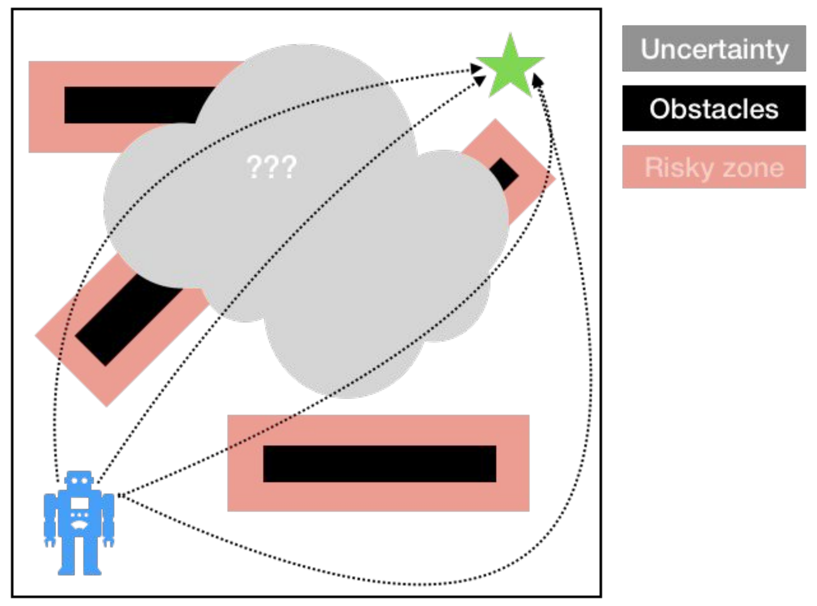
ICRA 2022 IEEE International Conference on Robotics and Automation

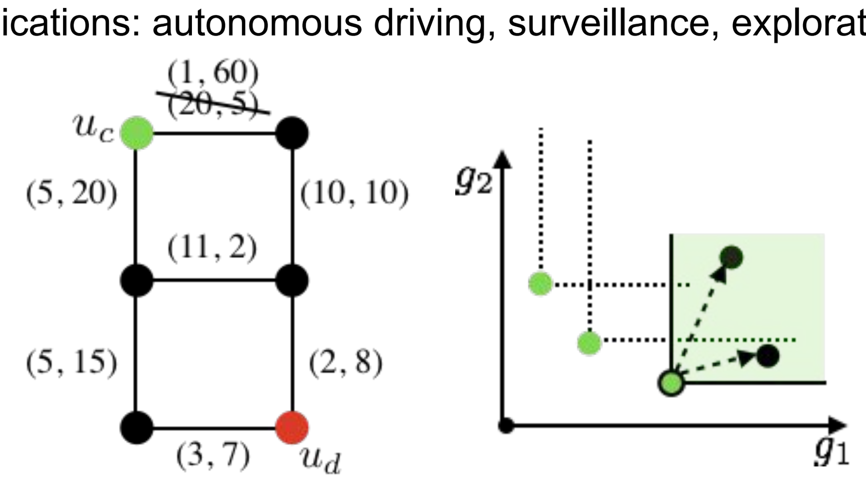
Multi-Objective Path-Based D* Lite

Zhongqiang (Richard) Ren, Sivakumar Rathinam, Maxim Likhachev, Howie Choset

1. Motivation and Problem



- Planning path in a dynamic environment.
- Optimizing multiple objectives: path length, risk, arrival time.
- Applications: autonomous driving, surveillance, exploration, etc.

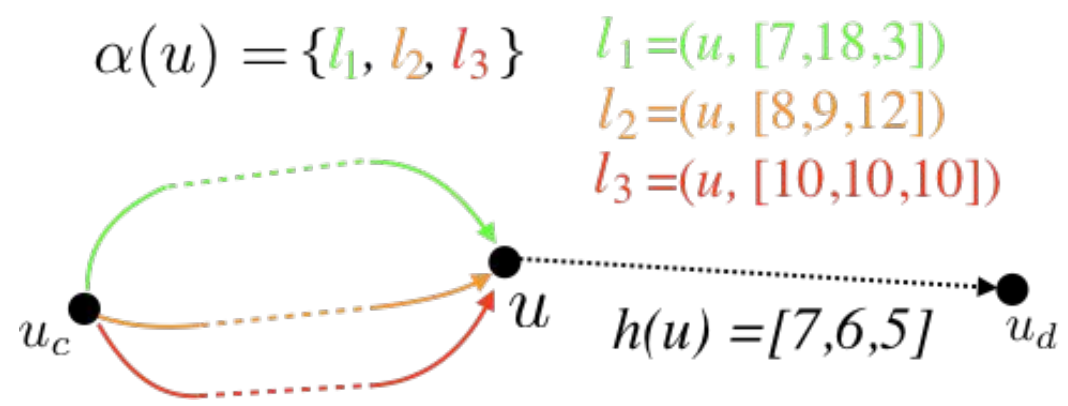


- Graph search problem
- Vector-cost edges.
- Dominance and Pareto-optimal

Def. Dominance
Given two M-dim vector a, b:
If $a(i) \leq b(i),\ \forall i \in \{1, 2, \ldots, M\}$
And $a(i) < b(i),\ \exists i \in \{1, 2, \ldots, M\}$
Then a dominates b.

- Goal: find all cost-unique Pareto-optimal paths from the start to the goal.

2. Background

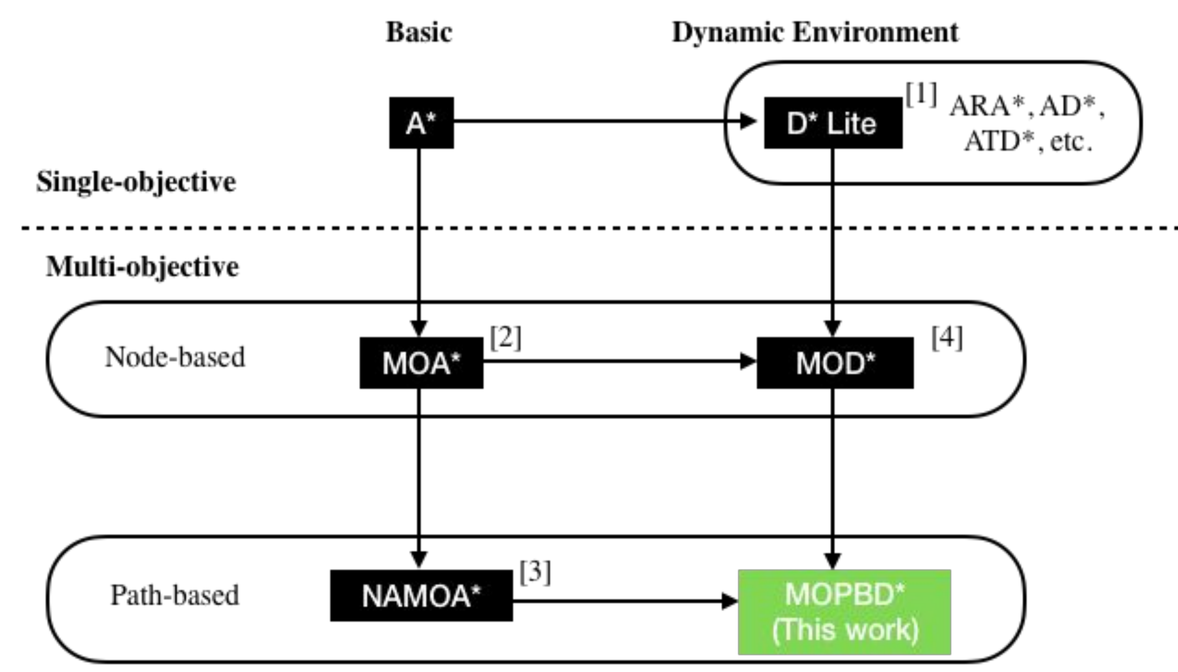


Node-based Expansion (MOA*, MOD*)

- In each iteration, a node is selected from OPEN and expanded.
- To expand a node, all labels (i.e. partial solution paths) are extended to adjacent nodes.

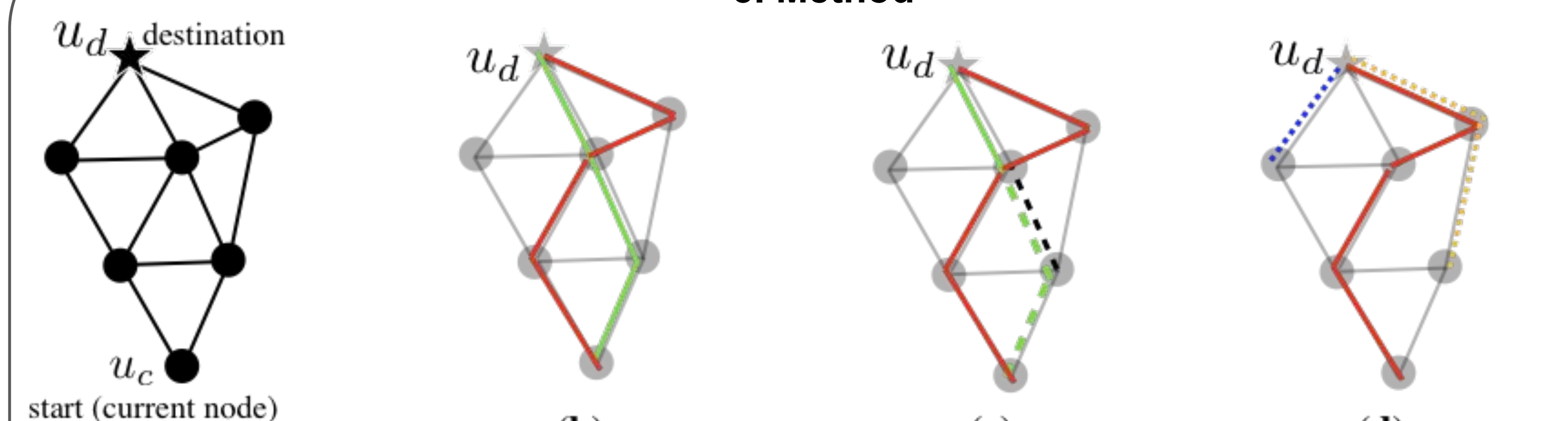
Path-based Expansion (NAMOA* and its variants)

- Labels (i.e. partial solution paths) are stored in OPEN and selected for expansion.
- When a new label is generated at a node, this label (rather than node) is inserted into OPEN.



[1] Koenig, Sven, and Maxim Likhachev. "Fast replanning for navigation in unknown terrain." IEEE Transactions on Robotics 21, no. 3 (2005): 354-363.
[2] Stewart, Bradley S., and Chelsea C. White III. "Multiobjective a." Journal of the ACM (JACM) 38, no. 4 (1991): 775-814.
[3] Mandow, Lawrence, and José Luis Pérez De La Cruz. "Multiobjective A* search with consistent heuristics." Journal of the ACM (JACM) 57, no. 5 (2008): 1-25.
[4] Oral, Tugcem, and Faruk Polat. "MOD* Lite: an incremental path planning algorithm taking care of multiple objectives." IEEE Transactions on Cybernetics 46, no. 1 (2015): 245-257.

3. Method

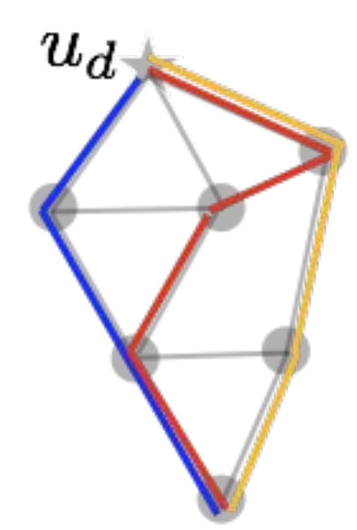


(a)

(b) Initially, search in as same way as NAMOA* (path-based) backwards from the goal to the start.

(c) When *cost(e)* changes (either increases or decreases), recursively find and delete all partial paths that go through *e*.

(d) Using a new notion of *consistency* to find all partial paths that need re-expansion.



(e)

Continue the search until all cost-unique Pareto-optimal paths are find.

Non-dominated paths		Path cost vectors
1st search, (Plot (b))	green	(10, 3.3, 331)
	red	(15, 2.1, 575)
2nd search, (Plot (e))	red	(15, 2.1, 575)
	yellow	(12, 3.6, 461)
	blue	(13, 3.4, 501)

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 2120219 and 2120529. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

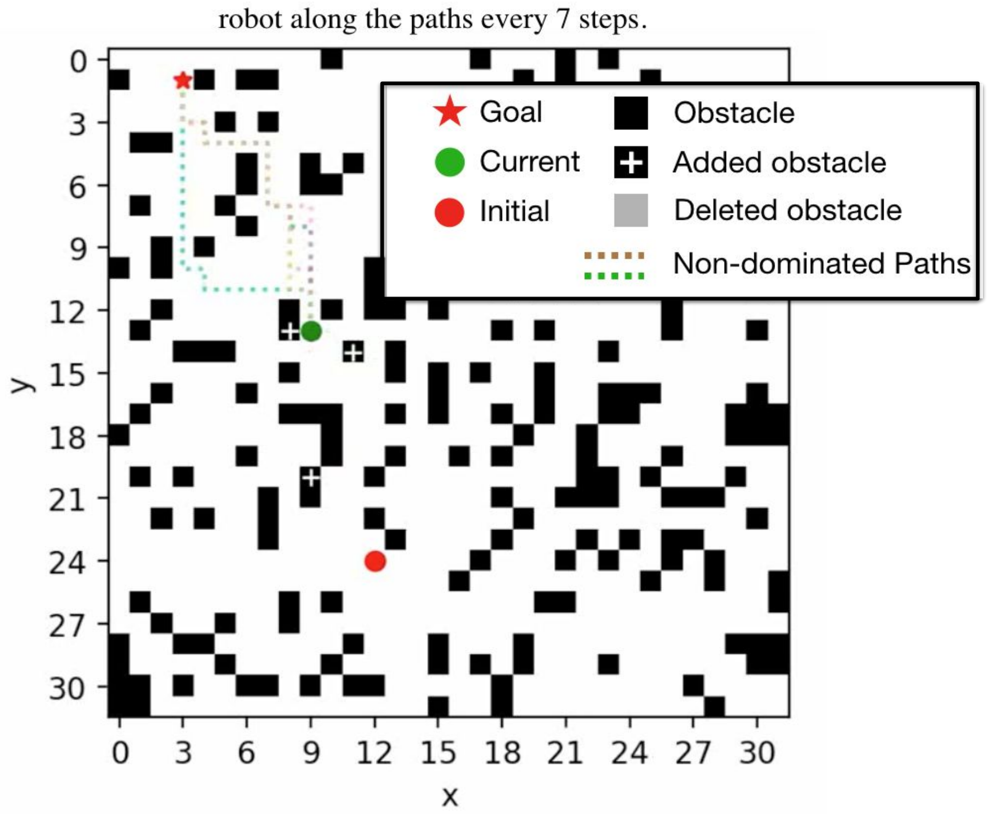
Paper

Google Scholar

4. Results

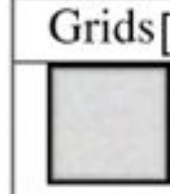
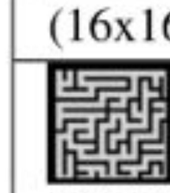
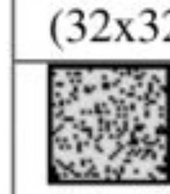
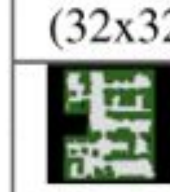
- Test with M=2,3,4 objectives, each component of edge cost vectors is randomly selected from [1,5].
- Iteratively (1) plan, (2) randomly select and execute, (3) add/delete obstacles.
- Run time limit 5 minutes.

Add an obstacles in the front of the robot along the paths every 7 steps.



Varying Maps, Varying #Objectives

Fixed M=2 (two objective), varying maps

Grids[1]	Algorithm	Exp.	R.T.	Sol.
 (16x16)	NAMOA*	111.8	**0.03**	3.0
	MOD*	39.1	0.35	3.0
	MOPBD*	**3.9**	0.06	3.0
 (32x32)	NAMOA*	1556.6	0.55	10.5
	MOD*	92.1	3.15	10.5
	MOPBD*	**19.7**	**0.17**	10.5
 (32x32)	NAMOA*	829.5	0.22	4.9
	MOD*	311.0	3.51	4.9
	MOPBD*	**35.0**	**0.12**	4.9
 (65x81)	NAMOA*	5923.3	2.85	16.3
	MOD*	208.4	12.6	12.3
	MOPBD*	**28.0**	**2.43**	16.3

Average over all instances
(*)Timeout in some instances

Fixed map (Maze), varying M.

M	Planner	Remove Obst.	Add Obst.
2	MOPBD* (ours)	**0.0060 (0.070)**	**0.018 (0.12)**
	NAMOA*	0.042 (0.15)	0.045 (0.19)
3	MOPBD* (ours)	**0.037 (4.6)**	**0.14 (14)**
	NAMOA*	0.099 (4.4)	0.17 (6.2)
4	MOPBD* (ours)	**0.062 (1.44)**	0.24 (15)
	NAMOA*	0.12 (2.13)	**0.17 (5.0)**

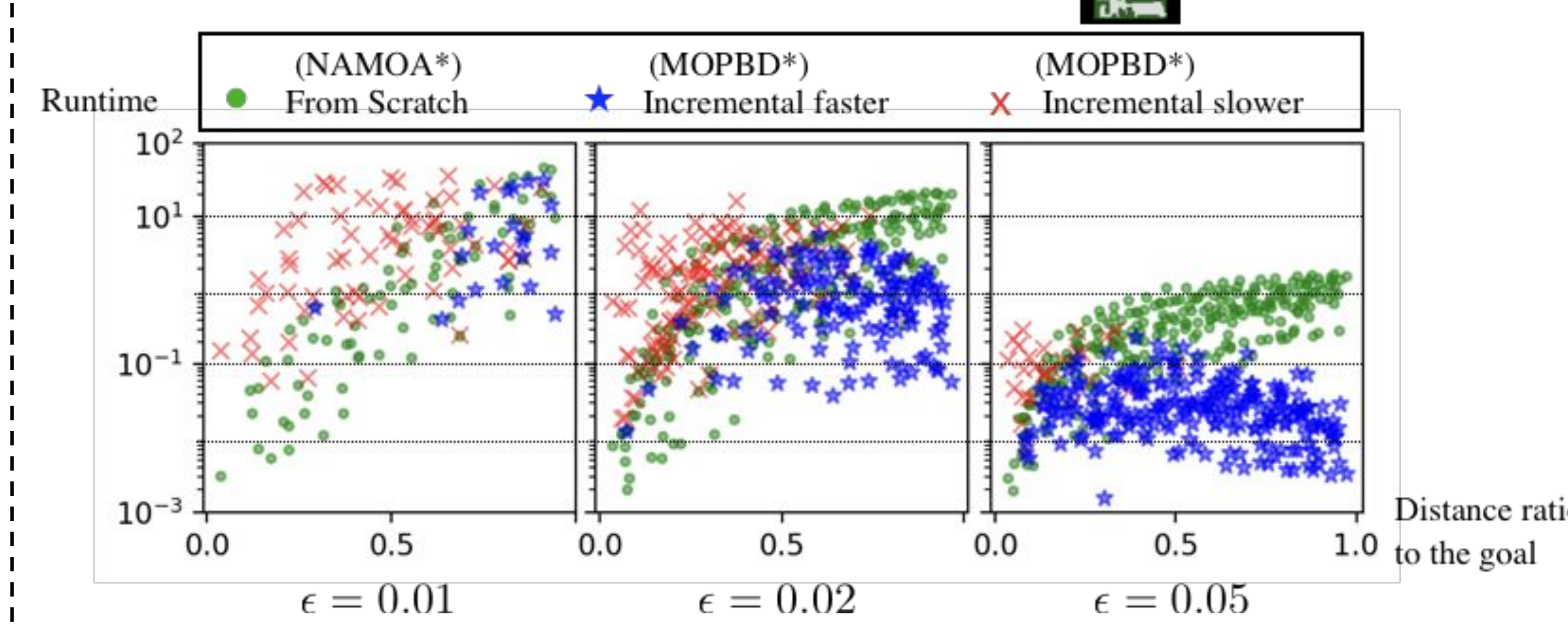
Runtime in format Median (Average)

- MOPBD* outperforms MOD* in all scenario;
- For run time, MOPBD*, in general, outperforms NAMOA* (search from scratch) on average;
- For number of expansion (path-based), MOPBD* outperforms NAMOA*;

[1] Stern, Roni, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li et al. "Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks." In Symposium on Combinatorial Search. 2019.

Sub-optimal Variant

Fixed M=3 (three objective), fixed map of size 65x81 



- Larger epsilon, the advantage of MOPBD* is more obvious.
- When the cost of an edge changes, MOPBD* needs to recursively delete all paths that go through the edge.
- Larger epsilon leads to more pruning (i.e. fewer partial paths at each node) computationally less expensive to delete.