Auction-Based Robot Coordination: Auction Robots


Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions

Structure of the Tutorial

- There are no prerequisites.
- We proceed in very small steps.
- We want everyone to understand everything.
- Please interrupt me if you have a question.

Coordination Task: Multi-Robot Routing

- Teams of robots are fault tolerant.
- Team of robots can operate in parallel.
- We study coordination tasks where a team of robots has to visit given targets in known or unknown terrain. Each target needs to be visited by one robot.
- Examples:
  - Planetary surface exploration
  - Facility surveillance
  - Search and rescue
- We use this coordination task only as an example.

Coordination Task: Multi-Robot Routing

- Properties (= important assumptions)
  - The robots are identical.
  - The robots know their own location.
  - The robots know the target locations.
  - The robots might not know where obstacles are.
  - The robots observe obstacles in their vicinity.
  - The robots can navigate without errors.
  - The path costs satisfy the triangle inequality.
  - The robots can communicate with each other.
Coordination Task: Multi-Robot Routing

(a possible solution, not necessarily the optimal one)

Coordination Task: Multi-Robot Routing

USC’s Player/Stage robot simulator

Coordination Task: Team Objectives

- Minimize SUM, the sum of the path costs over all robots (MINISUM = energy or distance)
- Minimize MAX, the maximal path cost over all robots (MINIMAX = makespan)
- Minimize AVE, the average arrival time over all targets (MINIAVE = flowtime)

Coordination Task: Team Objective 1 - MINISUM

- Minimize SUM, the sum of the path costs over all robots (MINISUM = energy or distance)
- Example: planetary surface exploration
Coordination Task:
Team Objective 1 - MINISUM
\[10 + 10 + 2 + 4 + 15 = 41\]

Coordination Task:
Team Objective 2 - MINIMAX
- Minimize \(\text{MAX}\), the maximal path cost over all robots
  \((\text{MINIMAX} = \text{makespan})\)
- Makespan = task completion time
- Example: mine clearing, facility surveillance

\[\max(10, 10, 2, 4, 15) = 15\]

Coordination Task:
Team Objective 3 - MINIAVE
- Minimize the AVE, average arrival time over all targets
  \((\text{MINIAVE} = \text{flowtime})\)
- Flowtime = average visitation time
- Example: search and rescue

\[\frac{1 + 2 + 3 + 4 + 6 + 9 + 10 + 1 + 4 + \ldots}{22} = 5.8\]

Coordination Task:
Team Objectives
- Minimize the sum of the path costs over all robots
  \((\text{MINISUM} = \text{energy or distance})\)
- Minimize the maximal path cost over all robots
  \((\text{MINIMAX} = \text{makespan})\)
- Minimize the average arrival time over all targets
  \((\text{MINIAVE} = \text{flowtime})\)
- In the following, we use MINISUM as example team objective unless stated otherwise!
Coordination Task
Mixed Integer Programming

Multi-robot routing is related to ...
- ... Vehicle/Location Routing Problems
- ... Traveling Salesman Problems (TSPs)
- ... Traveling Repairman Problems
except that the robots ...
- ... do not necessarily start at the same location
- ... are not required to return to their start location
- ... do not have capacity constraints

Multi-robot routing problems can be solved optimally with Mixed Integer Programming (MIP) methods, such as CPLEX:

Index sets and constants:
- $V_R = \text{Set of robot vertices}$
- $V_T = \text{Set of target vertices}$
- $c(i,j) = \text{Path cost from vertex } i \text{ to vertex } j$

Variables:
- $x_{ij} = \text{Is vertex } j \text{ visited by some robot directly after vertex } i$ (1 = yes, 0 = no)

Objective only

Objective and constraint C1 only

(a possible solution, not necessarily the optimal one)
Coordination Task: Mixed Integer Programming

Objective and constraints C1 and C2 only

(a possible solution, not necessarily the optimal one)

Coordination Task: Mixed Integer Programming

Objective and constraints C1, C2 and C3

(a possible solution, not necessarily the optimal one)

Coordination Task: Mixed Integer Programming

The number of subtour elimination constraints (C3) is exponential in the number of targets.

The MIPs are more complex for team objectives different from MINISUM.

Only small multi-robot routing problems can be solved optimally with MIP methods, even after tuning them (for example, by using cutting plane techniques).

Coordination Task: Mixed Integer Programming

In fact, only small multi-robot routing problems can be solved optimally with any method since MINISUM, MINIMAX and MINIAVE are NP-hard even if the terrain is completely known. The reduction is from Hamiltonian Path.

In fact, multi-robot routing problems resemble vehicle routing problems, which are notoriously harder than TSPs.

Since we cannot hope to minimize the team cost for multi-robot routing problems of realistic sizes and we need real-time performance, we only aim for a small but possibly suboptimal team cost (for example, one that is within a constant factor away from minimal).

Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions

Auctions for Agent Coordination: Research Issues

Our goal is to show:

- Auctions are an effective and practical approach to multi-robot routing and agent-coordination in general.
- Auctions have a small runtime.
- Auctions are communication efficient: information is compressed into bids
- Auctions are computation efficient: bids are calculated in parallel
- Auctions result in a small team cost.
Auctions for Agent Coordination: Research Issues

- There are some experimental results in the literature on agent coordination with auctions. Some publications report good team performance, others do not.
- We want to lay a firm theoretical foundation for agent coordination with auctions. Auction theory from economics is insufficient for such a foundation because we are dealing with cooperative (not: competitive) situations without
  - given preferences of the bidders
  - privacy concerns (revealing preferences)
  - concerns about gaming the system (untruthful bidding)

Auctions for Agent Coordination: Overview

<table>
<thead>
<tr>
<th>multi-robot routing</th>
<th>auctions</th>
</tr>
</thead>
<tbody>
<tr>
<td>robots</td>
<td>bidders</td>
</tr>
<tr>
<td>targets</td>
<td>items</td>
</tr>
</tbody>
</table>

Auctions for Agent Coordination: Known and Unknown Terrain

- Auctions can solve multi-robot routing problems
  - in known terrain
  - in unknown terrain

Auctions for Agent Coordination: Known Terrain

- We use an auction to solve the multi-robot routing problem for all targets.
- Each robot then determines a cost-minimal path to visit all targets it has won and follows it.
Auctions for Agent Coordination: Unknown Terrain

We use an auction to solve the multi-robot routing problem for all targets under the assumption that unknown terrain is easily traversable. (Other assumptions are possible, of course.) Each robot determines a cost-minimal path to visit all targets it has won and follows it.

Whenever a robot observes blocked terrain, it broadcasts this information to all other robots and we use another auction to solve the multi-robot routing problem for all remaining unvisited targets, again under the assumption that unknown terrain is easily traversable. Each robot determines a cost-minimal path to visit all targets it has won and follows it.

And so on, until all targets have been visited.

Auctions for Agent Coordination: Evaluation Criteria

- Ease of implementation
- Ease of decentralization
- Bid generation
- Bid communication
- Winner determination
- Team cost (with respect to team objective)

Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions
**Parallel Single-Item Auctions: Procedure**

- Classes of single-item auctions
  - First price auctions vs. second price auctions
  - Single-round auctions vs. multi-round auctions
- ...

**Parallel Single-Item Auctions: Example**

Each robot bids on each target in independent and simultaneous auctions.
- The robot that bids lowest on a target wins it.
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.
- Example: [Simmons et al. 2000]

**Warning:** I did all calculations for the examples late at night by hand!

**Example**

- Each robot bids on a target the minimal path cost it needs from its current location to visit the target.
Parallel Single-Item Auctions:

Example

- Minimal team cost (above) is not achieved.
- The team cost resulting from parallel single-item auctions is large because they cannot take synergies between targets into account.

Example

- It often does not make sense to send different robots to the same cluster of targets.

Synergies

- Each robot bids on a target the minimal path cost it needs from its current location to visit the target.
Parallel Single-Item Auctions:
Positive Synergy

A B

Smallest path cost to visit A: 5
Smallest path cost to visit B: 4
Smallest path cost to visit A and B: 5

Smallest path cost to visit A and B
<

Smallest path cost to visit A + Smallest path cost to visit B

(example: a cake is worth more than the sum of its ingredients)

Parallel Single-Item Auctions:
Negative Synergy

B C

Smallest path cost to visit B: 4
Smallest path cost to visit C: 4
Smallest path cost to visit B and C: 12

Smallest path cost to visit B and C
>

Smallest path cost to visit B + Smallest path cost to visit C

(example: two cars are worth less than the sum of the individual cars)

Parallel Single-Item Auctions:
Positive and Negative Synergies

A B C

Bid on A: 5
Bid on B: 4
Bid on C: 4

Parallel Single-Item Auctions:
Summary

- Ease of implementation: simple
- Ease of decentralization: simple
- Bid generation: cheap
- Bid communication: cheap
- Winner determination: cheap
- Team cost: large - no synergies taken into account

Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions

Ideal Combinatorial Auctions:
Procedure

- Each robot bids on all bundles (= subsets) of targets.
- Each robot wins at most one bundle, so that the number of targets won by all robots is maximal and, with second priority, the sum of the bids of the bundles won by robots is as small as possible.
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.
- Example: [Berhault et. al. 2003]
Ideal Combinatorial Auctions: Synergies

- Each robot bids on a bundle the minimal path cost it needs from its current location to visit all targets that the bundle contains.

**Example**

- Robot bids on {A}: 5
- Robot bids on {A,B}: 5
- Robot bids on {B}: 4
- Robot bids on {A,C}: 13
- Robot bids on {C}: 4
- Robot bids on {B,C}: 12
- Robot bids on {A,B,C}: 13

The team cost resulting from ideal combinatorial auctions is minimal since they take all synergies between targets into account, which solves an NP-hard problem. The number of bids is exponential in the number of targets. Bid generation, bid communication and winner determination are expensive.

Combinatorial Auctions: Procedure

- Each robot bids on some bundles (= sets) of targets.
- Each robot wins at most one bundle, so that the number of targets won by all robots is maximal and, with second priority, the sum of the bids of the bundles won by robots is as small as possible.
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.
- The team cost resulting from combinatorial auctions then is small but can be suboptimal. Bid generation, bid communication and winner determination are still relatively expensive.
- Example: [Berhault et. al. 2003]

Combinatorial Auctions: Bidding Strategies

- Which bundles to bid on is mostly unexplored in economics because good bundle-generation strategies are domain dependent. For example, one wants to exploit the spatial relationship of targets for multi-robot routing tasks.
- Good bundle-generation strategies:
  - generate a small number of bundles
  - generate bundles that cover the solution space
  - generate profitable bundles
  - generate bundles efficiently
Combinatorial Auctions: Domain-Independent Bundle Generation

Dumb bundle generation bids on all bundles (sort-of).
- THREE-COMBINATION
  - Bid on all bundles with 3 targets or less
- Note: It might be impossible to allocate all targets.

Combinatorial Auctions: Domain-Dependent Bundle Generation

Smart bundle generation bids on clusters of targets.
- GRAPH-CUT
  - Start with a bundle that contains all targets.
  - Bid on the new bundle.
  - Build a complete graph whose vertices are the targets in the bundle and whose edge costs correspond to the path costs between the vertices.
  - Split the graph into two subgraphs along (an approximation of) the maximal cut.
  - Recursively repeat the procedure twice, namely for the targets in each one of the two subgraphs.

Maximal cut = two sets that partition the vertices of a graph
Maximal cut = maxcut = cut that maximizes the sum of the costs of the edges that connect the two sets of vertices
Finding a maximal cut is NP-hard and needs to get approximated.
Combinatorial Auctions: Domain-Dependent Bundle Generation

Submit bids for the following bundles:
- \{A\}, \{B\}, \{C\}, \{D\}
- \{A,B\}, \{C,D\}
- \{A,B,C,D\}

Combinatorial Auctions: Experiments

3 robots in known terrain with 5 clusters of 4 targets each (doors are closed with 25 percent probability)

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Bids</th>
<th>SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel single-item auctions</td>
<td>635.1</td>
<td>426.5</td>
</tr>
<tr>
<td>Combinatorial auctions with THREE-COMBINATION</td>
<td>20506.5</td>
<td>247.9</td>
</tr>
<tr>
<td>Combinatorial auctions with GRAPH-CUT</td>
<td>1112.1</td>
<td>184.1</td>
</tr>
<tr>
<td>Optimal (MIP) = ideal combinatorial auctions</td>
<td>N/A</td>
<td>184.4</td>
</tr>
</tbody>
</table>

Note: 184.1 due to discretization issues
Combinatorial Auctions: Experiments

- Combinatorial auctions result in much smaller team costs than parallel single-item auctions.
- Smart bundle generation for combinatorial auctions decreases the number of bids and the team cost substantially over dumb bundle generation.
- The team cost of combinatorial auctions with smart bundle generation is near-minimal.
- Combinatorial auctions can be run in real time if the number of robots and targets are reasonably small.

Combinatorial Auctions: Summary

- Ease of implementation: difficult
- Ease of decentralization: unclear (form robot groups)
- Bid generation: expensive
  - Bundle generation: expensive (can be NP-hard)
  - Bid generation per bundle: ok (NP-hard)
- Bid communication: expensive
- Winner determination: expensive (NP-hard)
- Team cost: small (minimal) – many (all) synergies taken into account
  - Use a smart bundle generation method.
  - Approximate the various NP-hard problems.

Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions

Sequential Single-Item Auctions: Procedure

- Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BIDSUMPATH).
  - Example: [Lagoudakis et al. 2004, Tovey et al. 2005]

Sequential Single-Item Auctions: Synergy

- Example: **A** bids on **B** and **C**. **B** bids on **A** and **C**. **C** bids on **A**. The bids are as follows:
  - Bid on A: 5
  - Bid on B: 4
  - Bid on C: 4
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.
- Example: [Lagoudakis et al. 2004, Tovey et al. 2005]
Sequential Single-Item Auctions: Synergy

Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BIDSUMPATH). We give more details on this bidding rule later.

Sequential Single-Item Auctions: Example

Bid on A: (86)
Bid on B: (91)
Bid on C: (23)
Bid on D: (37)

Bid on A: (90)
Bid on B: (85)
Bid on C: (41)
Bid on D: 27

Sequential Single-Item Auctions: Example

Bid on A: (107)
Bid on B: (109)
Bid on C: (21)
Bid on D: (27)

Bid on A: (90)
Bid on B: (85)
Bid on C: (41)
Bid on D: 27
Sequential Single-Item Auctions: Example

Each robot needs to submit only one of its lowest bid.
Each robot needs to submit a new bid only directly after the target it bid on was won by some robot (either by itself or some other robot).
Thus, each robot submits at most one bid per round, and the number of rounds equals the number of targets. Consequently, the total number of bids is no larger than the one of parallel single-item auctions, and bid communication is cheap.
The bids that do not need to be submitted were shown in parentheses in the example.

Sequential Single-Item Auctions: Example

The team cost resulting from sequential single-item auctions is not guaranteed to be minimal since they take some but not all synergies between targets into account.

Sequential Single-Item Auctions: Procedure

- Each robot needs to submit only one of its lowest bid.
- Each robot needs to submit a new bid only directly after the target it bid on was won by some robot (either by itself or some other robot).
- Thus, each robot submits at most one bid per round, and the number of rounds equals the number of targets. Consequently, the total number of bids is no larger than the one of parallel single-item auctions, and bid communication is cheap.
- The bids that do not need to be submitted were shown in parentheses in the example.

Sequential Single-Item Auctions: Results

<table>
<thead>
<tr>
<th>parallel single-item auctions</th>
<th>sequential single-item auctions</th>
<th>optimal (MIP) = ideal combinatorial auctions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM = 426.98</td>
<td>SUM = 279.62</td>
<td>SUM = 271.04</td>
</tr>
</tbody>
</table>

Sequential Single-Item Auctions: Summary

- Ease of implementation: relatively simple
- Ease of decentralization: simple
- Bid generation: cheap (to be discussed later)
- Bid communication: cheap
- Winner determination: cheap
- Team cost: relatively small – some synergies taken into account
Sequential Single-Item Auctions: Derivation of Path Bidding Rules

- We suggest to use hill climbing to automatically derive bidding rules for sequential single-item auctions for a given team objective.
- Let a robot win a target so that some measure of the team cost increases the least.
- Robot r bids on target t the difference in the minimal measure of the team cost for the given team objective between the allocation of targets to all robots that results from the current allocation if robot r wins target t and the one of the current allocation. (Targets not yet won by robots are ignored.)

Sequential Single-Item Auctions: Derivation of Path Bidding Rules

- Measure of the team cost = team cost
- We suggest to use hill climbing to automatically derive bidding rules for sequential single-item auctions for a given team objective.
- Let a robot win a target so that the team cost increases the least.
- Robot r bids on target t the difference in the minimal team cost for the given team objective between the allocation of targets to all robots that results from the current allocation if robot r wins target t and the minimal team cost of the current allocation. (Targets not yet won by robots are ignored.)
Sequential Single-Item Auctions: Derivation of Path Bidding Rules

- Minimize the sum of the path costs over all robots (MINISUM = energy or distance)
- Minimize the maximal path cost over all robots (MINIMAX = makespan)
- Minimize the average arrival time over all targets (MINIAVE = flowtime)

How much to bid on target A?

- MINISUM = energy or distance

- Minimal path cost the robot needs from its current location to visit all targets it has won if it wins the target that it bids on

- Minimal path cost the robot needs from its current location to visit all targets it has won so far
Sequential Single-Item Auctions: Derivation of Path Bidding Rules

- **MINISUM = energy or distance**
  - Bid the increase in the minimal path cost the robot needs from its current location to visit all targets it has won if it wins the target it is bids on (BIDSUMPATH), which is exactly the common-sense bidding rule used earlier.

- **MINIMAX = makespan**
  - Bid the minimal path cost the robot needs from its current location to visit all targets it has won if it wins the target it is bids on (BIDMAXPATH), which balances the path costs of all robots.

- **MINIAVE = flowtime**
  - Bid the increase in the minimal sum of arrival times the robot needs from its current location to visit all targets it has won if it wins the target it is bids on (BIDAVEPATH).

Finding the minimal path cost for visiting a given set of targets is NP-hard. We therefore use the polynomial-time cheapest insertion heuristic (or more sophisticated heuristics based on two-opt, a TSP hill-climbing method).
Sequential Single-Item Auctions: Results for Path Bidding Rules

- **2 robots in known terrain of size 51 by 51 with 10 unclustered targets**
  
<table>
<thead>
<tr>
<th>SUM</th>
<th>MAX</th>
<th>AVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIDSUMPATH</td>
<td>134.18</td>
<td>97.17</td>
</tr>
<tr>
<td>BIDMAXPATH</td>
<td>144.84</td>
<td>90.10</td>
</tr>
<tr>
<td>BIDAVEPATH</td>
<td>157.29</td>
<td>100.56</td>
</tr>
<tr>
<td>optimal (MIP)</td>
<td>132.06</td>
<td>85.86</td>
</tr>
</tbody>
</table>

- **10 robots in unknown terrain of size 51 by 51 with 100 unclustered targets**
  
<table>
<thead>
<tr>
<th>SUM</th>
<th>MAX</th>
<th>AVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIDSUMPATH</td>
<td>596.10</td>
<td>223.20</td>
</tr>
<tr>
<td>BIDMAXPATH</td>
<td>677.80</td>
<td>110.60</td>
</tr>
<tr>
<td>BIDAVEPATH</td>
<td>697.80</td>
<td>121.50</td>
</tr>
<tr>
<td>optimal (MIP)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- **10 robots in unknown terrain of size 51 by 51 with 100 clustered targets**
  
<table>
<thead>
<tr>
<th>SUM</th>
<th>MAX</th>
<th>AVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIDSUMPATH</td>
<td>35.43</td>
<td>121.50</td>
</tr>
<tr>
<td>BIDMAXPATH</td>
<td>37.92</td>
<td>110.60</td>
</tr>
<tr>
<td>BIDAVEPATH</td>
<td>49.15</td>
<td>100.56</td>
</tr>
<tr>
<td>optimal (MIP)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
- Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions
Sequential Single-Item Auctions: Derivation of Tree Bidding Rules

- Path bidding rules ("direct approach")
  - Find paths directly
- Tree bidding rules ("indirect approach")
  - Find trees and convert them to paths

We suggest to use hill climbing to automatically derive bidding rules for sequential single-item auctions for a given team objective.

Every robot maintains a tree. Let a robot win a target so that the cost of the forest (the sum of the costs of its edges) increases the least.

Robot \( r \) bids on target \( t \) the difference in the minimal cost of the forest for the given team objective between the allocation of targets to all robots that results from the current allocation if robot \( r \) wins target \( t \) and the one of the current allocation. (Targets not yet won by robots are ignored.)

Sequential Single-Item Auctions: Derivation of Tree Bidding Rules

- Minimize the sum of the path costs over all robots (MINISUM = energy or distance)
- Minimize the maximal path cost over all robots (MINIMAX = makespan)
- Minimize the average arrival time over all targets (MINIAVE = flowtime)

The multi-robot routing problem is similar to TSP problems. Thus, we borrow ideas from solving TSP problems.

We find a minimum spanning tree (MST) - a tree that contains the current location of the robot and all targets so that the tree cost is minimal.

Circumnavigating the MST results in a constant-factor (= two) cost guarantee for TSP problems and thus also for single-robot routing problems.
Constructing an MST with the greedy Prim algorithm whose root is the current location of the robot is equivalent to a sequential single-item auction where the robot bids on a target the increase in the cost of the MST if it wins the target (= the minimal path cost from the target to either its current location or any of the targets it has already won).

The robot then circumnavigates the MST. (We use depth-first search of the tree, lightest branch first, with shortcuts to the next unvisited target and stop once all targets have been visited.)

Each robot bids on a target the increase in the cost of the MST if it wins the target = the minimal path cost from the target to either its current location or any of the targets it has already won.

Each robot bids on a target the increase in the cost of the MST if it wins the target = the minimal path cost from the target to either its current location or any of the targets it has already won.

Each robot bids on a target the increase in the cost of the MST if it wins the target = the minimal path cost from the target to either its current location or any of the targets it has already won.
We can generalize this idea to multiple robots.

We now find a minimum spanning forest (MSF) – a collection of trees, each of which contains the current location of one of the robots so that the collection contains all targets and the sum of the costs of its edges is minimal.

We again use a sequential single-item auction where each robot bids on a target the increase in the cost of its tree if it wins the target (= the minimal path cost from the target to either its current location or any of the targets it has already won).

Each robot then circumnavigates its tree. We again use depth-first search of the tree, lightest branch first, with shortcuts to the next unvisited target and stop once all targets have been visited.

Each robot bids on a target the increase in the cost of its tree if it wins the target = the minimal path cost from the target to either its current location or any of the targets it has already won (BIDSUMTREE).
Each robot bids on a target the increase in the cost of its tree if it wins the target = the minimal path cost from the target to either its current location or any of the targets it has already won (BIDSUMTREE).
Sequential Single-Item Auctions:
Derivation of Tree Bidding Rules

- Minimize the maximal path cost over all robots (MINIMAX = makespan)
- Minimize the average arrival time over all targets (MINIAVE = flowtime)

Bid the cost of its tree if the robot wins the target (BIDMAXTREE).

Sequential Single-Item Auctions:
Derivation of Path Bidding Rules

- MINIAVE = flowtime
  Bid the minimal path cost the robot needs from its current location to visit the target (BIDAVETREE).

Path bidding rules ("direct approach")
- Find paths directly (finding optimal paths: NP-hard, instead: use polynomial time cheapest insertion heuristic)
- Tree bidding rules ("indirect approach")
  - Find optimal trees (in polynomial time)
  - Convert trees to (possibly suboptimal) paths (optimal conversion: NP-hard, instead: use polynomial time circumnavigation heuristic)

Summary of Bidding Rules
Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Path bidding rules
  - Tree bidding rules
- Analytical results for sequential single-item auctions
- Conclusions

Sequential Single-Item Auctions: Analytical Results

- MINISUM, MINIMAX and MINIAVE are NP-hard.
- BIDSUMPATH/TREE ≥ factor 1.5 away from MINISUM
- BIDMAXPATH/TREE ≥ factor 3 away from MINIMAX
- BIDAVEPATH/TREE ≥ factor 2 away from MINIAVE

Sequential Single-Item Auctions: Analytical Results

- The first constant-factor cost guarantee for completely polynomial auction schemes:
  - BIDSUMTREE is 2-optimal for MINISUM even if the paths are obtained from the tree with the circumnavigation heuristic (trivial: a simple generalization of the proof of the constant-factor cost guarantee of the Prim algorithm for TSPs).
  - BIDSUMPATH is 2-optimal for MINISUM even if the path costs are approximated with the cheapest insertion heuristic (more difficult: see proof sketch on the next slide).

Sequential Single-Item Auctions: Analytical Results

- Greedy construction of MSTs

Sequential Single-Item Auctions: Analytical Results

- Greedy construction of MSTs
Sequential Single-Item Auctions: Analytical Results

- Greedy construction of MSTs

- Prim algorithm for TSPs: 2-optimal for TSP
Sequential Single-Item Auctions: Analytical Results

- Prim algorithm for TSPs

Sequential Single-Item Auctions: Analytical Results

- Greedy construction of MFTs

Sequential Single-Item Auctions: Analytical Results

- Greedy construction of MFTs
Sequential Single-Item Auctions: Derivation of Tree Bidding Rules

- **BIDSUMTREE**

Sequential Single-Item Auctions: Derivation of Tree Bidding Rules

- **BIDSUMTREE**

Sequential Single-Item Auctions: Analytical Results

- **BIDSUMPATH** is 2-optimal for MINISUM even if the path costs are approximated with the cheapest insertion heuristic (more difficult: see proof sketch on the next slide).

Sequential Single-Item Auctions: Analytical Results

- Such constant factor cost guarantees do not exist for BIDMAXPATH/TREE and BIDAVEPATH/TREE.

paths resulting from BIDMAXPATH
Such constant factor cost guarantees do not exist for BIDMAXPATH/TREE and BIDAVEPATH/TREE.

---

**3 team objectives**
- MINISUM, MINIMAX and MINIAVE

**6 bidding rules**
- 3 path bidding rules, one for each team objective
  - BIDSUMPATH, BIDMAXPATH and BIDAVEPATH
- 3 tree bidding rules, one for each team objective
  - BIDSUMTREE, BIDMAXTREE and BIDAVETREE

A lower bound and an upper bound on the worst-case cost ratio for the 18 combinations of team objectives and bidding rules.
Sequential Single-Item Auctions: Analytical Results

- The path bidding rule has experimentally a much larger runtime than the tree bidding rule (for a large number of robots and targets) but results in slightly smaller team costs.
- The difference in SUM in our experiments is about 5 percent for MINISUM in known and up to about 15 percent for MINISUM in unknown terrain.

Conclusions: Related Work

- Robotics
  - Auctions implemented and tested on robots as heuristics for complex coordination problems
- Computer Science Theory
  - MINISUM, MINIMAX and MINIAVE can be approximated efficiently with constant-factor cost guarantees
  - Centralized and complicated algorithms
- Operations Research
  - Algorithms for vehicle/location routing problems, that are similar to multi-robot routing problems

Conclusions: Main Lesson

- Auctions are an effective and practical approach to multi-robot routing and (likely) agent-coordination in general.

Structure of the Tutorial

- Overview of the coordination task
- Auction mechanisms
  - Parallel single-item auctions
  - Combinatorial auctions
  - Sequential single-item auctions
- Bidding rules for sequential single-item auctions
  - Tree bidding rules
  - Path bidding rules
- Analytical results for sequential single-item auctions
- Conclusions

Conclusions: Insights

- We presented a framework for auction-based agent coordination based on sequential single-item auctions and demonstrated it for multi-robot routing.
- Sequential single-item auctions provide a good trade-off between parallel single-item auctions and combinatorial auctions.
- The bidding rules for sequential single-item auctions can be derived automatically from given team objectives and do well experimentally.
- The properties of sequential single-item auctions can be analyzed theoretically. For example, they provide constant-factor cost guarantees in some cases.

Literature

- Find our papers at idm-lab.org/project-c.html
- Check out the papers by Anthony Stentz (CMU) and collaborators
  - Maja Matarić (USC) and collaborators
  - Barbara Grosz (Harvard) and collaborators
  - Stefano Rizzi (Bologna) and collaborators
  - and lots of other researchers…
A manufacturer produces animal feed with 2 active ingredients and a filler ingredient. One kg of feed mix must contain 90g of nutrient A, 50g of nutrient B, 20g of nutrient C and 2g of nutrient D. The ingredients have the following nutrient values and costs. What should be the amount of ingredients in one kg of feed mix to minimize cost? (The filler ingredient costs nothing.)

<table>
<thead>
<tr>
<th>Active Ingredient 1 (g/kg)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>$/kg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>80</td>
<td>40</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>Active Ingredient 2 (g/kg)</td>
<td>200</td>
<td>150</td>
<td>20</td>
<td></td>
<td>60</td>
</tr>
</tbody>
</table>

A = kg of ingredient 1 per kg of feed mix
B = kg of ingredient 2 per kg of feed mix

Minimize $40A + 60B$

Subject to
- $A + B \leq 1$
- $100A + 200B \geq 90$
- $80A + 150B \geq 50$
- $40A + 20B \geq 20$
- $10A \geq 2$
- $A \geq 0, B \geq 0$

LPs can be solved in polynomial time.
In practice, one often uses variants of the simplex method (which is not guaranteed to run in polynomial time but often runs very fast).

What if there is a fixed cost of $15 if any of active ingredient 2 is used?

A = kg of ingredient 1 per kg of feed mix
B = kg of ingredient 2 per kg of feed mix
X = 1 if B > 0

Minimize $40A + 60B + 15X$

Subject to
- $A + B \leq 1$
- $100A + 200B \geq 90$
- $80A + 150B \geq 50$
- $40A + 20B \geq 20$
- $10A \geq 2$
- $B \leq X$
- $A \geq 0, B \geq 0, X \geq 0$
- X is integer
Appendix  
Mixed Integer Programming (MIP)

What if they need to satisfy only three of the four nutrient constraints?

A = kg of ingredient 1 per kg of feed mix  
B = kg of ingredient 2 per kg of feed mix  
X1 = 0 if constraint 1 is not satisfied  
X2 = …

Minimize 40 A + 60 B + 15 X  
Subject to  
A + B ≤ 1  
100 A + 200 B ≥ 90 X1  
80 A + 150 B ≥ 5 X2  
40 A + 20 B ≥ 20 X3  
10 A ≥ 2 X4  
X1 + X2 + X3 + X4 ≥ 3  
A ≥ 0, B ≥ 0, 1 ≥ X1 ≥ 0, 1 ≥ X2 ≥ 0, 1 ≥ X3 ≥ 0, 1 ≥ X4 ≥ 0  
X1, X2, X3 and X4 are integer

Appendix  
Mixed Integer Programming (MIP)

MIPs can likely not be solved in polynomial time (solving them is NP-hard).  
Viewing MIPs as LPs and rounding variables to the closest integer can result in large errors.  
In practice, one often uses CPLEX, industrial-strength software that can solve LPs and MIPs.

Appendix  
Literature for LP and MIP Material

This problem and many more examples can be found in the OR Notes by J. E. Beasley at  
people.brunel.ac.uk/~mastjjb/jeb/or/contents.html

Funding

We acknowledge funding from NSF awards  
DMI-0113881  
IIS-0413196  
IIS-0412912

Thank you!