Resource-Constrained Planning: A Monte-Carlo Random Walk Approach

Hootan Nakhost[1] Jörg Hoffmann[2] Martin Müller[1]

[1]University of Alberta
[2]Saarland University

Reasoning about Resources

- Examples of limited resources
  - Fuel, energy, money, time
- Model: not replenishable resources
  - Initial supply
  - Some actions consume resources

Limitation of the Current Methods

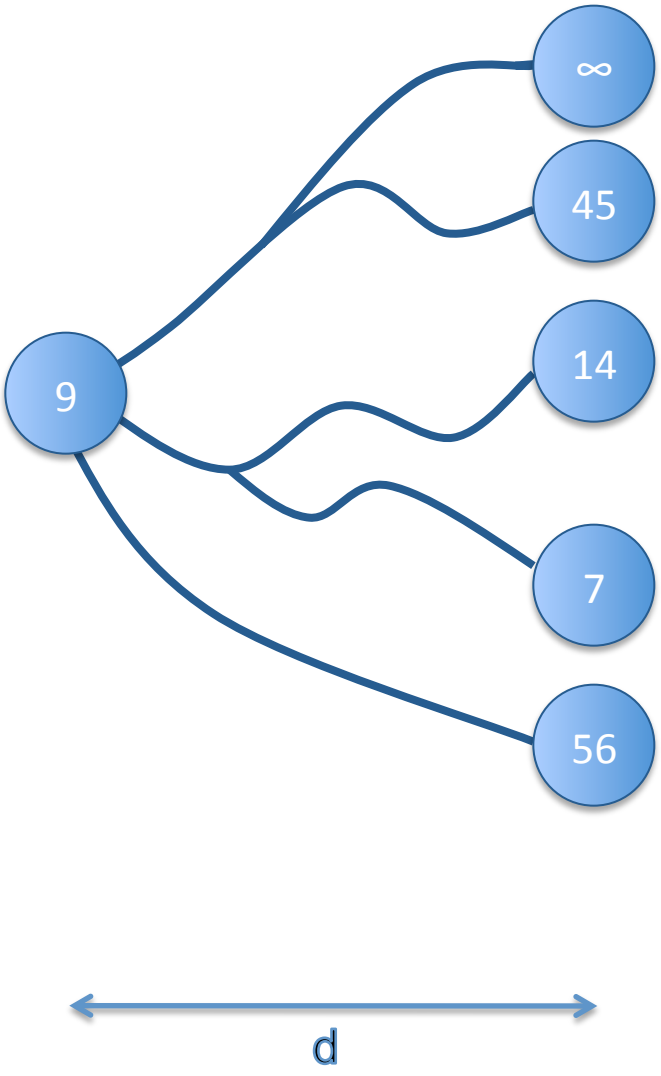
- Relaxation heuristics do not model resource consumption at all
- Greedy search algorithms add more problems

How to improve RCP?

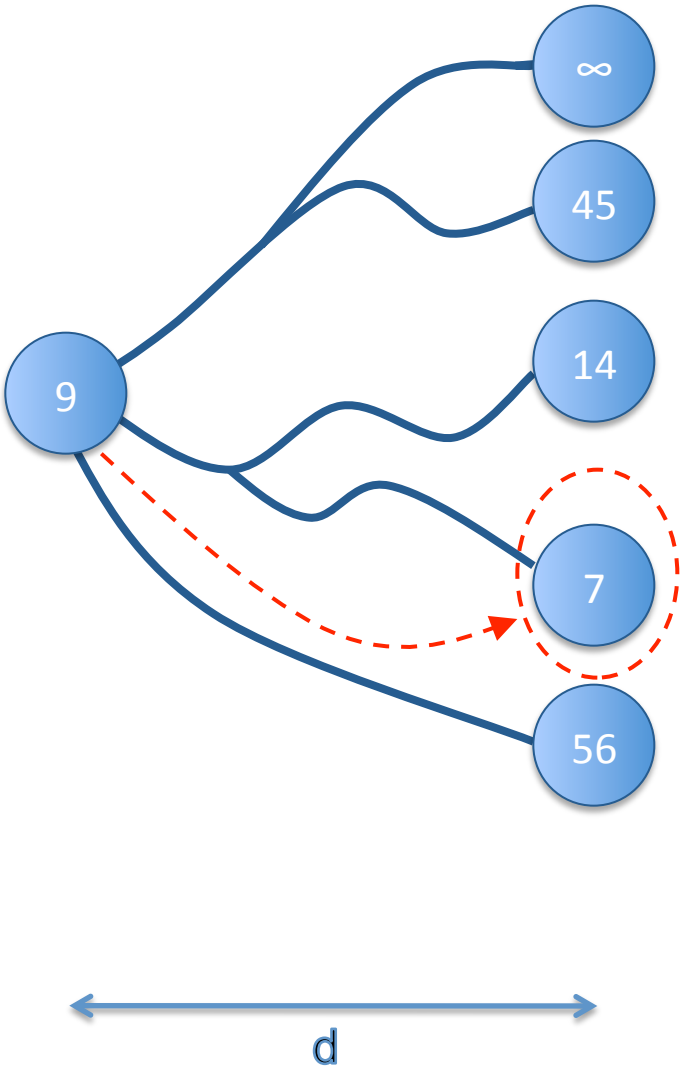
- Build new heuristic functions
  - LPRPG [Coles et. al. ICAPS'08]
- Our approach: design new better search algorithms
  - Focusing on local search
  - Extend RW planner Arvand

Review: Random Walks in Arvand

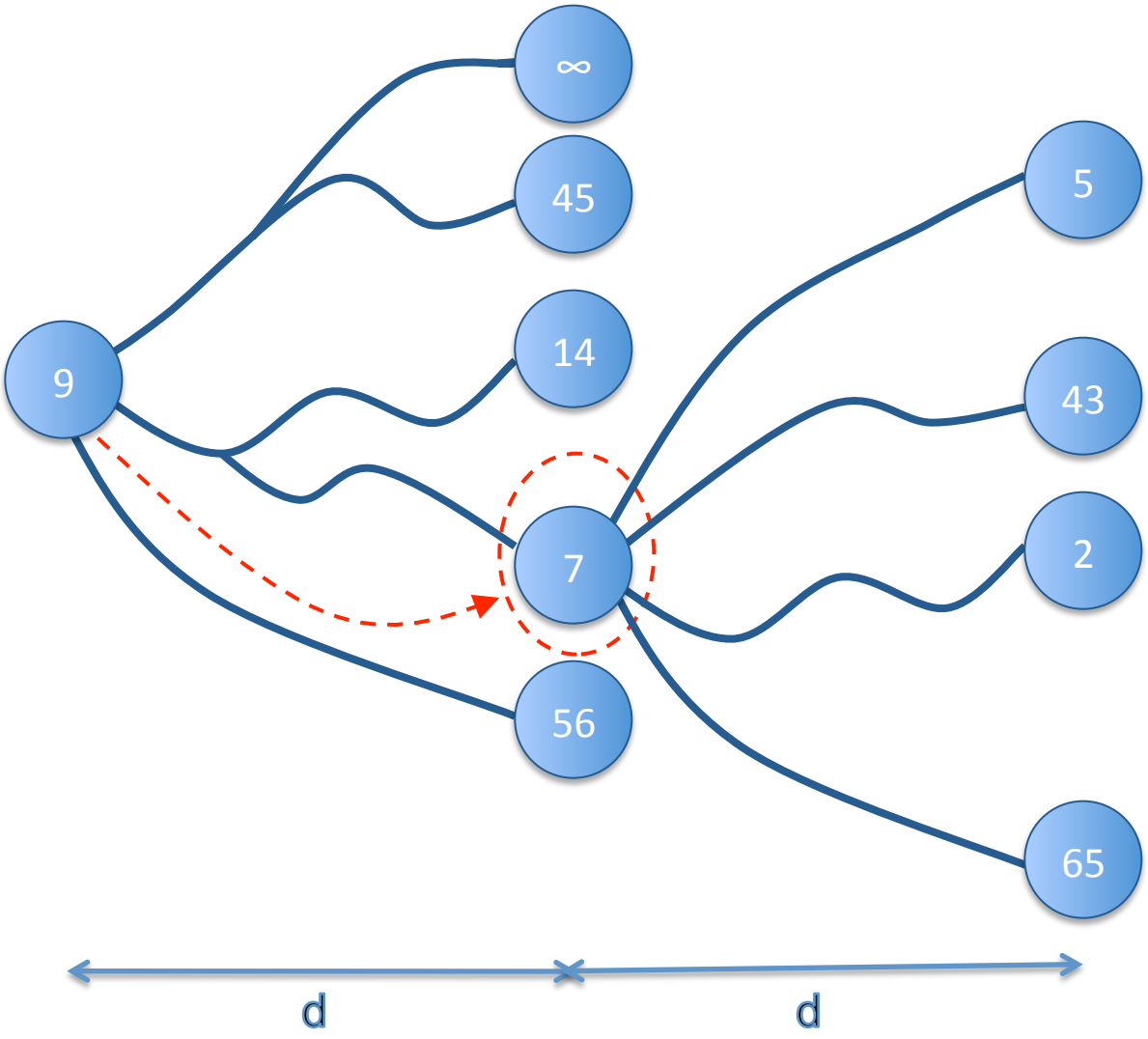
Review: Random Walks in Arvand



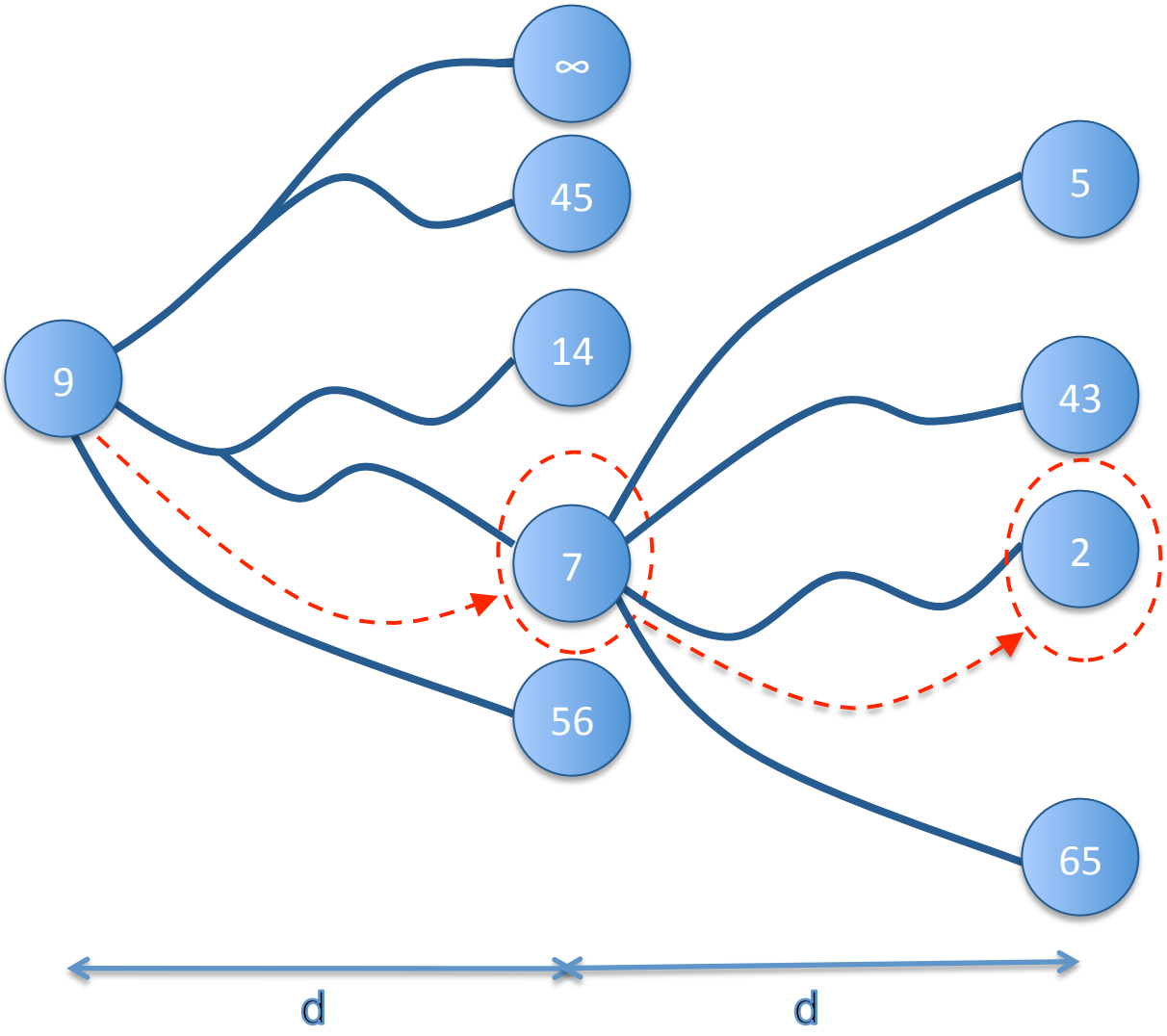
Review: Random Walks in Arvand



Review: Random Walks in Arvand



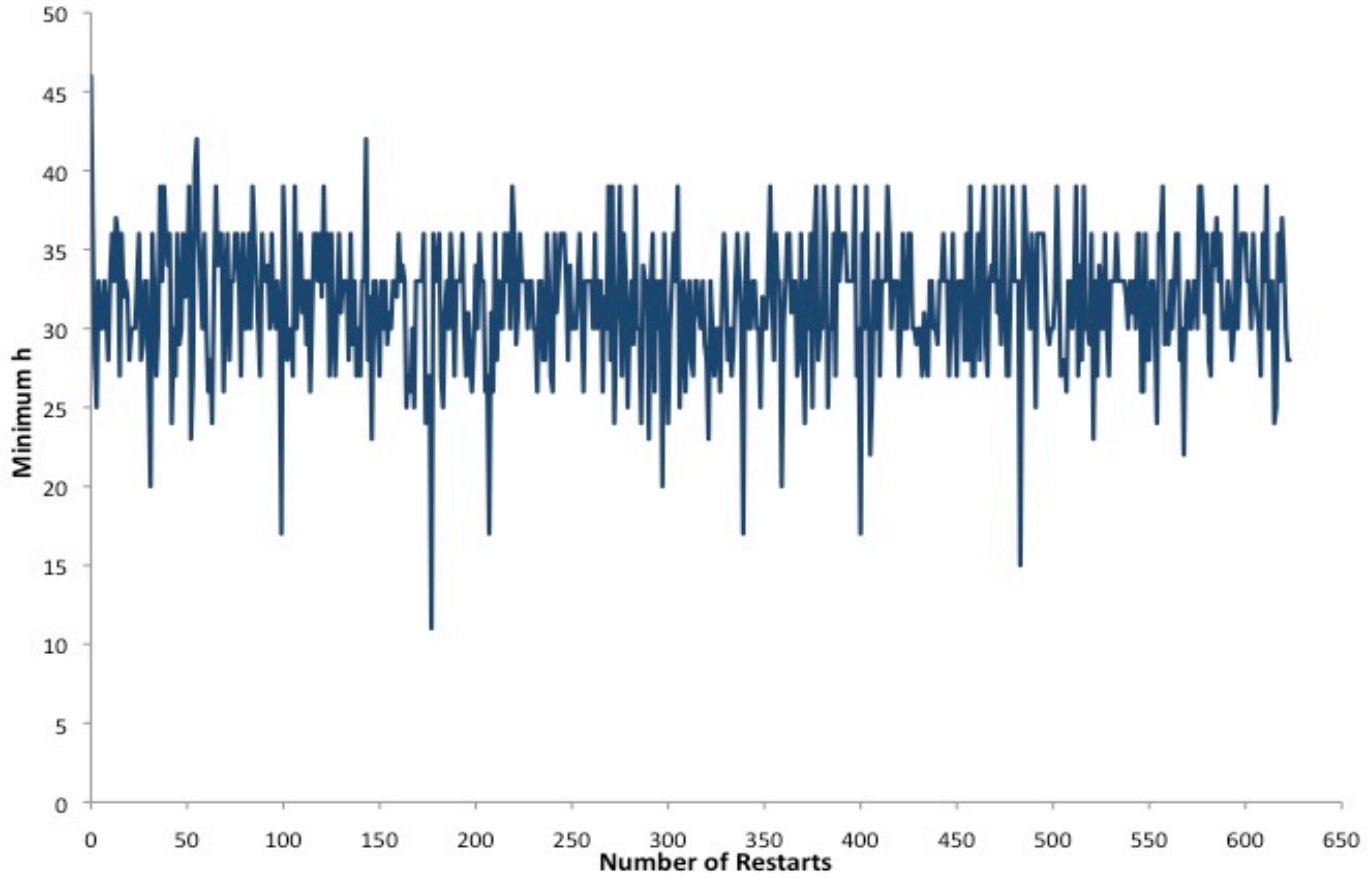
Review: Random Walks in Arvand



Improvements to Arvand for RCP

- Smart Restarting (SR)
- On-path Search Continuation (OPSC)

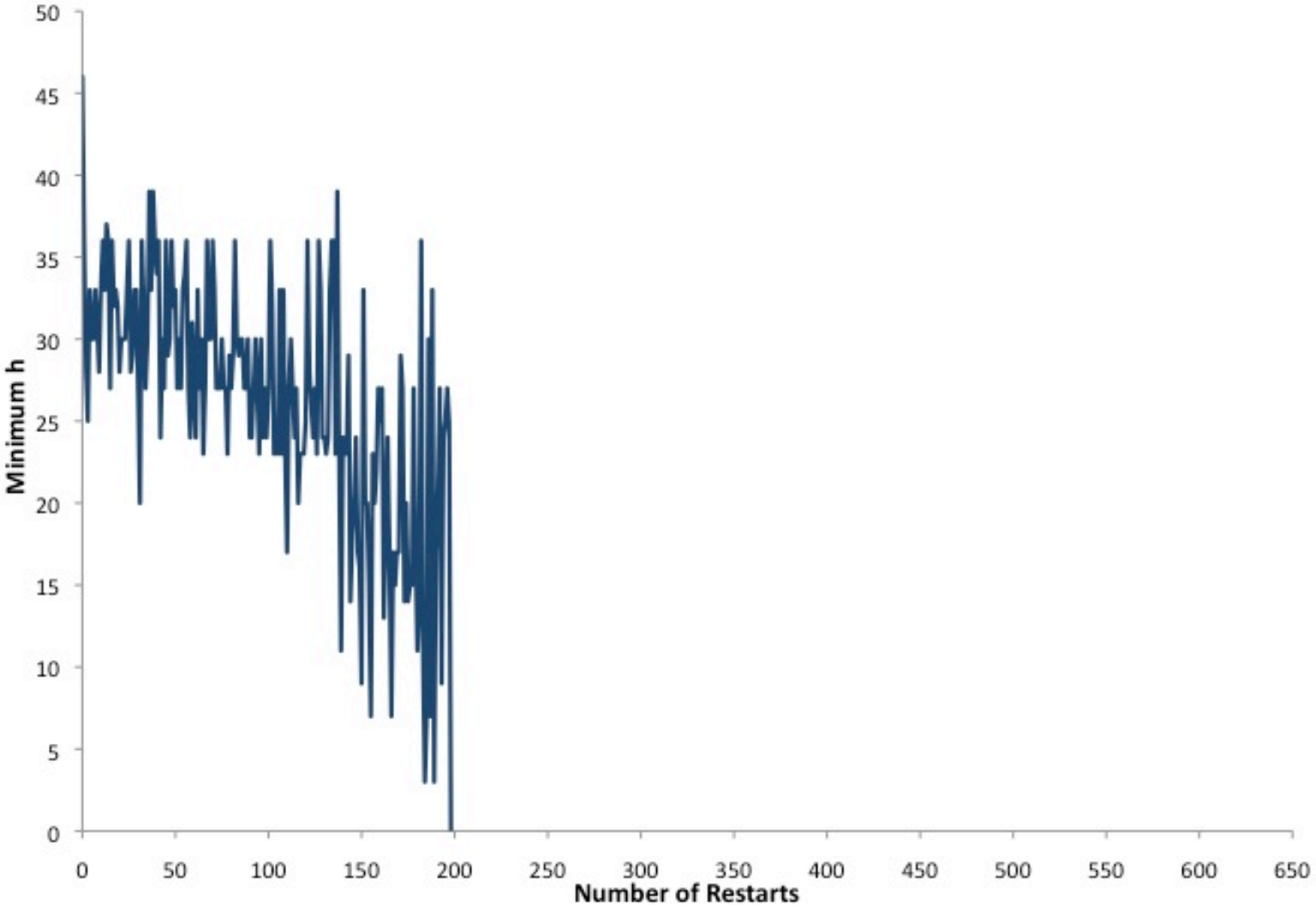
Basic Restarting in an Example



Smart Restarting

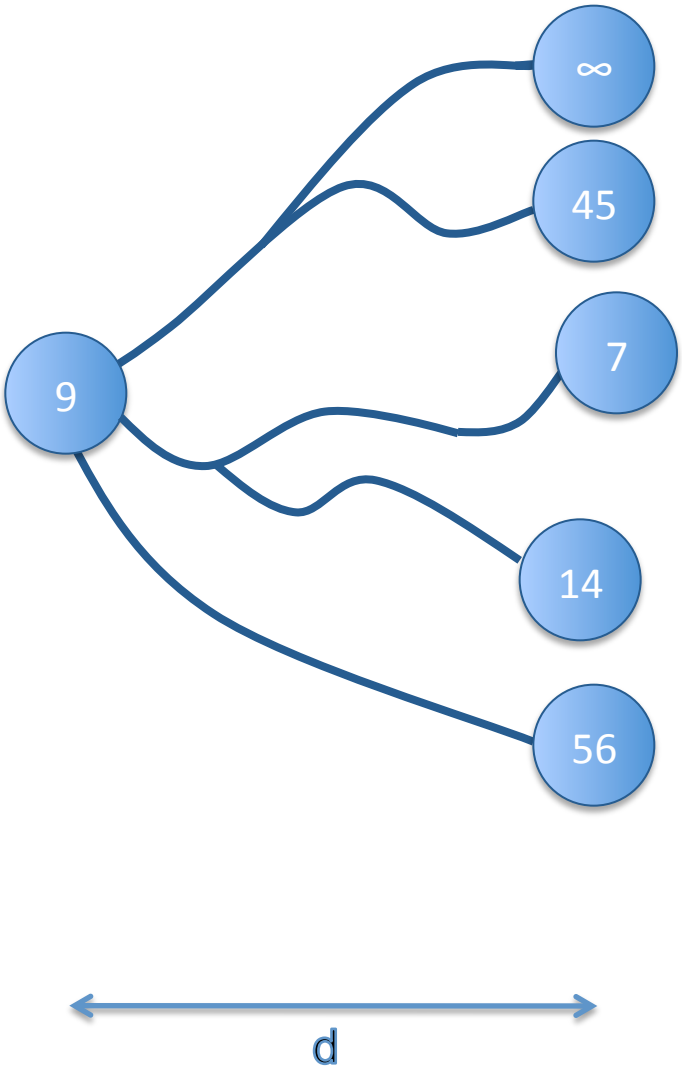
- Maintain a pool of most promising episodes performed so far
- When an episode gets stuck, instead of always restarting from the initial state, restart from a state visited in such an episode
- Parameters:
  - Pool size
  - When to start smart restarting

Smart Restarting in an Example

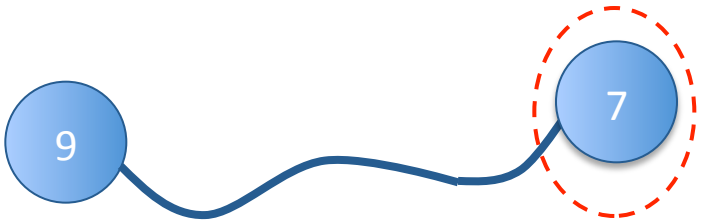


On-Path-Search Continuation

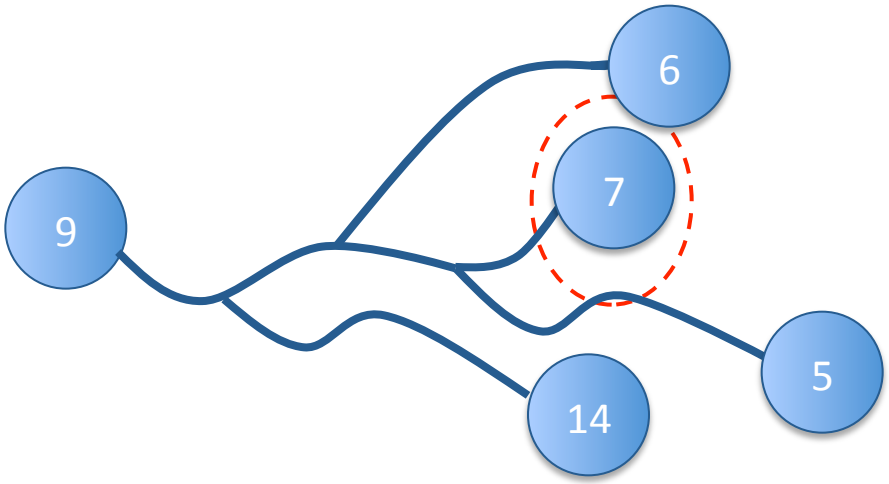
On-Path-Search Continuation



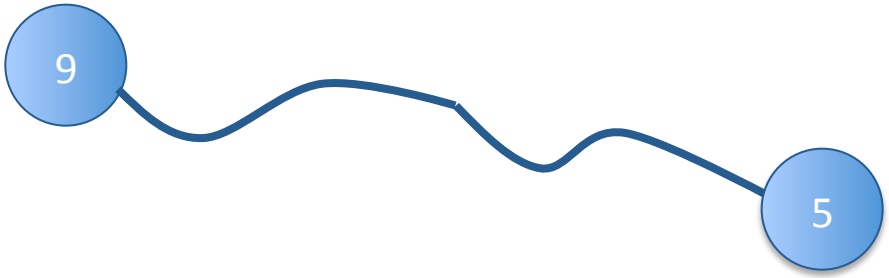
On-Path-Search Continuation



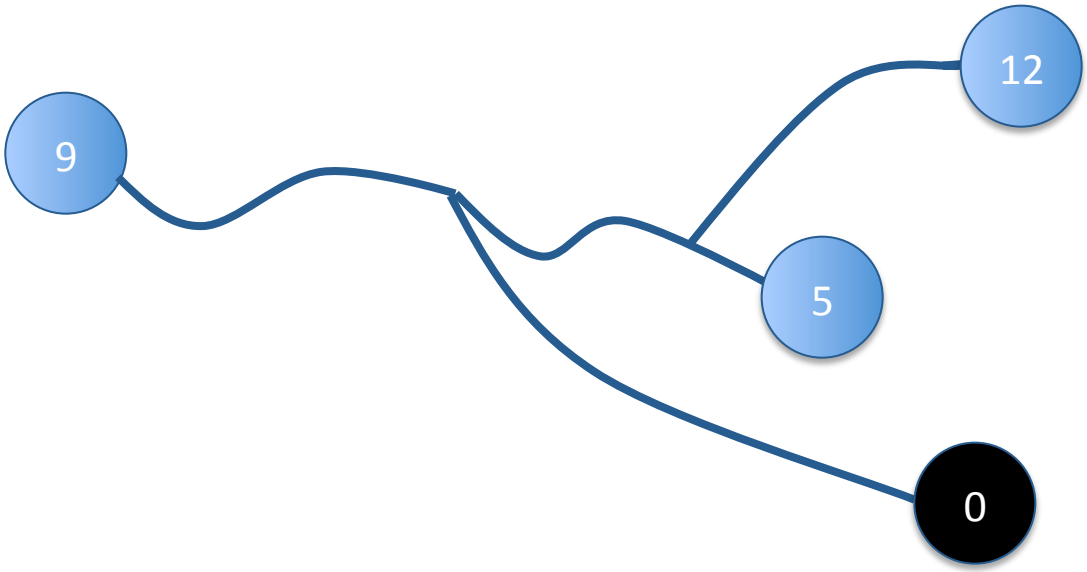
On-Path-Search Continuation



On-Path-Search Continuation



On-Path-Search Continuation



How to test RCP planners?

- The performance as a function of constrainedness
- Resource constrainedness C (Hoffmann et. al. IJCAI'07):

$$C = \frac{initial\ supply}{minimum\ need}$$

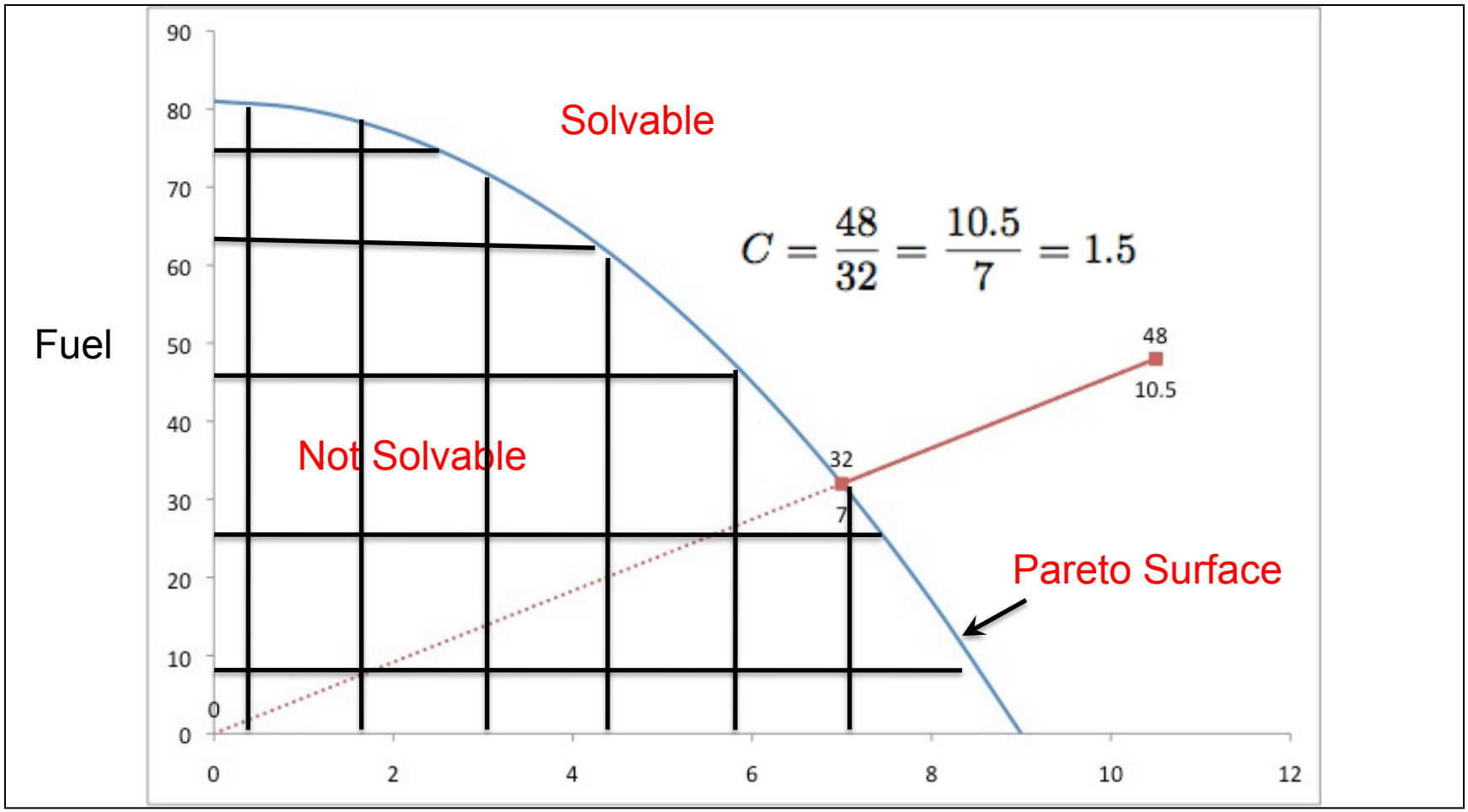
- The closer C is to 1, the more constrained is the problem

C for Multiple Resources

- The previous definition of C only works for problems with single resource

- New definition:
  - The largest factor by which we can downscale the initial resource supply without making the task unsolvable

Example


Fuel
Money
Solvable
Not Solvable
Pareto Surface
$C = \frac{48}{32} = \frac{10.5}{7} = 1.5$
48
10.5
32
7
0
90
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12

Experiments

- Extensive experiments on three RCP domains
  - NoMystery
  - Rovers
  - TPP
- 8 satisficing and 5 optimal planners were tested
  - Arvand, FD-AT1, FD-AT2, LAMA, FF, LPG, M, Mp, LPRPGP
  - Num-2-sat, LM-cut, Merge and Shrink, Selmax, FD-AT-OPT

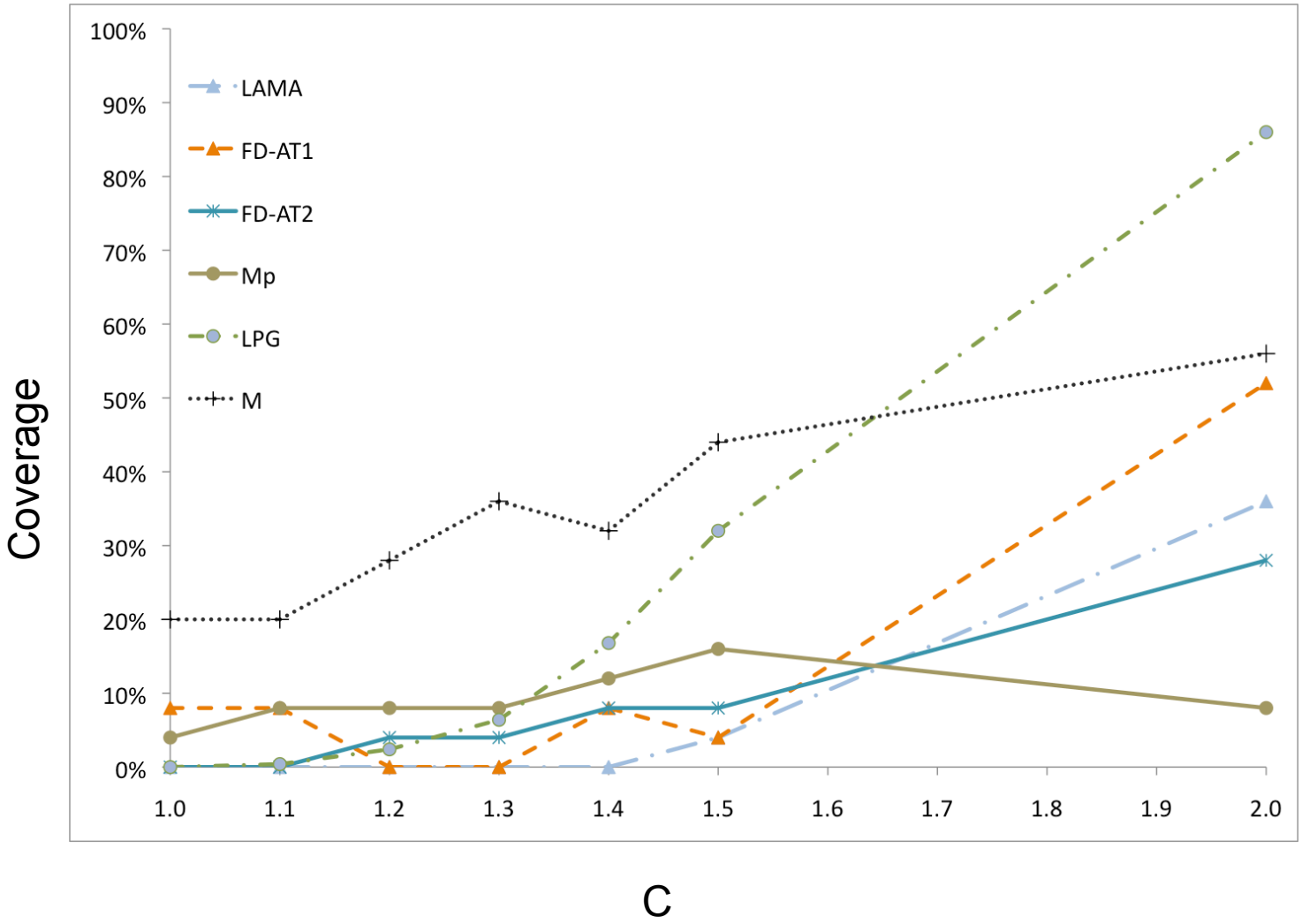
Encodings for RCP Problems

- Propositional
- Numerical
- Cost augmented
  - Only for single resource
- Costs with no hard constraints
  - Only for LAMA
- Preferences

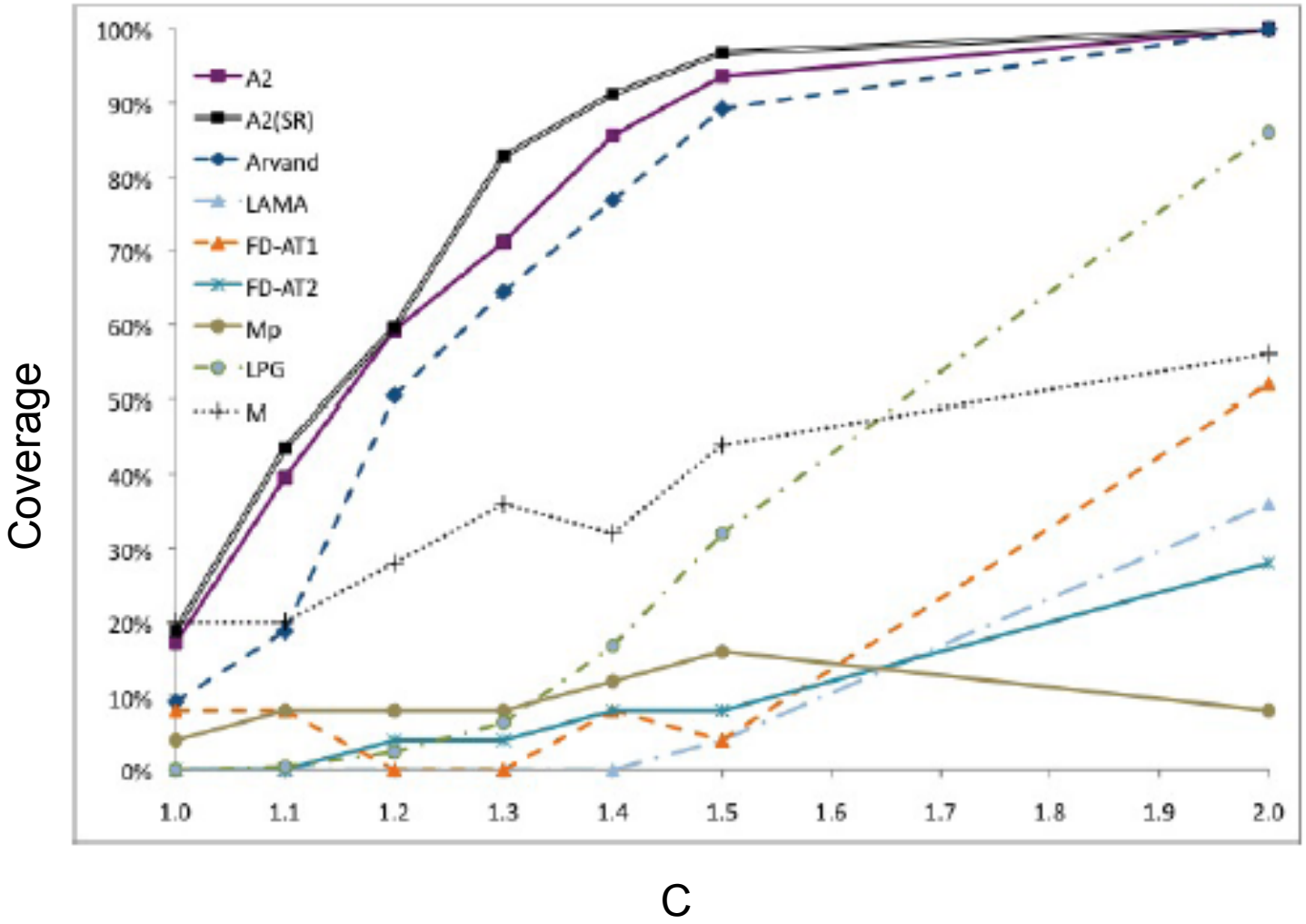
RCP Benchmarks:

- 450 problem instances
- Smaller set of base problems for C=1.0
- Other problems obtained by changing C={1.1, 1.2, 1.3, 1.4, 1.5, 2.0}

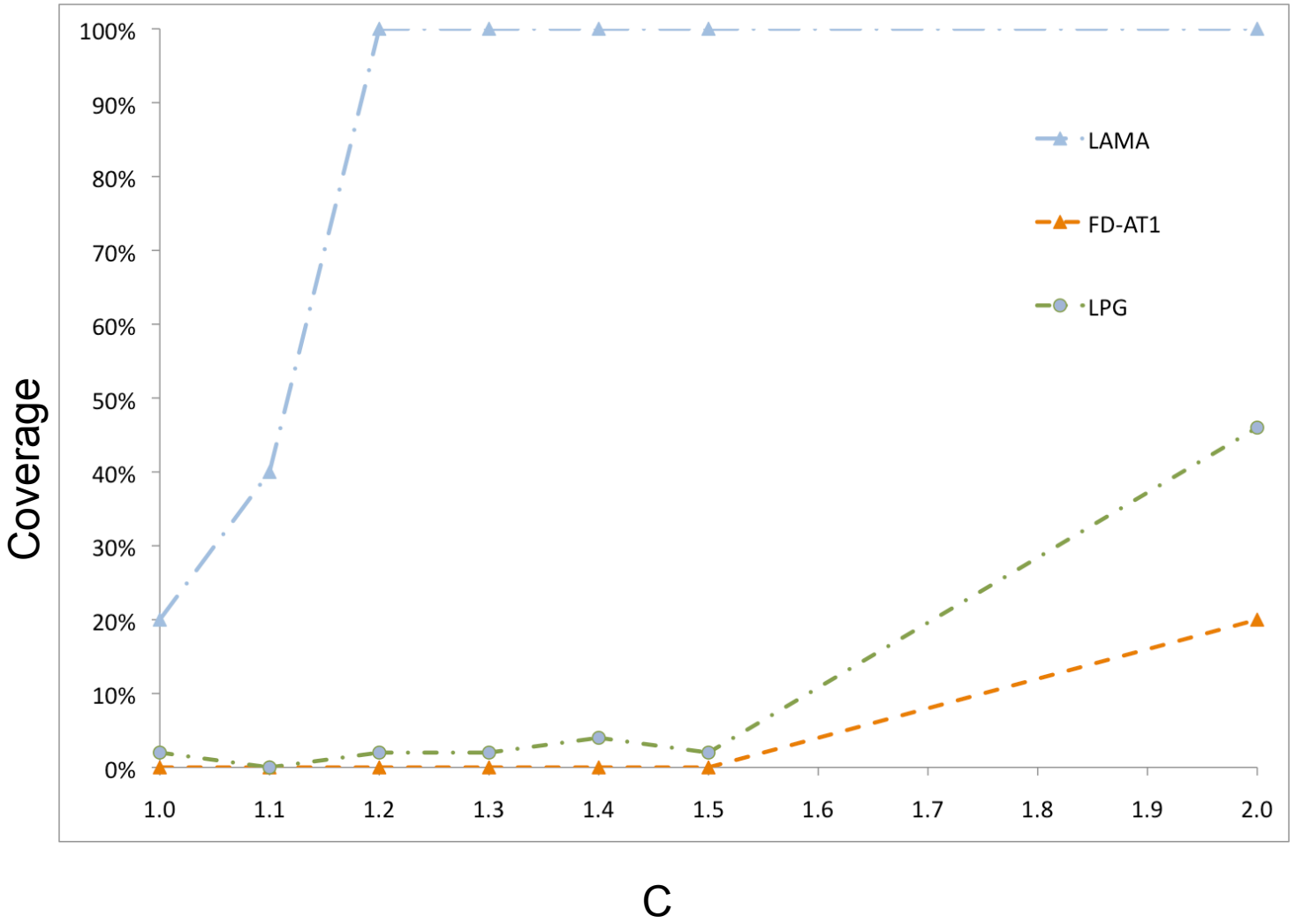
Results: Rovers, small



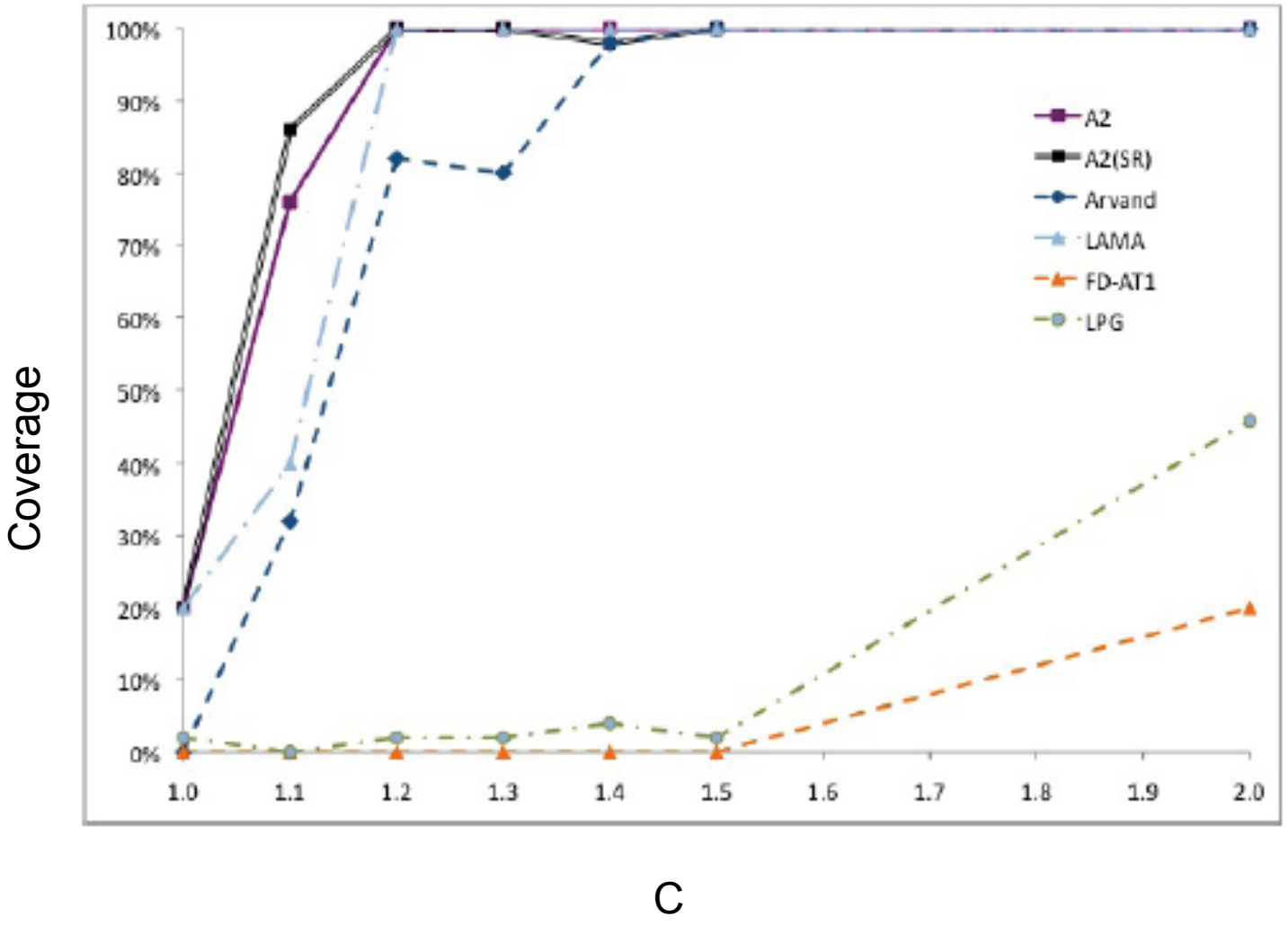
Results: Rovers, small



Results: Rovers, large

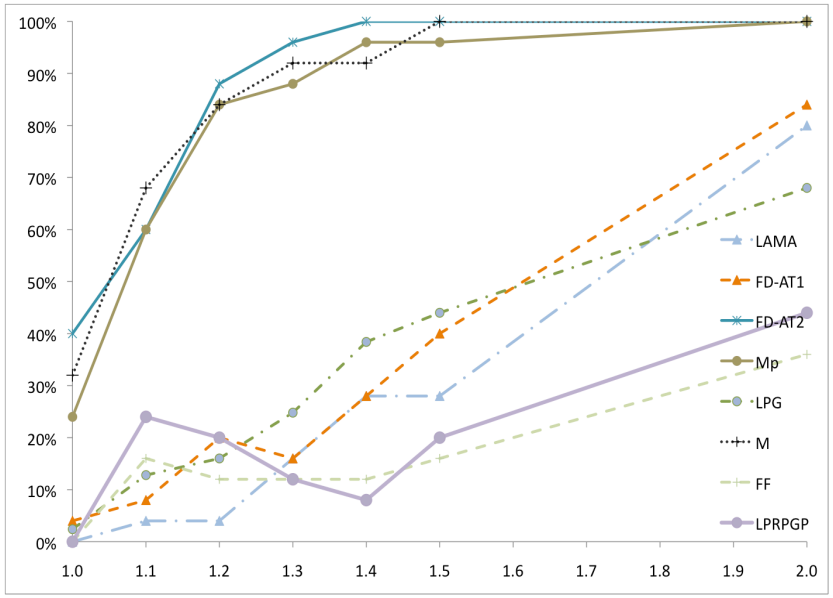


Results: Rovers, large

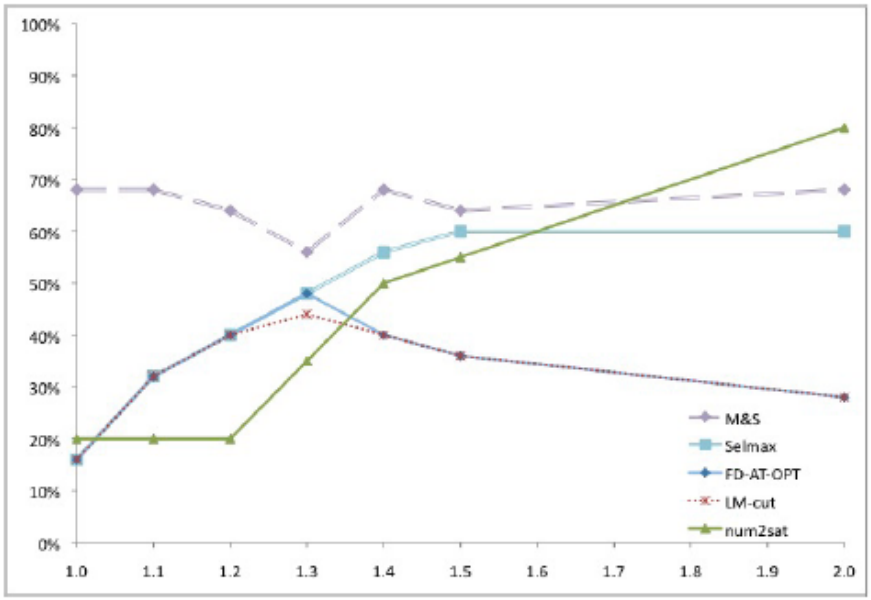


Results: NoMystery, small

Coverage

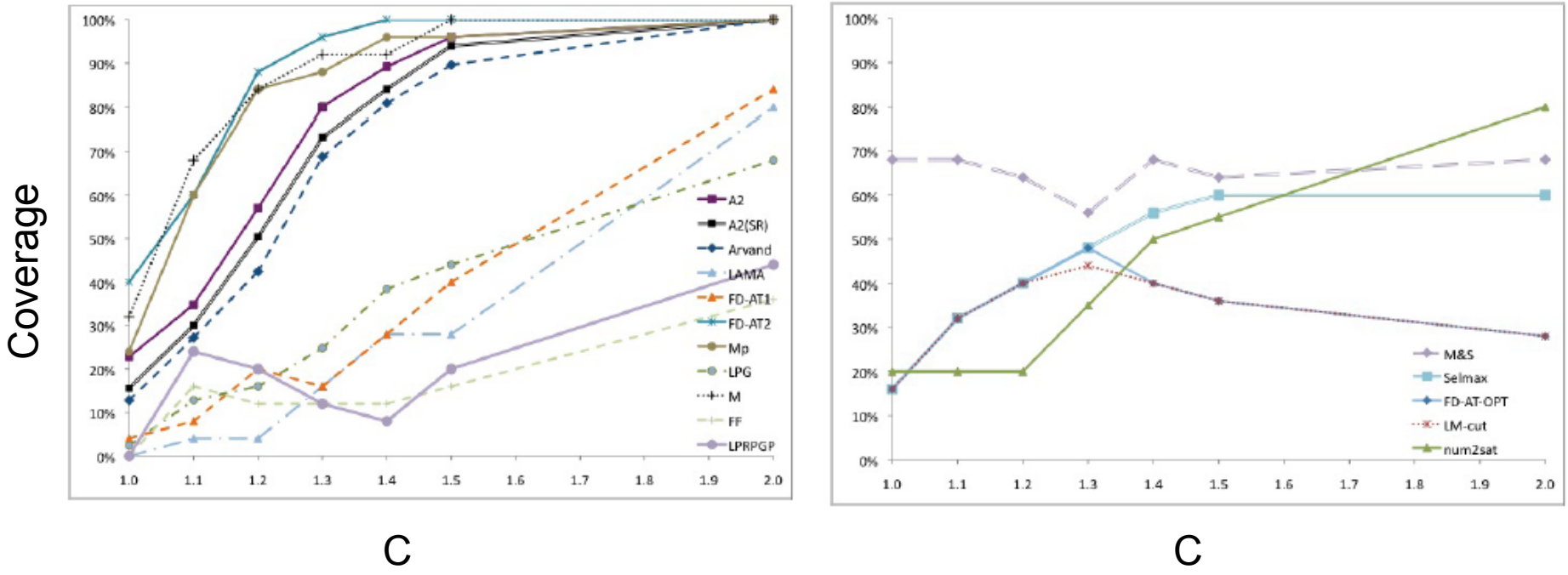


C

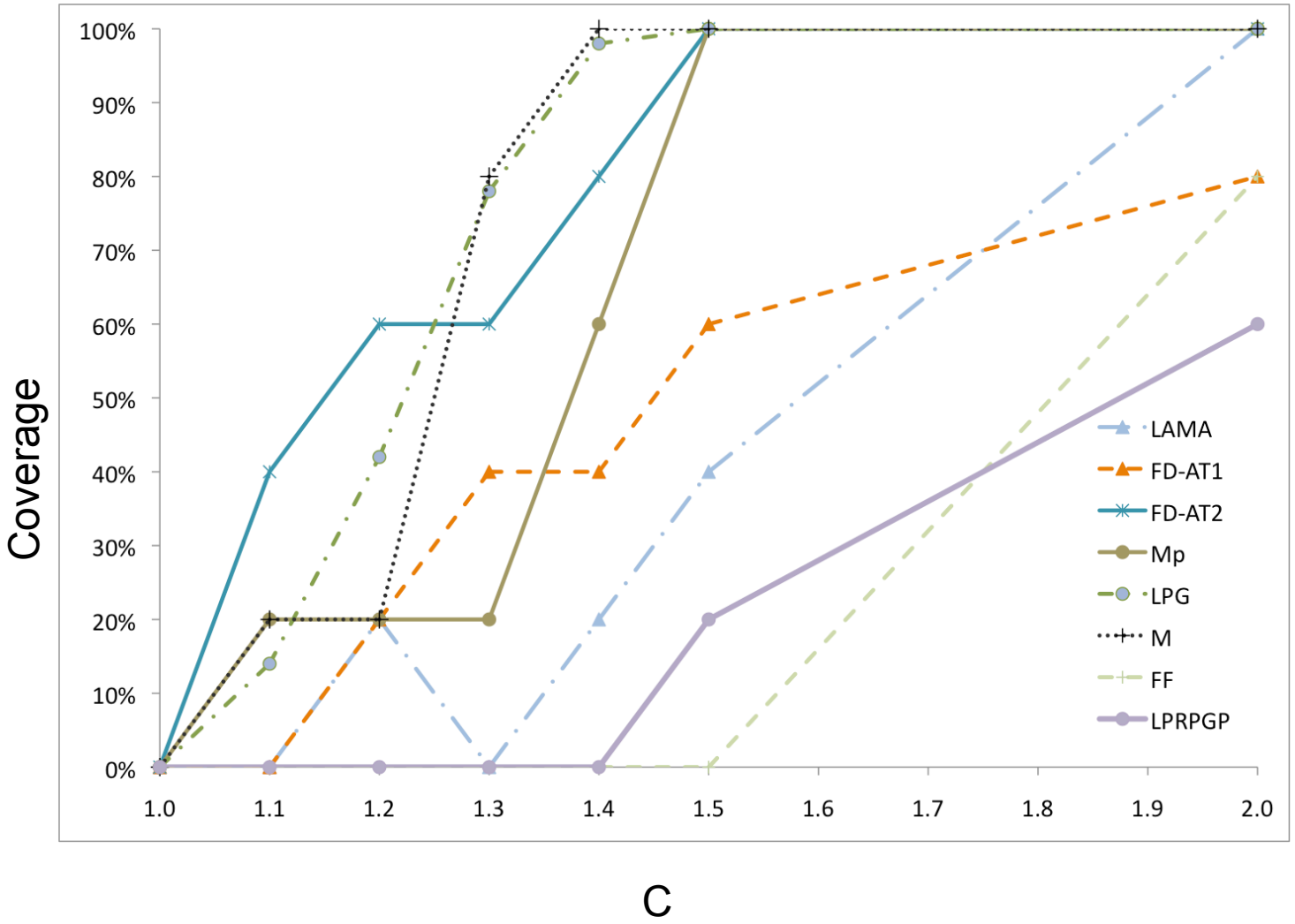


C

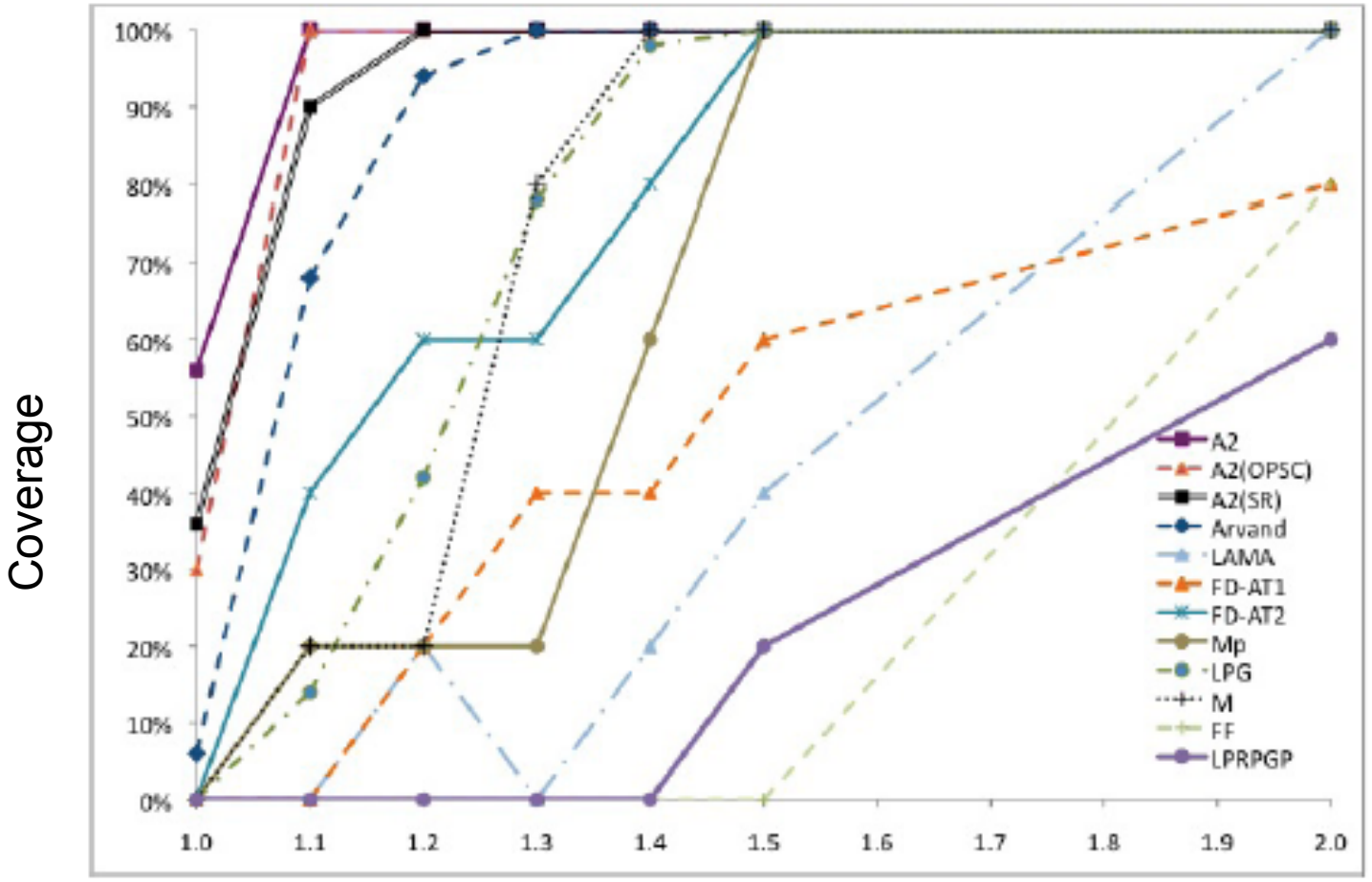
Results: NoMystery, small



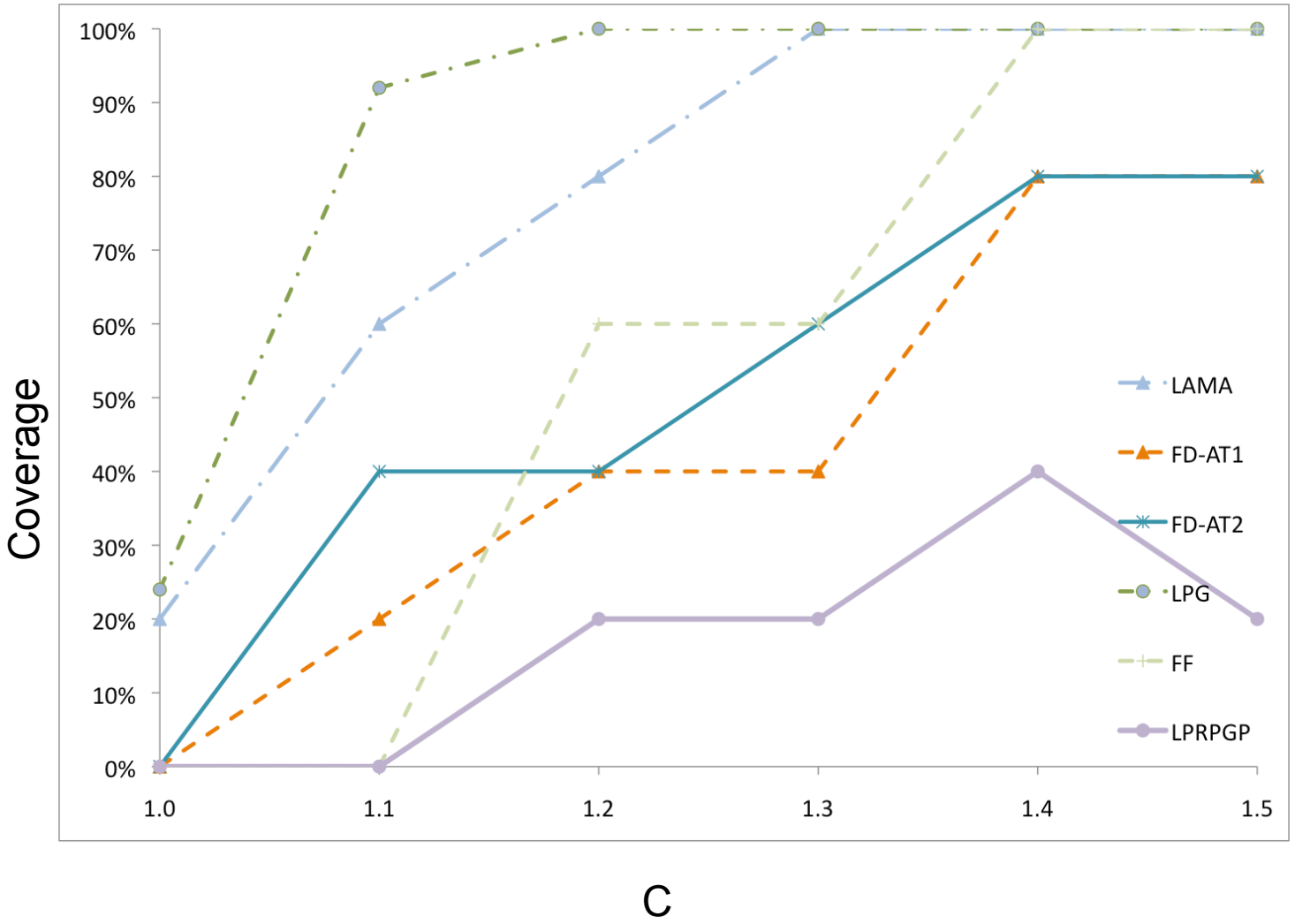
Results: NoMystery, Large



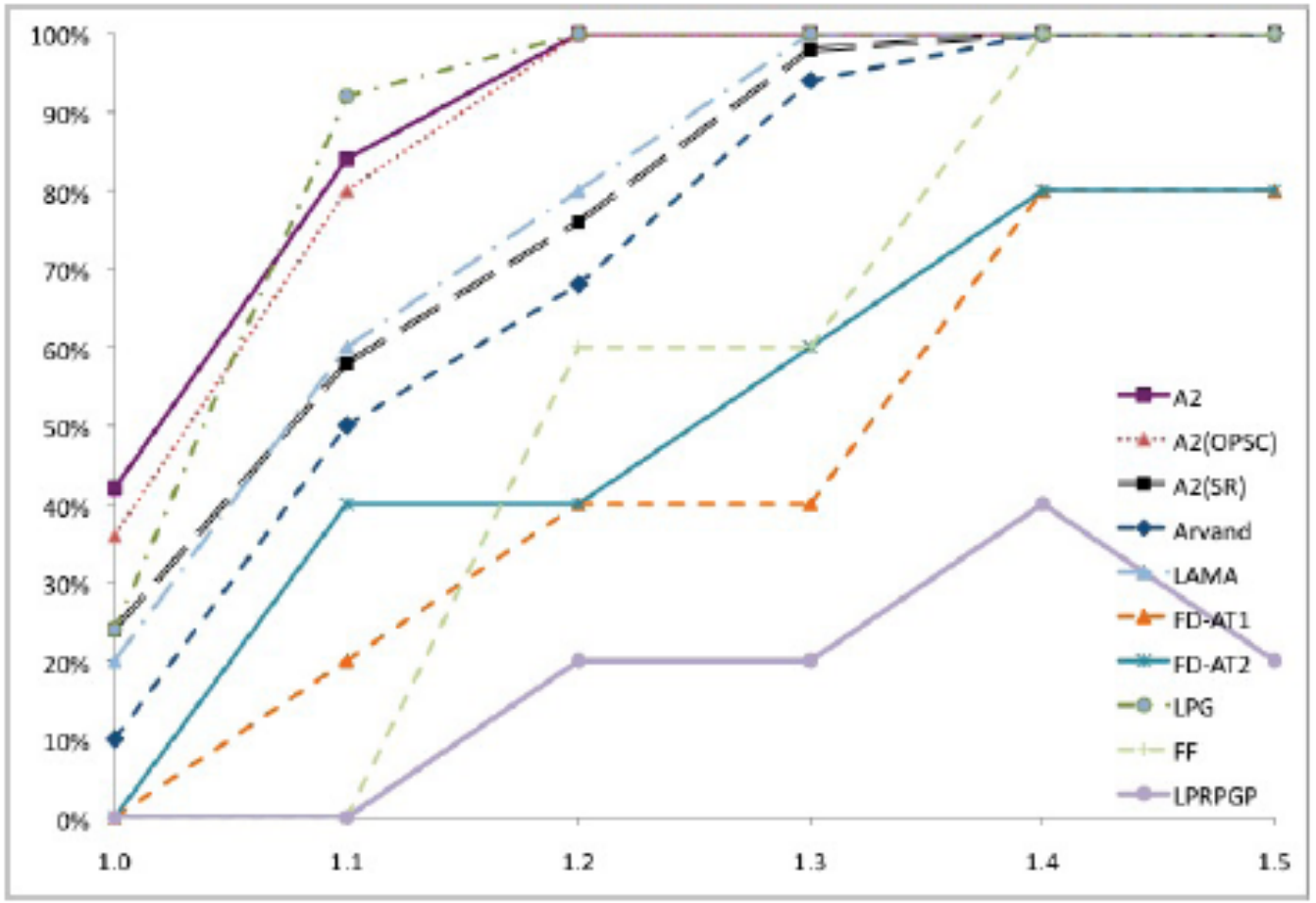
Results: NoMystery, Large



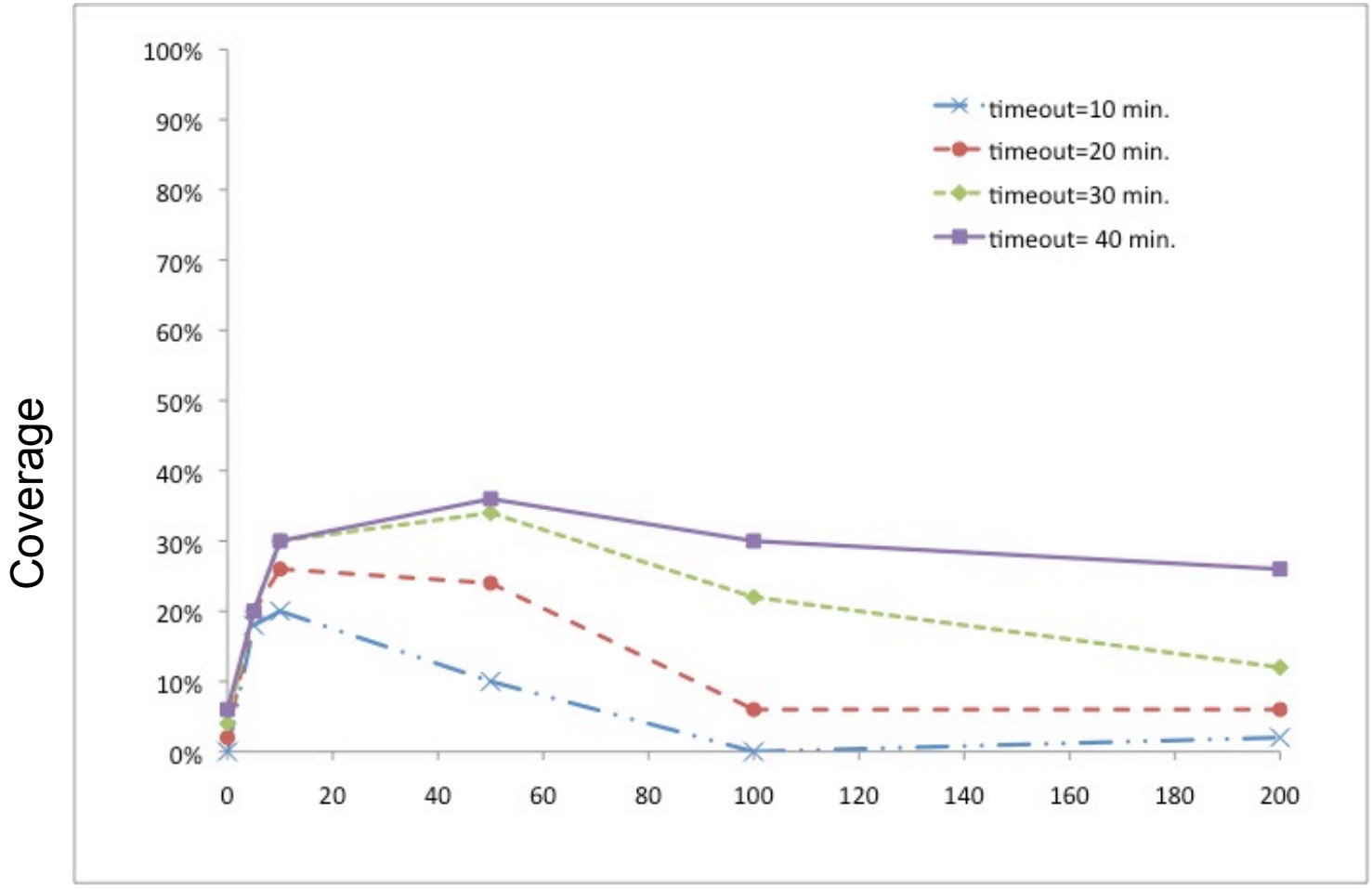
Results: TPP


Coverage
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1.0
1.1
1.2
1.3
1.4
1.5
C
LAMA
FD-AT1
FD-AT2
LPG
FF
LPRPGP

Results: TPP


Coverage
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1.0
1.1
1.2
1.3
1.4
1.5
A2
A2(OPSC)
A2(SR)
Arvand
LAMA
FD-AT1
FD-AT2
LPG
FF
LPRPGP
C

Other Experiments: the effect of pool size on SR



Domain: NoMystery

IPC-2011

Planner	Arvand	A2(SR)	A2(OPSC)	LAMA	FD-AT1	FD-AT2	M	Mp	LPRPGP
Coverage	66%	68%	50%	51%	76%	68%	30%	40%	46%

- SR
  - never worst
  - better in 4 domains
- OPSC
  - slow progress in the search space

Contributions

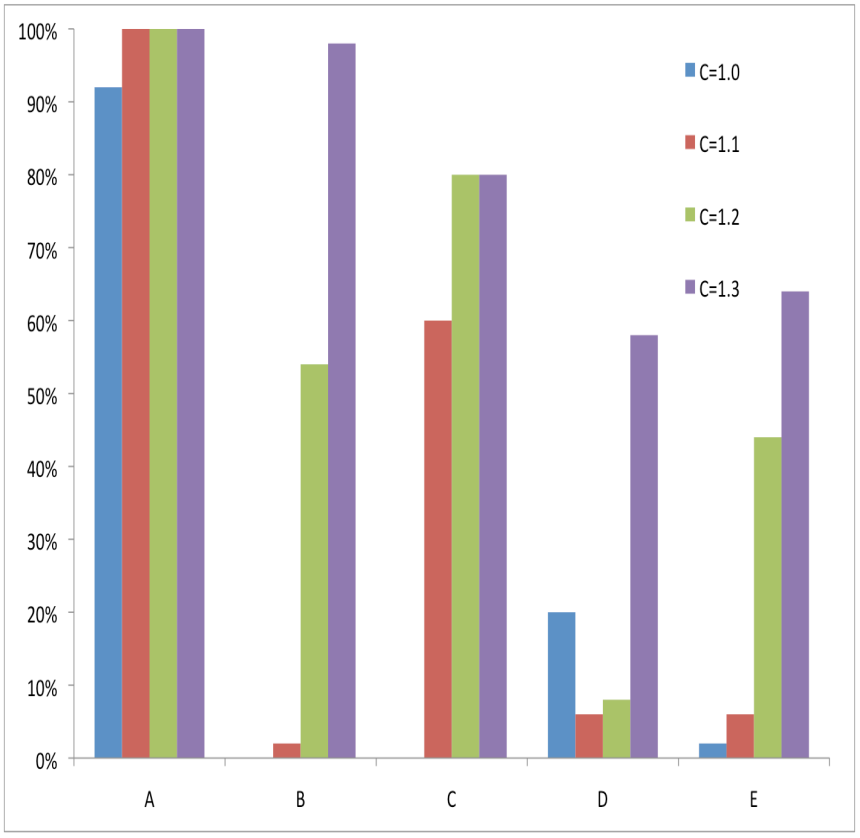
- Defined Resource constrainedness for multiple resources
- Extended benchmark suite controlling C
- large-scale study of the current state of the art as a function of C
- Two new techniques to improve Arvand

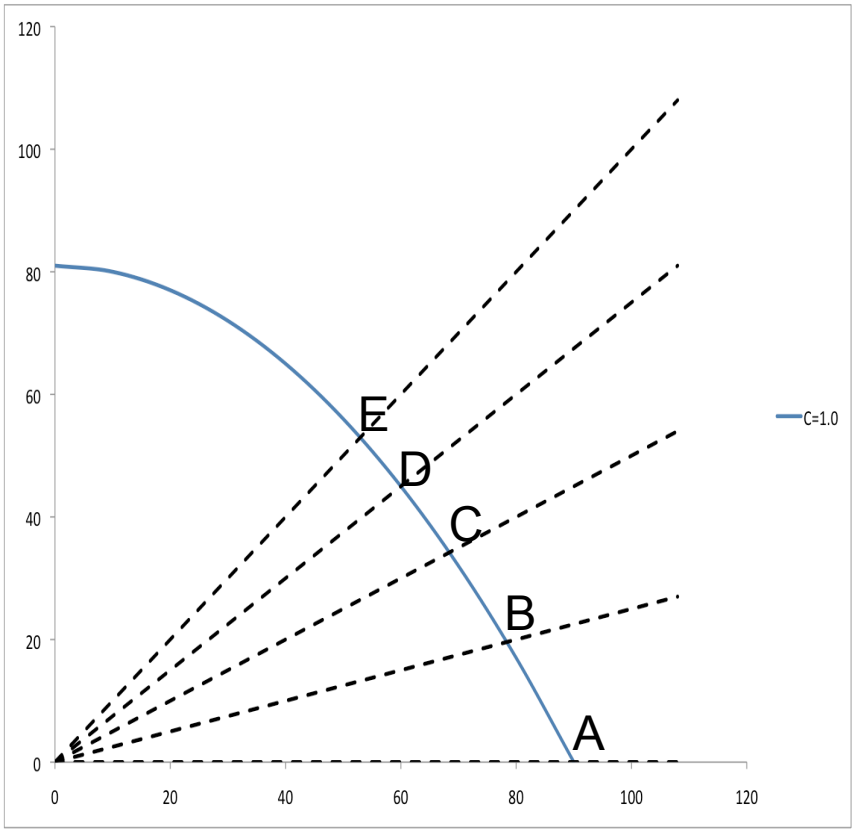
Future Work

- Deal with resources explicitly
- Design automatic configuration methods

Thank you for your attention!

Resource Distribution

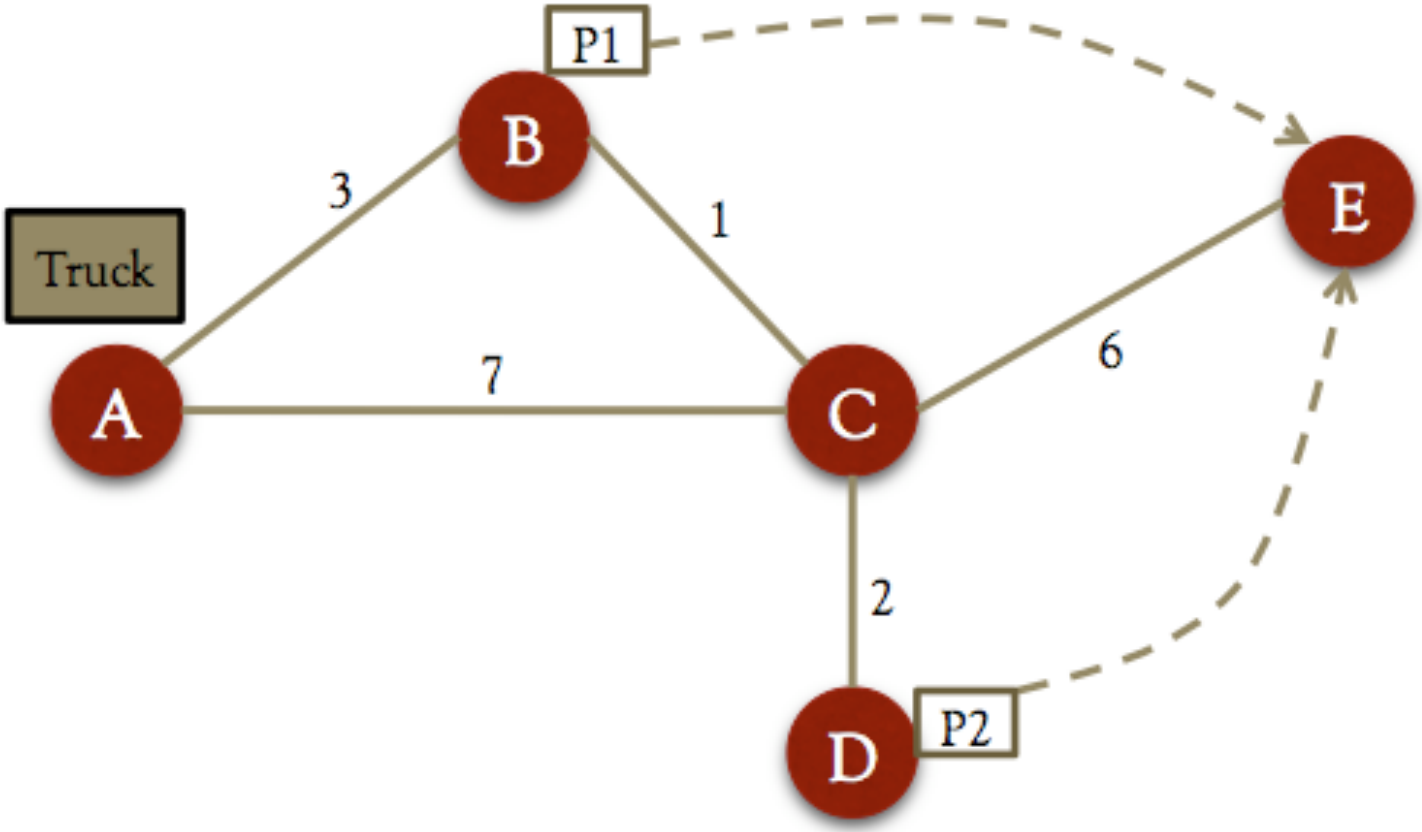




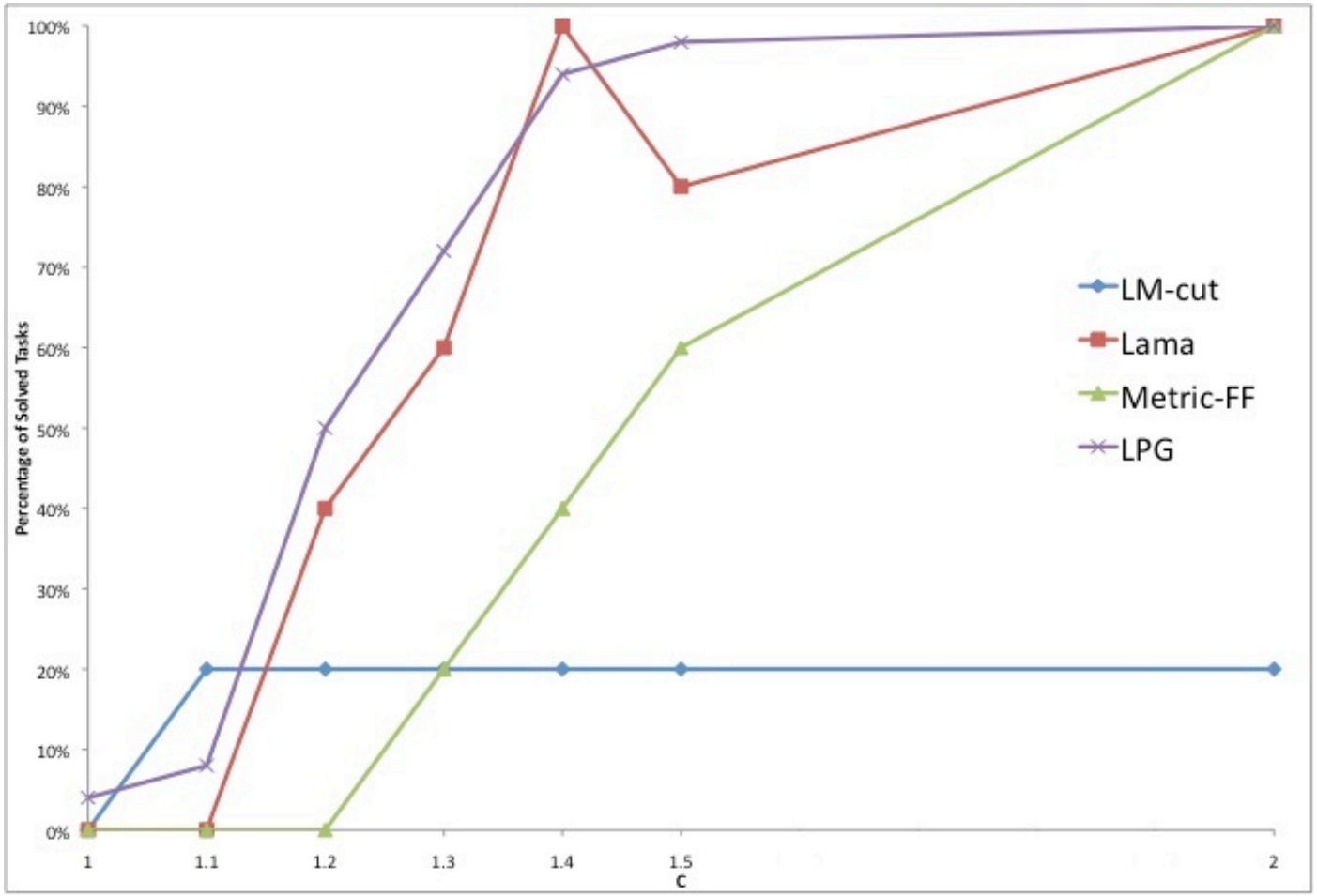
Contributions

- Generalized the previous notion of C to the case of multiple resources
- Introduced two new techniques
  - Smart Restarting (SR)
  - On Path Search Continuation (OPSC)
- Extended benchmark suite controlling C
- Large-scale study of current Planners on RCP

An Example



Study the performance as a function of C



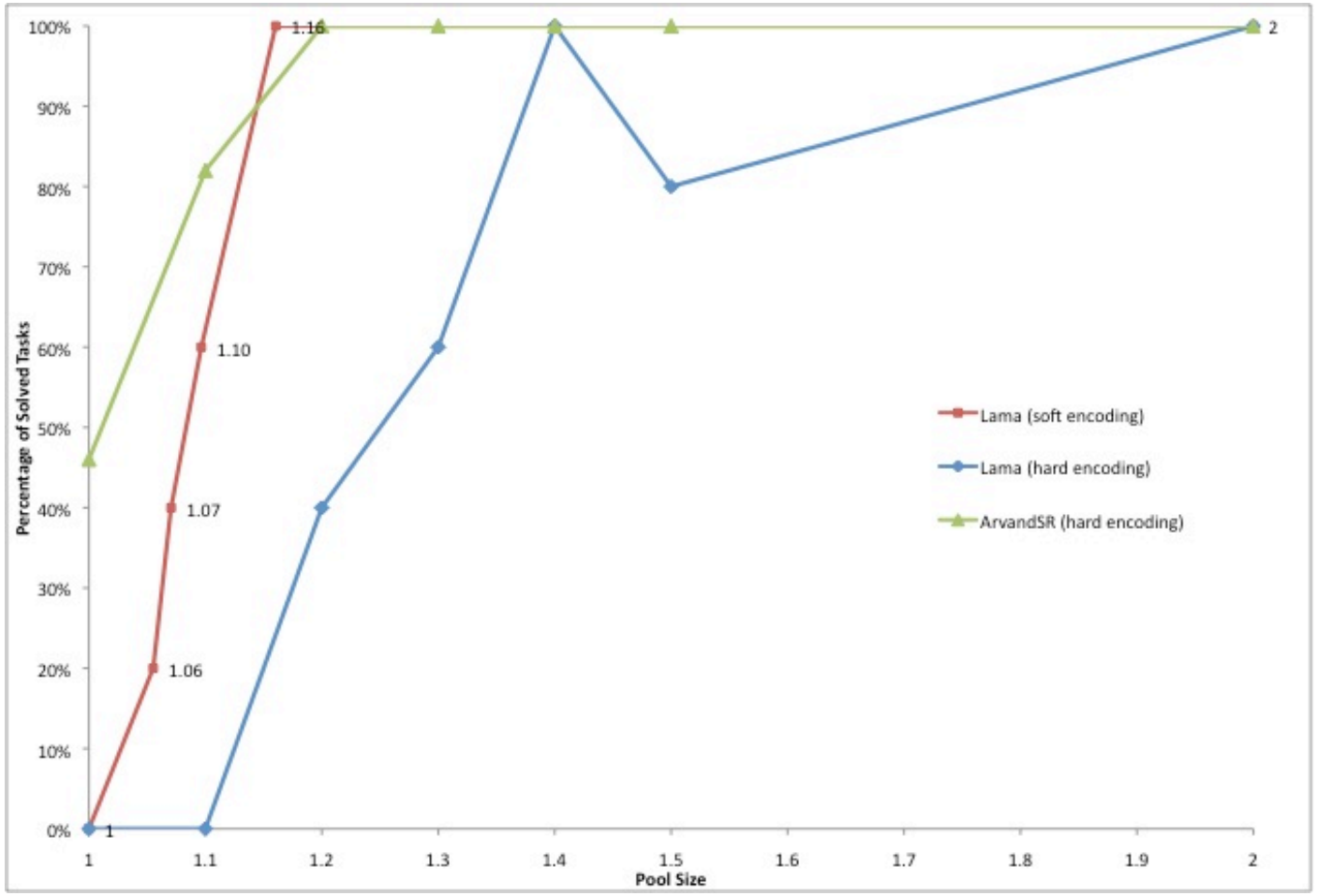
NoMystery: Summary

- Original Arvand already outperforms the state of the art
  - Coverage for C=1.0 is 18% compared to 4% for LPG
- Smart restarts can significantly improve Arvand
  - Coverage for C=1.0 improves to 46%
  - For C=1.1 coverage is 90% compared to 8% for LPG
  - Comparing to the previously known best planner LPG, there is a factor > 11 improvement in coverage for C=1.0
- The benefit of smart restarts tends to grow as C tends to 1

IPC Benchmarks

- Domains with resources: Mystery, Mprime, Trucks
- Puzzle domains: Pipesworld, Freecell
- Smart restarting improves the results in two domains and does not harm in other domains

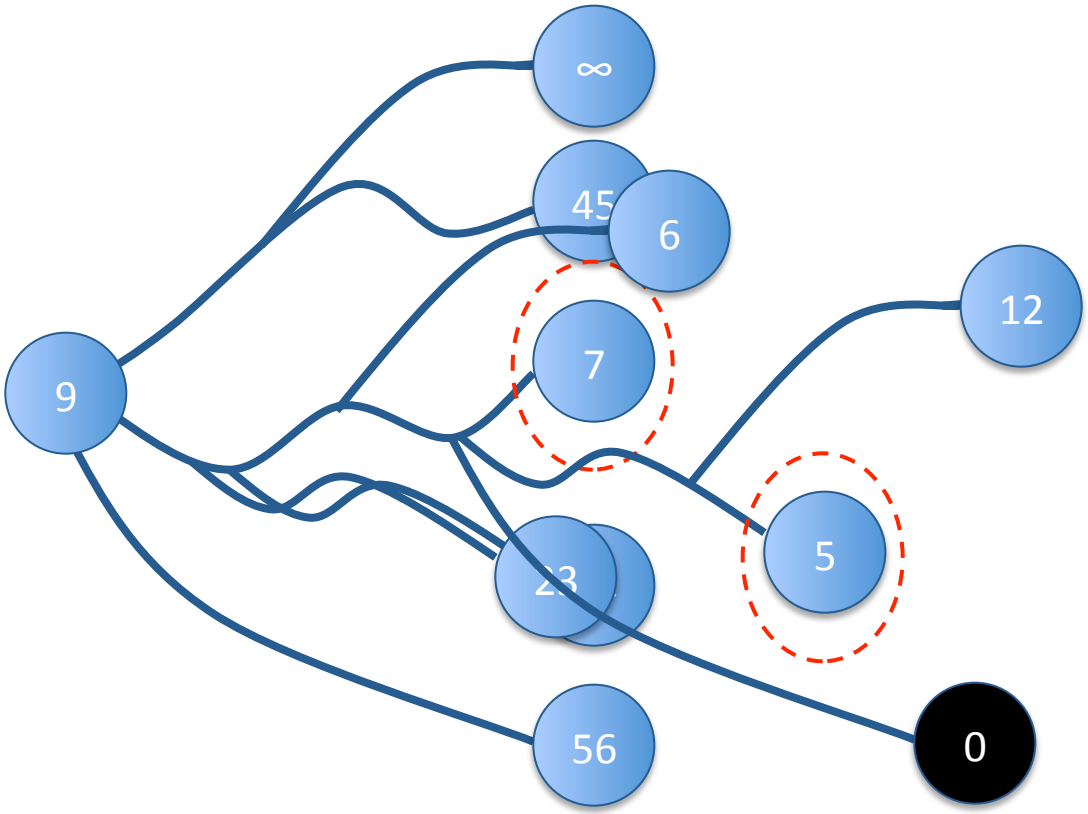
Soft Encoding vs. Hard Encoding



Conclusions

- Current stat-of-the-art planners are very bad at economizing limited resources
- Local search can help

On-Path-Search Continuation



IPC Domains

- Usually planners are evaluated on competition domains
- Wide range of domains with different characteristics
- Different problem instances with different sizes

Motivation

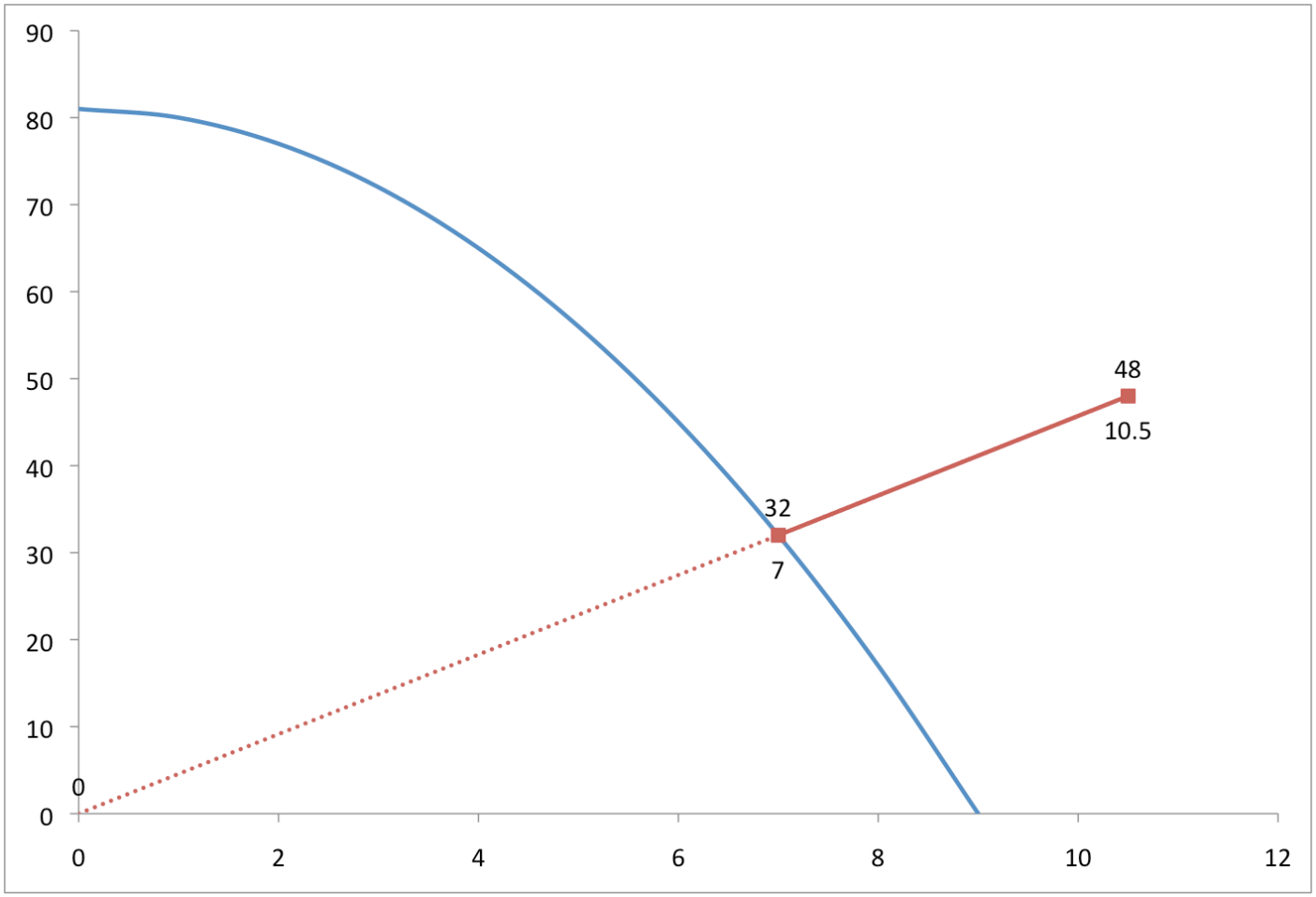
- Need to economize limited resources
- Limitation of heuristic planners in critically resource-constrained problems
  - Relaxation heuristics do not model resource consumption at all
  - Greedy search algorithms add more problems

Controlling Resource Availability

- How the behavior of algorithm changes when the problem becomes more constrained
- Let C be the ratio between the amount of available resources vs. the minimum amount required
- The problem becomes intuitively harder as C approaches 1

Arvand

- Forward chaining local search
- In each step, run random walks to find the next state
- If no improvement after several steps, then restart


90
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
48
10.5
32
7
0

RCP Benchmarks:

- TPP
  - 5 problems with 1 agent, 8 market, 8 products
- NoMystery, small
  - 25 problems with 2 trucks, 9 locations, 9 packages
- NoMystery, large
  - 5 problems with 1 truck, 12 locations, 15 packages
- Rovers, small
  - 25 problems with 2 rovers, 11 locations, 16 objectives
- Rovers, large
  - 5 problems with 1 rover, 15 locations, 20 objective

Resource Distribution

- 5 random problems with 5 pareto-optimal resource allocation for each of them