IIS-1734419: Final Year Report

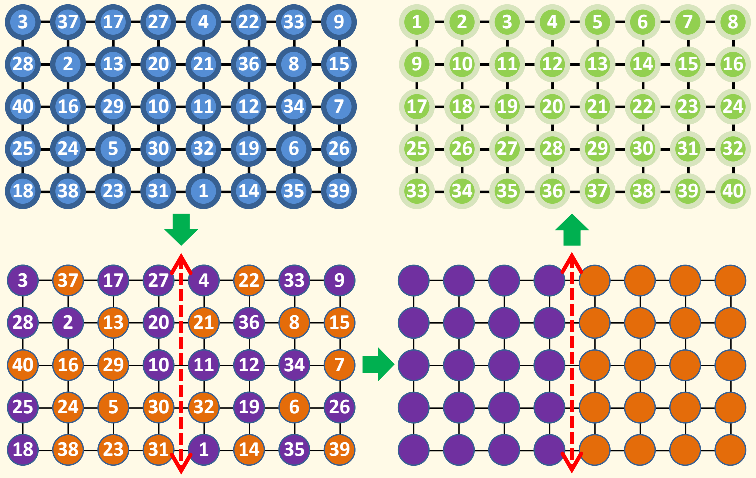
Optimal Multi-Robot Motion Planning: Theoretical Limits and Fast Algorithms

Jingjin Yu

Department of Computer Science | Rutgers University

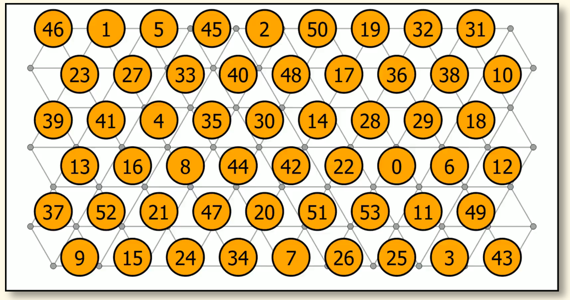
Highlights of Contributions

Theoretical Breakthroughs on Optimal Multi-Robot Motion Planning

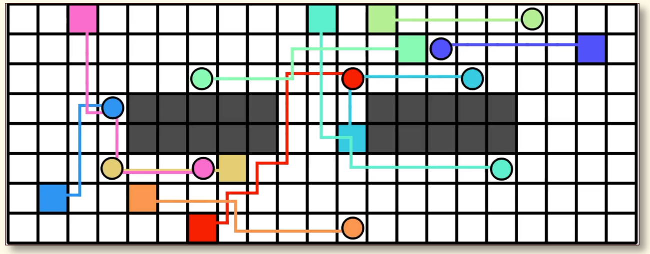


Polynomial time algorithm for constant factor multi-robot motion planning on grids under maximum density [RSS'18]

Effective and Principled Algorithms

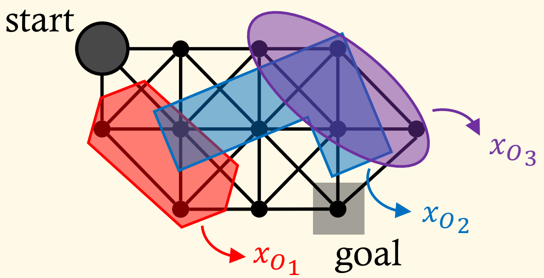


Fast algorithms for continuous, high density settings [WAFR'18]



Fast algorithms for dynamic settings [RA-L/ICRA'20]

General Methodology for Path-Based (Multi-Robot) Optimization



$$\sum_{v\in N(v^I)} x_{v^I,v} = \sum_{v\in N(v^G)} x_{v,v^G} = 1, \quad \sum_{v\in N(v^I)} x_{v,v^I} = \sum_{v\in N(v^G)} x_{v^G,v} = 0$$

$$\forall v_i \in V/\{v^I, v^G\}, \quad \sum_{v_j\in N(v_i)} x_{v_i,v_j} = \sum_{v_j\in N(v_i)} x_{v_j,v_i} \le 1$$

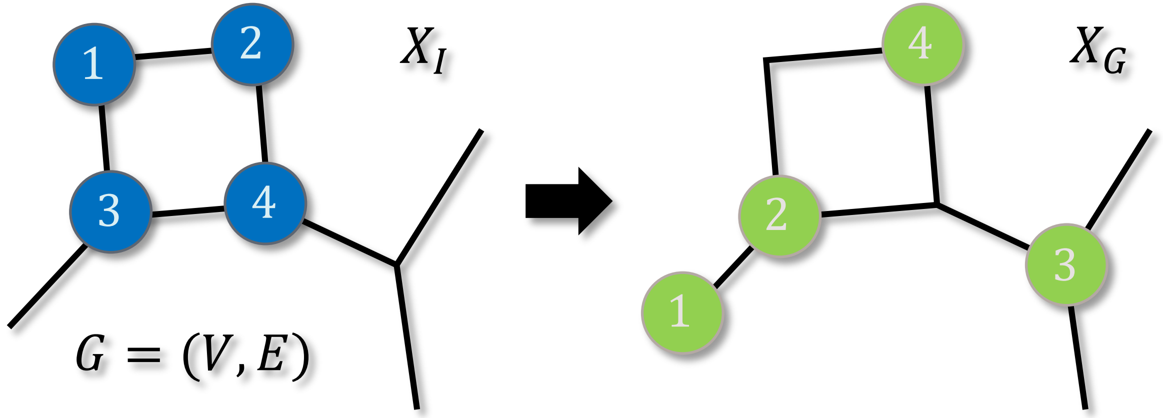
$$\forall O_k \in O, L x_{O_k} \ge \sum_{v_i\in O_k} \sum_{v_j\in N(v_i)} x_{v_i,v_j} + x_{v_j,v_i}$$

Minimize $\sum_{O_k\in O} x_{O_k}$

[IROS'19, Best Student Paper Finalist, Best Application Paper Finalist]

Optimal Multi-Robot Motion Planning on Graphs

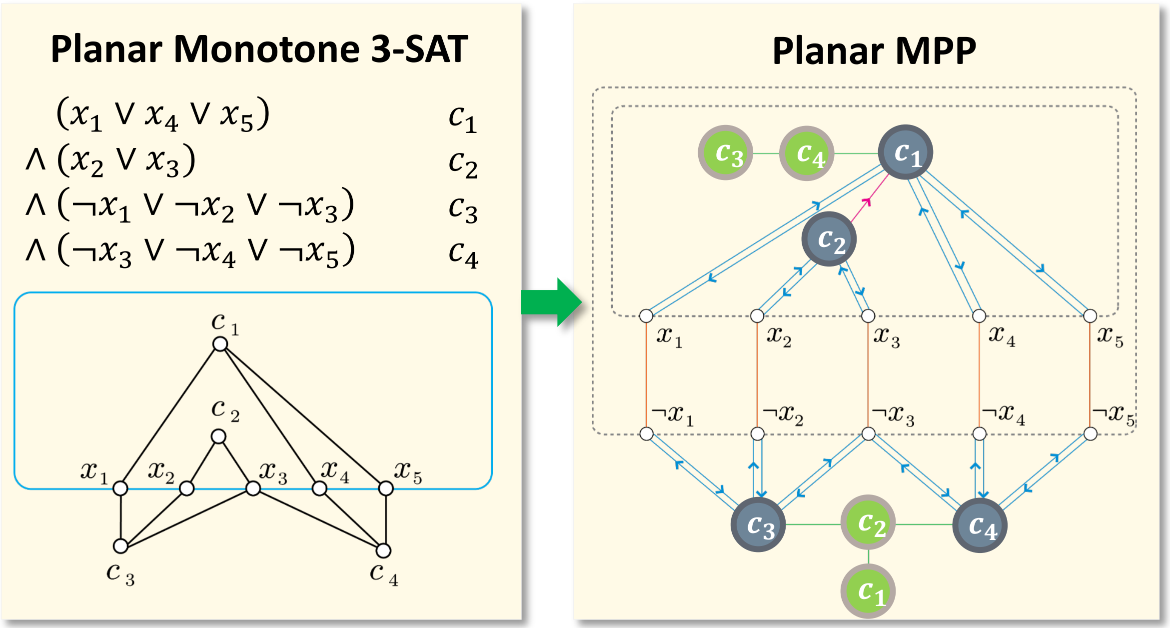
Optimal Multi-Robot Path and Motion Planning



MPP: (G, X_I, X_G), seek collision free $P = \{p_1, \ldots, p_n\}$

- ⇨ Min max time (makespan): $\min_{P \in \mathcal{P}} \max_{p_i \in P} time(p_i)$
- ⇨ Min total time: $\min_{P \in \mathcal{P}} \sum_{p_i \in P} time(p_i)$
- ⇨ ……
- ⇨ We seek (near-) optimal solutions (ideally, 1.x in practice)

Computational Intractability

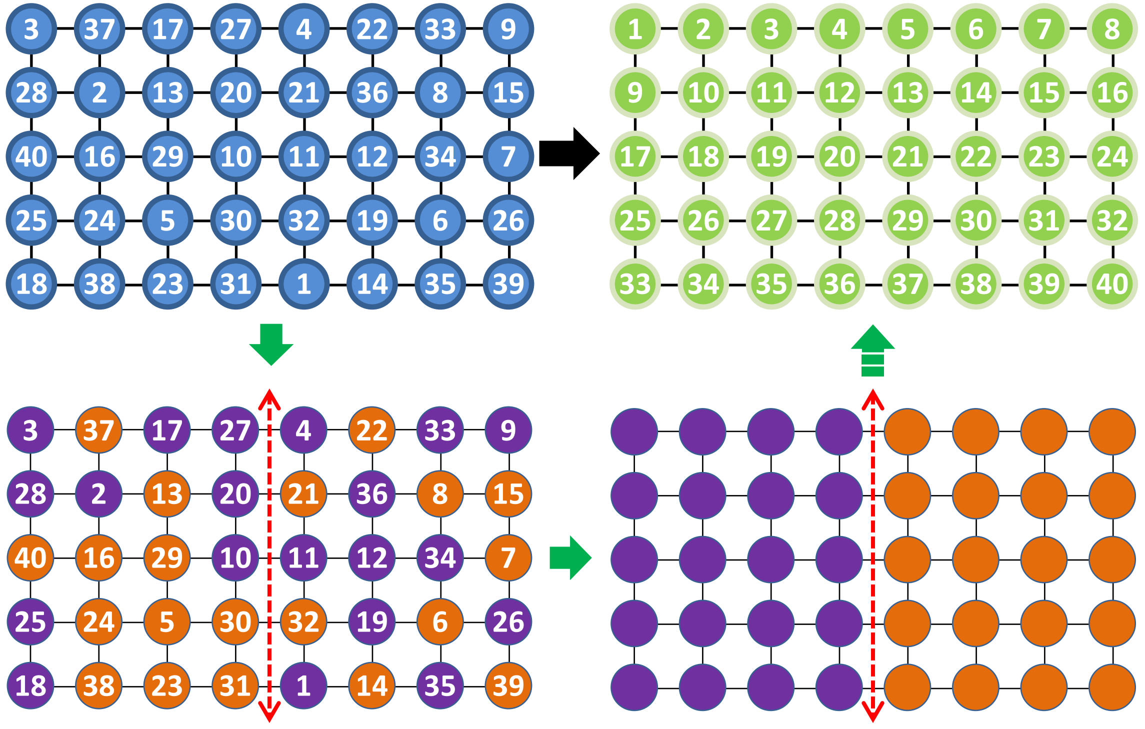


Theorem. Optimal Planar MPP is NP-hard for min makespan, min total time, min max distance, and min total distance objectives [RA-L' 16].

Polynomial Time $O(1)$-Optimal Algorithm (1)

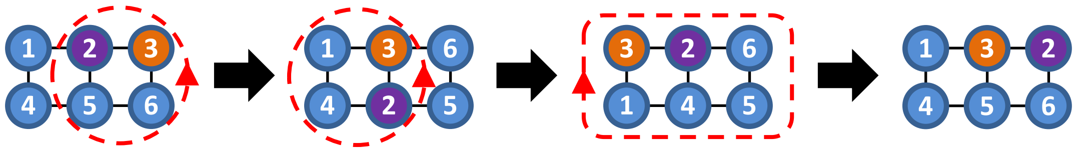
MRS'17

Divide-n-Conquer Approach for Exp. O(1)-Optimal Solution

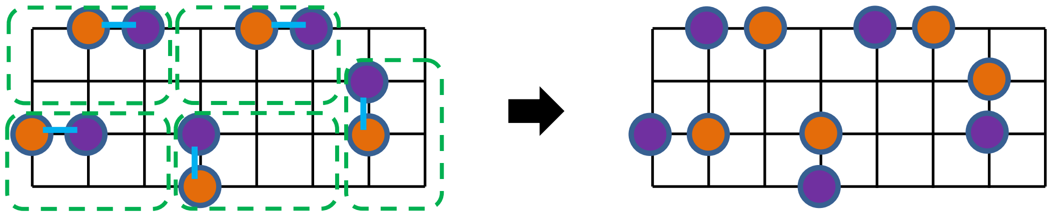


Treating a labeled problem as an unlabeled problem helps!

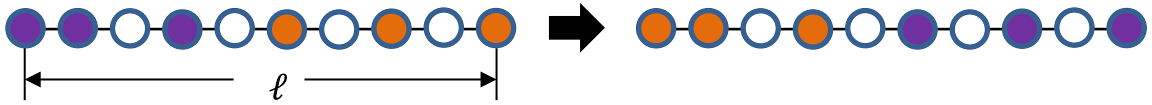
Motion Primitives



$O(1)$-step "edge flip" within a 2×3 grid



Multiple concurrent edge flips in **parallel** in $O(1)$ steps

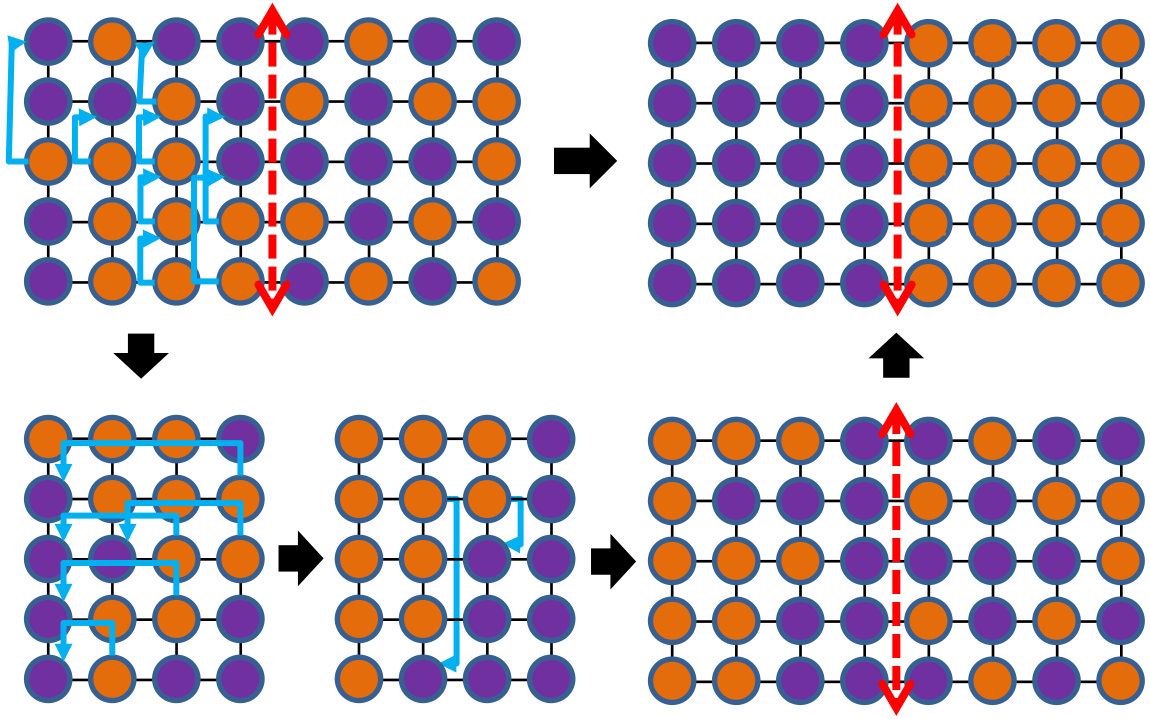


Exchange two groups on an **embedded line** in $O(\ell)$ steps

Polynomial Time $O(1)$-Optimal Algorithm (2)

MRS'17, RSS'18

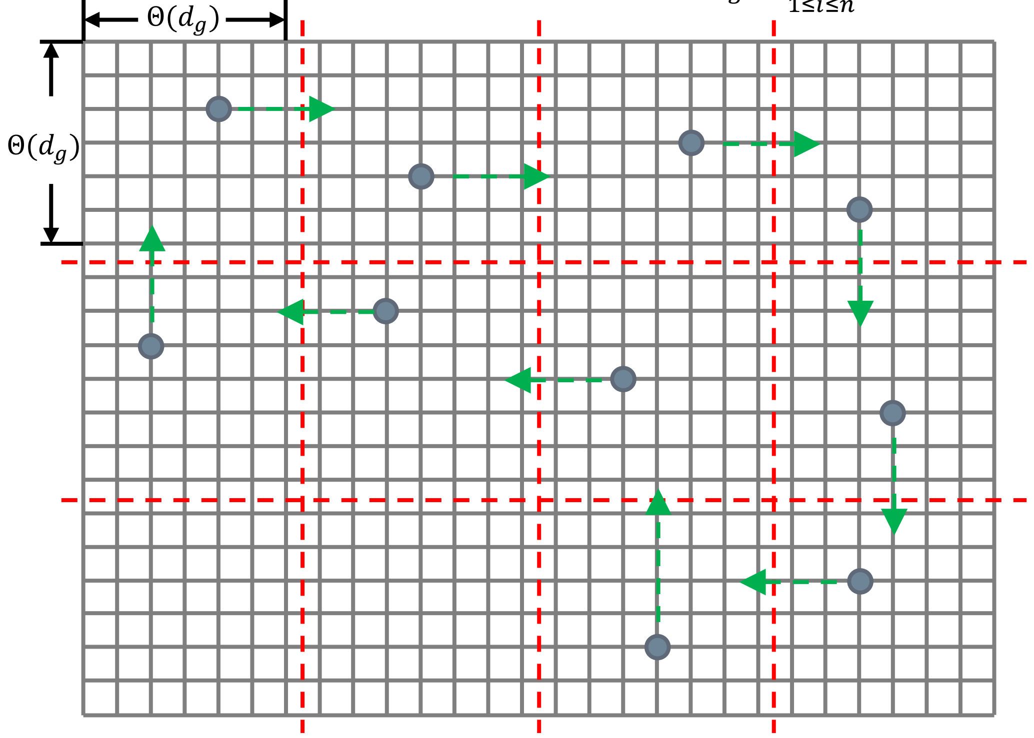
Split & Group via Multiple "Shuffles"



Split & Group (SaG) computes **expected** $O(1)$-optimal sol.

Global Partition & Flow

$$d_g = \max_{1 \le i \le n} dist(X_I(i), X_G(i))$$

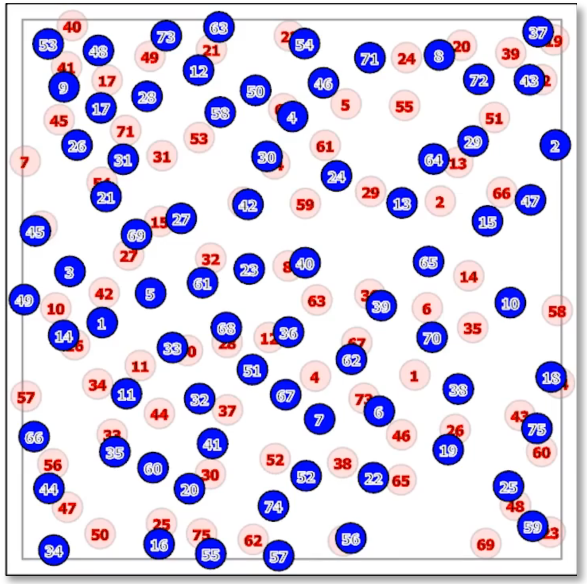


Applying global partition & flow yields $O(1)$-optimal solution

Fast Algorithms for Dense Continuous Problems

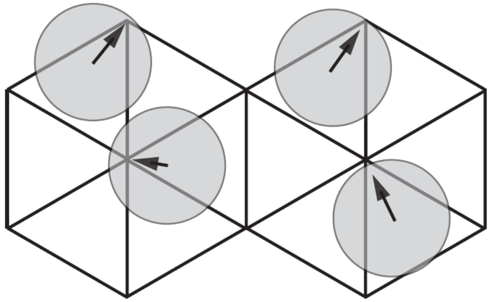
WAFR'18

Prior Work on Continuous Multi-Robot Motion Planning

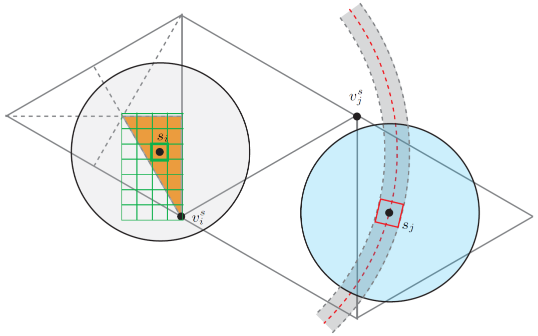


23% maximum density with provable optimality guarantee [ISRR'15]

New WAFR Work: Maximum Density of 51% with Completeness Guarantee

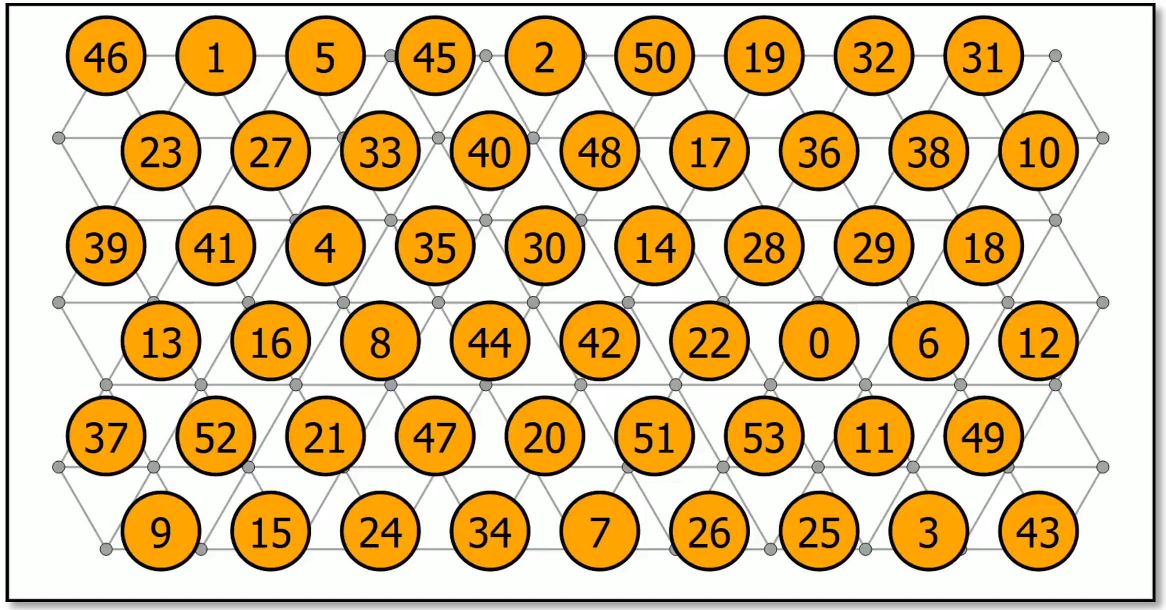


Novel discretization method



Computer assisted proof

- Opt. guarantee on discrete problem on triangular grid
- Completeness guarantee on continuous problem
- Fast computation for continuous problem
- All with highest supported density in the literature



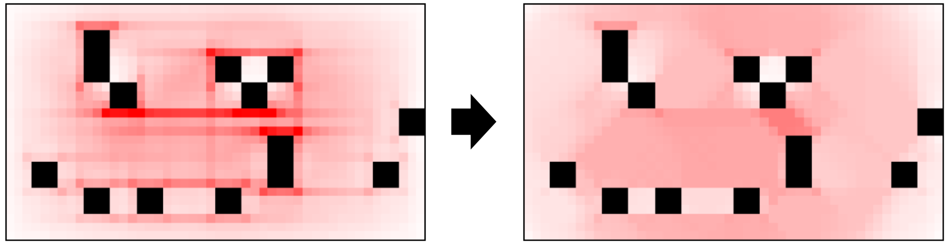
IIS-1734419 NRI: FND: Collaborative Multi-Robot Systems with Provable Availability, Safety, and Optimality Guarantees

JJ Yu | RUTGERS

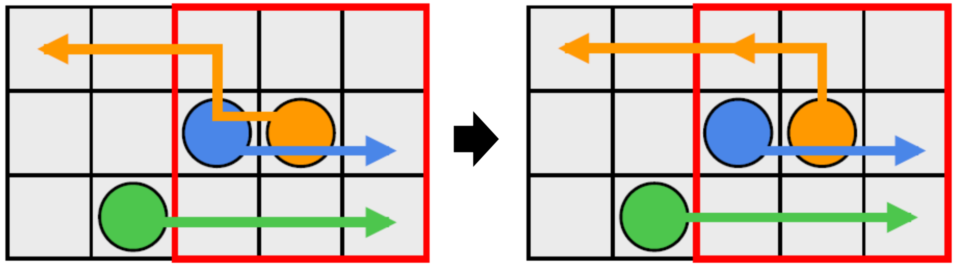
Fast Algorithms for Dynamic Re-Planning

RA-L/ICRA'20

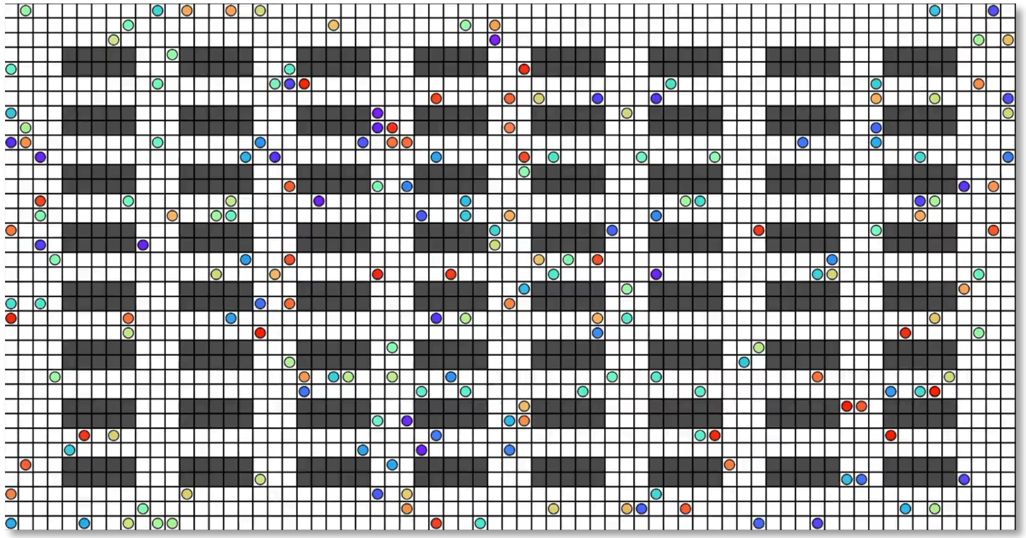
DDM: Multi-Robot Motion Planning w/ Path Diversification and Solution Database Lookup



Apply path diversification to reduce congestion

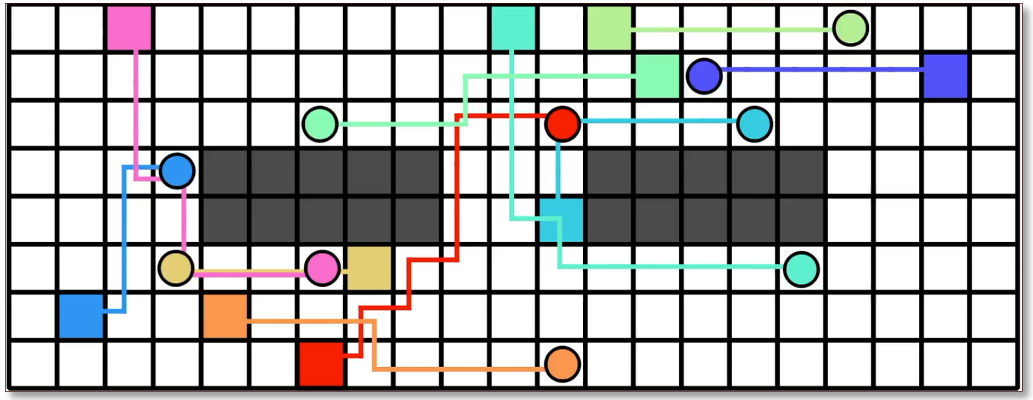


Using solution database to speed up local planning



Static settings:

- Fast
  Ex: 69 x 36 grid
  200 robots
  Time: 0.1 s
- High quality
  1.x optimal



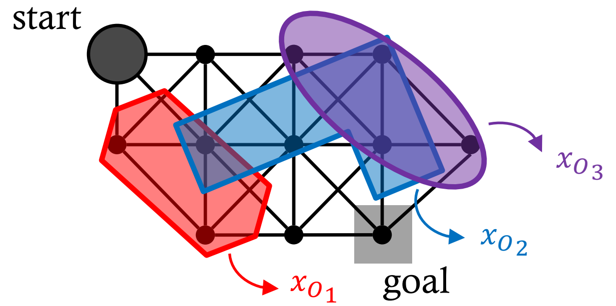
Dynamic settings:

- Fast replanning
- High throughput
- High quality

Methodology for Path-Based Multi-Robot Optimization

IROS'19, Best Application Paper Finalist, Best Student Paper Finalist

General IP Optimization Framework



$$\sum_{v\in N(v^I)} x_{v^I,v} = \sum_{v\in N(v^G)} x_{v,v^G} = 1$$

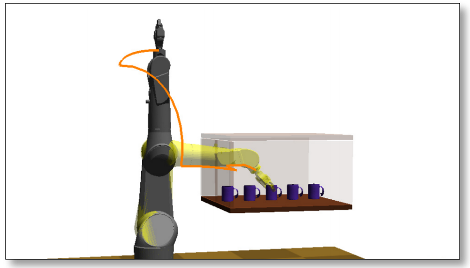
$$\sum_{v\in N(v^I)} x_{v,v^I} = \sum_{v\in N(v^G)} x_{v^G,v} = 0$$

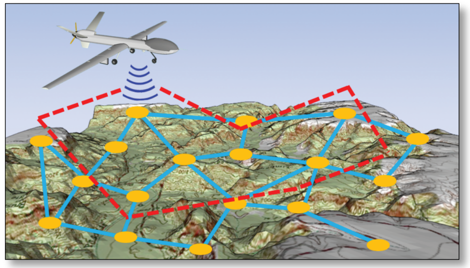
$$\forall v_i \in V/\{v^I, v^G\}, \sum_{v_j\in N(v_i)} x_{v_i,v_j} = \sum_{v_j\in N(v_i)} x_{v_j,v_i} \le 1$$

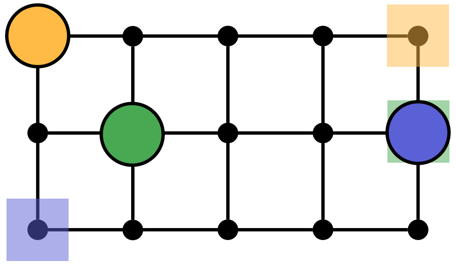
$$\forall O_k \in \mathcal{O}, L x_{O_k} \ge \sum_{v_i\in O_k} \sum_{v_j\in N(v_i)} x_{v_i,v_j} + x_{v_j,v_i}$$

Minimize $\sum_{O_k\in\mathcal{O}} x_{O_k}$

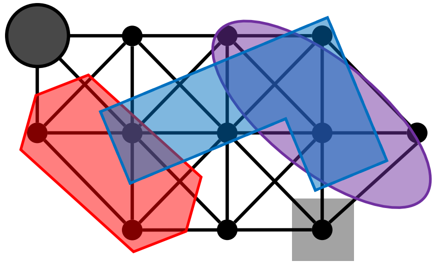
Applications to Many Path-Based Optimization Problems in Robotics



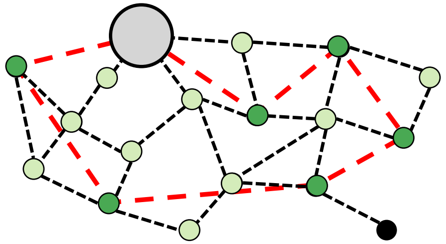




Multi-robot motion planning



Minimum constraint removal



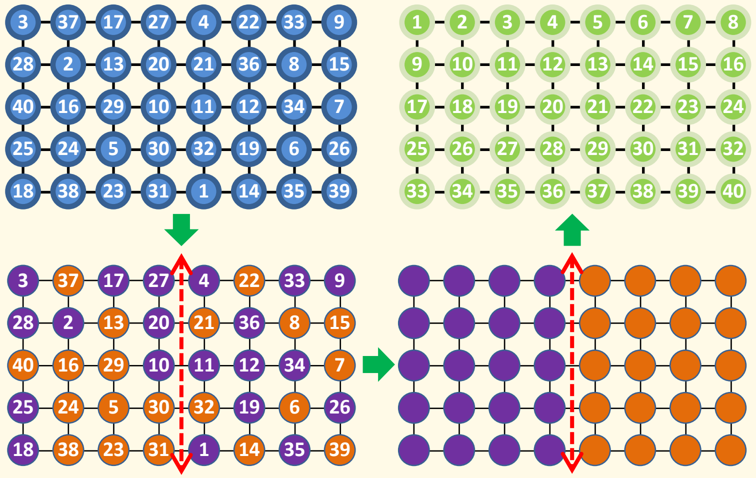
Reward collection problems

▶ A good first option for your path-based optimization problems – give it a try!

IIS-1734419 NRI: FND: Collaborative Multi-Robot Systems with Provable Availability, Safety, and Optimality Guarantees

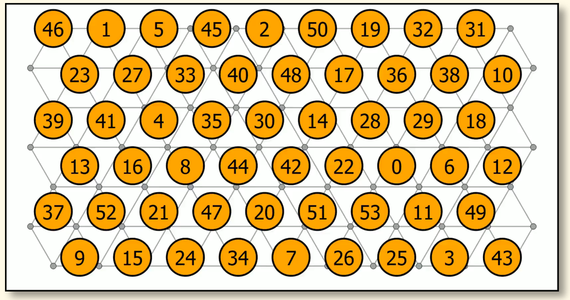
Highlights of Contributions

Theoretical Breakthroughs on Optimal Multi-Robot Motion Planning

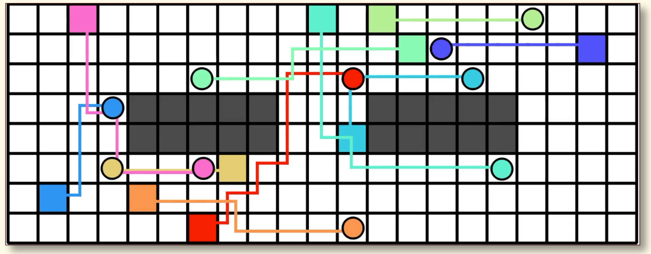


Polynomial time algorithm for constant factor multi-robot motion planning on grids under maximum density [RSS'18]

Effective Algorithmic Solutions

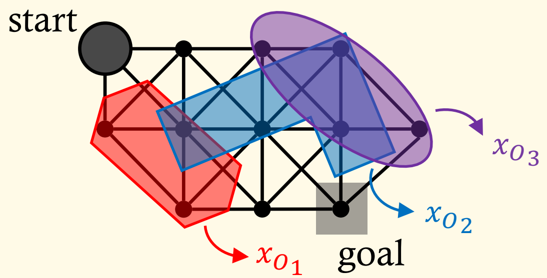


Fast algorithms for continuous, high density settings [WAFR'18]



Fast algorithms for dynamic settings [RA-L/ICRA'20]

General Methodology for Path-Based (Multi-Robot) Optimization



$$\sum_{v\in N(v^I)} x_{v^I,v} = \sum_{v\in N(v^G)} x_{v,v^G} = 1, \quad \sum_{v\in N(v^I)} x_{v,v^I} = \sum_{v\in N(v^G)} x_{v^G,v} = 0$$

$$\forall v_i \in V/\{v^I, v^G\}, \quad \sum_{v_j\in N(v_i)} x_{v_i,v_j} = \sum_{v_j\in N(v_i)} x_{v_j,v_i} \le 1$$

$$\forall O_k \in \mathcal{O}, L x_{O_k} \ge \sum_{v_i\in O_k} \sum_{v_j\in N(v_i)} x_{v_i,v_j} + x_{v_j,v_i}$$

Minimize $\sum_{O_k\in\mathcal{O}} x_{O_k}$

[IROS'19, Best Student Paper Finalist, Best Application Paper Finalist]