

16-350

Planning Techniques for Robotics

Multi-Robot Planning

Maxim Likhachev

Robotics Institute

Carnegie Mellon University

Different Categorizations of Multi-Robot Planning

- Centralized vs. Decentralized
 - **Centralized:** one central control of (planning for) all the robots
 - **Decentralized:** each robot decides/plans what to do on its own, so that the overall team of robots accomplishes the overall goal

Different Categorizations of Multi-Robot Planning

- Centralized vs. Decentralized
 - **Centralized:** one central control of (planning for) all the robots
 - **Decentralized:** each robot decides/plans what to do on its own

Why do we need decentralized planning approaches?

Different Categorizations of Multi-Robot Planning

- Centralized vs. Decentralized
 - **Centralized:** one central control of (planning for) all the robots
 - **Decentralized:** each robot decides/plans what to do on its own

Why do we need decentralized planning approaches?

Robust to limits on or loss of communication

Robust to losing some robots in the team

Computationally more scalable

Different Categorizations of Multi-Robot Planning

- Centralized vs. Decentralized
 - **Centralized:** one central control of (planning for) all the robots
 - **Decentralized:** each robot decides/plans what to do on its own

Why do we need decentralized planning approaches?

Robust to limits on or loss of communication

Robust to losing some robots in the team

Computationally more scalable

Challenges with decentralized planning?

Different Categorizations of Multi-Robot Planning

- Centralized vs. Decentralized
 - **Centralized:** one central control of (planning for) all the robots
 - **Decentralized:** each robot decides/plans what to do on its own

Why do we need decentralized planning approaches?

Robust to limits on or loss of communication

Robust to losing some robots in the team

Computationally more scalable

Challenges with decentralized planning?

How to guarantee that the overall team accomplishes its goal?

Different Categorizations of Multi-Robot Planning

- Multi-robot Path Planning vs. Multi-robot Cooperative Task Planning
 - **Multi-robot Path Planning:** how to plan paths for N robots so that they don't collide with each other during execution
 - **Multi-robot Cooperative Task Planning:** how to compute plans for N robots so that they achieve the overall goal that may require cooperation

Different Categorizations of Multi-Robot Planning

- Small teams vs. large teams (swarms) of robots
 - **Planning for small teams:** Compute plans for N (potentially heterogeneous) robots, where N is typically 2-10
 - **Planning for (control of) swarms of robots:** how to control a swarm of N (usually homogeneous) robots, where N is typically 10-1000

Different Categorizations of Multi-Robot Planning

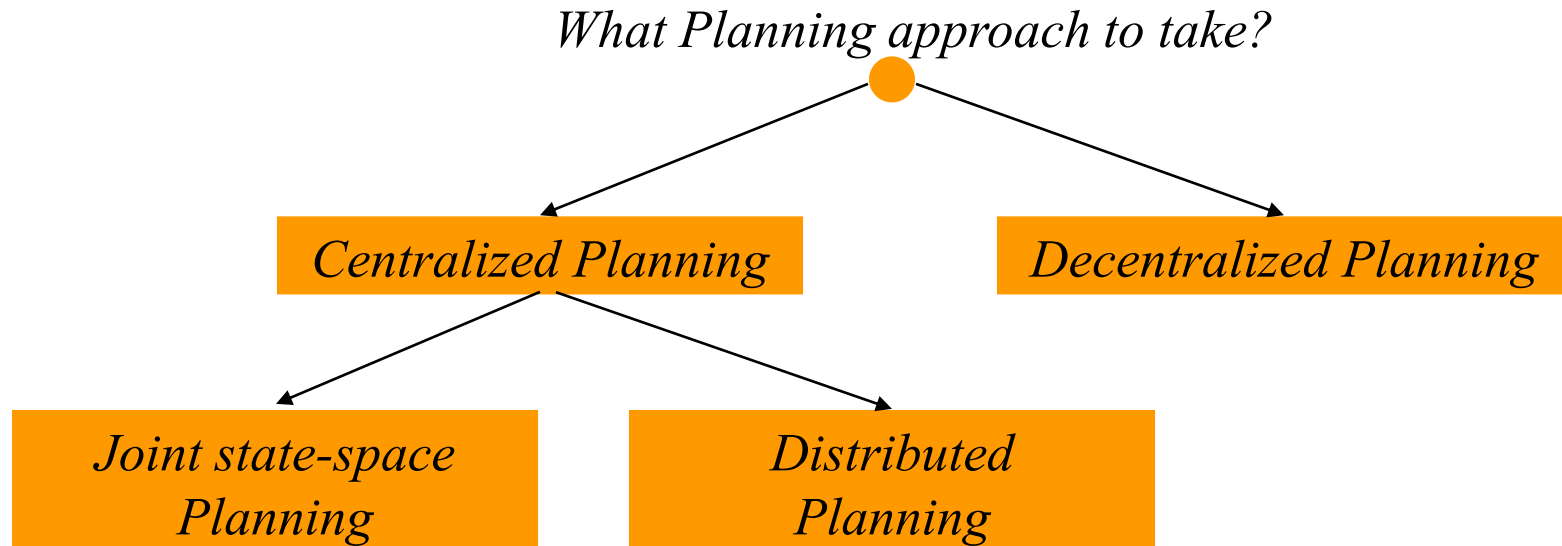
- Small teams vs. large teams (swarms) of robots
 - **Planning for small teams:** Compute plans for N (potentially heterogeneous) robots, where N is typically 2-10
 - **Planning for (control of) swarms of robots:** how to control a swarm of N (usually homogeneous) robots, where N is typically 10-1000

Where do you think centralized approaches are used more often?

Control of swarms is typically decentralized

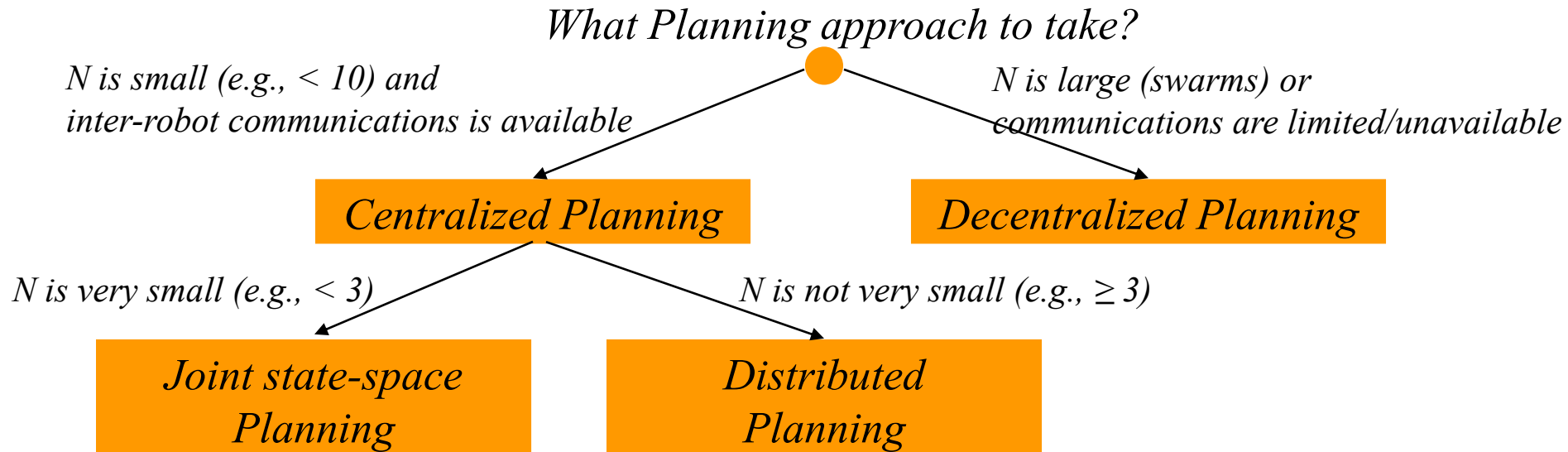
Different Categorizations of Multi-Robot Planning

- Joint state-space vs. distributed planning (within centralized)
 - **Joint state-space planning:** Planning for N robots in a state-space that represents joint configurations of robots
 - **Distributed planning:** Planning is split into N individual planners that share their results (and potentially re-plan) to obtain a final plan for all N robots



Different Categorizations of Multi-Robot Planning

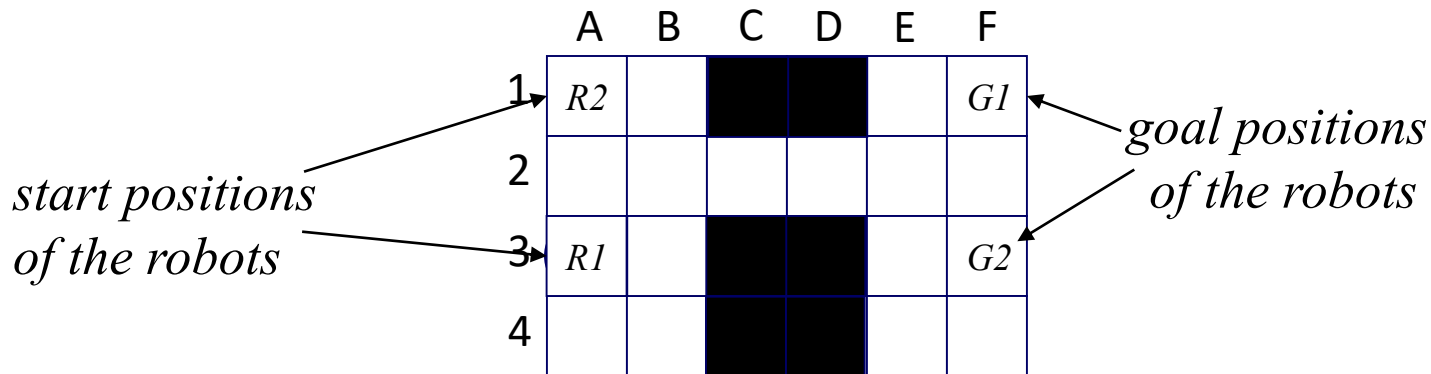
- Joint state-space vs. distributed planning (within centralized)
 - **Joint state-space planning:** Planning for N robots in a state-space that represents joint configurations of robots
 - **Distributed planning:** Planning is split into N individual planners that share their results (and potentially re-plan) to obtain a final plan for all N robots



Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

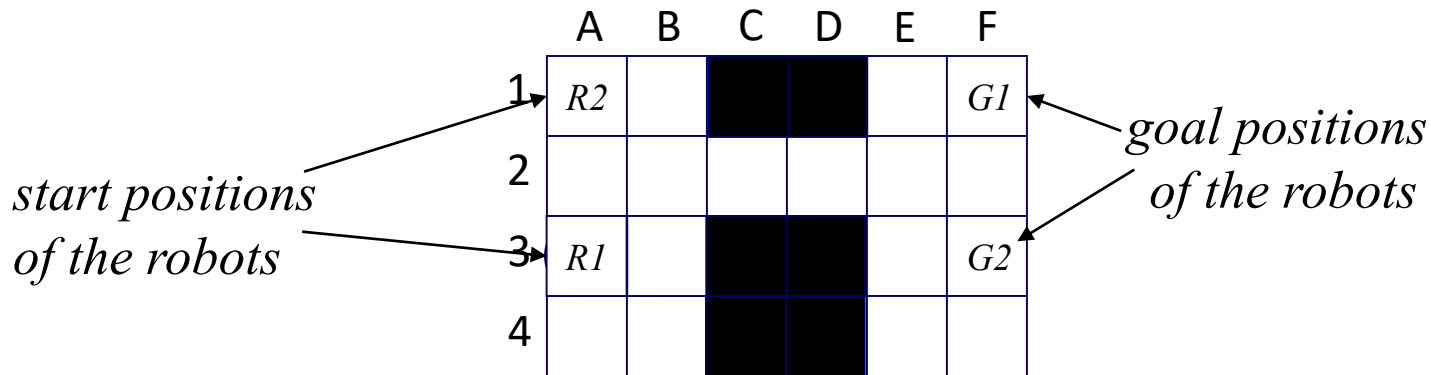
simple example for two omnidirectional point-size robots



Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

simple example for two omnidirectional point-size robots



Any examples of this in industry?

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Joint state-space planning

	A	B	C	D	E	F
1	$R2$					$G1$
2						
3	$R1$					$G2$
4						

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Joint state-space planning

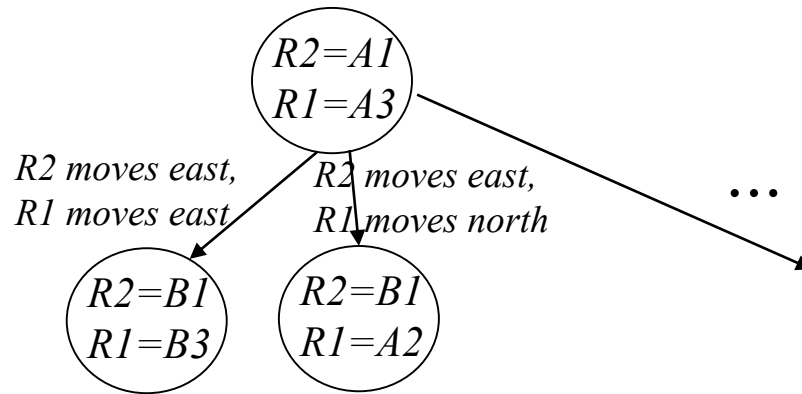
	A	B	C	D	E	F
1	$R2$					$G1$
2						
3	$R1$					$G2$
4						

The simplest approach: construct and search a graph, where each state encodes positions of all the robots and each action encodes all possible movements

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Joint state-space planning



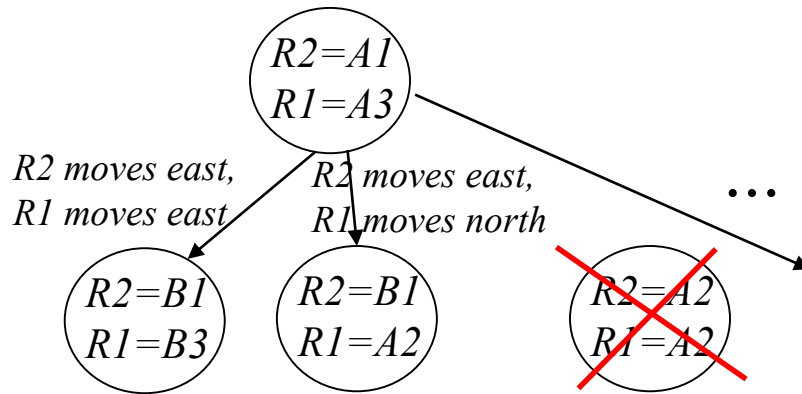
	A	B	C	D	E	F
1	$R2$		■	■		$G1$
2						
3	$R1$		■	■		$G2$
4			■	■		

The simplest approach: construct and search a graph, where each state encodes positions of all the robots and each action encodes all possible movements

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Joint state-space planning



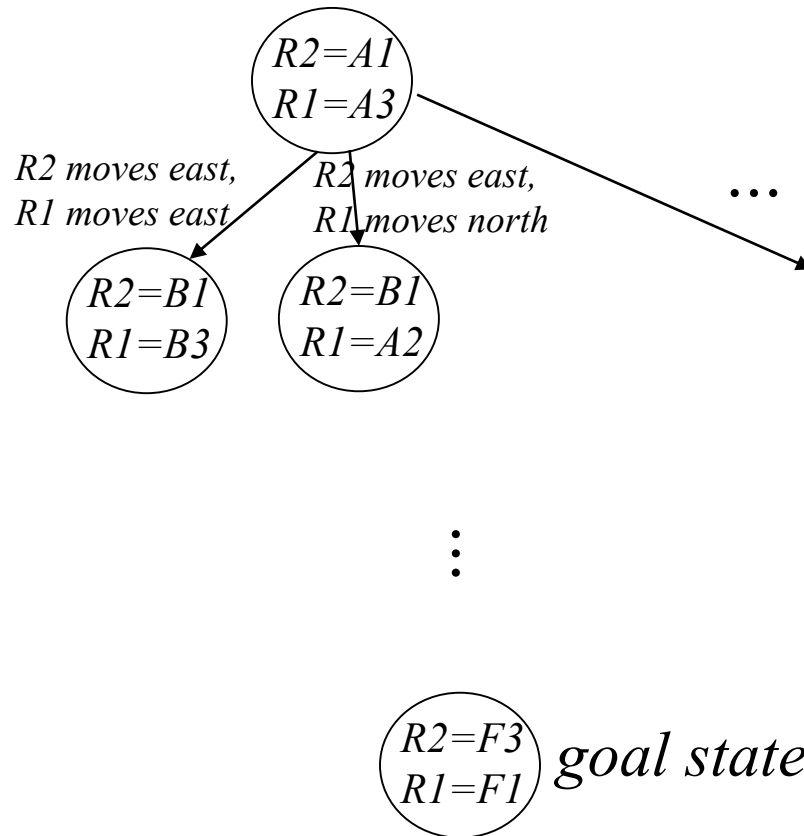
	A	B	C	D	E	F
1	$R2$		■	■		$G1$
2						
3	$R1$		■	■		$G2$
4			■	■		

The simplest approach: construct and search a graph, where each state encodes positions of all the robots and each action encodes all possible movements

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Joint state-space planning



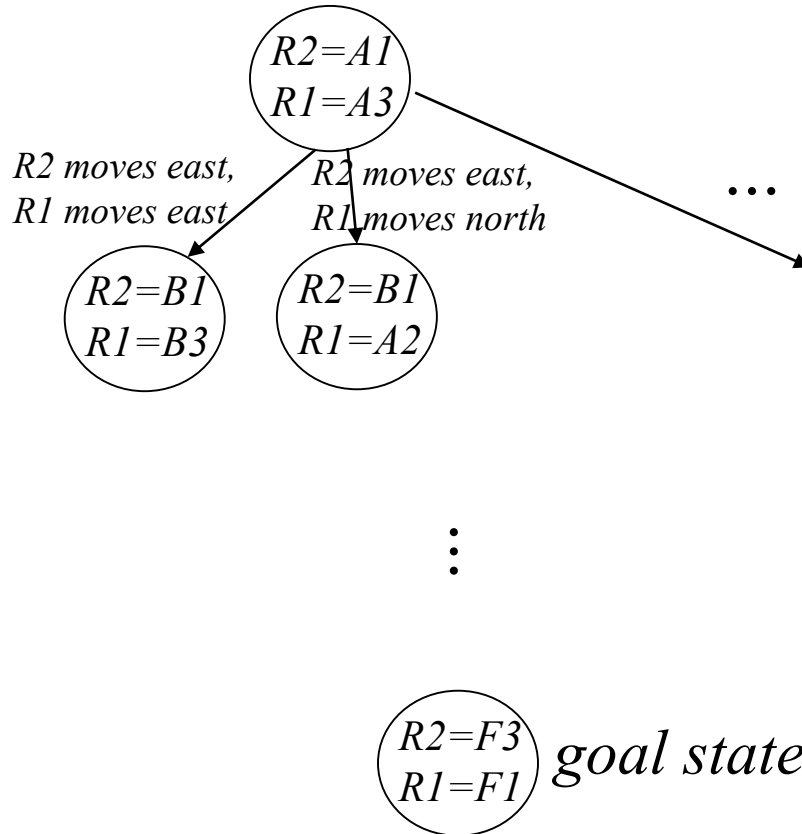
	A	B	C	D	E	F
1	$R2$		■	■		$G1$
2						
3	$R1$		■	■		$G2$
4			■	■		

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Assuming 4-connected grid,
what is the maximum branching factor
(how many actions/successors)?

Joint state-space planning



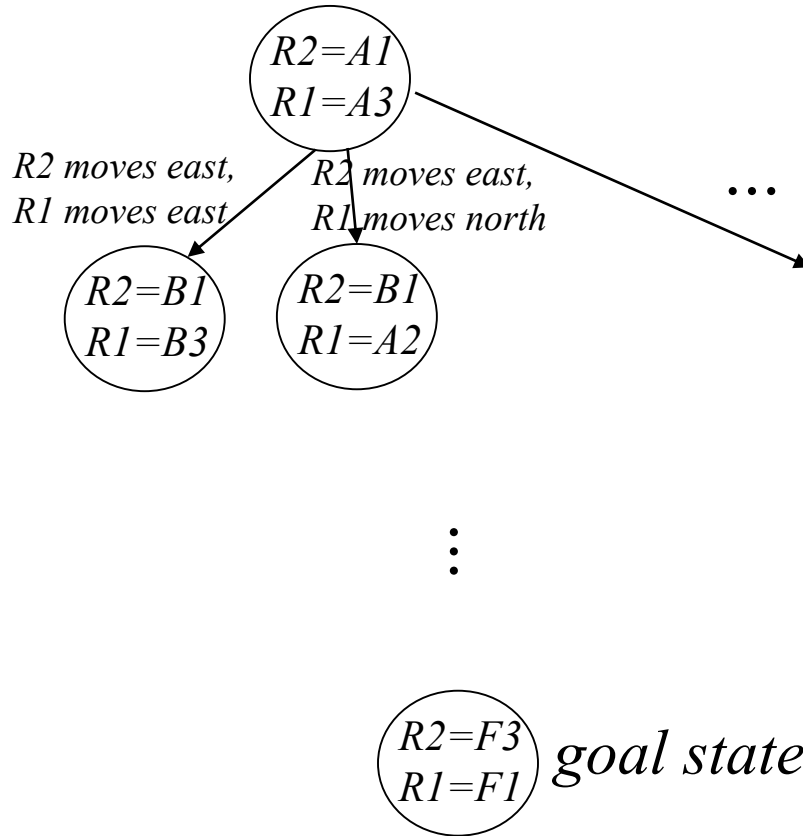
	A	B	C	D	E	F
1	$R2$					$G1$
2						
3	$R1$					$G2$
4						

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

What is the size of the graph?

Joint state-space planning

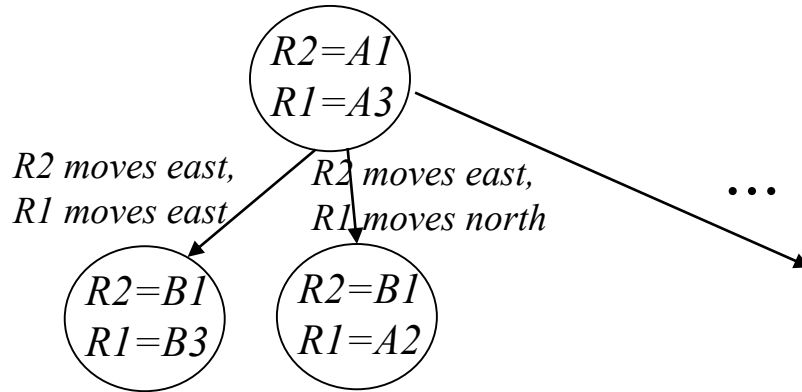


	A	B	C	D	E	F
1	R2					G1
2						
3	R1					G2
4						

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Joint state-space planning



⋮



What is the size of the graph?

	A	B	C	D	E	F
1	R2					G1
2						
3	R1					G2
4						

*Scalability w.r.t. N is clearly an issue!
Good heuristic function is critical*

Any ideas for the heuristics?

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Distributed planning

	A	B	C	D	E	F
1	$R2$		■	■		$G1$
2						
3	$R1$		■	■		$G2$
4			■	■		

One popular approach: Prioritized Planning

For $i = 1:N$

Compute path for robot R_i that avoids collisions with paths for robots $R_1..R_{i-1}$

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Distributed planning

	A	B	C	D	E	F
1	$R2$		■	■		$G1$
2						
3	$R1$		■	■		$G2$
4			■	■		

What will be the plan returned by Prioritized Planning?

One popular approach: Prioritized Planning

For $i = 1:N$

Compute path for robot R_i that avoids collisions with paths for robots $R_1..R_{i-1}$

Multi-Robot Path Planning

- Path planning for N robots to get to their goals w/o collisions

Distributed planning

Is it complete?

Is it optimal?

What is the complexity of Prioritized Planning?

One popular approach: Prioritized Planning

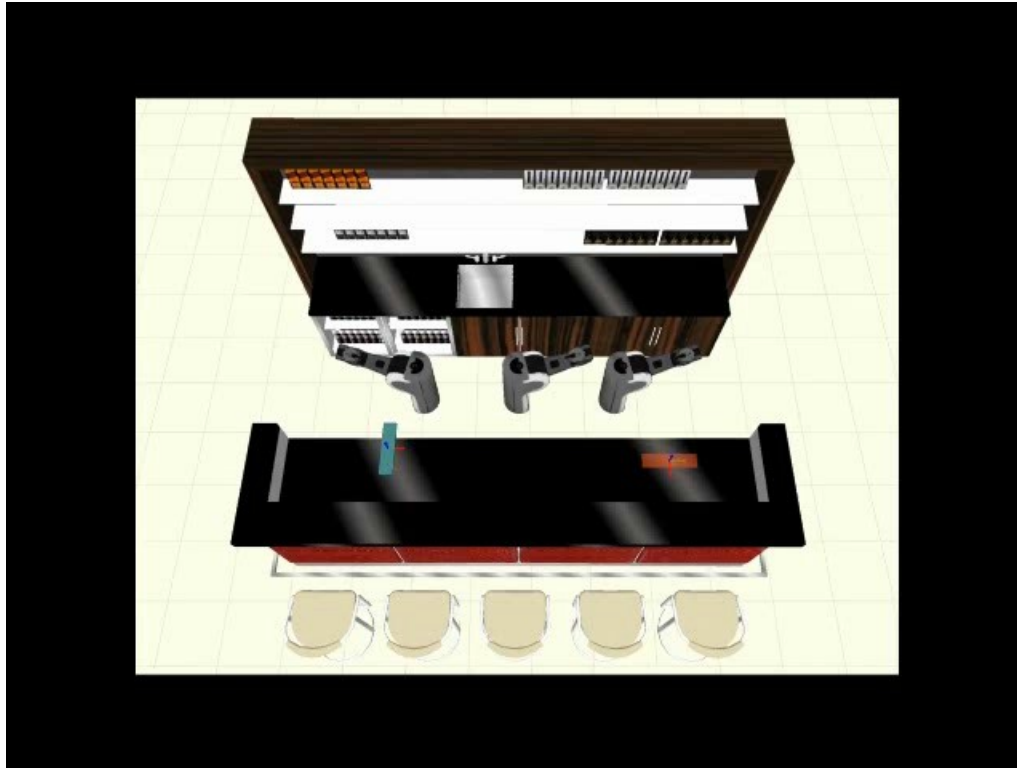
For $i = 1:N$

Compute path for robot R_i that avoids collisions with paths for robots $R_1..R_{i-1}$

	A	B	C	D	E	F
1	R_2		■	■		G_1
2						
3	R_1		■	■		G_2
4			■	■		

Multi-Robot Cooperative Planning/Task Allocation

- Example: planning for N robotic arms to move an object



*[Cohen et al., '14]
(performs joint state-space planning)*

Multi-Robot Cooperative Planning/Task Allocation

- Example: planning for N robotic arms to move an object



Search-Based Planning Lab

Master of Science
Robotic Systems Development | Carnegie Mellon University
The Robotics Institute

Collaborative Manipulation

Dr. Maxim Likhachev

Ishani Chatterjee	Clare Cui
Andrew Dornbush	Brad Factor
Sung Kyun Kim	Maitreya Naik
Tae-Hyung Kim	Angad Sidhu
Venkatraman Narayanan	Logan Wan

(planning is distributed: plan on Roman platform first, then on PR2)

Multi-Robot Cooperative Planning/Task Allocation

- Example: planning for multi-robot exploration/mapping

N robots need to explore and build a map of unknown environment

One approach: Distributed Greedy Mapping

For $i = 1:N$

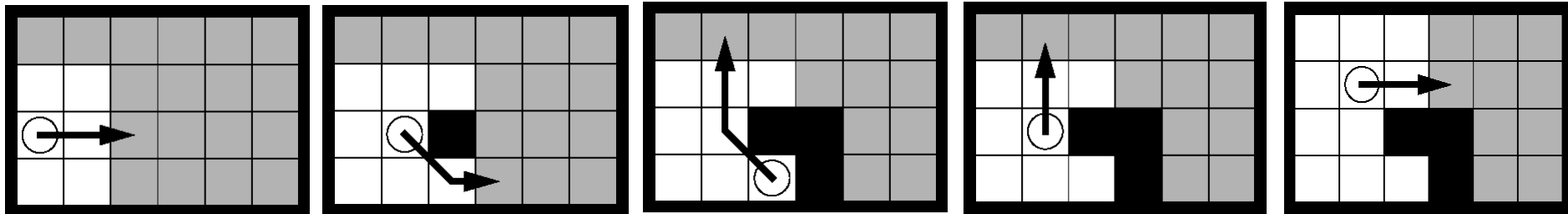
Compute a path using Greedy Mapping approach for robot R_i taking into account what paths were computed for $R_1..R_{i-1}$ (and what cells they will see)

Multi-Robot Cooperative Planning/Task Allocation

- Example: planning for multi-robot exploration/mapping

N robots need to explore and build a map of unknown environment

- Greedy Mapping for a single robot:
 - always move the robot on a shortest path to the closest unobserved (or unvisited) cell
 - it always achieves a gain in information.
 - thus, it is guaranteed to map the environment that is reachable (assuming all moves are reversible)



One approach: Distributed Greedy Mapping

For $i = 1:N$

Compute a path using Greedy Mapping approach for robot R_i taking into account what paths were computed for $R_1..R_{i-1}$ (and what cells they will see)

Multi-Robot Cooperative Planning/Task Allocation

- Example: planning for multi-robot exploration/mapping

N robots need to explore and build a map of unknown environment



[Butzke et al., '11]

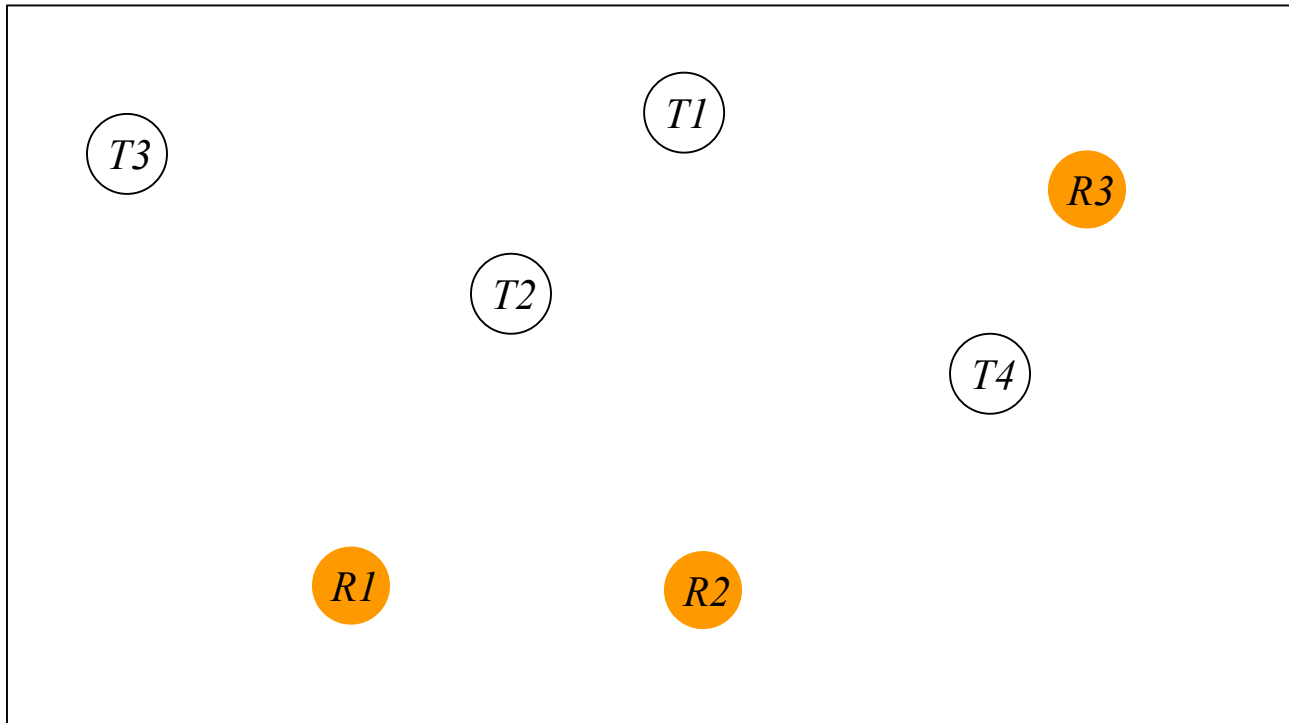
One approach: Distributed Greedy Mapping

For $i = 1:N$

Compute a path using Greedy Mapping approach for robot R_i taking into account what paths were computed for $R_1..R_{i-1}$ (and what cells they will see)

Multi-Robot Cooperative Planning/Task Allocation

- Market-based approach (very popular distributed approach)
 - Consider planning the allocation of tasks to N robots
 - General scheme: *robots auction out their tasks to their teammates with the goal of increasing their own revenue*

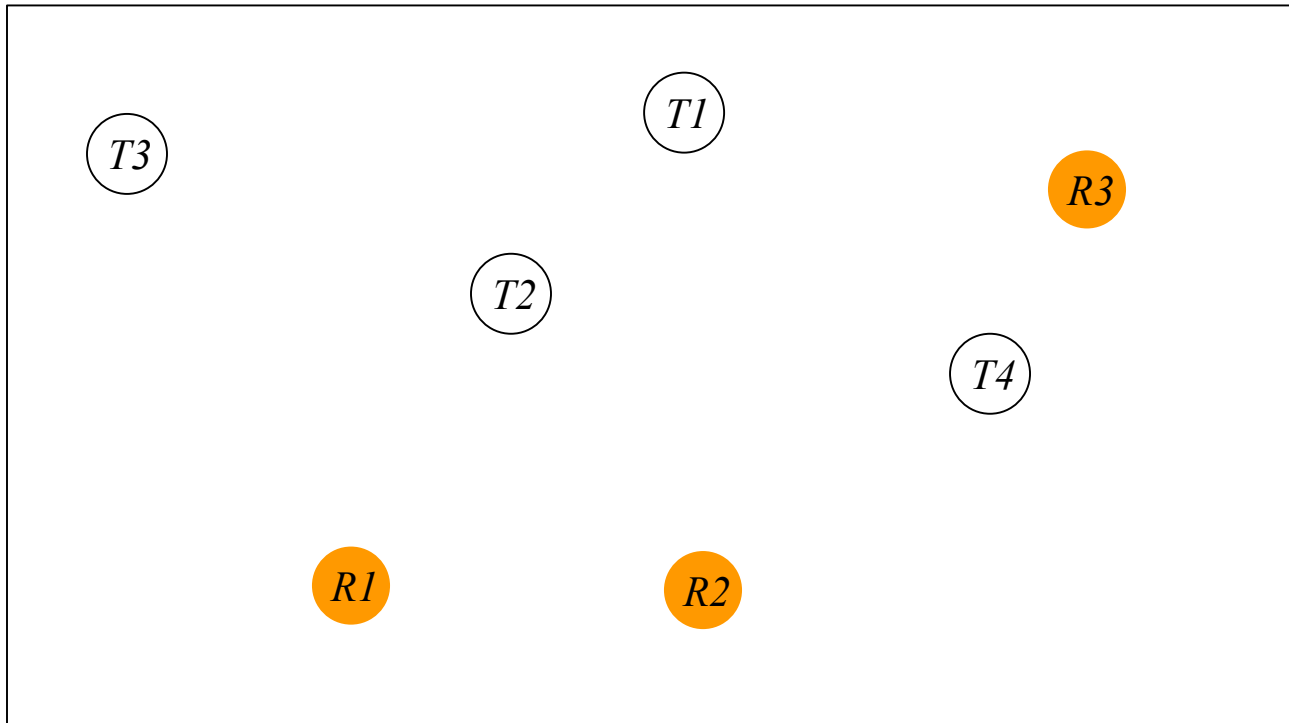


Market-based Approach

Given N robots $R_1 \dots R_N$, M tasks $T_1 \dots T_M$, and $C_i^{R_j}$ – cost of executing task i by robot R_j (cost may depend on other tasks executed by this robot)

Planner needs to decide: Which task gets executed by which robot?

Find a plan(mapping) π^ : $T_i \rightarrow R_j$ such that $\pi^* = \operatorname{argmin} \Sigma C_i^{\pi(T_i)}$*



Market-based Approach

Given N robots $R_1 \dots R_N$, M tasks $T_1 \dots T_M$, and $C_i^{R_j}$ – cost of executing task i by robot R_j (cost may depend on other tasks executed by this robot)

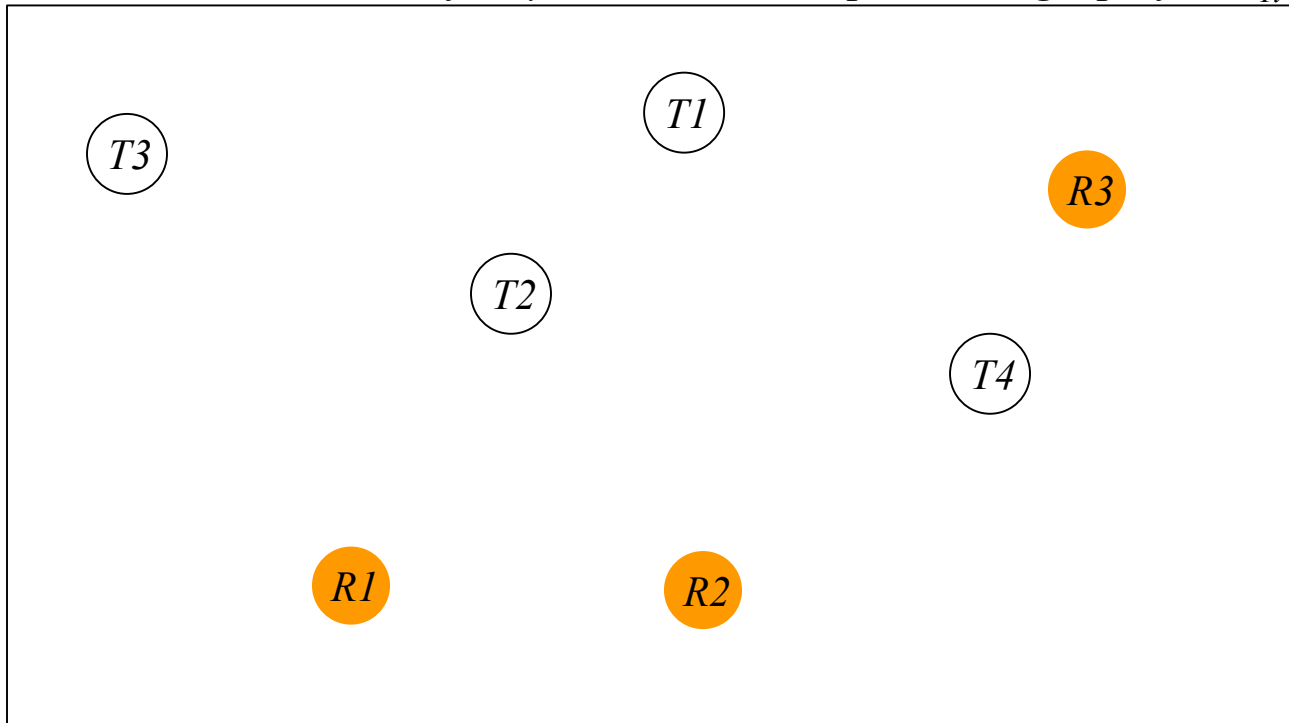
Iterate over steps 1-4 until convergence or planning time expires

Step 1: start with an arbitrary plan π

Step 2: all robots offer their tasks T_i at auction at the max. price of $C_{T_i}^{\pi(T_i)} - \epsilon$

Step 3: all robots R_j bid on the offered tasks T_i with the bid = $C_i^{R_j} + \epsilon$

Step 4: robots sell to the lowest bidders if they are below max. price and get profit: $C_{T_i}^{\pi(T_i)} - C_i^{R_j}$



Market-based Approach

Given N robots $R_1 \dots R_N$, M tasks $T_1 \dots T_M$, and $C_i^{R_j}$ – cost of executing task i by robot R_j (cost may depend on other tasks executed by this robot)

When does it converge in one iteration?

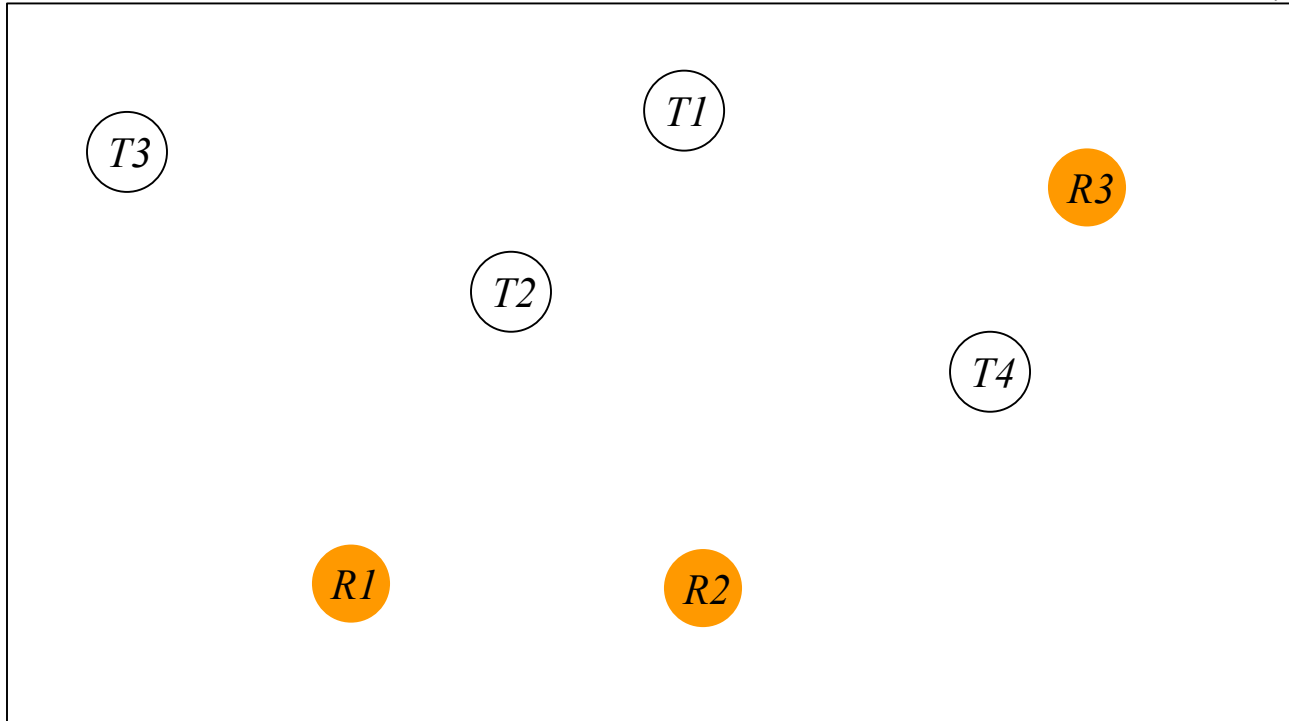
Iterate over steps 1-4 until convergence or planning time expires

Step 1: start with an arbitrary plan π

Step 2: all robots offer their tasks T_i at auction at the max. price of $C_{T_i}^{\pi(T_i)} - \epsilon$

Step 3: all robots R_j bid on the offered tasks T_i with the bid = $C_i^{R_j} + \epsilon$

Step 4: robots sell to the lowest bidders if they are below max. price and get profit: $C_{T_i}^{\pi(T_i)} - C_i^{R_j}$



Summary

- Multi-robot planning
 - currently very popular research area
 - has a large number of algorithms developed (class on its own)
 - still a challenging planning problem
- When solving, always think first about the requirements (e.g., team size, type of problem, communication availability)

