Lifelong Search-based Planning: From Incremental Planning to Planning with Experience

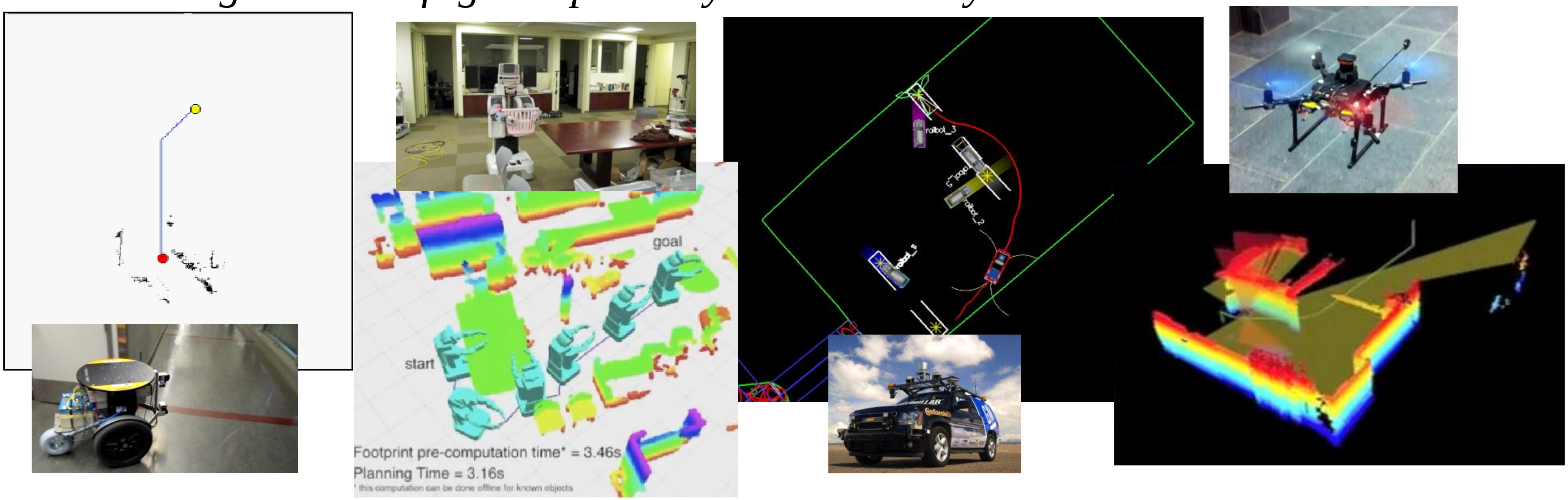
Maxim Likhachev
Robotics Institute & NREC
Carnegie Mellon University

based on the joint work with:
Mike Phillips, Ben Cohen, Andrew Dornbush
Sven Koenig, Sachin Chitta

Heuristic Search for Repeated Planning

Planning is often a repeated process:

navigation and flight in partially-known and dynamic environments



- low-dimensional graph

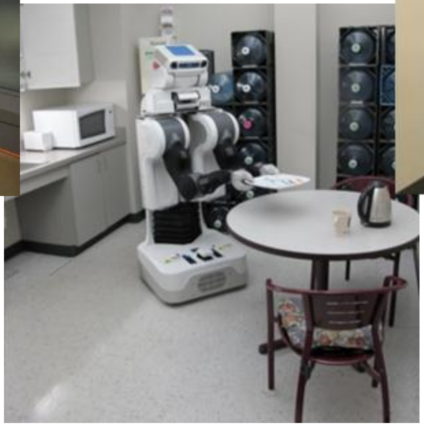
- (relatively) small changes in the graph plus moving start

incremental graph search techniques

Heuristic Search for Repeated Planning

Planning is often a repeated process:

solving similar planning problems for repetitive tasks



- high-dimensional graph

- larger changes in the graph plus different start and goal

graph search with Experience (E-graphs)

Outline

- Two Classes of Incremental Graph Search
  - *Basic idea behind D*, D* Lite, LPA* and its extensions*
  - *Basic idea behind Adaptive A* and its extensions*
  - *What these approaches can and cannot solve and why*

- Graph Search with Experience
  - *Overview of planning with E-graphs*

Outline

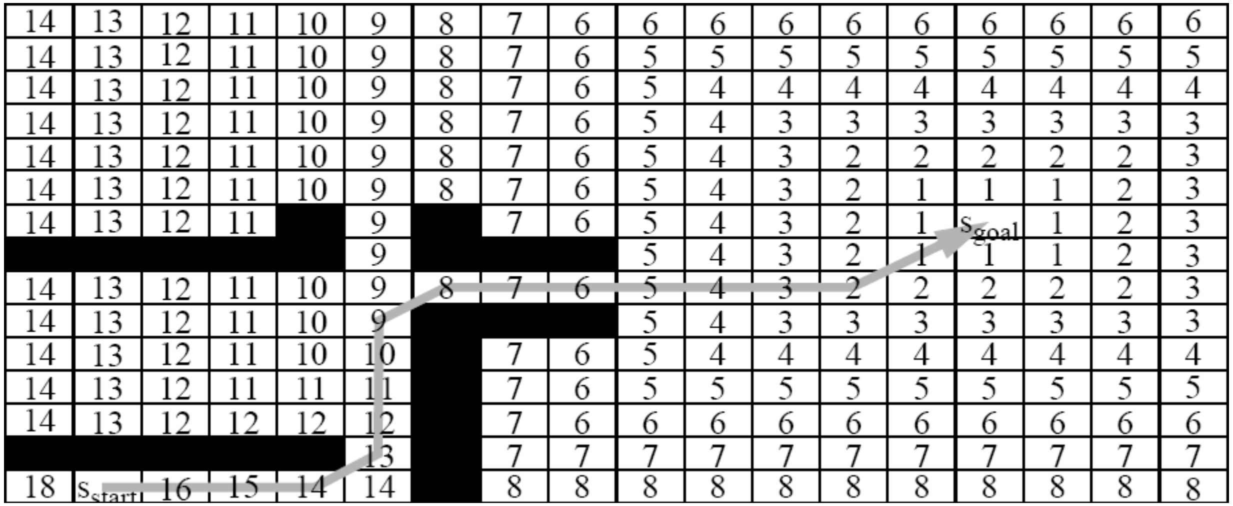
- Two Classes of Incremental Graph Search
  - *Basic idea behind D*, D* Lite, LPA* and its extensions*
  - *Basic idea behind Adaptive A* and its extensions*
  - *What these approaches can and cannot solve and why*

- Graph Search with Experience
  - *Overview of planning with E-graphs*

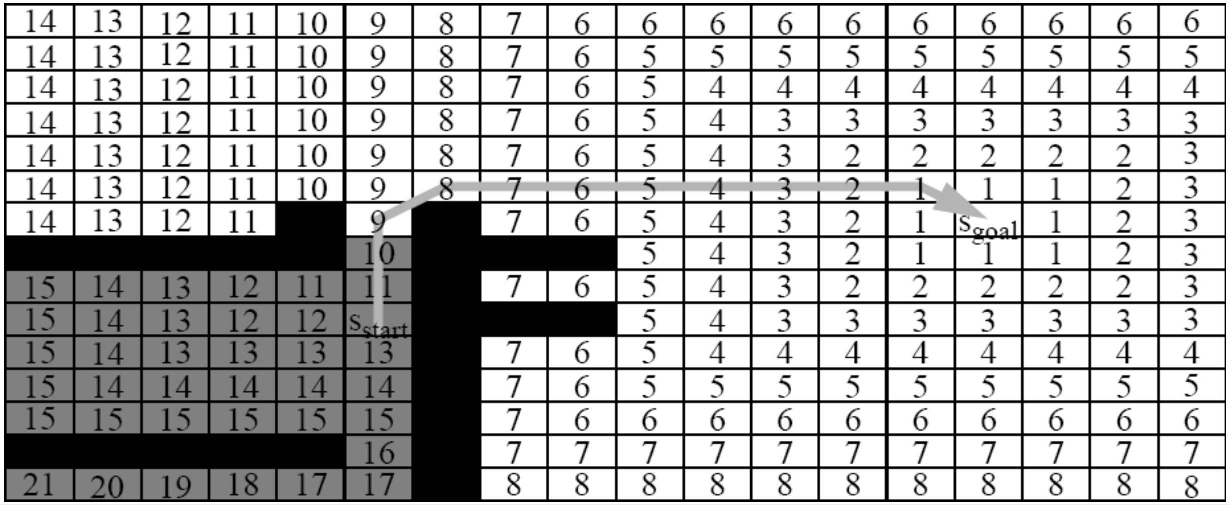
Basic Idea Behind D*, D* Lite, LPA* and etc.

- **Reuse state values from previous searches**

cost of least-cost paths to goal at first planning episode



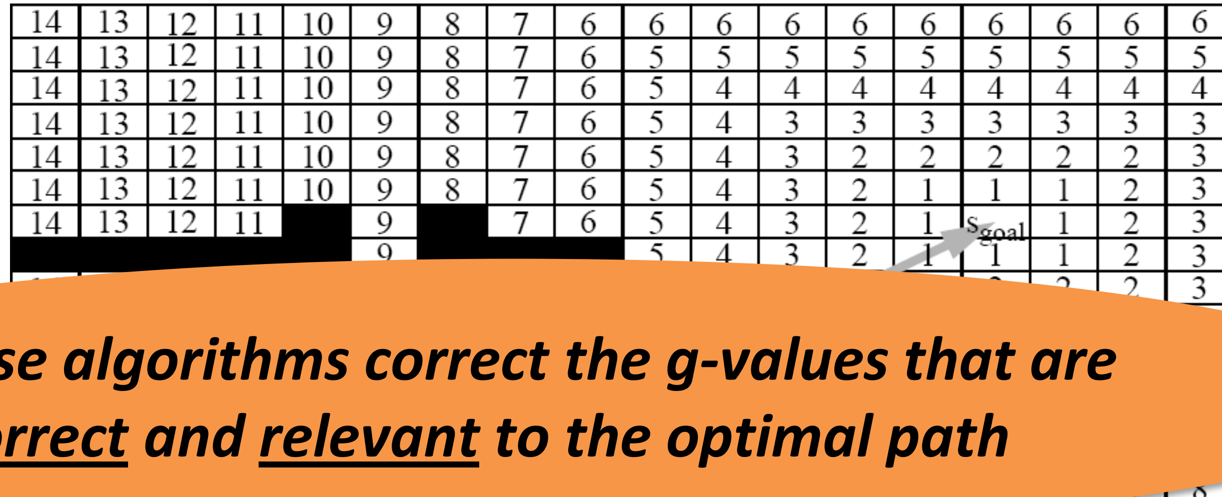
cost of least-cost paths to goal after the door turns out to be closed



Basic Idea Behind D*, D* Lite, LPA* and etc.

- **Reuse state values from previous searches**

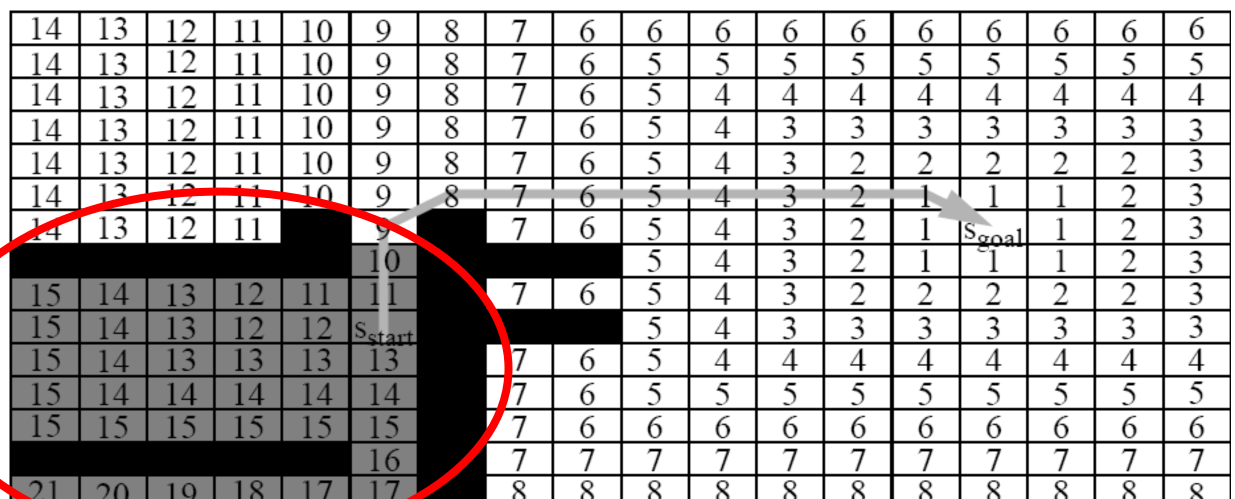
cost of least-cost paths to goal at first planning episode



These algorithms correct the g-values that are <u>incorrect</u> and <u>relevant</u> to the optimal path

These are the only changes to the g-values

cost of least-cost paths to goal after the door turns out to be closed

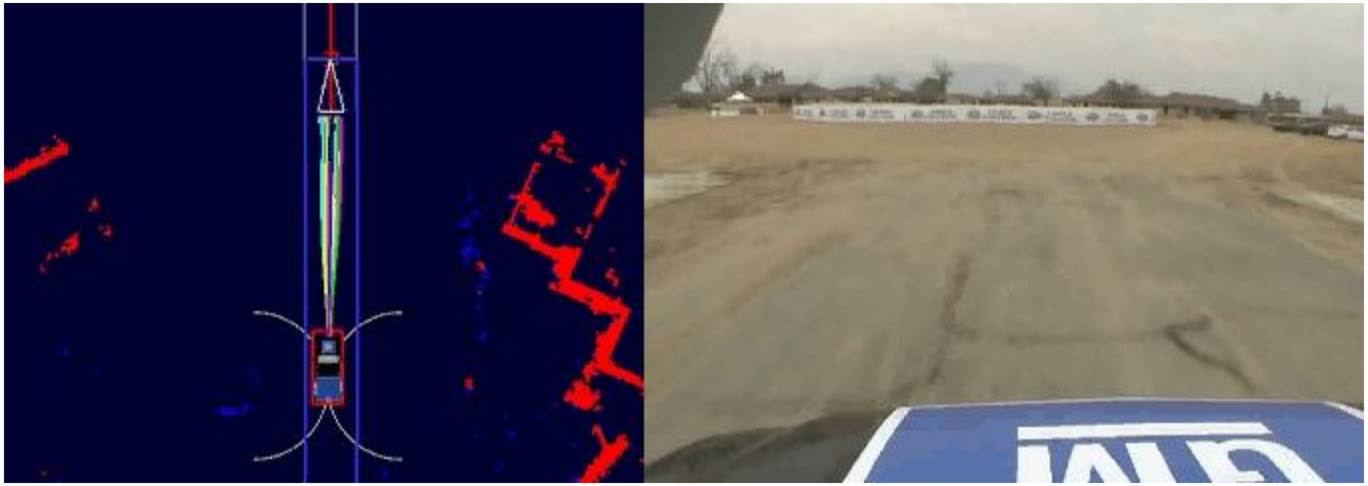


Application to Autonomous Flight and Navigation

- Anytime D* (=ARA*+ D* Lite) for 4D re-planning in real-time ($<x,y,z,\Theta>$ for flight and $<x,y,\Theta,v>$ for flight)

…but:

- *require iterating over all edges whose cost change*
- *effective only when changes are relatively small*



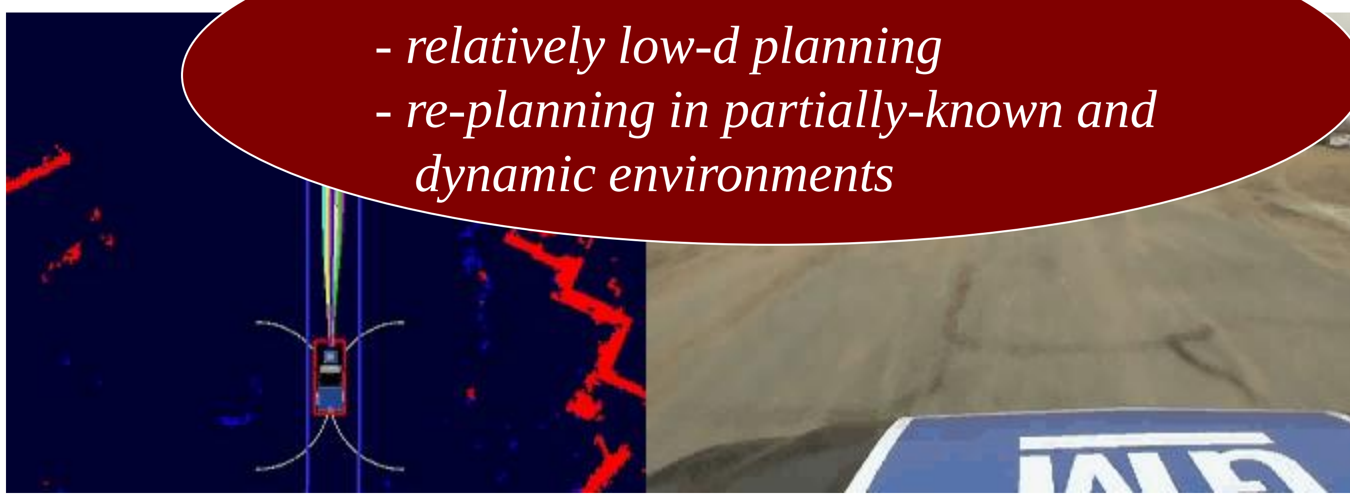
part of efforts by Tartanracing team from CMU for the Urban Challenge 2007 race

Application to Autonomous Flight and Navigation

- Anytime D* (=ARA*+ D* Lite) for 4D re-planning in real-time ($<x,y,z,\Theta>$ for flight and $<x,y,\Theta,v>$ for flight)

…but:
- *require iterating over all edges whose cost change*
- *effective only when changes are relatively small*

Limited to:
- *relatively low-d planning*
- *re-planning in partially-known and dynamic environments*



part of efforts by Tartanracing team from CMU for the Urban Challenge 2007 race

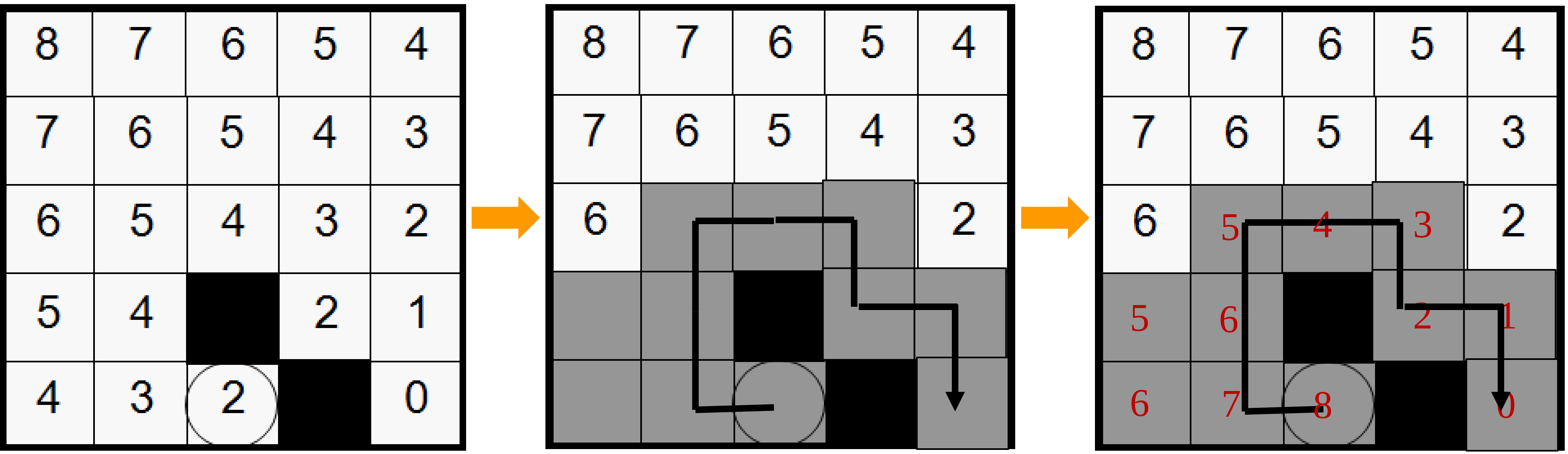
Basic idea behind Adaptive A* and its variants

- **Improve ("learn") heuristic values**

initial heuristics that estimate cost-to-goals

states expanded during planning

heuristics of expanded states improved according to: h(s) = g(s)-solution cost



little bookkeeping but:
- *less effective as incremental search*
- *mostly for low-d problems (e.g., 2D target pursuit)*
- *also limited to small changes*

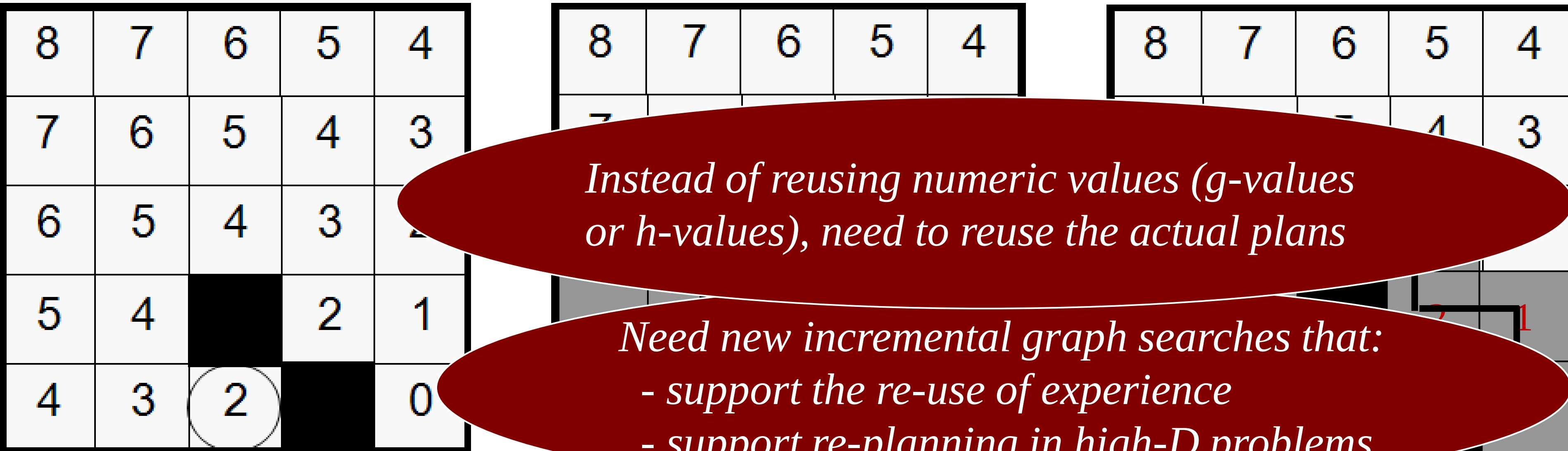
Basic idea behind Adaptive A* and its variants

- **Improve ("learn") heuristic values**

initial heuristics that estimate cost-to-goals

states expanded during planning

heuristics of expanded states improved according to: h(s) = g(s)-solution cost



Instead of reusing numeric values (g-values or h-values), need to reuse the actual plans

Need new incremental graph searches that:
- support the re-use of experience
- support re-planning in high-D problems

little bookkeeping but:
- less effective as incremental search
- mostly for low-d problems (e.g., 2D target pursuit)
- also limited to small changes

Outline

- Two Classes of Incremental Graph Search
  - *Basic idea behind D*, D* Lite, LPA* and its extensions*
  - *Basic idea behind Adaptive A* and its extensions*
  - *What these approaches can and cannot solve and why*

- Graph Search with Experience
  - *Overview of planning with E-graphs*

Planning with Experience Graphs

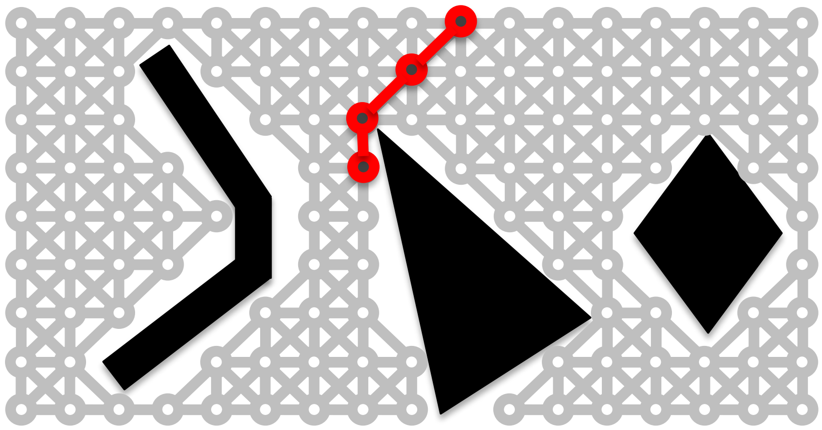
- Many planning tasks are repetitive
  - loading a dishwasher
  - opening doors
  - moving objects around a warehouse
  - …

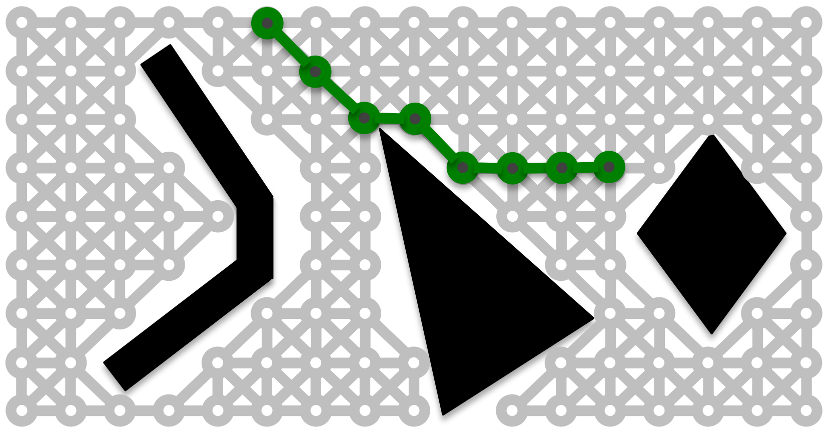
- Can we re-use prior experience to accelerate planning, in the context of search-based planning?

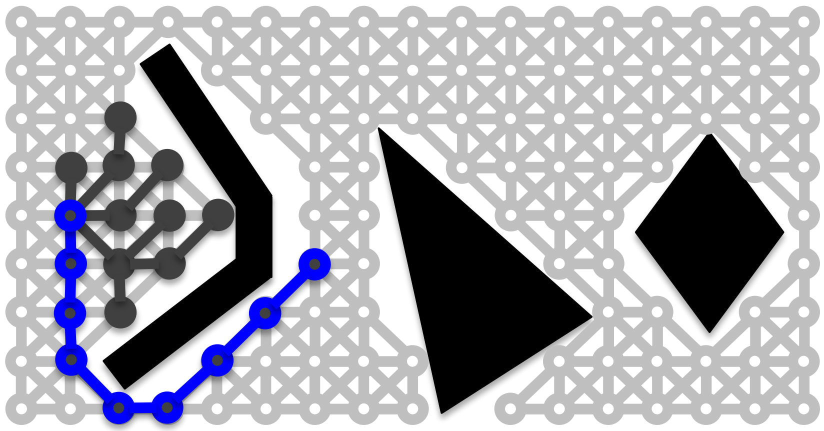
- Would be especially useful for high-dimensional problems such as mobile manipulation!

Planning with Experience Graphs

Given a set of previous paths (experiences)…

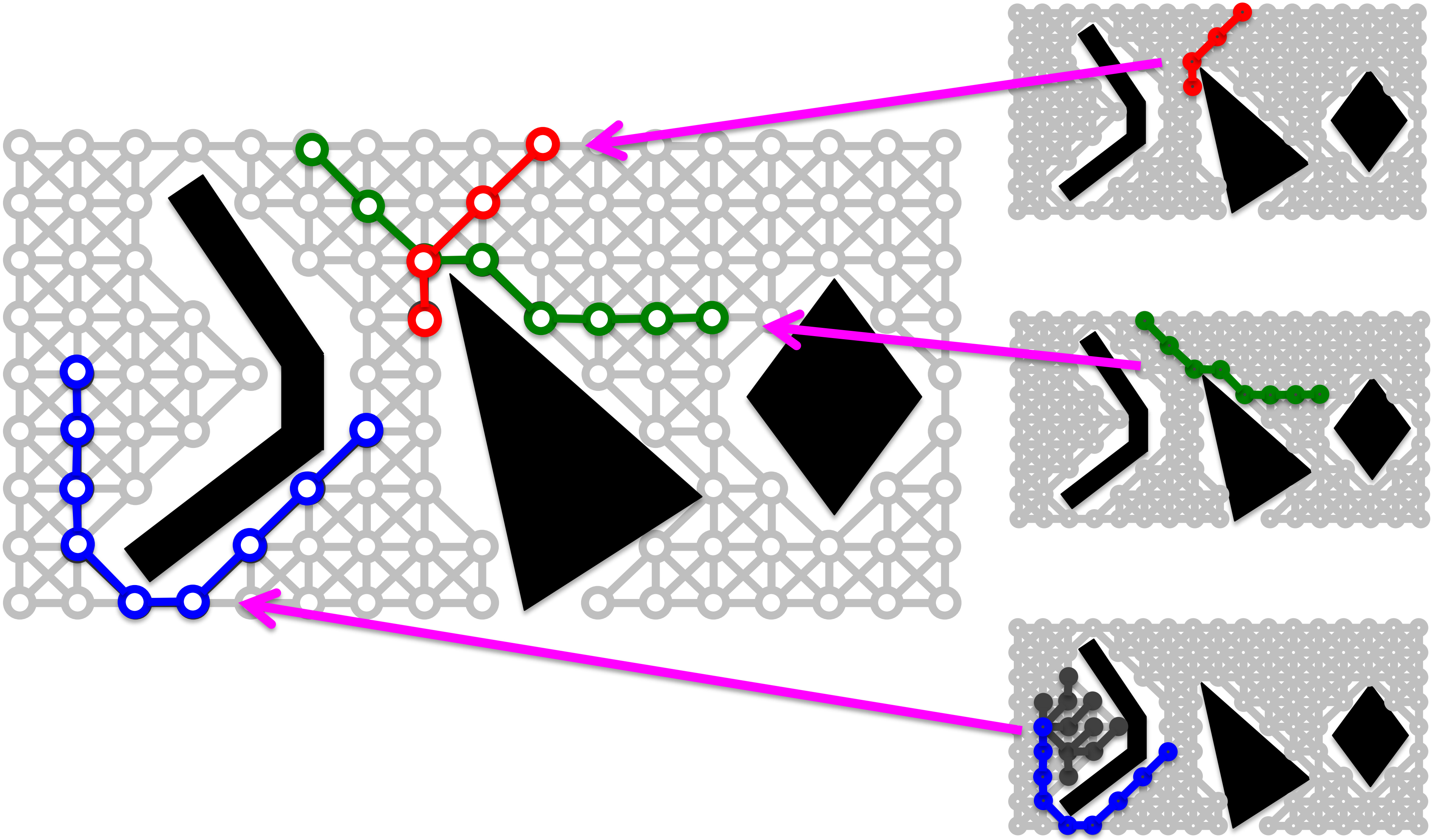






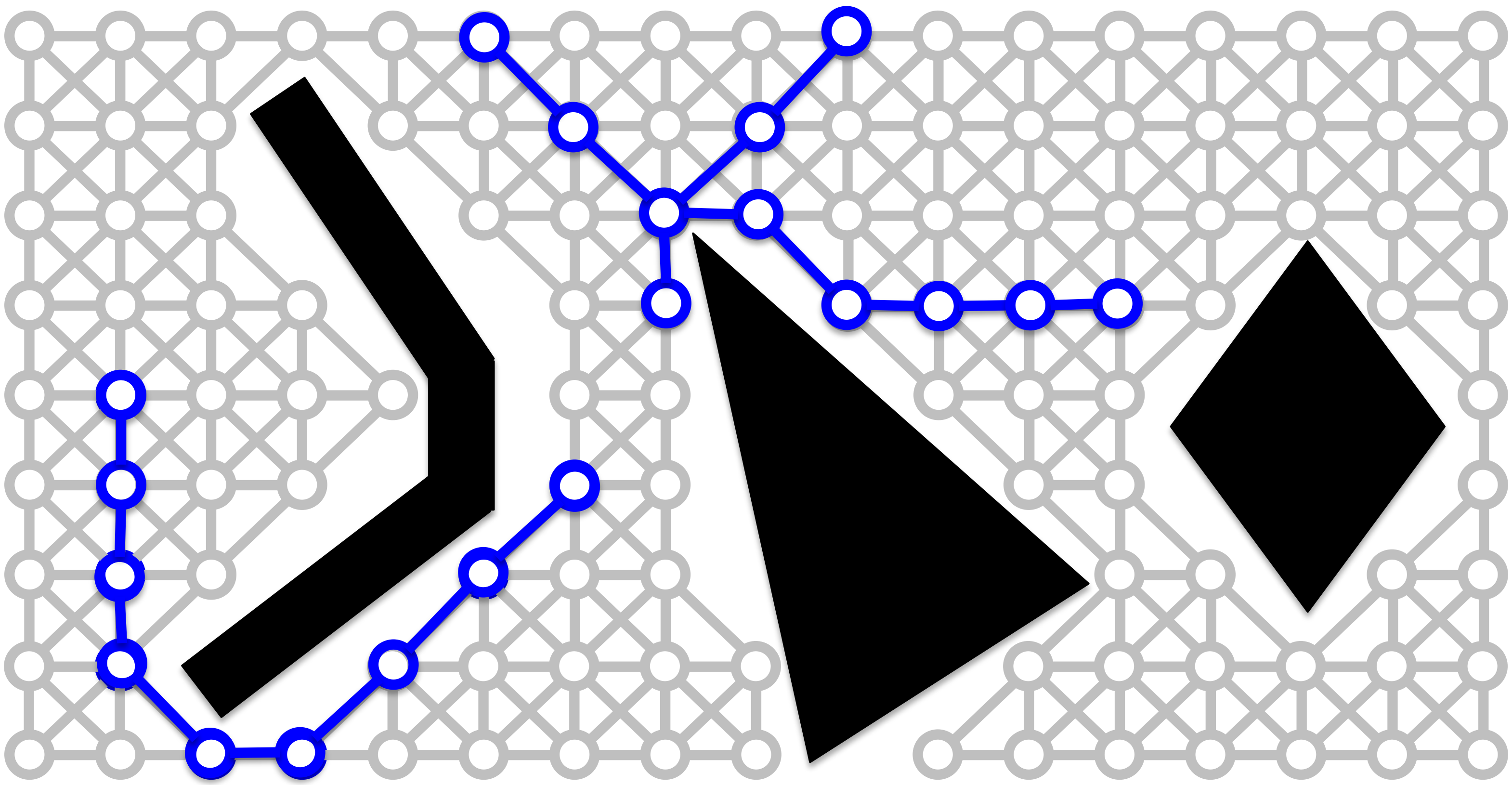
Planning with Experience Graphs

Put them together into an *E*-graph (Experience graph)



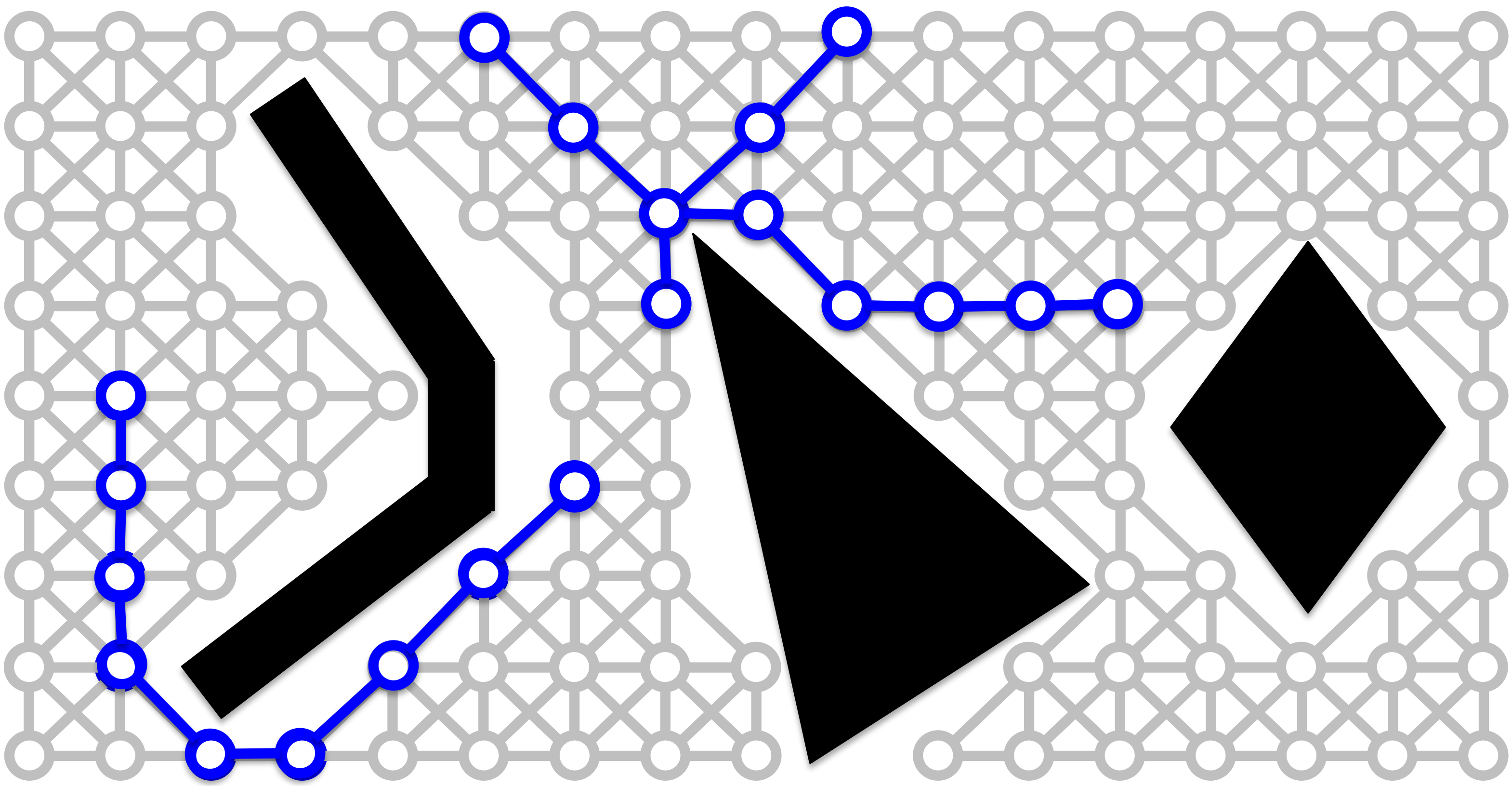
Planning with Experience Graphs

- *E*-Graph [Phillips et al., RSS'12]:
  - *Collection of previously computed paths or demonstrations*
  - *A sub-graph of the original graph*



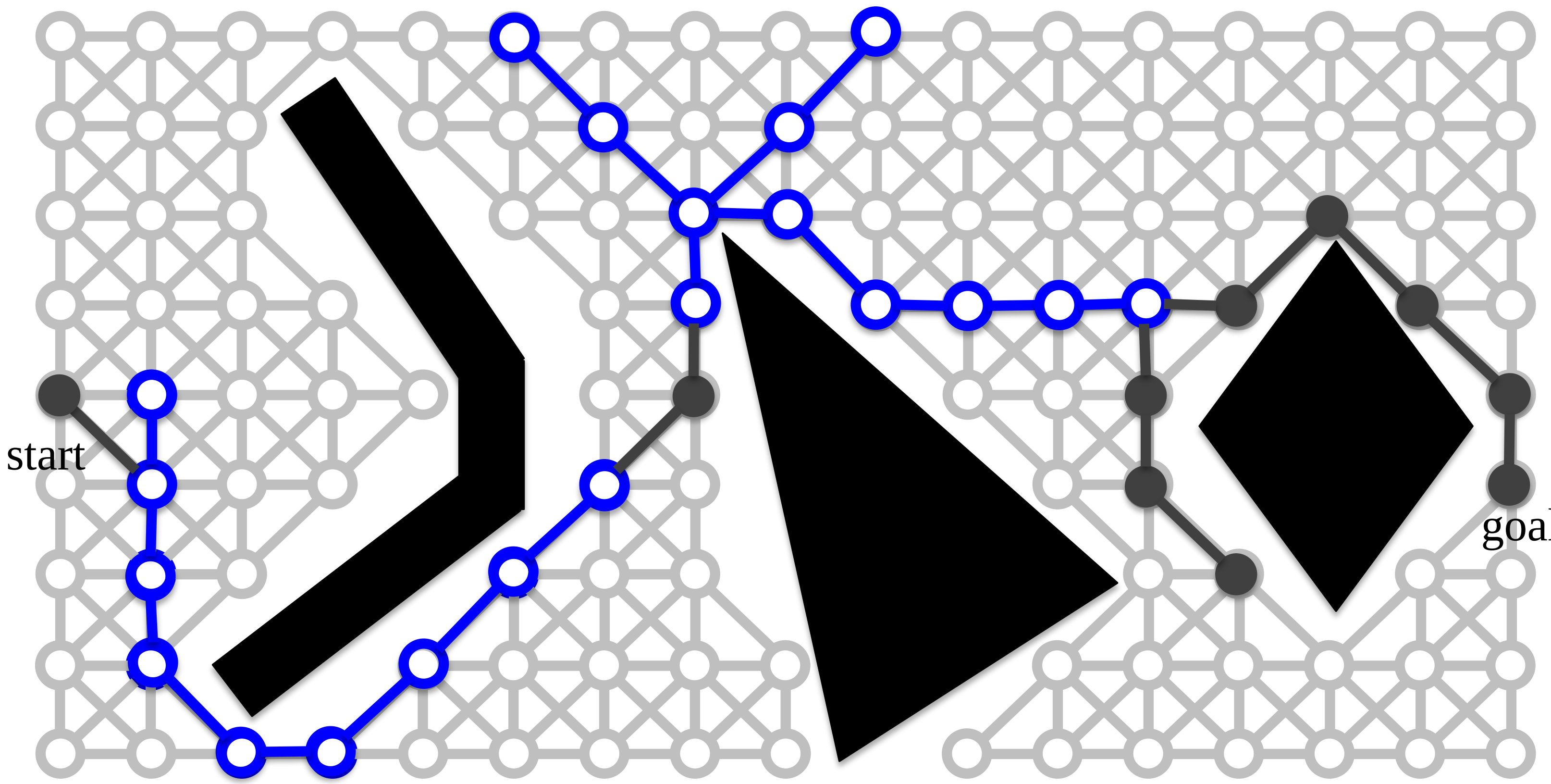
Planning with Experience Graphs

Given a new planning query…



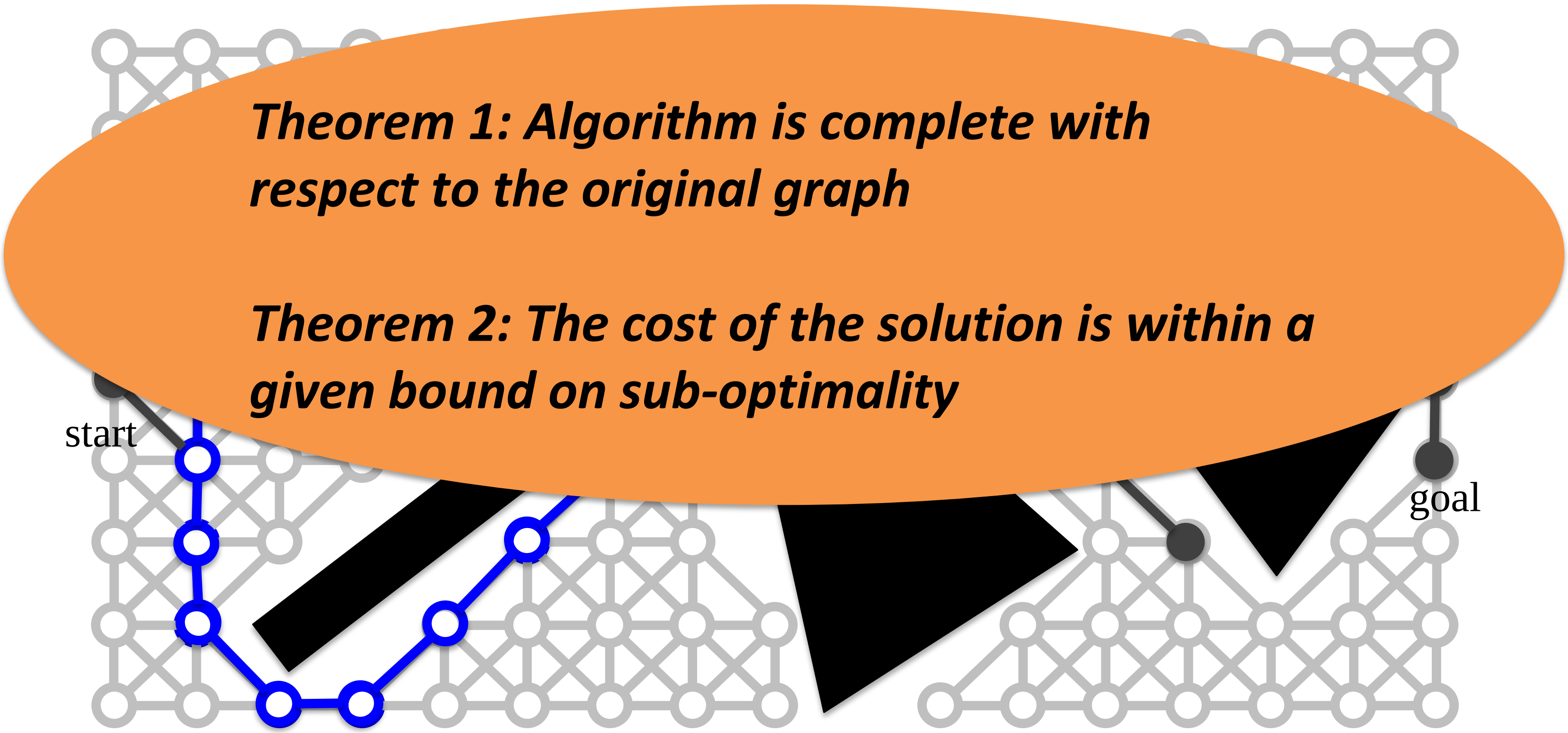
Planning with Experience Graphs

…re-use E-graph. For repetitive tasks, planning becomes much faster



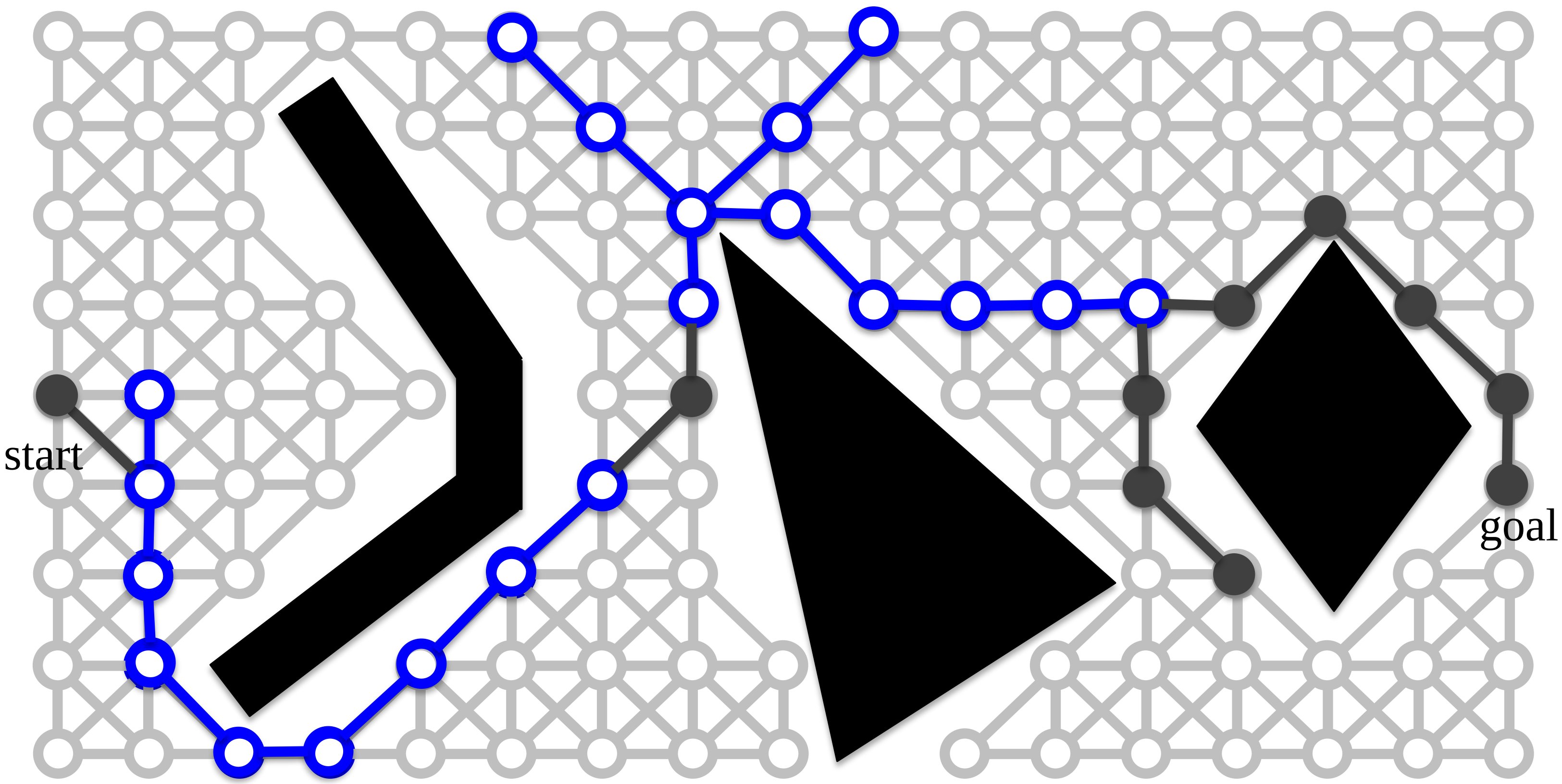
Planning with Experience Graphs

…re-use E-graph. For repetitive tasks, planning becomes much faster



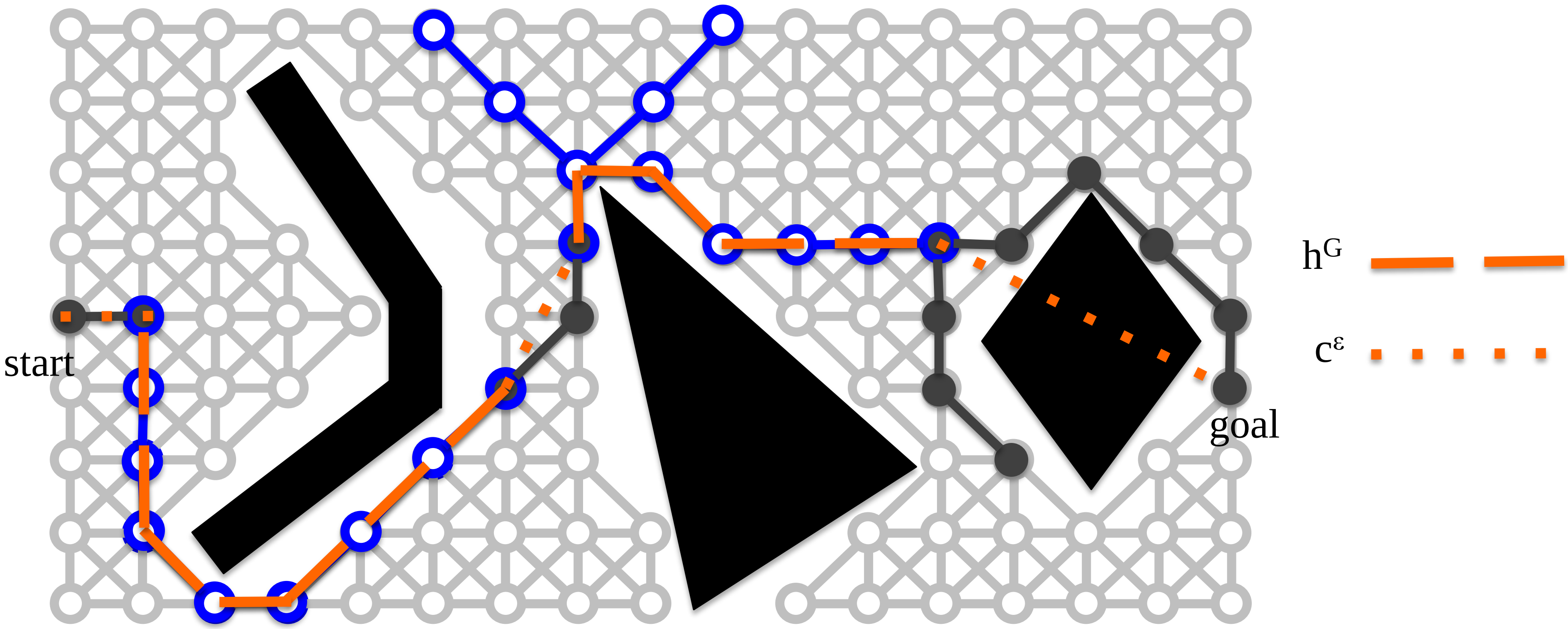
Planning with Experience Graphs

- Reuse *E*-Graph by:
  - Introducing a new heuristic function
  - Heuristic guides the search toward expanding states on the *E*-Graph **within sub-optimality bound ε**



Planning with Experience Graphs

- Focusing search towards E-graph **within sub-optimality bound *ε***

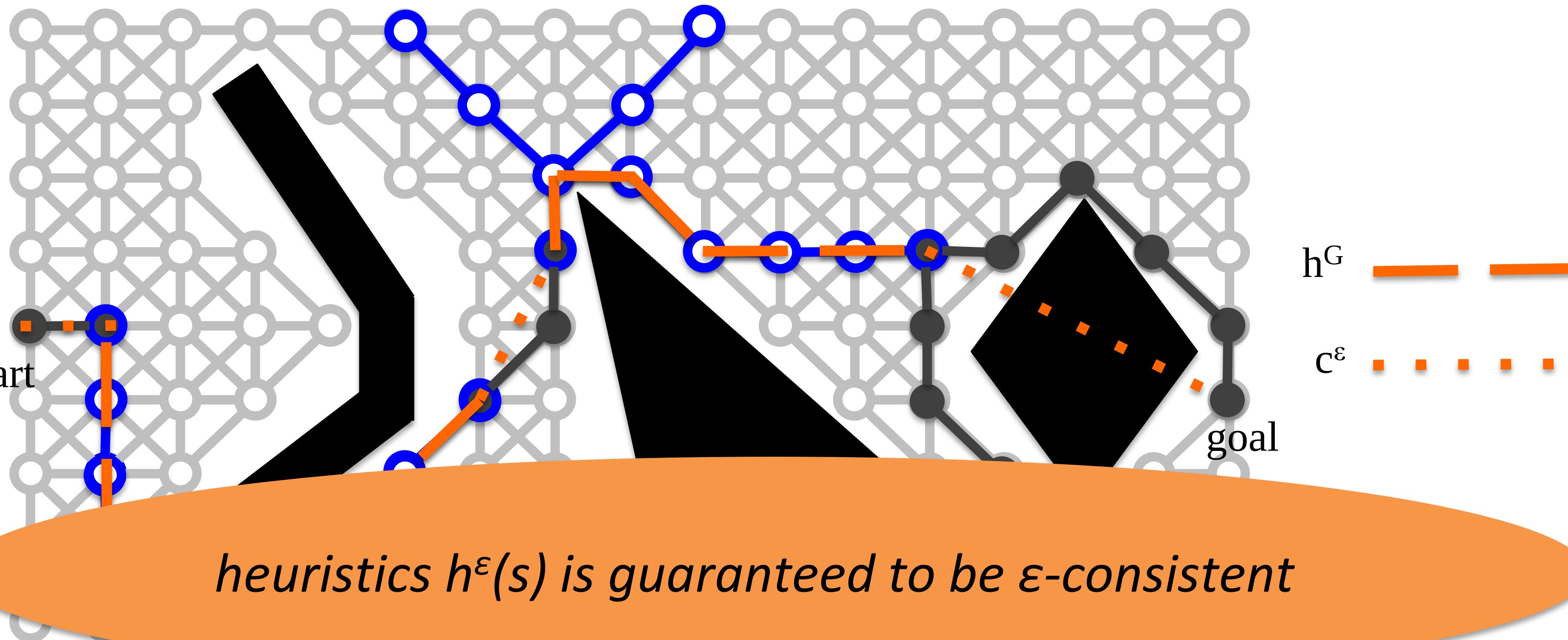


Heuristic computation finds a min cost path using two kinds of "edges"

$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\overbrace{\varepsilon^{\mathcal{E}} h^G(s_i, s_{i+1})}^{\text{Traveling off the } E\text{-Graph uses an inflated original heuristic}}, \overbrace{c^{\mathcal{E}}(s_i, s_{i+1})}^{\text{Traveling on } E\text{-Graph uses actual costs}}\}$$

Planning with Experience Graphs

- Focusing search towards E-graph **within sub-optimality bound *ε***



heuristics $h^\varepsilon(s)$ is guaranteed to be ε-consistent

Heuristic computation finds a min cost path using two kinds of "edges"

$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\underbrace{\varepsilon^{\mathcal{E}} h^G(s_i, s_{i+1})}_{\text{Traveling off the } E\text{-Graph uses an inflated original heuristic}}, \underbrace{c^{\mathcal{E}}(s_i, s_{i+1})}_{\text{Traveling on } E\text{-Graph uses actual costs}}\}$$

Planning with Experience Graphs

Theorem 5. Completeness w.r.t. the original graph G: *Planning with E-graphs is guaranteed to find a solution, if one exists in G*

Theorem 6. Bounds on sub-optimality: *The cost of the solution found by planning with E-graphs is guaranteed to be at most ε-suboptimal:*

$$cost(solution) \leq \varepsilon \; cost(optimal\ solution\ in\ G)$$

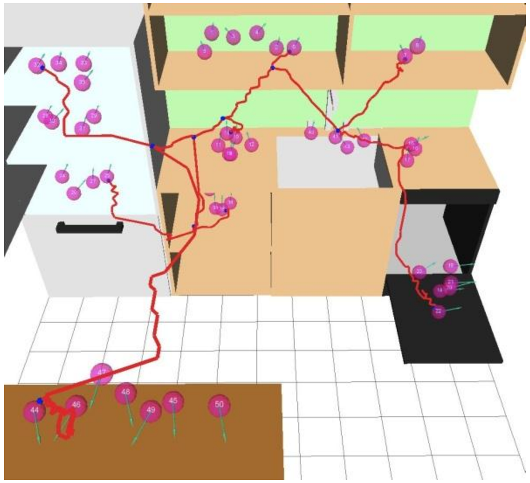
Planning with *E*-Graphs for Mobile Manipulation

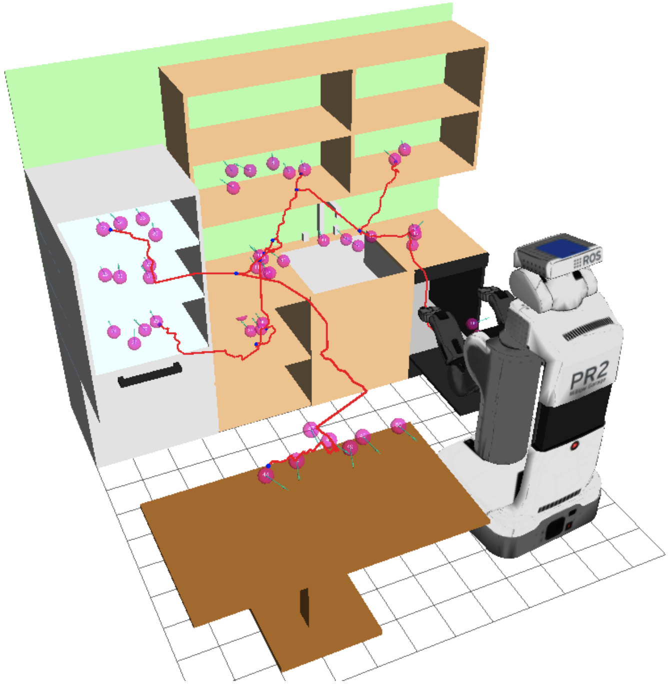
- Dual-arm mobile manipulation (10 DoF)

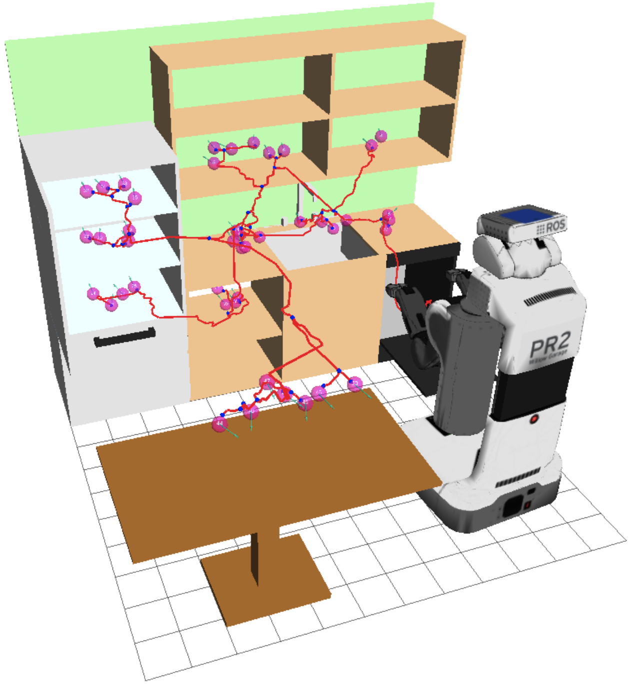
Planning with *E*-Graphs for Mobile Manipulation

Kitchen environment:

- moving objects around a kitchen
- bootstrap E-Graph with 10 representative goals
- tested on 40 goals in natural kitchen locations







Kitchen environment: planning times

	Success (of 40)	Mean Time (s)	Std. Dev. (s)	Max (s)
E-Graphs	40	0.33	0.25	1.00

Method	Success (of 40)	Mean Speed-up	Std. Dev.	Max
Weighted A*	37	34.62	87.74	506.78
RRT-Connect	40	1.97	2.35	11.32
PRM	25	16.52	74.25	372.90
RRT* (first solution)	23	50.99	141.35	613.54

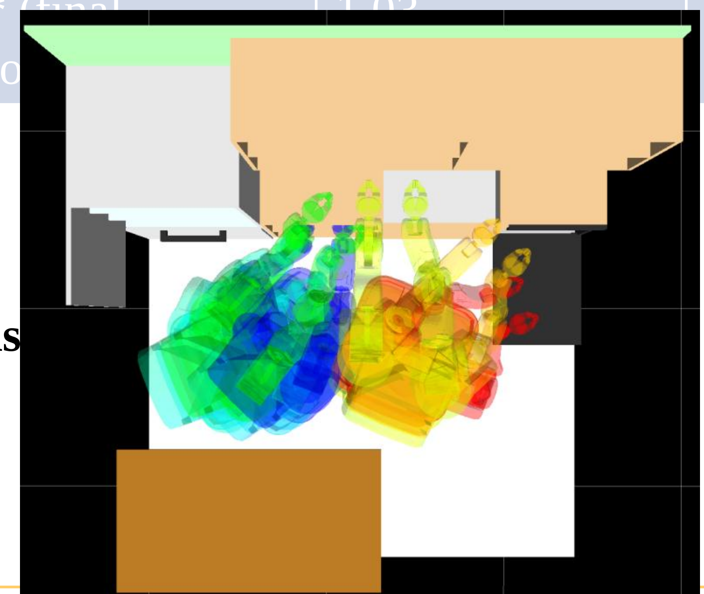
- Max planning time of 2 minutes
- Sub-optimality bound of 20 (for E-Graphs and Weighted A*)

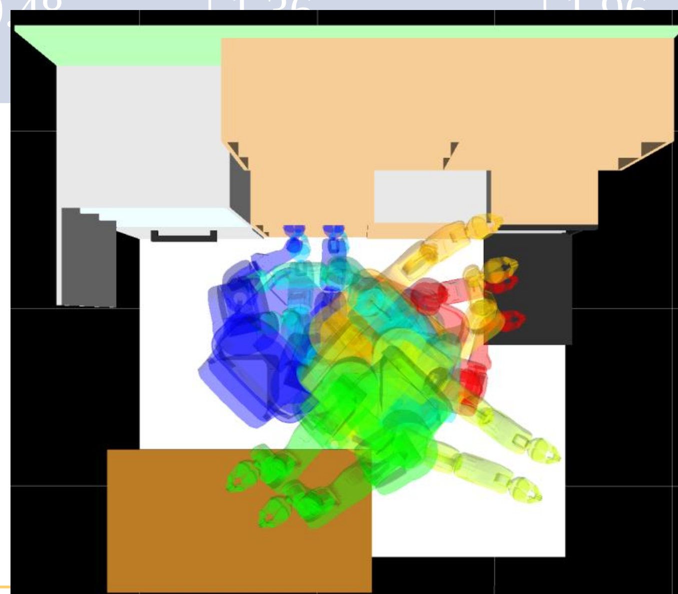
- All sampling methods are from OMPL
- Shortcutting was applied to sampling methods
- Sampling methods (which require configuration space goals) are given the goal found by E-Graphs

Kitchen environment: path quality ratio (method/E-graph)

Method	Object XYZ Path Length Ratio	Std. Dev.	Base XY Path Length Ratio	Std. Dev.
Weighted A*	0.91	0.68	1.14	1.40
RRT-Connect	2.54	4.67	3.45	9.67
PRM	0.85	0.32	0.88	0.48
RRT* (first solution)	1.08	0.60	1.39	1.79
RRT* (final solution)	1.03	0.48	1.36	1.96

E-Graphs





RRT-Connect

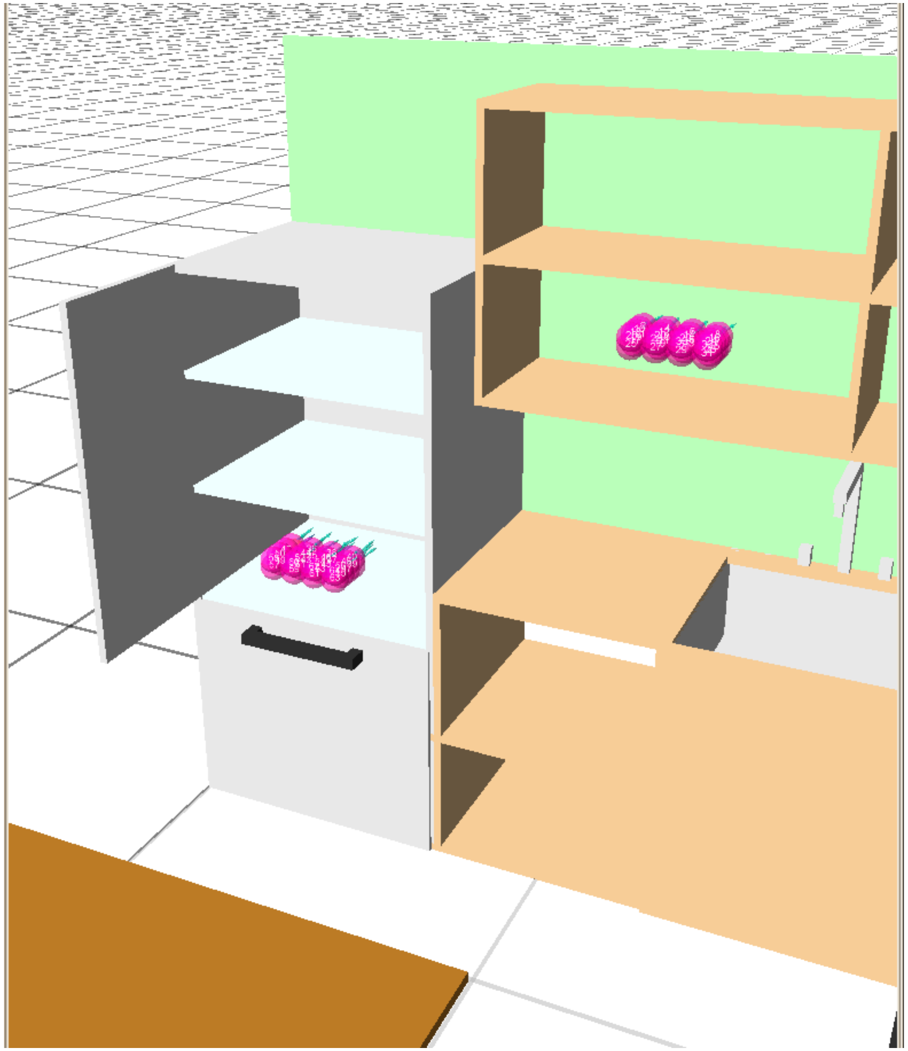
Kitchen environment: path consistency

58 goals (between the two locations)

	Similarity (without warping)	Dynamic Time Warping
E-Graphs	23447	407
RRT-Connect	101456	1512
PRM	42901	748
RRT*	N/A†	N/A†

† RRT* was unable to solve these cases

H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-26, no. 1, 1978.



Conclusions

- Existing incremental heuristic searches (D*, D* Lite, LPA*, Adaptive A*, etc.) are more suitable for
  - lower-dimensional planning problem
  - re-planning while operating in partially-known environments and dynamic environments
  - mostly because they “repair” the numeric value functions (g-values or h-values)
- Need new incremental heuristic searches that use plans to speed up planning rather than repair “value functions”
- Planning with Experience graphs is a step towards it
  - suitable for both high-D as well as low-D problems
  - developed mainly for improving planning for repetitive tasks

Future Directions

- Storing and loading Experience Graphs depending on the tasks and situations

- Use demonstrations as experiences

- Incremental searches for High-D planning problems

- **Students** who contributed to this work:
  - Ben Cohen
  - Mike Phillips
  - Andrew Dornbush
  - Jon Butzke
  - Brian MacAllister
  - Alex Kushleyev
- **Collaborators:**
  - Sachin Chitta
  - Sven Koenig
  - Dave Ferguson
- **Sponsors: Willow Garage, ARL, DARPA**

- Some of the software is available open-source (standalone and ROS compatible): http://www.sbpl.net/Software