Motion Planning for Unlabeled Discs with Optimality Guarantees

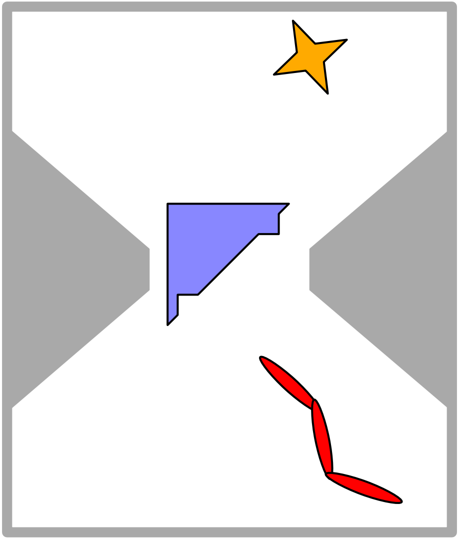
Kiril Solovey

Tel Aviv University, Israel

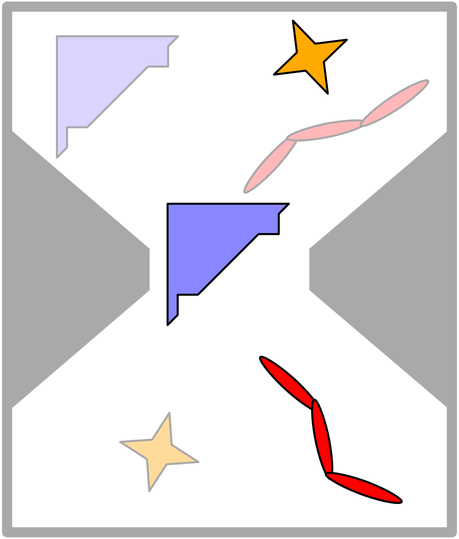
Israeli Conference on Robotics, 2016

* Joint work with Jingjin Yu, Or Zamir and Dan Halperin

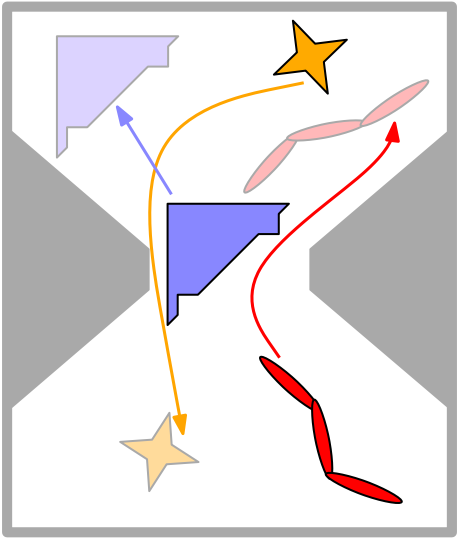
Multi-robot motion planning



Multi-robot motion planning



Multi-robot motion planning



Complete combinatorial techniques, related work

Planning using exact analytic methods, often using an explicit construction of the free space.

- Fixed number of robots:
  - Two discs in $O(n^3)$ [Schwartz and Sharir, 83]
  - Two discs in $O(n^2)$ [Yap, 83]
  - Two discs (and other types) in $O(n^2)$ [Sharir and Sifrony, 91]

Complete combinatorial techniques, related work

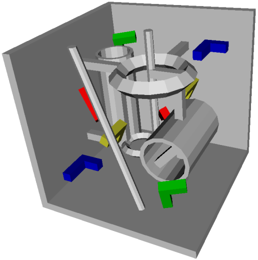
Planning using exact analytic methods, often using an explicit construction of the free space.

- Fixed number of robots:
  - Two discs in $O(n^3)$ [Schwartz and Sharir, 83]
  - Two discs in $O(n^2)$ [Yap, 83]
  - Two discs (and other types) in $O(n^2)$ [Sharir and Sifrony, 91]
- Multiple robots:
  - PSPACE-hardness for translating rectangles [Hopcroft et al., 84], [Hearn and Demaine, 05]
  - NP-hardness for discs [Spirakis and Yap, 84]

Sampling-based methods for multi-robot, related work

Capturing the connectivity of the free space by random sampling.

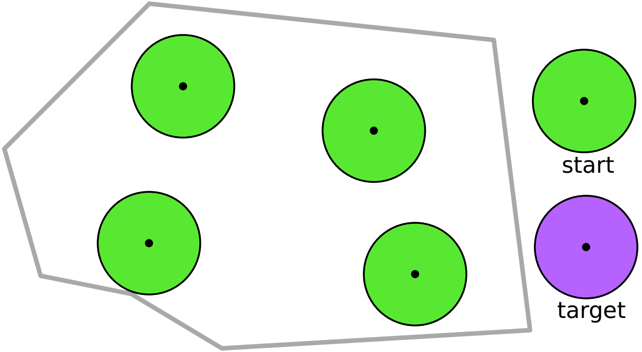
- Sampling-based planners that are tailored for the multi-robot case have been proposed [Švestka and Overmars, 98], [van den Berg et al., 09], [Wagner and Choset, 11], [S. et al., 14] ...
- Sampling-based planners can cope with complex types of scenarios and robots
- However, they provide only (weak) asymptotic guarantees



Unlabeled multi-robot motion planning

A variant of the multi-robot problem in which the robots are identical and indistinguishable [Kloder and Hutchinson, 06].

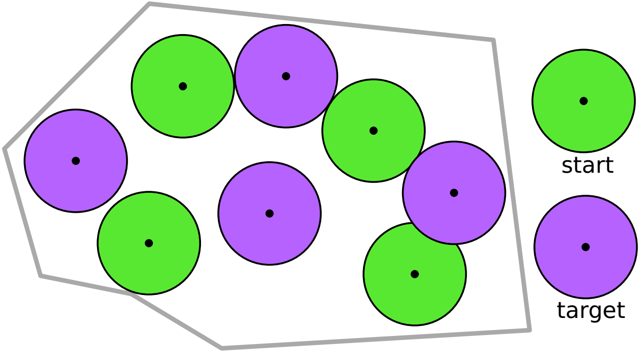
Goal: Move the robots such that each target will be occupied by *some* robot, while avoiding collisions.



Unlabeled multi-robot motion planning

A variant of the multi-robot problem in which the robots are identical and indistinguishable [Kloder and Hutchinson, 06].

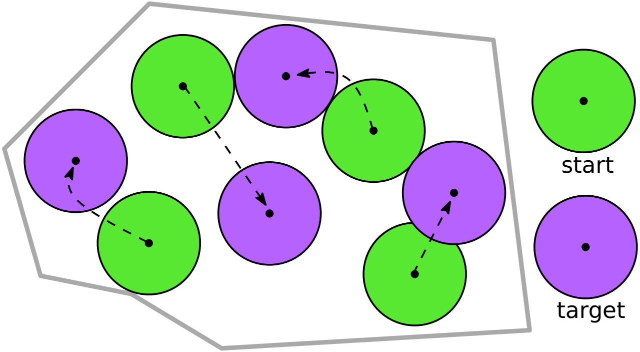
Goal: Move the robots such that each target will be occupied by *some* robot, while avoiding collisions.



Unlabeled multi-robot motion planning

A variant of the multi-robot problem in which the robots are identical and indistinguishable [Kloder and Hutchinson, 06].

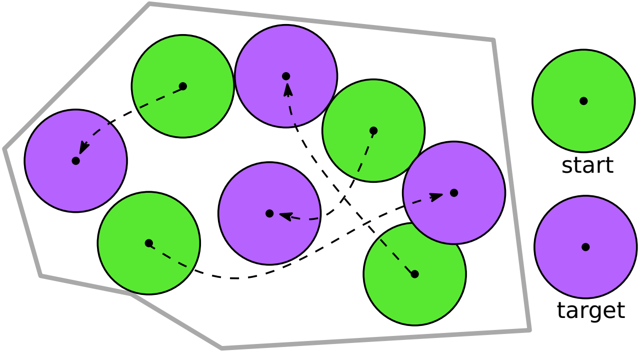
Goal: Move the robots such that each target will be occupied by *some* robot, while avoiding collisions.



Unlabeled multi-robot motion planning

A variant of the multi-robot problem in which the robots are identical and indistinguishable [Kloder and Hutchinson, 06].

Goal: Move the robots such that each target will be occupied by *some* robot, while avoiding collisions.

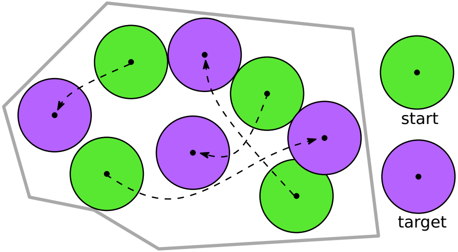


Unlabeled multi-robot motion planning: related results

In general, the unlabeled problem is PSPACE-hard [S. & Halperin, 15].

However, by making certain separation assumptions the unlabeled problem can be solved efficiently. The following algorithms work in the setting of m unit-disc robots in an environment of complexity n:

- $O(m^2 + mn)$-time algorithm [Adler et al., 14]
- $\tilde{O}(m^4 + m^2 n)$-time algorithm that returns the optimal solution in terms of the longest path length [Turpin et al., 13]



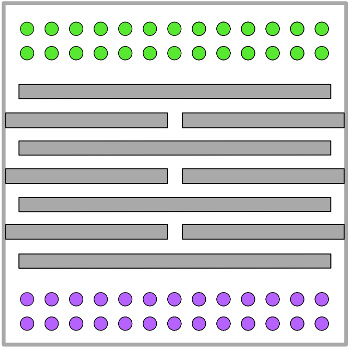
Contribution

We consider the specific setting of unit-disc robot operating in a workspace cluttered with polygonal obstacles.

We develop an $\tilde{O}(m^4 + m^2 n^2)$-time algorithm, where m is the number of robots and n is the complexity of the workspace.

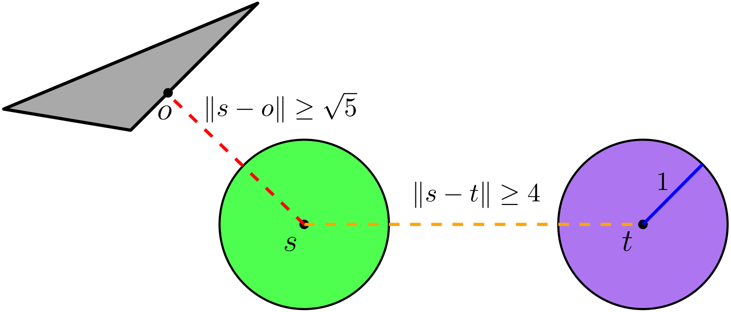
The algorithm minimizes the total path length, i.e., the sum of lengths of the individual paths of the robots.

Specifically, the algorithm is guaranteed to return a solution whose cost is at most $\mathrm{OPT} + O(m)$.



Our setting

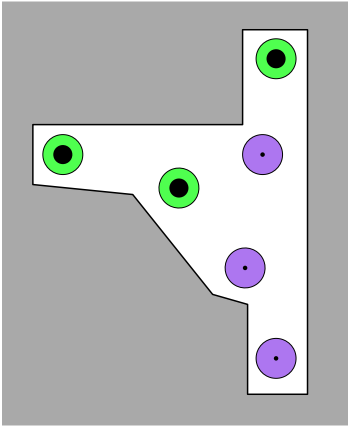
We study the unlabeled problem for unit discs moving amid polygonal obstacles, and make two simplifying assumptions:



Standalone targets

The main ingredient in our algorithm is the following observation.

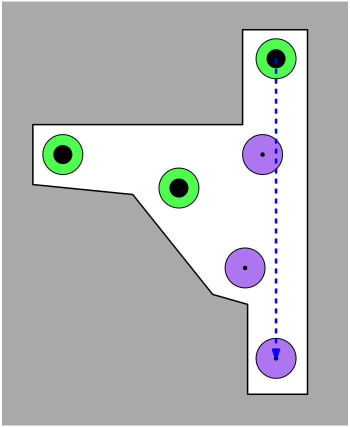
- Suppose that the robots are resting at their start positions
- One of the robots can be moved to one of the targets positions while the remaining robots stay put
- The other robots would still be able to reach their targets afterwards



Standalone targets

The main ingredient in our algorithm is the following observation.

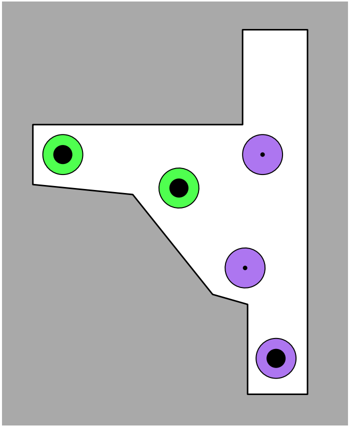
- Suppose that the robots are resting at their start positions
- One of the robots can be moved to one of the targets positions while the remaining robots stay put
- The other robots would still be able to reach their targets afterwards



Standalone targets

The main ingredient in our algorithm is the following observation.

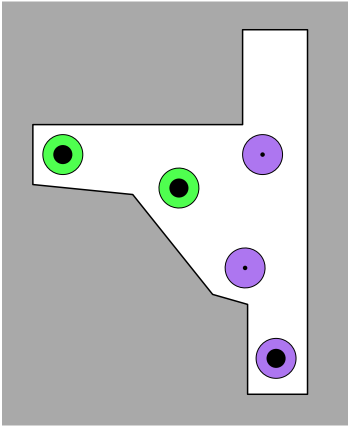
- Suppose that the robots are resting at their start positions
- One of the robots can be moved to one of the targets positions while the remaining robots stay put
- The other robots would still be able to reach their targets afterwards



Algorithmic implications

We perform the following steps in each iteration of the algorithm:

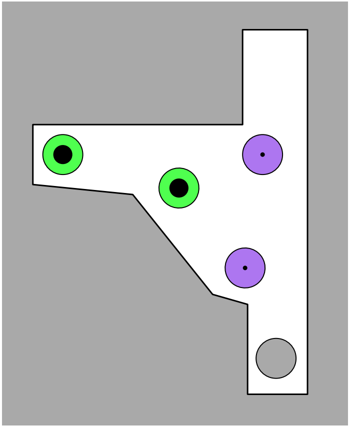
1. Find a robot that can be moved to a target without blocking consecutive moves and execute the motion
2. This robot will stay put for the rest of the execution, i.e., will be treated as an obstacle
3. Repeat



Algorithmic implications

We perform the following steps in each iteration of the algorithm:

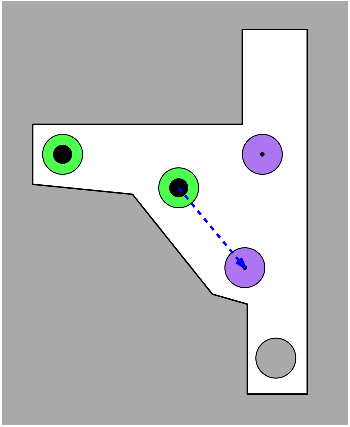
1. Find a robot that can be moved to a target without blocking consecutive moves and execute the motion
2. This robot will stay put for the rest of the execution, i.e., will be treated as an obstacle
3. Repeat



Algorithmic implications

We perform the following steps in each iteration of the algorithm:

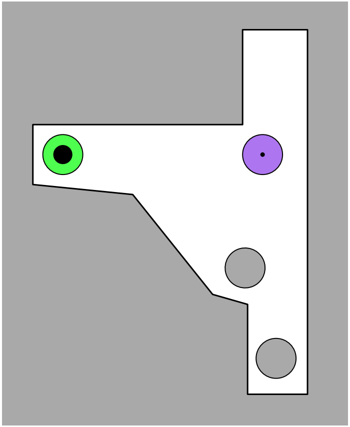
1. Find a robot that can be moved to a target without blocking consecutive moves and execute the motion
2. This robot will stay put for the rest of the execution, i.e., will be treated as an obstacle
3. Repeat



Algorithmic implications

We perform the following steps in each iteration of the algorithm:

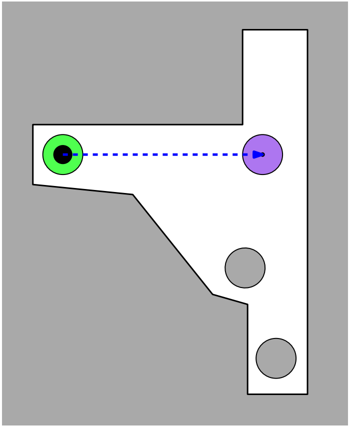
1. Find a robot that can be moved to a target without blocking consecutive moves and execute the motion
2. This robot will stay put for the rest of the execution, i.e., will be treated as an obstacle
3. Repeat



Algorithmic implications

We perform the following steps in each iteration of the algorithm:

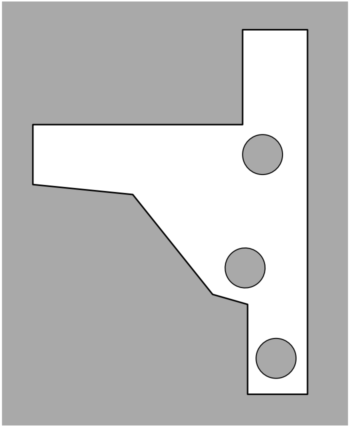
1. Find a robot that can be moved to a target without blocking consecutive moves and execute the motion
2. This robot will stay put for the rest of the execution, i.e., will be treated as an obstacle
3. Repeat



Algorithmic implications

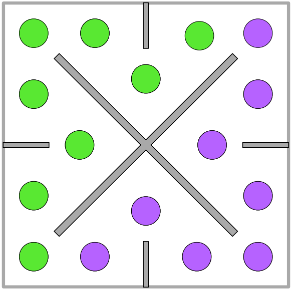
We perform the following steps in each iteration of the algorithm:

1. Find a robot that can be moved to a target without blocking consecutive moves and execute the motion
2. This robot will stay put for the rest of the execution, i.e., will be treated as an obstacle
3. Repeat

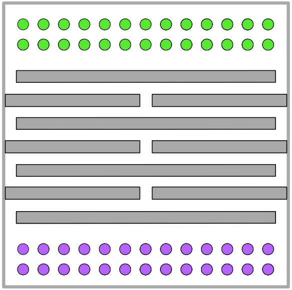


Experimental results

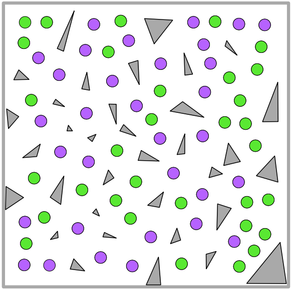
We implemented the algorithm in C++ with exact geometric methods that are provided by CGAL. As such, our implementation is complete, robust, deterministic and parameter free.



8 robots
19 sec.



26 robots
800 sec.

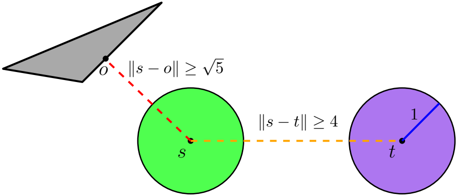


32 robots
6000 sec.

Motion Planning for Unlabeled Discs with Optimality Guarantees

Solovey, Yu, Zamir & Halperin

We study the unlabeled problem in the setting of unit-disc robots translating amid polygonal obstacles, with separation constraints:



We develop an $\tilde{O}(m^4 + m^2n^2)$-time algorithm that returns a solution whose cost is at most OPT+4m, in terms of the total path length.

Thank you!