly simplifies resources allocation mechanism

Ascending Auction Protocol

Mechanism Bidding Rules

- *Bid price*, β_i = highest bid so far
- *Ask price*, $\alpha_i = \beta_i + \epsilon$ or q_i if undefined
- Agent must bid at least *ask price*

Agent Bidding Policies

Agent bids *ask prices* for the set of goods, maximizing his surplus. No anticipation of other agents' strategies.

Ascending Auction Problems

1. Protocol may not find an equilibrium solution.
   ▸ AA Example 1

2. Protocol can produce a solution arbitrary far from optimal.
   ▸ AA Example 2

3. Protocol restricted to single-unit length job,
   is still not guaranteed to reach equilibrium.
   ▸ AA Example 3

Incremental Auction Closing

- Sunk costs are considered
- Positive or negative effects on the solution
  ▸ AAIC Example 1
- No effect for:
  1. Single-unit problem, no sunk costs
  2. If allocation represents a price equilibrium
- Order of reopening matters

Combinatorial Auction Needs

- Ascending auction mostly works well for single-unit problem.
- Ascending auction cannot always find existing equilibria in multiple-unit problem.

Combinatorial Auction Needs

- Ascending auction mostly works well for single-unit problem.
- Ascending auction cannot always find existing equilibria in multiple-unit problem.
- Combinatorial auctions help complementary issues.
  But, computationally more complex.

Problem Allocation Reformulation

Definition

- G, a set of n discrete *basic goods*
- G', a expanded set of *market goods*
  $good(y, z)$, denotes "bundle of y slots no later than slot z"
- A, a set of m agents
- $\perp$, the seller
- P', set of prices $p(y, z)$ for all market goods in G'

Scheduling Computational Tractability

Order

- No need to consider all 2^n combinations
- $\theta(l \cdot n)$ market goods in G' and prices in P'
  where l is a bound on y, i.e. $y \leq l$ $(l \geq \max_{j \in A} \lambda_j)$
- Because additional structure (similar to Rothkopf at al. 1998)
  Agents will want some number of slots before some deadline
- Goal is to preserve tractability

Market Good Allocation Solution

- A mapping, ϕ, assigns market goods to agents :

$$\phi : G' \rightarrow A \cup \perp$$

Market Good Allocation Solution

- A mapping, ϕ, assigns market goods to agents :

$$\phi : G' \rightarrow A \cup \perp$$

- Set of market goods allocated to agent j :

$$\Phi_j \equiv \{i|\phi(i) = j\}$$

Market Good Allocation Solution

- A mapping, ϕ, assigns market goods to agents :

$$\phi : G' \to A \cup \perp$$

- Set of market goods allocated to agent j :

$$\Phi_j \equiv \{i | \phi(i) = j\}$$

- A market allocation ϕ is *consistent with* a solution f if f gives each agent what is promised by ϕ

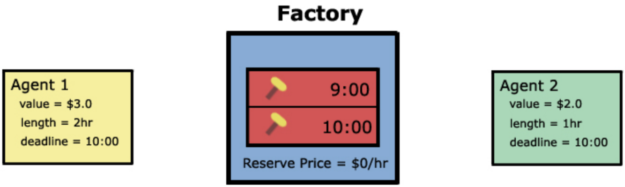
Combinatorial Price Equilibrium

Definition

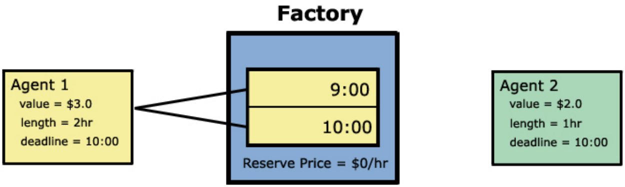
A solution ϕ is in *equilibrium* at prices p iff :

1. For all agent j, Φ_j maximizes j's guaranteed surplus at p
2. Market good price at least min consistent reserve price. For all (y, z), $p(y, z) \geq \min_B \sum_{i \in B} q_i$
3. There exists an implementing solution f, consistent with ϕ s.t.
   1. Allocated market good price $\geq$ sum of basic good prices comprising market good in f
   2. When market good could be satisfied by basic goods unallocated, reserve prices of those goods define an upper bound on its price

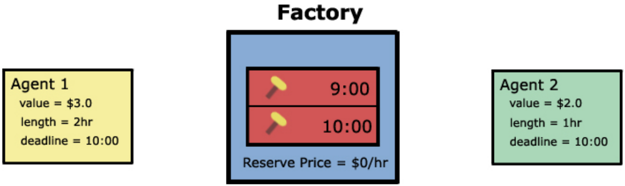
Agents' Jobs


Factory
Agent 1
value = $3.0
length = 2hr
deadline = 10:00
9:00
10:00
Reserve Price = $0/hr
Agent 2
value = $2.0
length = 1hr
deadline = 10:00

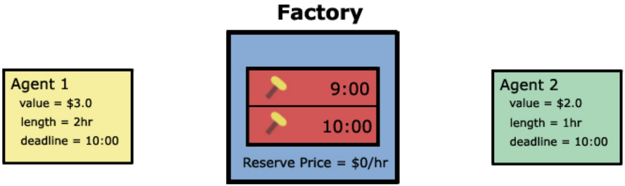
Optimal Solution



Combinatorial Auction

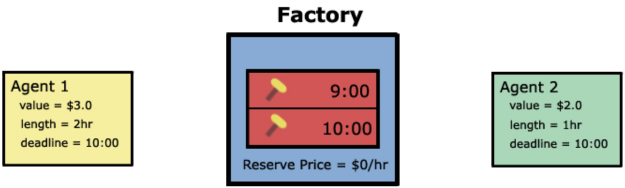


Combinatorial Auction



- Consider $I = 2$, $p(1, 9:00) = p(1, 10:00) = 2.1$ and $p(2, 10:00) = 2.9$

Combinatorial Auction



- Consider $I = 2$, $p(1, 9:00) = p(1, 10:00) = 2.1$ and $p(2, 10:00) = 2.9$
- Computed allocation $\Phi_1 = \{(2, 10:00)\}$, $\Phi_2 = \emptyset$ Satisfies combinatorial equilibrium conditions.

Optimal and Equilibrium

- Combinatorial equilibrium prices can support :
  1. Optimal solution
  2. But also non-optimal solution.

- Sub-optimality is not usefully bounded
  -even without reserve prices.

- Optimal solution supported by equilibria in original formulation are retained in the combinatorial one.
- Given monotone reserve prices, optimal solution can be supported with $\theta(l \cdot n)$ price system.

Generalized Vickery Auction

- Neither ascending nor combinatorial auction guarantee optimal solution to scheduling problem.

Generalized Vickery Auction

- Neither ascending nor combinatorial auction guarantee optimal solution to scheduling problem.
- GVA finds efficient schedules for all our scheduling problem.
- GVA is a direct revelation mechanism :
  - GVA is not a price system.
  - Rather GVA computes overall payments for agents' allocations.

VGA Protocol

Mechanism Bidding Rules

- Each agent j announces his alleged utility function $\tilde{v}_j$. Not constrained to be truthful.
- Auction knows the reserve values, q_i.

Allocation Rules and Optimality

After receiving bids, GVA returns :

1. Allocation solution f^*,
2. Payments to agents.

VGA Payments

- Payments to agent j:

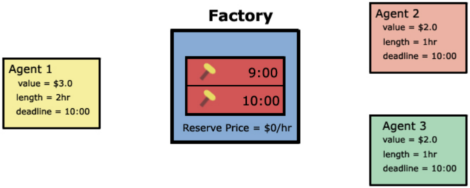
$$V_{-j} \equiv W_{-j}(f^*) - P_j(\tilde{v}_j)$$

where :

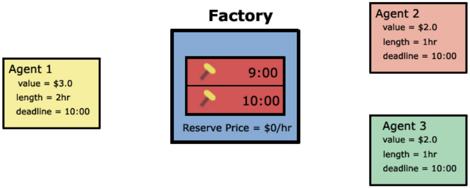
1. W_{-j} = agents' total reported value at f^*, excluding j
2. P_j = residual payment (function of other agent's reported valuations)

- Payments force truthful bidding as a dominant strategy. Optimal allocation is computed on truthful bids, therefore allocation is globally optimal.

VGA

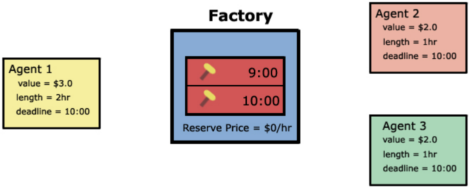

Factory
9:00
10:00
Reserve Price = $0/hr
Agent 1
value = $3.0
length = 2hr
deadline = 10:00
Agent 2
value = $2.0
length = 1hr
deadline = 10:00
Agent 3
value = $2.0
length = 1hr
deadline = 10:00

VGA



- Mechanism finds optimal solution : $f^*(9:00) = 2$ and $f^*(10:00) = 3$

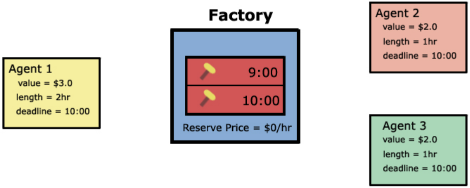
VGA



- Mechanism finds optimal solution : $f^*(9:00) = 2$ and $f^*(10:00) = 3$

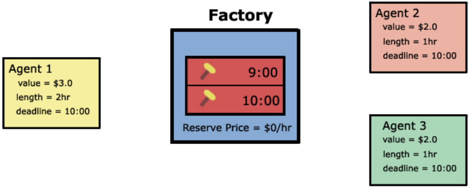
j	Agent 1	Agent 2	Agent3
W_{-j}	4	2	2
$v_j(F_j) + V_{-j}$	$0 + [4 - P_1]$	$2 + [2 - P_2]$	$2 + [2 - P_3]$

VGA



- For participation, received total value $v_j(F_j) + V_{-j} \geq 0$
  $P_j \leq 4$ for $j \in \{1, 2, 3\}$

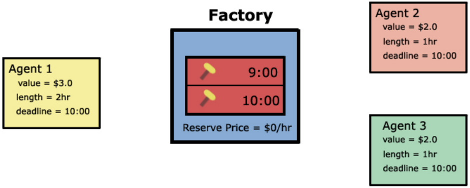
VGA



- For participation, received total value $v_j(F_j) + V_{-j} \geq 0$
  $P_j \leq 4$ for $j \in \{1, 2, 3\}$

j	Agent 1	Agent 2	Agent3
$v_j(F_j) + V_{-j}$	$0 + [4 - P_1]$	$2 + [2 - P_2]$	$2 + [2 - P_3]$
P_j	4 (pays 0)	3 (pays 1)	3 (pays 1)

VGA



- For participation, received total value $v_j(F_j) + V_{-j} \geq 0$
  $P_j \leq 4$ for $j \in \{1, 2, 3\}$

j	Agent 1	Agent 2	Agent3
$v_j(F_j) + V_{-j}$	$0 + [4 - P_1]$	$2 + [2 - P_2]$	$2 + [2 - P_3]$
P_j	4 (pays 0)	3 (pays 1)	3 (pays 1)

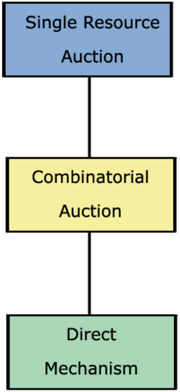
- Net revenue $2.0

Performance

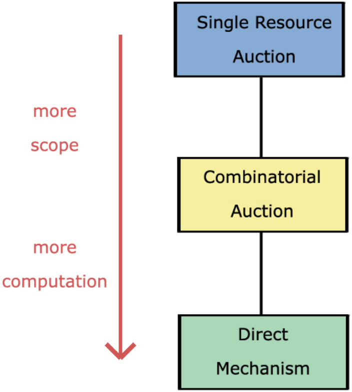
- Single-unit, fixed-deadline has optimal solution
  Greedy algorithm running in $\theta(m \lg m)$
- VGA mechanism must solve multiple optimization problems :
  1. One to determine optimal solution
  2. One for each agent j with his bids removed to find P_j

  Therefore VGA adds a factor of m to the computation
- Single-unit, fixed-deadline has optimal VGA solution
  With preference revelation needs $\theta(m^2 \lg m)$
- Multiple-unit scheduling problem is NP-complete

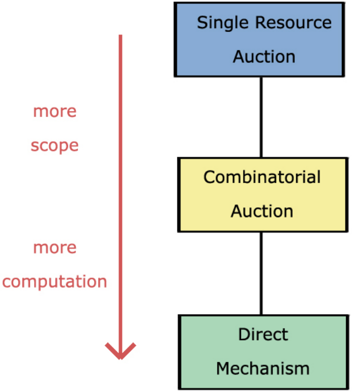
Scope and Computation Tradeoffs



Scope and Computation Tradeoffs



Scope and Computation Tradeoffs



But there exists more scheduling problems,
If we have time, for example....

Online Real-time Scheduling Problem

- Online scheduling of jobs on a single processor
  Online = not all jobs are known in advance
- Jobs are owned by seperate, self-interested agents
  1. Decide when to submit job after true release time
  2. Can inflate job's length
  3. Can declare arbitrary value and deadline for job
- Strategic agent can manipulate the system by annoucing false characteristics of job, if beneficial for its completion
- Sellers schedule jobs and determine amount to charge to buyers

Online Real-time Scheduling Goals

1. Schedule needs to be constructed in real-time
2. Maximizing sum of job's values completed on time
3. Online algorithm needs to compare well against the optimal offline one
4. Preemption of a running job by a newly arrived job is possible

Online Real-time Scheduling Direct Mechanism

- Input : job declared by each agent
- Output : schedule and payment to be made by each agent to mechanism
- Goal = incentive compatibility
  Agent's best interests :
  1. To submit job upon release
  2. To declare truthfully value, length and deadline of job
- Approximate solutions compare well with offline solutions

To Take Home

- Scheduling is important
- Many types of scheduling problem exist
- Most scheduling problems are hard,
  and most often NP-complete
- Price systems and auctions are a promising new approach for multiple scheduling problems
- Auction mechanisms encourage truth revelation about jobs
  Crucial for distributed scheduling

Questions ?

Challenges

Challenges

1. Message passing / closure / final schedule determination
   Protocol problem : asynchronous communication

Challenges

1. Message passing / closure / final schedule determination
   Protocol problem : asynchronous communication

2. Appropriate messages elicited
   Mechanism design problem : socially desirable outcome

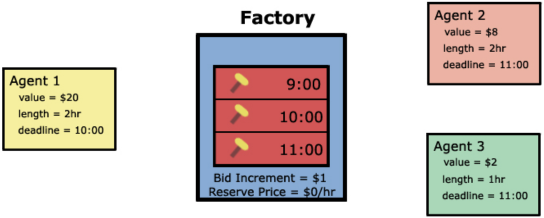
Combinatorial Price Equilibrium

Definition

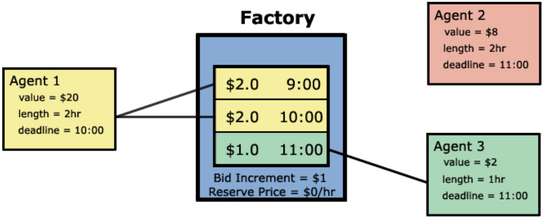
A solution ϕ is in *equilibrium* at prices p iff :

1. For all agent j, Φ_j maximizes j's guaranteed surplus at p
2. For all (y, z), $p(y, z) \geq \min_{\{B \subseteq G_z : |B| = y\}} \sum_{i \in B} q_i$
3. There exists an implementing solution f s.t.
    1. For all j, $\sum_{(y,z) \in \Phi_j} p(y, z) \geq \sum_{i \in F_j} q_i$
    2. For all "unallocated (y,z)", $p(y, z) \leq min_B \sum_{i \in B} q_i$

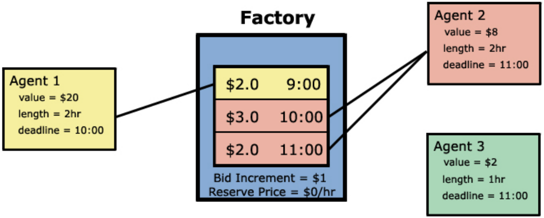
Agent jobs


Agent 1
value = $20
length = 2hr
deadline = 10:00
Factory
9:00
10:00
11:00
Bid Increment = $1
Reserve Price = $0/hr
Agent 2
value = $8
length = 2hr
deadline = 11:00
Agent 3
value = $2
length = 1hr
deadline = 11:00

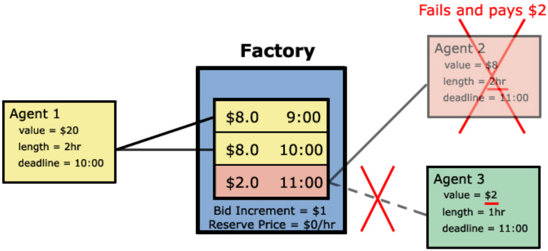
Bids


Factory
$2.0 9:00
$2.0 10:00
$1.0 11:00
Bid Increment = $1
Reserve Price = $0/hr
Agent 1
value = $20
length = 2hr
deadline = 10:00
Agent 2
value = $8
length = 2hr
deadline = 11:00
Agent 3
value = $2
length = 1hr
deadline = 11:00

Bids

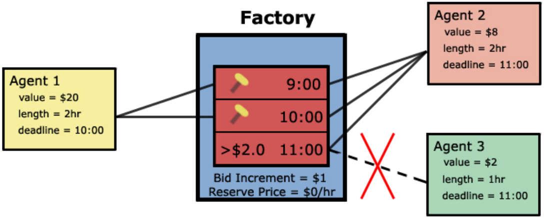

Factory
Agent 1
value = $20
length = 2hr
deadline = 10:00
$2.0 9:00
$3.0 10:00
$2.0 11:00
Bid Increment = $1
Reserve Price = $0/hr
Agent 2
value = $8
length = 2hr
deadline = 11:00
Agent 3
value = $2
length = 1hr
deadline = 11:00

Bids

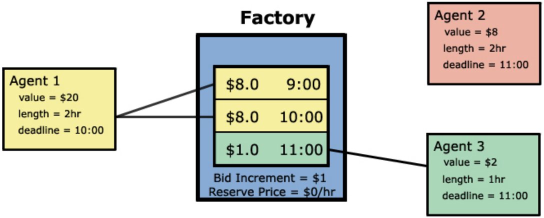


- Agent 2 wins slot 3 but cannot complete his job
- Agent 3 cannot get slot 3, $p_3 > 2$ blocked by Agent 2
- Not an optimal solution. Solution global value = $20.0

Problem


Factory
Agent 1
value = $20
length = 2hr
deadline = 10:00
9:00
10:00
>$2.0 11:00
Bid Increment = $1
Reserve Price = $0/hr
Agent 2
value = $8
length = 2hr
deadline = 11:00
Agent 3
value = $2
length = 1hr
deadline = 11:00

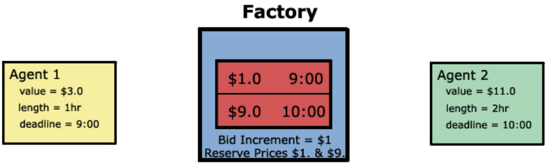
Equilibrium Solution



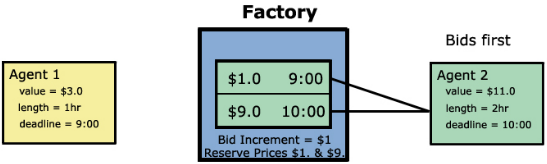
- Price equilibrium if Agent3 wins slot 3 at $p_3 \leq 2$
- Optimal solution. Solution global value $= \$22.0$

Return

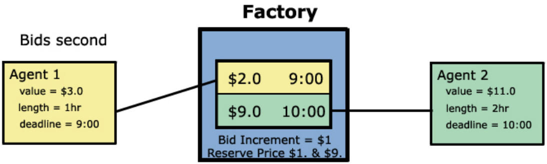
Agent jobs


Factory
Agent 1
value = $3.0
length = 1hr
deadline = 9:00
$1.0 9:00
$9.0 10:00
Bid Increment = $1
Reserve Prices $1. & $9.
Agent 2
value = $11.0
length = 2hr
deadline = 10:00

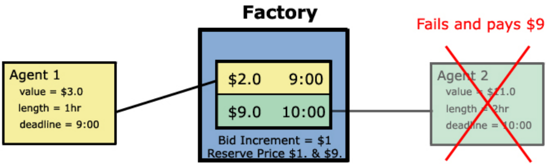
A2 Bids First



A1 Bids Second

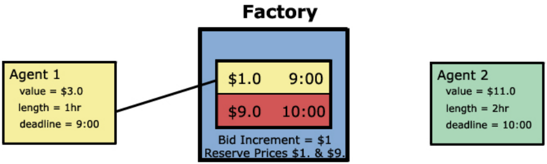


Allocation



- Agent 2 wins slot 2 but cannot complete his job
- Solution global value = $3.0

Optimal Solution



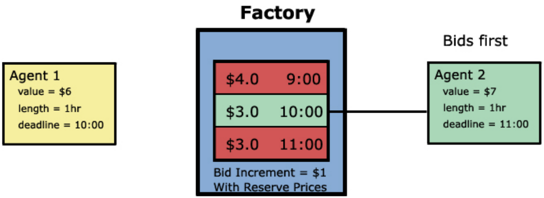
- Optimal solution (not equilibrium).
  Solution global value = $12.0
- Solution can be arbitrary far from optimal

Return

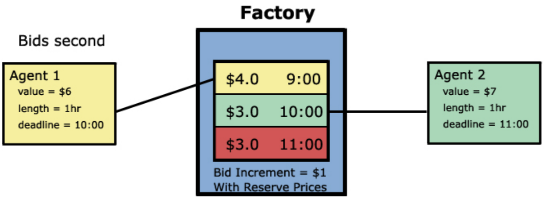
Agents' jobs


Factory
Agent 1
value = $6
length = 1hr
deadline = 10:00
$4.0 9:00
$3.0 10:00
$3.0 11:00
Bid Increment = $1
With Reserve Prices
Agent 2
value = $7
length = 1hr
deadline = 11:00

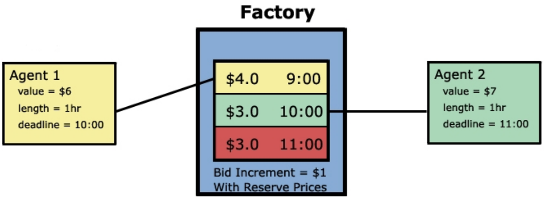
A2 Bids First



A1 Bids Second

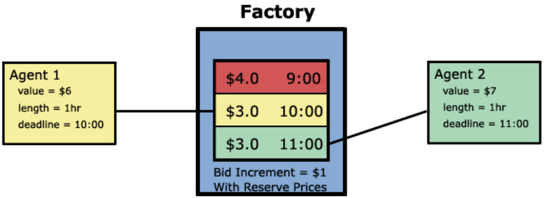


Allocation



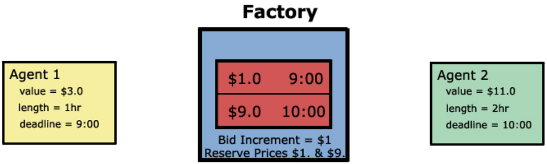
- But $p_2 = \$3 < p_1$ not an equilibrium
- Agent 1 would maximize his surplus by demanding p_2 at the final prices

Equilibrium Solution

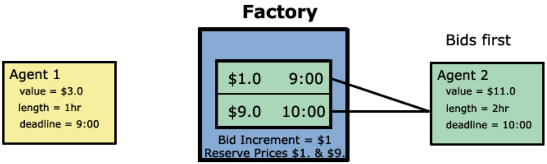


Return

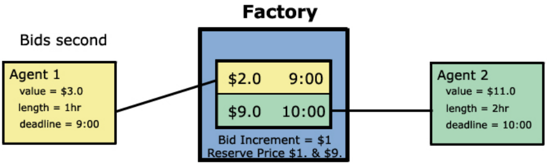
Agents' jobs


Factory
Agent 1
value = $3.0
length = 1hr
deadline = 9:00
$1.0 9:00
$9.0 10:00
Bid Increment = $1
Reserve Prices $1. & $9.
Agent 2
value = $11.0
length = 2hr
deadline = 10:00

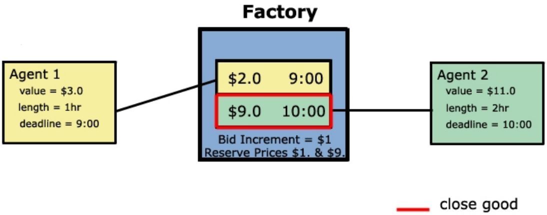
A2 Bids First



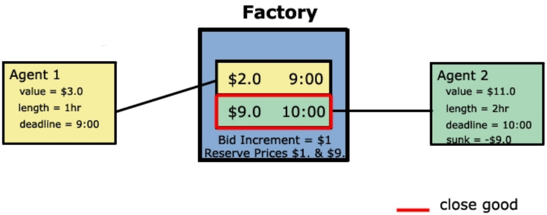
A1 Bids Second



Auction Closed for Slot 2

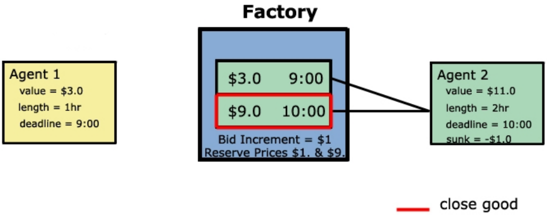


Agent 2 sunk cost



- Agent 2 treats his payment as sunk, and value slot 1 at $11

Allocation



- Agent 2 outbids Agent 1 for slot 1
- Solution global value = \$11
  (better >\$3 but not optimal <\$12)

Return