Tutorial on Auction-Based Agent Coordination at AAAI 2006

Abstract

Teams of agents are more robust and potentially more efficient than single agents. However, coordinating teams of agents so that they can successfully complete their mission is a challenging task. This tutorial will cover one way of efficiently and effectively coordinating teams of agents, namely with auctions. Coordination involves the allocation and execution of individual tasks through an efficient (preferably decentralized) mechanism. The tutorial on "Auction-Based Agent Coordination" covers empirical, algorithmic, and theoretical aspects of auction-based methods for agent coordination, where agents bid on tasks and the tasks are then allocated to the agents by methods that resemble winner determination methods in auctions. Auction-based methods balance the trade-off between purely centralized coordination methods which require a central controller and purely decentralized coordination methods without any communication between agents, both in terms of communication efficiency, computation efficiency, and the quality of the solution.

The tutorial will use the coordination of a team of mobile robots as a running example. Robot teams are increasingly becoming a popular alternative to single robots for a variety of difficult tasks, such as planetary exploration or planetary base assembly. The tutorial covers auction-based agent coordination using examples of multi-robot routing tasks, a class of problems where a team of mobile robots must visit a given set of locations (for example, to deliver material at construction sites or acquire rock probes from Martian rocks) so that their routes are optimized based on certain criteria, for example, minimize the consumed energy, completion time, or average latency. Examples of multi-robot routing tasks include search-and-rescue in areas hit by disasters, surveillance, placement of sensors, material delivery, and localized measurements. We also discuss agent-coordination tasks from domains other than robotics. We give an overview of various auction-based methods for agent coordination, discuss their advantages and disadvantages and compare them to each other and other coordination methods. The tutorial also covers recent theoretical advances (including constant-factor performance guarantees) as well as experimental results and implementation issues.

Intended Audience

The tutorial makes no assumptions about the background of the audience, other than a very general understanding of algorithms, and should be of interest to all researchers who are interested in robotics, autonomous agents and multi-agent systems. Thus, the tutorial is appropriate undergraduate and graduate students as well as researchers and practitioners who are interested in learning more about how to coordinate teams of agents using auction-based mechanisms.

Additional Information

For pointers to lots of additional material visit the tutorial webpage:

- idm-lab.org/auction-tutorial.html (scroll to the bottom)
- metropolis.cta.ri.cmu.edu/markets/wiki

For questions or requests for additional information, please send email to Sven Koenig (skoenig@usc.edu).

Speakers

The speakers will be Bernardine Dias, Sven Koenig, Michail Lagoudakis, Robert Zlot, Nidhi Kalra, and Gil Jones. The presented material is provided by the researchers listed below and includes material by their co-workers A. Stentz, D. Kempe, A. Meyerson, V. Markakis, A. Kleywegt and C. Tovey. Special thanks go to Anthony Stentz, a research professor with the Robotics Institute of Carnegie Mellon University and the associate director of the National Robotics Engineering Consortium at Carnegie Mellon University, and Craig Tovey, a professor in Industrial and System Engineering at Georgia Institute of Technology.

Bernardine Dias (Carnegie Mellon University, USA)
www.ri.cmu.edu/people/dias_m.html

M. Bernardine Dias is research faculty at the Robotics Institute at Carnegie Mellon University. Her research interests are in technology for developing communities, multirobot coordination, space robotics, and diversity in computer science. Her dissertation developed the TraderBots framework for market-based multirobot coordination and she has published extensively on a variety of topics in robotics.

E. Gil Jones (Carnegie Mellon University, USA)
www.ri.cmu.edu/people/jones_edward.html

E. Gil Jones is a Ph.D. student at the Robotics Institute at Carnegie Mellon University. His primary interest is market-based multi-robot coordination. He received his BA in Computer Science from Swarthmore College in 2001, and spent two years as a software engineer at Bluefin Robotics in Cambridge, Mass.

Nidhi R. Kalra (Carnegie Mellon University, USA)
www.cs.cmu.edu/~nidhi/

Nidhi R. Kalra is a Ph.D. student at the Robotics Institute at Carnegie Mellon University. She is interested in developing coordination strategies for robots working on complex real-world problems. To this end, she is developing the market-based Hoplites framework for tight multirobot coordination.

Pinar Keskinocak (Georgia Institute of Technology, USA)
www.isye.gatech.edu/people/faculty/Pinar_Keskinocak/home.html

Pinar Keskinocak is an associate professor at Georgia Institute of Technology. She is interested in electronic commerce, routing and scheduling applications, production planning, multi-criteria decision making, approximation algorithms, and their application to a variety of problems. Pinar has published extensively in operation research.

Sven Koenig (University of Southern California, USA)
idm-lab.org

Sven Koenig is an associate professor at the University of Southern California. From 1995 to 1997, Sven demonstrated that it is possible to combine ideas from different decision-making disciplines by developing a robust mobile robot architecture based on POMDPs from operations research. Since then, he has published over 100 papers in robotics and artificial intelligence, continuing his interdisciplinary research.

Michail G. Lagoudakis (Technical University of Crete, Greece)
www.intelligence.tuc.gr/~lagoudakis/

Michail G. Lagoudakis is an assistant professor at the Technical University of Crete. He is interested in machine learning (reinforcement learning), decision making under uncertainty, numeric artificial intelligence, as well as robots and other complex systems. He has published extensively in artificial intelligence and robotics.

Robert Zlot (Carnegie Mellon University, USA)
www.cs.cmu.edu/~robz/

Robert Zlot is a PhD student at the Robotics Institute at Carnegie Mellon University, where he earned a Master's degree in Robotics in 2002. Robert's main interests are in multirobot coordination and space robotics. His current research focuses on market-based algorithms for tasks that exhibit complex structure.

AAAI 2006 Tutorial on Auction-Based Agent Coordination

M. Bernardine Dias, Gil Jones, Nidhi R. Kalra, Pinar Keskinocak, Sven Koenig, Michail G. Lagoudakis, Robert Zlot

includes material or ideas by
D. Kempe, A. Kleywegt, V. Markakis, A. Meyerson, A. Stentz, C. Tovey
with special thanks to
A. Stentz and C. Tovey

Tutorial Guidelines

- There are no prerequisites.
- We proceed in very small steps.
- We want everyone to understand everything.
- Please ask if you have questions.

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

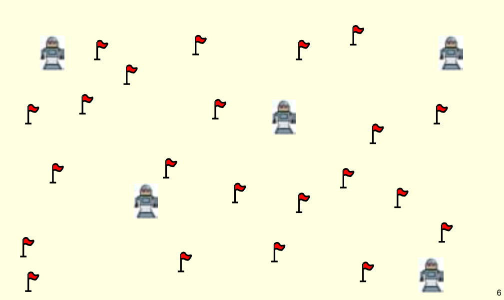
A Typical Coordination Task: Multi-Robot Routing

- Agents=Robots, Tasks=Targets
- A team of robots has to visit given targets spread over some known or unknown terrain. Each target must be visited by one robot.
- Examples:
  - Planetary surface exploration
  - Facility surveillance
  - Search and rescue

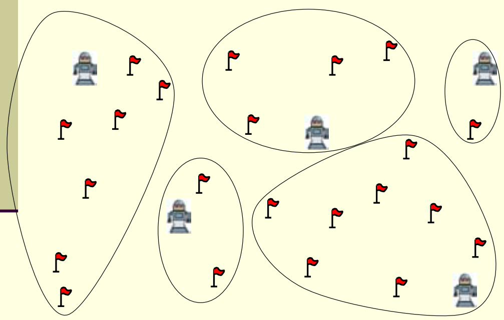
A Typical Coordination Task: Multi-Robot Routing Assumptions

- 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.

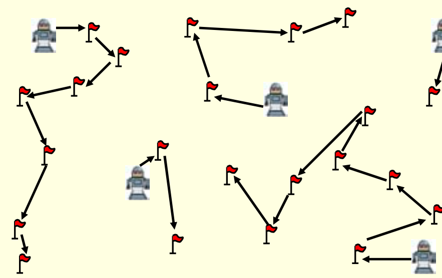
A Typical Coordination Task: Multi-Robot Routing



A Typical Coordination Task: Multi-Robot Routing



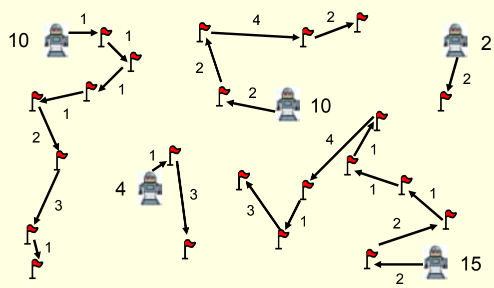
A Typical Coordination Task: Multi-Robot Routing



(a possible solution, not necessarily the optimal one)

A Typical Coordination Task: MiniSum Team Objective

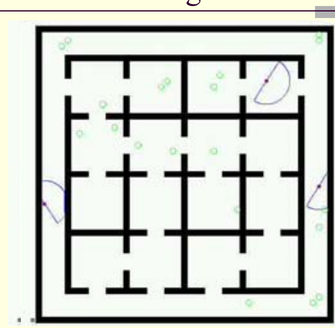
10+10+2+4+15 = 41



A Typical Coordination Task: Multi-Robot Routing

- 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

A Typical Coordination Task: Multi-Robot Routing



USC's Player/Stage robot simulator

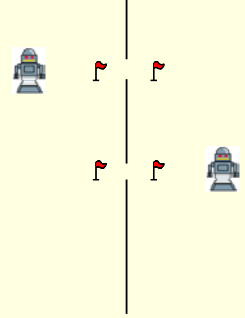
Auctions for Robot Coordination: Overview

Agent coordination	Auctions
agents	bidders
tasks	items
cost	currency

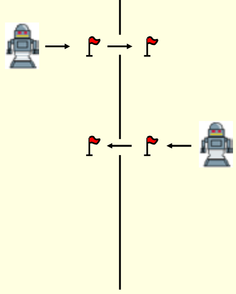
Auctions for Robot Coordination: Advantages

- Auctions are an effective and practical approach to agent-coordination.
- 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 can be used if the terrain or the knowledge of the robots about the terrain changes.

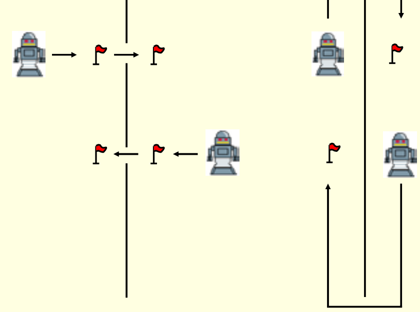
Auctions for Robot Coordination: Known Terrain



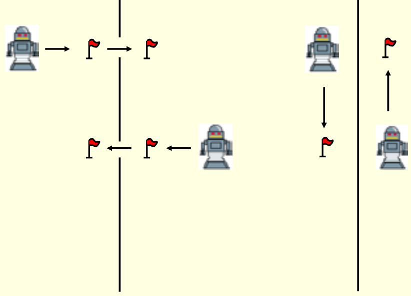
Auctions for Robot Coordination: Known Terrain



Auctions for Robot Coordination: Unknown Terrain



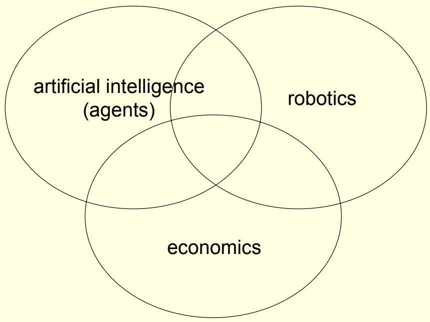
Auctions for Robot Coordination: Unknown Terrain



Auctions for Robot Coordination: Overview of the Tutorial

- 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.
- We want to show experimentally that auctions can be successfully applied to a variety of agent-coordination problems.

Auctions for Robot Coordination: Disciplines



19

Auctions for Robot Coordination: Who are we?

- We are researchers from two different groups with active research on auctions who have never published together.
- One of the groups is at CMU, with research(ers) centered on robotics.
- The other group is distributed across different universities, with research(ers) in artificial intelligence, robotics, economics and theoretical computer science.

20

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

21

Structure of the Tutorial

- We now give an overview of the results of research on auctions in economics.
- We then explain why we can build on that research but need additional results to apply auctions to agent coordination.

22

What is an auction?

- Definition [McAfee & McMillan, JEL 1987]:
  - a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants.
- Examples:

FreeMarkets
SOTHEBYS
CHRISTIE'S
NASDAQ
ebay

Going once, …
going twice, ...

23

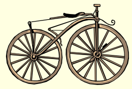
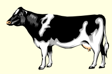
Why are we interested in auctions?

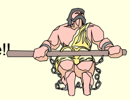
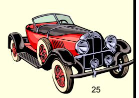
- Auctions have been widely used for many years...

24

Why are we interested in auctions?

- ... and many commodities
  - Antiques and art
  - Livestock and other agricultural produce
  - Real estate
  - Mineral and timber rights
  - Radio frequencies
  - Diamonds
  - Corporate stock
  - Treasury bonds
  - Used automobiles
  - Wives and slaves
  - Body parts and human tissue!!

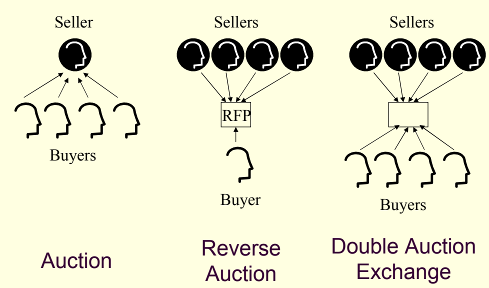
Pricing models

- Posted prices
  - Static
  - Dynamic
    - Change dynamically over time
    - Customized pricing
- Price discovery mechanisms
  - Negotiations
  - Auctions

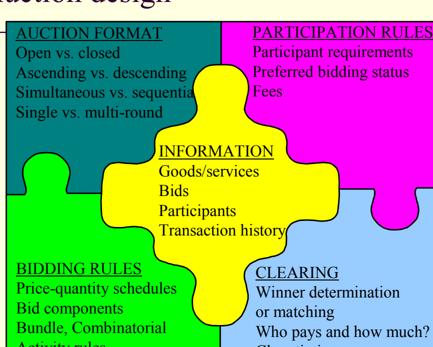
Why auctions?

- For object(s) of unknown value
- Mechanized
  - reduces the complexity of negotiations
  - ideal for computer implementation
- Creates a sense of “fairness” in allocation when demand exceeds supply

Auction formats



Auction design



Bidding strategies

- At which auctions to participate?
  - Participation cost, auction duration, number of bidders
- When to bid?
- How much to bid? (price and/or quantity)
  - Effects of synergies or economies of scale

Important issues on designing auctions with human participants

- "Efficient" allocation:
  the bidders who values an item most gets it
  - Incentives for truthful bidding
- Maximize the auctioneer's revenue
- Things to avoid:
  - Collusion
    - If some bidders collude, they might do better by lying. Collusion among buyers, sellers, and/or auctioneer.
  - Hide-in-the-grass strategy
  - Predatory bidding
  - Jump bidding
  - Shilling
  - Bid shielding
  - Winner's curse

Differences of auctions with robot participants

- Robots don't game the system, e.g. by bidding untruthfully. They bid as we ask them to!
- Robots do not intentionally "hide" information and thus do not have privacy concerns.
- Robots do not have inherent utilities (preferences). We define their utilities so that the result of the auction serves a common "team" objective.
- Robots don't care if the outcome is not "fair."

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

Outline

- Common auction mechanisms used for agent coordination
- Protocols and practical issues

Types of Auction Mechanisms

- Mechanism for allocating items (= goods, tasks, resources, …) for agent coordination
  - Single seller, multiple buyers
  - Seller wants to acquire the maximum amount of revenue from the bidders for items (*e.g.*, contract tasks for the minimum cost)
- Open-cry vs. sealed bid
- Reserve prices

Types of Auction Mechanisms

- Common auction types for agent coordination
  - Single-item auctions
  - Multi-item auctions
  - Combinatorial auctions
- We will use the example of tasks for during the descriptions of the protocols

Single-Item Auctions

- Auctioneer is selling a single task
- First-price auction
  - Protocol: Each bidder submits a bid containing a single number representing its cost for the task. The bidder with the lowest bid wins and is awarded the task, agreeing to perform it for the price of its bid.
- Vickrey (second-price) auction
  - Protocol: Same as above, but bidder with the lowest bid agrees to perform task for the price of the second-lowest bidder's bid.
  - Incentive compatible.
- Which mechanism?
  - Doesn't matter if robots bid truthfully

Multi-Item Auctions

- Protocol: Auctioneer offers a set of *t* tasks. Each bidder may submit bids on some/all of the tasks. The auctioneer awards one or more tasks to bidders, with *at most one* task awarded to each bidder.
  - No multiple awards: bids do not consider cost dependencies.
- Protocol may specify a fixed number of awards, *e.g.*:
  1) m tasks awarded, $1 \leq m \leq$ *#bidders*
  2) Every bidder awarded one task (m = *#bidders*)
  3) The one best award ($m = 1$)
- For 2) the assignment can be done optimally [Gerkey and Matarić 04]
  - Greedy algorithm common: Award the lowest bidder with the associated task, eliminate that bidder and task from contention, and repeat until you run out of tasks or bidders.

Combinatorial Auctions

- Protocol: Auctioneer offers a set of tasks *T*. Each bidder may submit bids on any task *bundles* (subsets of *T*), and the auctioneer awards a combination of bundles to multiple bidders (at most one bundle awarded per bidder). The awards should maximize the revenue for the auctioneer.
- Exponential number of bundles, $2^{|T|}$
  - Winner determination is NP-hard
  - But, fast optimal *winner determination* algorithms exist that take advantage of the sparseness of the bid set [*e.g.* CABOB, Sandholm 2002]
- Number of bundles can be reduced
  - Auctioneer: only allow certain bundles
    - Roles [Hunsberger and Grosz 00]
    - Rings or nested structure [Rothkopf *et al.* 98]
  - Bidders: task clustering algorithms [Berhault *et al.* 03, Dias *et al.* 02, Nair *et al.* 02]
    - Clustering (spanning tree, greedy nearest neighbor)
    - Limit bundle size
    - Recursive max graph cuts

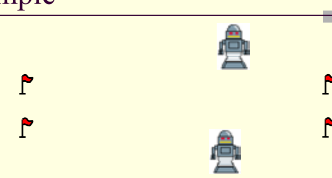
Auctions for Robot Coordination: Types of auctions

- We now discuss 3 auction types in more detail
  - Parallel Auctions
  - Combinatorial Auctions
  - Sequential Auctions

Parallel Auctions: Procedure

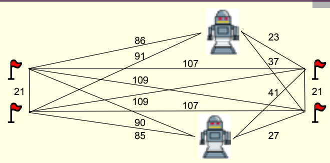
- Each robot bids on each target in independent and simultaneous auctions.
- The robot that bids lowest on a target wins it.
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.

Parallel Auctions: Example



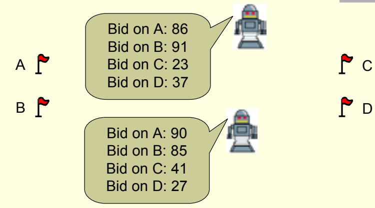
- Each robot bids on a target the minimal path cost it needs from its current location to visit the target.

Parallel Auctions: Example



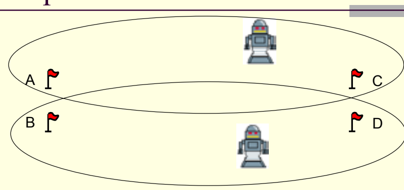
- Each robot bids on a target the minimal path cost it needs from its current location to visit the target.

Parallel Auctions: Example

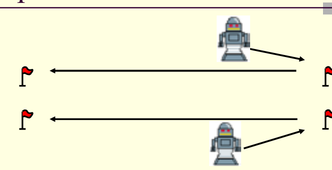


- Each robot bids on a target the minimal path cost it needs from its current location to visit the target.

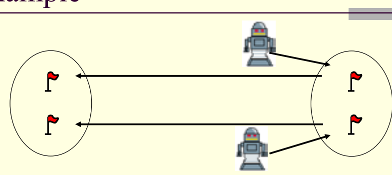
Parallel Auctions: Example



Parallel Auctions: Example

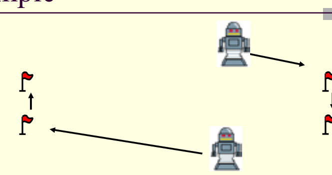


Parallel Auctions: Example



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

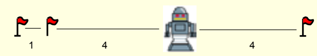
Parallel Auctions: Example



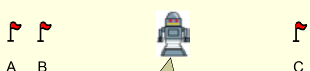
- Minimal team cost (above) is not achieved.
- The team cost resulting from parallel auctions is large because they cannot take synergies between targets into account.

Parallel Auctions: Synergies

Parallel Auctions: Synergies



Parallel Auctions: Synergies



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

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

Parallel Auctions: Positive Synergy

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)

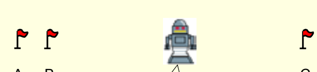
Parallel Auctions: Negative Synergy

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 Auctions: Positive and Negative Synergies



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

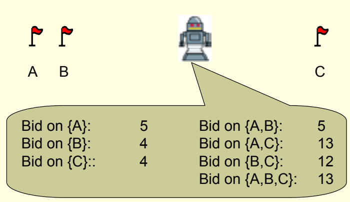
Parallel Auctions: Summary

- Ease of implementation: simple
- Ease of decentralization: simple
- Bid generation: cheap
- Bid communication: cheap
- Auction clearing: cheap
- Team performance: poor
  - no synergies taken into account

Ideal Combinatorial Auctions: Procedure

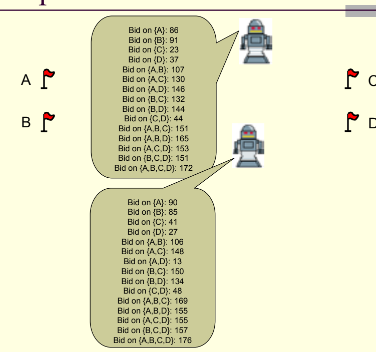
- Each robot bids on all bundles (= subsets) of targets.
- Each robot wins at most one bundle, so that the number of targets won by all robots is maximal and, with second priority, the sum of the bids of the bundles won by robots is as small as possible.
- Each robot determines a cost-minimal path to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

Ideal Combinatorial Auctions: Synergies

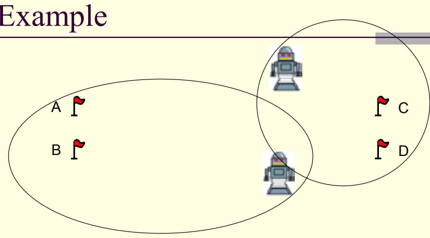


- Each robot bids on a bundle the minimal path cost it needs from its current location to visit all targets that the bundle contains.

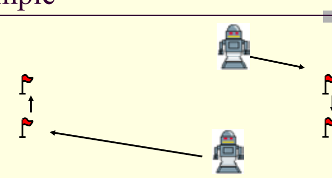
Ideal Combinatorial Auctions: Example



Ideal Combinatorial Auctions: Example



Ideal Combinatorial Auctions: Example



- 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).

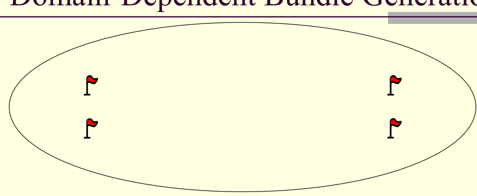
- THREE-COMBINATION
  - Bid on all bundles with 3 targets or less
- Note: It might be impossible to allocate all targets.

Combinatorial Auctions: Domain-Dependent Bundle Generation

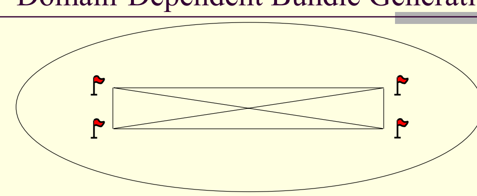
Smart bundle generation bids on clusters of targets.

- 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 sub graphs along (an approximation of) the maximal cut.
  - Recursively repeat the procedure twice, namely for the targets in each one of the two sub graphs.

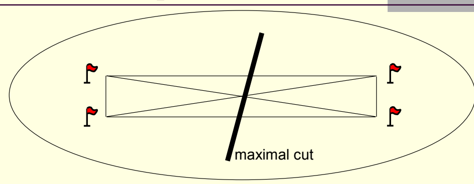
Combinatorial Auctions: Domain-Dependent Bundle Generation



Combinatorial Auctions: Domain-Dependent Bundle Generation



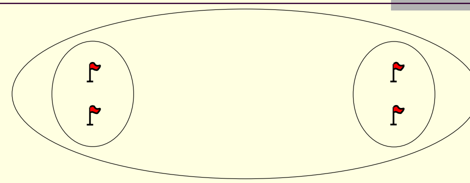
Combinatorial Auctions: Domain-Dependent Bundle Generation



- 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.

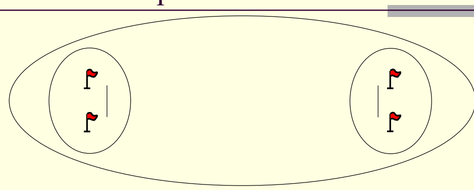
97

Combinatorial Auctions: Domain-Dependent Bundle Generation



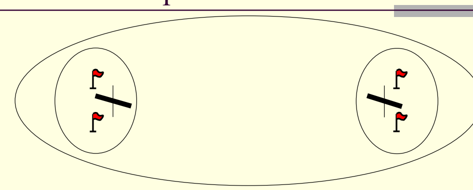
98

Combinatorial Auctions: Domain-Dependent Bundle Generation



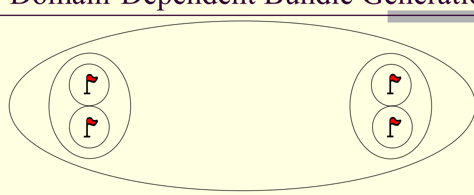
99

Combinatorial Auctions: Domain-Dependent Bundle Generation



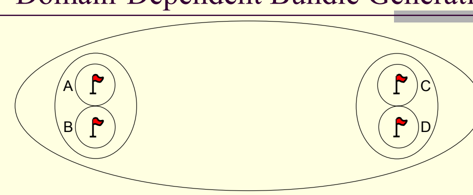
100

Combinatorial Auctions: Domain-Dependent Bundle Generation



101

Combinatorial Auctions: Domain-Dependent Bundle Generation



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

102

Combinatorial Auctions: Experiments in Known Terrain

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

	number of bids	SUM
parallel single-item auctions	635.1	426.5
combinatorial auctions with THREE-COMBINATION	20506.5	247.9
combinatorial auctions with GRAPH-CUT	1112.1	184.1
optimal (MIP) = ideal combinatorial auctions	N/A	184.4 (due to discretization issues)

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
- Auction clearing: expensive (NP-hard)
- Team performance: very good (optimal)
  - many (all) synergies taken into account

- Use a smart bundle generation method.
- Approximate the various NP-hard problems.

Sequential Auctions: Procedure

Parallel Auctions

- Ease of implementation: simple
- Ease of decentralization: simple
- Bid generation: cheap
- Bid communication: cheap
- Auction clearing: cheap
- Team performance: poor

Combinatorial Auctions

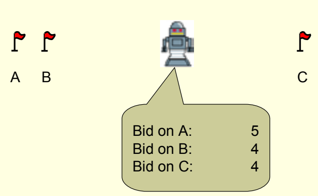
- Ease of implementation: difficult
- East of decentralization: unclear
- Bid generation: expensive
- Bid communication: expensive
- Auction clearing: expensive
- Team performance: "optimal"

Sequential auctions provide a good trade-off between parallel auctions and combinatorial auctions.

Sequential Auctions: Procedure

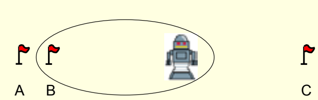
- There are several bidding rounds until all targets have been won by robots. Only one target is won in each round.
- During each round, each robot bids on all targets not yet won by any robot. The minimum bid over all robots and targets wins. (The corresponding robot wins the corresponding target.)
- 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 Auctions: Synergy



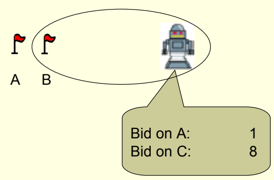
- Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BidSumPath). We give more details on this bidding rule later.

Sequential Auctions: Synergy



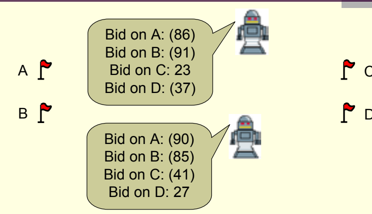
- Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BidSumPath). We give more details on this bidding rule later.

Sequential Auctions: Synergy

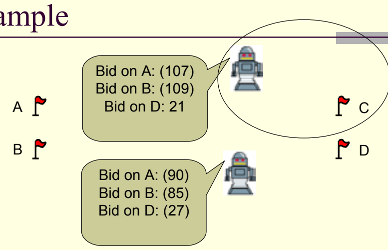


- Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BidSumPath). We give more details on this bidding rule later.

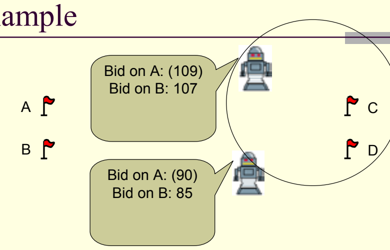
Sequential Auctions: Example



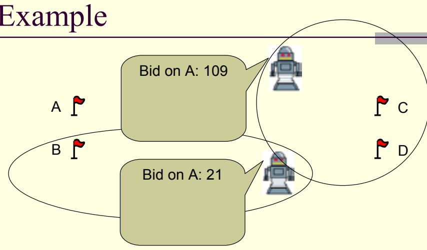
Sequential Auctions: Example



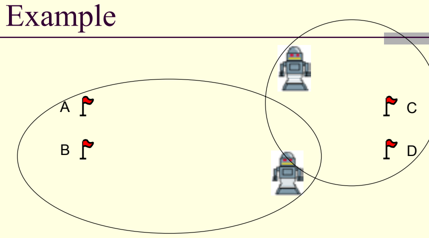
Sequential Auctions: Example



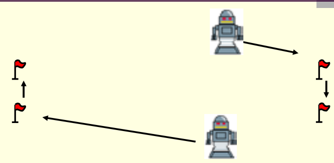
Sequential Auctions: Example



Sequential Auctions: Example



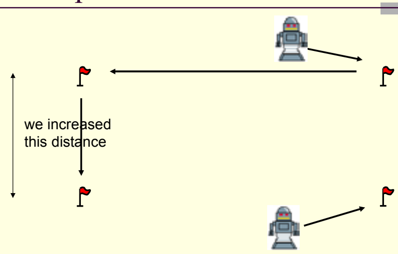
Sequential Auctions: Example



Sequential 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 auctions, and bid communication is cheap.
- The bids that do not need to be submitted were shown in parentheses in the example.

Sequential Auctions: Example



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

Sequential Auctions: Summary

- Ease of implementation: relatively simple
- Ease of decentralization: simple
- Bid generation: cheap (to be discussed later)
- Bid communication: cheap
- Auction clearing: cheap
- Team performance: very good
  - some synergies taken into account

Sequential Auctions: Derivation of Bidding Rules

- We suggest to use hill climbing to automatically derive bidding rules for sequential 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 Auctions: Derivation of Bidding Rules

- Path bidding rules ("direct approach")
  - Find paths directly
  - Will be explained in this tutorial
- Tree bidding rules ("indirect approach")
  - Find trees and convert them to paths
  - Similar, will not be explained in this tutorial

Sequential 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 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 Auctions: Derivation of Path Bidding Rules

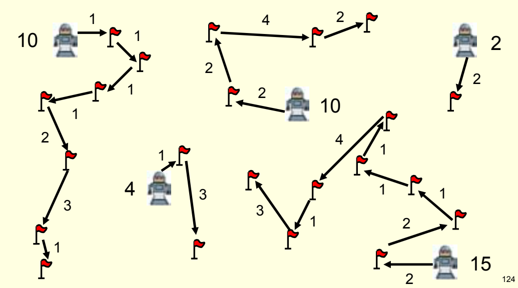
- We now show that robots can implement the resulting bidding rules in form of a sequential auction without having to know which targets the other robots have won already.

Sequential Auctions: Derivation of Path Bidding Rules

- **MiniSum**
  - Minimize the sum of the path costs over all robots
  - Minimization of total energy or distance
  - Application: planetary surface exploration
- **MiniMax**
  - Minimize the maximum path cost over all robots
  - Minimization of total completion time (makespan)
  - Application: facility surveilance, mine clearing
- **MiniAve**
  - Minimize the average arrival time over all targets
  - Minimization of average service time (flowtime)
  - Application: search and rescue

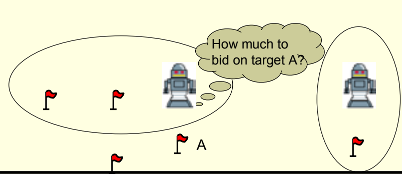
A Typical Coordination Task: MiniSum Team Objective

10+10+2+4+15 = 41



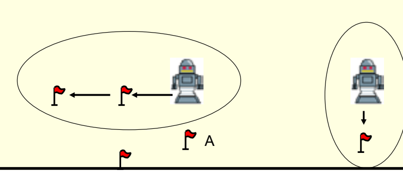
Sequential Auctions: Derivation of Path Bidding Rules

- MiniSum = energy or distance



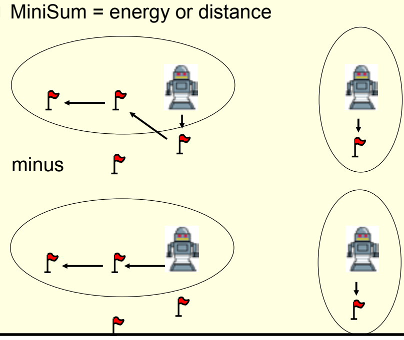
Sequential Auctions: Derivation of Path Bidding Rules

- MiniSum = energy or distance



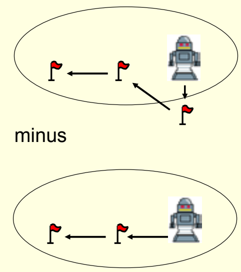
Sequential Auctions: Derivation of Path Bidding Rules

- MiniSum = energy or distance



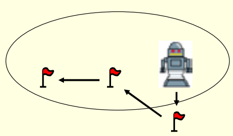
Sequential Auctions: Derivation of Path Bidding Rules

- MiniSum = energy or distance



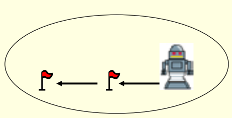
Sequential Auctions: Derivation of Path Bidding Rules

- 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

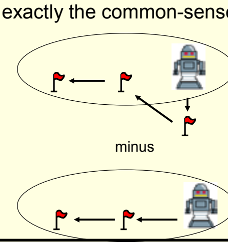
minus minus



minimal path cost the robot needs from its current location to visit all targets it has won so far

Sequential 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.

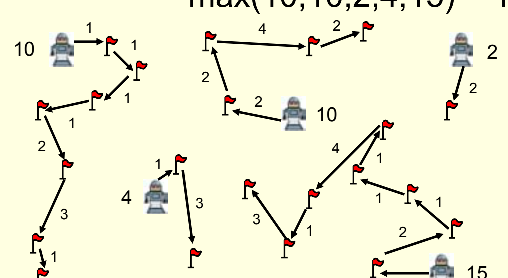


Sequential Auctions: Derivation of Path Bidding Rules

- **MiniSum**
  - Minimize the sum of the path costs over all robots
  - Minimization of total energy or distance
  - Application: planetary surface exploration
- **MiniMax**
  - Minimize the maximum path cost over all robots
  - Minimization of total completion time (makespan)
  - Application: facility surveilance, mine clearing
- **MiniAve**
  - Minimize the average arrival time over all targets
  - Minimization of average service time (flowtime)
  - Application: search and rescue

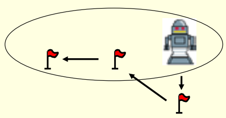
A Typical Coordination Task: MiniMax Team Objective

max(10,10,2,4,15) = 15



Sequential Auctions: Derivation of Path Bidding Rules

- 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.



133

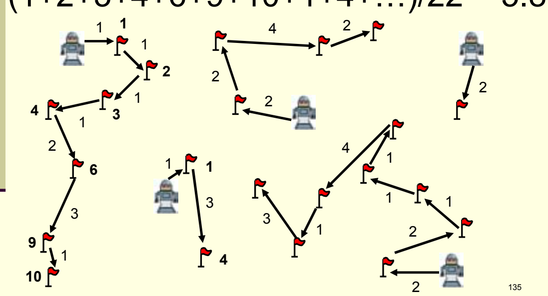
Sequential Auctions: Derivation of Path Bidding Rules

- **MiniSum**
  - Minimize the sum of the path costs over all robots
  - Minimization of total energy or distance
  - Application: planetary surface exploration
- **MiniMax**
  - Minimize the maximum path cost over all robots
  - Minimization of total completion time (makespan)
  - Application: facility surveilance, mine clearing
- **MiniAve**
  - Minimize the average arrival time over all targets
  - Minimization of average service time (flowtime)
  - Application: search and rescue

134

A Typical Coordination Task: MiniAve Team Objective

(1+2+3+4+6+9+10+1+4+…)/22 = 5.8



135

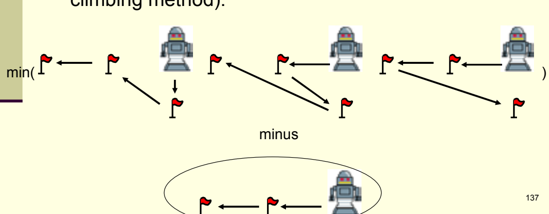
Sequential Auctions: Derivation of Path Bidding Rules

- 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).

136

Sequential Auctions: Derivation of Path Bidding Rules

- 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).



137

Sequential Auctions: Comparison of Bidding Rules

- **BidSumPath, BidMaxPath, BidAvePath**
  - Computation: local
  - Optimal bids: NP-hard
  - *Convention*: simple TSP insertion heuristic
  - Optimal conversion: none
- **BidSumTree, BidMaxTree, BidAveTree**
  - Computation: local
  - Optimal bids: polynomial
  - Optimal conversion: NP-hard
  - *Convention*: simple MST heuristic

138

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

Complexity of Auction Mechanisms

- Time complexity (amount of computation)
  - bid valuation in a single auction
  - winner determination in a single auction
  - number of auctions required to sell all tasks
- Communication complexity (message bandwidth)
  - call for bids
  - bid submission
  - awarding tasks to winners
    - may or may not inform losers in addition to winners
- Solution Quality (team cost)

Time Complexity

Auction type	Bid valuation	Winner determination	Number of auctions
Single-item	v	$O(r)$	n
Multi-item (greedy)	$O(n \cdot v)$	$O(n \cdot r \cdot m)$	$\lceil n/m \rceil$
Multi-item (optimal)	$O(n \cdot v)$	$O(r \cdot n^2)$	$\lceil n/m \rceil$
Combinatorial	$O(2^n \cdot V)$	$O((b+n)^n)$	1

n = # of items
r = # of bidders
b = # of submitted bid bundles (combinatorial auctions)
m = max # of awards per auction (multi-item auctions), $1 \le m \le r$
v / V = time required for item/bundle valuation (domain dependent)

* - [Gerkey and Matarić IJRR 23(9), 2004]
** - [Sandholm, Artificial Intelligence 135(1), 2002]

Communication Complexity

= worst-case message bandwidth

Auction type	Auction call	Bid submission	Award	Award (+ losers)
Single-item	$O(r)$	$O(r)$	$O(1)$	$O(r)$
Multi-item	$O(r \cdot n)$	$O(r \cdot n)$	$O(m)$	$O(r)$
Combinatorial	$O(r \cdot n)$	$O(r \cdot 2^n)$	$O(n)$	$O(r+n)$

n = # of items
r = # of bidders
m = max # of awards per auction (multi-item auctions), $1 \le m \le r$

"winners" = auctioneer only informs the winners of auctions
"winners + losers" = auctioneer also informs the losers that they've lost

Multi-Robot Routing: Optimal Solutions through MIP

- Use of Mixed Integer Programming (MIP) and CPLEX to solve multi-robot routing problems optimally for MiniSum, MiniMax, and MiniAve

Index sets and constants:

V_R = Set of robot vertices
V_T = Set of target vertices
c(i,j) = Path cost from vertex i to vertex j

Variables:

x_{ij} = Is vertex j visited by some robot directly after vertex i? (1 = yes, 0 = no)

Multi-Robot Routing: Optimal MiniSum Solution

Minimize

$$\sum_{i \in V_T \cup V_R, j \in V_T} c(i,j) x_{ij}$$

subject to

$$\sum_{i \in V_T \cup V_R} x_{ij} = 1 \quad \forall j \in V_T \qquad \text{(C1)}$$

$$\sum_{j \in V_T} x_{ij} \le 1 \quad \forall i \in V_T \cup V_R \qquad \text{(C2)}$$

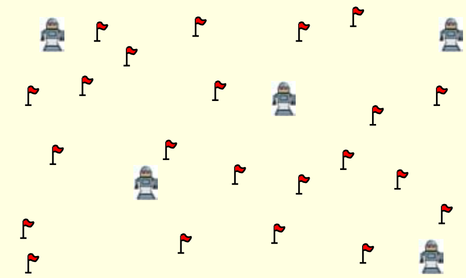
$$\sum_{i,j \in U} x_{ij} \le |U| - 1 \quad \forall U \subseteq V_T : |U| \ge 2 \qquad \text{(C3)}$$

Multi-Robot Routing: MIP Constraints

- Constraints (C1)
  - Each target vertex is entered exactly once
- Constraints (C2)
  - Each (robot or target) vertex is left at most once
- Constraints (C3)
  - There are no subtours (= cycles)

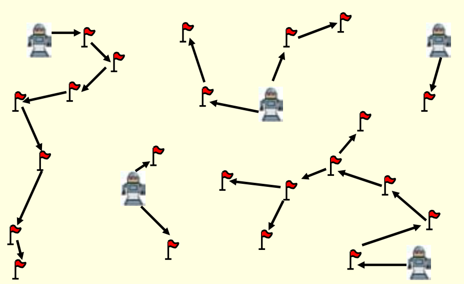
Multi-Robot Routing: Optimal MiniSum Solution

- Objective only



Multi-Robot Routing: Optimal MiniSum Solution

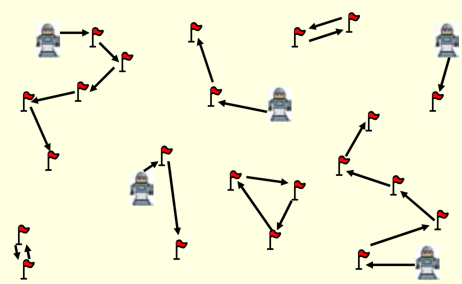
- Objective and constraint C1 only



(a possible solution, not necessarily the optimal one)

Multi-Robot Routing: Optimal MiniSum Solution

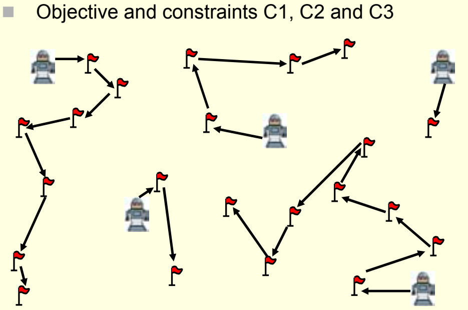
- Objective and constraints C1 and C2 only



(a possible solution, not necessarily the optimal one)

Multi-Robot Routing: Optimal MiniSum Solution

- Objective and constraints C1, C2 and C3



(a possible solution, not necessarily the optimal one)

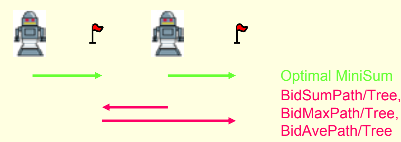
Multi-Robot Routing: Limitations of the MIP formulation

- 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).

Multi-Robot Routing: Hardness of Optimal Solutions

- Task allocation in general is NP-hard
- Only small multi-robot routing problems can be solved optimally since MiniSum, MiniMax, MiniAve are NP-hard even if the terrain is completely known. The reduction is from Hamiltonian Path.
- Multi-robot routing problems resemble vehicle routing problems, which are notoriously harder than TSPs.
- We cannot hope to minimize the team cost of realistic multi-robot routing problems in realistic running times.
- We hope for a small, possibly suboptimal team costs (for example, within a constant factor from optimal).

Sequential Auctions: Suboptimal Team Performance



- BidSumPath/Tree ≥ factor 1.5 away from MiniSum
- BidMaxPath/Tree ≥ factor 3 away from MiniMax
- BidAvePath/Tree ≥ factor 2 away from MiniAve

What is the best possible and the best known of the worst case?

Sequential Auctions: Theoretical Analysis

- **3 team objectives for multi-robot routing**
  - MiniSum, MiniMax, MiniAve
- **6 bidding rules for multi-robot routing**
  - 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
- **18 lower and upper bounds on team performance**
  - worst-case cost ratio
  - compared to optimal cost
  - first theoretical guarantees for auction-based coordination

Sequential Auctions: Analytical Results

cost ratio = $\dfrac{\text{team cost resulting from bidding rule}}{\text{minimum team cost}}$

Bidding Rule	Team Objective MiniSum		MiniMax		MiniAve	
	Lower	Upper	Lower	Upper	Lower	Upper
BidSumPath	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxPath	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAvePath	m	$2m^2$	$\frac{n+1}{2}$	$2m^2n$	$\Omega(m^{1/3})$	$2m^2$
BidSumTree	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxTree	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAveTree	m	$2m$	$\frac{n+1}{2}$	$2mn$	$\Omega(m^{1/3})$	$2m^2$

n robots and m targets

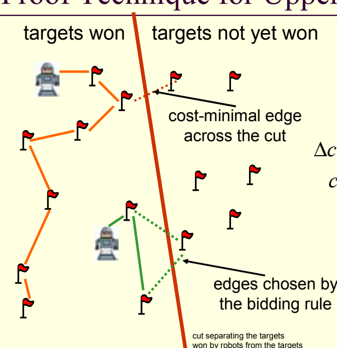
Sequential Auctions: Analytical Results

cost ratio = $\dfrac{\text{team cost resulting from bidding rule}}{\text{minimum team cost}}$

Bidding Rule	Team Objective MiniSum		MiniMax		MiniAve	
	Lower	Upper	Lower	Upper	Lower	Upper
BidSumPath	$\boxed{1.5}$	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxPath	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAvePath	m	$2m^2$	$\frac{n+1}{2}$	$2m^2n$	$\Omega(m^{1/3})$	$2m^2$
BidSumTree	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxTree	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAveTree	m	$2m$	$\frac{n+1}{2}$	$2mn$	$\Omega(m^{1/3})$	$2m^2$

n robots and m targets

Sequential Auctions: Proof Technique for Upper Bounds



BidSumPath

$$\Delta c(S) \le \alpha c^*$$
$$c(S) \le \alpha \sum c^*$$
$$\le \alpha c(\text{MSF})$$
$$\le \alpha c(\text{Optimum})$$

Sequential Auctions: Analytical Results

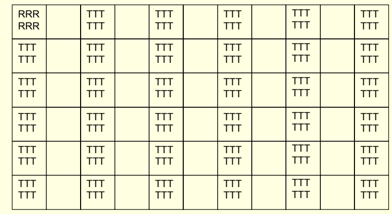
cost ratio = $\frac{\text{team cost resulting from bidding rule}}{\text{minimum team cost}}$

Bidding Rule	MiniSum Lower	MiniSum Upper	MiniMax Lower	MiniMax Upper	MiniAve Lower	MiniAve Upper
BidSumPath	1.5	$\boxed{2}$	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxPath	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAvePath	m	$2m^2$	$\frac{n+1}{2}$	$2m^2n$	$\Omega(m^{1/3})$	$2m^2$
BidSumTree	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxTree	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAveTree	m	$2m$	$\frac{n+1}{2}$	$2mn$	$\Omega(m^{1/3})$	$2m^2$

n robots and m targets

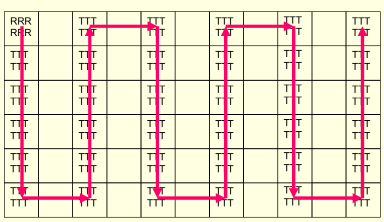
Sequential Auctions: Proof Technique for Lower Bounds

- Constant factor guarantees do not exist for BidMaxPath/Tree and BidAvePath/Tree



Sequential Auctions: Proof Technique for Lower Bounds

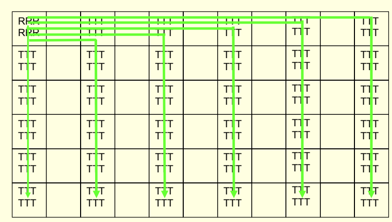
- Constant factor guarantees do not exist for BidMaxPath/Tree and BidAvePath/Tree



paths resulting from BidMaxPath

Sequential Auctions: Proof Technique for Lower Bounds

- Constant factor guarantees do not exist for BidMaxPath/Tree and BidAvePath/Tree



paths with small team cost

Sequential Auctions: Analytical Results

cost ratio = $\frac{\text{team cost resulting from bidding rule}}{\text{minimum team cost}}$

Bidding Rule	MiniSum Lower	MiniSum Upper	MiniMax Lower	MiniMax Upper	MiniAve Lower	MiniAve Upper
BidSumPath	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxPath	n	$2n$	$\frac{n+1}{2}$	$\boxed{2n}$	$\Omega(m^{1/3})$	$2m$
BidAvePath	m	$2m^2$	$\frac{n+1}{2}$	$2m^2n$	$\Omega(m^{1/3})$	$\boxed{2m^2}$
BidSumTree	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxTree	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAveTree	m	$2m$	$\frac{n+1}{2}$	$2mn$	$\Omega(m^{1/3})$	$2m^2$

n robots and m targets

Sequential Auctions: Observations

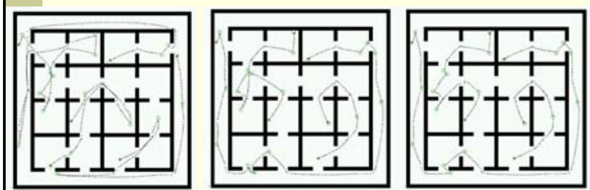
- **Looking at team objectives**
  - Best guarantees offered for MiniSum
  - MiniSum: constant-factor (2) approximation
  - MiniMax: linear in the number of robots
  - MiniMax: linear in the number of targets
- **Looking at bidding rules**
  - Best guarantees given by BidSumPath, BidSumTree
  - Each rule is best for the corresponding objective
  - Exception: BidAvePath, BidAveTree

Sequential Auctions: Experimental Evidence

- Experimental Performance
  - Bounds = extreme cases
  - Experiments = average cases
  - Bidding rules perform better in practice
- Experimental Bounds
  - Much smaller than the theoretical worst-case
  - Within a factor of 1.4 in most cases
- Time Complexity
  - Path rules are more expensive
  - Tree rules are more efficient
  - Path rules result in somewhat better performance

Sequential Auctions: Experimental Comparison

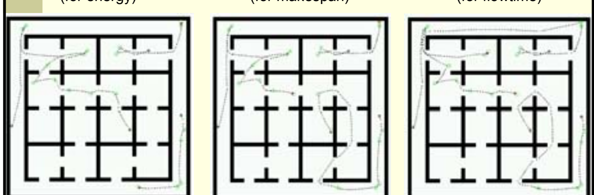
parallel auctions | sequential auctions | optimal (MIP) = ideal combinatorial auctions



SUM = 426.98 | SUM = 279.62 | SUM = 271.04

Sequential Auctions: Appropriateness of Bidding Rules

BidSumPath (for energy) | BidMaxPath (for makespan) | BidAvePath (for flowtime)



SUM = 182.50, MAX = 113.36, AVE = 48.61 | SUM = 218.12, MAX = 93.87, AVE = 46.01 | SUM = 269.27, MAX = 109.39, AVE = 45.15

pictures are from USC's Player/Stage robot simulator

Sequential Auctions: Results for Path Bidding Rules

- 2 robots and 10 unclustered targets
- known terrain of size 51×51

	SUM	MAX	AVE
BidSumPath	193.50	168.50	79.21
BidMaxPath	219.15	125.84	61.39
BidAvePath	219.16	128.45	59.12
optimal (MIP) = ideal combinatorial auctions	189.15	109.34	55.45

Sequential Auctions: Results for Path Bidding Rules

- 2 robots and 10 clustered targets
- known terrain of size 51×51

	SUM	MAX	AVE
BidSumPath	134.18	97.17	62.47
BidMaxPath	144.84	90.10	57.38
BidAvePath	157.29	100.56	49.15
optimal (MIP) = ideal combinatorial auctions	132.06	85.86	47.63

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

Outline

- What are the practical issues that we encounter when implementing market-based coordination on a team of robots?
- We will focus on:
  - Dynamic environments
  - Robustness to failures
  - Uncertainty

Market-Based Robot Implementations

- Several domains:
  Distributed sensing, Mapping, Exploration, Surveillance, Perimeter Sweeping, Assembly, Box Pushing, Reconnaissance, Soccer, and Treasure Hunt
- Some approaches have been demonstrated on multiple domains:
  TraderBots and *MURDOCH*
- A variety of cost/reward models, bidding strategies, and auction-clearing mechanisms are used
- No clear guidelines for how to pick the best approach for a given domain or application

Deciding which approach to use

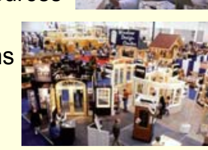
- Some comparative studies: Gerkey and Matarić, Dias and Stentz, and Rabideau et al.
- Market-based approaches do well in these comparative studies
- Different application requirements and tradeoffs in implementation make it difficult to construct a single market-based approach that can be successful in all domains
- A well-designed market-based approach with sufficient plug-and-play options for altering different tradeoffs can be successful in a wide range of applications

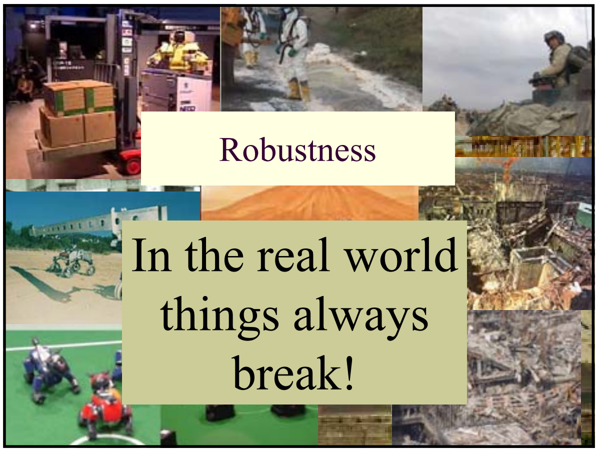
Some considerations when designing your coordination approach

- How dynamic is your environment?
- What are your requirements for robustness?
- How reliable is your information?
- How will you balance scalability vs. solution quality?
- What type of information will you have access to?
- What resources/capabilities does your team possess?
- What do you want to optimize?
- How often will your mission/tasks change?
- What guarantees do you require?

Characteristics of dynamic environments

- Unreliable/incomplete information
- Changing/moving obstacles
- Changing task requirements
- Changing limited resources and capabilities
- Evolving ad-hoc teams





Robustness

In the real world things always break!

Generally a team is robust if it can …

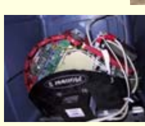
- Operate in dynamic environments
- Provide a basic level of capability without dependence on communication, but improve performance if communication is possible
- Respond to new tasks, modified tasks, or deleted tasks during execution
- Survive loss (or malfunction) of one or more team members and continue to operate efficiently

Categories of Failure

- Communication Failure

- Partial Robot Malfunction

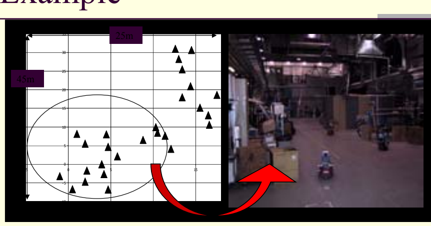
- Robot Death



Dealing with communication failures

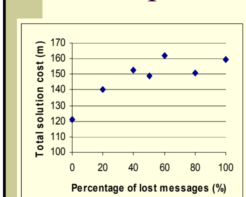
- Acknowledgements can help ensure task completion but delay task allocation
- Tradeoff between repeated tasks and incomplete tasks
- Message loss often results in loss in solution quality

Example



- Nominal case: 23 goals assigned
- Note: Some assigned tasks may not be completed due to dynamic conditions

Example



	Cost (m)		Tasks Completed (#) (23 assigned)	
Description	Mean	+/-	Mean	+/-
Nominal	121	12	21.0	2.0
20% msg. loss	140	5	24.0	0.3
40% msg. loss	153	3	24.7	2.0
50% msg. loss	149	10	24.0	0.7
60% msg. loss	162	9	25.3	0.7
80% msg. loss	151	3	22.3	0.7
100% msg. loss	159	5	21.0	2.0

- Acknowledgements help ensure task completion
- Repeated tasks vs. incomplete tasks
- Message loss results in loss of efficiency but tasks are completed if resources permit

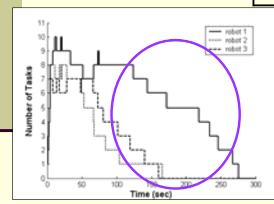
Dealing with partial malfunctions

- Identifying the malfunction may be done as an individual or as a team
- Key advantage is that malfunctioning teammate can re-auction tasks it cannot complete
- If complete failure (robot death) is anticipated, a quicker allocation method should be chosen
- Possible new tasks can be generated to enable recovery from malfunction
- Malfunctions often results in loss in solution quality

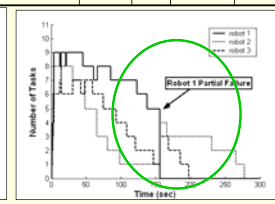
Example

- Laser failure or gyro error is detected
- Robot greedily auctions all its tasks to other robots

	Cost (m)		Tasks Completed (#) (23 assigned)	
Description	Mean	+/-	Mean	+/-
Nominal	121	12	21.0	2.0
Partial Failure	140	5	22.0	1.0



Nominal Performance

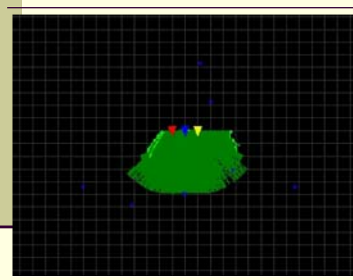


Partial Malfunction

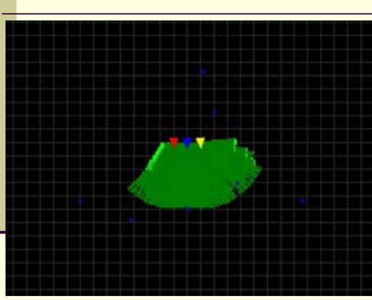
Dealing with robot death

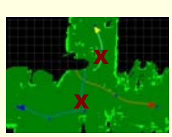
- Detecting the death must be done by the team
- Can detect potential deaths by keeping track of communication links
- Need to seek confirmation of suspected deaths
- Need to query other robots about tasks assigned to dead robot(s) and repair subcontract links
- If no new contract can be made, the owner of the task must complete it

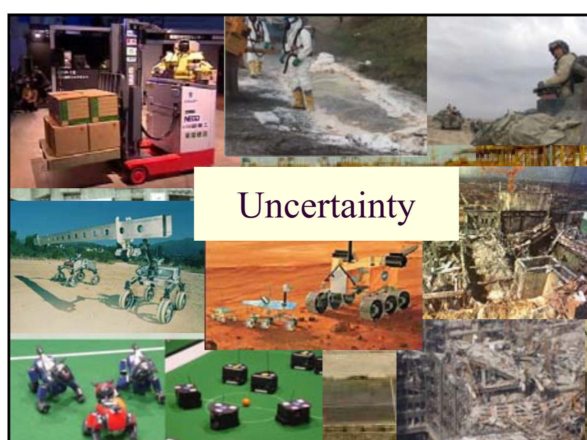
Example



Example





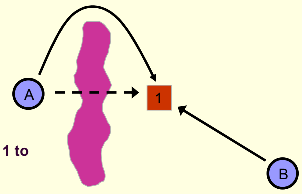


Uncertainty

Uncertain and changing environments

- Robots discover that a task cannot be executed for the bid cost
- Robots auction the task to another robot, default, or execute at a loss (learning to estimate better in the future)

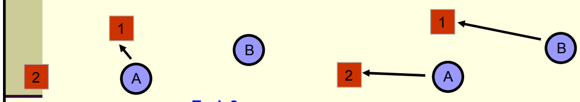
Robot A encounters obstacle, making Task 1 more costly than expected



Robot A sells Task 1 to Robot B

New, deleted, and changing tasks

- New tasks trigger new auction rounds
- Tasks can be re-prioritized by changing revenue function
- Tasks can be deleted – compensation may be necessary
- Subcontracting can help deal with changing situation

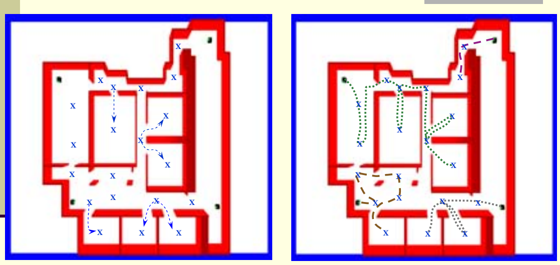


Robot A is committed to execute Task 1

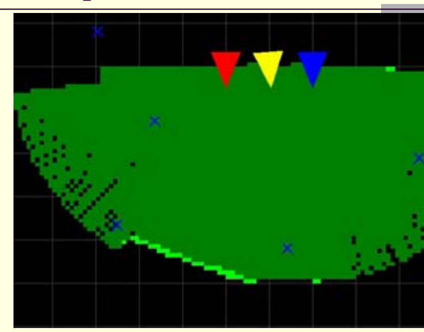
Task 2 appears which is worth 10X revenue, but Tasks 1 and 2 must be executed exclusively

Robot A sells Task 1 to B so that it can purchase Task 2—even though B requires higher cost than A to execute Task 1

Example: Imperfect information



Example: Unknown world



Open Challenges

- Benchmarks for effective comparisons of coordination approaches
- Detailed guidelines for designing a market-based coordination approach for a given application domain
- Improved robustness (efficient detection of failures and cooperative recovery strategies)
- Effective information-sharing using market-based approaches
- Demonstrated coordination of large teams using market-based approach
- Demonstrated effective learning applied to market-based coordination of teams
- Varied and rigorous testing in a variety of application domains

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

Outline

- Where do typical multirobot planning issues arise in market-based systems? How are they incorporated into the market framework?
- Task Allocation
  - who does each task?
- Complex Task Allocation and Decomposition
  - who does the task, and how is the task achieved?
- Tight Coordination
  - how to accomplish joint tasks that may require close interaction?

Task Allocation

Task Allocation

- How is the general problem different from previous multirobot routing example?
  - Agents may have different cost functions
  - There may be constraints between tasks
  - Tasks may be distributed across agents and may need to be reallocated
  - Agents may need to form subteams to complete some tasks
  - We may be dealing with roles (allocated for an indeterminate amount of time)
  - The environment may be extremely unknown or dynamic

Task Allocation Definition #1

- Given
  - a set of tasks, T
  - a set of agents, A
  - a cost function c_i: $2^T \rightarrow \mathbf{R} \cup \{\infty\}$ (states the cost agent i incurs by handling a subset of tasks)
  - an initial allocation of tasks among agents $< T_1^{init}, \ldots, T_{|A|}^{init} >$, where $\cup T_i^{init} = T$ and $T_i^{init} \cap T_j^{init}$ for all $i \neq j$
- Find
  - the allocation $< T_1, \ldots, T_{|A|} >$ that minimizes $\sum c_i(T_i)$

[T. Sandholm, *Contract Types for Satisficing Task Allocation: I Theoretical Results*, AAAI Spring Symposium, 1998]

- Extended from "Task Oriented Domains"
  - here, cost function is assumed to be symmetric and finite

[Rosenschein and Zlotkin, *A Domain Theory for Task Oriented Negotiation*, IJCAI, 1993]

Task Allocation Definition #2

- Given
  - a set of tasks, T
  - a set of robots, R
  - $\Re = 2^R$ is the set of all possible robot subteams
  - a cost function c_r:$2^T \rightarrow \mathbf{R}^+ \cup \{\infty\}$ (states the cost subteam r incurs by handling a subset of tasks)
- Then
  - an allocation is a function A:$T \rightarrow \Re$ mapping each task to a subset of robots

  or, equivalently
  - $\Re^T$ is the set of all possible allocations
- Find
  - the allocation $A^* \in \Re^T$ that minimizes a global objective function C: $\Re^T \rightarrow \mathbf{R}^+ \cup \{\infty\}$

[Dias, Zlot, Kalra, Stentz, *Market-based Multirobot Coordination: A Survey and Analysis*, Proceedings of the IEEE Special Issue on Multi-robot Systems, 2006]

What's missing?

- Tasks T and robots R may be changing over time
  - Can represent as T(t) and R(t)
- Robots can only be in one subteam
  - Cost function of a subteam can change if one or more members are performing other tasks individually or as part of other subteams

A taxonomy

- Single-task robots (ST) vs multi-task robots (MT)
  - ST: each robot is capable of handling only one task at a time
  - MT: robots can execute multiple tasks simultaneously
- Single-robot tasks (SR) vs multi-robot tasks (MR)
  - SR: Each task requires exactly one robot
  - MR: Tasks may require more than one robot
- Instantaneous assignment (IA) vs time-extended assignment (TA)
  - IA: Available information on tasks/robots/environment permits only an instantaneous allocation of tasks to robots and no planning for future allocations
  - TA: More information is available (*e.g.* a full list of tasks, or a model of how they will arrive) and robots can plan into the future (*e.g.* can maintain schedules or task sequences)

[Gerkey and Matarić, *A Formal Analysis and Taxonomy of Task Allocation in Multi-robot Systems*, IJRR, 23(9), 2004]

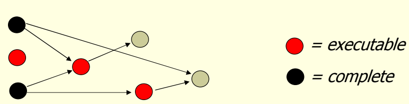
Example: MURDOCH

- Multirobot box-pushing and loosely-coupled tasks
  - Box pushing: one watcher, two pushers
  - Loosely-coupled: tracking, monitoring, cleanup
- Single task auctions: each task is auctioned when introduced, *available* robots bid, task awarded
  - *Available* robots: have not committed to any other tasks
  - Heterogeneous robots: participation by resource-centric publish/subscribe protocol
- ST-SR-IA (with online tasks)
- Solution quality: 3-competitive (utility maximization only)

[Gerkey and Matarić, IEEE Trans. R&A 2002 / IJRR 2004]

Example: M+

- Load transfer, hospital servicing
  - task precedence constraints
- Negotiation protocol - distributed auction
  - *Available* robots announce bids for *executable* tasks (those with precedence constraints satisfied)
  - Robot with the lowest cost awarded the task, although it can transfer to another robot with a lower cost before execution
  - one-task lookahead



- SR-ST-TA*

[Botelho and Alami, ICRA 1999]

Example: *TraderBots*

- Distributed sensing, exploration, area reconnaissance, treasure hunt
- SR-ST-TA
  - Task scheduling and sequencing (unlimited lookahead)
- 1) Multi-task auctions (*OpTraders)*
  - Greedy clearing algorithm: 2-approximation (one-shot, no iteration)
    - Optimal clearing algorithm possible in polynomial time
  - MAPA - maximum number of awards per auction
    - Increasing MAPA $\rightarrow$ poorer solution quality but faster allocation [Dias *et al.*, i-SAIRAS 03]

TraderBots (cont'd)

- 2) Distributed / peer-to-peer auctions (*RoboTraders)*
  - Multi-task auctions with MAPA = 1
  - Anytime / local search algorithm
  - Task reallocation for unknown / dynamic environments
  - Optimal solution guaranteed in a finite number of trades with a sufficiently expressive set of contract types [Sandholm, AAAI Spring Symp. 98]
    - Single-task; Multi-task; Swap; Multi-party (OCSM)
  - In a limited number of rounds, combinations of single- and multi-task contracts performed best [Andersson and Sandholm, ICDCS 00]
  - Allowing non-individual rational trades can lead to better solutions [Vidal, AAMAS 02]
  - Other P2P-trading examples: TRACONET [Sandholm, IWDAI 93], swap-based protocol [Golfarelli 97], UAV application [Lemaire, ICRA 02]

TraderBots (cont'd)

- 3) Leaders [Dias and Stentz, IROS 02]
  - Optimize allocations/plans within subgroups
    - "pockets" of centralized optimization
  - Example: leader collects task info from a subgroup; holds a combinatorial exchange; if a better solution is found, leader retains the surplus as profit

[Dias *et al.*, multiple publications 1999-2006]

Example: Multi-robot tasks (MR-ST-IA)

How to form coalitions / subteams?

- Robots must hire helpers to move found objects
  - Foraging [Guerrero and Oliver, CCIA 03]
- Auctioneer chooses subteam based on robot capabilities / costs
  - Subgroup accepts or rejects task
  - Furniture moving [Lin and Zheng, ICRA 05]
- Subteams agree upon “plays” before sending bid to auctioneer
  - Treasure hunt [Jones et al, ICRA 06]

Summary: Task Allocation

- Covered applications: box-pushing, distributed sensing, surveillance, load transfer, hospital servicing, foraging, furniture moving, treasure hunt
- Different mechanisms are used in different scenarios; choice depends on:
  - Quality/scalability tradeoff
  - Uncertainty / dynamicity of environment
  - Task constraints/duration
  - Ability to plan / replan
  - Required speed of allocation

Complex Task Allocation

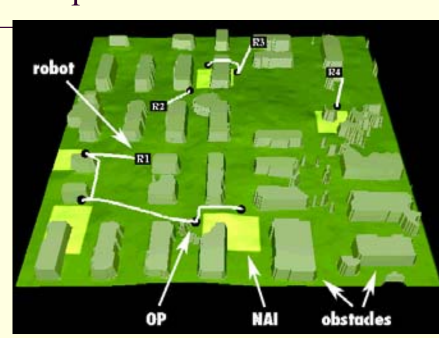
Complex Task Allocation

- What’s different from previous problems?
  - Tasks may be complex or abstract so subtasks that need to be allocated might not be specifically predefined

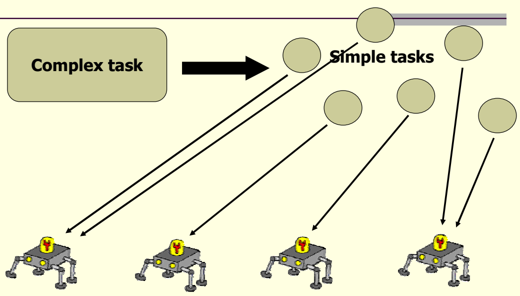
Complex Tasks

- *Simple* tasks can be executed in a straightforward, prescriptive manner (*e.g.* plan a path from point *A* to point *B*)
- *Complex* tasks
  - Tasks that have many potential solution strategies
  - Abstract description
  - Often involves solving an NP-hard problem
- We’ll focus on: complex tasks that can be decomposed into multiple inter-related subtasks

Example: Area Reconnaissance



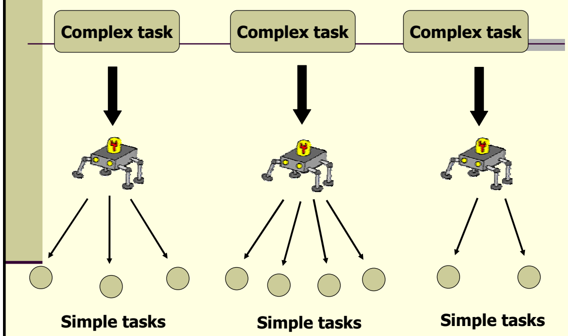
Complex Task Allocation



Problem: how can we know how to decompose the complex task(s) efficiently before we know which robots are going to be assigned the resulting simple tasks?

212

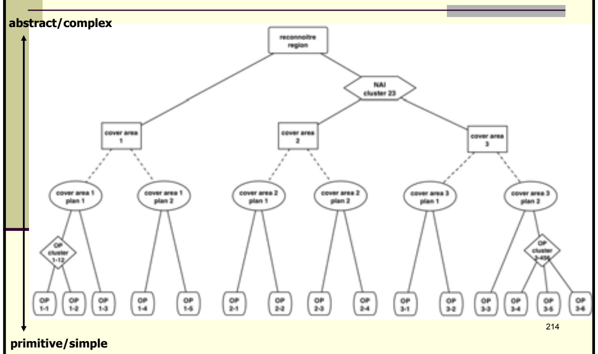
Complex Task Allocation



Problem: how can we know how to best allocate the complex tasks if we don't yet know how they will be decomposed?

213

Task Trees



214

Task Tree Auctions

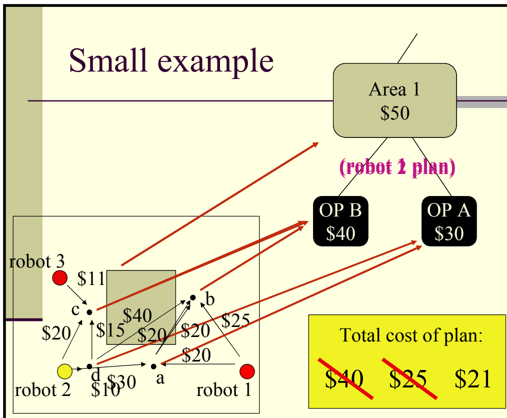
- *Task trees* are traded on the market
- Bids are placed for tasks at any level of a task tree
- First pass: bid on auctioneers plan (valuation)
- Second pass: redecompse abstract tasks (decomposition)

- Avoids premature commitment on allocation and decomposition decisions
- Mechanism enables:
  - Tasks can be reallocated or redecomposed
  - Robots can develop their own plans for complex tasks
  - Subtasks of a single complex task can be shared among multiple robots

[Zlot and Stentz, ICRA 2005 / IJRR 2006]

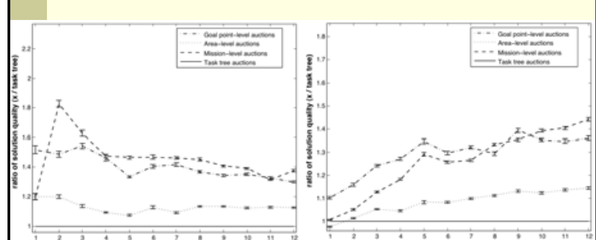
215

Small example



216

Comparison to Single-Level Simple Task Allocation



217

Field Experiments

Summary: Complex Task Allocation

- Application: area reconnaissance
- If tasks are complex, can incorporate task decomposition into the allocation mechanism
  - If agents have different preferences on the possible task decompositions, outcome can be made more efficient by coupling task allocation and decomposition

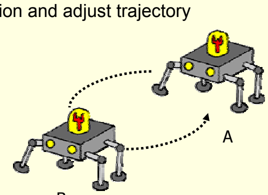
Tight Coordination

Loose v Tight Coordination

- Loose:
  - task can be completed by a single agent
  - task easily decomposed into discrete subtasks
  - teammates coordinate during decomposition, allocation but not during execution
  - Research Question: Who does which task?
  - e.g. exploration, Burgard et. al., ICRA 2000
- Tight:
  - task *requires* participation from multiple agents
  - task not easily decomposed into subtasks
  - teammates coordinate during all stages of task and continuously coordinate during execution
  - Research Question: Who does *what and how?*
  - e.g. box carrying, Caloud et. al., IROS 1990

Tight Coordination

- Informally, we say that robot A coordinates with robot B if it considers the state of B when choosing its own. This coordination is *tight* if A considers B's state at a high frequency throughout execution.
- Example: following a teammate: continuously observe B's position and adjust trajectory



Approach I:

- Achieve tight coordination indirectly through task allocation
- Role of Market: allocate IA tasks.
- Benefit: the auction provides a simple interface between robots
- Drawback: Limited applicability (to tasks where robots don't *need* to directly interact)

Box Pushing, Gerkey & Matarić, ICRA 2001

- Goal: move box to goal using “watcher” and 2 “pushers”
- IDEA: facilitate a form of indirect coordination by selecting new tasks according to success of previous actions
- Market-based Approach
  - continuously auction ‘push-right-side’ and ‘push-left-side’ tasks
  - tasks are very short lived
  - new task depends on success of previous task
- Observations
  - actions of one pusher certainly affects actions of other
  - pushers never interact directly, just via watcher & tasks
  - mission could be completed by single pusher & watcher

Exploration, Lemaire et. al., ICRA 2004

- Goal: traverse route while maintaining communication with base station
- IDEA: encode planning/coordination into tasks.
- Market-based Approach
  - simplify exploration task: fixed, known trajectory
  - simplify relay task: stay in fixed location for fixed duration
- Observations
  - actions of explorer determine task of relay robot
  - robots do not interact after allocation phase
  - Similar to Murdoch approach for box pushing
  - Limited approach to constrained exploration problem

Approach II:

- Achieve tight coordination using reactive approach
- Role of Market: allocate roles to robots.
- Benefit: reactive approaches can work very well for tight coordination
- Drawback: limited applicability (to tasks where interactions are simple)

Construction Simmons et. al. NRL, Wshp 2002

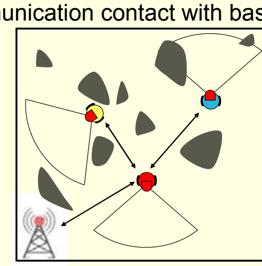
- Goal: dock a beam using a crane, roving eye, precise manipulator
- IDEA: hybrid approach - use auctions to assign tasks, achieve tight coordination with reactive approach. Similar to other MR tasks
- Market-based Approach
  - auction tasks such as “watch fiducials” and “push beam”
- Observations:
  - robots must interact closely on tight sense-act loop
  - achieved using simple reactive approach (simple interactions only)

Approach III:

- Achieve tight coordination by buying and selling joint plans online
- Role of Market: determine when joint plans are required, make contracts between teammates during execution
- Benefit: can handle complex tight coordination tasks
- Drawback: may be very complex

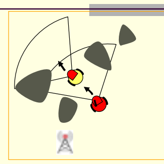
Constrained Exploration

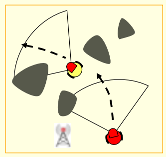
- Explore an environment while maintaining communication contact with base station



Complex Tight Coordination

- Tight coordination to ensure current constraints are met
- Extensive coordination of plans to ensure that future constraints are met
- Cannot be encoded as task allocation
- Too complex for reactive approach





Perimeter Sweeping, Exploration -

Kalra et. al., ICRA 2005

- Goal: perimeter sweeping & constrained exploration
- Q1: How do we decide what a robot should do if task is not decomposable into independent subtasks?
- IDEA 1: evaluate cost and revenue of *actions*
  - i.e. every action has cost and revenue, not just every task
  - this allows evaluation of action at fine granularity
  - and we no longer need to define problems as set of finite tasks
  - e.g. instead of profit(path-to-city-a), profit(path)

Kalra et. al. (cont)

- Q2: How do we incorporate constraints between robots into cost/revenue function?
- IDEA 2: couple cost and revenue between robots
  - i.e. profit of A's actions depends on B's simultaneous actions
  - e.g. if robot A loses comms with teammate B, both incur cost

Kalra et. al. (cont)

- Q3: How do we make this tractable?
- IDEA 3: decouple robots' planning whenever possible, auction joint plans when necessary
  - e.g. robots A & B frequently share their intended actions
  - each chooses its own trajectory while considering the other's expected trajectory
  - when constraint violation is expected, they propose and bid on joint plans that solve the constraints.
  - related to use of leaders/opportunistic centralization in TraderBots

Summary

- Choice approach depends on:
  - Type of tight coordination
    - Can it be encoded as a task allocation problem?
    - Is coordination simple enough to use a reactive approach?
  - Quality of solution desired
    - Are benefits of a complex approach "worth it"?

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Heterogeneous domains
- Conclusion

Section Outline

- Overview of heterogeneous Teams and the domains in which they operate
- Market-based allocation for heterogeneous teams
  - Special requirements for human-multirobot teams
- Conclusions

Heterogeneous Teams In Action

(1)


- Construction (1)
- Urban Search and Rescue
  - Real Robots (2)
  - Simulated (3)
- Planetary Exploration (4)
- Treasure Hunt (5)
- Robocup Segway League (6)

(2)


(3)


(4)


(6)


(1) F. Heger, L. Hiatt, B.P. Sellner, R. Simmons, and S. Singh. "Results in Sliding Autonomy for Multi-robot Spatial Assembly", Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, September, 2005.
(2) http://www.itl.nist.gov/iaui/vvrg/hri/IMAGESusar.html
(3) N. Schurr, J. Marecki, P. Scerri, J.P. Lewis and M. Tambe. "The DEFACTO System: Training Tool for Incident Commanders" Innovative Applications of Artificial Intelligence, 2005.
(4) J. Schneider, D. Apfelbaum, D. Bagnell, R. Simmons, "Learning Opportunity Costs in Multi-Robot Market Based Planners", International Conference on Robotics and Automation, 2005.
(5) E.G. Jones, B. Browning, M.B. Dias, B. Argall, M. Veloso, and A. Stentz, "Dynamically formed heterogeneous robot teams performing tightly-coupled tasks", to appear in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),* 2006.
(6) B. Argall, Y. Gu, B. Browning, and M. Veloso. The First Segway Soccer Experience: Towards Peer-to-Peer Human-Robot Teams. Carnegie Mellon University, 2005. Image from http://www.cs.cmu.edu/~coral-downloads/segway/images/ .

Heterogeneous Teams

- Members of team are equipped differently, have different skills, or play different roles.
- Why heterogeneous teams?
  - For complex missions, many specialists better than a few generalists
    - In TRESTLE, 3 different robots preferred to a single monolithic construction robot.
    - For USAR, robots need different form factors and sensing modalities
  - Specialists often easier to design than generalists.
  - Enabling coordinated heterogeneous teams means easier reuse across applications
    - TRESTLE "Roving Eye" broadly useful

Heterogeneous Teams

- How does a heterogeneous domain differ from multirobot routing?
  - Completing different tasks may now require using a number of different capabilities (instead of simply visiting a target).
  - Agents may have capabilities that make them better suited to address some tasks than others (instead of all agents being identical)
  - We now have to consider capabilities when forming bids and awarding auctions (instead of only considering a metric like cost)

Allocation for Heterogeneous Teams

- Allocation requires reasoning about different robots' capabilities.
- Markets well suited for allocation in these domains
  - Each bid can encapsulate a robot's ability to complete the task.
    - Robots need not bid if they can't do the task.
    - Individual robot needs only to be able to assess its own abilities and resources.
  - Auctioneer can award task only based on bids, not individual knowledge of individual capabilities.
- Valuation of different allocations difficult
  - For a visual inspection task should a very busy Binocular Roving-Eye bid lower or higher than an idle Pioneer with a web cam?

Human as Leader Example

- Human operator and a team of fire truck robots are tasked with extinguishing fires in a city
  - Goal of domain to prevent as much damage as possible to burning buildings
- Domain work flow:
  - Human operator discovers a fire
  - Operator generates a fire-fighting task parameterized with building location, magnitude of the fire, and estimated building value
  - Human sends task to autonomous dispatcher
  - Dispatcher determines which fire truck robot should attend to the fire

Human Perspective

- Human operator(s) trying to accomplish some task
- Operator generates tasks to address domain requirements
  - Task is fully parameterized
    - Description
    - Value function
- Task gets executed by some agent in the system
  - Operator does not care which agent completes the task
- Allocation solution for generated tasks should maximize over operator's preferences

Allocation Perspective

- Tasks periodically arrive in a stream
  - Rate of arrival may be governed by some distribution
- Tasks should be allocated to maximize some objective function
  - Some tasks more important in objective function
  - A task's value has a temporal component
    - Maximum value given for immediate completion
    - Value for completion degrades as a function of time
  - Objective function may have additional components
    - Cost of resources
    - Penalty for failure to complete allocated task by a deadline

Using Market-based Allocation

- Translate from objective value to market currency
  - Offer rewards offered for task completion
    - Maximum reward given for immediate completion
    - Reward decays, mirroring decay of task value in the objective function
- Self-interested agents attempt to accumulate as much reward as possible
- As tasks are issued by the operator, auction is conducted
- Allocation strategy awards task to highest positive bidder
  - If no agent has a positive bid, task goes unallocated

Incorporating human preferences

- Instantiating human preference in an objective function can be difficult
  - Literature scarce on this topic, but for interesting analysis see D. Wolpert, K. Tumer. "An Introduction to Collective Intelligence" *NASA tech rep NASA-ARC-IC-99-63*, 2000.
- Many interactions between objective function and solution quality
  - Success of allocation strategy contingent on many factors
    - System load
    - Types of tasks (values and rates of decay)
    - Learning capabilities of agents
- Can we somehow incorporate user feedback?
- What happens when the human is part of the team?

Conclusions

- Many interesting domains require interfacing humans with team of robots, or generally interfacing different types of agents with each other.
- If we can express human preference in an objective function, then we can construct a reasonable market-based allocation approach.
- Task valuation is difficult for domains with heterogeneous agents, especially with online tasks; learning valuations in such domains seems a fruitful research direction.

Structure of the Tutorial

- Overview
- Auctions in Economics
- Theory of Robot Coordination with Auctions
  - Auctions and task allocation
  - Analytical results
- Practice of Robot Coordination with Auctions
  - Implementations and practical issues
  - Planning for market-based teams
  - Human-multirobot domains
- Conclusion

Conclusions

- Auctions are indeed a promising means of coordinating teams of agents (including robots).
- In particular, auctions can be an effective and practical approach to multi-robot routing.
- There are lots of opportunities for further research on agent coordination with auctions.

Conclusions

- There is a workshop on Auction Mechanisms for Robot Coordination at AAAI 2006 that you might want to participate in!
- Additional material can be found at:
  - idm-lab.org/auction-tutorial.html (scroll to the bottom)
  - metropolis.cta.ri.cmu.edu/markets/wiki

Conclusions

- We thank the members of our research teams:
  - C. Casinghino, M. Dias, D. Ferguson, J. Gonzalez, E. Jones, N. Kalra, M. Sarnoff, K. Shaban, A. Stentz (group lead), L. Xu, M. Zinck, and R. Zlot.
  - M. Berhault, H. Huang, D. Kempe, S. Jain, P. Keskinocak (group lead), A. Kleywegt, S. Koenig (group lead), M. Lagoudakis (group lead), V. Markakis, C. Tovey, A. Meyerson and X. Zheng.
- We owe special thanks to:
  - www.itl.nist.gov/iaui/vvrg/hri/IMAGESusar.html

Conclusions

- We appreciate funding for this research from:
  - Army Research Laboratory (CMU)
  - The Boeing Company (CMU)
  - Defense Advanced Research Projects Agency (CMU)
  - Jet Propulsion Laboratory (USC)
  - National Aeronautics and Space Administration (CMU)
  - 2 NSF grants (USC and Georgia Tech)