15-382 Collective Intelligence – S18

Lecture 25:
Task Allocation 4

Instructor:
Gianni A. Di Caro

جامعة كارنيجي ميلون في قطر
Carnegie Mellon University Qatar

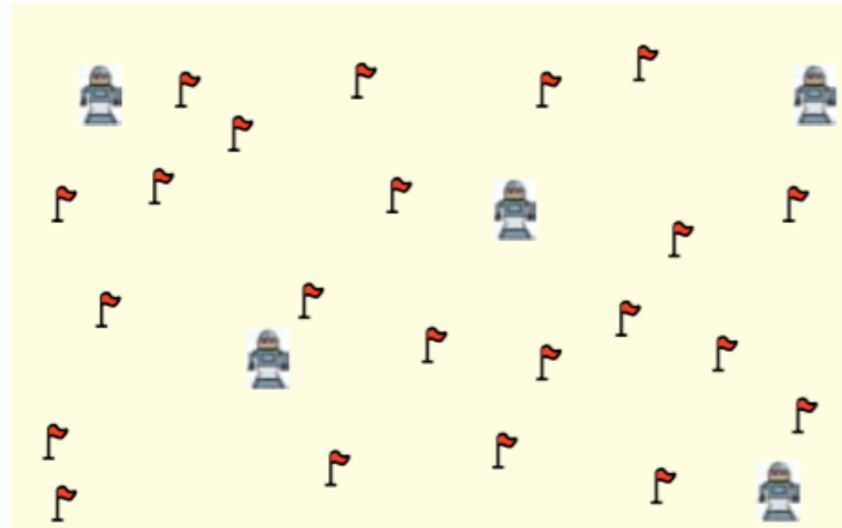
Types of Auctions for Task Allocation

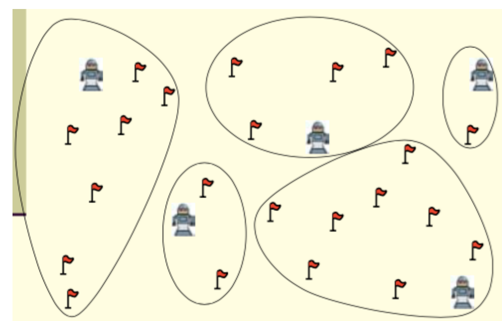
- **Parallel Auctions**
  - Each robot bids *on each task* (=single-item) in independent and simultaneous auctions
- **Combinatorial Auctions**
  - Each robot bids *on some bundles* (= subsets) of tasks
- **Sequential Auctions**
  - There are *several parallel auctions bidding rounds* until all tasks have been assigned to robots. Only one task is assigned in each round. A bundle is defined/assigned at the end of the rounds

Guiding example: Multi-Robot Routing

Scenario

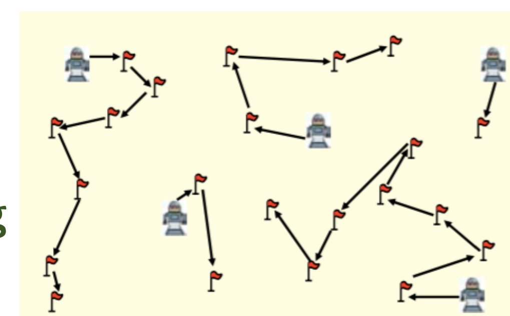
- **Agents = Robots**, **Tasks = Targets**
- A team of robots has to visit given targets spread over some terrain, *minimizing costs*
- A subset of tasks has to be assigned to each robot such that all tasks are serviced
- Each target must be visited by one and only one robot
- The cost of servicing *any task by any robot* is a constant *c*: the allocation has to minimize the costs related to traveling to tasks under the constraint of servicing all tasks
- *Examples:*
  - Goods delivery to spatially spread customers (Uber/Amazon)
  - Planetary surface exploration
  - Facility surveillance
  - Search and rescue





Task Assignment

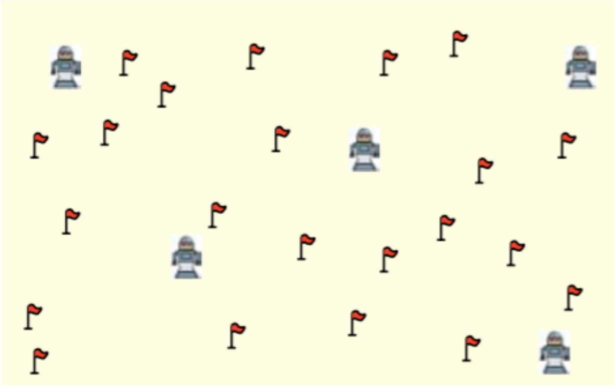
Paths / Tasks ordering

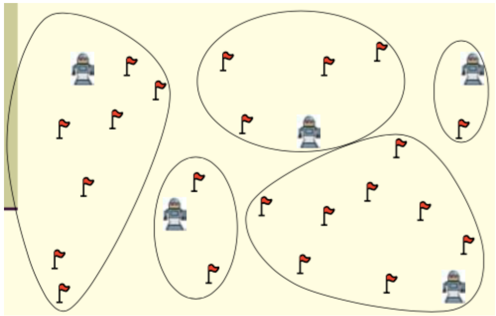


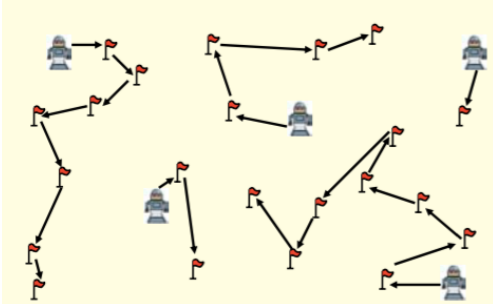
Guiding example: Multi-Robot Routing

Assumptions

- The robots are identical (c cost for servicing any task))
- The robots know their own location
- The robots know task 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 (auctioning)
- Tasks have *no service dependencies* (e.g., τ_i before τ_j)
- Each task only require one robot to be serviced







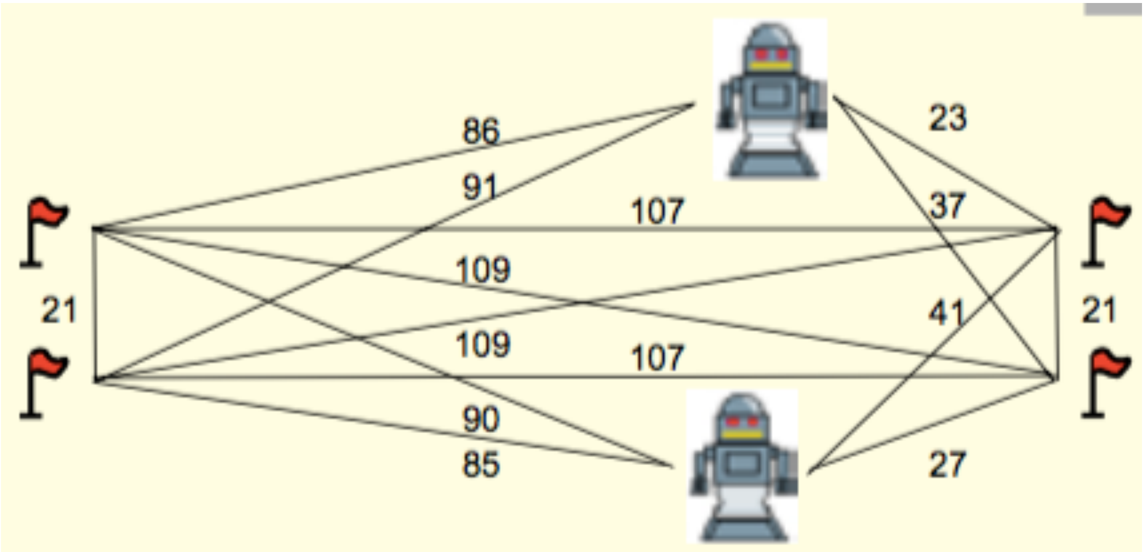
- Note: *Simplified scenario* compared to the general ones considered so far
- **SR – ST – TA**
- Complication: Utility function is *not linear (task dependencies are in the costs)*
  - $U^r(\tau_i, \tau_j) \neq U^r(\tau_i) + U^r(\tau_j)$

Parallel Auctions

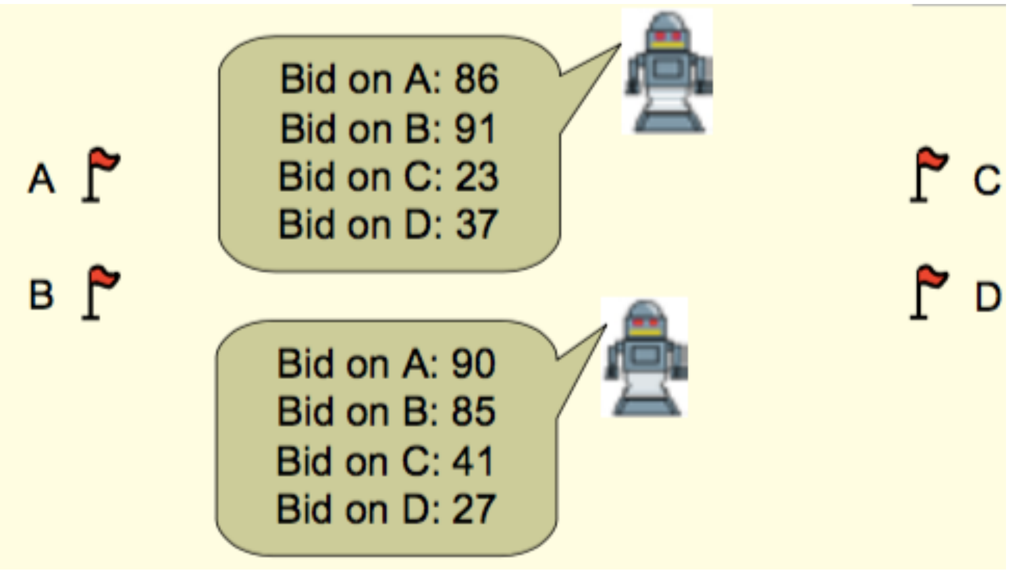
- Each robot bids on each target/task in *independent and simultaneous auctions.*
- The robot that <u>bids lowest on a target wins it</u> (minum cost / energy / time to perform the task)
- Each robot determines a <u>cost-minimal path</u> to visit all targets it has won and follows it → **Sequence of tasks to deal with**

- Each robot bids on a target the <u>minimal path cost</u> it needs from its current location to visit the target
- This might be an *estimate*

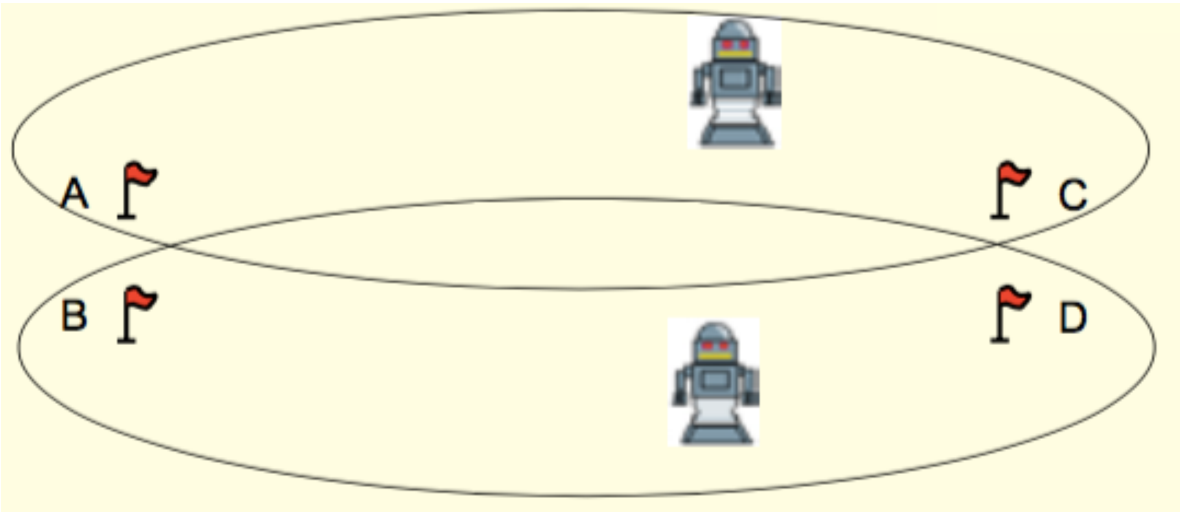
Parallel Auctions



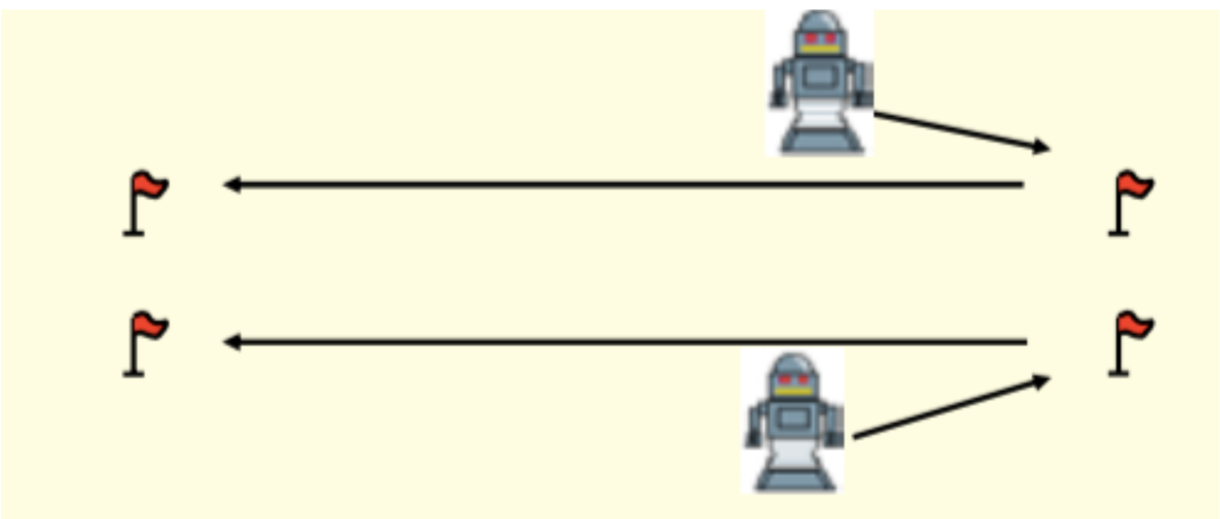
- Each robot bids on a target the <u>minimal path cost</u> it needs from its current location to visit the target
- This might be an *estimate*



Parallel Auctions



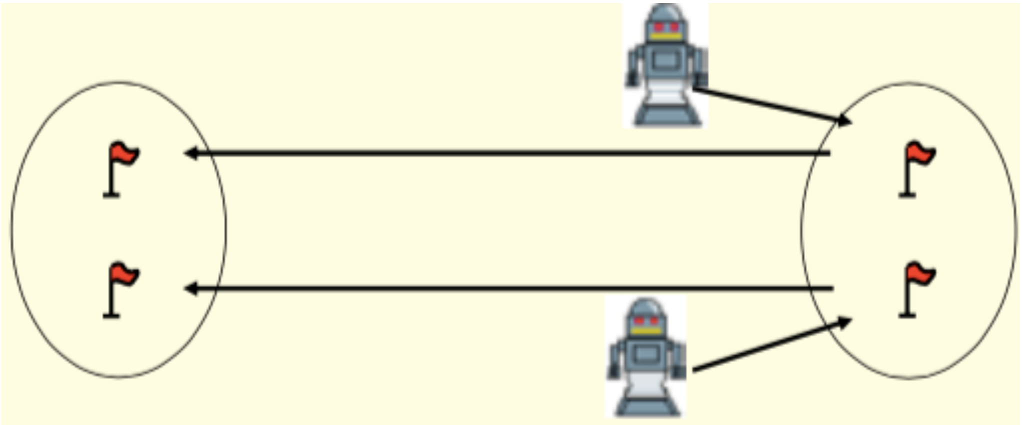
Task Assignment



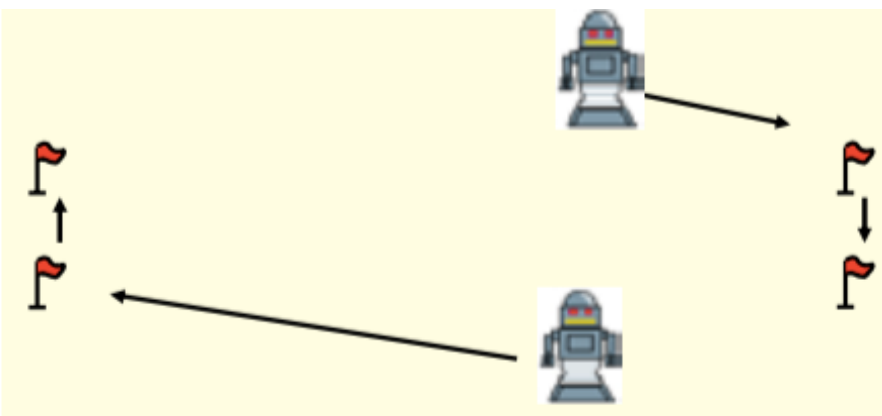
Robot Paths

Does it seem optimized?

Parallel Auctions



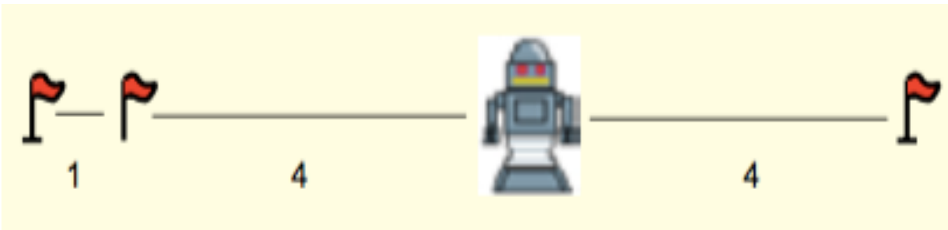
Sub-optimal Task Assignment: it is often the case that it is not convenient to send different robots to deal with tasks that are clustered (in space)

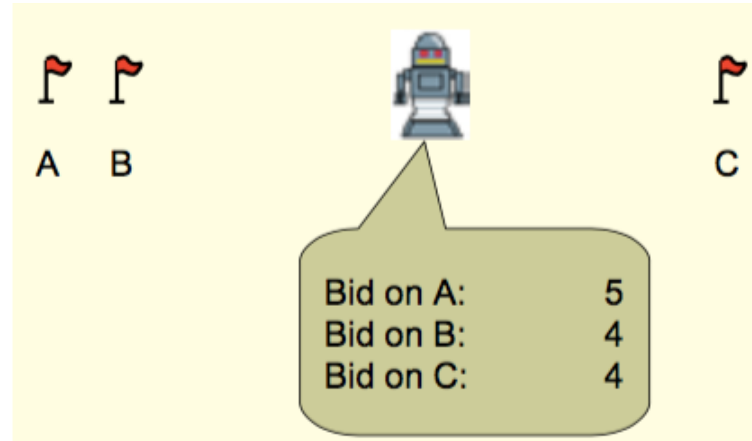


<u>Optimal solution, with minimal team cost</u>

- <u>Minimal team cost is not achieved</u>
- The team cost resulting from parallel auctions is large because they **cannot take *synergies* between tasks into account.**

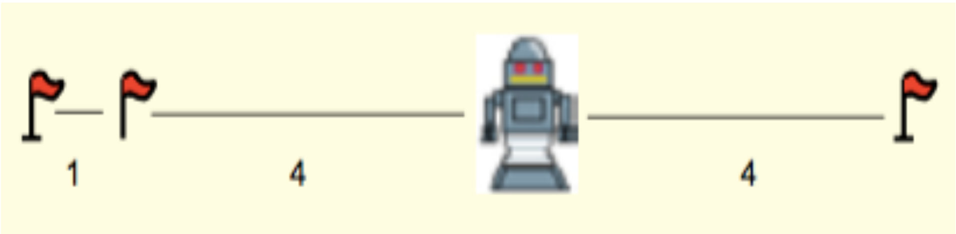
Parallel Auctions: Not considering Synergies

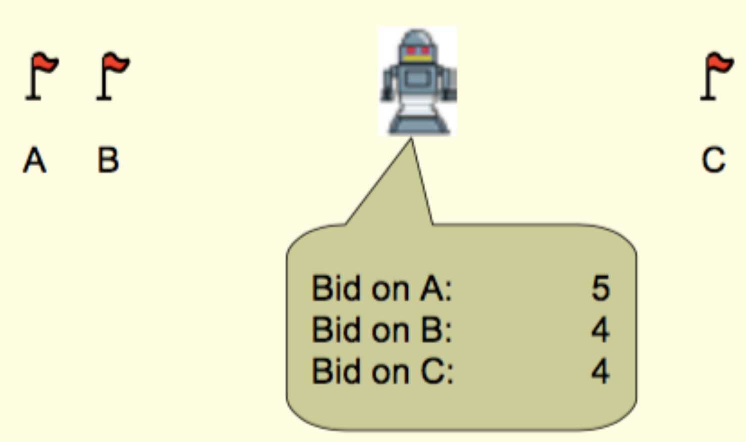




- Each robot bids on a target the minimal path cost it needs from *its current location* to visit the target
- No synergies among tasks are accounted for: the **order** of performing the tasks (i.e., of visiting the targets) is not considered
- Effects: wrong estimates of the real costs
  - **Overestimate total costs** in case of positive task synergies
  - **Underestimate total costs** in case of negative task synergies

Parallel Auctions: Positive Synergies



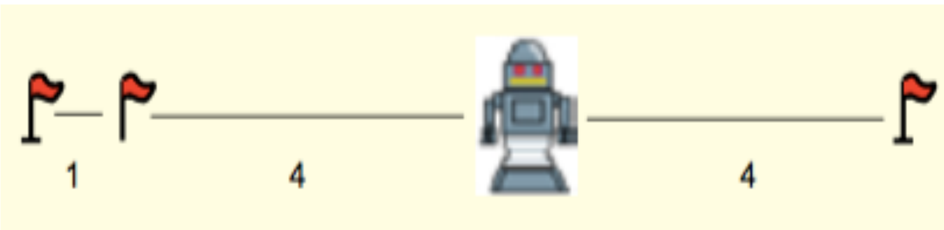


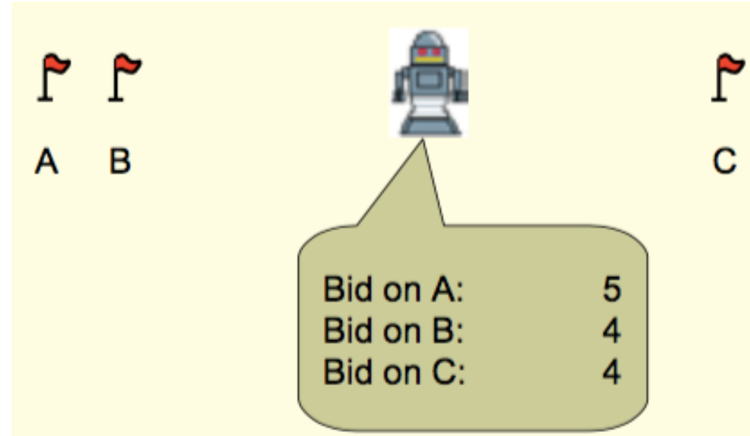
Smallest path cost to visit A: 5
Smallest path cost to visit B: 4
Smallest path cost to visit A and B: 5

smallest path cost to visit A and B
<
smallest path cost to visit A + smallest path cost to visit B

Overestimate of total costs

Parallel Auctions: Negative Synergies





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

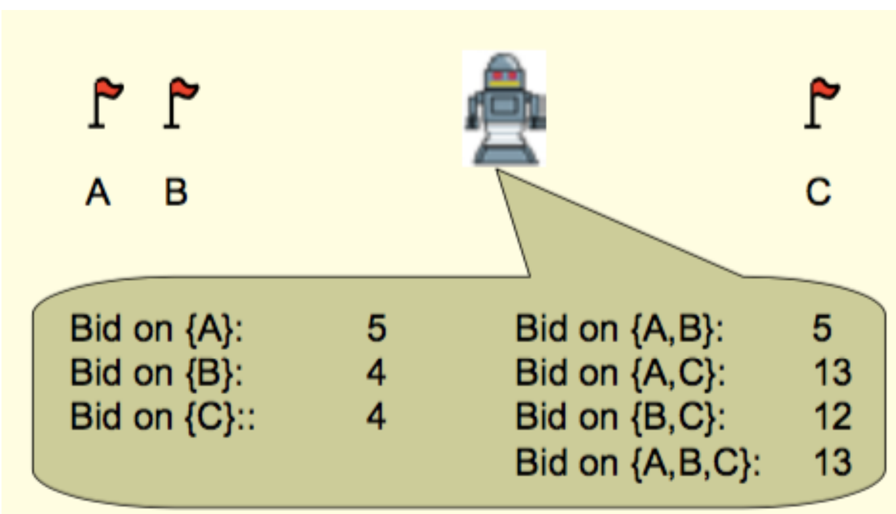
Underestimate of total costs

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

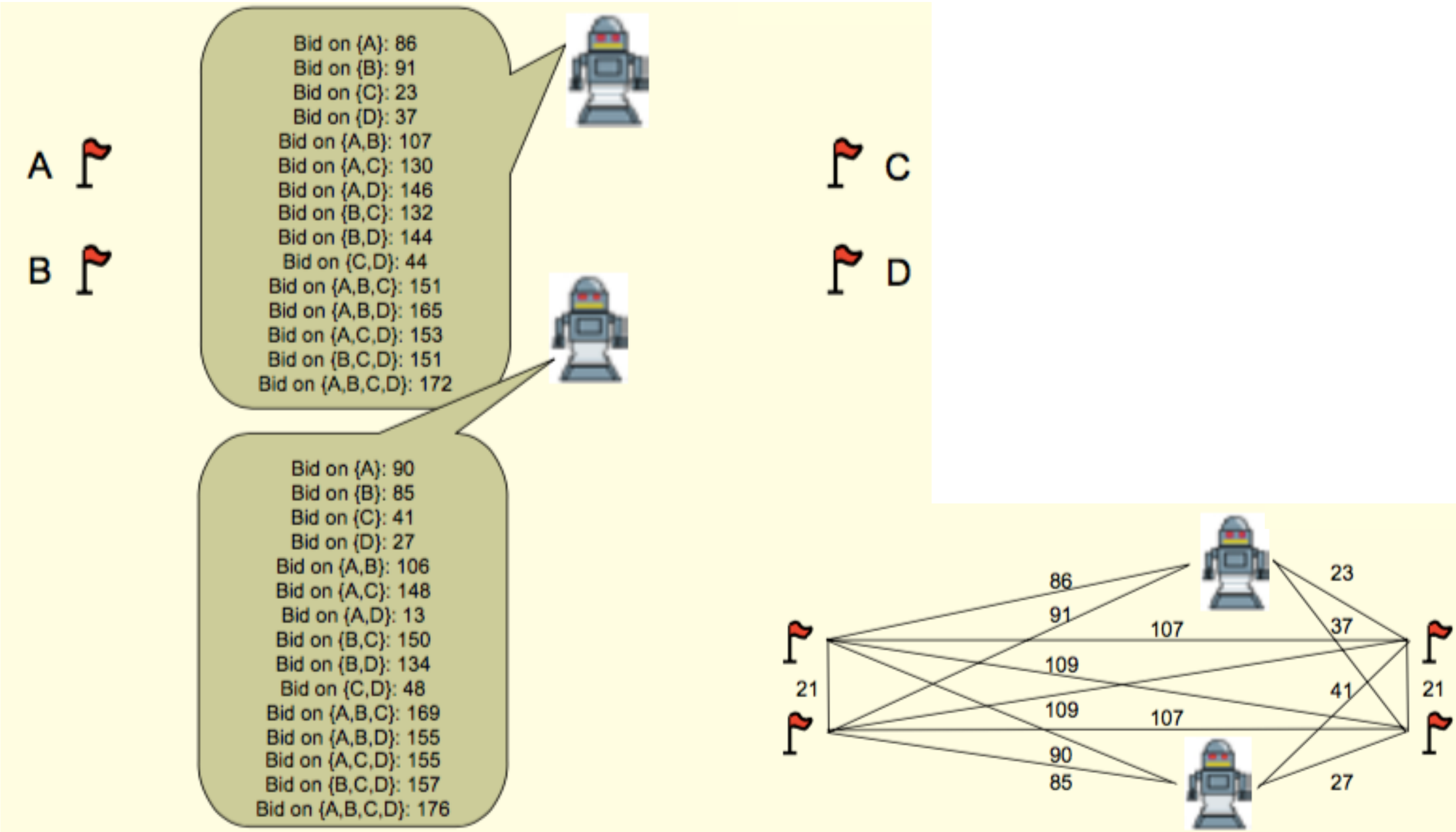
COMBINATORIAL AUCTIONS: *IDEAL* SCENARIO

- Each robot bids on ***all* bundles** (= subsets) of tasks
- Each robot gets assigned at most one bundle, with the **goal** of:
  - *Maximizing the number of tasks assigned* to the robots [first priority]
  - *Minimizing the total team cost* (= sum of the bids of the bundles won by robots) [second priority]
- Each robot determines a cost-minimal path to service all tasks (visit all targets) it has been assigned to, and follows it

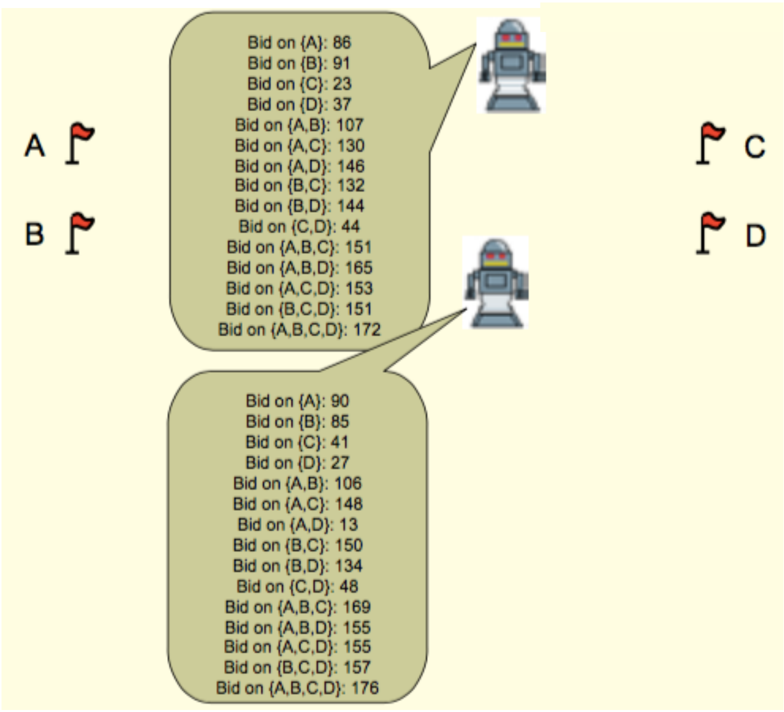


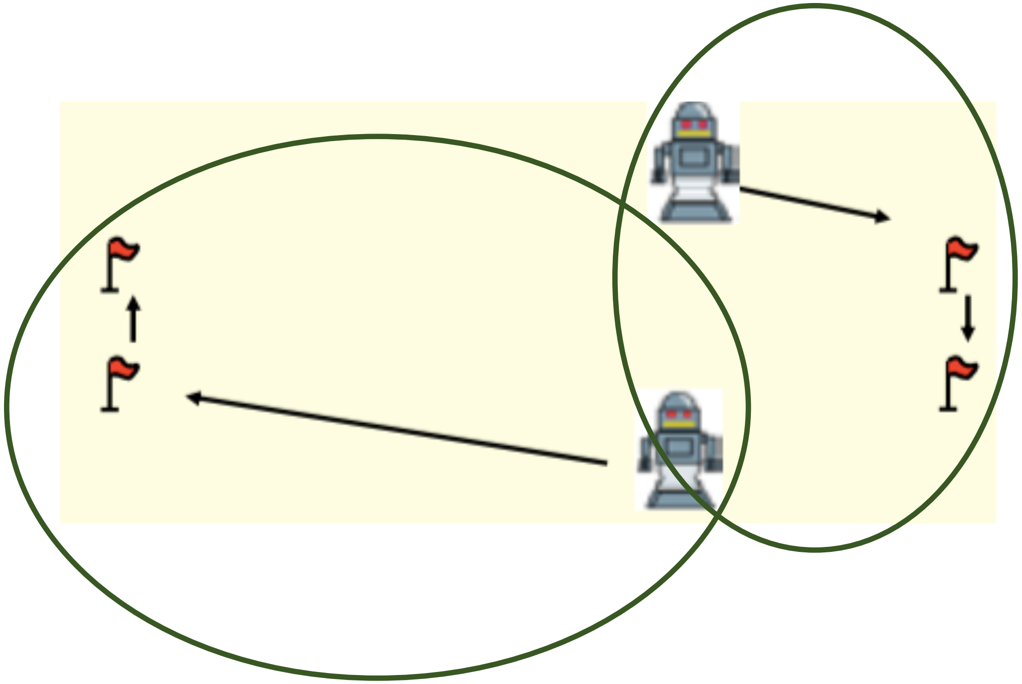
- Each robot bids on a bundle the **minimal path cost** it needs from its current location to service all tasks in the bundle
- → *Synergies are accounted for!*

Combinatorial Auctions: *Ideal* scenario



Combinatorial Auctions: *Ideal* scenario





- The team cost resulting from *ideal combinatorial auctions* is minimized since all synergies between tasks are accounted for **solving an NP-hard problem**
- The number of bids is exponential in the number of tasks
- Bid generation, bid communication and winner determination are expensive

Combinatorial Auctions: Scenario in *Practice*

- Each robot bids on ***some*** **bundles** (= subsets) of tasks
- Each robot gets assigned at most one bundle, with the **goal** of:
  - *Maximizing the number of tasks assigned* to the robots [first priority]
  - *Minimizing the total team cost* (= sum of the bids of the bundles won by robots) [second priority]
- Each robot determines a cost-minimal path to service all tasks (visit all targets) it has been assigned to, and follows it
- The team cost resulting from practical combinatorial auctions is expected to be small but can be suboptimal
- Bid generation, bid communication and winner determination are still relatively expensive

Combinatorial Auctions: Bidding Strategies

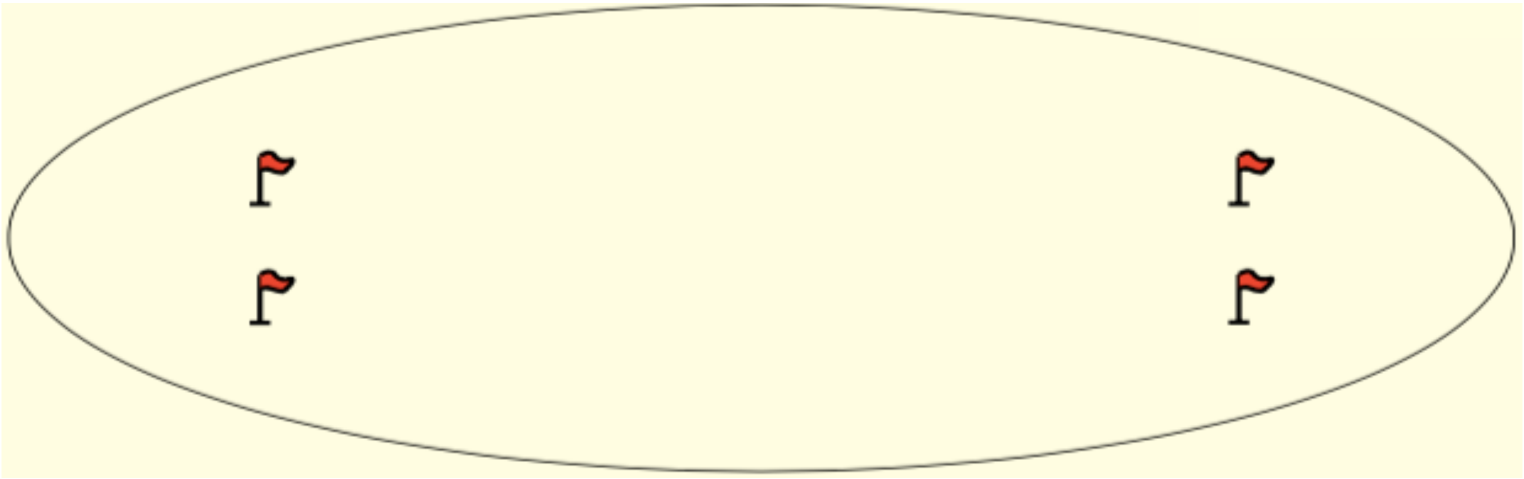
- Which bundles to bid on is mostly unexplored in economics because good bundle-generation strategies are usually <u>domain dependent</u>
- E.g., for multi-robot routing tasks one wants to exploit the spatial relationship of targets, but for other types of tasks different relations would make more sense
- <u>Good bundle-generation strategies:</u>
  - generate a small number of bundles
  - generate bundles that cover the solution space
  - generate profitable bundles
  - generate bundles efficiently
  - ....
- <u>Basic (*dumb*) domain-independent bundle-generation strategy:</u>
  - Generate (some of) all n-tasks bundles, e.g., all 3-targets subsets

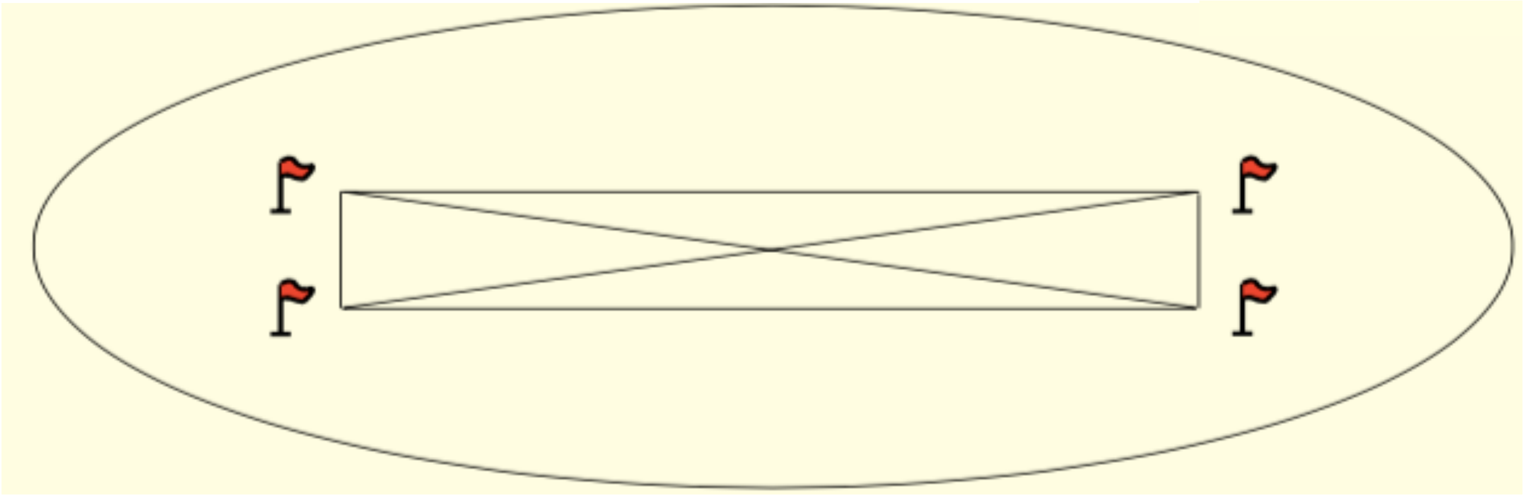
Combinatorial Auctions: Domain-Dependent Bundle Generation

- In our multi-robot routing problem, spatial relationships between tasks play an important role determining the cost of a bundle → Smart bundle generation can be obtained by spatial clustering of tasks

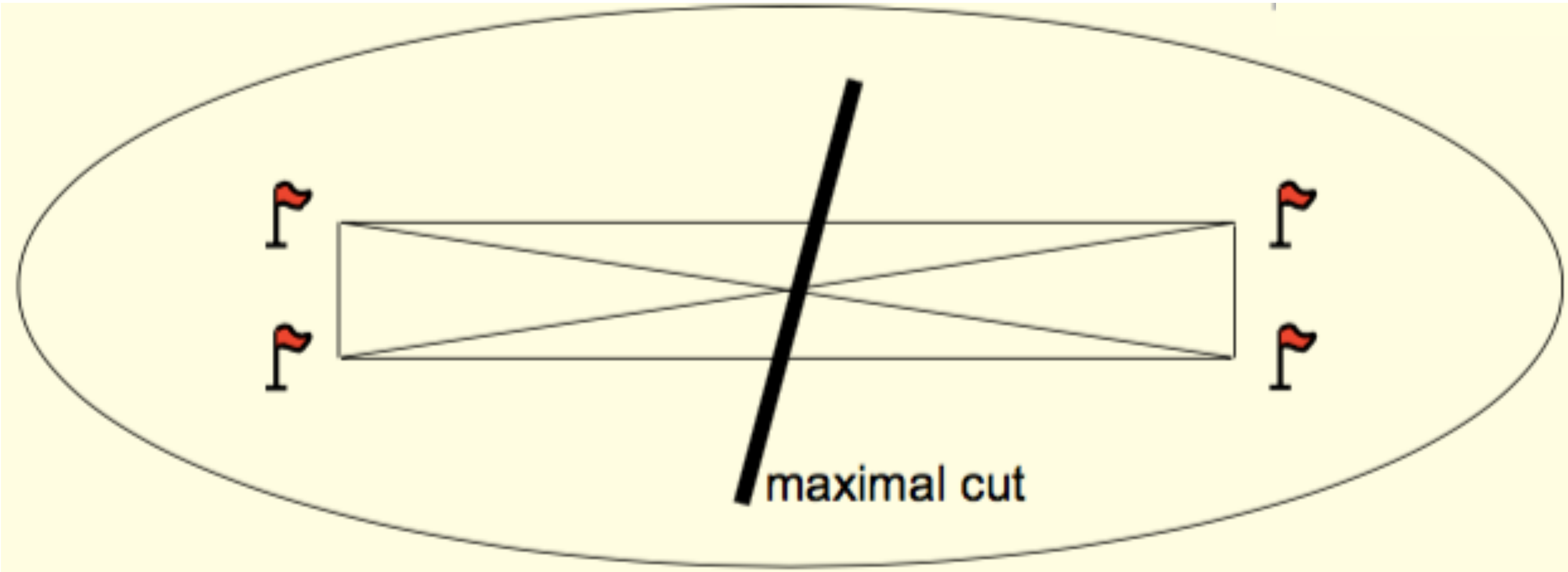
- **Procedure GRAPH-CUT:**
  - Start with a bundle that contains *all targets*
  - Bid on the new bundle
  - Build a *complete graph* whose vertices are the tasks in the bundle and edge costs correspond to the path costs between the vertices
  - Split the graph into two sub graphs along (an approximation of) the maximal cut
  - Bid on the two bundles
  - Recursively *repeat the procedure twice*, namely for the tasks in each one of the two sub graphs (bundles)

Combinatorial Auctions: Domain-Dependent Bundle Generation



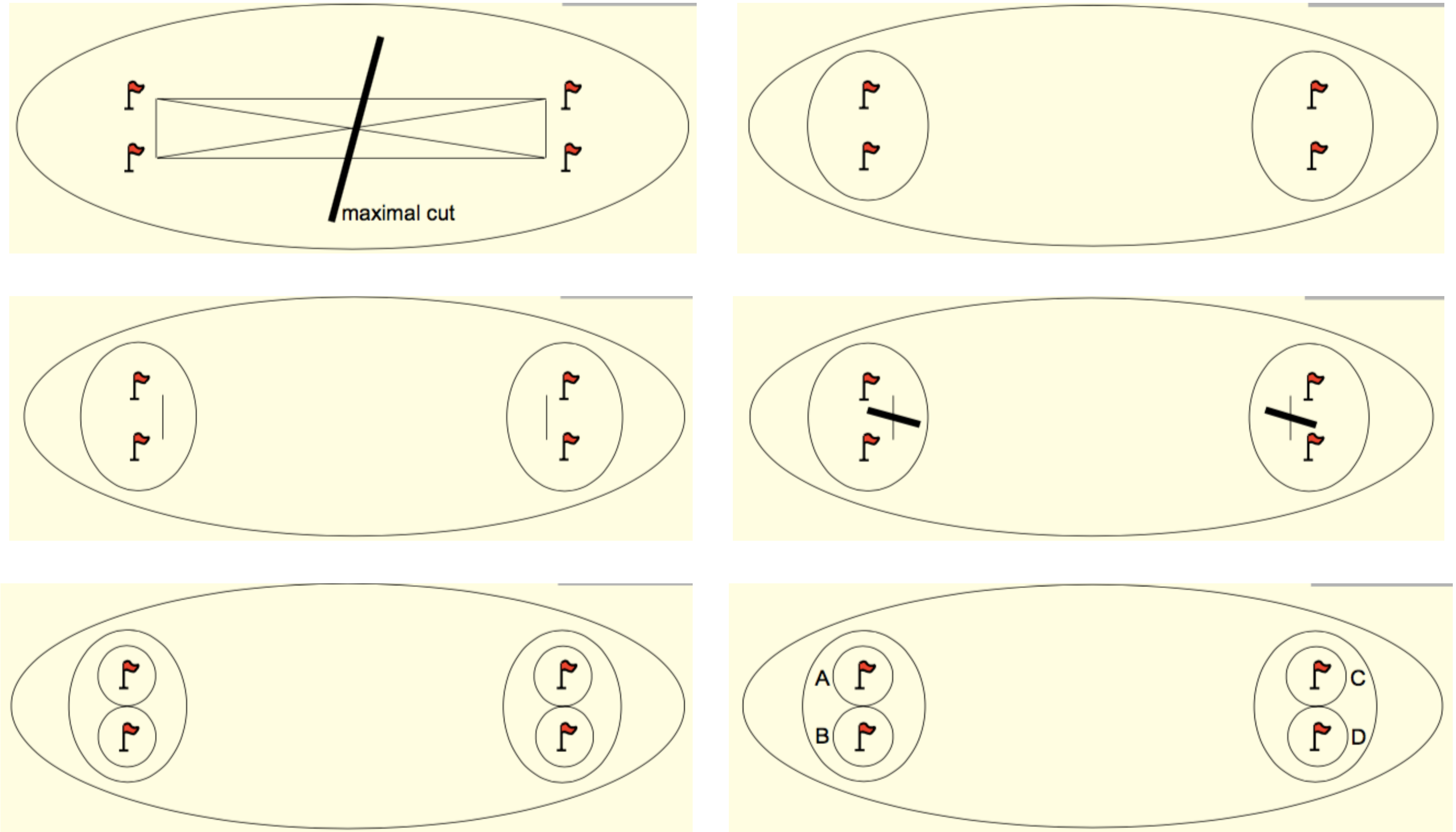


Combinatorial Auctions: Domain-Dependent Bundle Generation



- **Cut** = A partition of the vertices of a graph in <u>two disjoint sets</u>
- **Weighted Maximal Cut** (= weighted maxcut) = cut that maximizes the sum of the costs of the edges that connect the two sets of vertices
- In our case, this means to <u>avoid expensive partitions</u>
- Finding a maximal cut is ***NP-hard*** and needs to get <u>approximated</u>

Combinatorial Auctions: Domain-Dependent Bundle Generation



Submit bids for bundles: {A}, {B}, {C}, {D}, {A,B}, {C,D}, {A,B,C,D}

Numeric experiment

- 3 robots in known terrain with 5 clusters of 4 targets each

	Number of bids	Team cost (sum)
Parallel single—item auctions	635	426
Combinatorial auctions with fixed 3-bundles	20506	248
Combinatorial auctions with GRAPH-CUT	1112	184
Optimal combinatorial auctions (with MIP)	N/A	184

Combinatorial Auctions: Summary

- Ease of implementation: difficult
- Ease of decentralization: unclear (depends on task scenario)
- Bid generation: expensive
  - Bundle generation: expensive (can be NP-hard)
  - Bid generation per bundle: can be NP-hard
- Bid communication: expensive
- Auction clearing: expensive (NP-hard)
- Team performance: very good (<u>optimal</u>)
  - Many (all) synergies taken into account
- <u>Workarounds:</u>
  - Use a smart bundle generation method
  - Approximate the various NP-hard problems

Sequential Auctions

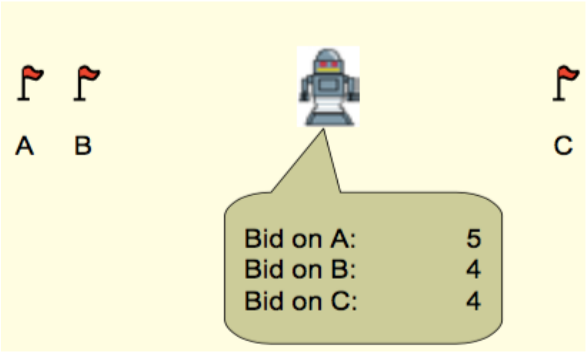
Parallel Auctions	Combinatorial Auctions
Ease of implementation: simple	Ease of implementation: difficult
Ease of decentralization: simple	East of decentralization: unclear
Bid generation: cheap	Bid generation: expensive
Bid communication: cheap	Bid communication: expensive
Auction clearing: cheap	Auction clearing: expensive
Team performance: poor	Team performance: "optimal"

- **Sequential auctions:** a good *trade-off* between parallel auctions and combinatorial auctions
- Several bidding rounds, until all tasks have been assigned to robots
  - Only one task is assigned in each round
- During each round, each robot bids on all tasks not yet assigned
  - The minimum bid over all robots and tasks wins, and the corresponding robot gets the corresponding task
- Each robot determines a cost-minimal path to service all tasks it has been assigned, and follows it

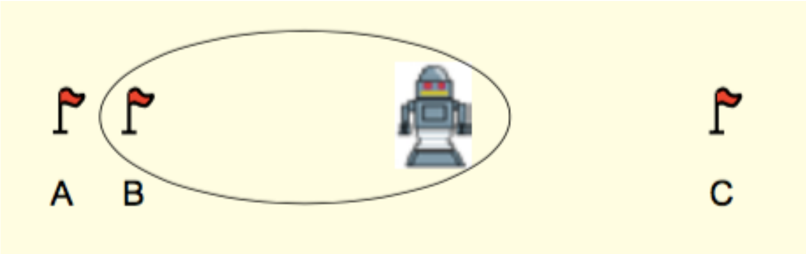
Sequential Auctions: Exploiting Synergies

- Each robot bids on a task. The amount of the bid is chosen to optimize *team performance.* This can be realized in many different ways, depending on what performance is of interest (e.g., total time, total energy/traveling, ...)
- E.g., *Performance:* Minimize the sum of path costs for all robots → A robot bids the minimal increase in path cost it needs from its current location to visit all of the targets it has been assigned so far + the new task

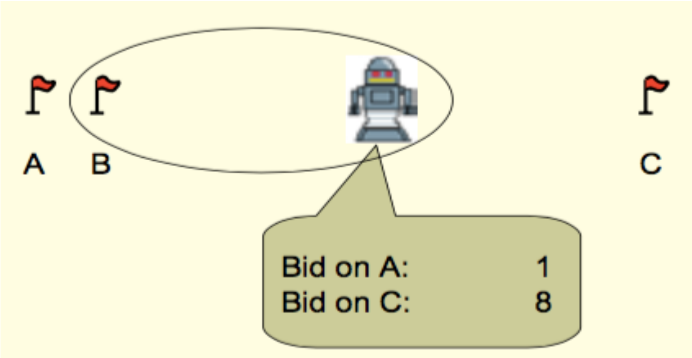
Initial bid, on all individual tasks



Robots has won Task B

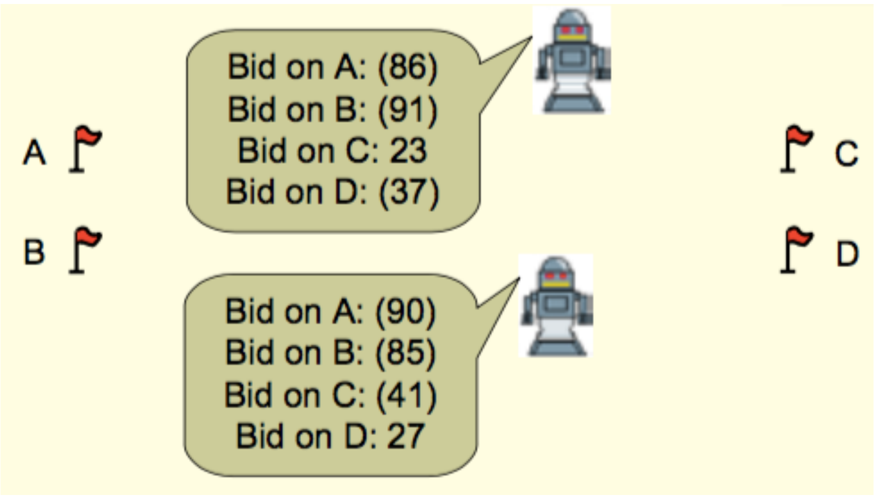


Robots new bids on unassigned Tasks

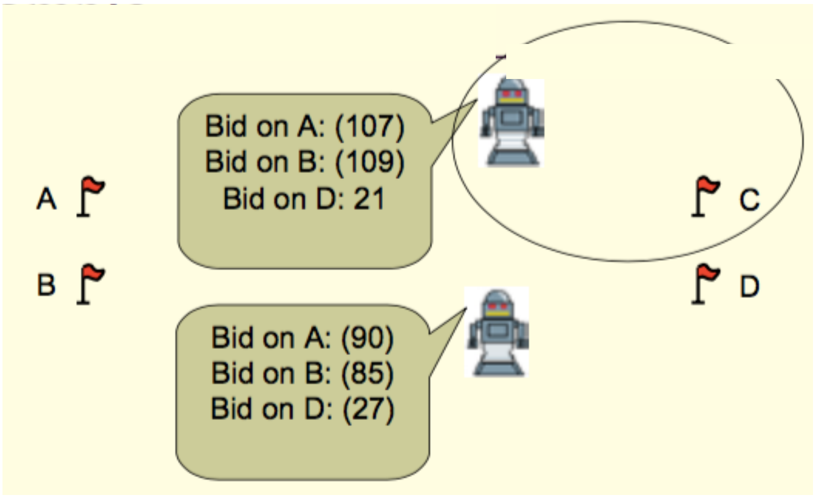


Sequential Auctions: Another Example

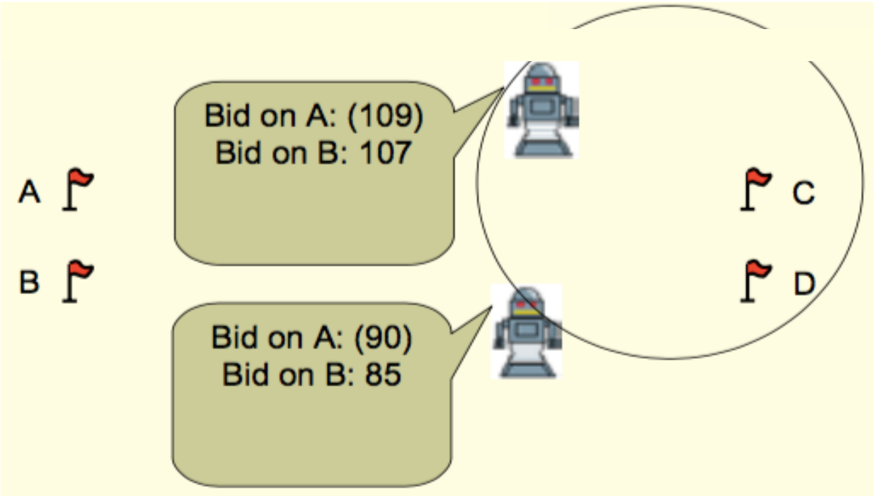
Initial bids, on all individual tasks



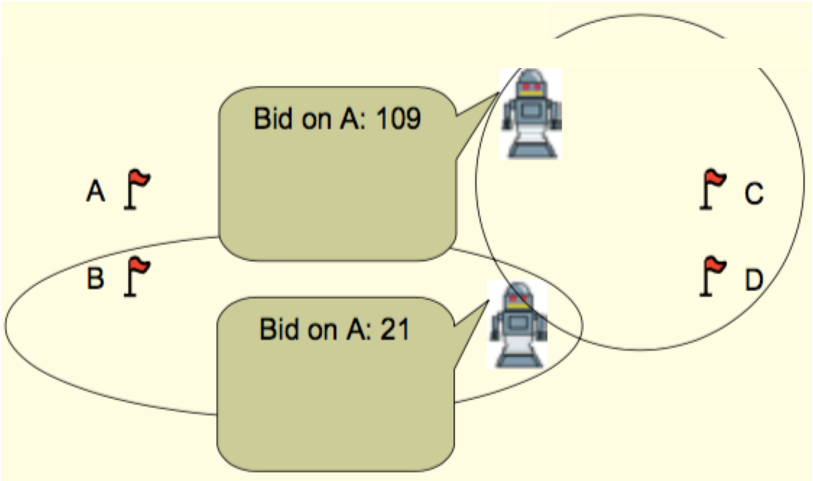
After one task assignment



After two task assignments

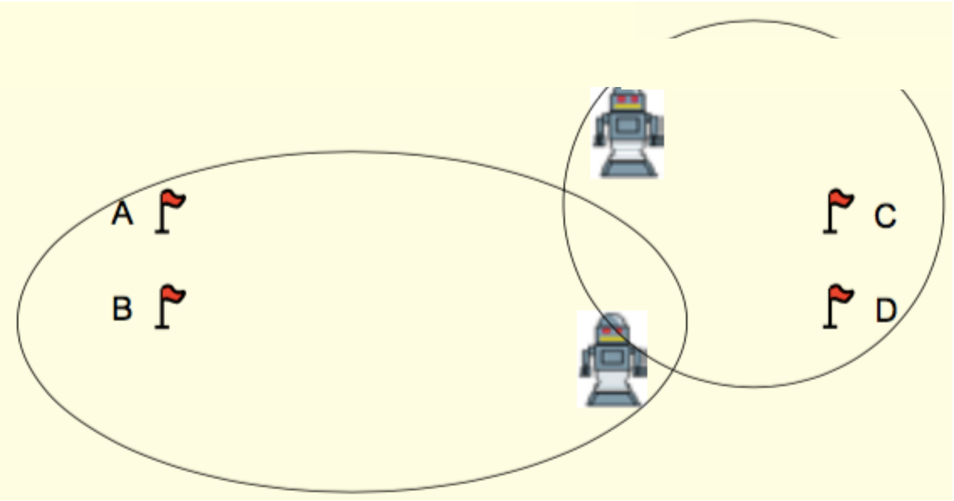


After three task assignments

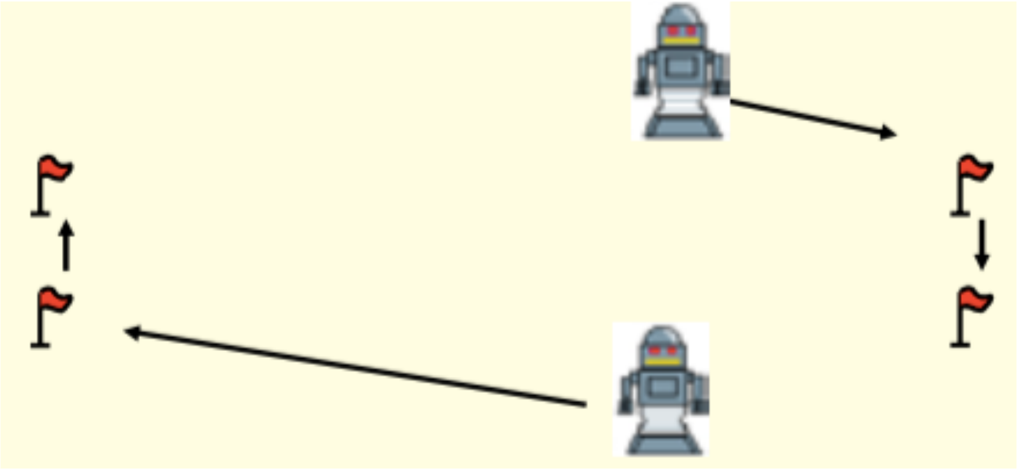


Sequential Auctions: Another Example

Complete task assignment



Robot paths

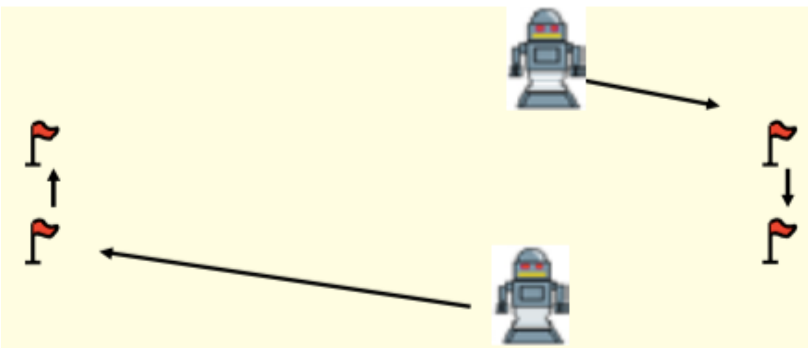


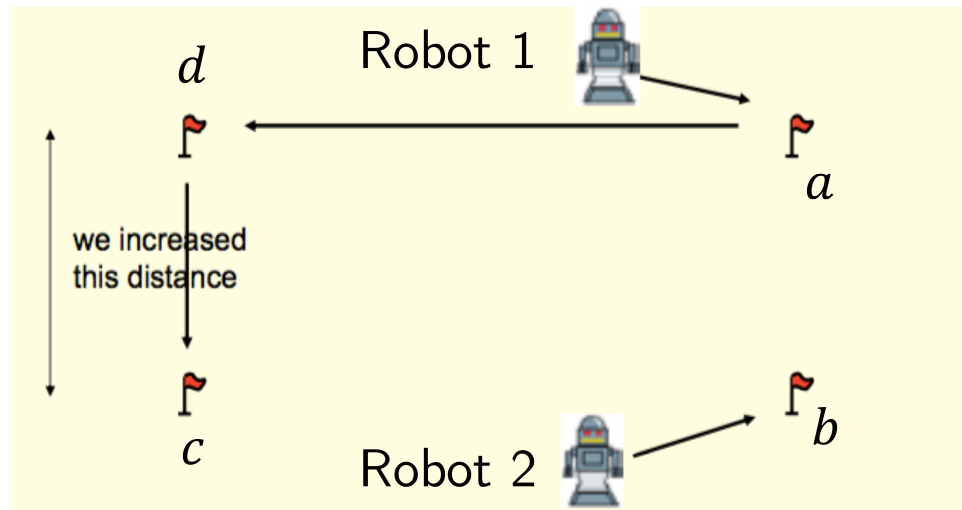
Sequential Auctions: Remarks

- Each robot needs to submit <u>only one of its lowest bid</u>
- 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)
- → <u>Each robot submits at most one bid per round</u>, and the **number of rounds equals the number of tasks**
- → <u>The total number of bids is no larger than the one of parallel auctions</u>, and <u>bid communication is cheap.</u>
- The bids that do not need to be submitted were shown in parentheses in previous examples

Sequential Auctions: Remarks

- Not always capable to exploit synergies because of the sequential nature of the process
- **No guarantees of optimality**





Robot 1: a
Robot 2: b
Robot 1: d
Robot 1: c

Sequential Auctions: Summary

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

Sequential Auctions: Bidding Rules

- Robots' bids should be quantified to let a robot win a task so that some selected measure of *team performance is optimized*
- Let's assume that team performance is in terms of costs minimization → A robot's bid should be related to the increase of some measure of team cost, such that the best bid increases the cost (of the team) least
- Robot r bids on task 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.)
- Team cost minimization can be achieved in a fully distributed way: if each robot bids to minimize its cost difference, it actually minimizes the cost different for the team

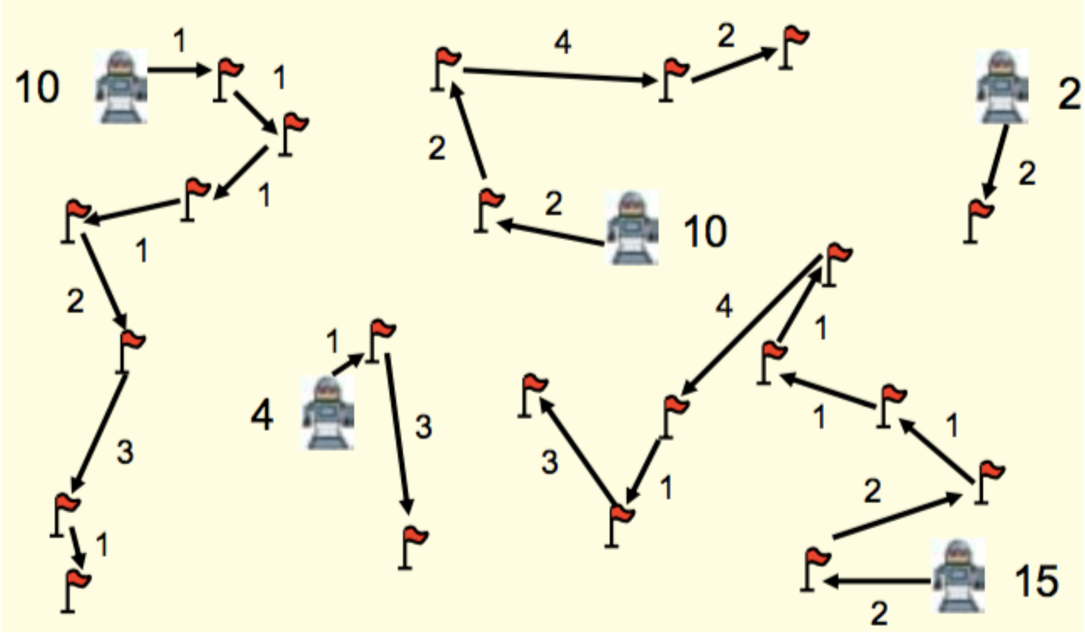
Team Performance Criteria

- **MiniSum**
  - *Team goal:* Minimize the sum of the path costs over all robots
  - → Minimization of total used *energy* or traveled *distance*
  - Application: logistics / goods delivery, planetary surface exploration
- **MiniMax**
  - *Team goal:* Minimize the maximum path cost over all robots
  - → Minimization of total completion time (*makespan*)
  - Application: facility surveillance, mine clearing, time-critical task set
- **MiniAvg**
  - *Team goal:* Minimize the average arrival time over all targets
  - → Minimization of average service time (*flowtime*)
  - Application: search and rescue

MiniSum and Bidding Rules in Seq Auctions

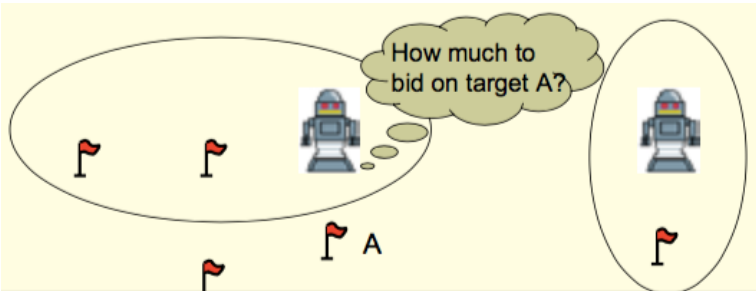
- **MiniSum**
  - *Team Goal:* Minimize the sum of the path costs over all robots
  - → Minimization of total used *energy* or traveled *distance*
  - Application: logistics / goods delivery, planetary surface exploration

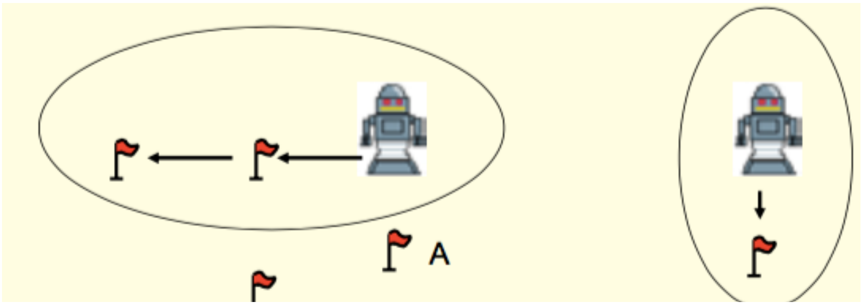
E.g., Team performance: 10+10+2+4+15 = 41



MINISUM AND BIDDING RULES IN SEQ AUCTIONS

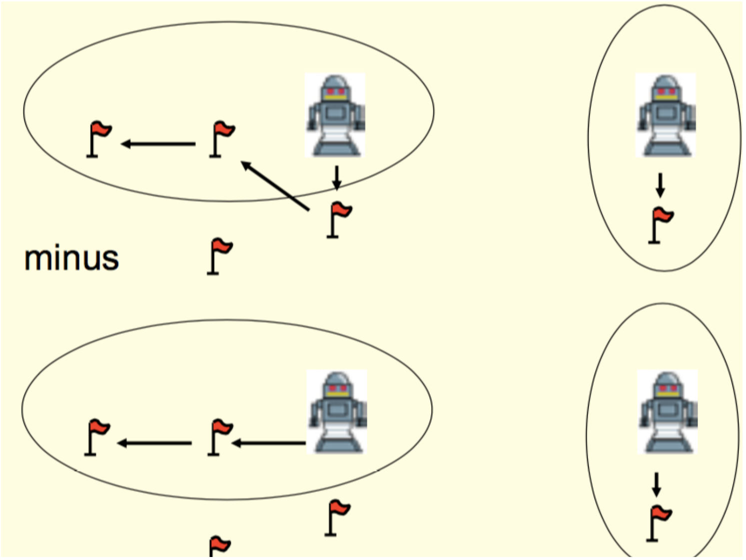
- *Bids ↔ Team Goal:* Minimize the sum of the path costs over all robots





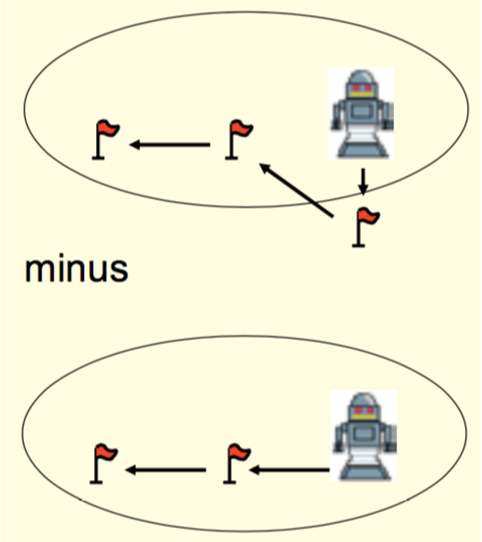
Current allocations and costs

Bid that increases the team cost the least



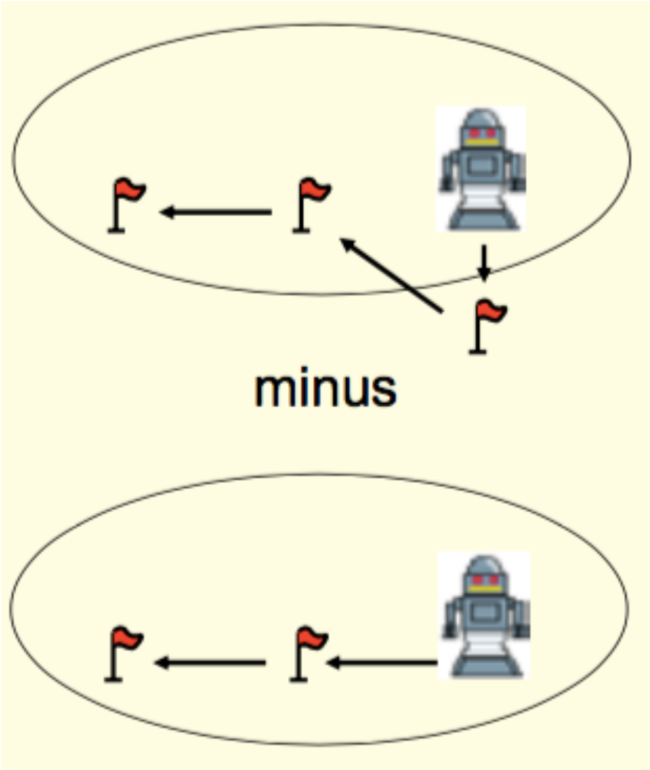
Equivalent to

A robot doesn't need to know about other robots



MiniSum and Bidding Rules in Seq Auctions

- *Bids ↔ Team Goal:* Minimize the sum of the path costs over all robots



- 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

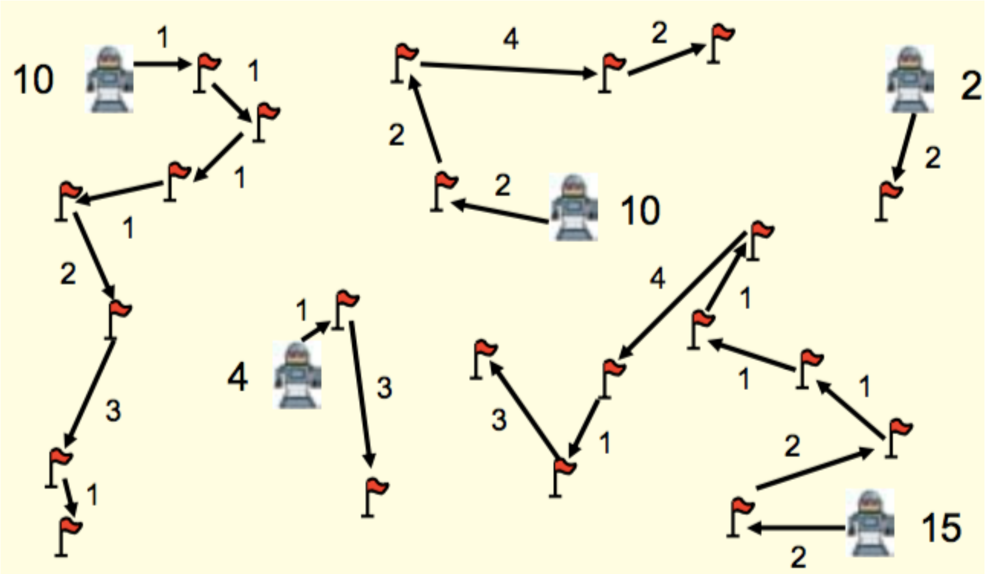
- minimal path cost the robot needs from its current location to visit all targets it has won so far

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 bidding on (***BidSumPath***)

MiniMax and Bidding Rules in Seq Auctions

- **MiniMax**
  - *Team goal:* Minimize the maximum path cost over all robots
  - → Minimization of total completion time (*makespan*)
  - Application: facility surveillance, mine clearing, time-critical task set

E.g., Team performance: $\max(10, 10, 2, 4, 15) = 15$

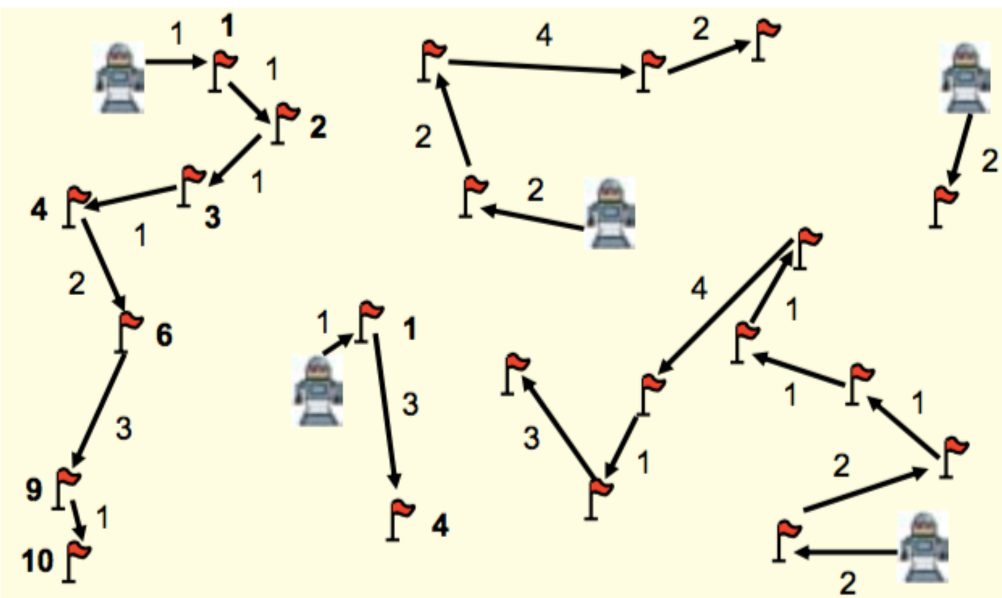


- **Bid** the minimal path cost the robot needs from its current location to visit all targets it has won *if* it wins the target it is bidding on (***BidMaxPath***)
- This bid automatically balances the path costs of all team robots, without the need for knowing about their bids

MiniAvg and Bidding Rules in Seq Auctions

- **MiniAvg**
  - *Team goal:* Minimize the average arrival time over all targets
  - → Minimization of average service time (*flowtime*)
  - Application: search and rescue

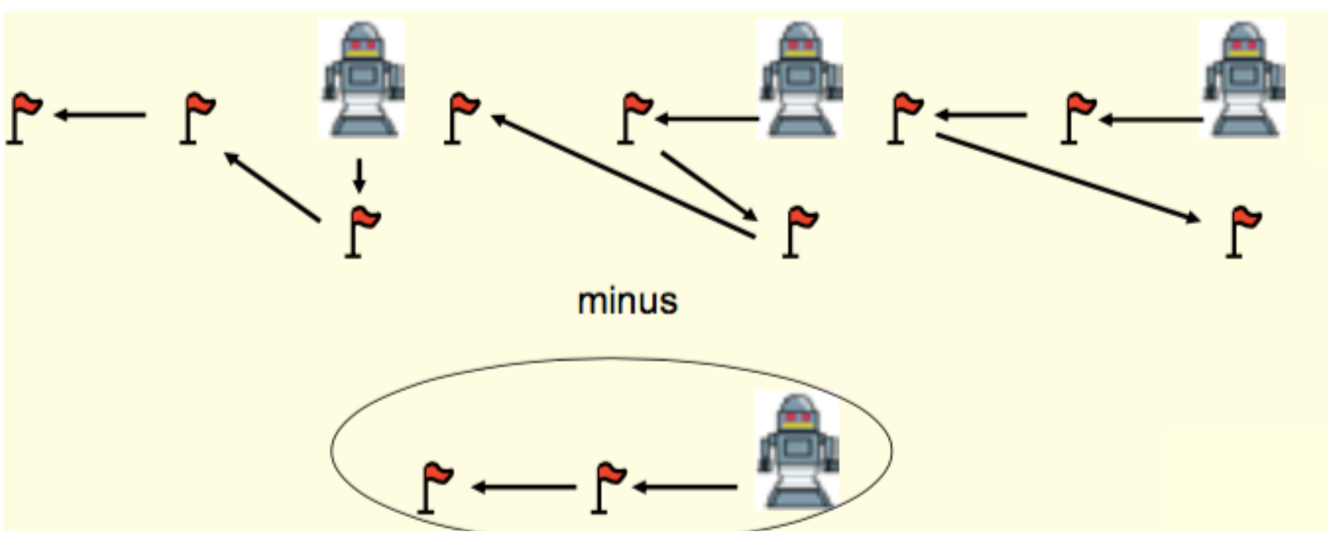
E.g., Team performance: (1+2+3+4+6+9+10+1+4+...)/22 = 5.8



- **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 bidding on (***BidAvePath***)
- This bid automatically balances the avg of the arrival times of all team robots, without the need for knowing about their bids

Computing Optimal Bids is *NP-Hard*

- **Minimal path cost over a set of assigned tasks → *TSP !***
  - For large sets of tasks it might be intractable
  - Even for small subsets, if bidding has to be done online in real-time, optimal computation of bids might be unfeasible
  - A number of heuristics: *2-Opt, 3-Opt, ACO, NN, Insertion heuristics*
- ***Full task allocation problem: Multi-robot routing → VRP***

Minimum (

…)

Complexity of auction mechanisms

- **Time complexity** (amount of computation)
  - (Distributed) Bid computations, in a single auction
  - + Winner determination, in a single auction
  - + Number of auctions required to assign all tasks
- **Communication complexity** (bandwidth for information exchanges)
  - Call for bids from the auctioneer
  - + Bids submission from the agents to the auctioneer
  - + Awarding tasks to winners (may or may not inform losers in addition to winners)

Solution Quality (team cost) → It depends whether the above complexity it allows to deal with all subproblems in optimal way or not

Time Complexity

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

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

Results from:

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

Communication Complexity

Auction type	Auction call (from auctioneer)	Bid submission (from agents)	Task awards (from auctioneer)	Task awards (+ ¬awarded)
Single—item	$O(r)$	$O(r)$	$O(1)$	$O(r)$
Multi-item	$O(rn)$	$O(rn)$	$O(m)$	$O(r)$
Combinatorial	$O(rn)$	$O(r \cdot 2^n)$	$O(n)$	$O(r+n)$

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

Optimal Task Allocation: VRP

$$\min_{\boldsymbol{x}} \sum_{i \in V_T \cup V_R,\ j \in V_T} c_{ij} x_{ij}$$ Performance function (for MiniSum)

$s.t.$

$$\sum_{i \in V_T \cup V_R} x_{ij} = 1 \quad \forall j \in V_T$$ Each target vertex is entered exactly once

$$\sum_{j \in V_T} x_{ij} \leq 1 \quad \forall i \in V_T \cup V_R$$ Each (robot or target) vertex is left at most once

$$\sum_{i,j \in U} x_{ij} \leq |U| - 1 \quad \forall U \subseteq V_T \colon |U| \geq 2$$ Sub-tour elimination

V_R = Set of robot vertices
V_T = Set of task / target vertices
Variables: x_{ij} = edge (i, j) is in the solution
c_{ij} = Path cost from vertex i to vertex j

Note: this is referred to as a MIP (Mixed Integer Programming model) in the following

Worst-Case deviation from optimality

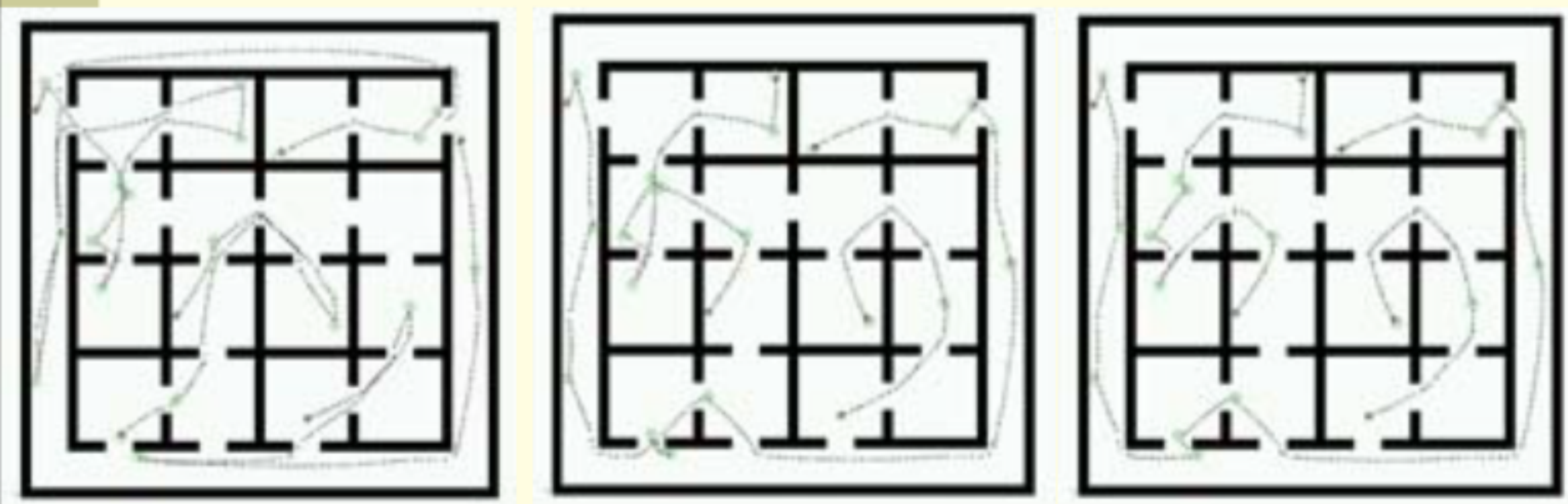
When using the mentioned bidding rules in sequential auctions, it is possible to compute the worst-case loss bounds (lower and upper) with respect to the optimal allocation computed using the optimization model (e.g., 1.5 means that using the BidSumPath bidding rule to optimize the MiniSum criterion, in the worst-case the team cost is at least 1.5 times the optimal team cost)

Bidding rule	Team Performance Criterion					
	MiniSum Lower - Upper		*MiniMax* Lower – Upper		*MiniAvg* Lower - Upper	
BidSumPath	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxPath	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega\left(m^{1/3}\right)$	$2m$
BidAvgPath	m	$2m^2$	$\frac{n+1}{2}$	$2m^2 n$	$\Omega\left(m^{1/3}\right)$	$2m^2$

EXPERIMENTAL RESULTS FOR SEQUENTIAL AUCTIONS

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

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



SUM = 426.98	SUM = 279.62	SUM = 271.04

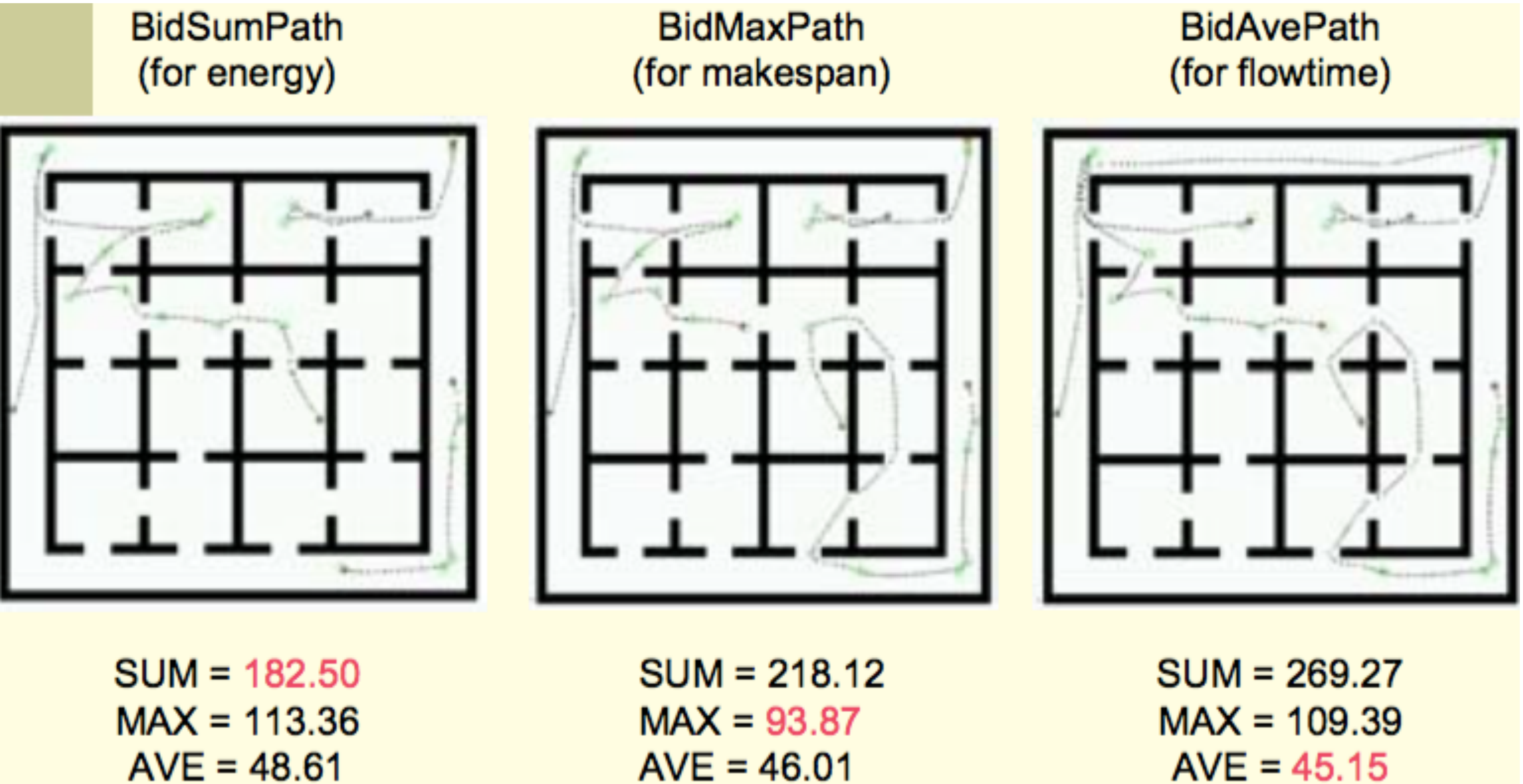
Experimental results for sequential auctions

- 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

Experimental results for sequential auctions

Impact of different bidding rules



TASK ALLOCATION SUMMARY

- Task Allocation as a model for coordination, division of labor, role assignment in multi-agent/robot systems
- General formalization and taxonomy of multi-robot task allocation (MRTA) problems
- Optimization models for different classes of TA problems
- Computational complexity of the different classes / models
- Basic solution approaches exploiting the optimization models
- Intentional vs. Emergent task allocation
- Distributed approaches
- Stigmergy-based (emergent) methods
- Market-based methods: auction models, properties of different auction models (parallel, combinatorial, sequential), complexity