SAGL: A NEW HEURISTIC FOR MULTI-ROBOT ROUTING WITH COMPLEX TASKS

Hong Xu[*], T. K. Satish Kumar[*], Dylan Johnke[†],
Nora Ayanian[*] and Sven Koenig[*]

[*] University of Southern Calfornia, Los Angeles, CA 90089

[†] Cornell University, Ithaca, NY 14853

ICTAI November 7, 2016

AGENDA

- Complex Routing Problem (CRP)
- Our algorithm: SAGL
- Experimental evaluation: SAGL vs others

AGENDA

- Complex Routing Problem (CRP)
- Our algorithm: SAGL
- Experimental evaluation: SAGL vs others

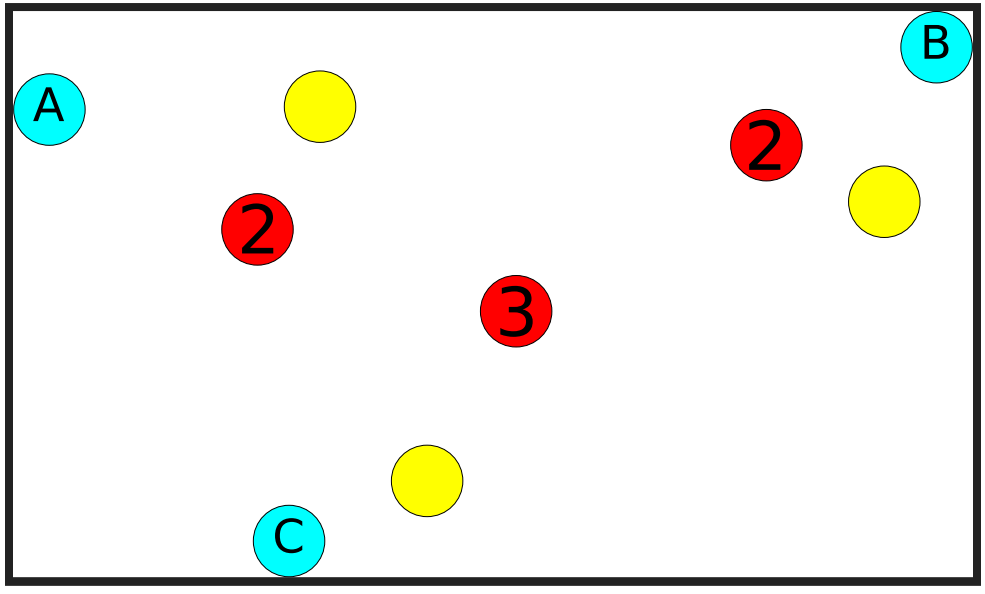
MOTIVATION

Search-and-rescue: lifting heavy debris (source: https://www.fema.gov/media-library/assets/images/100223)

COMPLEX ROUTING PROBLEM (CRP)

- Multiple homogeneous robots
  - same moving speed
  - same ability to accomplish tasks
- Multiple tasks in different locations
- Some tasks require more than one robots to accomplish
- Cooperative settings
- Solution: Task visitation order for each robot
- Solution evaluation: Makespan (total time required to accomplish all tasks)

COMPLEX ROUTING PROBLEM EXAMPLE



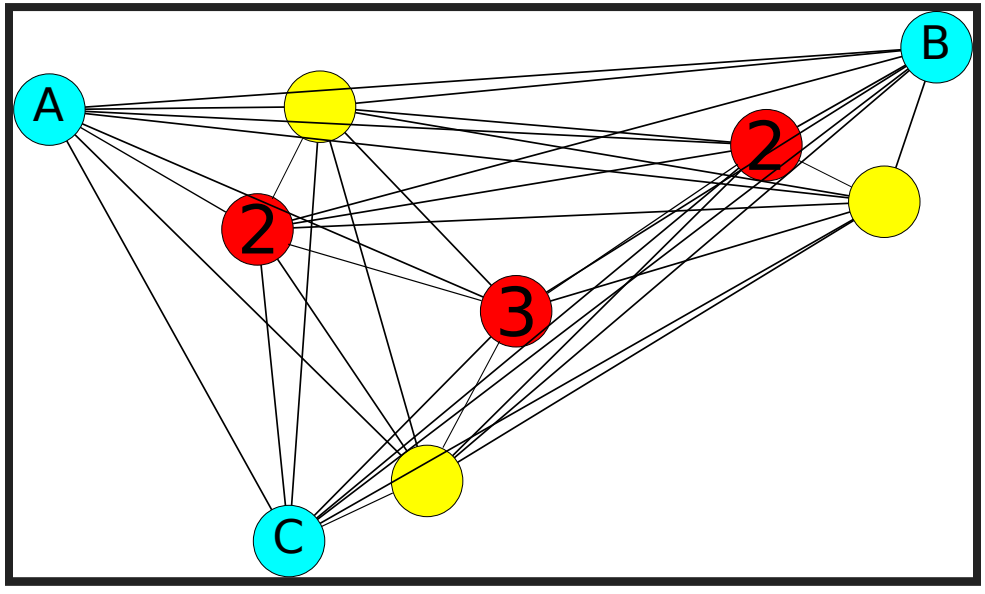
- Cyan: robots
- Yellow: tasks requiring only 1 robot (simple tasks)
- Red: tasks requiring N robots ($N \geq 2$) (complex tasks with a complexity level of N)
- Reduces to TSP-Path if only one robot and no complex tasks.

- State-of-the-art: Approach with Reaction Functions (ARF) [Zheng et al. '08, '11]
  - Based on auction mechanism
  - Produces good solution
  - Does not scale
    - e.g., cannot solve 20 complex tasks with a complexity level of 2 within 1 hour
- SAGL
  - Produces decent solution
  - Polynomial time complexity and scalable
    - Can handle high complexity levels and a large number of complex tasks

AGENDA

- Complex Routing Problem (CRP)
- Our algorithm: SAGL
- Experimental evaluation: SAGL vs others

EMBED COMPLEX ROUTING PROBLEM INTO A GRAPH



We embed a problem instance into
a **complete undirected edge-weighted graph**

- Vertices represent task and robot initial locations.
- Edges represent distances between the locations.

ASSUMPTIONS

- No collisions between robots.
- Distances satisfy the triangle inequality.
- Distances are symmetric.
- All robots move with unit speed.
- All tasks are accomplished immediately once all robots arrive—time span required for accomplishing tasks can be amortized into the incident edges.

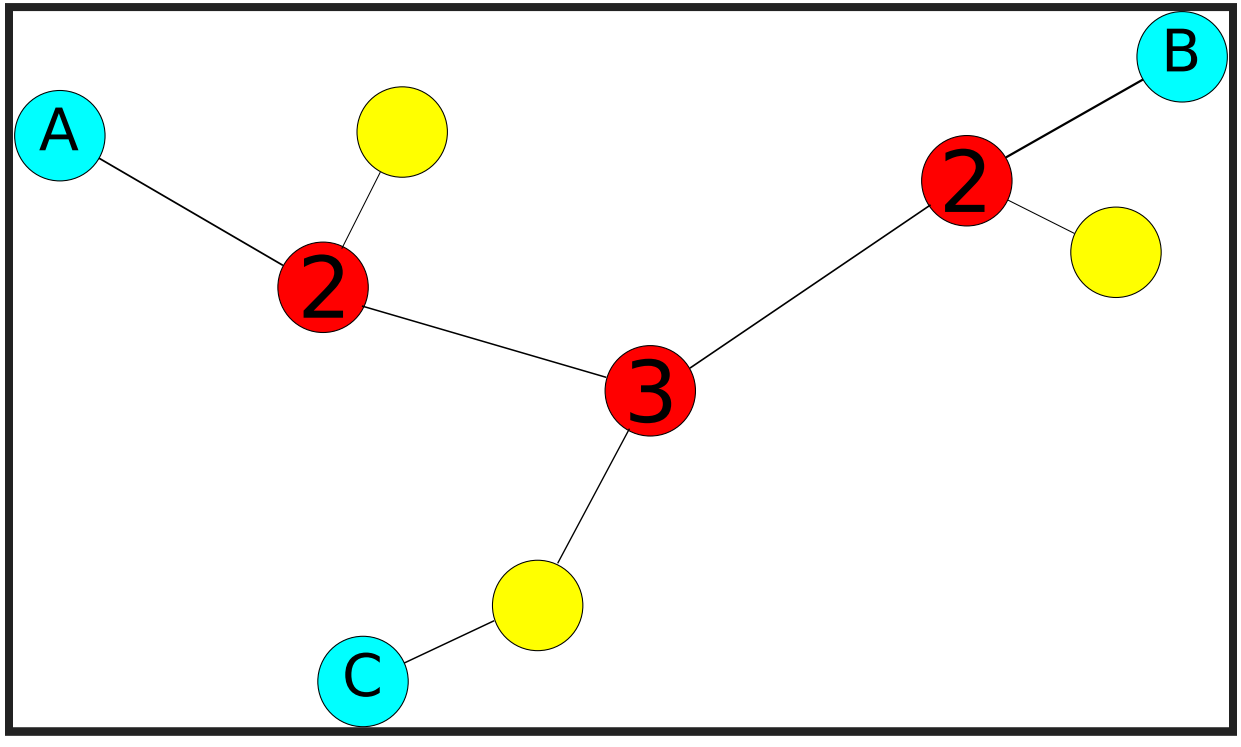
SAGL OVERVIEW

- Which robots should visit which tasks
  1. Spanning tree construction
  2. Task assignment
- What visitation order should the robots use
  3. Global visitation order determination for complex tasks
  4. Local visitation order determination

WHICH ROBOTS SHOULD VISIT WHICH TASKS

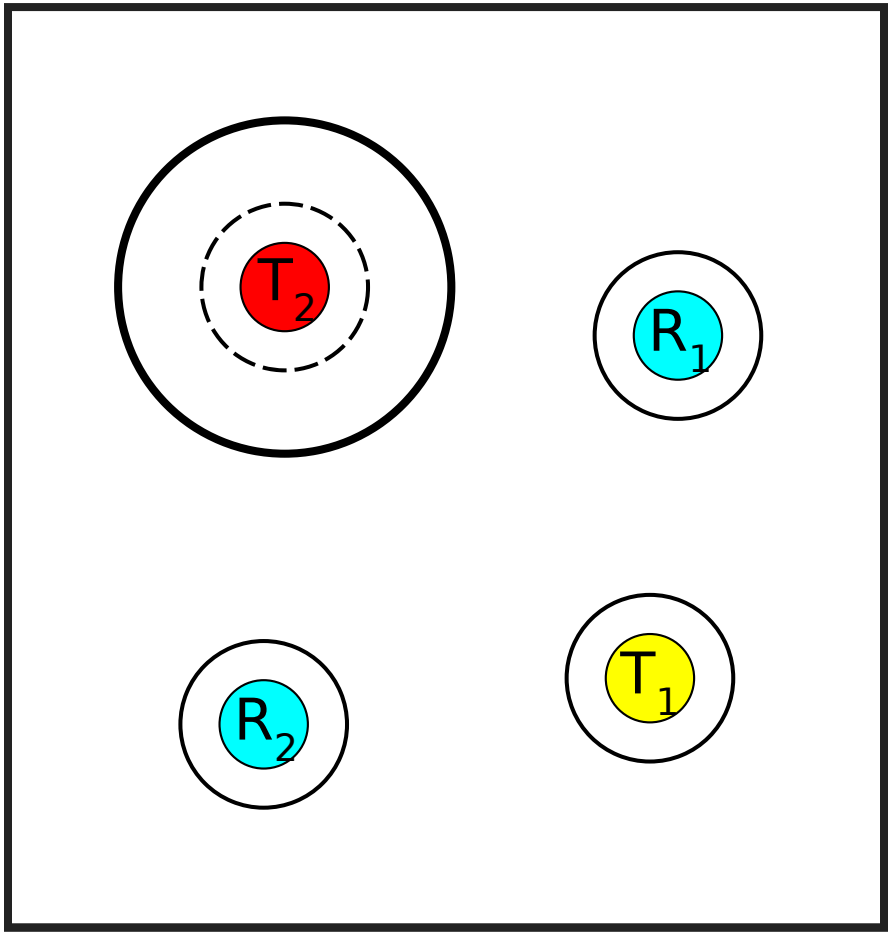
1. Spanning tree construction
    - Provides a base for task assignments
2. Task assignment

CONSTRUCT THE SPANNING TREE

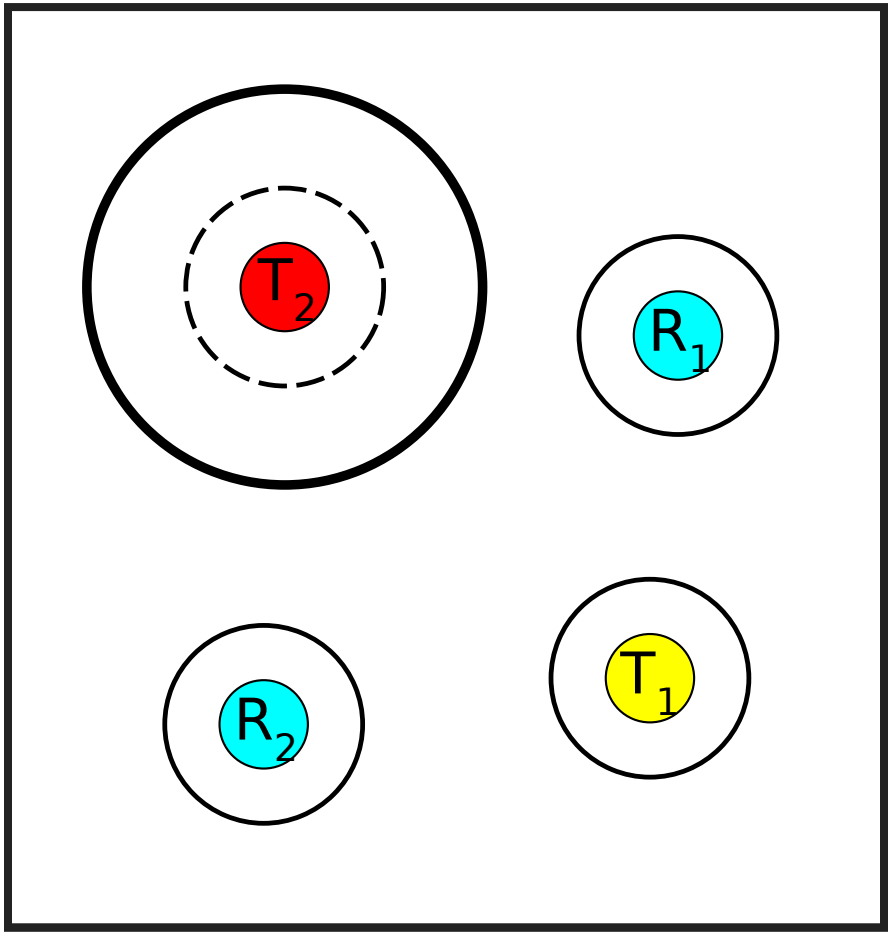


Provides a base for task assignments:
inspired by 2-approximation TSP

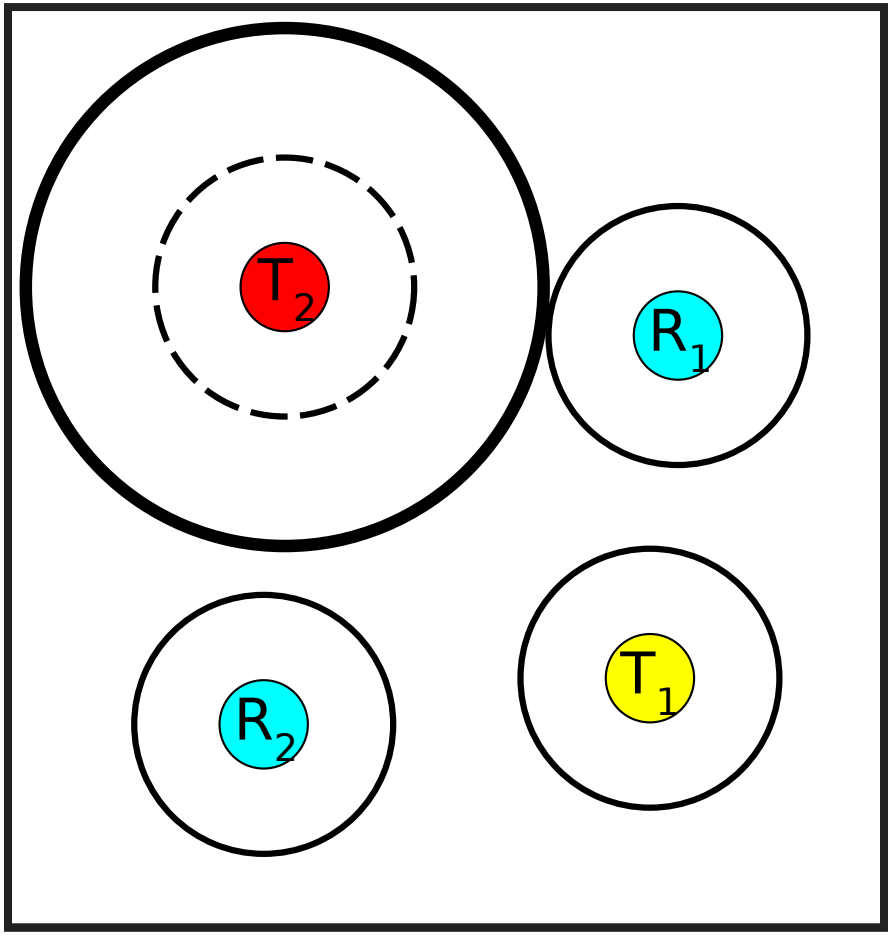
CONSTRUCT THE SPANNING TREE



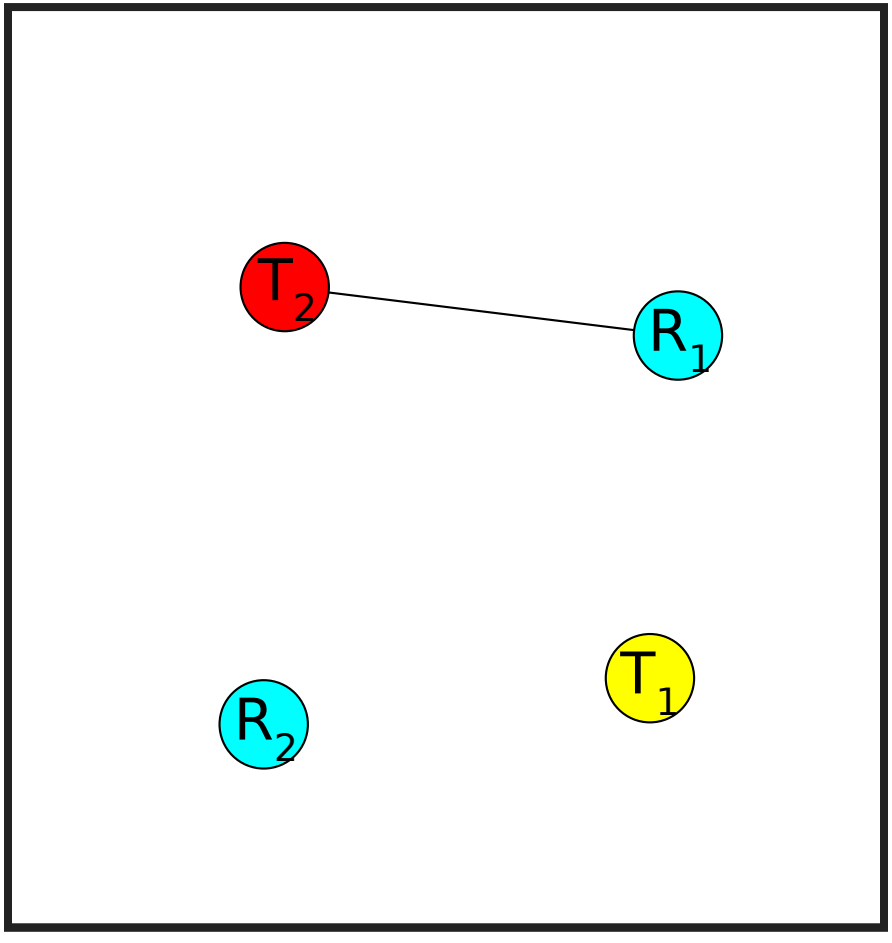
CONSTRUCT THE SPANNING TREE



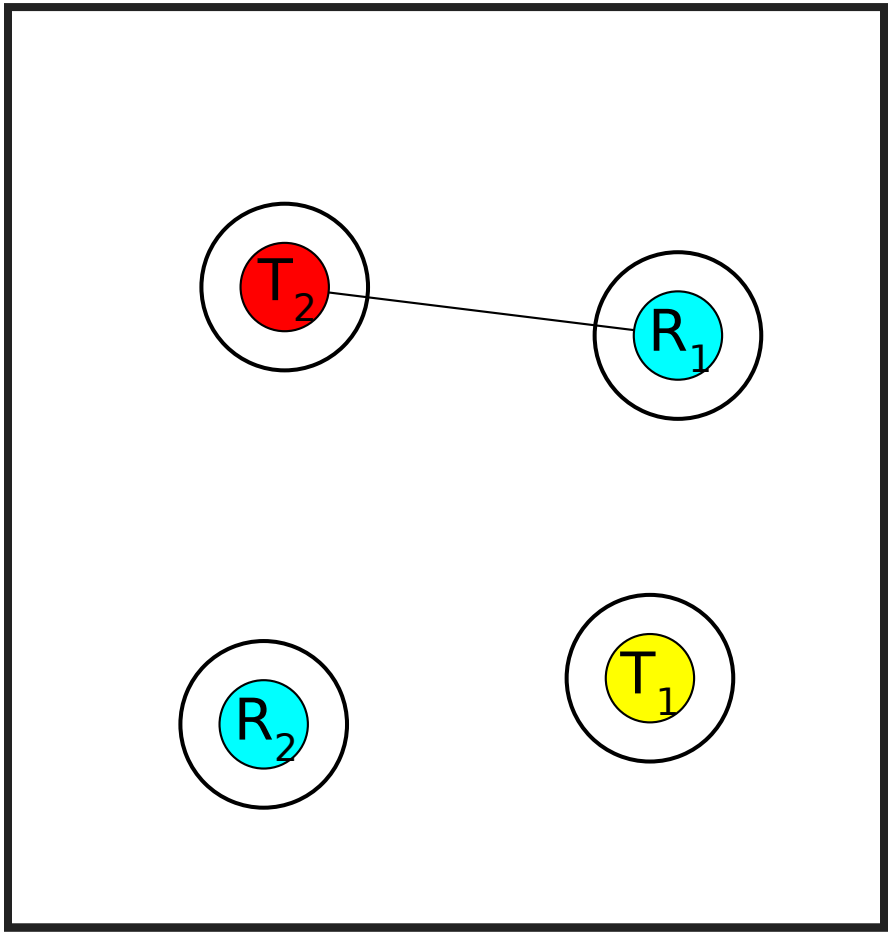
CONSTRUCT THE SPANNING TREE



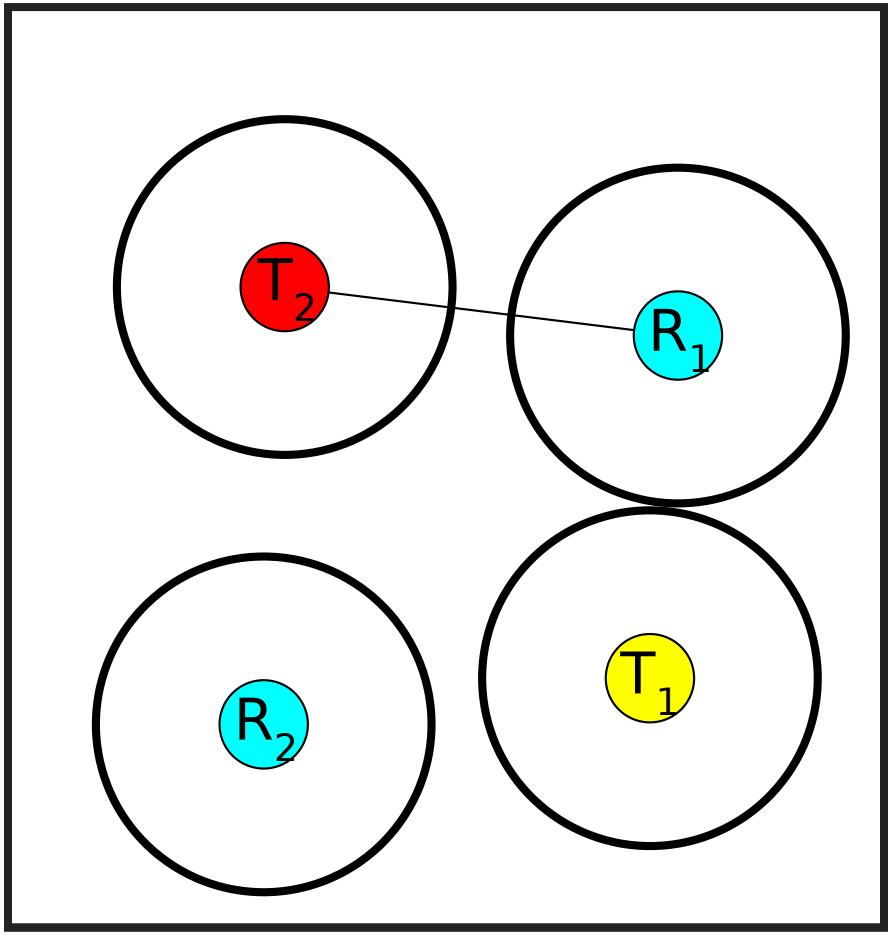
CONSTRUCT THE SPANNING TREE



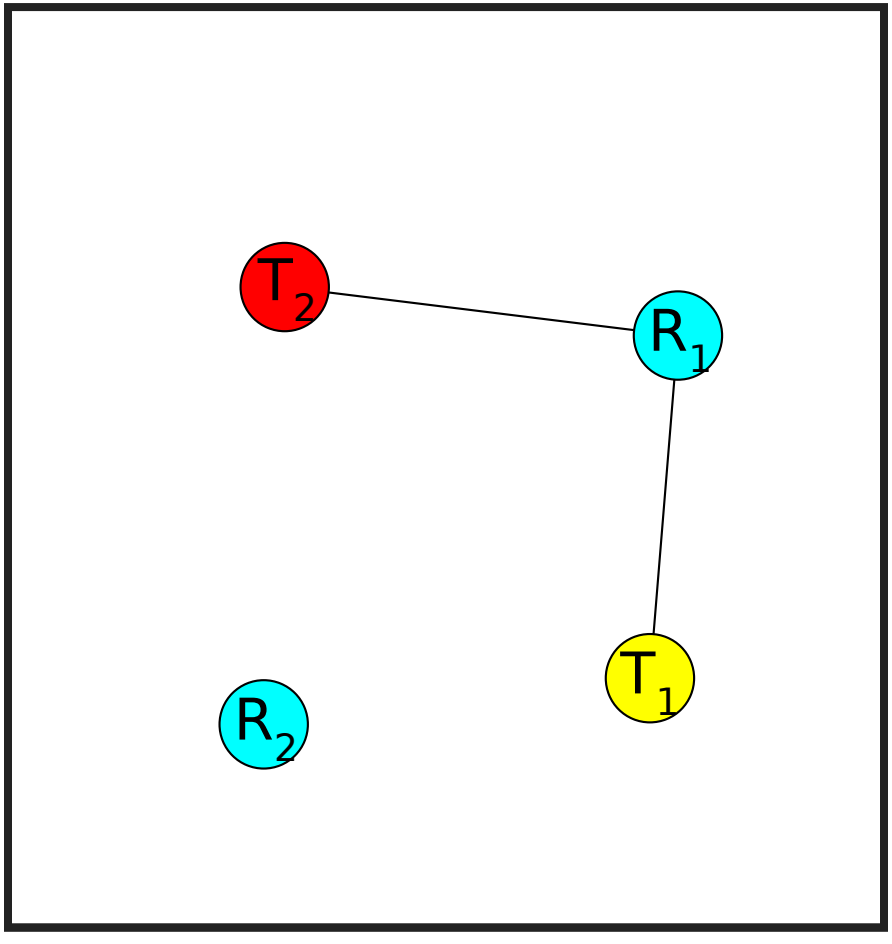
CONSTRUCT THE SPANNING TREE



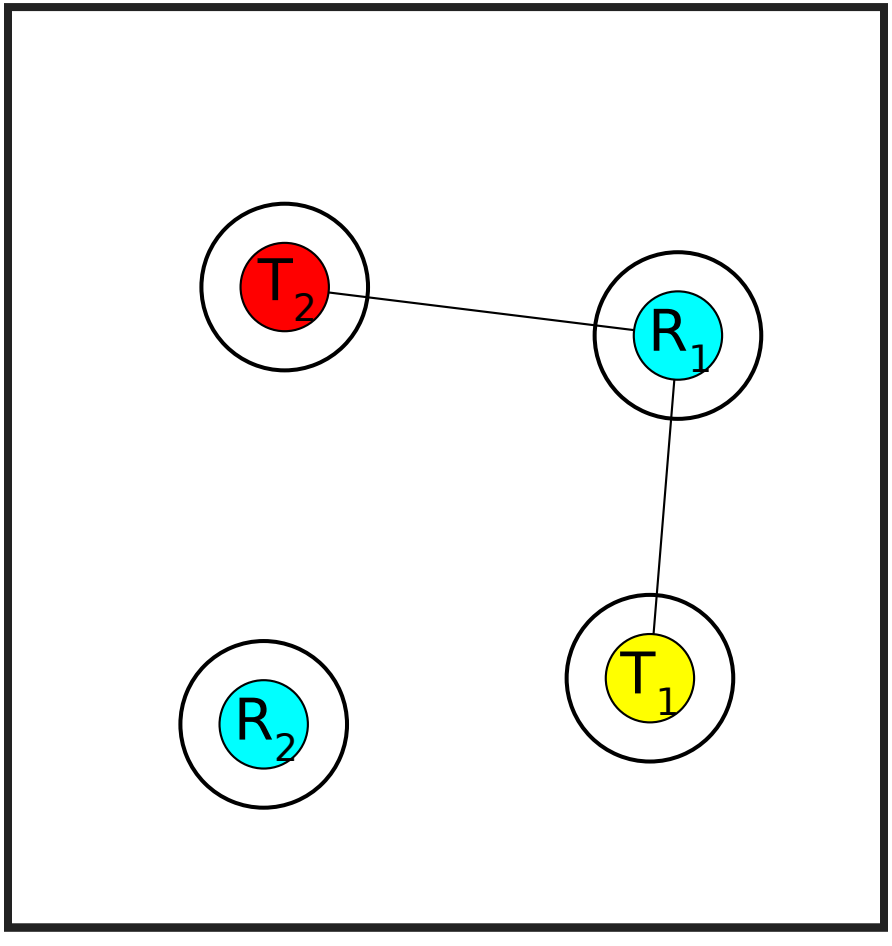
CONSTRUCT THE SPANNING TREE



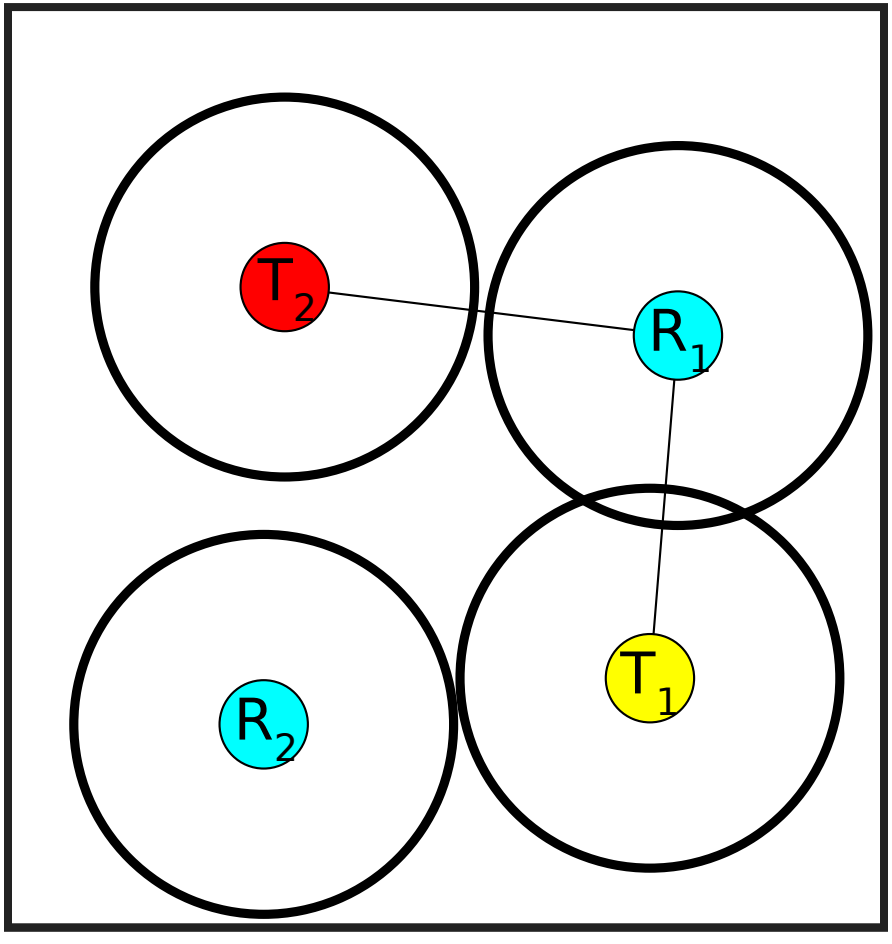
CONSTRUCT THE SPANNING TREE



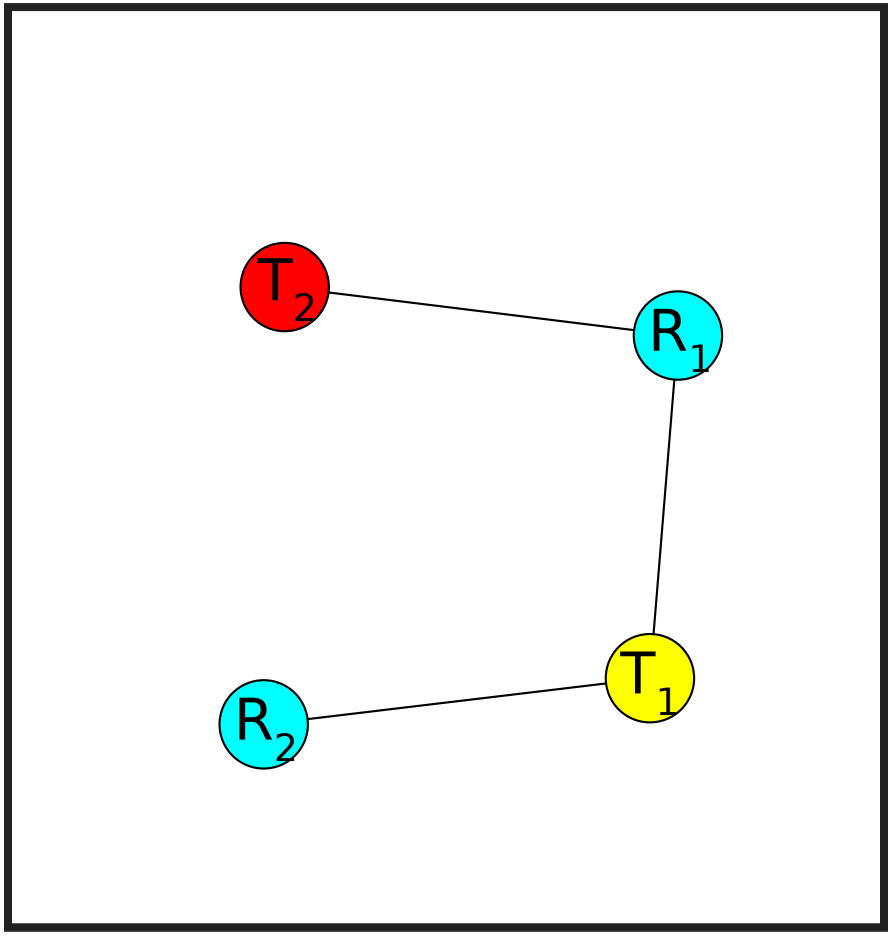
CONSTRUCT THE SPANNING TREE



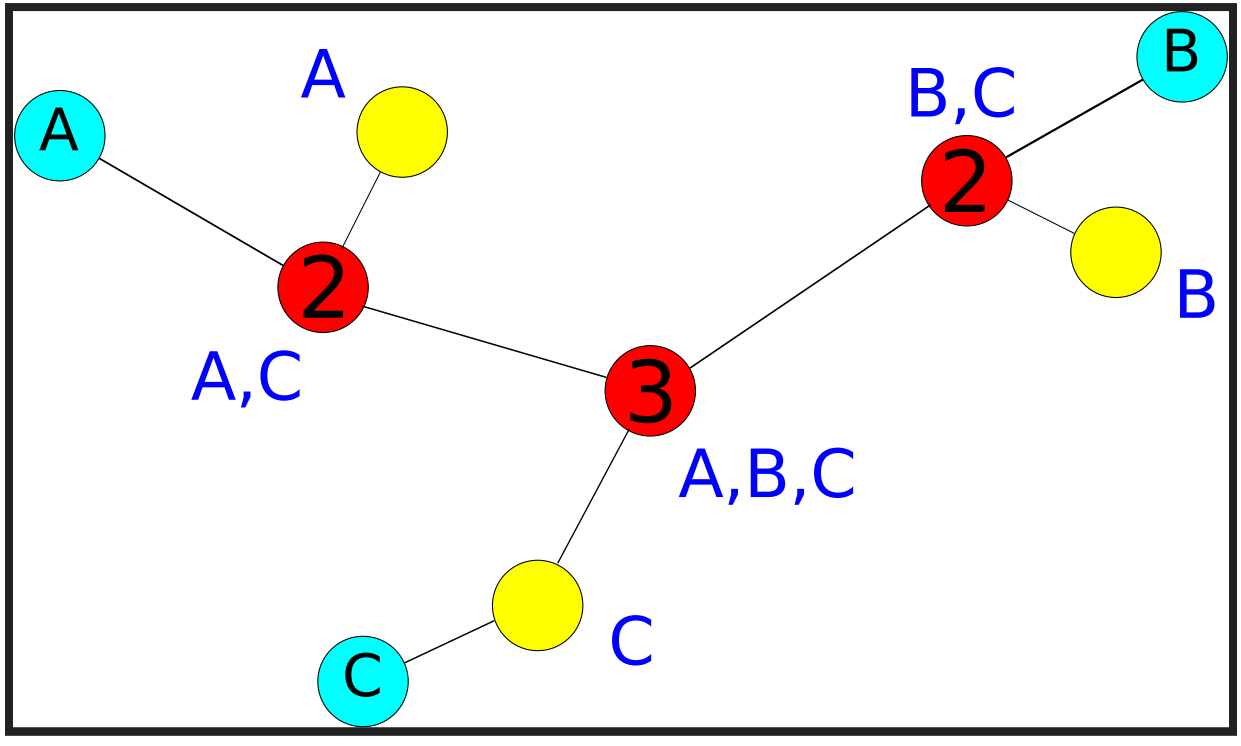
CONSTRUCT THE SPANNING TREE



CONSTRUCT THE SPANNING TREE



ASSIGN TASKS TO ROBOTS

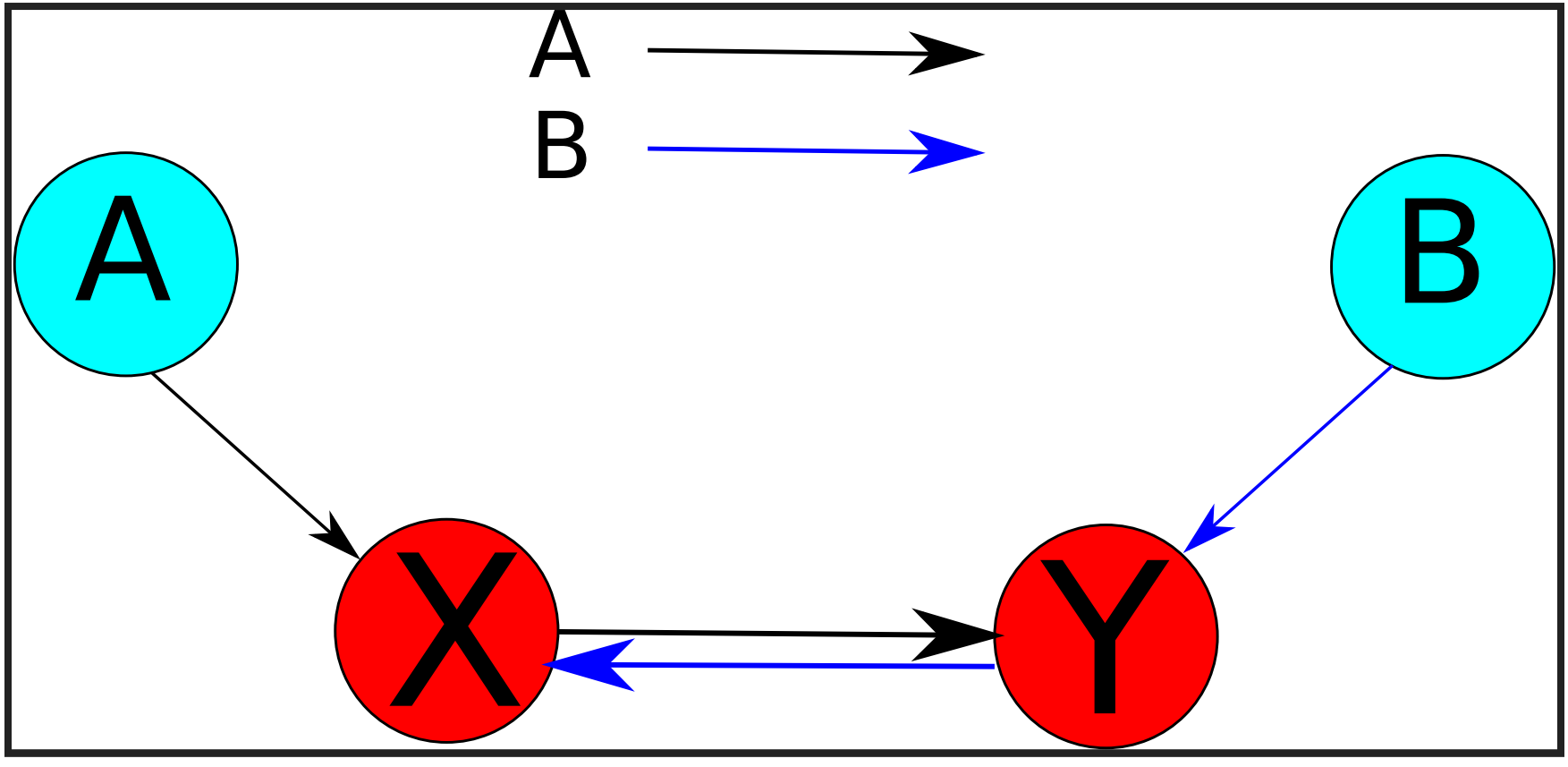


According to the distances on the spanning tree.

WHAT VISITATION ORDER SHOULD THE ROBOTS USE

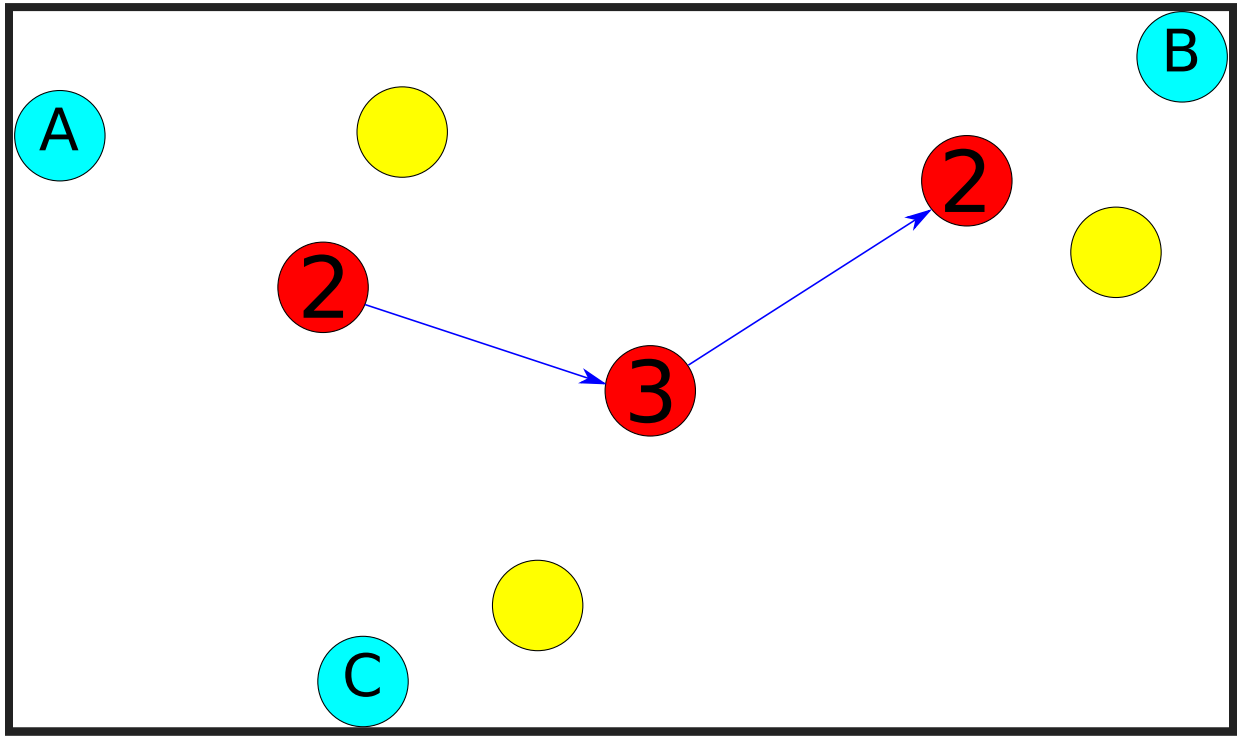
1. Global visitation order determination for complex tasks
    - Prevent deadlocks
2. Local visitation order determination

DEADLOCKS



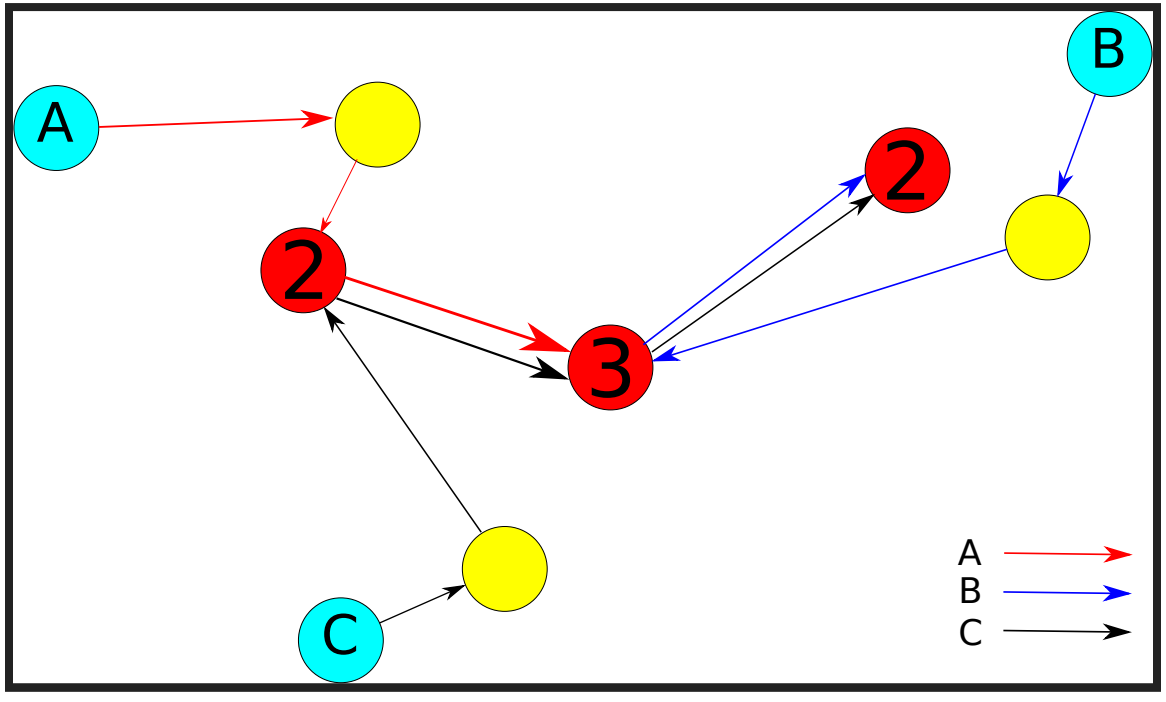
- Robot A waits at X forever.
- Robot B waits at Y forever.

GLOBAL VISITATION ORDER OF COMPLEX TASKS



Prevent deadlocks

LOCAL VISITATION ORDER



Path-constrained TSP [Bachrach et al. '05]:
Consistent with the global visitation order

AGENDA

- Complex Routing Problem (CRP)
- Our algorithm: SAGL
- Experimental evaluation: SAGL vs others

EXPERIMENT SET 1

- Compared with Approach with Reaction Functions (ARF) [Zheng et al. '08, '11]
- Map: 51x51 grid office environment
- 200 CRP instances with random vertices for
  - 10 robots
  - 80 simple tasks
  - various numbers of complex tasks with a complexity level of 2

OFFICE MAP

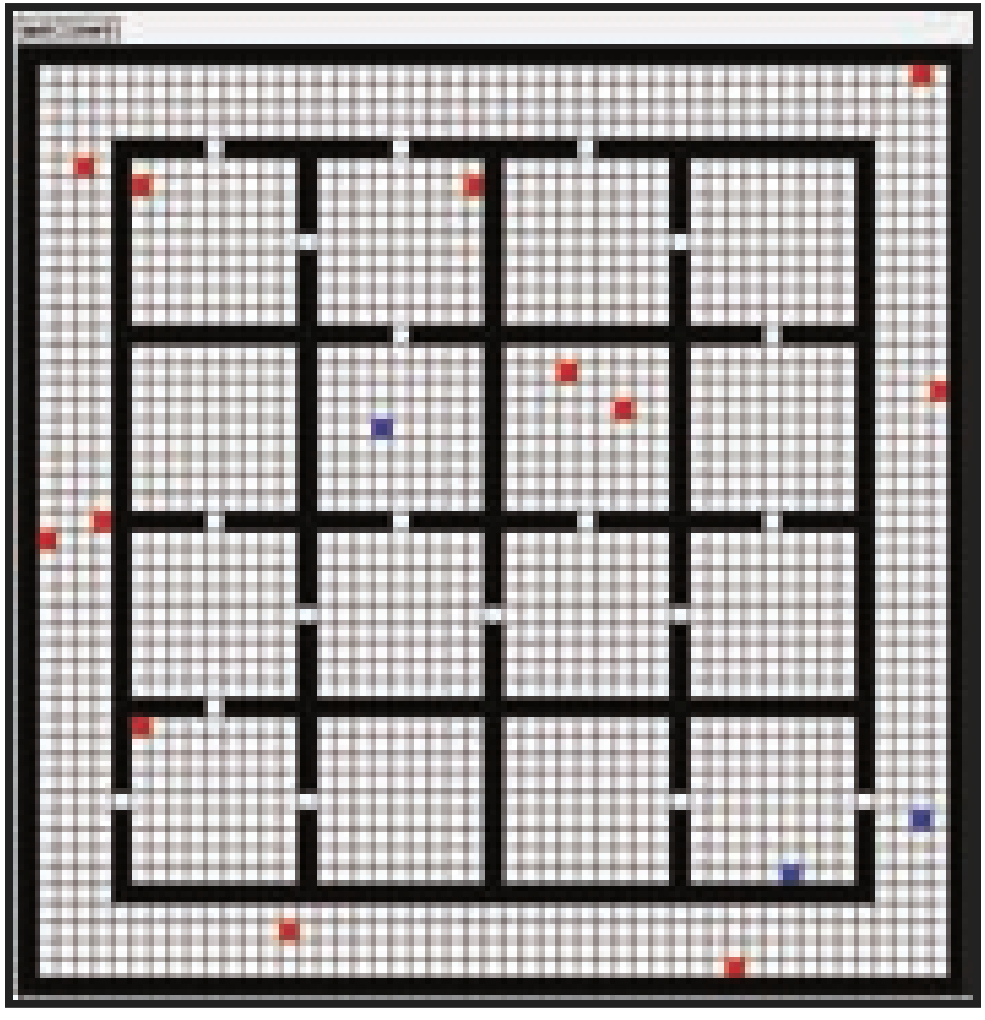
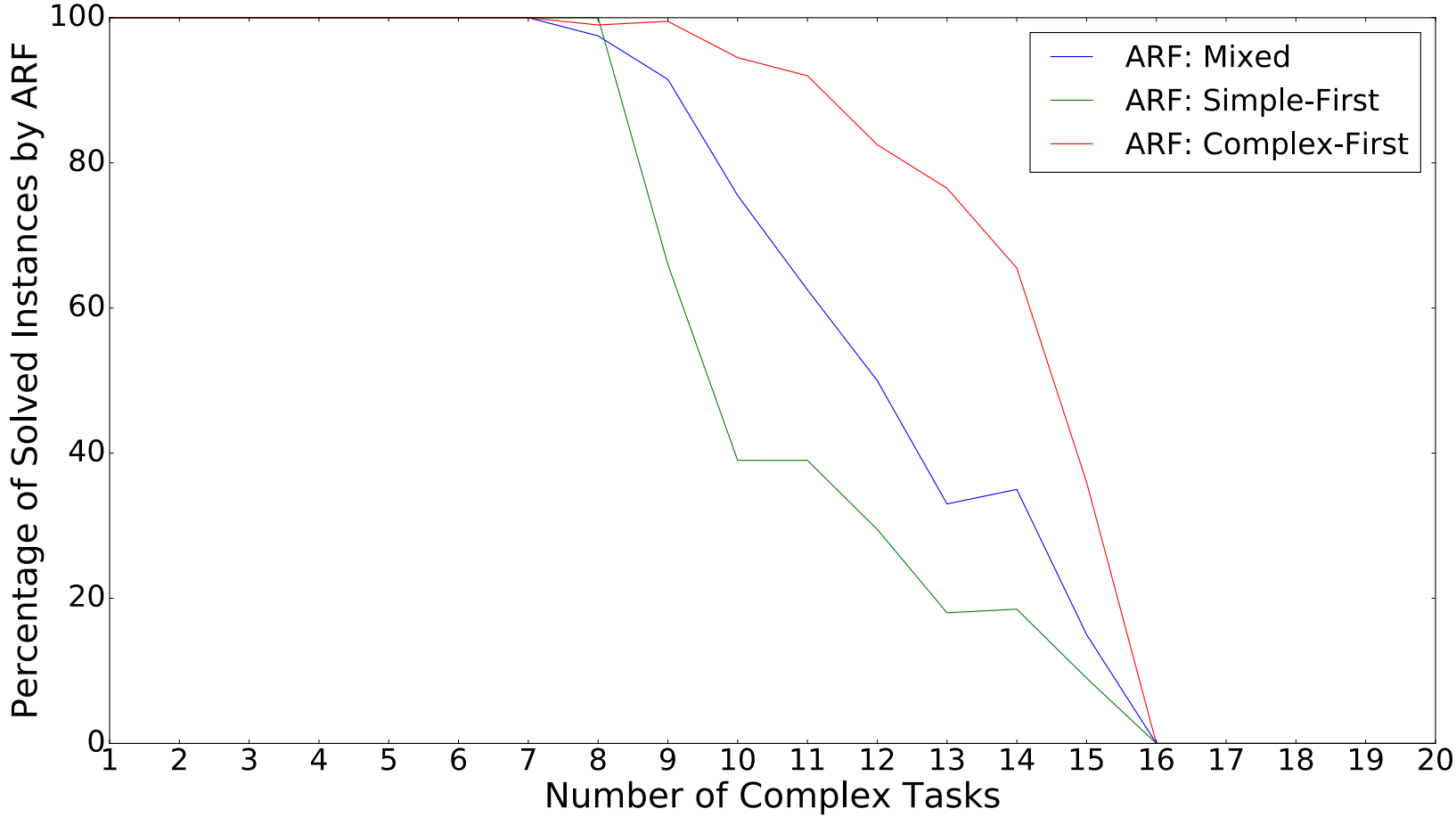


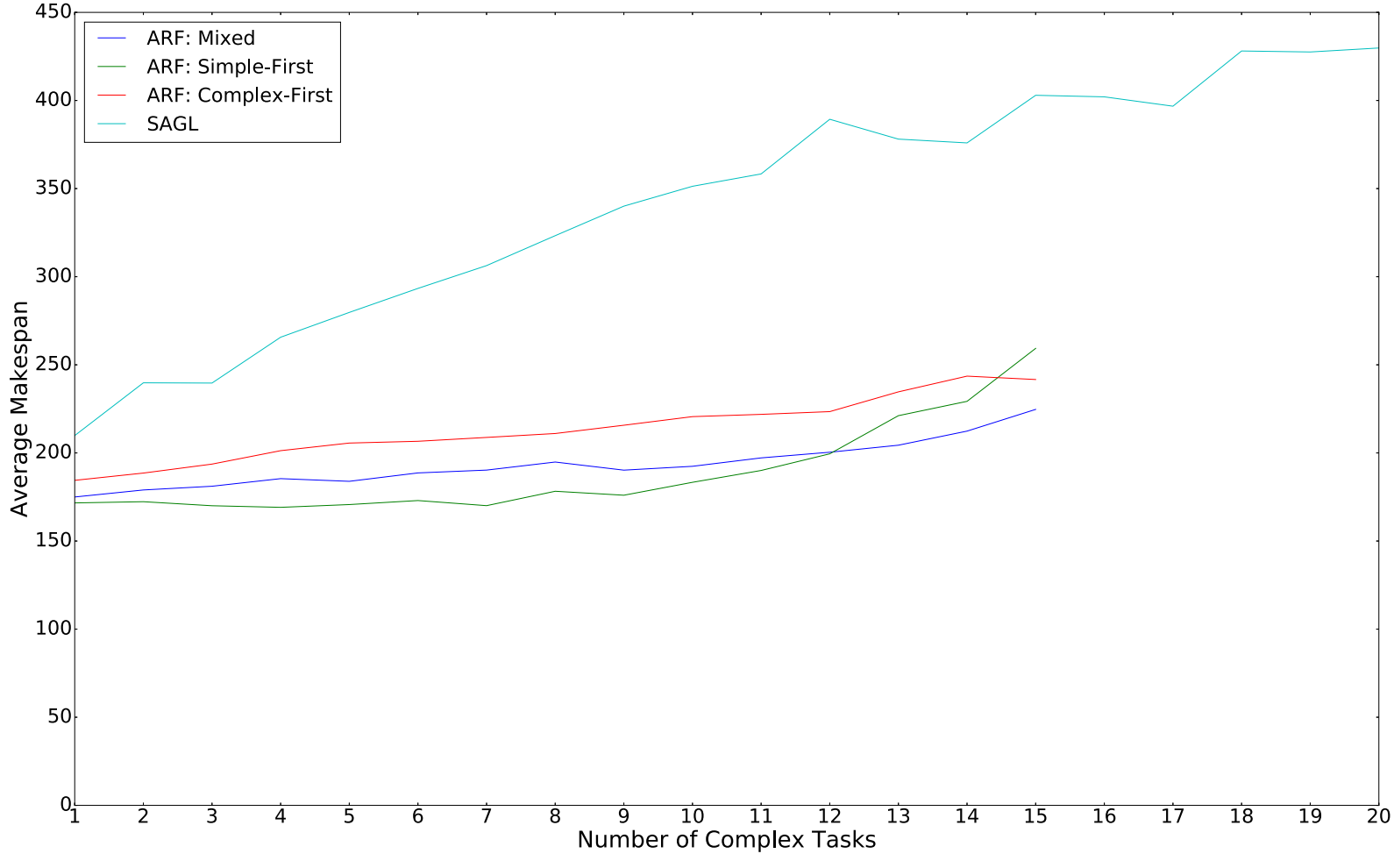
image source: [Koenig et al. '07]

COMPARED WITH ARF: EFFICIENCY



Percentage of instances solved by ARF within 2 minutes.
SAGL solved each of them within one second.

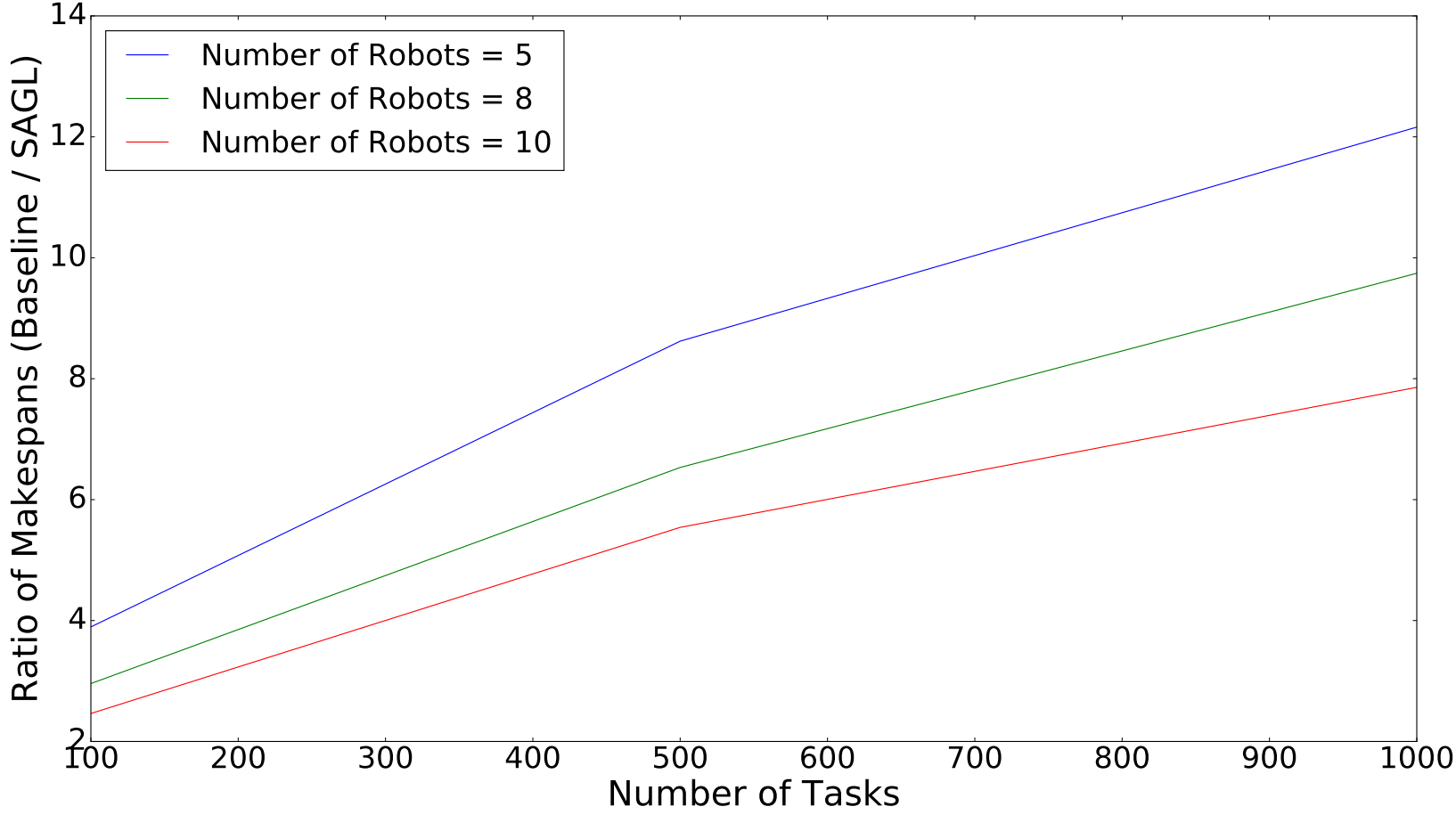
COMPARED WITH ARF: MAKESPAN



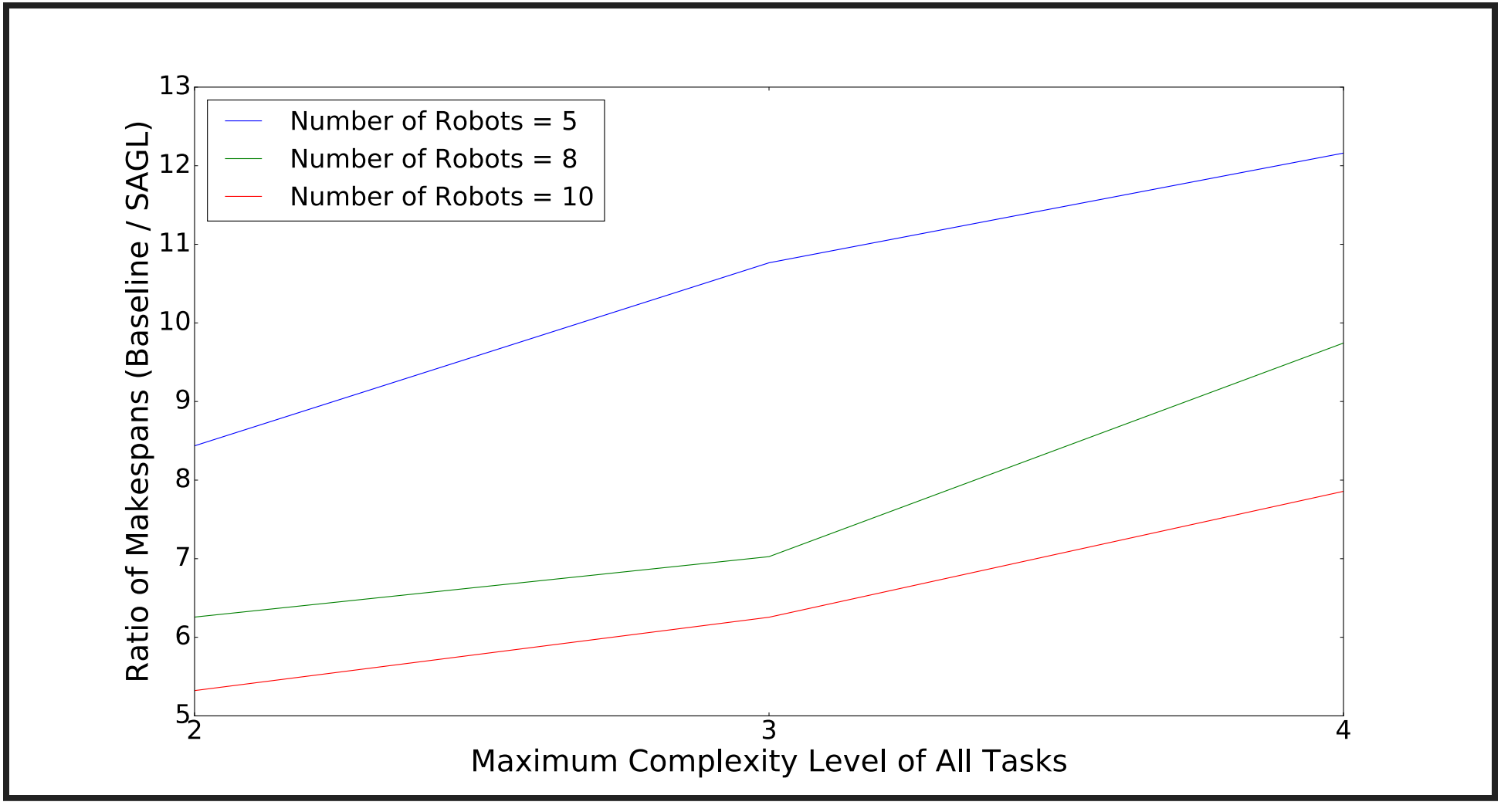
EXPERIMENT SET 2

- Large instances
- ARF cannot solve large instances
- Compared with a baseline algorithm
  - No spanning tree
  - Random global visitation order
  - No use of path-constrained TSP
- Map: obstacle free 300x300 continuous square
- 15 CRP instances for random
  - 5, 8 or 10 robots
  - 100, 500 or 1000 tasks
  - max complexity levels of 2, 3 or 4

SAGL VS BASELINE: MAKESPAN VS NUMBER OF TASKS



SAGL VS BASELINE: MAKESPAN VS MAXIMUM COMPLEXITY LEVELS



FUTURE WORK

- Heterogenous robots
- Distributed version
- Flexible complexity levels (task accomplishment time depends on number of robots)

CONCLUSION

- SAGL is
  - A polynomial time solver for the Complex Routing Problem
  - Four steps:
    1. Spanning tree construction
    2. Task assignment
    3. Global visitation order determination for complex tasks
    4. Local visitation order determination
  - More scalable than ARF
  - decent solution quality