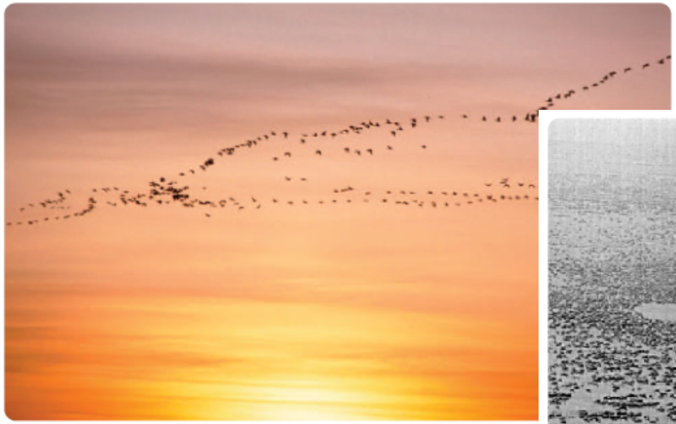
Decision Making
in Robots and Autonomous Agents

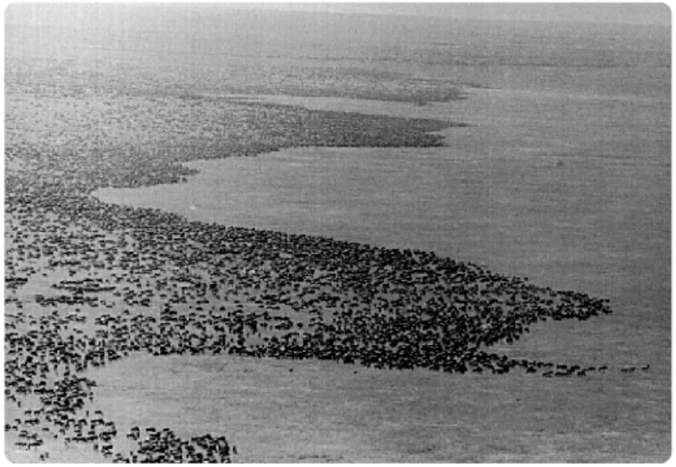
Case Study: Decentralized Resource Allocation and Multi-robot Systems

Subramanian Ramamoorthy
School of Informatics

6 March 2018

Motion Problems with Many Agents





What kind of knowledge does any one agent have?
How does the local knowledge get utilized in a global control strategy?

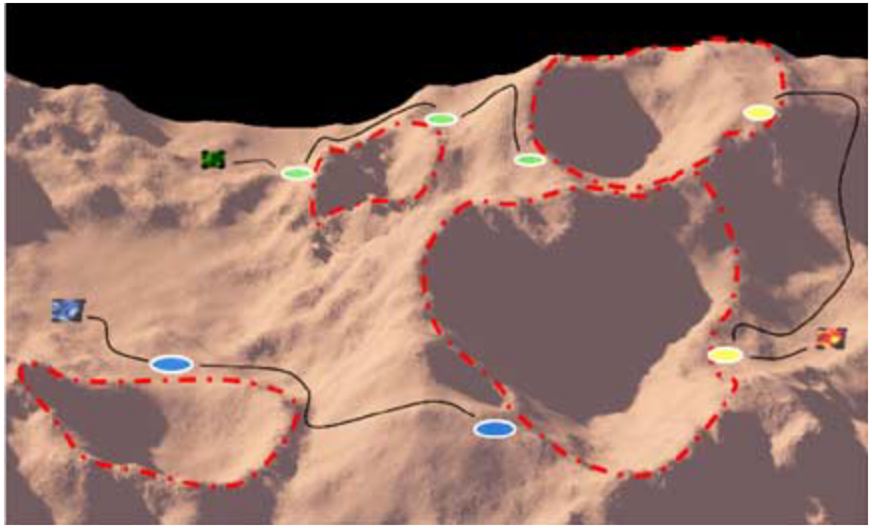
Market Based Approaches

Overview article:

M.B. Dias, R. Zlot, N. Kalra, A. Stentz, Market-based multirobot coordination: A survey and analysis. *Proc. IEEE* 94(7) : 1257 –1270, 2006.

The following slides are an abridged and mildly edited version of a presentation (at ICRA/AAMAS 2006) by the authors of the assigned paper.

Motivating Example: Robots Exploring on Mars

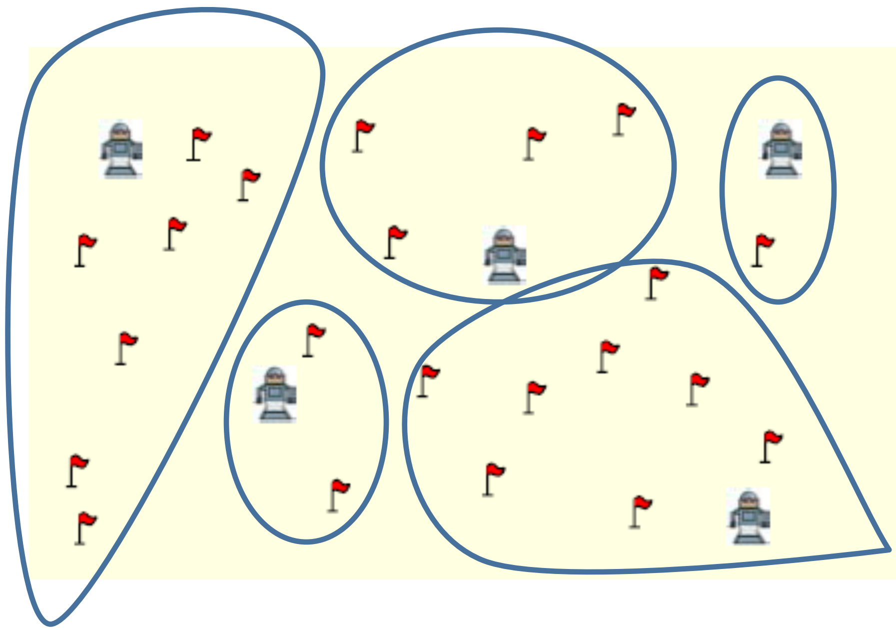


Multi-robot routing:
A team of robots has to visit given targets spread over some known or unknown terrain. Each target must be visited by one robot.

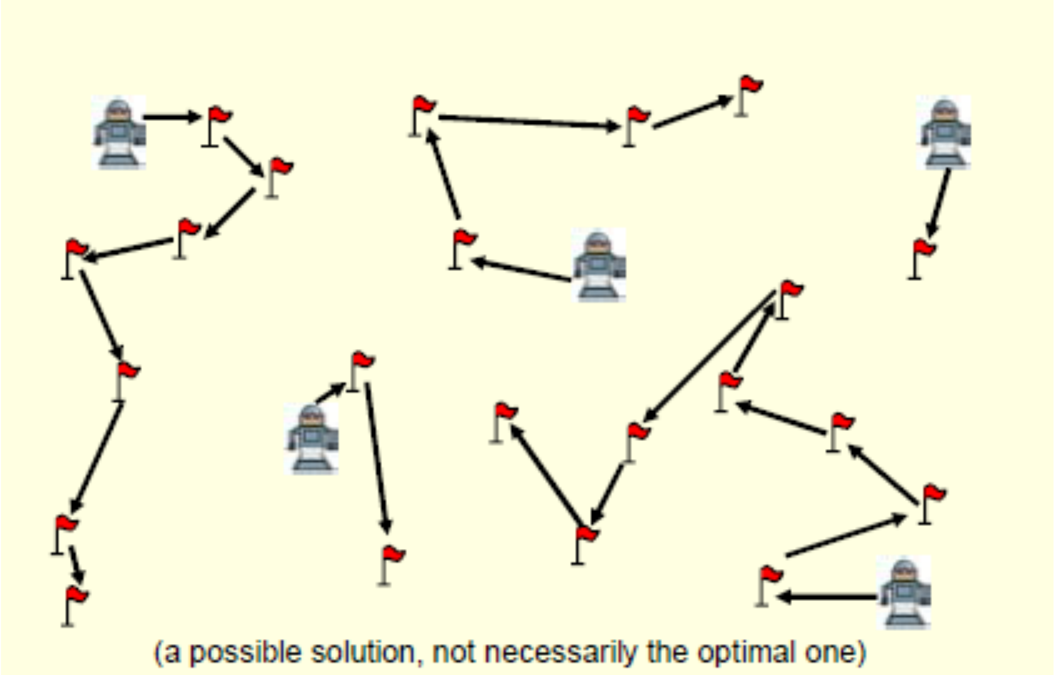
Multi-robot Routing: Assumptions

- 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.

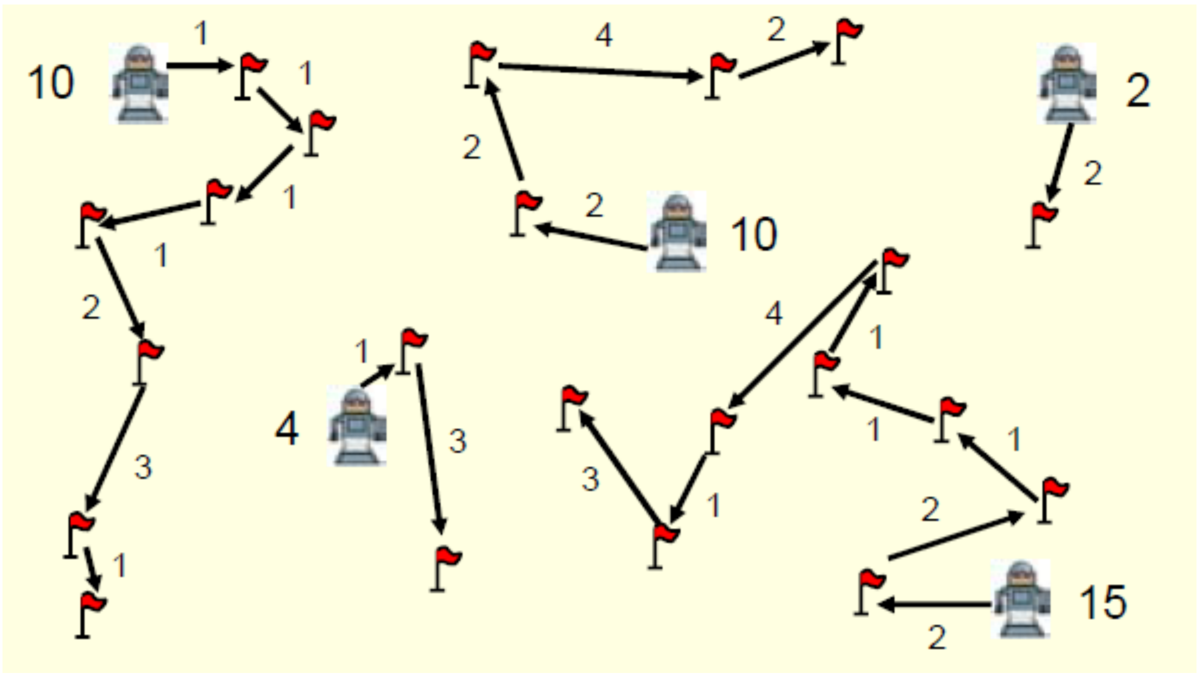
Multi-robot Routing



Multi-robot Routing



Routing: Minimum Sum Team Objective



10+10+2+4+15 = 41

History of Coordination Problem

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

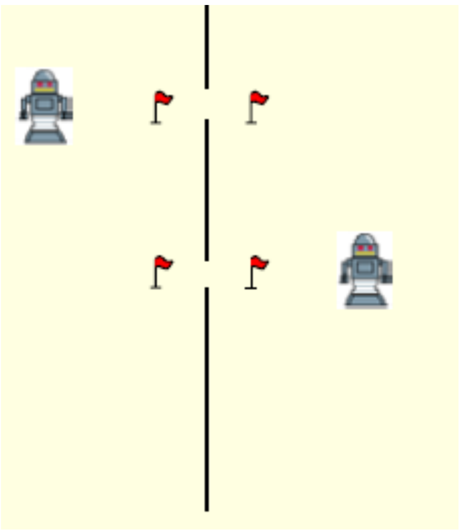
… are not required to return to their start location

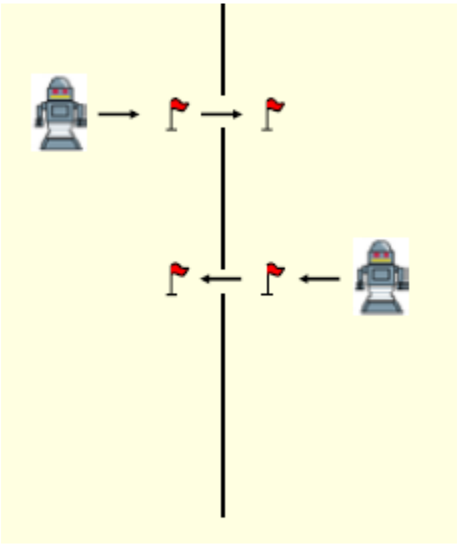
… do not have capacity constraints

Auctions for Multi-Robot Coordination

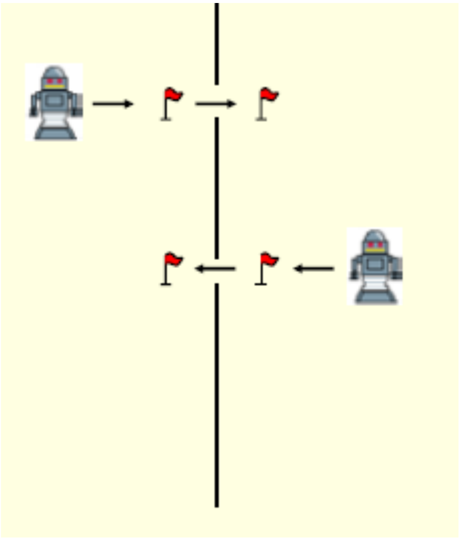
Agent coordination	Auctions
▪ agents	▪ bidders
▪ tasks	▪ items
▪ cost	▪ currency

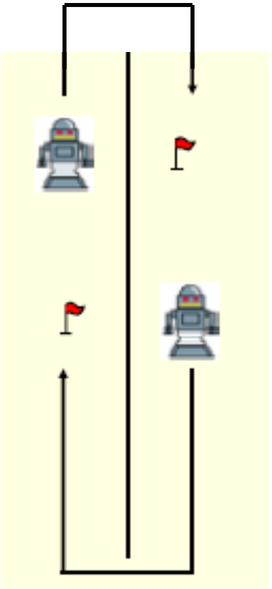
Auctions for Agent Coordination: Known Terrain



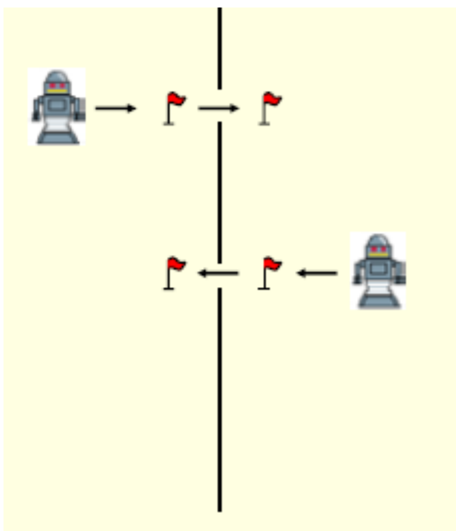


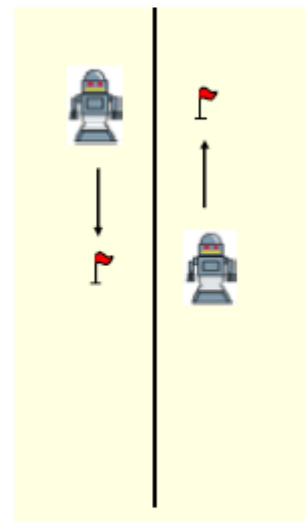
Auctions for Agent Coordination: Unknown Terrain
Plan 1





Auctions for Agent Coordination: Unknown Terrain Plan 2

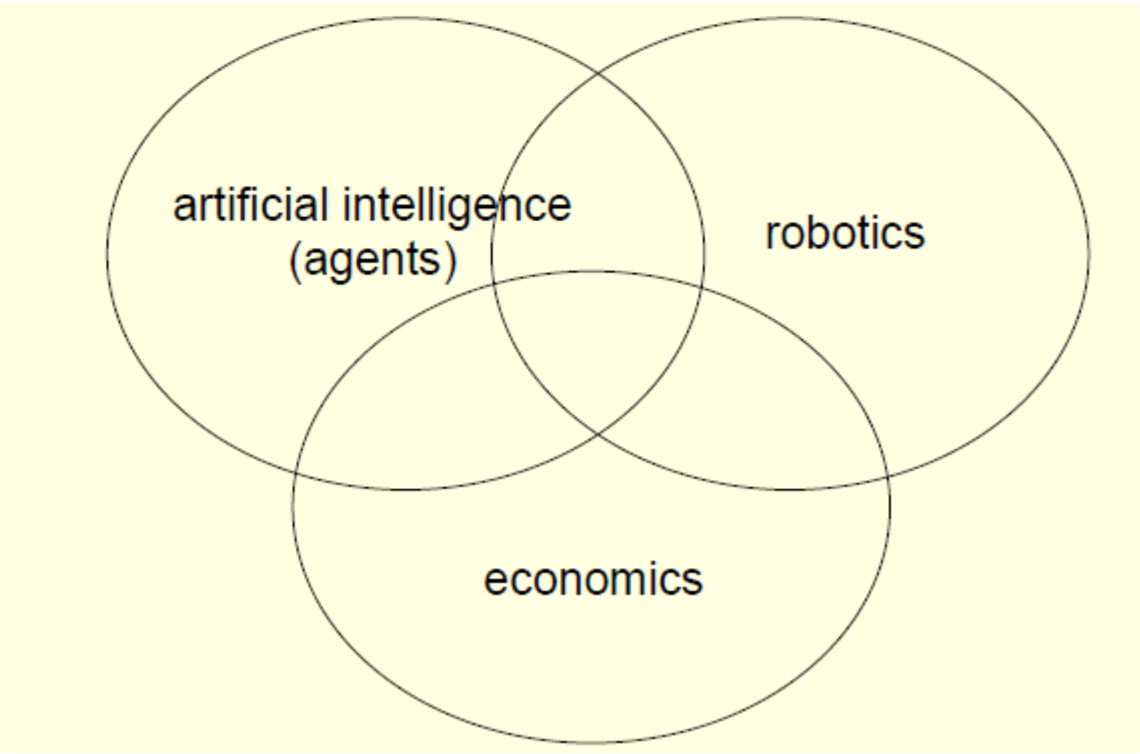




Auctions for Agent Coordination

- 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 Agent Coordination



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:

- eBay
- NASDAQ
- Sothebys

Key Feature: Pricing Mechanism

- Posted prices
  - Static
  - Dynamic
    - Change dynamically over time
    - Customized pricing

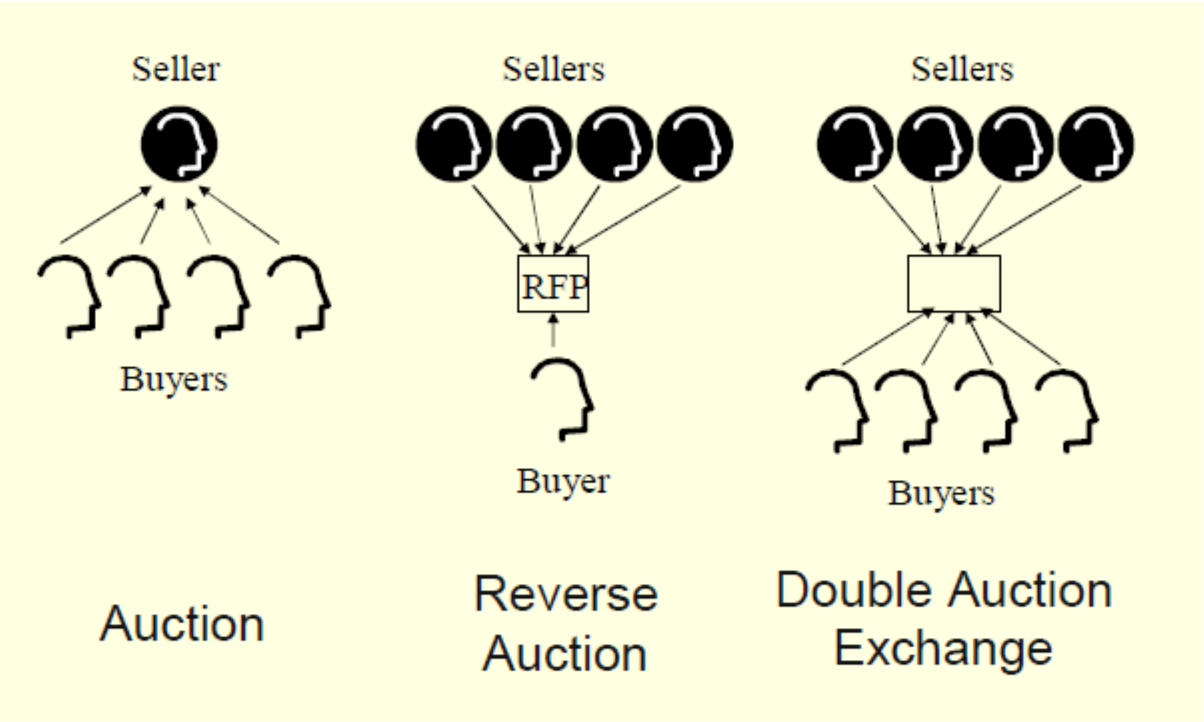
- Price discovery mechanisms
  - Auctions
  - Negotiations

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

Can you think of robotics scenarios with the above characteristics?

Auction Formats



Auctions for Robot Coordination

- 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. Why (we'll discuss…)

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 ≤ m ≤ #bidders

  2. Every bidder awarded one task (*m = #bidders)*

  3. The one best award (*m = 1)*
- For (2) assignment can be done optimally [Gerkey and Mataric 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.

Why/when not Auctions?

- 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

Time Complexity of Auctions

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)	$O(n \cdot v)$	$O(r \cdot n^2)$ [6]	$\lceil n/m \rceil$
Combinatorial	$O(2^n \cdot V)$	$O((b+n)^n)$[5]	1

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

Communication Complexity of Auctions [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), $1 \leq m \leq r$
"winners" = auctioneer only informs the winners of auctions
"winners + losers" = auctioneer also informs the losers that they've lost

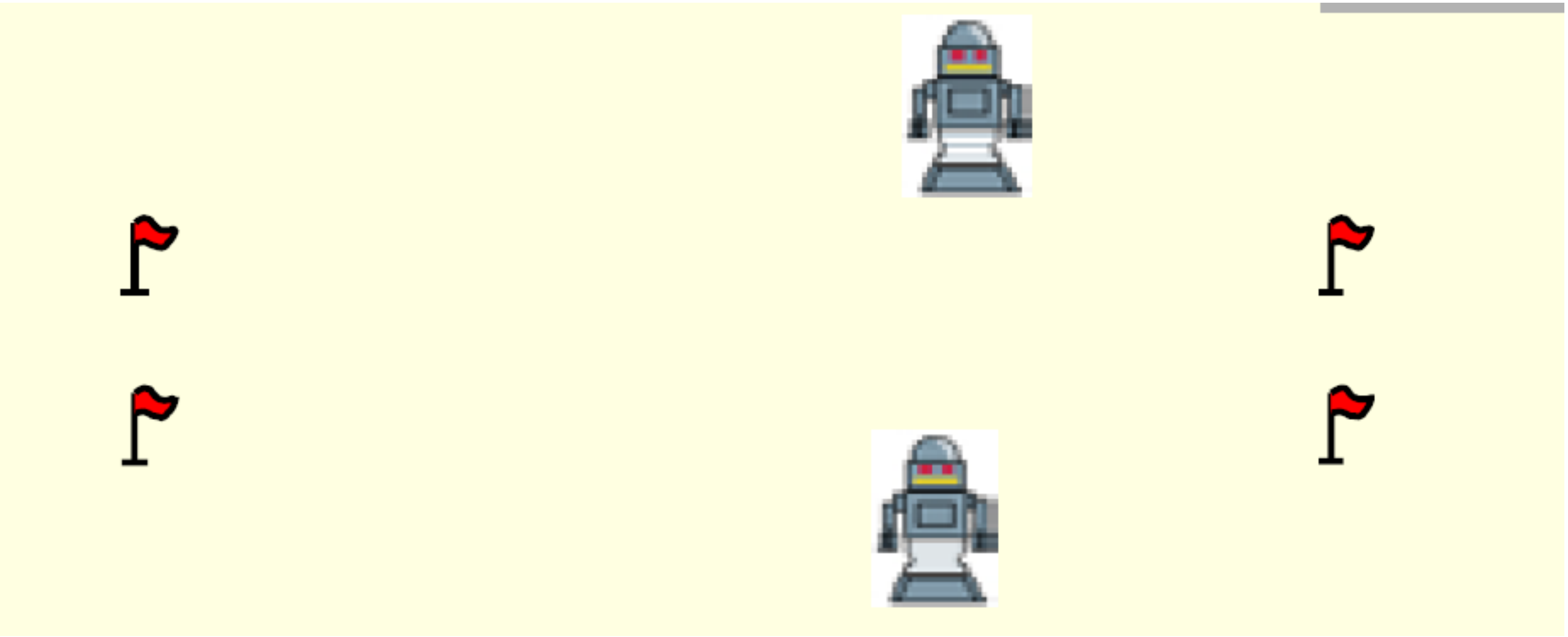
How exactly does this process work?

Let us look at some scenarios:

- Parallel auctions
  - 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…
- Sequential auctions
- Combinatorial auctions

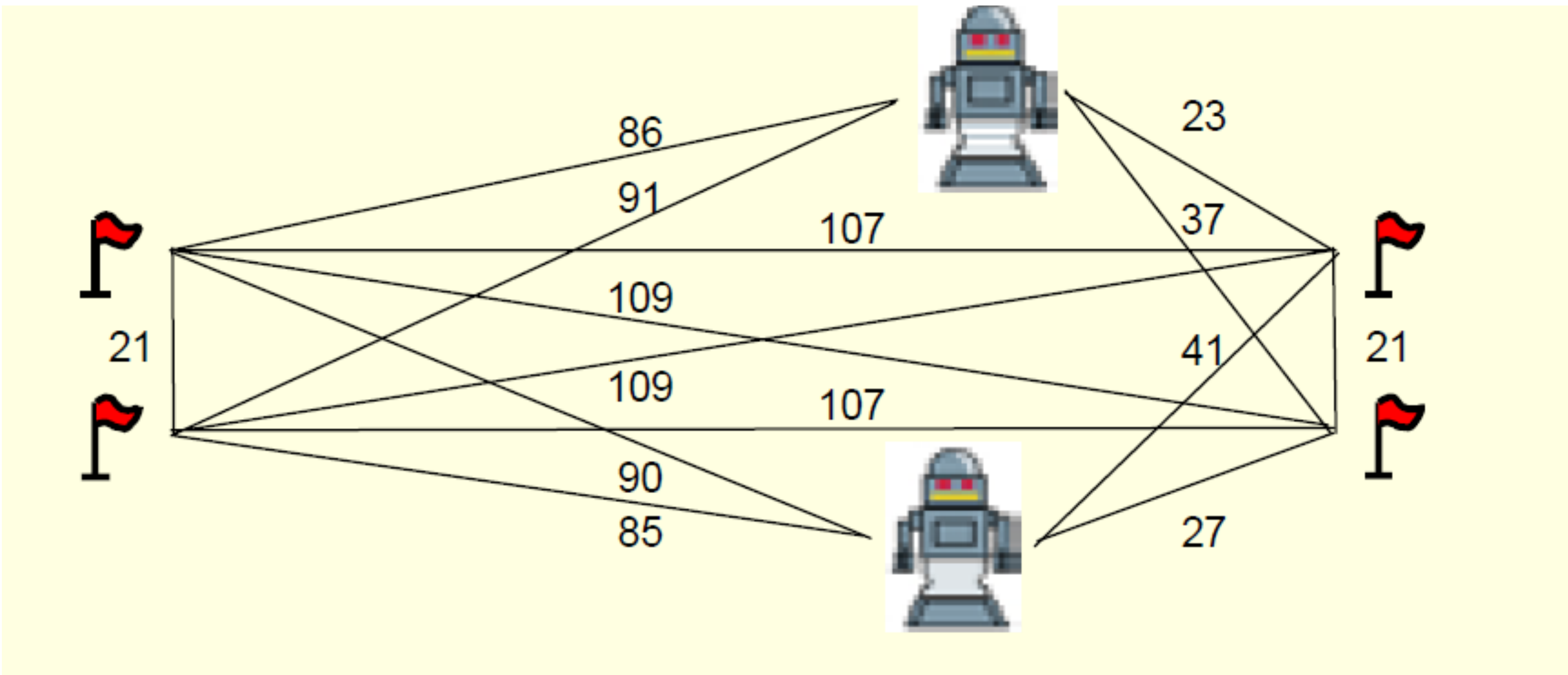
Parallel Auctions

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

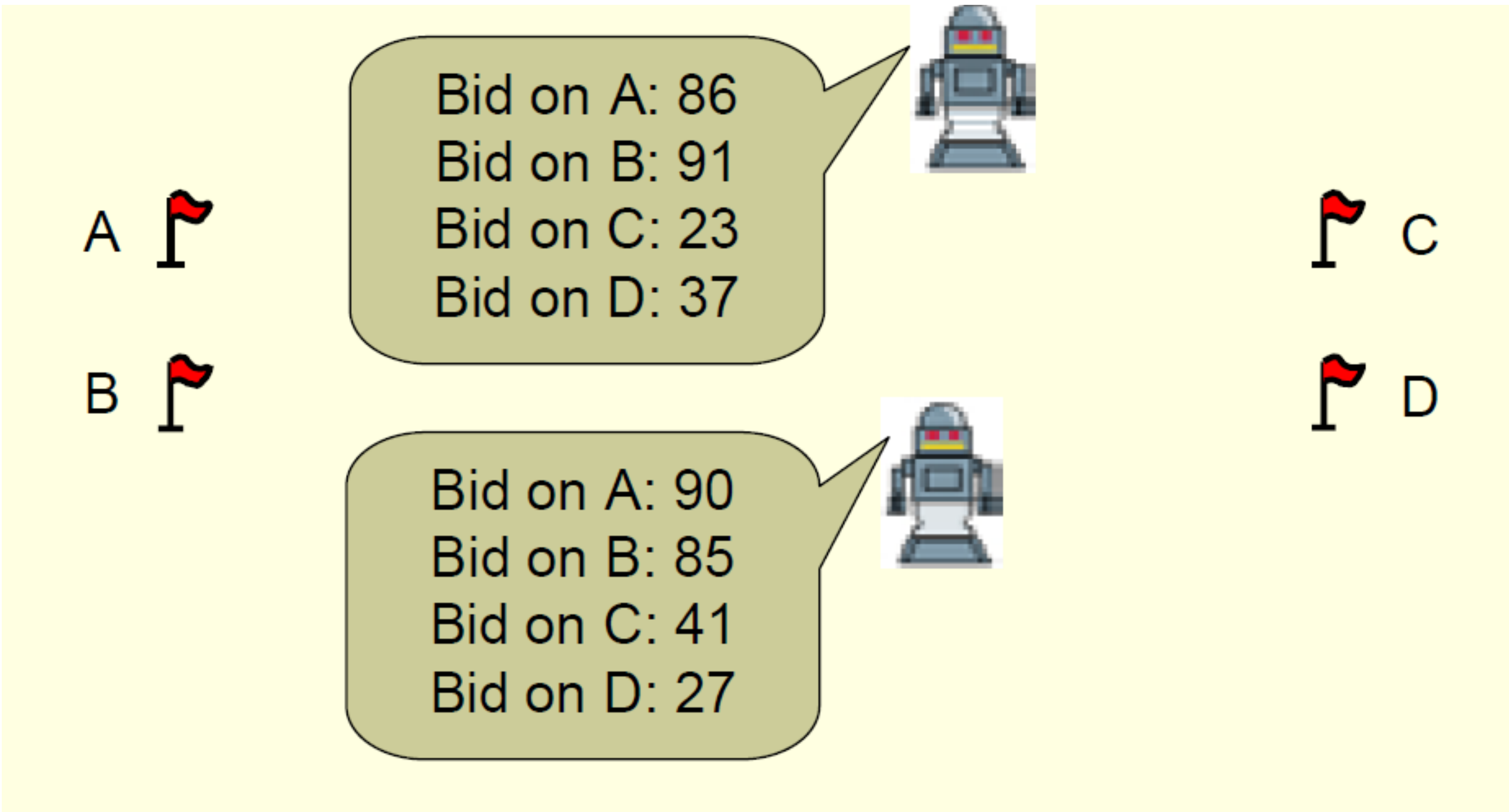


Parallel Auctions

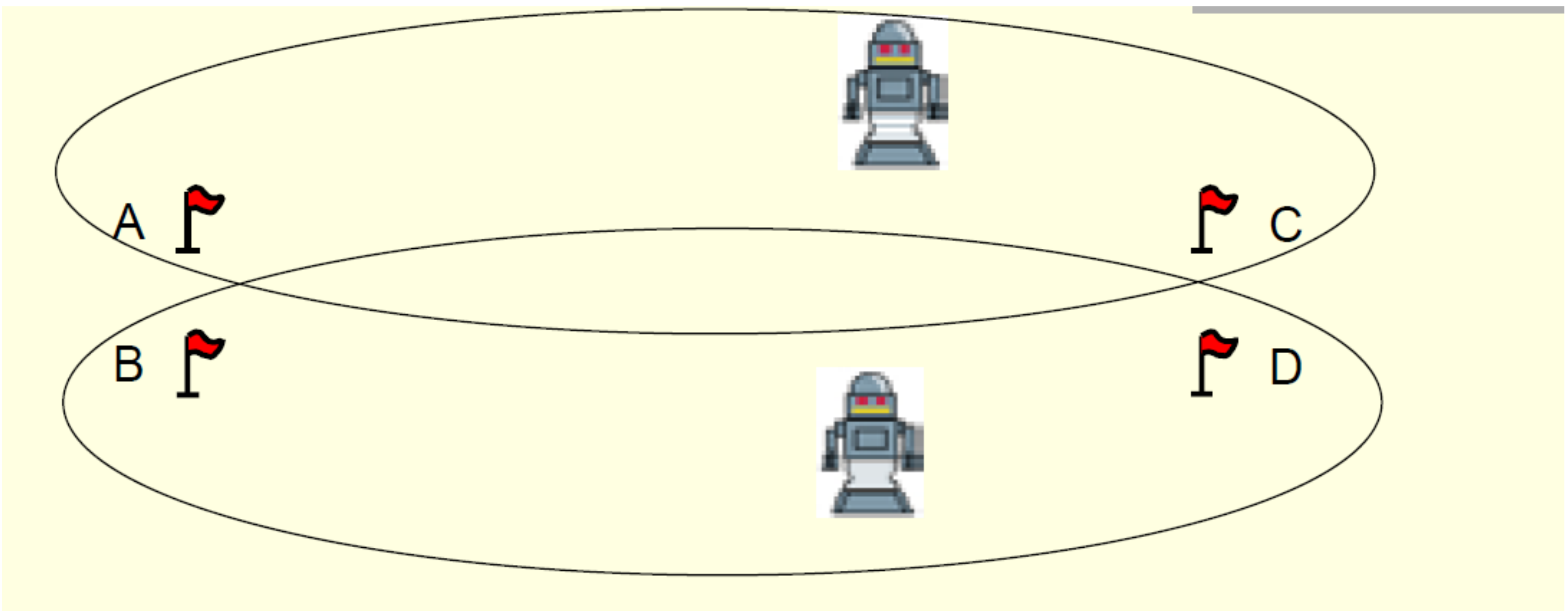
Each robot bids on a target the minimal path cost it needs from its current location to visit the target.



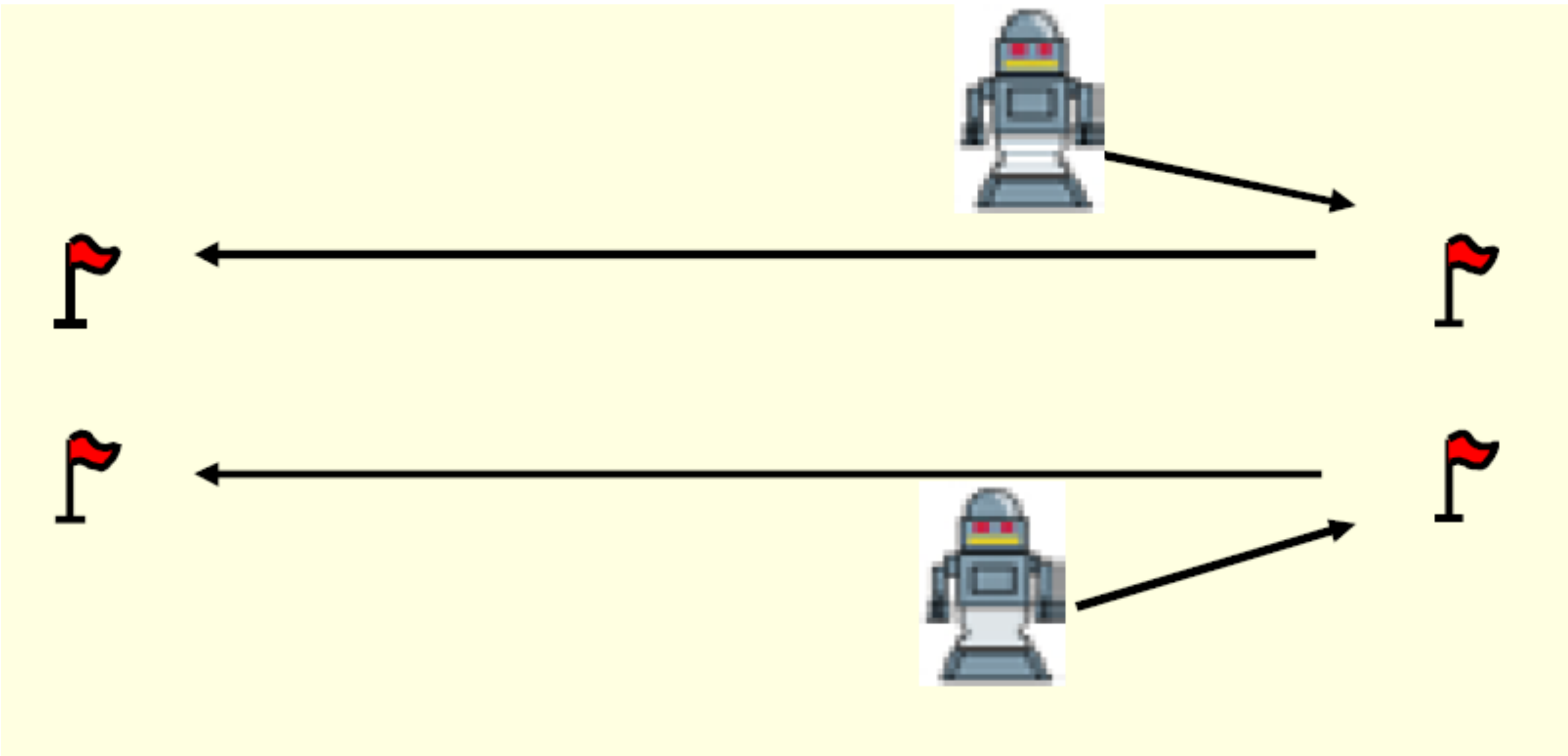
Parallel Auctions



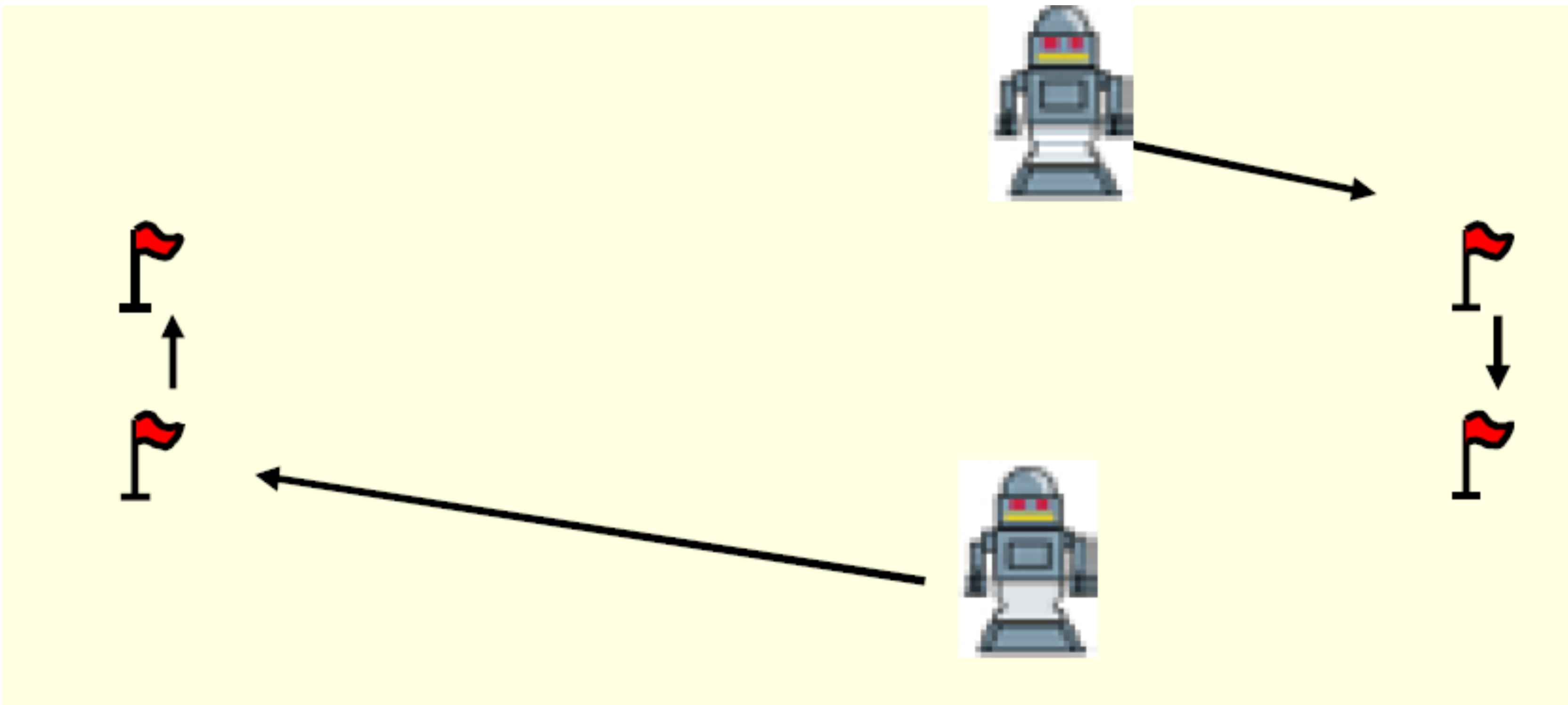
Parallel Auctions



Generated Plan



Limitations of Parallel Auctions



- 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: Good and Bad

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)

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)

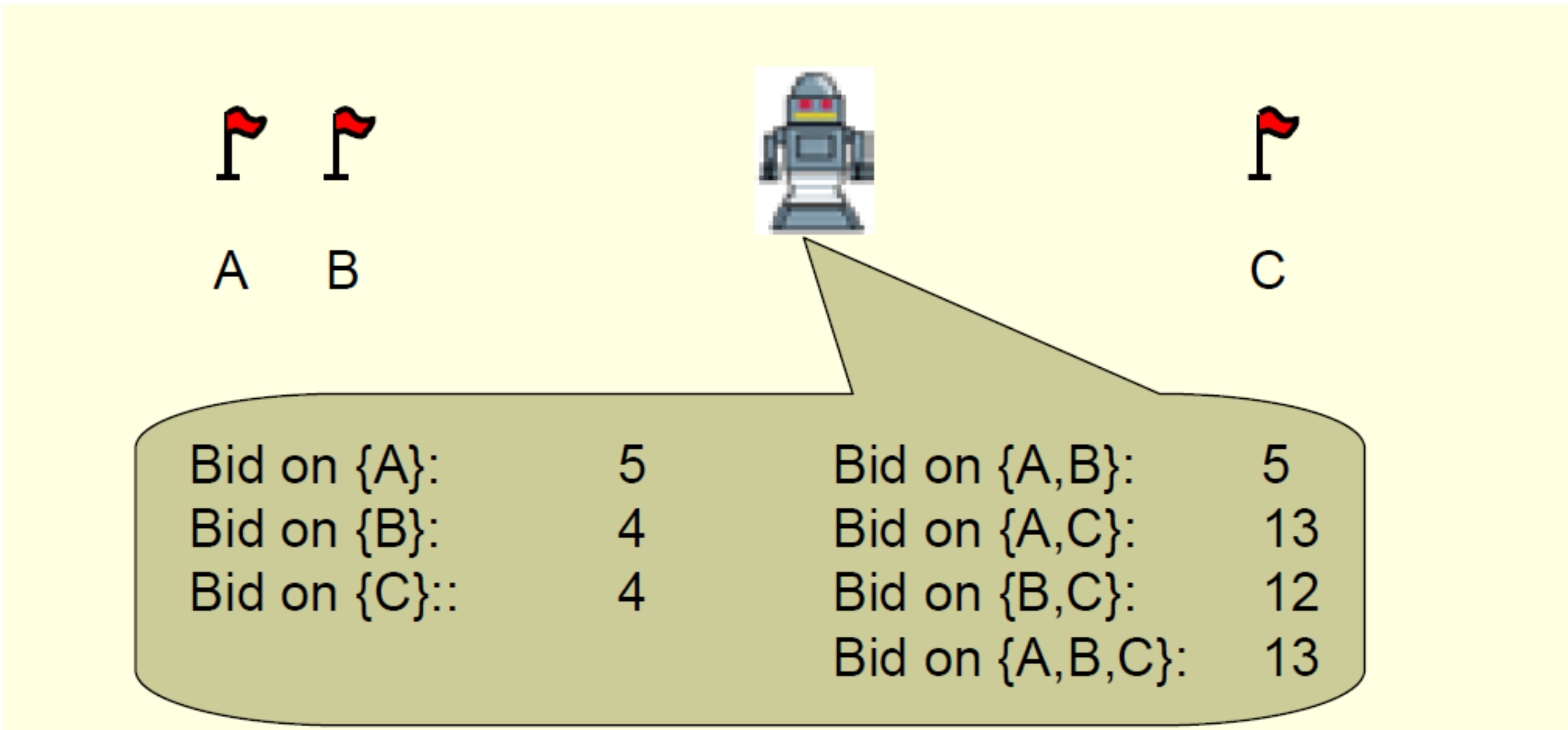
Combinatorial Auctions

- 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.

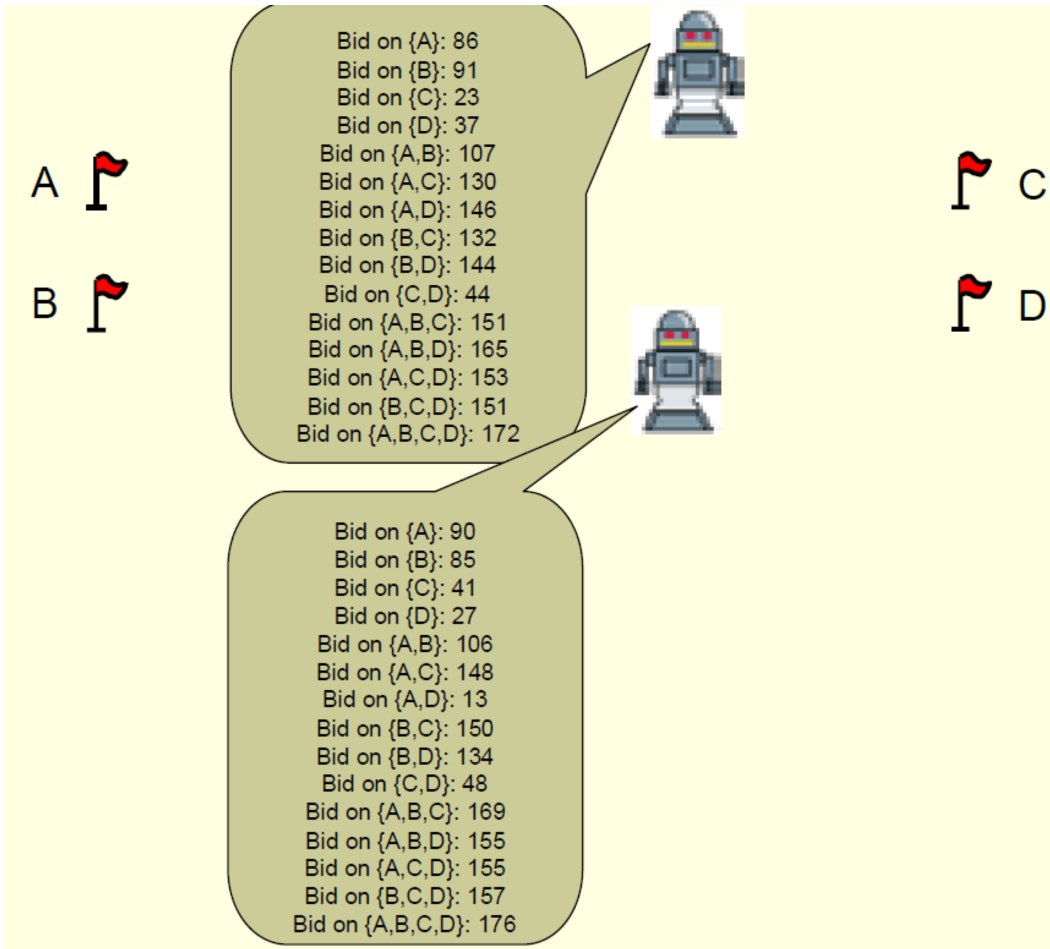
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.

Exploiting Synergies via Combinatorial Auctions

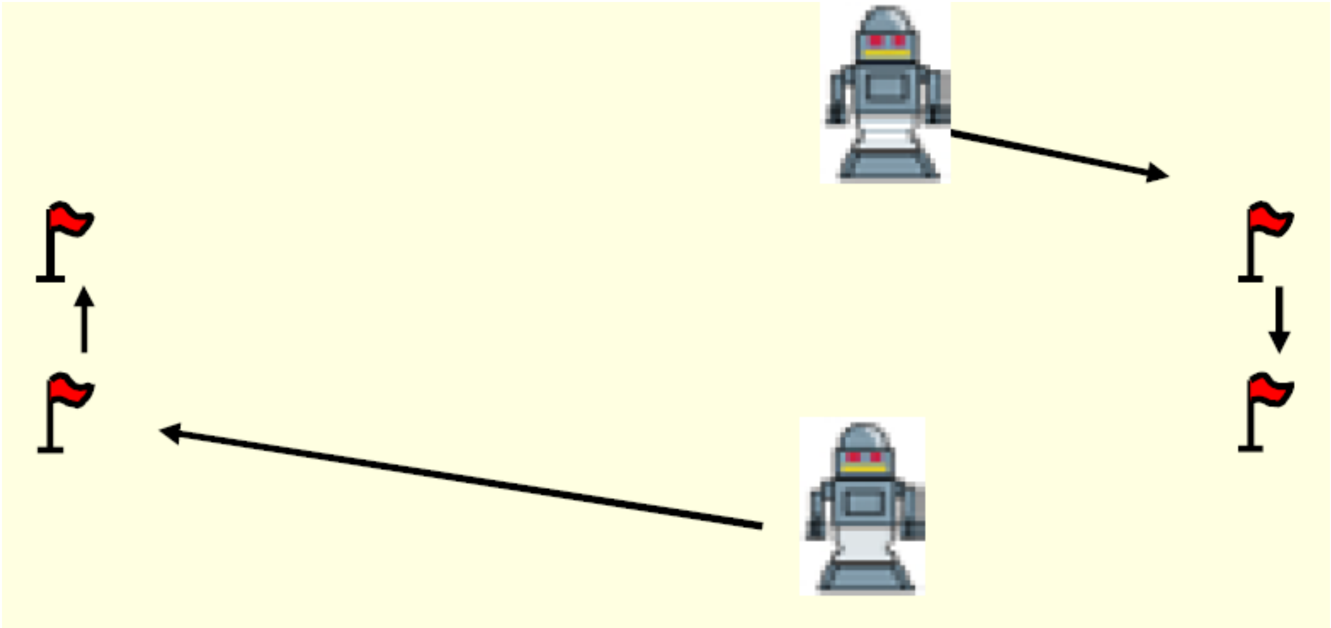


Each robot bids on a bundle the minimal path cost it needs from its current location to visit all targets that the bundle contains.

Multi-robot Combinatorial Auction



Combinatorial Auction Result



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.

Bidding Strategies in Combinatorial Auctions

- 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).

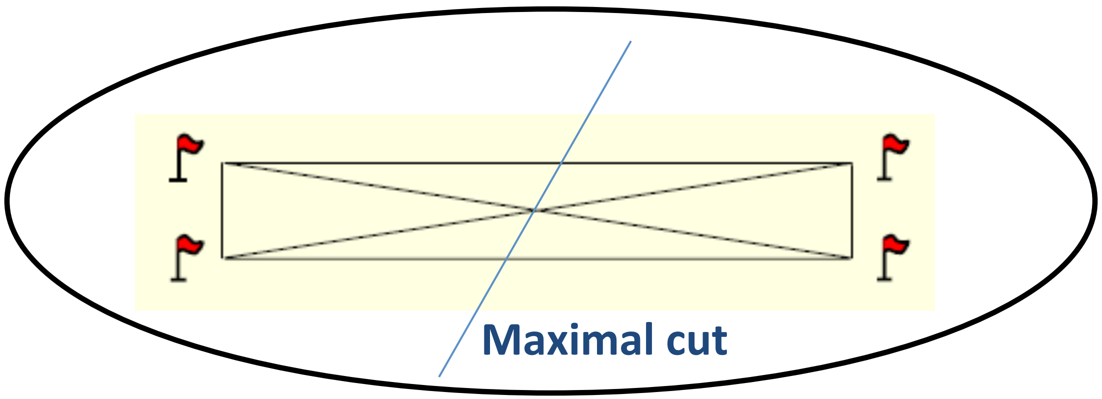
- THREE-COMBINATION
  - Bid on all bundles with 3 targets or less

Note: It might be impossible to allocate all targets.

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.

Domain Dependent Bundle Generation

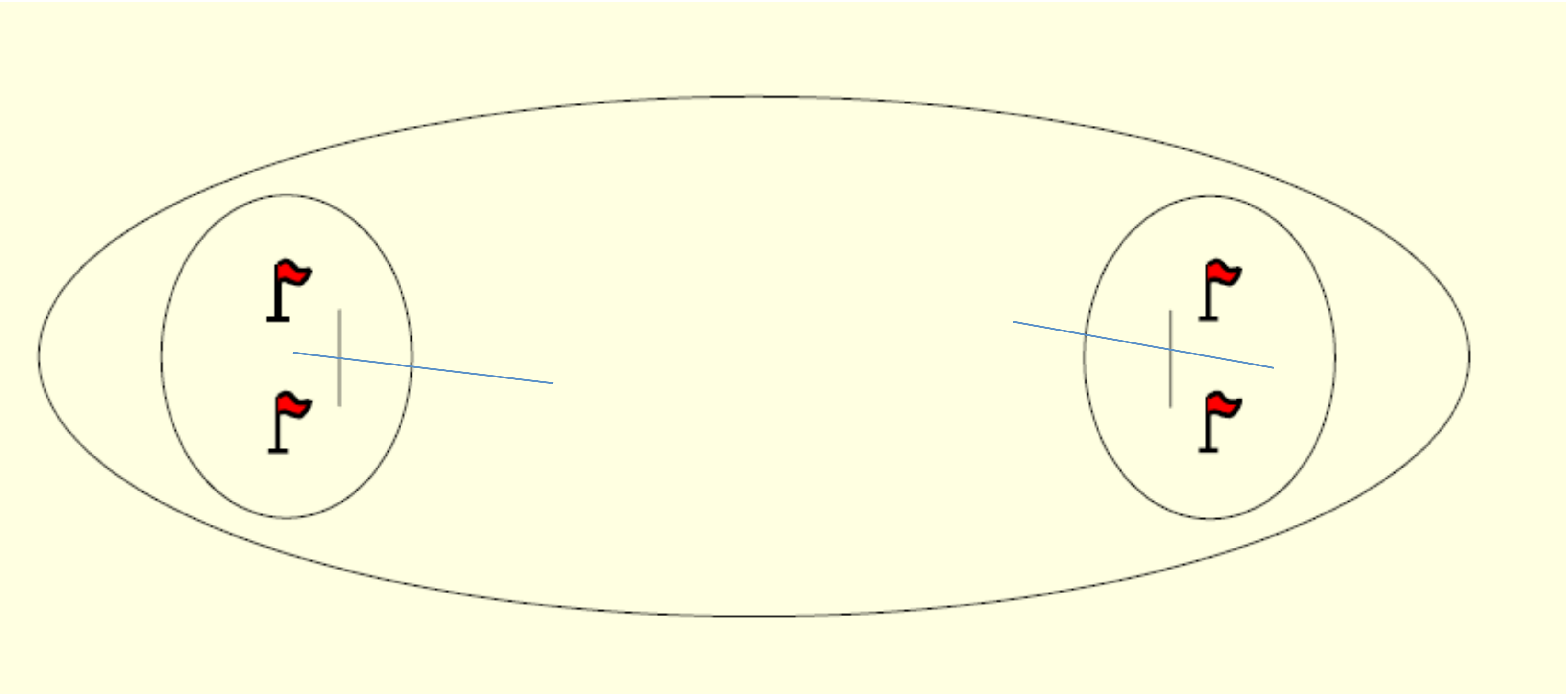


Cut = two sets that partition the vertices of a graph

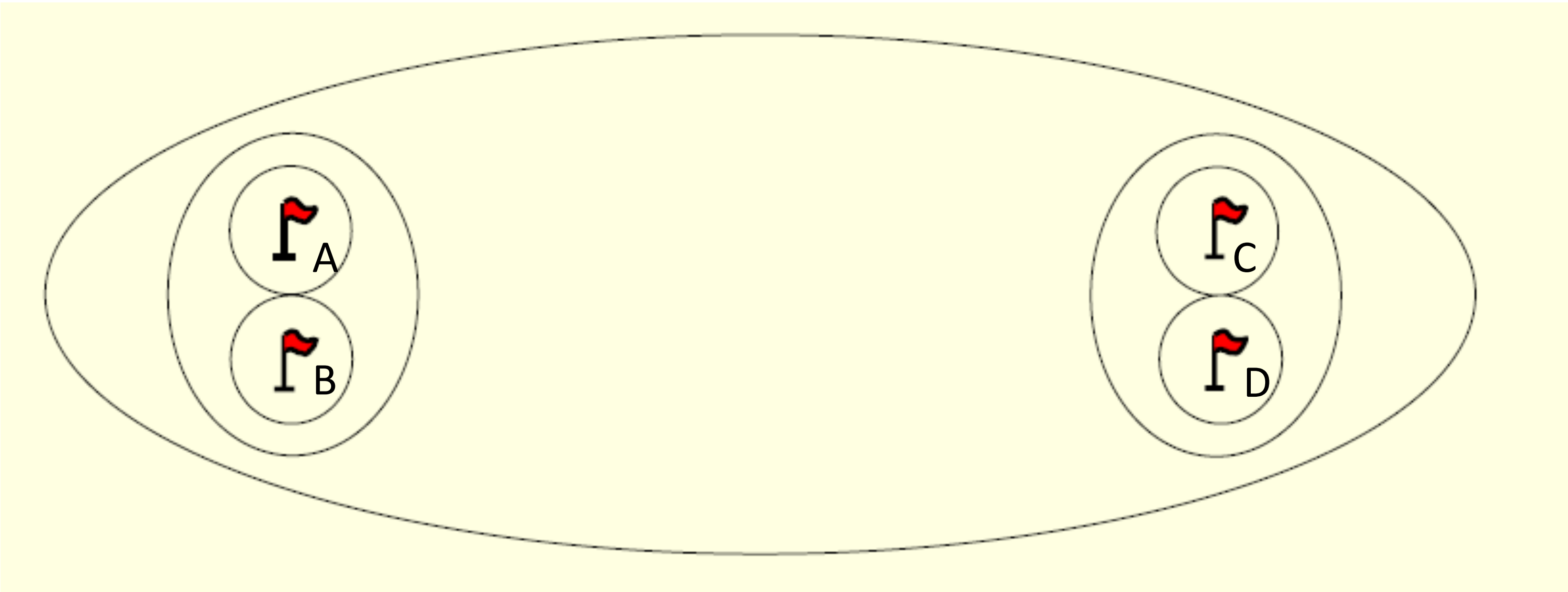
Maximal cut = maxcut = cut that maximizes the sum of the costs of the edges that connect the two sets of vertices

Finding a maximal cut is NP-hard and needs to get approximated.

Domain Dependent Bundle Generation



Domain Dependent Bundle Generation



Submit bids for the following bundles

- {A}, {B}, {C}, {D}
- {A,B}, {C,D}
- {A,B,C,D}

Performance

	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)

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

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.

Parallel vs. Combinatorial Auctions

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.

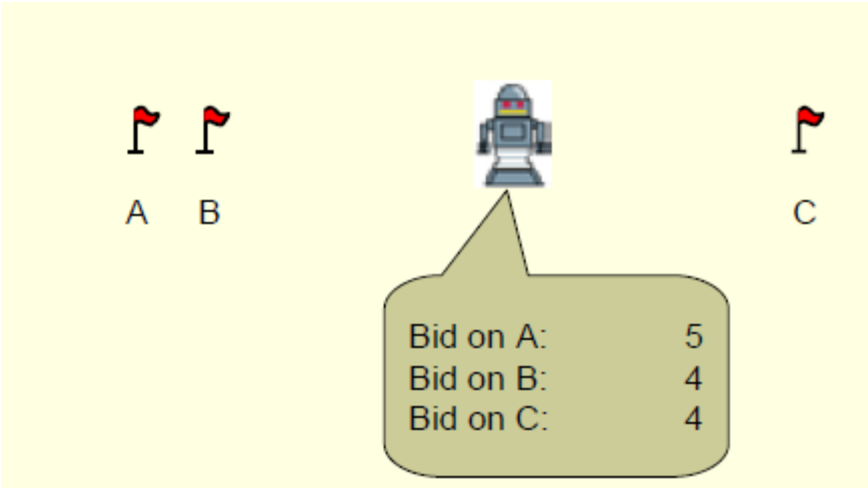
Sequential Auction Procedure

- 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.)

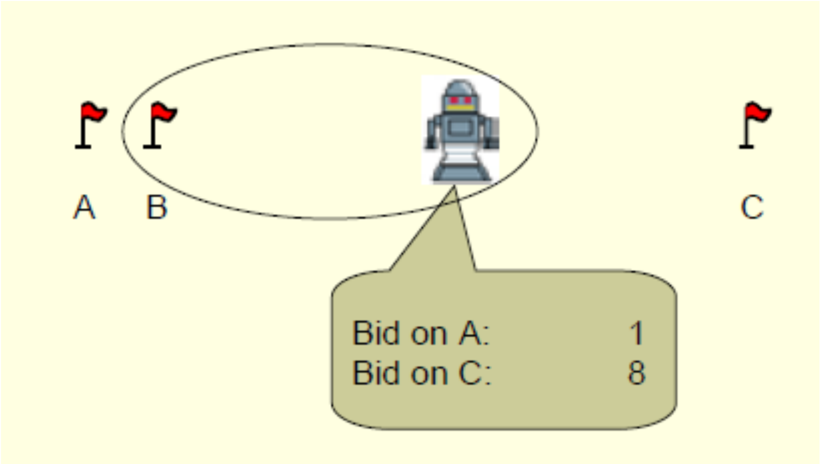
- Each robot determines a cost-minimal path to visit all targets it has won and follows it.

Sequential Auctions: Synergy



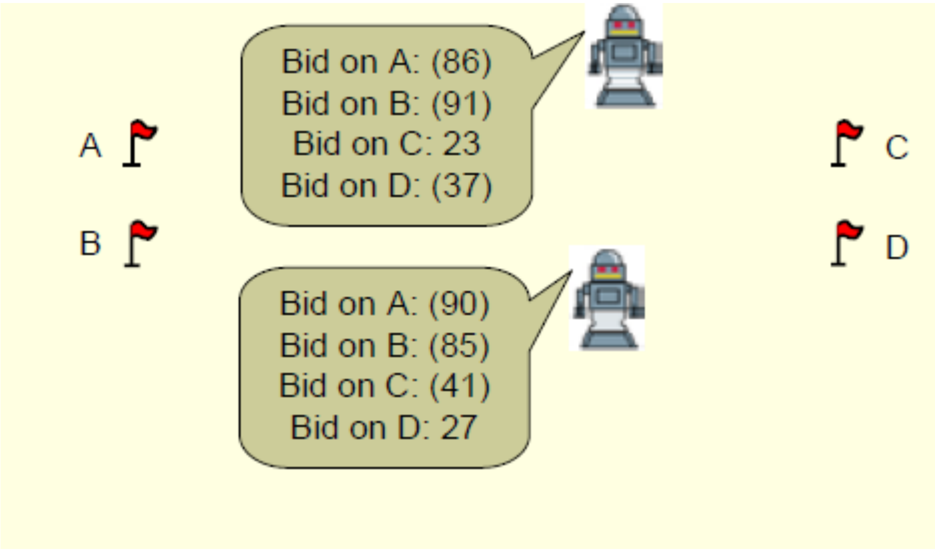
Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BidSumPath).

Sequential Auctions: Synergy

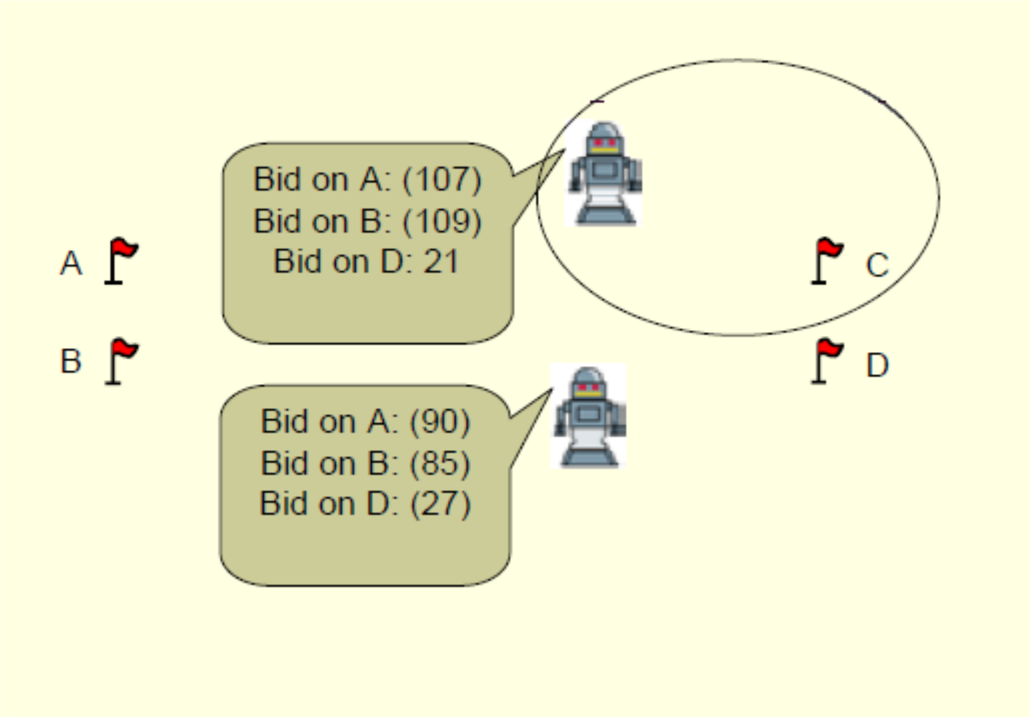


Each robot bids on a target the increase in minimal path cost it needs from its current location to visit all of the targets it has won if it wins the target (BidSumPath).

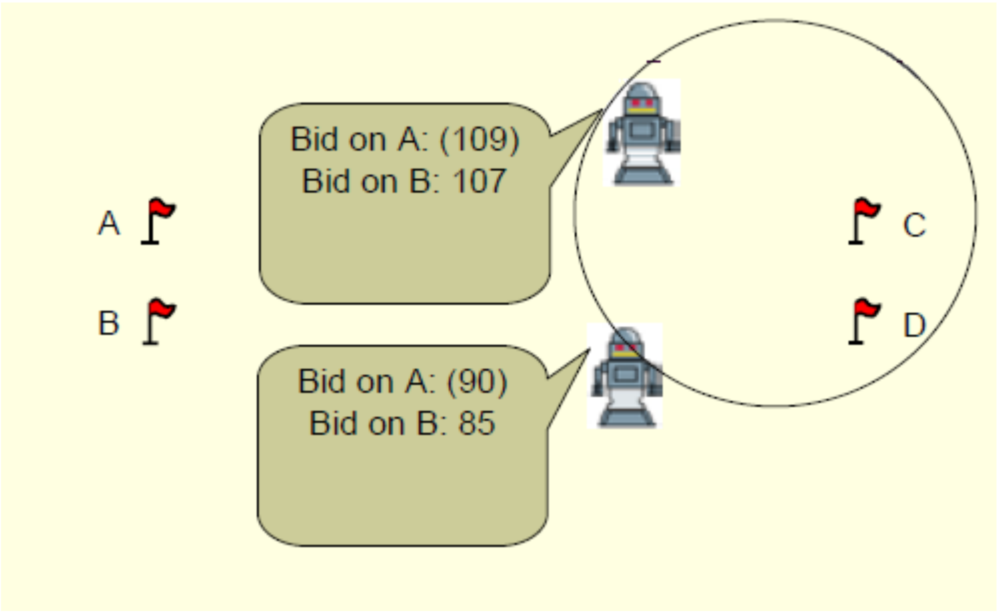
Sequential Auctions with multiple robots



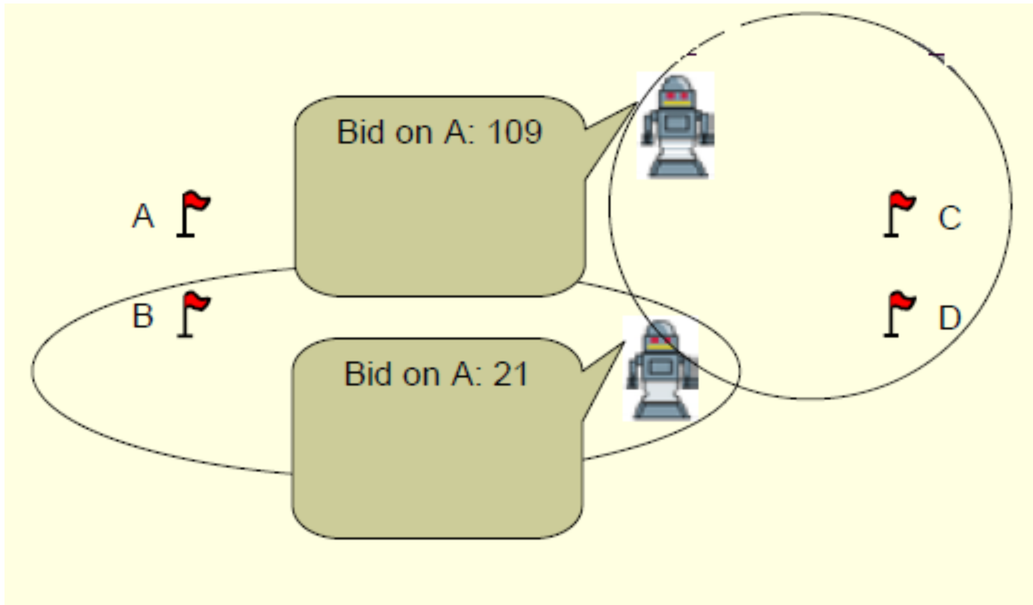
Sequential Auctions with Multiple Robots

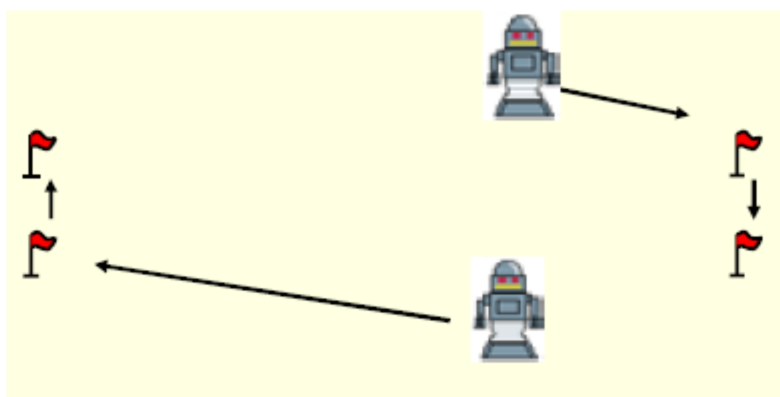


Sequential Auctions with Multiple Robots



Sequential Auctions with Multiple Robots

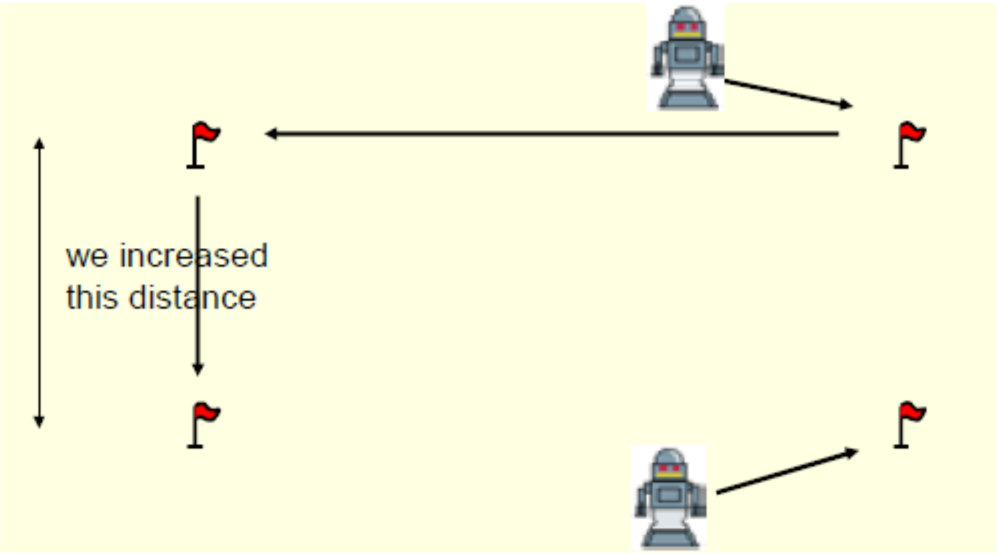




Sequential Auctions Procedure

- 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

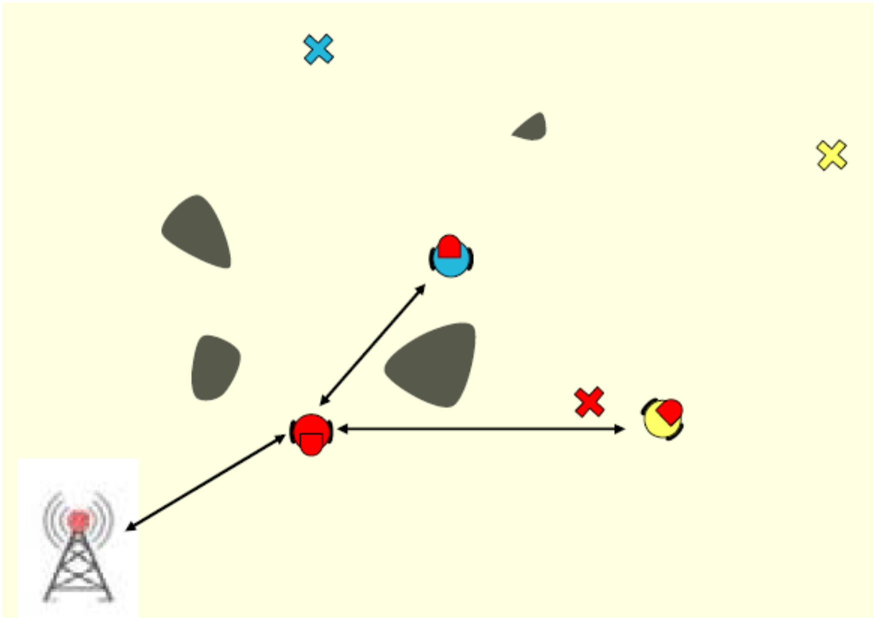
- Ease of implementation: relatively simple
- Ease of decentralization: simple
- Bid generation: cheap
- Bid communication: cheap
- Auction clearing: cheap
- Team performance: very good
  - some synergies taken into account

Various Kinds 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

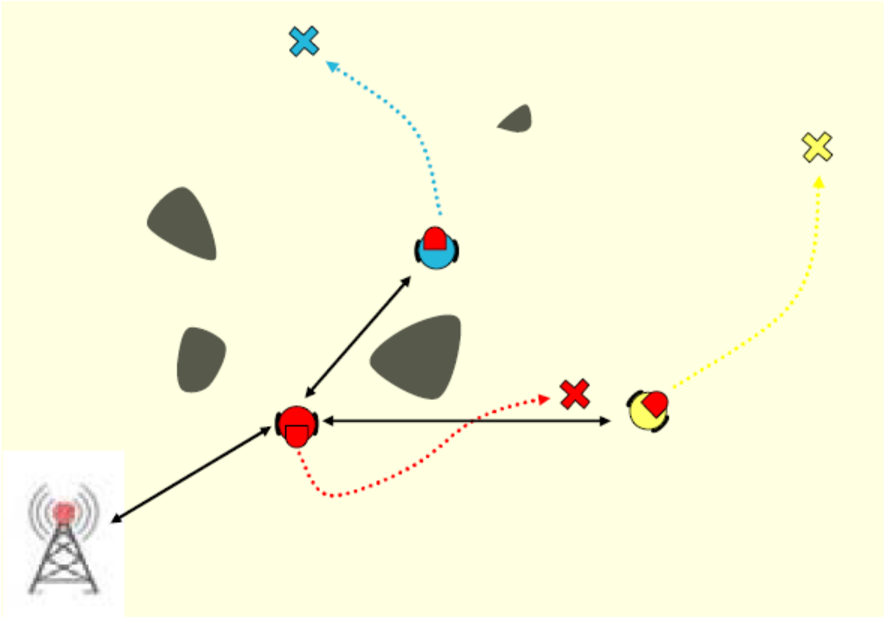
Small Example of Coordinated Motion

Setup: each robot must go to its goal target without losing contact with the radio tower. The cost of travel is relatively small compared to the high cost of LOS communication.



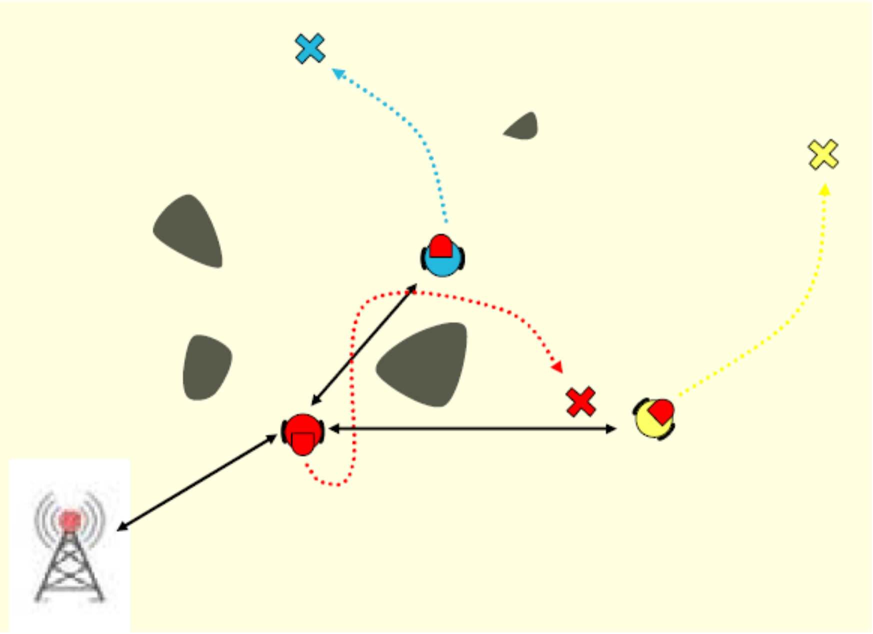
Small Example of Coordinated Motion

Robots independently generate paths to their goals while considering their teammates' paths. The LOS between red and yellow will not break so they do not need to actively coordinate. But LOS will break between red and blue. Both red and blue will be penalized if they follow their current paths.



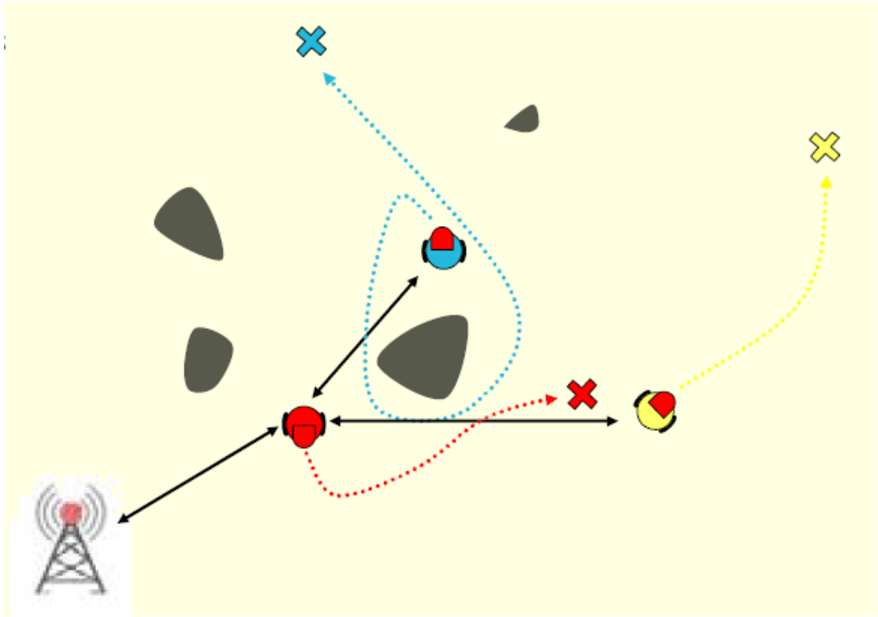
Small Example of Coordinated Motion

The blue robot proposes this joint plan to the red robot and requests a bid from the red robot for its participation. Red's bid will be too expensive because the proposed plan causes LOS loss between red and yellow.



Small Example of Coordinated Motion

The red robot sends blue a counter offer of this joint plan to the blue robot and requests a bid from the blue robot. Although the path is long, blue's bid will be less costly because it will have communication with the tower. This path will be adopted by the two robots.



Considerations when designing Coordination Mechanisms

- 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?