Continuous Arvand: Motion Planning with Monte Carlo Random Walks

Weifeng Chen and Martin Müller
Presented by Robert Holte
Department of Computing Science
University of Alberta

Introduction

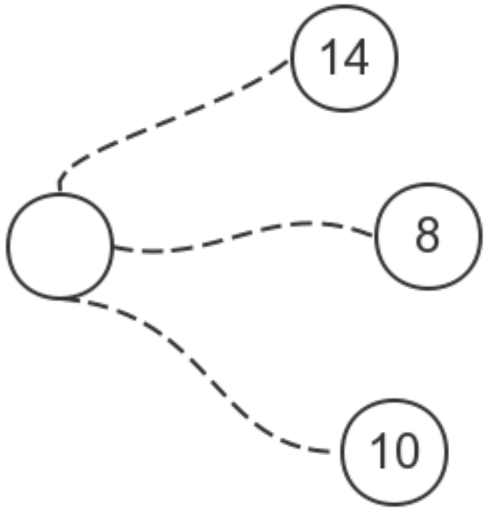
- Monte Carlo random walks (MRW) have been successful in classical deterministic planning with discrete states and actions.
- MRW uses random exploration of the local neighbourhood of a search state.
- Arvand is a family of planners using MRW approach in classical planning.
- The current work is an initial study adapting MRW to plan in continuous spaces.

Random Walks in Discrete State Spaces

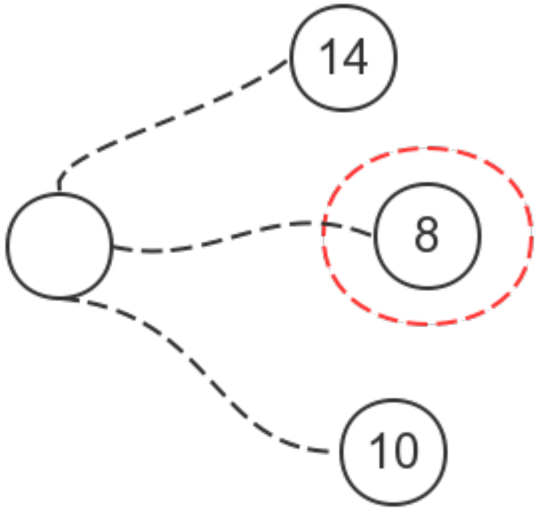
- MRW Procedure:
  - Start state s
  - Apply a sequence of randomly selected actions.
  - Use heuristic *h* to evaluate the endpoint.
  - Do this several times for s.
  - If no improvement, restart, otherwise repeat from best endpoint.
- Advantages:
  - Escape faster from local minima and plateaus
  - Combines greedy exploitation with random exploration
  - Avoid exhaustive search of dead-ends

Example of MRW

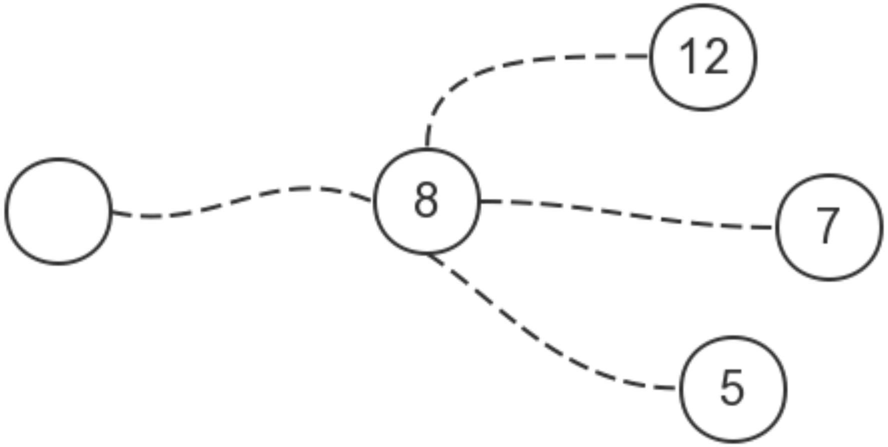
Example of MRW



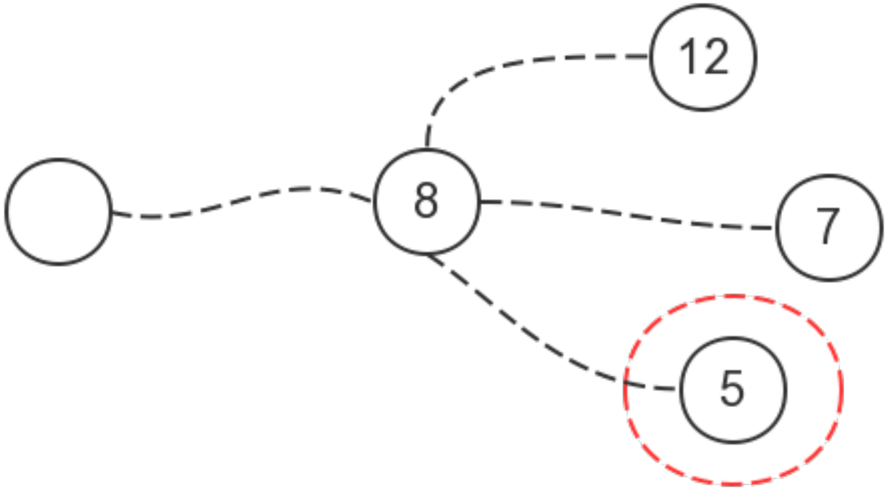
Example of MRW



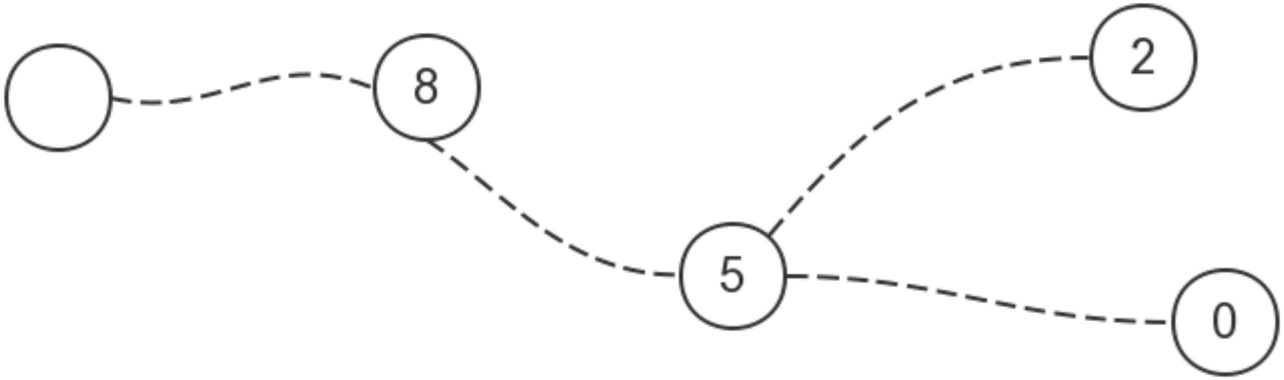
Example of MRW



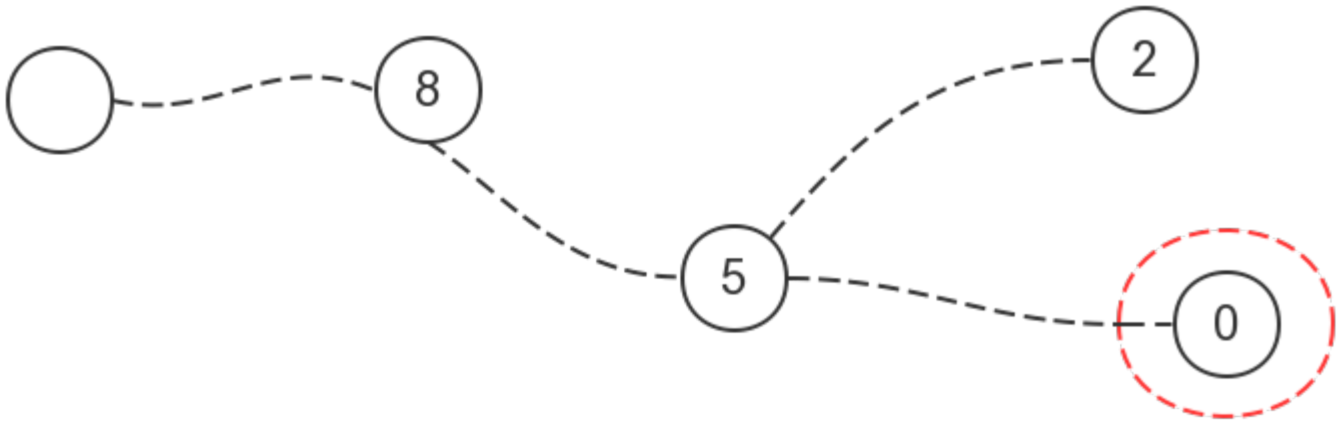
Example of MRW



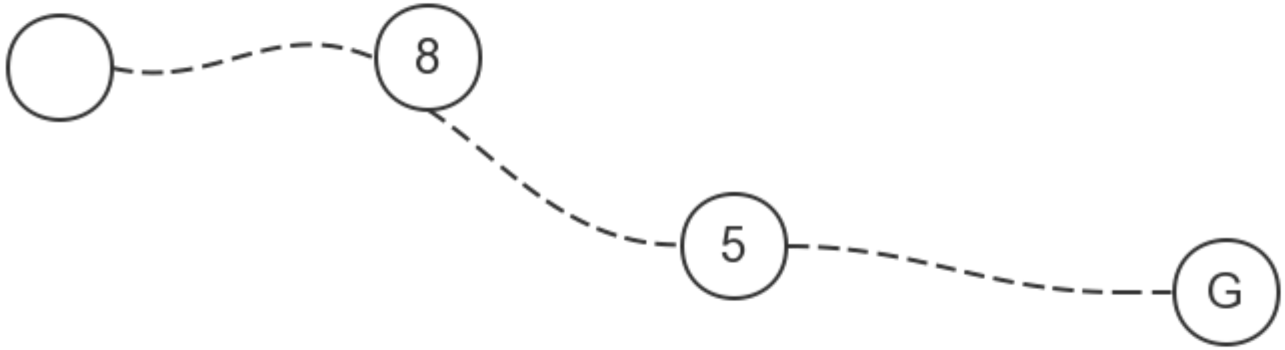
Example of MRW



Example of MRW



Example of MRW



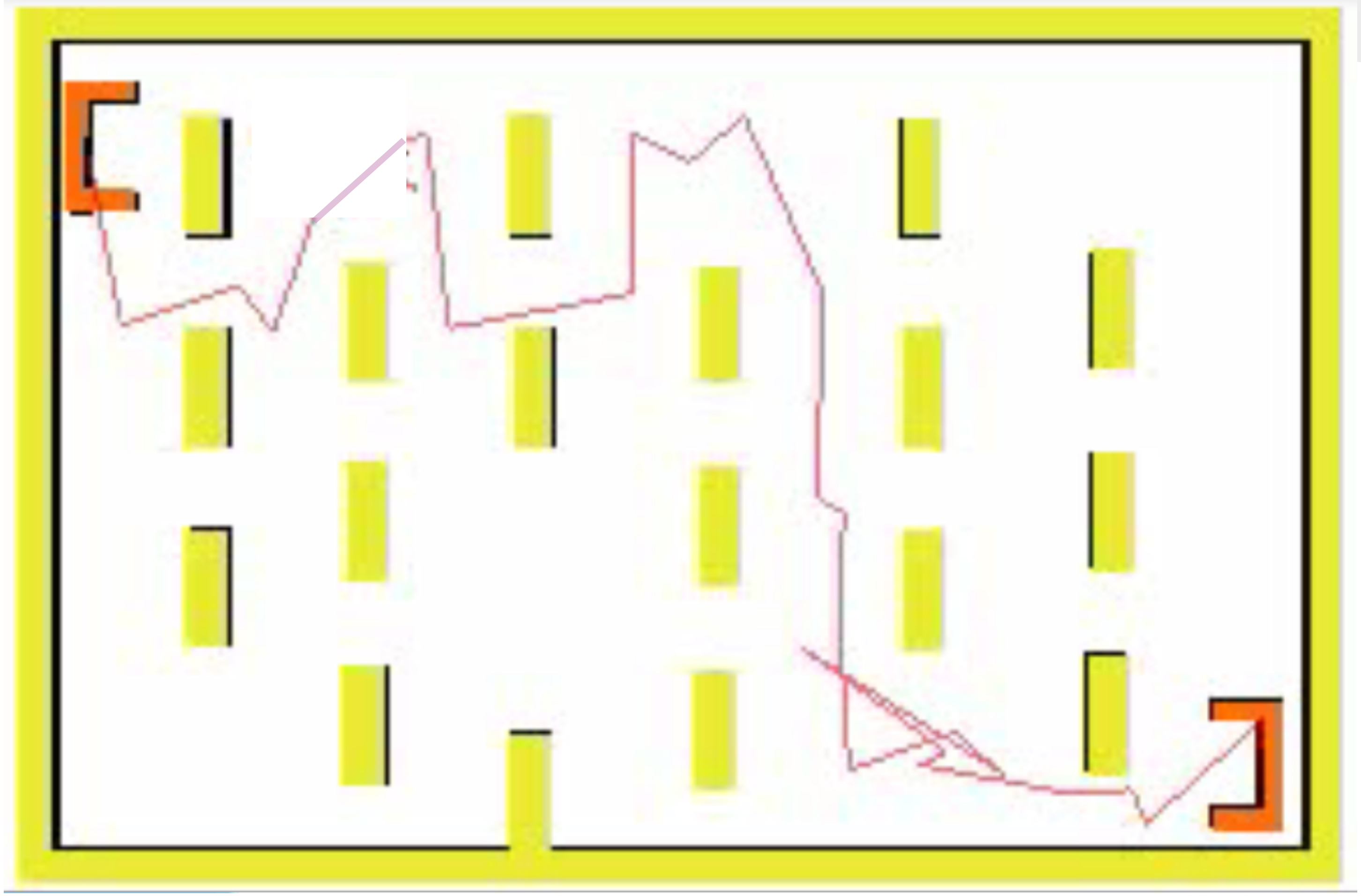
Random Walk Parameters

- Choices for terminating a random walk
  - Fixed length
  - Initial length, multiply when stuck
  - Local restarting rate r
    - Terminate walk with probability r at each step

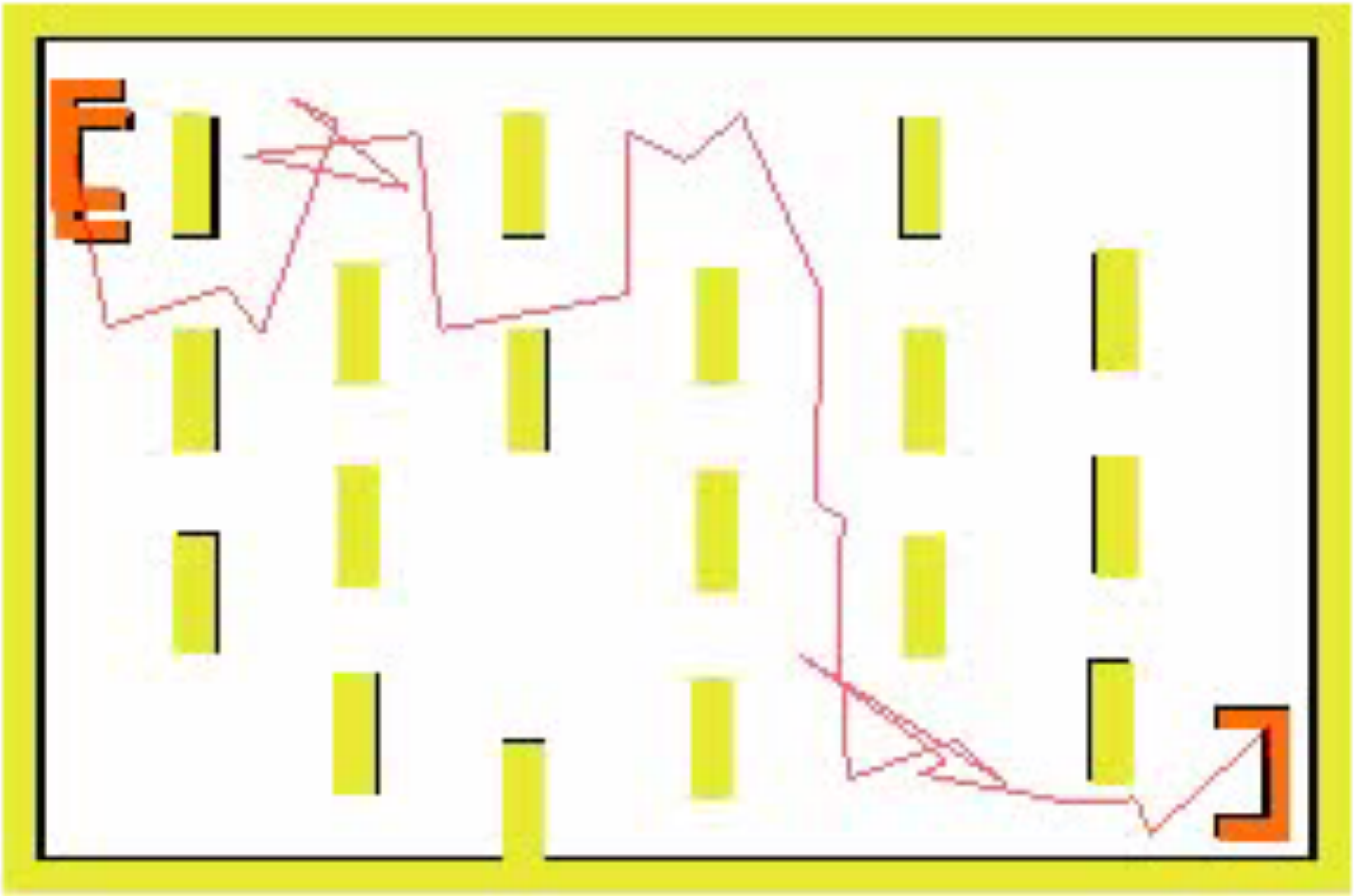
- Global restart mechanisms
  - Fixed number of search episodes
  - Restarting threshold t:
    - Restart when no improvement in last t walks
    - t is calculated adaptively*

* http://webdocs.cs.ualberta.ca/~mmueller/ps/2013/2013-IJCAI-arvand.pdf

Example – Barriers



Example – Barriers (video)



Classical vs Motion Planning

Main differences for MRW:

Component	Classical planning	Motion planning
State space	discrete	continuous
Goal checker	deterministic	approximate
Action execution	instant	continuous
Random walk	sample action $\rightarrow$ new state	sample state $\rightarrow$ new motion
Heuristic	Instance-specific, e.g. Fast Forward	C-space-specific, e.g. geometric distance

MRW for Motion Planning

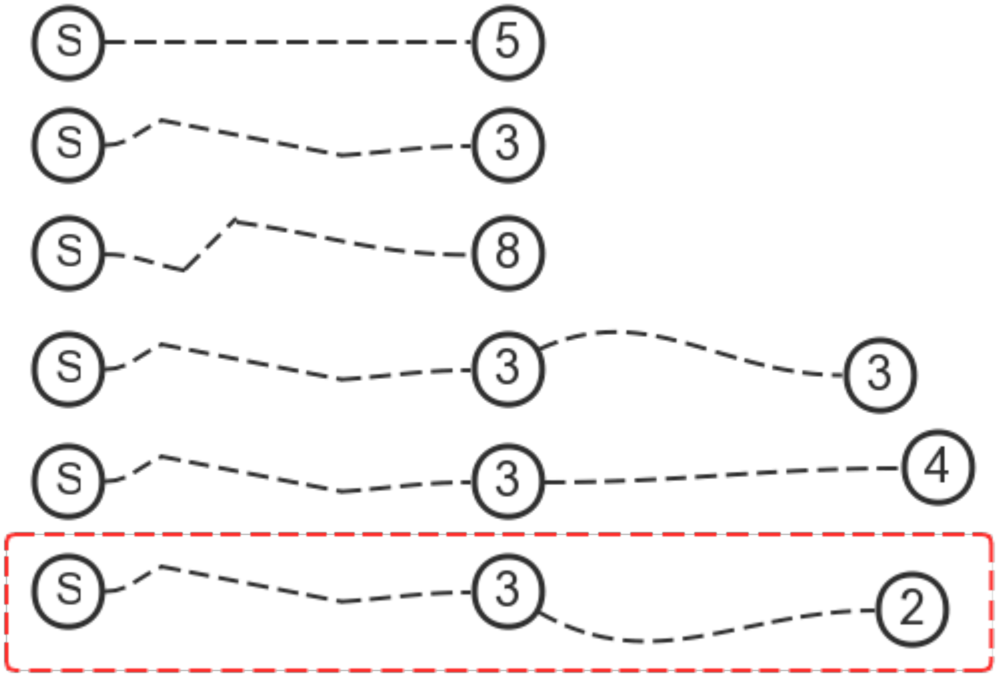
- Using a path pool
- Bidirectional search
- Anytime planning – Arvand*

Path Pool

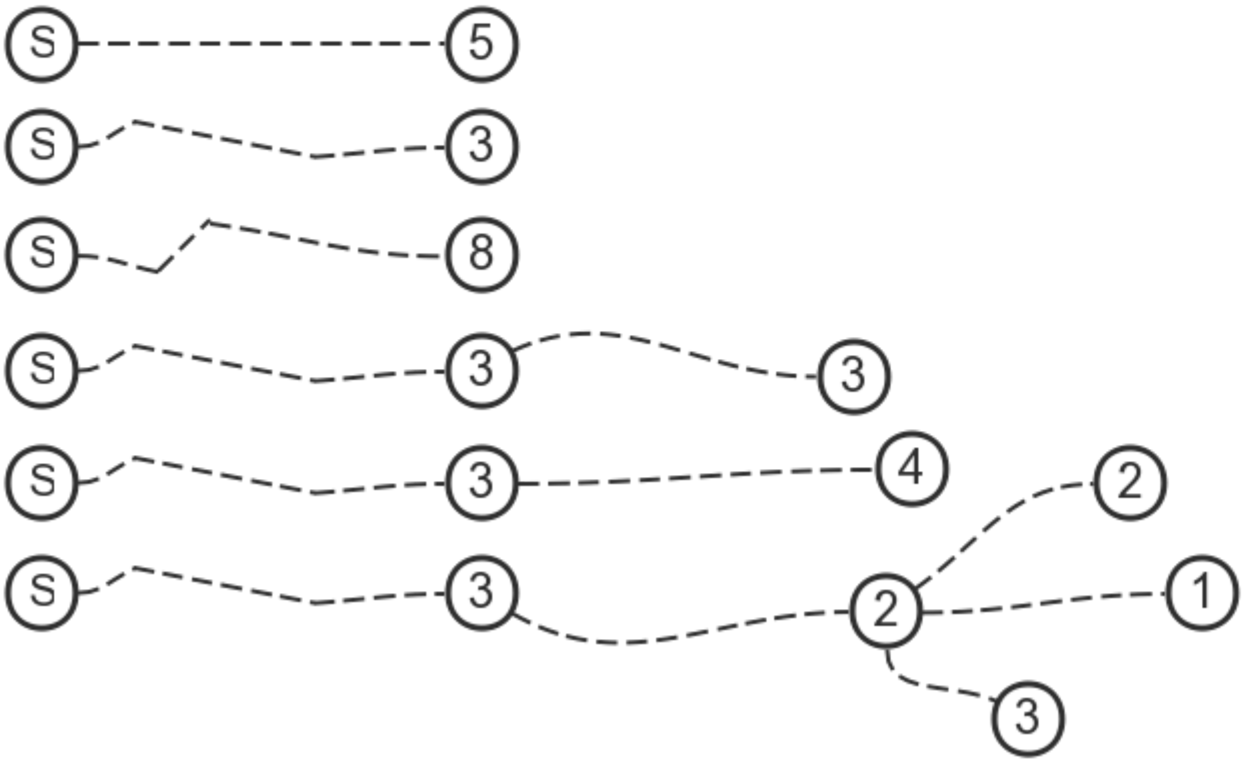
- Store a set of up to N random walks
- Utilize them for improving later searches
- Empty pool at global (re-)start
- Add/replace n < N paths at each time
  - Example: Pool size N = 6, n = 3

Path Selection

Pick path p with minimum *h*-value from pool

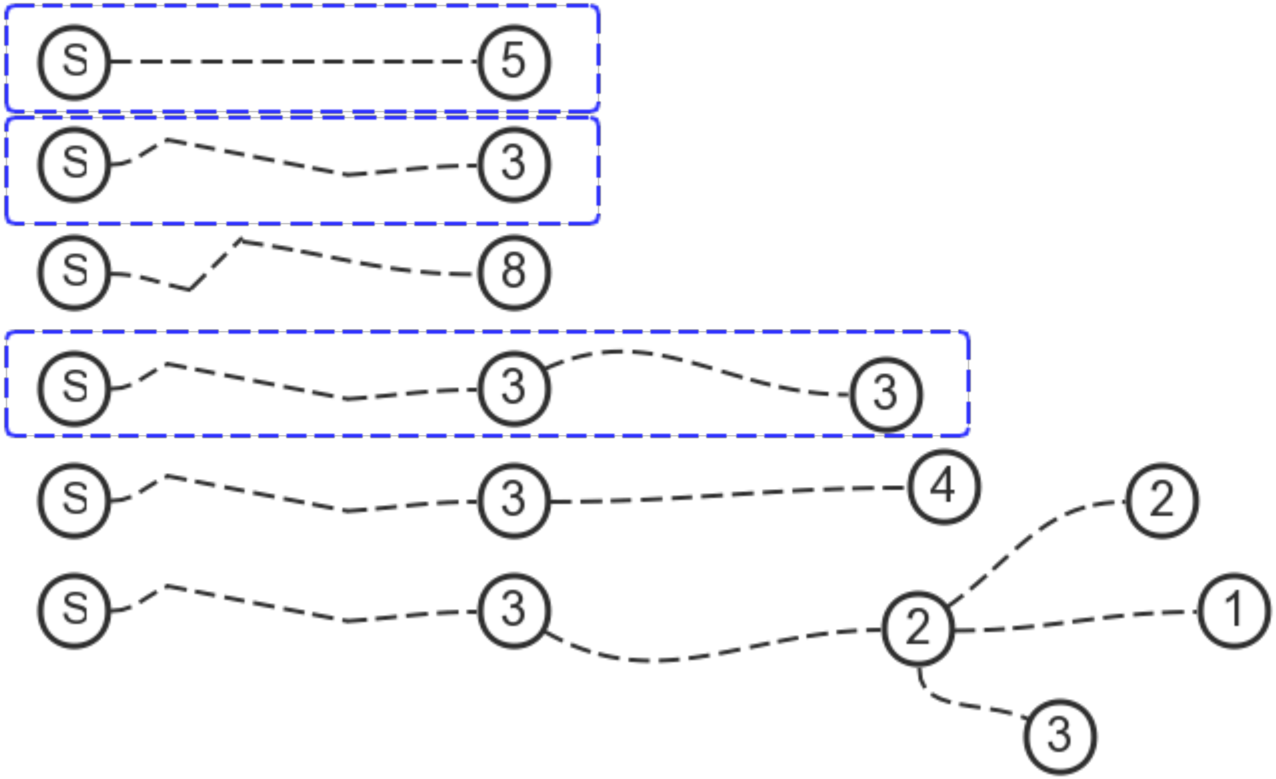


Path Expansion

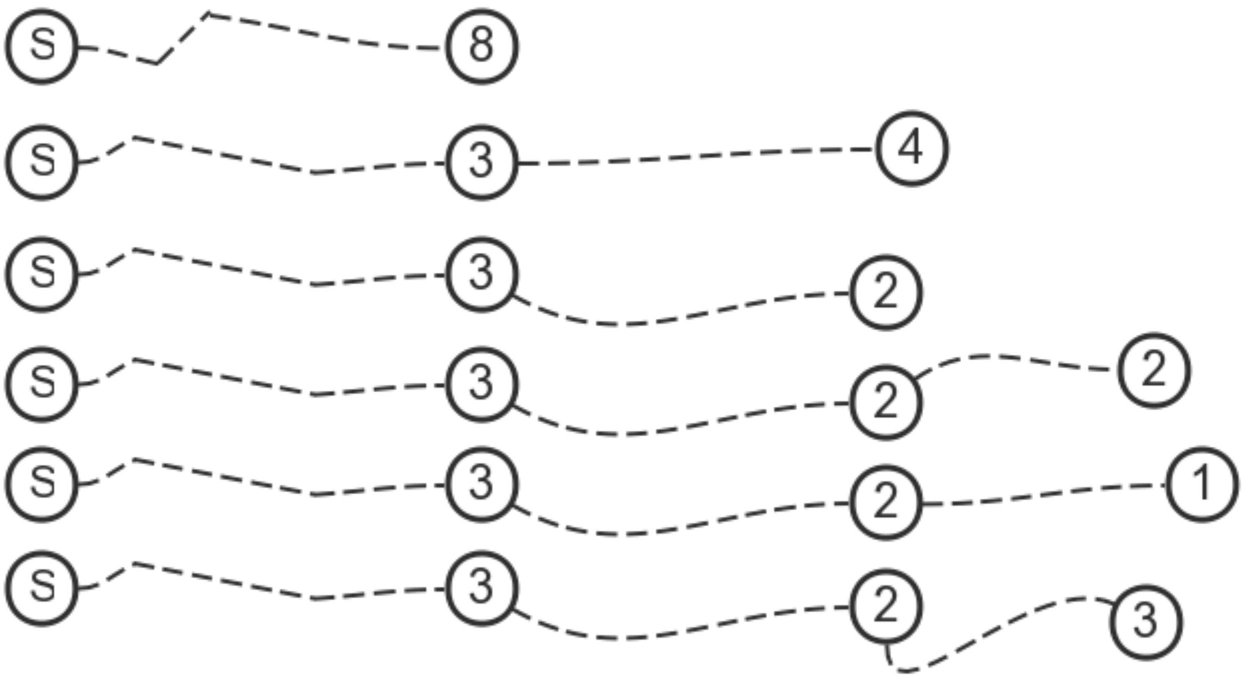


Choose Paths to be Replaced

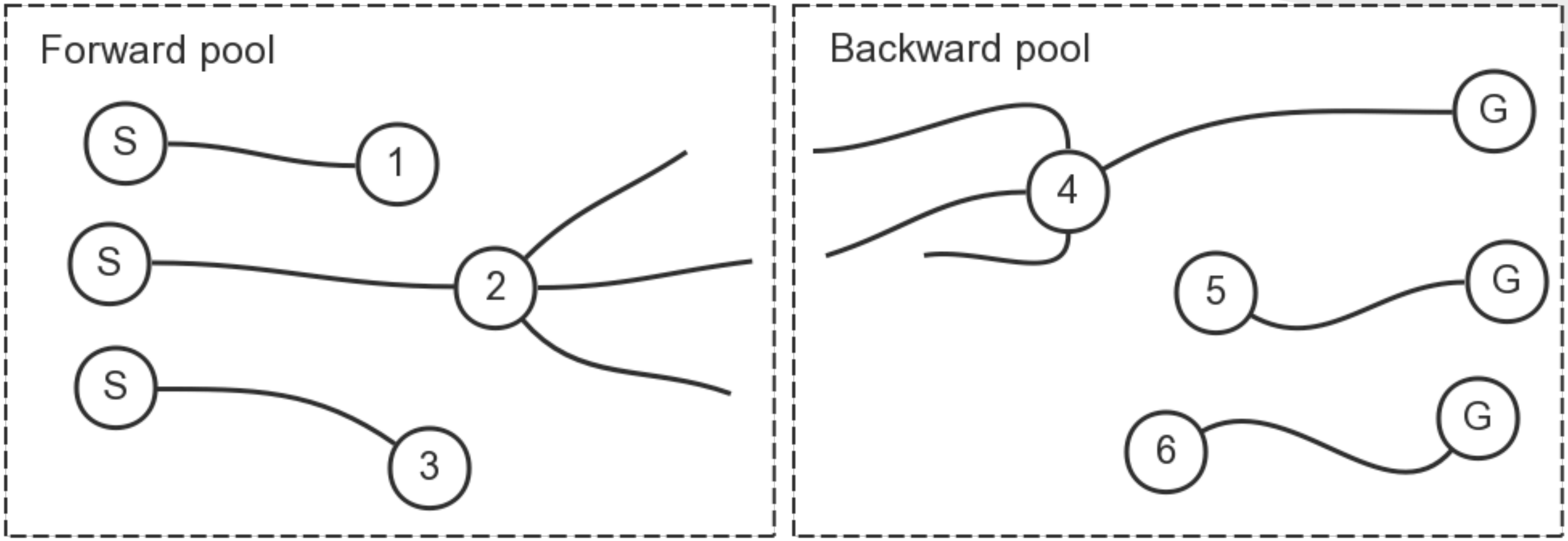
- Randomly choose n paths



Add New Paths to Pool



Bidirectional Arvand



- Alternate directions
- Choose the pair of endpoints that are closest, extend one of them, use the other as the goal.

Anytime Planning

- Most motion planners stop after they find the first valid plan is found.
- Anytime planning: restart and keep searching to find a better plan.

Implementation

- Continuous Arvand is built on top of Open Motion Planning Library (OMPL)
- Uses many OMPL primitives
  - pre-defined state space
  - state sampler
  - distance function
  - plan simplifier

Continuous Arvand Variants

Arvand_fixed	Constant parameters for walk length, number of walk...
Arvand_extend	Initial walk length = 10, doubled after every 100 walks
Arvand2	Number of walks = 1, restarting rate r = 0.01
Arvand2_AGR	Restart search when the last *t* walks did not lower heuristic, *t* is calculated adaptively
BArvand	Bidirectional Arvand
Arvand*	Find a best plan within the time limit

Experiments - Setup

- 5+1 other planners from OMPL:
  - KPIECE, EST, PDST, RRT, PRM
  - Optimizing planner RRT*, compared with Arvand*

- 13 motion planning problems from OMPL:
  - Maze, Barriers, Abstract, Apartment, BugTrap, Alpha, RandomPolygons, UniqueSolutionMaze, Cubicles, Pipedream, Easy, Home and Spirelli

Plan Length (Maze)

Planner	Path length	Simplified path length
KPIECE	285.35	149.64
EST	189.72	118.11
PDST	195.17	117.50
RRT	152.16	125.07
PRM	134.95	116.70
Arvand_fixed	**120.68**	**88.72**
Arvand_extend	187.00	105.30
Arvand2	4,630.43	139.96
Arvand2_AGR	10,739.10	153.31
BArvand	364.63	108.33

Rank of Arvand Versions

Metric	Arvand _fixed	Arvand _extend	Arvand2	Arvand2 _AGR	BArvand
Best in Memory	5/13	2/13	1/13	0/13	2/13
Avg Rank Memory	**1.2/10**	2.0/10	3.5/10	5.2/10	4.7/10

Rank of Arvand Versions

Metric	Arvand _fixed	Arvand _extend	Arvand2	Arvand2 _AGR	BArvand
Best in Memory	5/13	2/13	1/13	0/13	2/13
Avg Rank Memory	**1.2/10**	2.0/10	3.5/10	5.2/10	4.7/10
Best in Path Length	2/13	1/13	0/13	0/13	3/13
Avg rank Path Length	**1.8/10**	4.2/10	5.6/10	5.4/10	4.1/10

Rank of Arvand Versions

Metric	Arvand _fixed	Arvand _extend	Arvand2	Arvand2 _AGR	BArvand
Best in Memory	5/13	2/13	1/13	0/13	2/13
Avg Rank Memory	**1.2/10**	2.0/10	3.5/10	5.2/10	4.7/10
Best in Path Length	2/13	1/13	0/13	0/13	3/13
Avg rank Path Length	**1.8/10**	4.2/10	5.6/10	5.4/10	4.1/10
Best in Time	0/13	0/13	0/13	1/13	1/13
Avg Rank Time	8.0/10	8.5/10	5.8/10	**5.2/10**	5.5/10

Best Arvand vs Top 3 Other

Metric	Best Arvand	RRT	PRM	KPIECE	Other
Best in Memory	**10/13**	1/13	0/13	1/13	1/13
Avg Rank Memory	**1.3/10**	5.2/10	6.9/10	5.5/10	6.8/10

Best Arvand vs Top 3 Other

Metric	Best Arvand	RRT	PRM	KPIECE	Other
Best in Memory	**10/13**	1/13	0/13	1/13	1/13
Avg Rank Memory	**1.3/10**	5.2/10	6.9/10	5.5/10	6.8/10
Best in Path Length	**6/13**	1/13	**6/13**	0/13	0/13
Avg rank Path Length	**1.8/10**	4.9/10	3.1/10	7.8/10	5.5/10

Best Arvand vs Top 3 Other

Metric	Best Arvand	RRT	PRM	KPIECE	Other
Best in Memory	**10/13**	1/13	0/13	1/13	1/13
Avg Rank Memory	**1.3/10**	5.2/10	6.9/10	5.5/10	6.8/10
Best in Path Length	**6/13**	1/13	**6/13**	0/13	0/13
Avg rank Path Length	**1.8/10**	4.9/10	3.1/10	7.8/10	5.5/10
Best in Time	2/13	**5/13**	0/13	3/13	3/13
Avg Rank Time	3.5/10	**2.4/10**	5.9/10	3.0/10	3.9/10

Four Categories of Problems

- **Easy** (solvable in ~1 second by most planners)
  - Maze, BugTrap, RandomPolygons, Easy
- **Intermediate**
  - Alpha, Barriers, Apartment
- **Intermediate with long detour**
  - UniqueSolutionMaze, Cubicles, Pipedream_ring, Abstract
- **Hard** (avg. time > 1 minute, some time out)
  - Home, Spirelli

Results - Qualitative

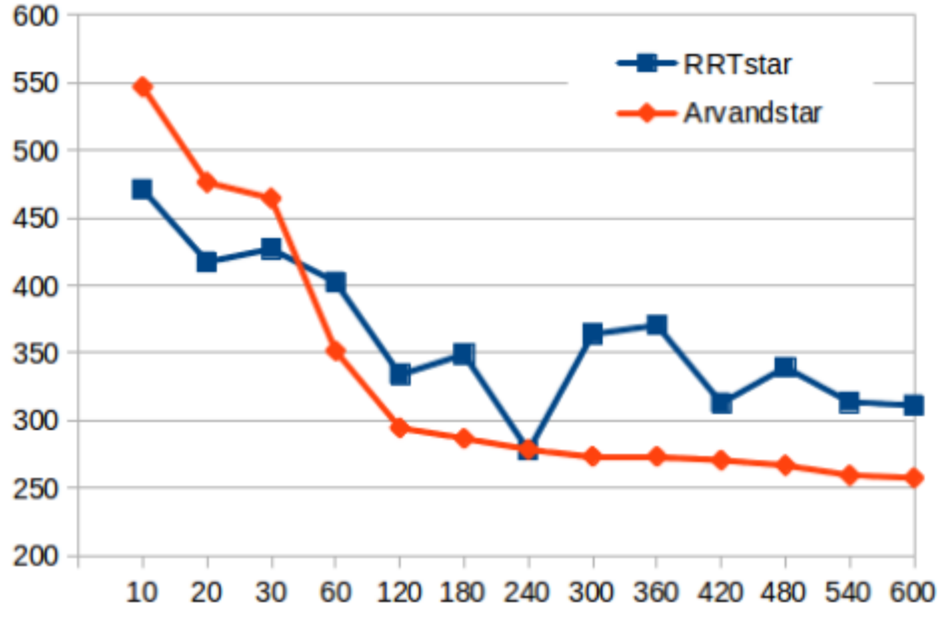
- Continuous Arvand produces competitive short solutions for Easy problems in a short time.
- BArvand outperforms all other planners in the intermediate problems Alpha and Barriers.
- Poor performance for problems requiring long detours.
- Arvand2_AGR and BArvand can solve the hard problem Spirelli, other variants time out.

Experiments - Summary

- Overall, the family of continuous Arvand planners are competitive
- Can outperform other planners in some motion planning problems
- Usually use much less memory
- Do not perform well when long detours are required

Anytime Plan Length

Plan length as a function of time for Arvand* and RRT*



- Problem: Alpha
- Data averaged over 10 runs

Future Work

- Try further MRW techniques from classical planning
  - On-Path Search Continuation
  - Smart Restarts
  - Adaptive local restarting
  - Evaluation of intermediate states along the walk
- Investigate other ways of using memory to speed up MRW, improve its plan quality, etc.
- Create a Portfolio Motion Planner

Conclusions

- Applied MRW approach to motion planning
- Works well for problems that do not require long detours
- Uses much less memory than other planners
- Highly configurable
- Different strengths and weaknesses compared to previous methods, and among our variations