Implementation of Sequential Single Item Auctions

UNSW COMP 3431
Assignment 2

Harold Jacob Hoare
z5021639
November 2015

Background

The general framework for sharing tasks amongst multiple agents is known as a contract net protocol[1]. The process is usually described as a type of auction and can be broken down into 5 phases,

1. Task announcements
2. Task announcement processing
3. Bidding
4. Bid processing
5. Contract processing, reporting results

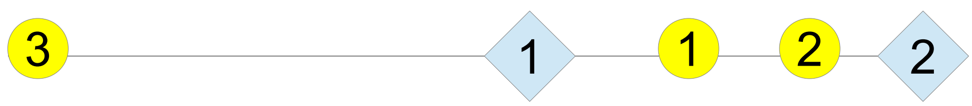
Here we concern ourselves primarily with bidding and bid processing.

The specific objective is to allocate a set of targets amongst a set of agents residing in a 2-dimensional world with a known map. The agents are tasked with collectively visiting all targets in the minimum amount of time.

The optimal solution to this problem can be arrived at by a single round combinatorial auction[2]. This requires every agent to bid on every combination of targets. The drawbacks of this approach are that the number of combinations is exponential in the number of targets and the bid for any combination is itself a NP-hard task of finding the optimal path for visiting the targets. Additionally, the determination of the winner is NP-hard.

Because of the computational complexity of single round combinatorial auctions a number of approximations have been devised that trade off calculation time against finding potentially less than optimal solutions. One such approximation is parallel single item auctions (PSIA), which take place over a number of rounds. Every agent bids its estimated time to reach every target in every round. The target is allocated to the agent with the lowest bid and the process is repeated for the remaining targets. The advantage of PSIA is that they run in polynomial time but the significant drawback is the lack of incorporation of synergies between targets.

As an example of this weakness, consider the 1-dimensional scenario below where the blue diamonds represent agents and the yellow circles represent the targets.



PSIA results in targets 1 and 3 being allocated to agent 1 and target 2 being allocated to agent 2. However we note that there is a synergy such that having moved to target 2, agent 2 can proceed to target 1. The optimal solution is for agent 2 to visit both targets 1 and 2 whilst agent 1 visits only target 3. In general PSIA can be an arbitrary factor away from the optimal solution[2].

Sequential single item auctions (SSIA) are an attempt to include synergies in the bidding process whilst avoiding the exponential time complexity of single round combinatorial auctions.

In SSIA each agent also bids on each target at every round. However the bid is the estimated time to visit the new target as well as the agent's already existing list of allocated targets. In terms of the 1-dimensional example above the process proceeds as follows,

- Round 1 : The winner is closest agent to any target, so agent 2 for target 2.
- Round 2 : Agent 2 incorporates the synergy from reaching target 2 in its bid for target 1 and wins target 1.
- Round 3 : Agent 3 wins target 3.

There are 2 possible modes for SSIA, MiniMax and MiniSum[3]. MiniMax is considered here and aims to minimise the time taken to visit all targets. The bids that each agent makes for each target are the time to visit all targets currently allocated to that agent plus the new target. In contrast MiniSum minimises the aggregate time (or distance) taken by all agents to visit all targets. MiniSum bids are the incremental increase in time to visit a new target in addition to the current path of allocated targets. When considering time rather than distance it is more natural consider MiniMax as we do here but the alternative MiniSum approach is very similar to implement.

Development Process

The ROS stage simulator[4] was used to develop an initial auction framework with the intention then to port this to actual turtlebots to execute the target visits, each turtlebot running its own adaptive monte carlo (AMCL) navigation in its own namespace. However when running multiple turthlebots across the network we experienced delayed laser scan and transform data which resulted in an inability for the turtlebots to accurately localise and plan. Hence the target visit process was also implemented in stage.

Bid Calculation

A simple bidding quantity would be the straight line distance between agent start position and target. This forms a lower bound on the distance but may be highly inaccurate in real environments due to obstacles and walls. By enhancing the distance to be the actual path followed by the agent when moving between start position and target we accurately align the bidding with the actual movement. It is critical here that the bidding reflects the actual agent motion. If different algorithms are used to calculate bids for the auction and then to move the agents between targets we introduce a source of error and the auction result may not be the minimum cumulative distance. Here we use the ROS move_base package make_plan service to produce paths (lists of poses) which are then used to calculate bids. By using the same move_base package to perform the motion we reduce the separation between bids and actual movement.

Instead of minimising distance this project aims to minimise the time taken to visit all targets. The first added complexity by using time is consideration of angular rotation. Each agent's initial rotational pose and each target rotational pose is arbitrarily chosen to be along the positive x direction, thus to move to a target in the negative x direction requires 360 degrees of angular rotation. When calculating the bid to move to a location the cost of angular rotation between each pose in the sequence was added to the linear movement cost.

Although by combining cumulative distance with cumulative rotation we arrive at a more accurate model there are further considerations of linear and angular acceleration as well as the smoothing employed by the global path planning. The problem of selecting a trajectory that minimises travel time can be approached by Delaunay triangualtion of the space followed by A* search amongst paths created by parameterized cubic splines, then path smoothing by waypoint tuning by gradient descent[5]. In light of the complexity of such approaches we make the simplifying assumption that all motion takes place the full linear or angular speed. We return to the issue of the accuracy of this assumption later.

Testing Environments

3 different environments were considered,

1. A realistic floorplan derived by mapping an actual building. This is a rich environment consisting of rooms, corridors and irregularly shaped obstacles as shown in figure 1.

2. A maze consisting only of perpendicular walls and free space. This controlled environment allows more specific testing as shown in figure 2.

3. An empty environment where targets can be randomly positioned so as to statistically test the algorithms.

Figure 1 : Real world environment

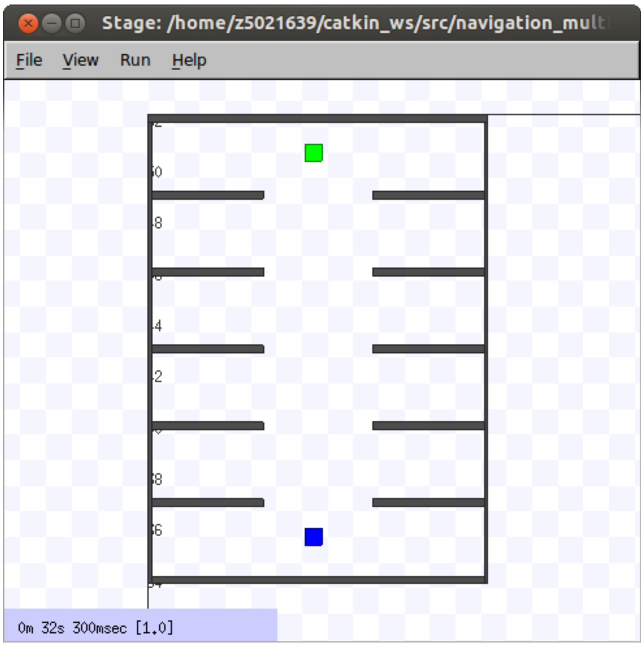


Figure 2 : Maze environment

Results - Floorplan and Maze

In the floorplan environment with 8 targets as labelled in figure 3 the results were as follows,

```
    Parallel Single Item Auction
Cost for blue  is 9.82165 to visit targets 1 4
Cost for green is 25.5934 to visit targets 3 5
Cost for red   is 74.4715 to visit targets 0 2 6 7

   Sequential Single Item Auction
Cost for blue  is 14.5664 to visit targets 4 1
Cost for green is 23.5637 to visit targets 5 3
Cost for red   is 39.8004 to visit targets 2 6 0 7
```

Figure 3 : Floorplan targets

The SSIA results in a time cost of only 53% of the PSIA as a result of much better ordering of the targets. For example, PSIA red agent visits its 4 targets in numerical order resulting in covering additional distance to return to target 2, whilst SSIA visits in the more efficient order.

Taking a more detailed approach and analysing the bids we consider the maze environment with targets allocated as in figure 4 and record results as follows,

```
*** RUNNING PARALLEL SINGLE ITEM AUCTION ***
  Robot 0 bids 22.6688 for target 0
  Robot 1 bids 10.5798 for target 0
    Winner is 1 with 10.5798 for target 0
  Robot 0 bids 27.0000 for target 1
  Robot 1 bids 9.97549 for target 1
    Winner is 1 with 9.97549 for target 1
  Robot 0 bids 3.69231 for target 2
  Robot 1 bids 27.6726 for target 2
    Winner is 0 with 3.69231 for target 2
  Robot 0 bids 18.0000 for target 3
  Robot 1 bids 13.1675 for target 3
    Winner is 1 with 13.1675 for target 3

  Cost for robot 0 is 3.69231 to visit targets 2
  Cost for robot 1 is 32.651 to visit targets 0 1 3
  Maximum cost is 32.651
```

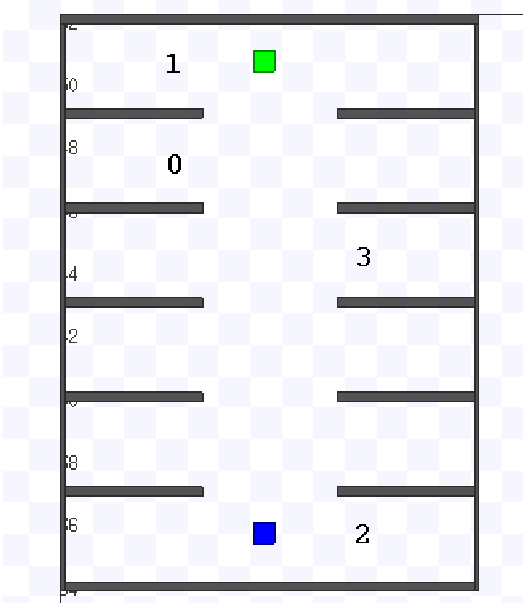


Figure 4 : Maze targets (robot 0 = blue, robot 1 = green)

We can clearly see the weakness of the PSIA approach in that blue robot 0 is only allocated the nearby target 2, leaving green robot 1 to visit the remaining 3 targets.

Switching to SSIA we see a more equitable division of the work,

```
*** RUNNING SEQUENTIAL SINGLE ITEM AUCTION ***
  Robot 0 bids 22.6688 for target 0 at pos 0
  Robot 0 bids 27.0000 for target 1 at pos 0
  Robot 0 bids 3.69231 for target 2 at pos 0
  Robot 0 bids 18.0000 for target 3 at pos 0
  Robot 1 bids 10.5798 for target 0 at pos 0
  Robot 1 bids 9.97549 for target 1 at pos 0
  Robot 1 bids 27.6726 for target 2 at pos 0
  Robot 1 bids 13.1675 for target 3 at pos 0
    Winner is 0 with 3.69231 for target 2 at pos 0
  Robot 0 bids 46.3768 for target 0 at pos 0
  Robot 0 bids 30.2005 for target 0 at pos 1
  Robot 0 bids 54.9471 for target 1 at pos 0
  Robot 0 bids 34.6393 for target 1 at pos 1
  Robot 0 bids 36.8175 for target 3 at pos 0
  Robot 0 bids 23.3051 for target 3 at pos 1
  Robot 1 bids 10.5798 for target 0 at pos 0
  Robot 1 bids 9.97549 for target 1 at pos 0
  Robot 1 bids 13.1675 for target 3 at pos 0
    Winner is 1 with 9.97549 for target 1 at pos 0
  Robot 0 bids 46.3768 for target 0 at pos 0
  Robot 0 bids 30.2005 for target 0 at pos 1
  Robot 0 bids 36.8175 for target 3 at pos 0
  Robot 0 bids 23.3051 for target 3 at pos 1
  Robot 1 bids 18.1830 for target 0 at pos 0
  Robot 1 bids 17.5786 for target 0 at pos 1
  Robot 1 bids 30.0045 for target 3 at pos 0
  Robot 1 bids 24.4435 for target 3 at pos 1
    Winner is 1 with 17.5786 for target 0 at pos 1
  Robot 0 bids 36.8175 for target 3 at pos 0
  Robot 0 bids 23.3051 for target 3 at pos 1
  Robot 1 bids 37.6077 for target 3 at pos 0
  Robot 1 bids 39.8085 for target 3 at pos 1
  Robot 1 bids 27.9066 for target 3 at pos 2
    Winner is 0 with 23.3051 for target 3 at pos 1

  Cost for robot 0 is 23.3051 to visit targets 2 3
  Cost for robot 1 is 17.5786 to visit targets 1 0
  Maximum cost is 23.3051
```

Target 3 is now allocated to blue robot 0 even though it is closer to green robot 1, leading to a time cost that is 73% of the PSIA result. Note however that PSIA requires 8 bids whereas SSIA takes 30 because it considers each remaining target at each round of the auction and at each position in the list of targets already allocated to that robot. In this case we also get the same result if we only consider adding new targets at the end of the list of already allocated targets (which takes 20 bids) but we see later that is not always the case.

We now return to the issue of the accuracy of the bids compared to actual time to move between the allocated targets. Figure 5 below shows this relationship for a variety of paths. Note that actual time is always more than 100% of bid cost since the robot can only ever move slower than full speed. For shorter paths there is a greater discrepancy because the fraction of the journey time spent accelerating and decelerating is greater. As path length increases the inpact of the acceleration reduces. It would be possible to scale the bids by a best fit function derived from this data in order to arrive at a more accurate heuristic bid.

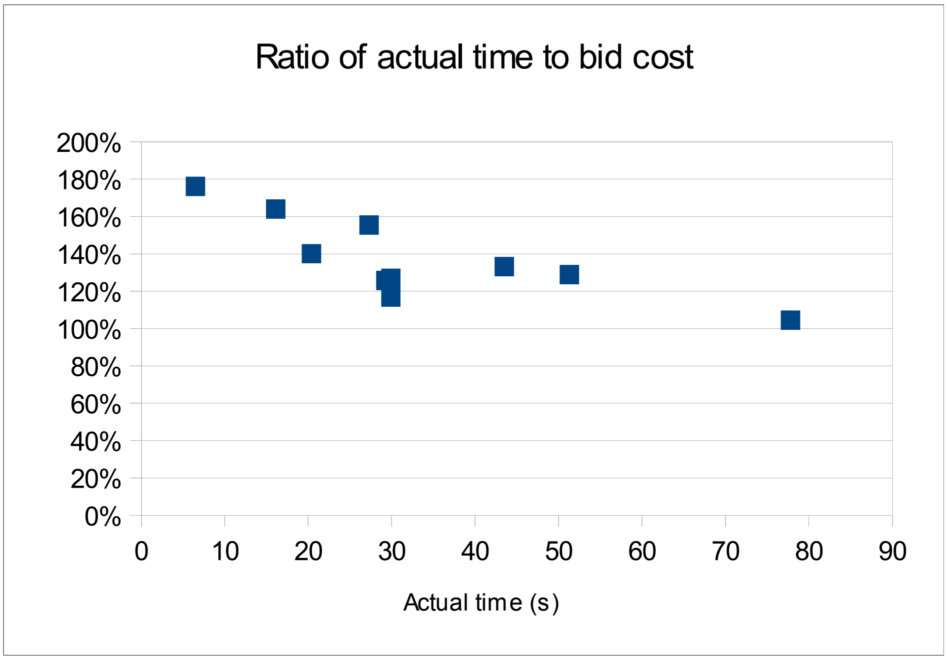


Figure 5 : Actual travel time and bid time

Results - Random Placements

Finally for the results we consider the empty environment and quantify the greater efficiency of SSIA compared to PSIA. The following chart show the resulting auction times for 4 different scenarios of increasing numbers of robots and targets. The horizontal black lines are the average times for PSIA and SSIA across 10 runs with random placements. The vertical coloured bars represent the standard deviations.

For the simpler scenario of 2 robots visiting 5 targets SSIA produced an average time of 64% of PSIA. With the added complexity of 5 robots visiting 20 targets SSIA was better able to create synergies, finding paths that took 39% of PSIA on average. In even larger scenarios with 10 robots visiting 50 targets SSIA also took 39% of PSIA. Note also that the range of PSIA times was very significant for large scenarios where the positioning and ordering of the tasks may fortuitously result in an efficient or very inefficient path. By comparison SSIA consistently produces solutions around 100 seconds with 15% standard deviation.

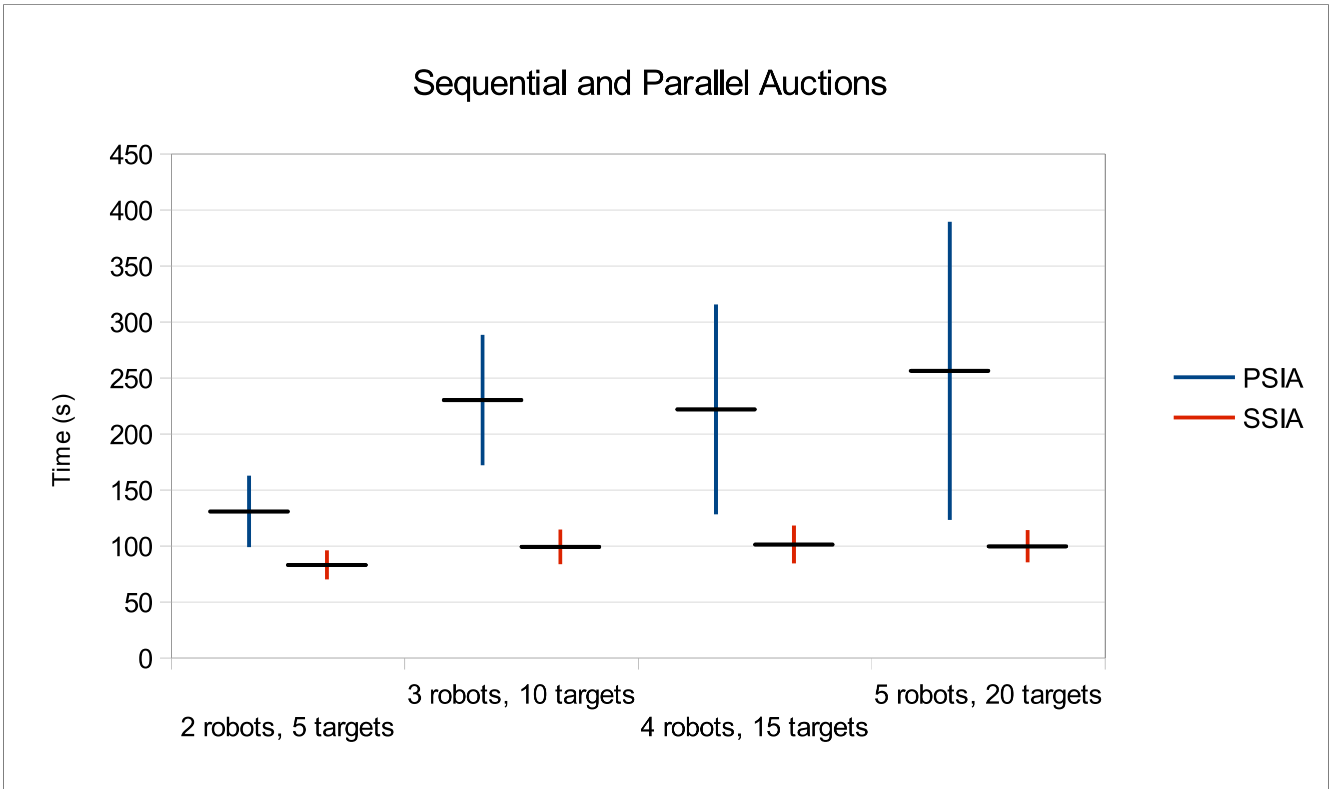


Figure 6 : SSIA vs PSIA for random placements

As mentioned in the maze environment resullts we can consider a simplification to SSIA whereby new targets that are being bid upon are only added to the end of existing agent target allocation paths instead of at any point on the path. It will usually be the case that new targets are best added to the end of the path because they are generally further away than targets that have already been won, hence making such an assumption should not impact optimality significantly but at the same time reduce the auction run time. Figure 7 shows a comparison of the full SSIA auctions and the simplified version for the same scenarios as figure 6.

The simplifed solution is only 0.3% worse than the full solution for the 2 robot cases and 2.2% worse for 5 robots, making it an attractive proposition. Note that the full solution is not even guaranteed to be at least as good as the simple solution because there are situations where a slightly more efficient ordering of the full SSIA results in a robot location that is then further away from the next target, ultimately resulting in a total time increase that more than offsets the original efficiency gain.

Figure 7 : SSIA full vs simple for random placements

Conclusions and Further Work

We have demonstrated the ability of sequential single item auctions to produce more optimal task allocations than parallel auctions in a variety of environments. The advantage of SSIA increases as the number of agents and tasks increases, resulting in total time costs of less than 40% of PSIA for large cases. SSIA also has an advantage in that the standard deviation of results are signicantly lower than PSIA.

We have also shown that simplifying SSIA to bid on new targets that only extend the path of existing targets (instead of being inserted at any point) reduces run time at only a small cost to optimality.

As an extension, the bidding calculation could be enhanced to account for linear and angular acceleration. Instead of more complex calculations it is suggested that a simple heuristic scaling based upon the bid path length would produce an improvement for minimal calculational cost.

References

[1] The Contract Net Protocol, Smith, IEEE Transactions on Computers, 1980.

[2] The Power of Sequential Single-Item Auctions for Agent Coordination, Koenig et al, AIII, 2006.

[3] Sequential Single-Cluster Auctions for Multi-Robot Task Allocation, Heap, UNSW PhD thesis, 2013.

[4] Multi-Robot Systems, Yehoshua, http://u.cs.biu.ac.il/~yehoshr1/89-689/

[5] Towards an Efficient Optimal Trajectory Planner for Multiple Mobile Robots, Thomas et al, IROS, 2003.