Sequential Single-Cluster Auctions for Robot Task Allocation

Bradford Heap and Maurice Pagnucco

Never Stand Still | Faculty of Engineering | Computer Science and Engineering

ARC Centre of Excellence for Autonomous Systems
School of Computer Science and Engineering
The University of New South Wales

Scenario

Consider this:

- Gas leak in an office building
- Too dangerous to send in humans
- Team of 10 autonomous robots
- 50 rooms to check

How can we allocate those 50 rooms to those 10 robots such that all rooms are checked in the shortest time frame?

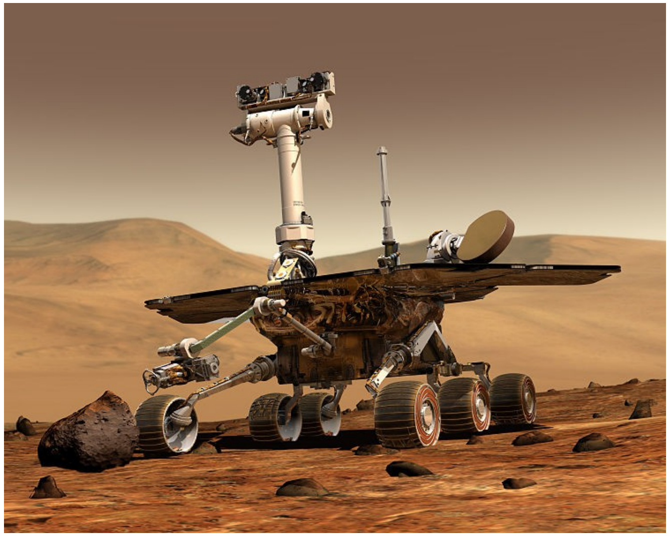
Multi-Robot Applications

Rescue Robots

- Shared global goal, individual tasks
- Often unknown terrain

Domestic Robots

- Heterogeneous tasks
- Known terrain with dynamic obstacles



Interplanetary Robots

- Hostile environment
- Limited knowledge about terrain

Introduction

The use of multiple robots gives us advantages over single robots such as robustness due to redundancy and efficiency due to parallelism.

However, optimal allocation of tasks amongst teams of robots is NP-Hard

Over the past decade much work has been done investigating methods of distributed task allocation in particular in the area of distributed auctions.

Background: Existing Auction Algorithms

Single-Round Combinatorial Auction (NP-Hard)

- Robots bid lowest path cost to complete all tasks in each bundle on all combinations of task bundles.
- Optimal Solution – takes in all synergies between tasks

Parallel Auction

- Robots bid lowest path cost to complete each task from current location in parallel
- Highly suboptimal – does not take in any synergies between tasks

Background: Existing Auction Algorithms

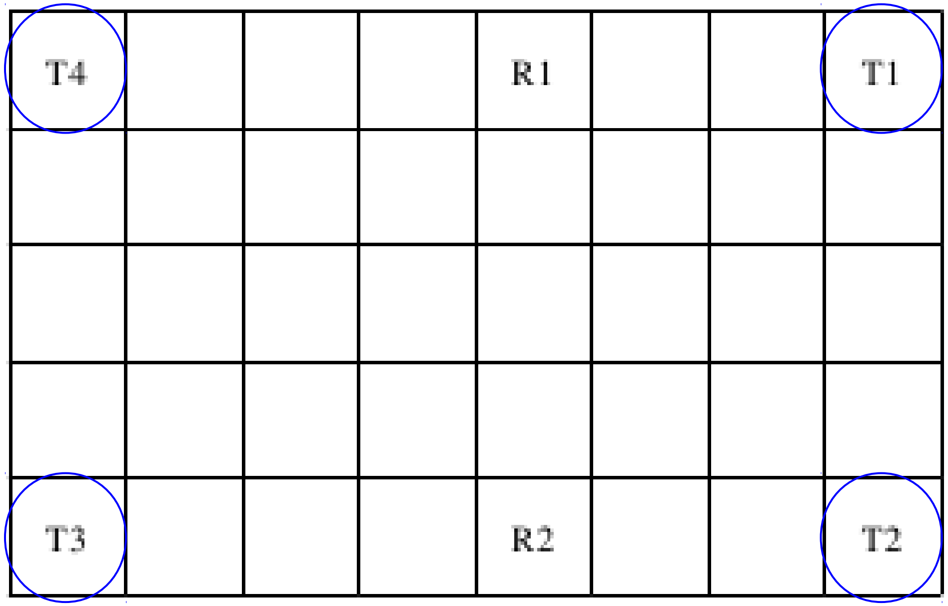
Sequential Single Item Auction (SSI)

- One task allocated per round
- Robots individually calculate bids for all unallocated tasks.
- Each task bid calculation considers the increase in path cost for doing that task – some synergy is considered.
- Lowest bid for any task wins that task.
- Repeat for unallocated tasks.

Sequential Single Item Auction with Bundle Bids

- Same as SSI but robots can bid for bundles of tasks like in Combinatorial except there is a max bundle size.
- Shown to be closer to optimal than SSI.

Auctions Summary



Combinatorial Solution:

R1 → T1 → T2

R2 → T3 → T4

Sum = 15 | # Bids = 30

Parallel Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 | # Bids = 8

SSI Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 |

Bids = 8

SSI Bundle Bids:

R1 → T1 → T2

R2 → T3 → T4

Sum = 20 |

Bids = 26

Auctions Summary

Combinatorial Solution:

R1 → T1 → T2

R2 → T3 → T4

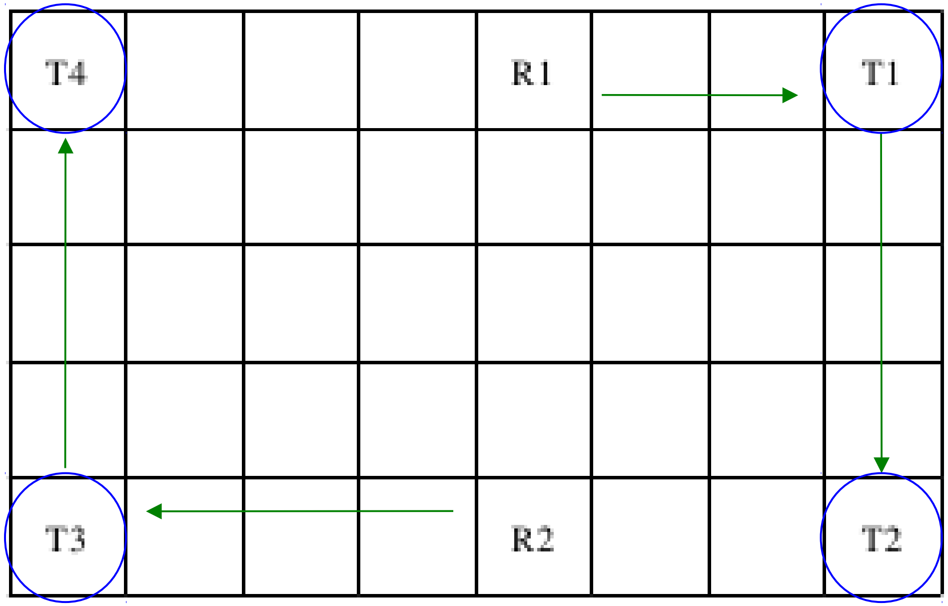
Sum = 15 | # Bids = 30

Parallel Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 | # Bids = 8



SSI Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 |

Bids = 8

SSI Bundle Bids:

R1 → T1 → T2

R2 → T3 → T4

Sum = 20 |

Bids = 26

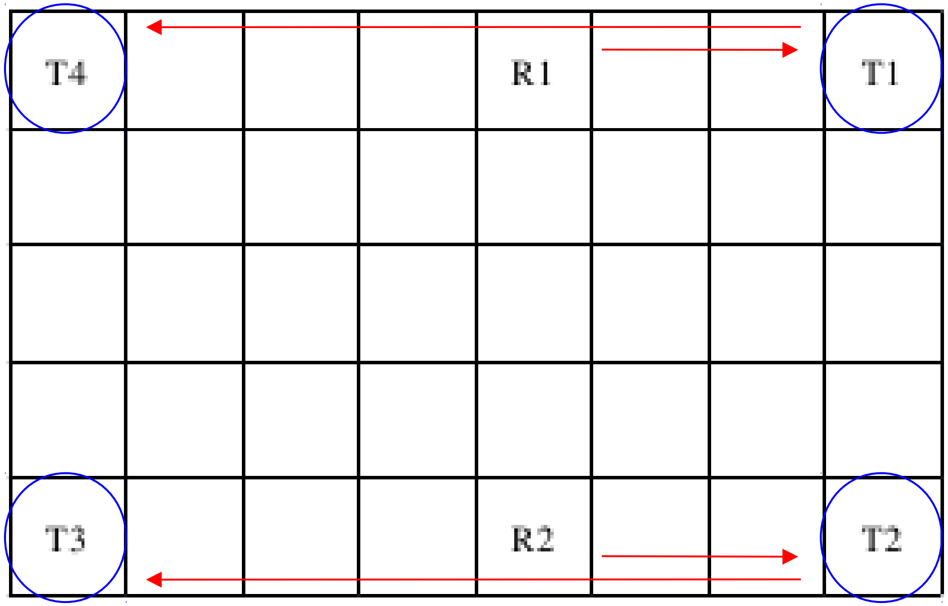
Auctions Summary

Combinatorial Solution:

R1 → T1 → T2

R2 → T3 → T4

Sum = 15 | # Bids = 30



Parallel Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 | # Bids = 8

SSI Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 |

Bids = 8

SSI Bundle Bids:

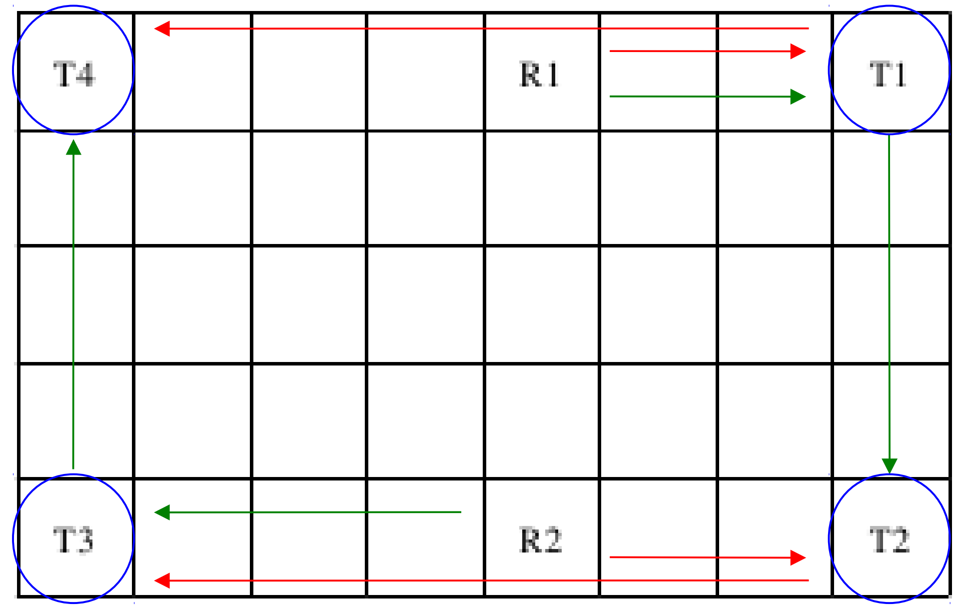
R1 → T1 → T2

R2 → T3 → T4

Sum = 20 |

Bids = 26

Auctions Summary



Combinatorial Solution:

R1 → T1 → T2

R2 → T3 → T4

Sum = 15 | # Bids = 30

Parallel Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 | # Bids = 8

SSI Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 |

Bids = 8

SSI Bundle Bids:

R1 → T1 → T2

R2 → T3 → T4

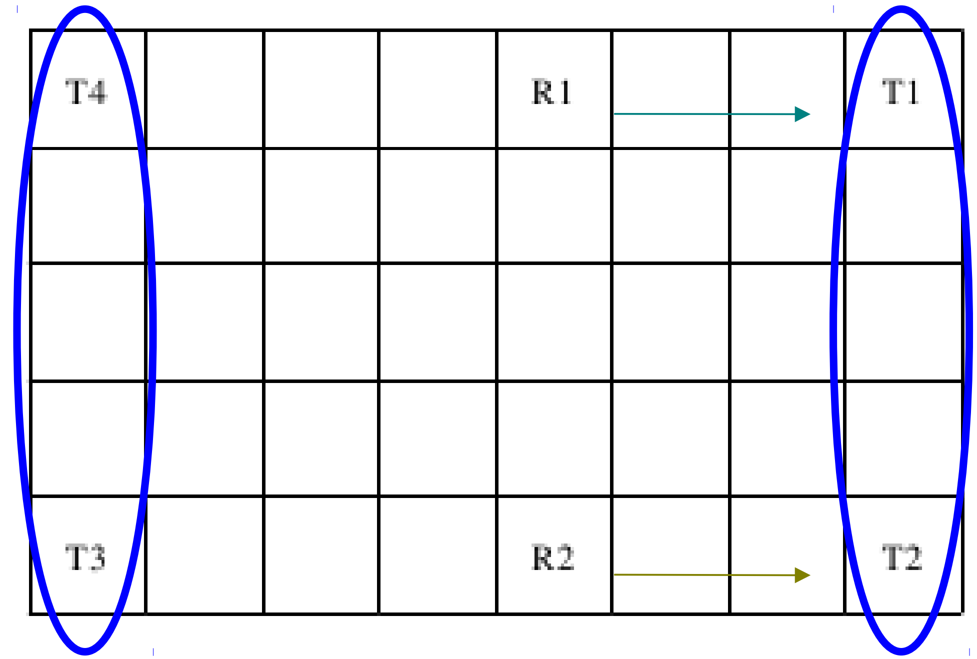
Sum = 20 |

Bids = 26

Clustering – Round 1

Our approach:

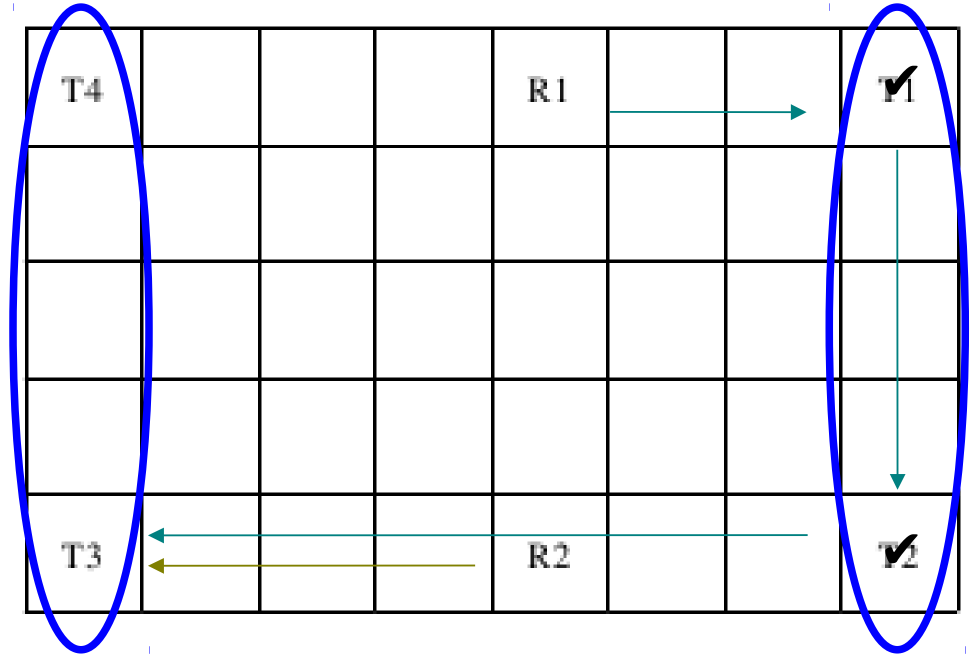
- Group tasks into "logical" clusters
- Each task is only assigned to one clusters
- Each cluster can have a varying number of tasks
- Run the equivalent of SSI Auction on clusters:
  - Where, robots bid on clusters of tasks, auction winner must complete all tasks in clusters



Clustering – Round 2

Our approach:

- Group tasks into "logical" clusters
- Each task is only assigned to one clusters
- Each cluster can have a varying number of tasks
- Run the equivalent of SSI Auction on clusters:
  - Where, robots bid on clusters of tasks, auction winner must complete all tasks in clusters



Auctions Summary

Combinatorial Solution:

R1 → T1 → T2

R2 → T3 → T4

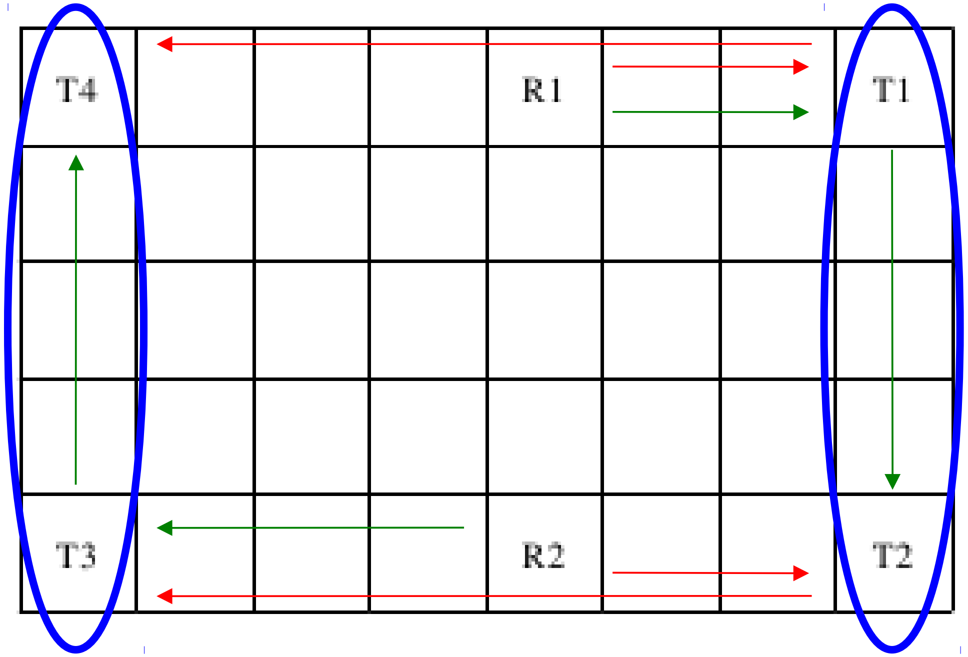
Sum = 15 | # Bids = 30

Parallel Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 | # Bids = 8



SSI Solution:

R1 → T4 → T3

R2 → T2 → T3

Sum = 20 |

Bids = 8

SSI Bundle Bids:

R1 → T1 → T2

R2 → T3 → T4

Sum = 15 |

Bids = 26

Cluster Solution:

R1 → T1 → T2

R2 → T3 → T4

Sum = 15 |

Bids = 4

Properties

1. The number of rounds in a SSC auction is no more than the number of rounds in a SSI auction.

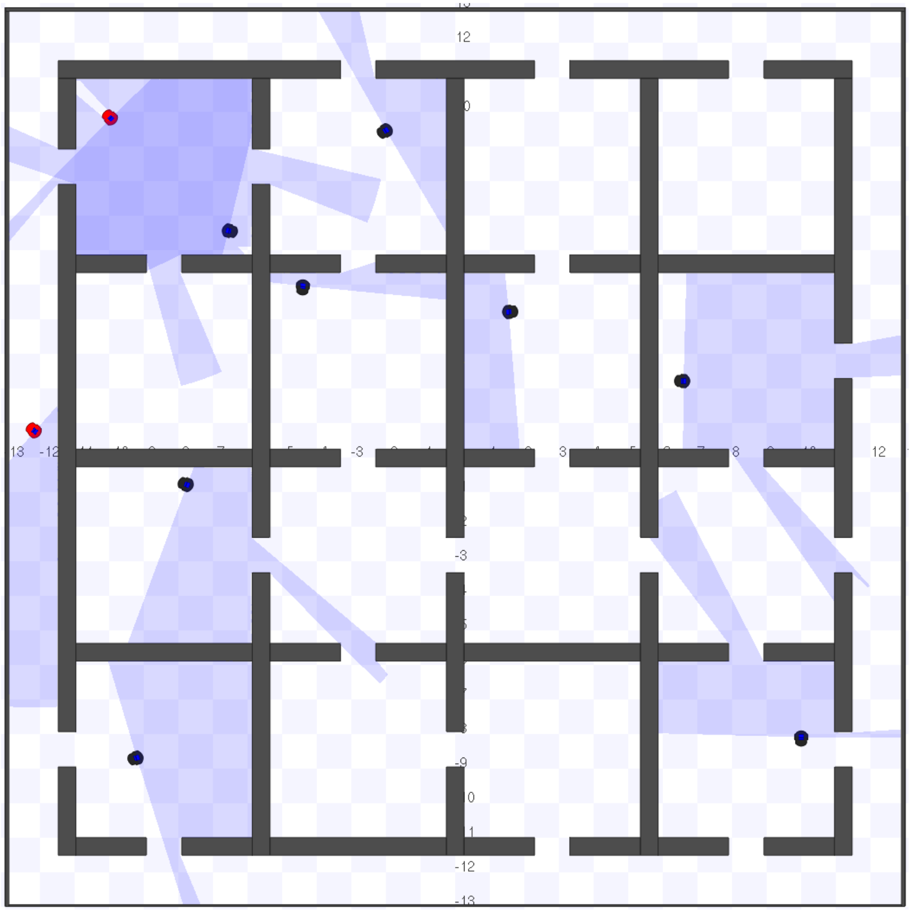
2. Winner determination time in a SSC auction is equal to the winner determination time in a SSI auction.

3. When clusters employ positive synergies between tasks the resultant team cost in a SSC auction is less than in a SSI auction.

Experiments

Environment:

- Standard Testbed
- 16 Rooms
- 40 Interconnecting Doors
- Doors fixed open/closed
- Robots only travel through open doors



Setup:

- Robots start in random locations
- 2 – 10 robots
- 6 – 60 tasks
- Robots have capacity constraints

Experiments

Tested with two team objectives:

- MiniMax – minimise the maximum distance a robot travels
- MiniSum – minimise the total distance all robots travel

K-Means clustering used with two different calculations for K.

Results compared with:

- Parallel auctions,
- SSI auctions,
- SSI with Bundle Bid auctions.

Main Results and Conclusions

We show:

- Mean improvement compared to SSI of 20 - 25% for MiniMax, 8 - 12% for MiniSum
- SSI with Bundle Bids performs better than Cluster Auctions
- However, SSI with Bundle Bids much slower than Clustering
- Cluster Auctions run almost as fast as SSI auctions

Future Work:

- Repeated auctions with dynamic re-clustering – allows robots to exchange uncompleted tasks while completing tasks.
- Ring auctions with clustering – robots grouped into teams and bid on tasks as a collective, then redistribute tasks amongst the team.

Questions

Sequential Single-Cluster Auctions for Robot Task Allocation

Bradford Heap and Maurice Pagnucco

Never Stand Still | Faculty of Engineering | Computer Science and Engineering

ARC Centre of Excellence for Autonomous Systems
School of Computer Science and Engineering
The University of New South Wales