OMCoRP: An Online Mechanism for Competitive Robot Prioritization

Sankar N. Das[1], Swaprava Nath[2], and Indranil Saha[2]

[1]Accenture Labs, India, [2]IIT Kanpur India

sankar.c64@gmail.com, swaprava@cse.iitk.ac.in, isaha@cse.iitk.ac.in

Collision Avoidance

- A central problem in various autonomous movement environments – e.g., **warehouse**, **autonomous vehicles**
- Our focus: **dynamic arrival** and **independent control** of the robots
- Difficulty: robots are **competitive** rather than **cooperative**
- **Design goals**:
  - Prioritize the more 'valuable' robots – without any prior knowledge of the robots that are independently owned/controlled
  - Scalable to 1000s of robots
  - Collision-free and Deadlock-free
  - Handle dynamic arrival of robots

Our Model and Approach

- Model is similar to the auction-like collision avoidance setting
- But analysis is a full-blown mechanism design approach with **competitive** robots
- The movement is given by a graph with edges being full-duplex paths
- Edges are incident on nodes which are roundabouts
- Robot i:
  - current location $\ell_i(t)$
  - intended next location $n_i(t)$
  - *value* for moving to that location $v_i(t)$, zero otherwise

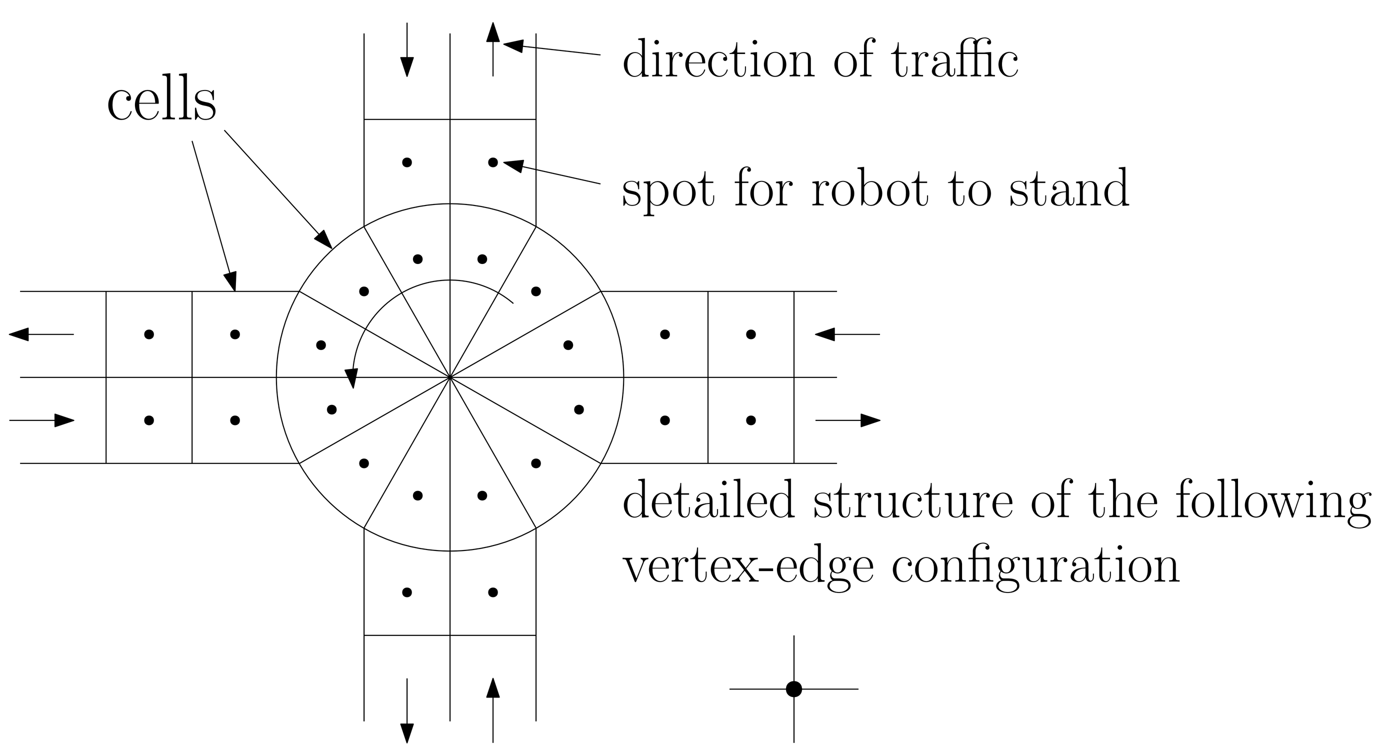


Illustration of a vertex with four incident full- duplex edges

State-of-the-art

1. Offline multi-robot planning algorithms
   - generate the collision-free paths for all the robots statically
   - *computation time* for generating path plans prohibitively large
   - cannot deal with the *dynamic arrival* of new robots
2. Robots independently generate their trajectories offline
   - trajectories are not *collision-free*
   - collision situations are resolved online in a decentralized manner through information exchange
   - **cooperative** robots
3. Robots interact in an auction-like setting and bid for their makespan or demand for resource
   - only *bid* part of the auction is used
   - assumed to bid *truthfully*

Competitive Robots

- The robots have different private valuation for movement
- Robots compete with each other to get a prioritized treatment
- The valuation indicates priority of a robot
  - Premium: high value, carries priority shipping items
  - Regular: moderate value, regular items
  - Economy: low priority, non-priority items

Algorithm: OMCoRP run by the LIM at $I_k(V)$ at time t

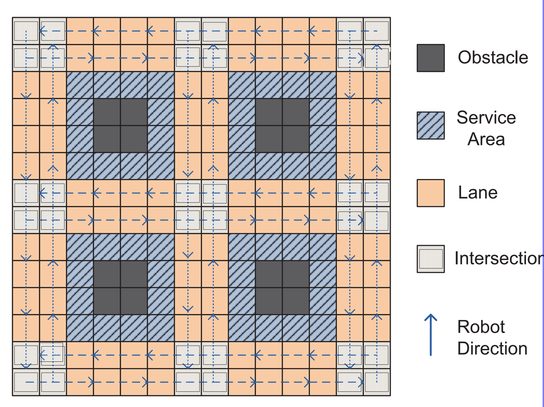
1: **Input:** For every robot $i \in N_k(t)$, the current and future cells $\ell_i(t), n_i(t)$, and reported value $\hat{v}_i(t)$

2: **Output:** Decisions for every robot $i \in N_k(t)$ to STOP/GO/FOLLOW

3: Compute $a_k^*(t+1)$ (eqn:efficient) and suggest the robots $i \in N_k(t)$ to STOP/GO/FOLLOW according to that recommendation in time $t+1$

4: Charge $p_i(t)$ (eqn:payment) to every $i \in N_k(t)$

5: Collect this money and distribute equally to all $j \in N \setminus N_k(t)$

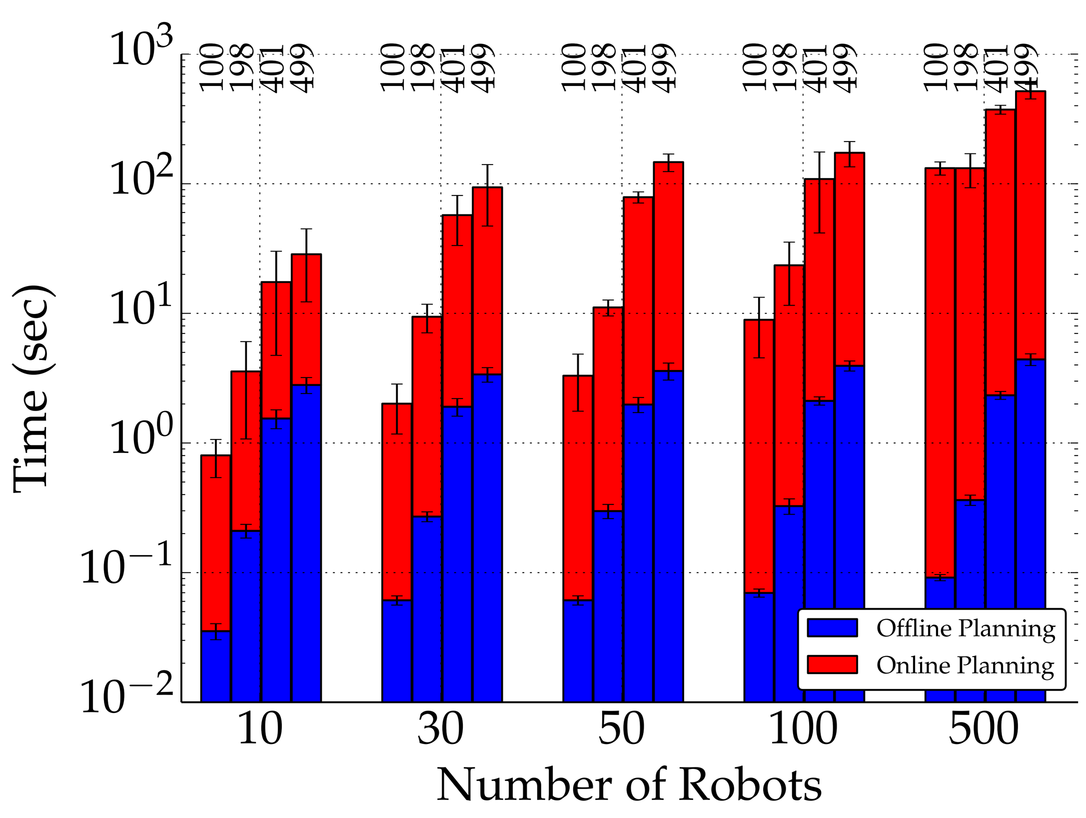


Setup: A 14×14 workspace.

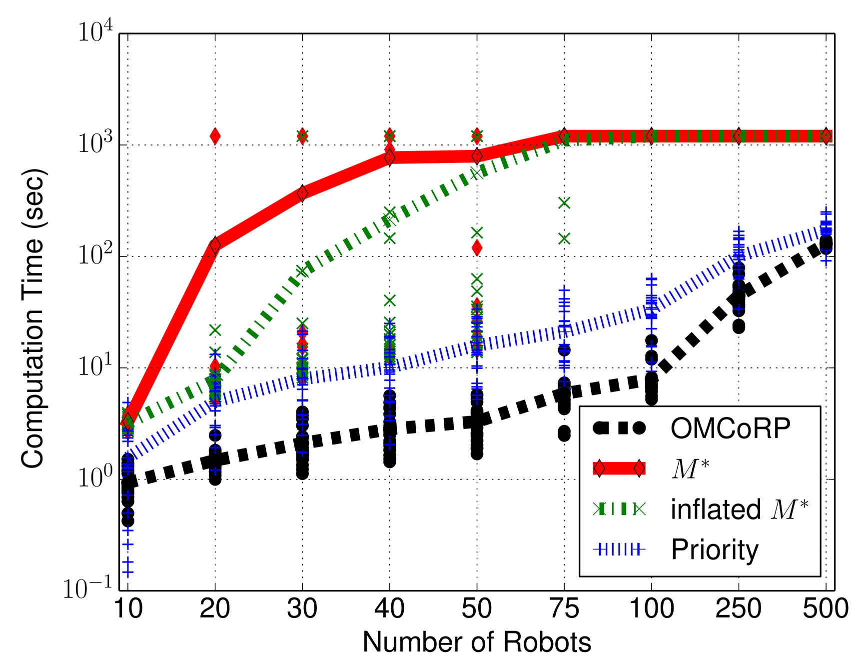
Properties

- Collision-free, decentralized, deadlock-free, and locally efficient
- Dominant strategy truthful: reporting true value is best
- Budget balanced : no net in or outflow of money
- Robust against entry or exit : new robots can join at any time

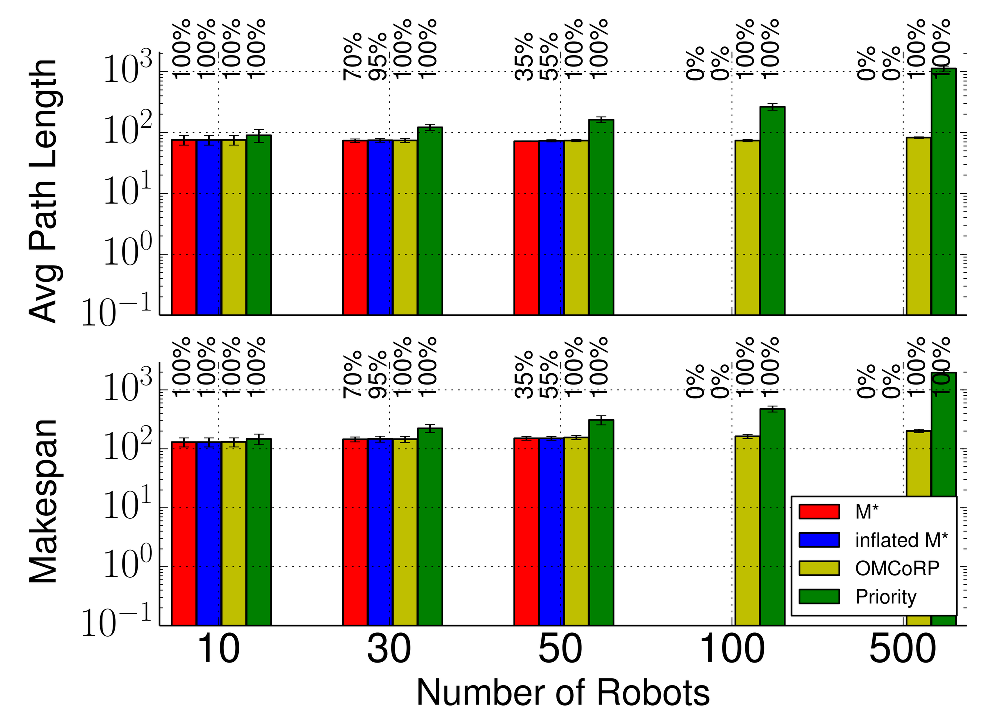
Experiments



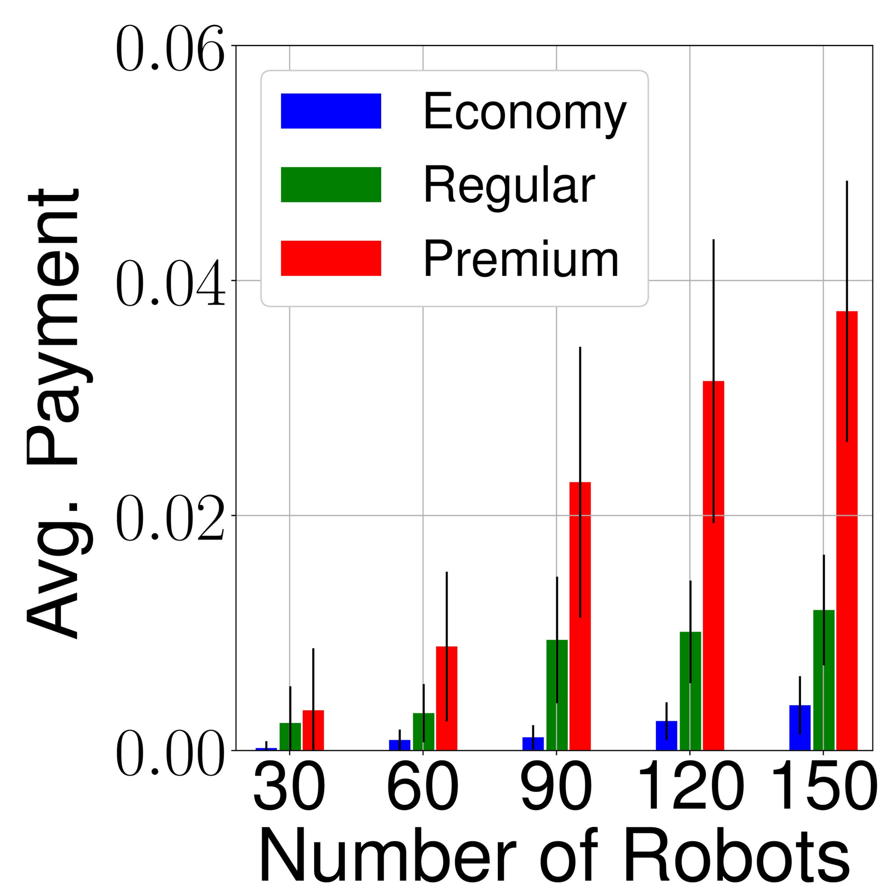
Scalability: Path planning and auction times of OMCoRP. The numbers on the bars show the workspace width for each number of robots.



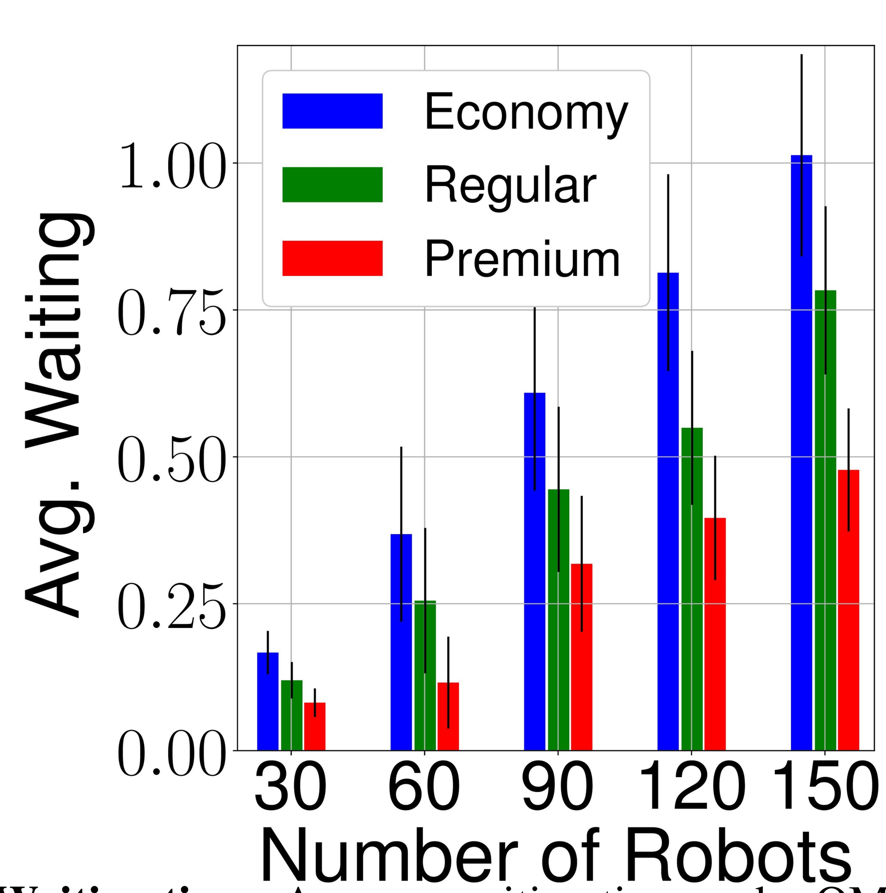
PlanningTime: The y-axis shows the total planning time of the different algorithms and the x-axis shows the number of robots. For OMCoRP, the planning time is the sum of the offline path computation and online spot auction times.



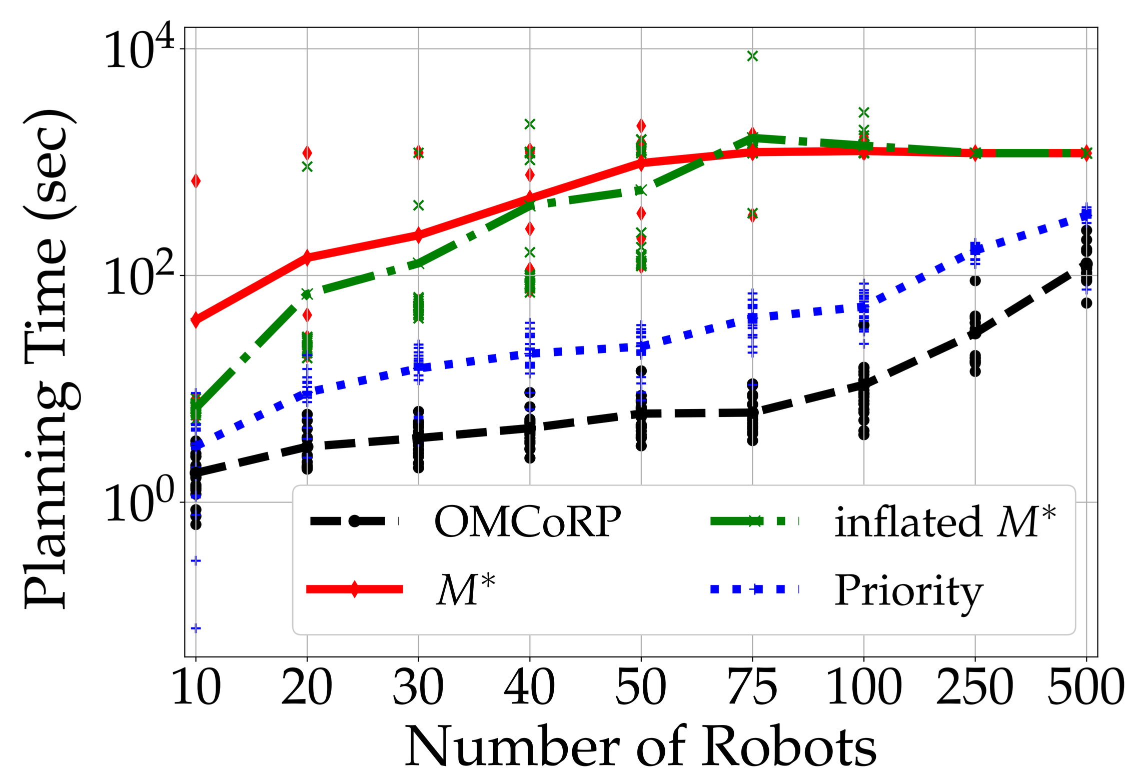
Path Length : Average path length and makespan for different mechanisms. The numbers on the bar denote the percentage of cases where the mechanism could find a path within the timeout period.



Payment: Average payments under OMCoRP for different classes of robots.



Waiting time: Average waiting time under OMCoRP for different classes of robots.



Dynamic Arrival: Comparison of Total Planning Time