On the hardness of of unlabeled motion planning

Kiril Solovey

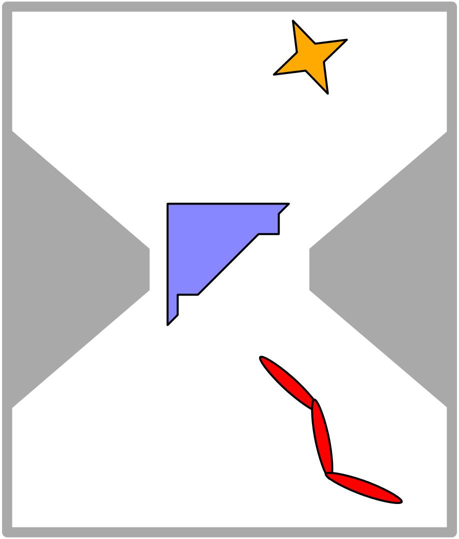
Tel Aviv University, Israel

January 20, 2015

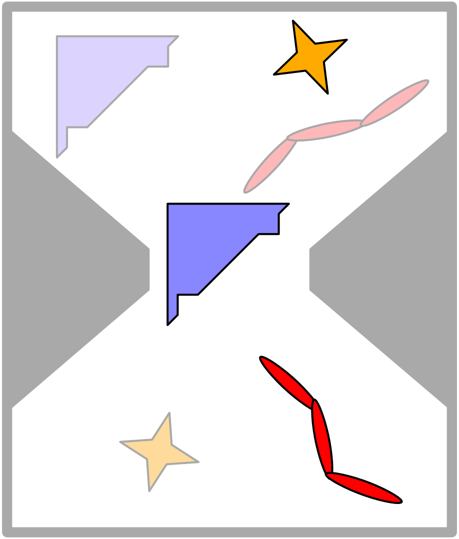
* Joint work with Dan Halperin

Introduction

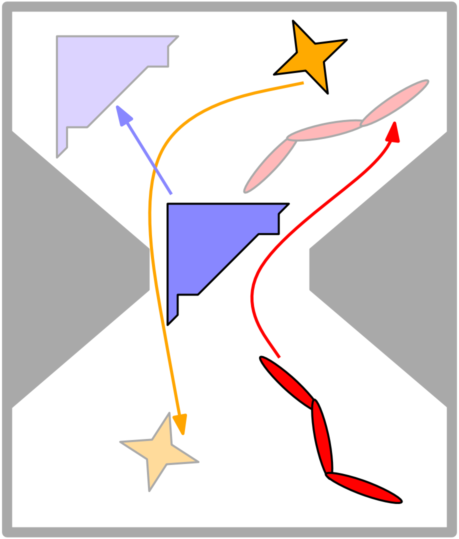
Multi-robot motion planning



Multi-robot motion planning



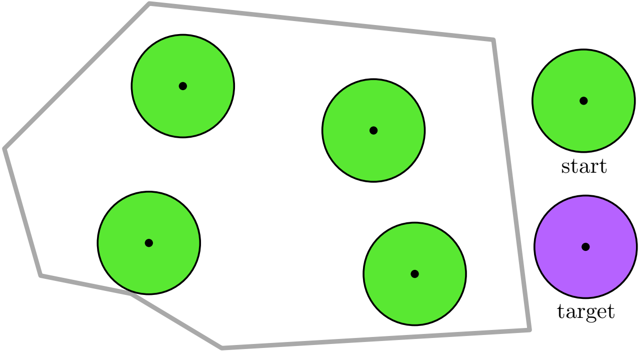
Multi-robot motion planning



Unlabeled multi-robot motion planning

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

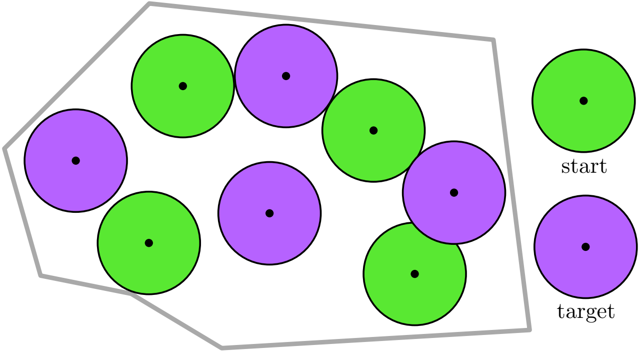
Goal: Move every robot to *some* target position.



Unlabeled multi-robot motion planning

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

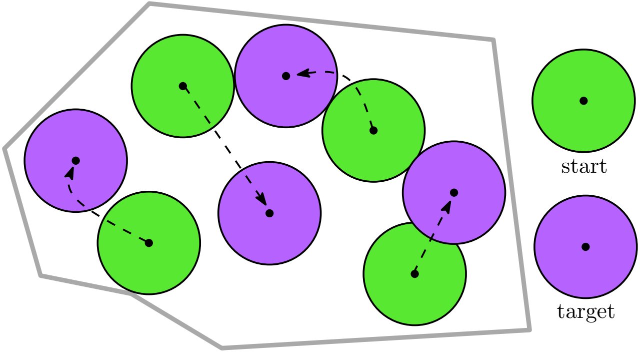
Goal: Move every robot to *some* target position.



Unlabeled multi-robot motion planning

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

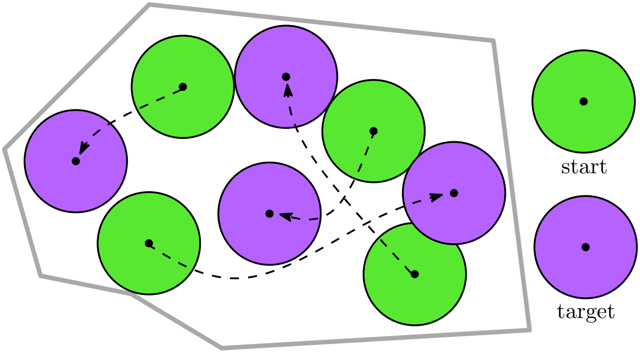
Goal: Move every robot to *some* target position.



Unlabeled multi-robot motion planning

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

Goal: Move every robot to *some* target position.



The complete combinatorial approach, related work

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

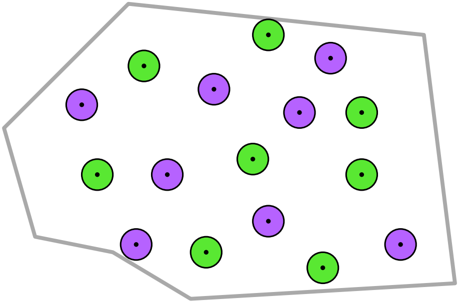
- Fixed number of robots:
  - Two discs in $O(n^3)$ [Schwartz and Sharir, 83]
  - Two discs in $O(n^2)$ [Yap, 83]
  - Two discs (and other types) in $O(n^2)$ [Sharir and Sifrony, 91]
- Multiple robots:
  - PSPACE-hardness for translating rectangles [Hopcroft et al., 84], [Hearn and Demaine, 05]
  - NP-hardness for discs [Spirakis and Yap, 84]
- Unlabeled motion planning:
  - $\tilde{O}(m^3)$ optimal algorithm for a special case [Turpin, Michael and Kumar, 13]

Unlabeled planning with separation

In WAFR 2014 we presented a complete combinatorial algorithm for motion planning of m unlabeled discs in a simple polygon.

The complexity of our algorithm is $O(n \log n + mn + m^2)$, where n is the complexity of the polygon, and m is the number of robots.

We make a simplifying assumption:



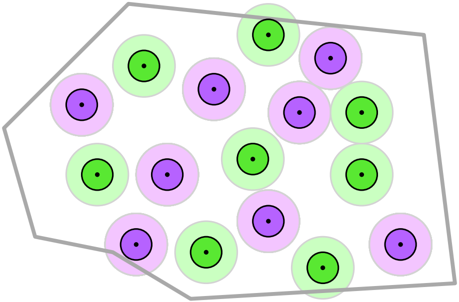
* Joint work with Aviv Adler, Mark de Berg, and Dan Halperin.

Unlabeled planning with separation

In WAFR 2014 we presented a complete combinatorial algorithm for motion planning of m unlabeled discs in a simple polygon.

The complexity of our algorithm is $O(n \log n + mn + m^2)$, where n is the complexity of the polygon, and m is the number of robots.

We make a simplifying assumption:



* Joint work with Aviv Adler, Mark de Berg, and Dan Halperin.

PSPACE-hardness of unlabeled planning

For the well-separated case the problem is in P. However, when no separation is assumed the problem becomes PSPACE-hard.

Contributor	Problem	Complexity	Robots
[Hopcroft et al., 84]	labeled	PSPACE-hard	rectangles
[Spirakis and Yap, 84]	labeled	NP-hard	discs
[Hearn and Demaine, 05]	labeled[1]	PSPACE-hard	2×1 rectangles
our work	unlabeled	PSPACE-hard	unit squares

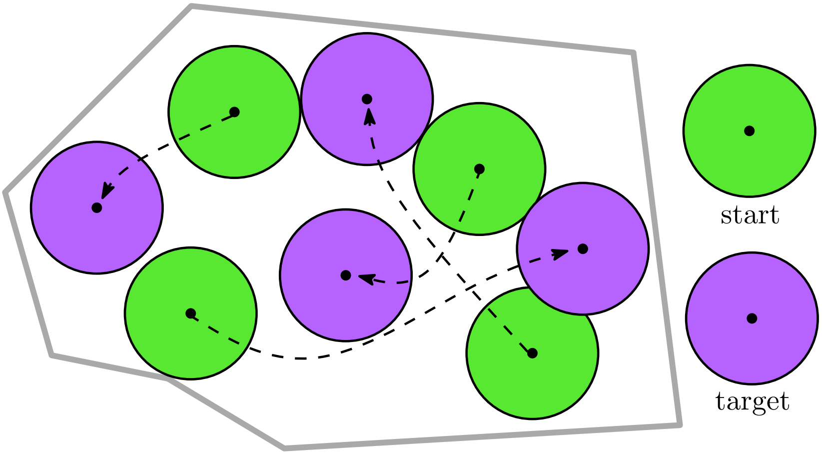
Table: Hardness results related to the multi-robot problem.

* Joint work with Dan Halperin.

[1]Hearn and Demaine consider various motion planning problems in their paper, including SOKOBAN and RUSH-HOUR.

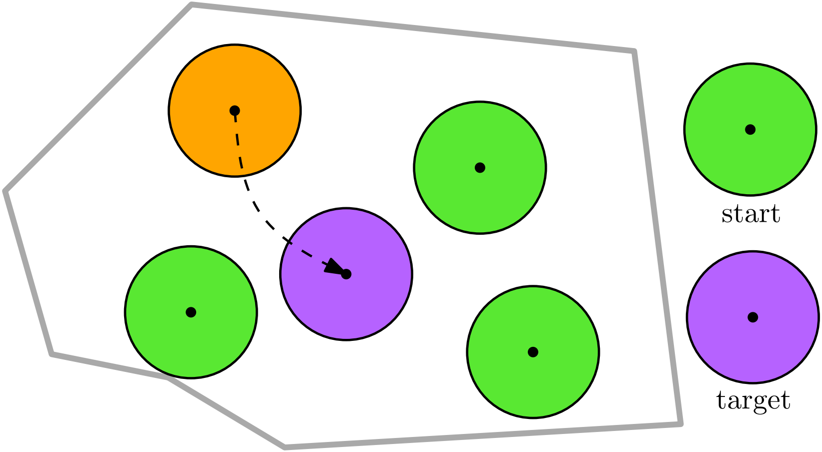
PSPACE-hard unlabeled problems

Move **any** robot to **any** target:



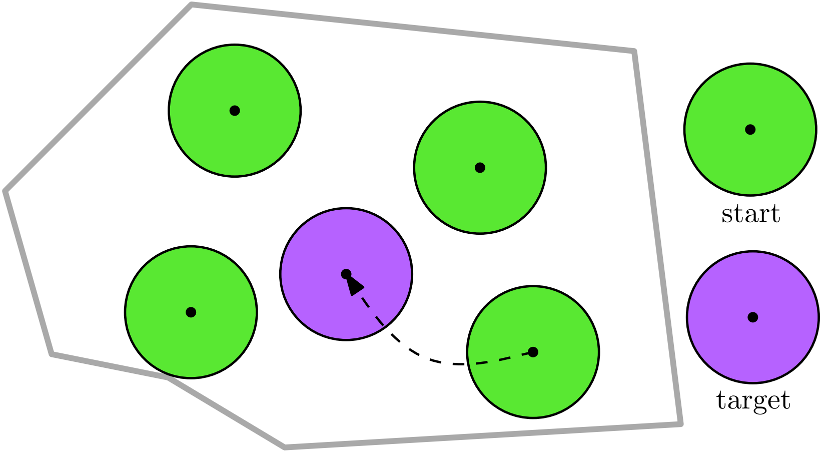
PSPACE-hard unlabeled problems

Move a **specific** robot to a **single** given target:



PSPACE-hard unlabeled problems

Move a **any** robot to a **single** given target:



Preliminaries

Basic definitions

Let r be a robot operating in a workspace $\mathcal{W} \subset \mathbb{R}^2$.

- $\mathcal{C}(r)$: configuration space of r
- $\mathcal{F}(r) \subset \mathcal{C}(r)$: free space of r
- For $s, t \in \mathcal{F}(r)$ a path is a cont. function $\pi : [0,1] \rightarrow \mathcal{F}(r)$, with $\pi(0) = s, \pi(1) = t$
- Given $c \in \mathcal{C}$ denote by $r(c) \subset \mathcal{C}$ the are occupied by r placed in c

Unlabeled motion planning

Two robots r, r' are geometrically identical if $\mathcal{F}(r) = \mathcal{F}(r')$ for $\mathcal{W}$.
Let $R = \{r_1, \ldots, r_m\}$ be a collection of m geometrically identical robots operating in $\mathcal{W}$. We use $\mathcal{F}$ to denote $\mathcal{F}(r_i)$ for any $r_i \in R$.

Definition 1

A collection of m configurations $C = \{c_1, \ldots, c_m | c_i \in \mathcal{C}\}$ is termed multi-configuration. It is also free if $C \subset \mathcal{F}$, and for every $c, c' \in C, c \neq c'$, it holds that $r(c) \cap r'(c') = \emptyset$

Unlabeled motion planning

Definition 2

Let $C = \{c_1, \ldots, c_m\}, C' = \{c'_1, \ldots, c'_m\}$ be two multi-confs. They are equivalent $(C \equiv C')$ if there exist m paths $\Pi = \{\pi_1, \ldots, \pi_m\}$ that move the robots from C to C', such that

- $\pi_i(0) = c_i$
- for every $c' \in C'$ exists j such that $\pi_j(1) = c'$
- for every $\tau \in [0, 1]$ the multi-conf $\Pi(\tau) = \{\pi_1(\tau), \ldots, \pi_m(\tau)\}$ is free.

Given two equivalent multi-confs C, C', denote by $\Pi(C, C')$ the set of m paths.

Unlabeled problems

We show that the following four decision problems are PSPACE-hard:

- *multi-to-multi*: given S, T, is true that $S \equiv T$?
- *multi-to-single*: given $S, t \in \mathcal{F}$, is there a multi-conf T such that $t \in T, S \equiv \mathcal{T}$?
- *multi-to-single-restricted*: given $S, s \in S, t \in \mathcal{F}$, is there a multi-conf T such that $t \in T, S \equiv T$, for which exists $\pi \in \Pi(S, T)$ such that $\pi(0) = s, \pi(1) = t$?
- *single-to-single*: given $s, t \in \mathcal{F}$, are there two multi-confs S, T such taht $s \in S, t \in T, S \equiv T$?

Nondeterministic constraint logic

NCL model

We show a reduction from the NCL model [Hearn and Demaine, 05]. NCL machine is defined by

- constraint graph $G = (V, E)$;
- weight function $w : E \to \mathbb{N}$;
- minimum-flow constraint $c : V \to \mathbb{N}$.

Definition 3

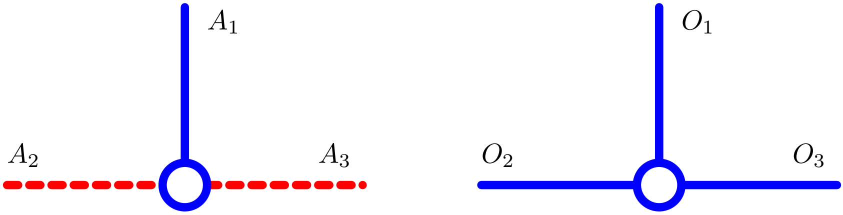
A machine configuration is an orientation function o over the edges of G such that the overall weight of incoming edges satisfy the minimum capacity requirement of every vertex.

Definition 4

A single *move* consists of reversing the orientation of one specific edge. Two orientations o, o' are equivalent ($o \equiv o'$), if o can be transformed to o' by a series of moves.

Example

Blue vertices and edges have a value of 2, while red have a value of 1.



NCL problems

The following decision problems are defined by [Hearn and Demaine, 05]:

- *orientation-to-orientation*: given two orientations o_S, o_T, is it true that $o_S \equiv o_T$?
- *orientation-to-edge*: given an orientation o_S and an edge $(v, v') \in E$ is there another orientation o_T such that $o_S \equiv o_T$ and $o_S(\{v, v'\}) \neq o_T(\{v, v'\})$?
- *edge-to-edge*: let $(v, v'), (u, u') \in E$. Additionally, let $o_{(v,v')}, o_{(u,u')}$ be two orientations for these specific edges. Are there two configurations o_S, o_T such that $o_S \equiv o_T$ and $o_S(v, v') = o_{(v,v')}, o_T(u, u') = o_{(u,u')}$?

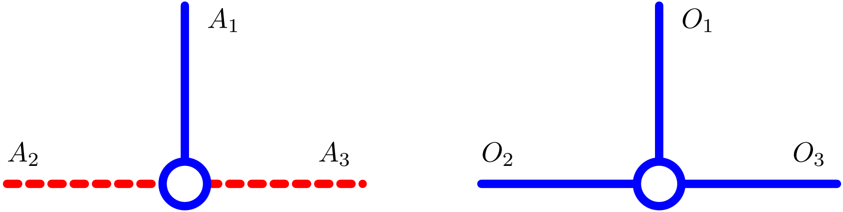
Harndess of NCL problems

The following Theorem will play a central role in our analysis.

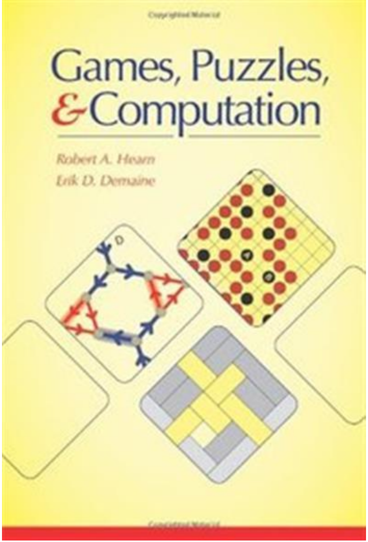
Theorem 5 (Hearn and Demaine)

orientation-to-orientation,orientation-to-edge, and edge-to-edge are PSPACE*-complete, even when the constraints graph is simple, planar, and consists of only* AND *and* OR *vertices.*

AND: OR:

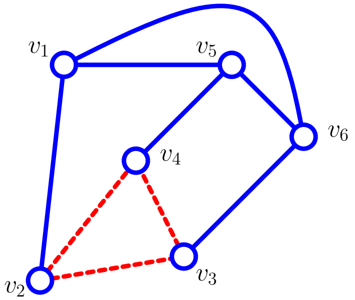


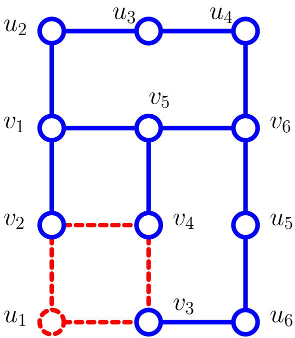
Examples for hardness proofs via NCL



Grid-embedded constraint graphs

As G is planar and of degree 3, it can be transformed, in polynomial time, into a grid graph G' whose vertices lie on points of a grid, and whose edges are axis-aligned segments of unit length.

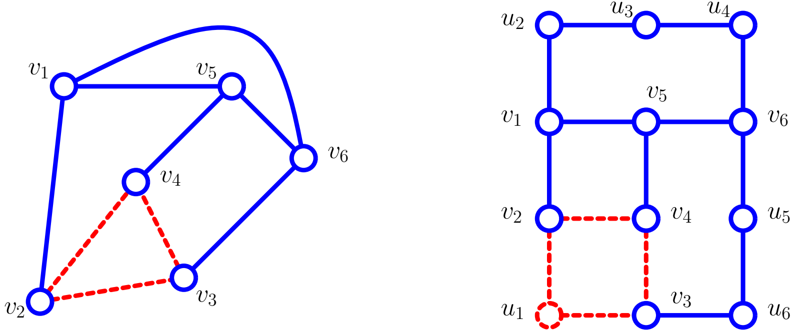




Note that this introduces a new type of CON vertices.

Grid-embedded constraint graphs

We assign capacities to CON vertices according to the weights of adjacent edges.



Lemma 6

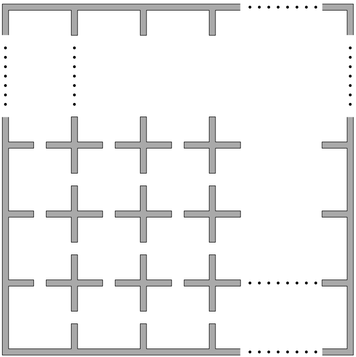
orientation-to-orientation,orientation-to-edge, and edge-to-edge, are PSPACE*-complete, even for the grid-embedded constraint graph G that consists of only* AND,OR, *and* CON *vertices.*

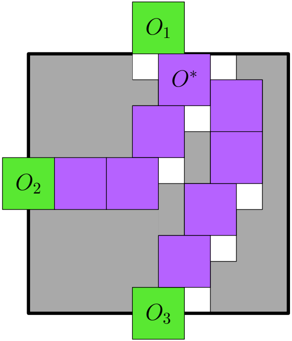
From NCL to multi-robot motion planning

Overview

Given a grid-embedded constraint graph we generate an unlabeled problem that consists of unit-square robots and polygonal obstacles.
We use the following grid layout that consists of

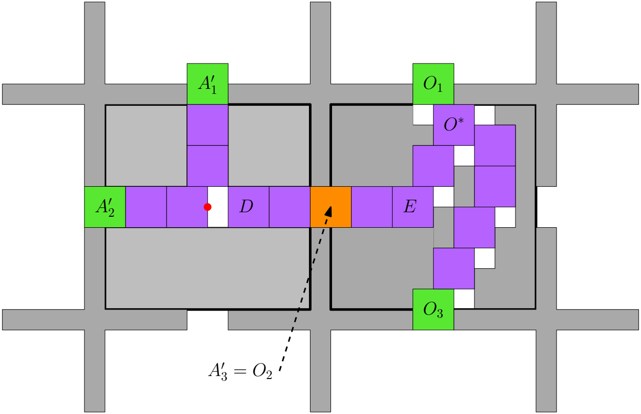
- 5×5 grid cells
- walls of thickness $1/2$ seperating the cells





Overview

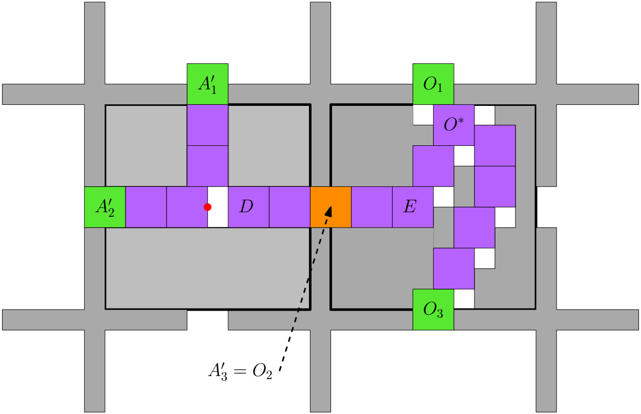
Each vertex from G is represented by the respective gadget. Vertices with shared edges are represented by adjacent cells which share a doorway.



Gadgets

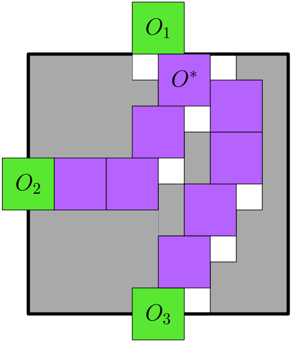
Each gadget contains two types of robots:

- robots representing edges (green or orange)
- robots representing vertices (purple)



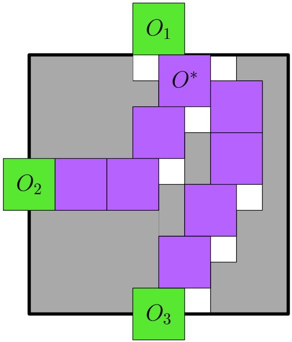
Terminal configurations

We will refer to configuration which are points on the $\frac{1}{2} \times \frac{1}{2}$ as *terminal configurations*. The robots can be located only in terminal confs or be in transit between two adjacent terminal configurations.

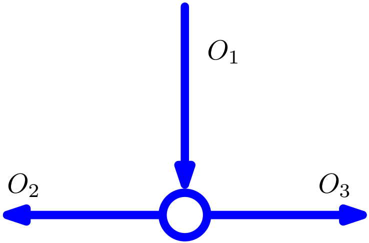


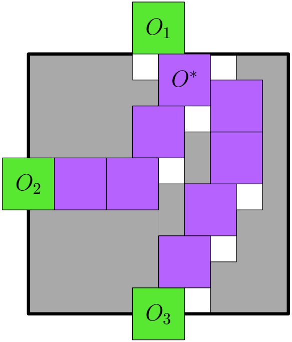
Inside/outside configurations

An edge robot is considered to be *inside* a gadget if it's a half-step inside the gadget. It is *outside* if it touches the 5×5 square of the gadget from the outside.



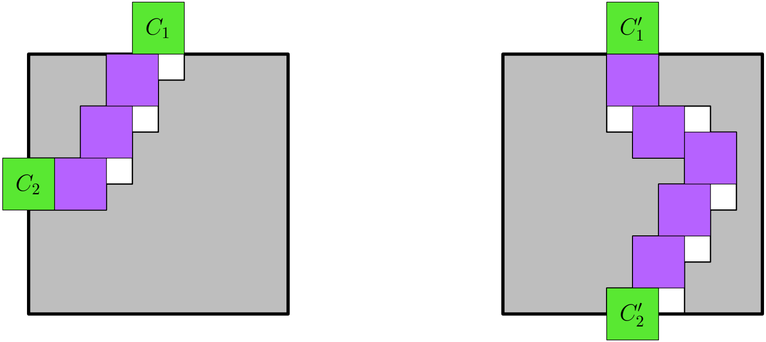
In-directed edge is represented by an outside configuration, while *out*-directed edge is represented by an inside configuration.





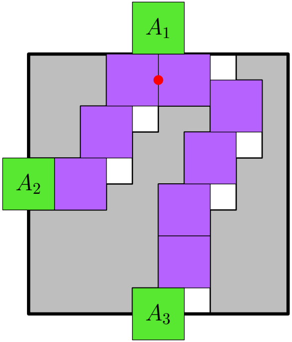
CON gadgets

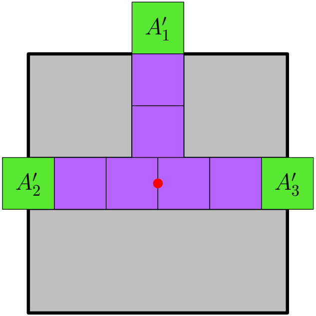
The two edge robots cannot be simultaneously inside the gadget.



AND gadgets

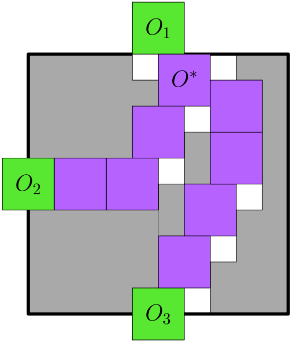
A_1, A'_1 can be inside only if the other edge robots are outside.





OR gadgets

At least one edge robot must be outside the gadget.



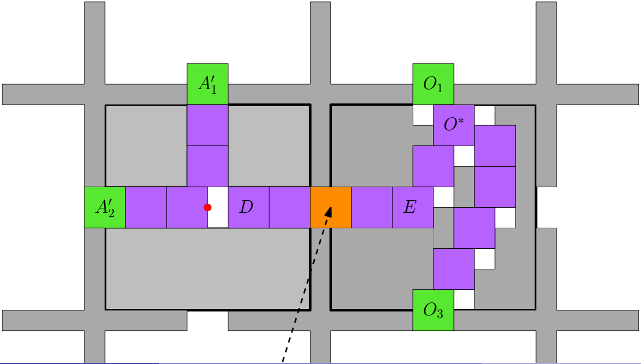
Basic properties

Lemma 7

Each edge robot can in at most two distinct terminal configurations.

Proof.

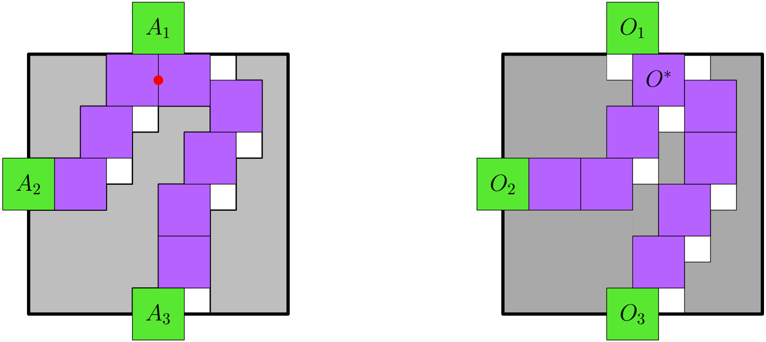
For every possible combination of two gadgets, the corresponding edge robot remains stuck. □



Basic properties

Lemma 8

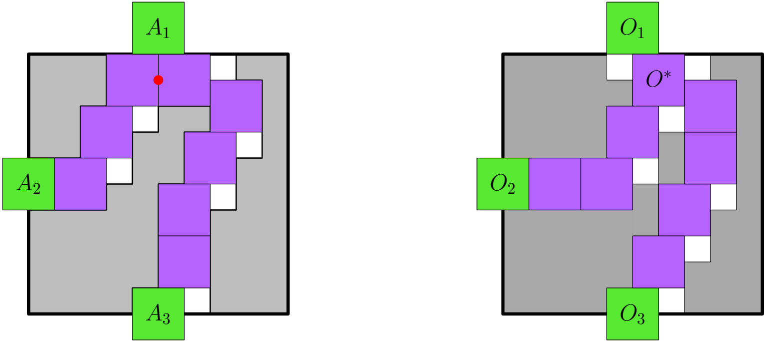
Each vertex robot can be in at most two distinct terminal configurations, save robot O^* in the OR *gadget, which can be in at most three terminal configurations.*



Basic properties

Lemma 9

Given a specific terminal configuration, there is at most one robot that can be in it.



Main Theorem

Theorem 10

multi-to-multi, multi-to-single, multi-to-single-restricted, and single-to-single are PSPACE*-hard.*

Reductions:

- *orientation-to-orientation* $\Longrightarrow$ *multi-to-multi*
- *orientation-to-edge* $\Longrightarrow$ *multi-to-single*
- *orientation-to-edge* $\Longrightarrow$ *multi-to-single-restricted*
- *edge-to-edge* $\Longrightarrow$ *single-to-single*

First reduction

In *orientation-to-orientation* we get two orientations o_S, o_T and ask whether $o_S \equiv o_T$. In *multi-to-multi* we get two multi-confs S, T and as whether $S \equiv T$.

The two orientations o_S, o_T are transformed into two multi-confs S, T, such that

$$o_S \equiv o_T \iff S \equiv T.$$

Orientation to multi-configuration

Lemma 11

An orientation o induces a valid multi-configuration C (and vice versa).

Proof.

- Given an edge in E, its orientation induces a terminal configuration for the respective edge robot
- Positions of edge robots induce positions for vertex robots
- By definition of o, C must be valid
- For the other direction, note that every edge robot can be in at most two terminal configurations, which represent the two direction of an edge in G

□

$o_S \equiv o_T \implies S \equiv T$

If $o_S \equiv o_T$ then there exists a sequence of edge flips that lead from o_S to o_T.

- Let o, o' be two orientation such that o' results from o by a single edge flip
- Denote by C, C' the multi-confs induced by the orientations
- Denote by c, c' the two configurations between which the edge robot has to move
- If the move between c to c' is not possible then one of the multi-confs must be invalid

$S \equiv T \implies o_S \equiv o_T$

Suppose that $S \equiv T$. We need to show that there exists a special $\Pi(S, T)$:

- If an edge robot is in transit between two terminal configurations
- Then the rest of edge robots are stationary in terminal configurations

We prove this for every individual gadget

$S \equiv T \implies o_S \equiv o_T$

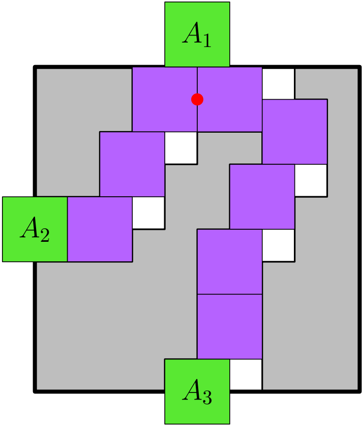
Suppose that $S \equiv T$. We need to show that there exists a special $\Pi(S, T)$:

- If an edge robot is in transit between two terminal configurations
- Then the rest of edge robots are stationary in terminal configurations

We prove this for every individual gadget.

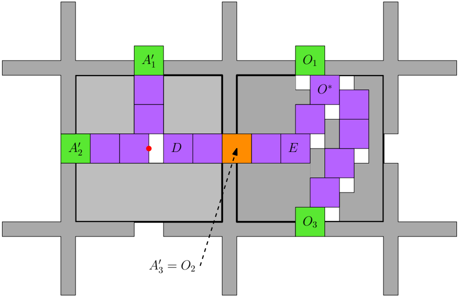
$S \equiv T \implies o_S \equiv o_T$

- If both A_2, A_3 move inside then A_1 must be outside
- Thus, A_2, A_3 do not depend on each other and can make the move sequentially



$S \equiv T \implies o_S \equiv o_T$

- Suppose that both A_2, A_3 move outside
- Then the two adjacent gadgets already made the adjustments to let A_2, A_3 enter
- Then A_2, A_3 do not depend on each other



Second + Third reductions

- In *orientation-to-edge* we get an orientations o_S and an edge $e \in E$ and ask whether there exists $o_T \equiv o_S$ such that $o_T(e) \neq o_S(e)$.
- In *multi-to-single* we get a multi-conf S and a configuration t and ask whether there exists $T \equiv S$ such that $t \in T$.
- In *multi-to-single-restricted* we require that a specific robot will reach t.

PSPACE-hardness of the labeled case

Theorem 12

The labeled variants of multi-to-multi and multi-to-single-restricted are PSPACE*-hard.*

We showed that given a specific terminal configuration, there is at most one robot that can be in it.

Open problems in combinatorial planning

- The case of two discs:
  - Subquadratic algorithm
  - Optimal motion (might be hard)
- Multiple robots without separation:
  - Hardness of **unlabeled** planning for discs
  - Hardness of **unlabeled** planning in a **simple polygon**
- Multiple robots with separation:
  - Efficient planning for **labeled** discs with separation 4
  - Optimal planning for **unlabeled** discs with separation 4 [work in progress, with Jingjin Yu and Or Zamir]

Question

Is it possible to solve the problem with separation $2 + \varepsilon$ in time polynomial in m, n, and $1/\varepsilon$?