

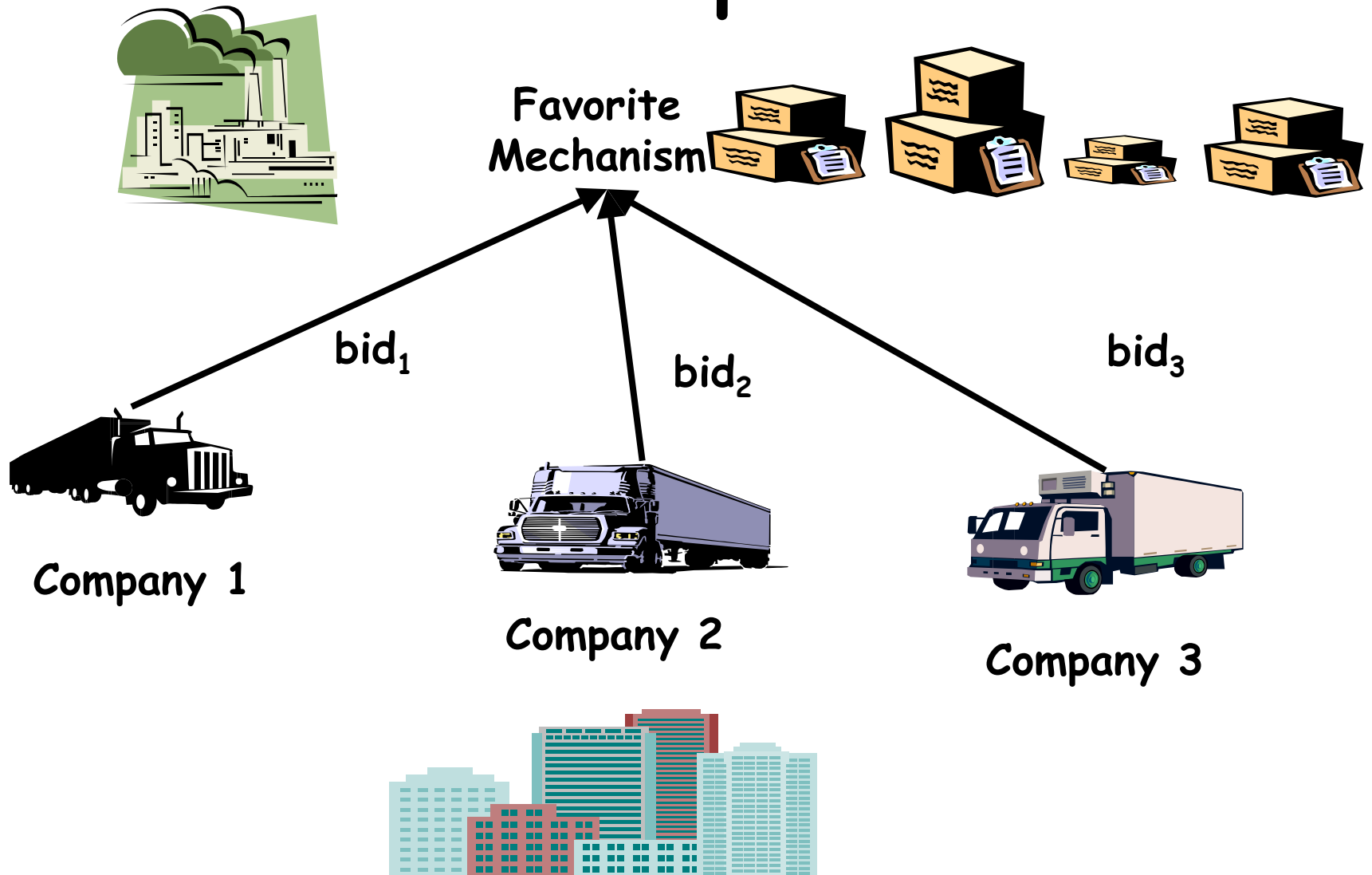
Mechanism Design and Computationally-Limited Agents

Kate Larson

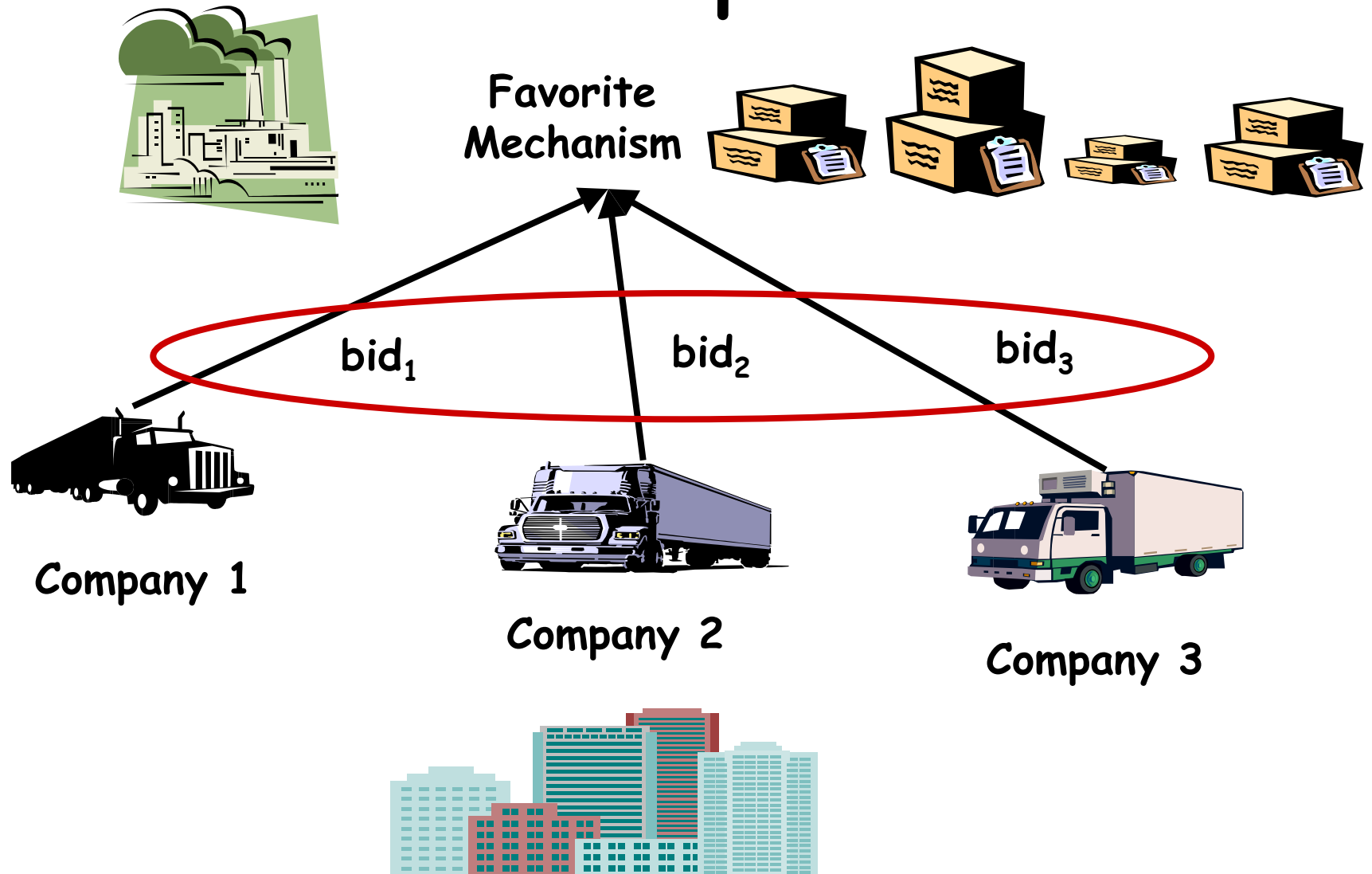
Computer Science Department

Carnegie Mellon University

Example



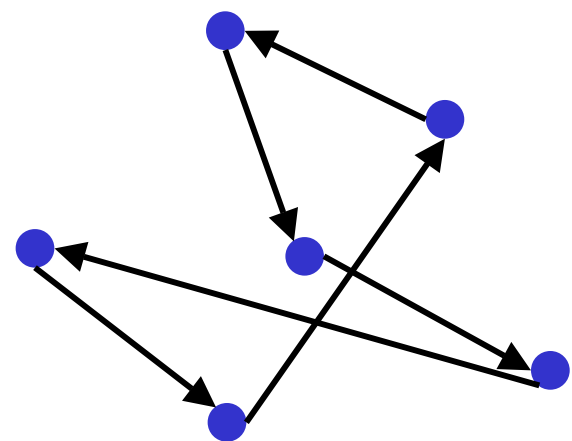
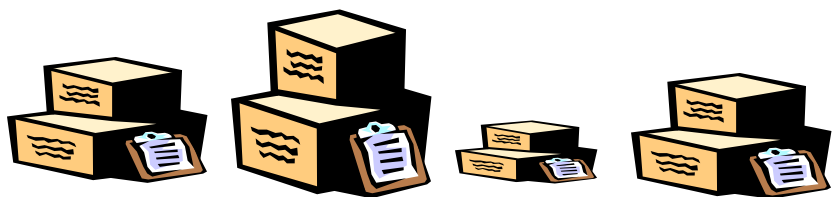
Example



Value complexity



Favorite Mechanism

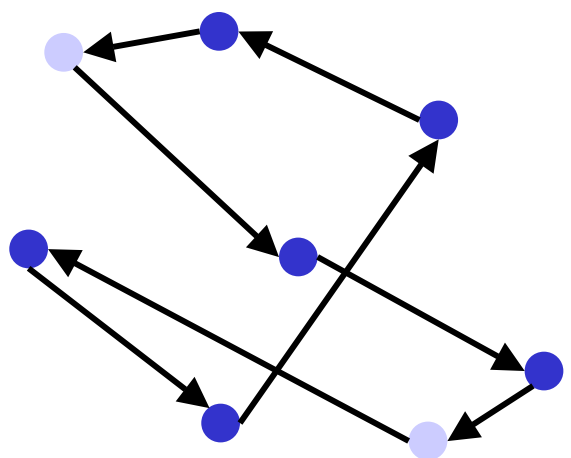


Current delivery schedule

Bid ?



Deliveries committed to



Delivery schedule with additional package

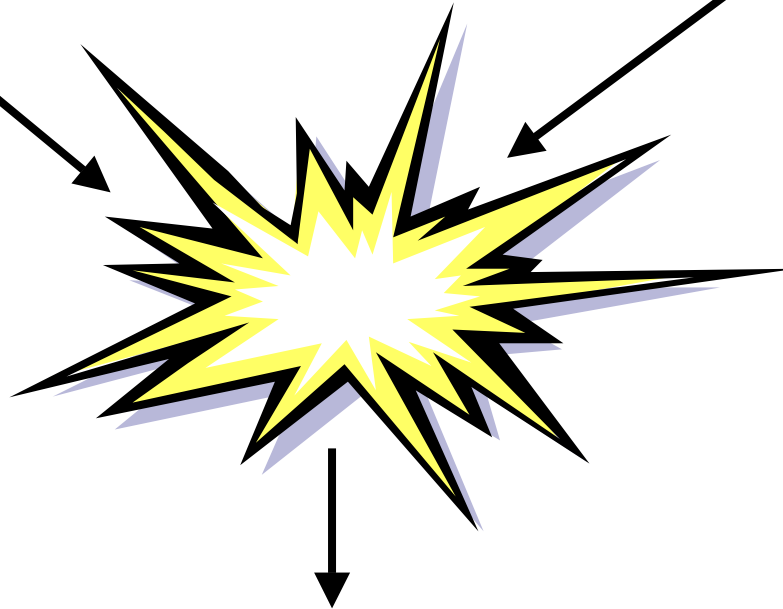
Value complexity

- Agents need values in order to participate (preferences)
- Figuring out values can have high computational overhead
- How should computational issues be handled game theoretically?
- Do computational issues have strategic implications?

Our approach

Resource
bounded
reasoning
from AI

Game theory and
mechanism
design

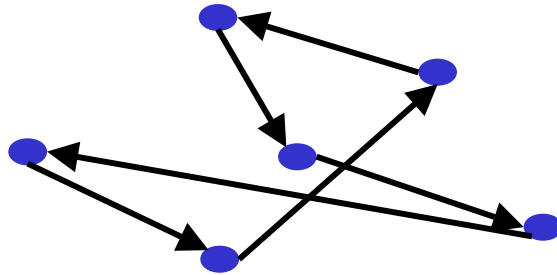


A Model of bounded rationality

Mechanism design for deliberative agents

Deliberative agents

- **Deliberative agent** — an agent that computes or gathers information



- Agents have
 - Anytime algorithms
 - Performance profile deliberation control
 - Cost functions which limit their deliberative capabilities

Anytime algorithms

- Anytime algorithms approximate values
 - Return a solution at any time
 - Solution improves over time
- Allows a tradeoff between computing time and solution quality
- Examples
 - **Iterative refinement algorithms:** Local search, simulated annealing
 - **Search algorithms:** Depth first search, branch and bound

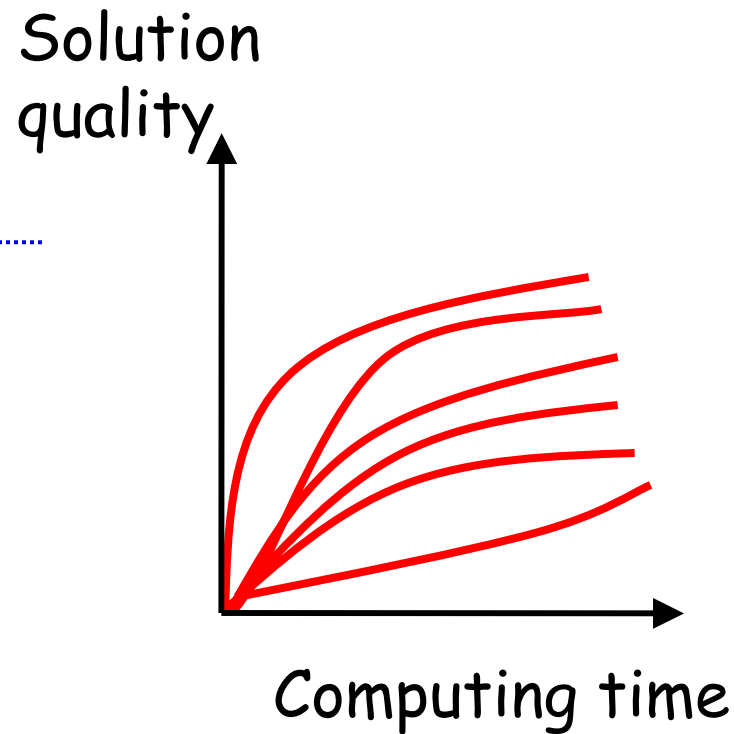
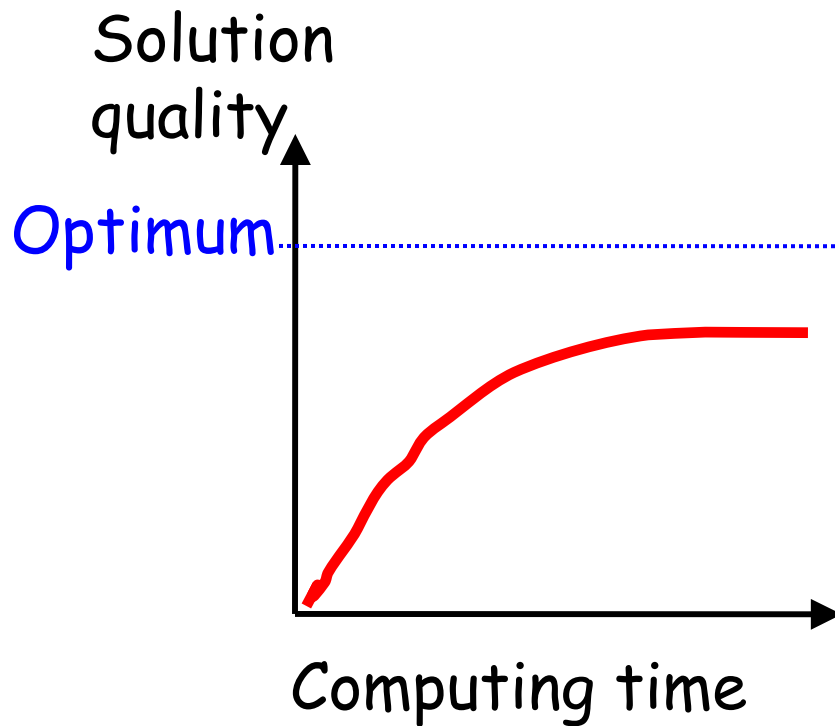
Performance profiles

- Performance profiles describe how computing changes solution quality
 - Characterize quality of algorithm's output as a function of computing time
- Use performance profiles to build **deliberation control policies**
- **Representation** is important
 - Earlier methods were not fully descriptive
 - They did not capture all possible ways an agent could control its deliberation

Performance profiles

Deterministic
performance profile

Variance introduced by
different problem instances



[Horvitz 87, 89, Dean & Boddy 89]

Table-based performance profile

[Zilberstein & Russell 91, 96]

Solution quality q

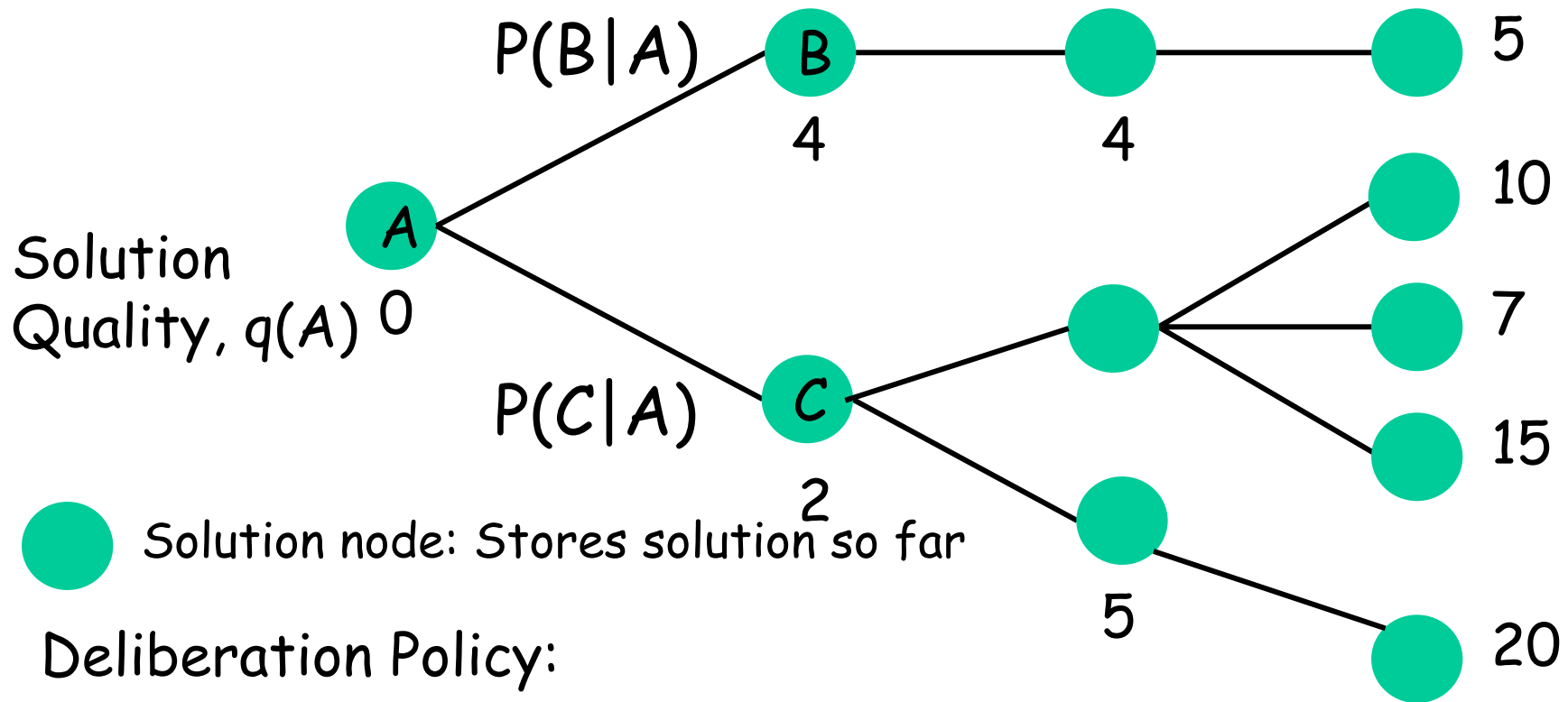
							.08	.19	.24
						.15	.30	.17	.39
				.16	.10	.16	.25	.30	.22
	.08	.04	.17	.20	.22	.30	.24	.19	.15
.09	.10	.20	.22	.23	.37	.31	.13	.15	
.11	.14	.33	.18	.21	.18	.08			
.22	.17	.25	.24	.15	.13				
.40	.31	.15	.19	.05					
.15	.20	.03							
.03									

† Computing time

Conditioning on solution quality so far [Hansen & Zilberstein 01]

Ignores conditioning on the path

Performance profile tree



Deliberation Policy:

$$\pi(q(n), t) = \operatorname{argmax}_d [q(n) - \operatorname{cost}(t) \text{ if } d = \text{stop}, \sum_{n'} P(n'|n) V(q(n'), t + \Delta t) \text{ if } d = \text{continue}]$$

Where

$$V(q(n), t) = \operatorname{max}_d [q(n) - \operatorname{cost}(t) \text{ if } d = \text{stop}, \sum_{n'} P(n'|n) V(q(n'), t + \Delta t) \text{ if } d = \text{continue}]$$

$n' \in \{\text{nodes at depth } \Delta t \text{ in the subtree rooted at node } n\}$

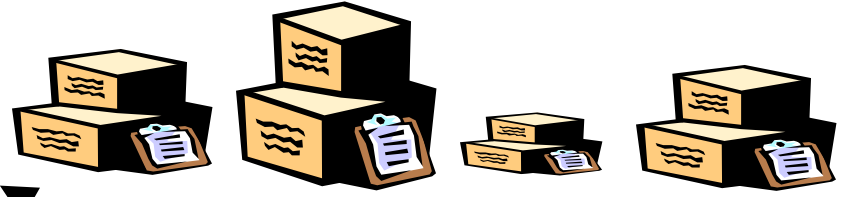
Properties of performance profile trees

- **Fully descriptive** - allows conditioning on everything including
 - Path of solution quality
 - Path of other solution features
 - Problem instance features
- Can model **uncertainty** from
 - Randomized algorithms
 - Lack of knowledge about what algorithms other agents use
 - Problem instances

Example



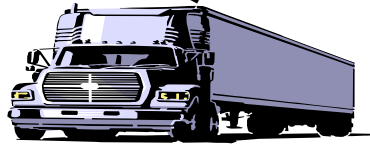
Favorite
Mechanism



bid_1

bid_2

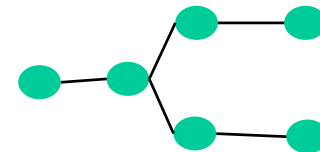
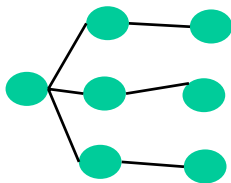
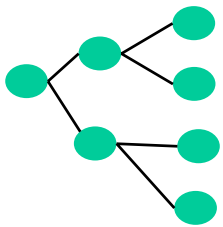
bid_3



Company 1

Company 2

Company 3



Anytime algorithms
Cost functions

Anytime algorithms
Cost functions

Anytime algorithms
Cost functions

Strategies of Agents

- A strategy includes both the **deliberating actions** as well as other (bidding) actions

DS=set of deliberation states, DA = set of deliberation actions,
 c^z = action of deliberating for one step on problem z

Vickrey Auction

(Bidding at time T)

$S_i: \text{Time} \times \text{DS} \rightarrow \text{DA} \times \mathbb{R}$

$S_i(t, ds) = (c^z, \emptyset)$ if $t \neq T$

$S_i(t, ds) = (c^z, V(n_i(T)))$ if $t = T$

Ascending Auction

$S_i: \text{Time} \times \text{DS} \times \text{Price} \rightarrow \text{DA} \times \{\text{stay, leave}\}$

$S_i(t, ds, p) = (c^z, a)$ $a \in \{\text{stay, leave}\}$

Strategic Deliberation

- Good estimates of other agents' values allow an agent to tailor its bidding strategy
- **Strategic deliberating:**
An agent uses some of its computational resources to approximate another's valuation

Auctions and strategic deliberating

	Auction mechanism	Classical Counter-speculation	Strategic deliberation	
			Deadlines	Costly
Single item for sale	First price sealed-bid	yes	yes	yes
	Descending	yes	yes	yes
	2 nd price sealed bid	no	no	yes
	Ascending	no	no	yes
Multiple items for sale	Generalized Vickrey On which $\langle \text{agent, bundle} \rangle$ pair to allocate next computation step ?	no	yes	yes

Computational Limitations and Mechanisms

- Allow agents to submit their **tools** for determining preferences
 - Performance profiles, algorithms, cost functions, problem instances.....
 - Truthful revelation, no strategic deliberation
 - Unreasonable assumptions
 - Can the mechanism handle all this information?
 - Can the agents practically submit all this information ?

Computational Limitations and Mechanisms

- Restrict to (interesting) mechanisms where agents only reveal (partially) computed results
 - Strategic deliberation may occur in any direct mechanism
 - Indirect mechanisms do not solve the problem:
 - Strategic deliberation can still occur
 - Issues with "non-truthfulness"

The Future?

- Allow the mechanism to help with some deliberation control
 - What information should the agents submit?
 - How to guarantee that truthful revelation will occur?
 - Who is solving the preference problems?
- Embrace strategic deliberation
 - If an agent is good at solving a certain type of problem, let it solve it for everyone!
 - Interesting incentive issues