# Multi Agent Systems

# Paper Implementation: “Auction-Based Multi-Robot Routing”

Vedant Bhatia 2016113
Naman Kedia 2016058

We have implemented all three routing algorithms as presented in the paper, i.e.:

a. **MiniSum**: Minimize the total cost incurred by the robots
b. **MiniMax**: Minimize the cost of the most expensive traversal by robots
c. **MiniAve**: Minimize the average cost over all targets

**Running the code**:

*python main.py [OPTION]*

*Options:*

1. *Optimization objective: [VAL]*
   a. *Values = ‘ave’,’max’,’sum’*
2. *Scenario: [VAL]*
   a. *Values: 1,2,3,4 (as per scenarios detailed below)*
3. *Verbose: --verbose (print each step)*

***Example usage:***

*python main.py ‘sum’ 1 --verbose*
*python main.py ‘max’  3*

***Note:***
***For each given scenario, the number of robots can be changed in the function definition.***
***For scenario 2, beta and epsilon can be changed in the function definition.***

The paper presents 4 example scenarios detailing how their algorithms performs in each scenario. We have successfully replicated these, detailed below:

**Scenario 1:**

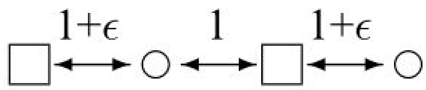
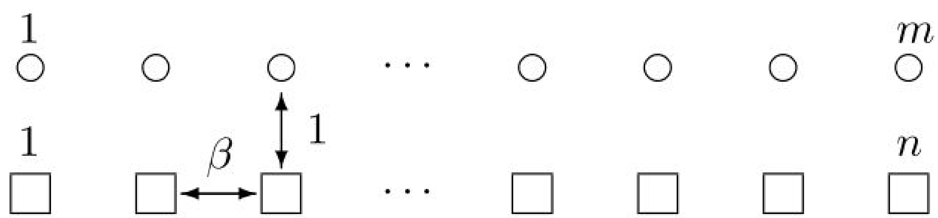



Fig. 1. A simple instance with 2 robots (squares) and two targets (circles).

**Result**: With MiniSum objective, algorithm allocates both targets to the robot on the right.

**Scenario 2:**




e

Fig. 2. Parallel lines construction: $n$ robots and $m = n$ targets spaced evenly (distance of $\beta$) on two parallel lines (one for robots, one for targets). The distance between the two lines is 1, except for the left-most robot which is a little closer to its corresponding target.

**Result**:

a. With MiniSum objective and ***beta = 1 - epsilon,*** all targets are allocated to the left most robot and a path that runs through all targets from left to right.
b. With MiniMax objective and ***beta = epsilon,*** one target is allocated to each robot

**Scenario 3:**

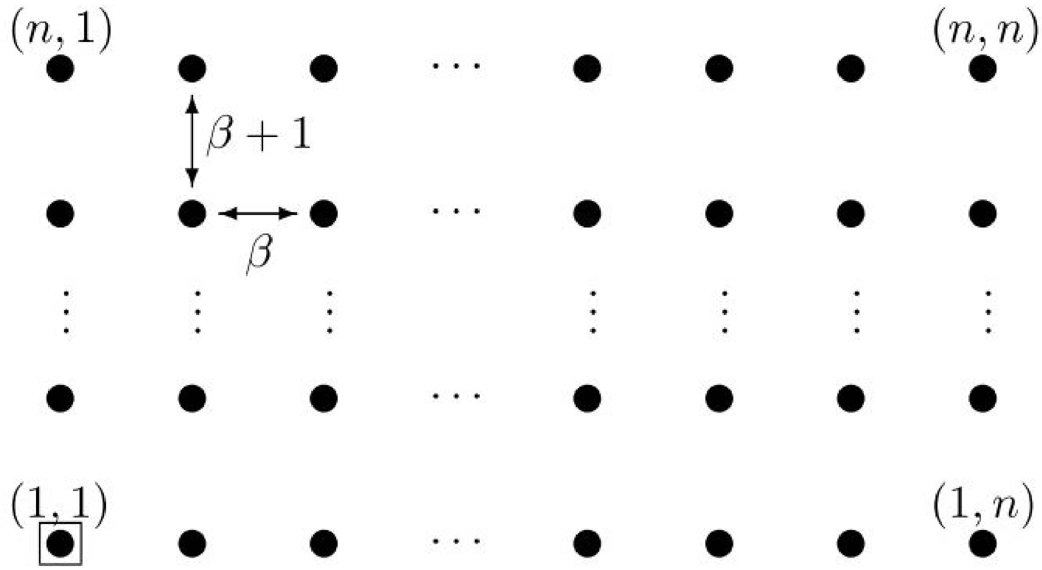



Fig. 3. Grid construction: $n$ robots at $(1, 1)$ and $m = n^3$ targets on a $(n \times n)$ rectangular grid; each of the $n^2$ gridpoints is a cluster of $n$ targets. The intra-row distance is $\beta$, whereas the intra-column distance is $\beta + 1$.

**Result**: With MiniMax or MiniAve objective, each robot is allocated one target in each cluster. Path for each robot is from left to right and right to left and bottom to top.

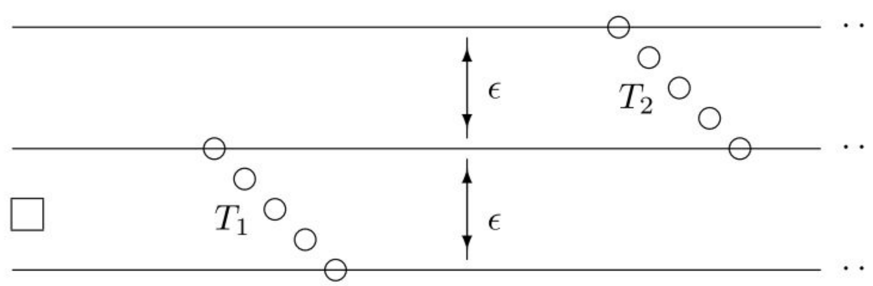



Fig. 4. Cluster arrangement in grid construction: example with 2 clusters of 5 targets each and 5 robots clustered on the left. Targets in cluster $T_1$ are arranged evenly on a line of slope -45° within a small cost from each other. Each of the 5 robots visits one target in $T_1$, since the straight line is the shortest path. Targets in $T_2$ have the same arrangement, but are shifted up by $\epsilon$ to ensure that the cost between corresponding targets in $T_1$ and $T_2$ is less than any other inter-cluster cost. The robot that visits the first target in $T_1$ will also visit the first target in $T_2$, and so on. This pattern repeats along the horizontal axis, but it can also be used for clusters arranged vertically.

**Scenario 4:**

Result: With MiniAve, the described allocation in the figure takes place.