Temporal Planning with Preferences and Time-Dependent Continuous Costs

J. Benton
j.benton@asu.edu
Arizona State University

Amanda Coles
amanda.coles@kcl.ac.uk

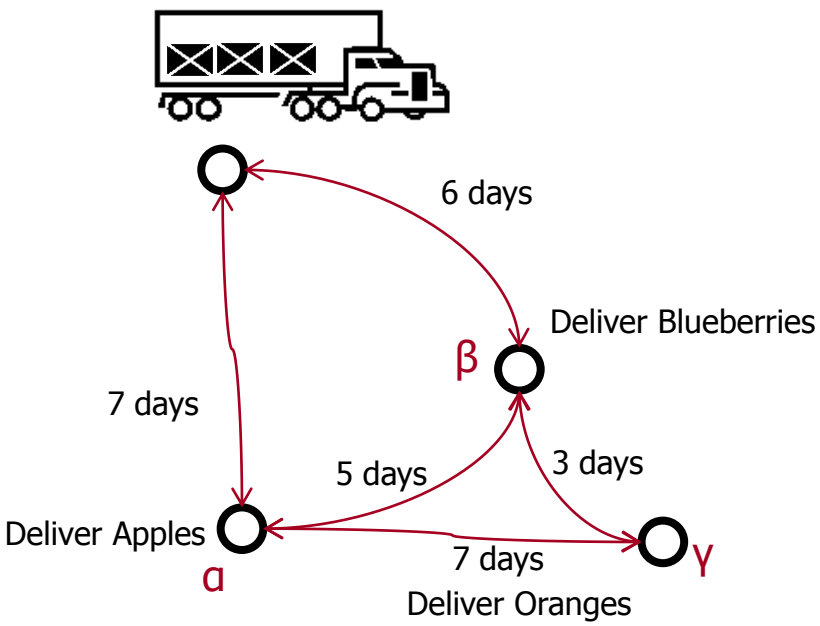
Andrew Coles
andrew.coles@kcl.ac.uk

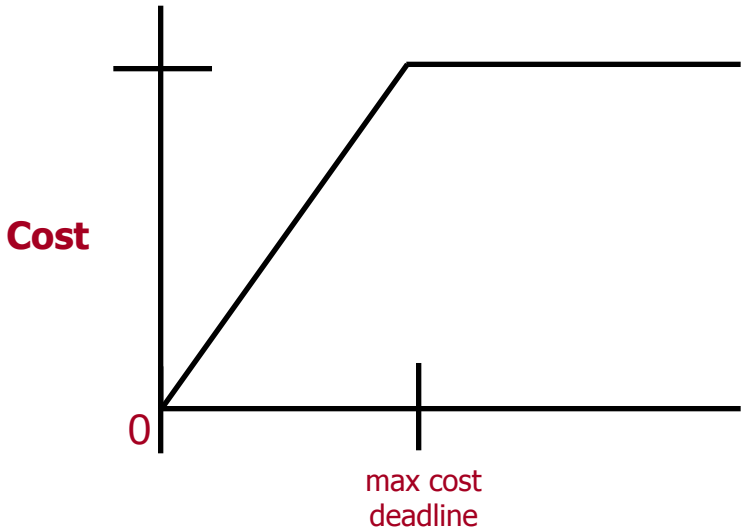
King's College London

soon... SIFT

Makespan != Plan Utility

The Dilemma of the Perishable Food

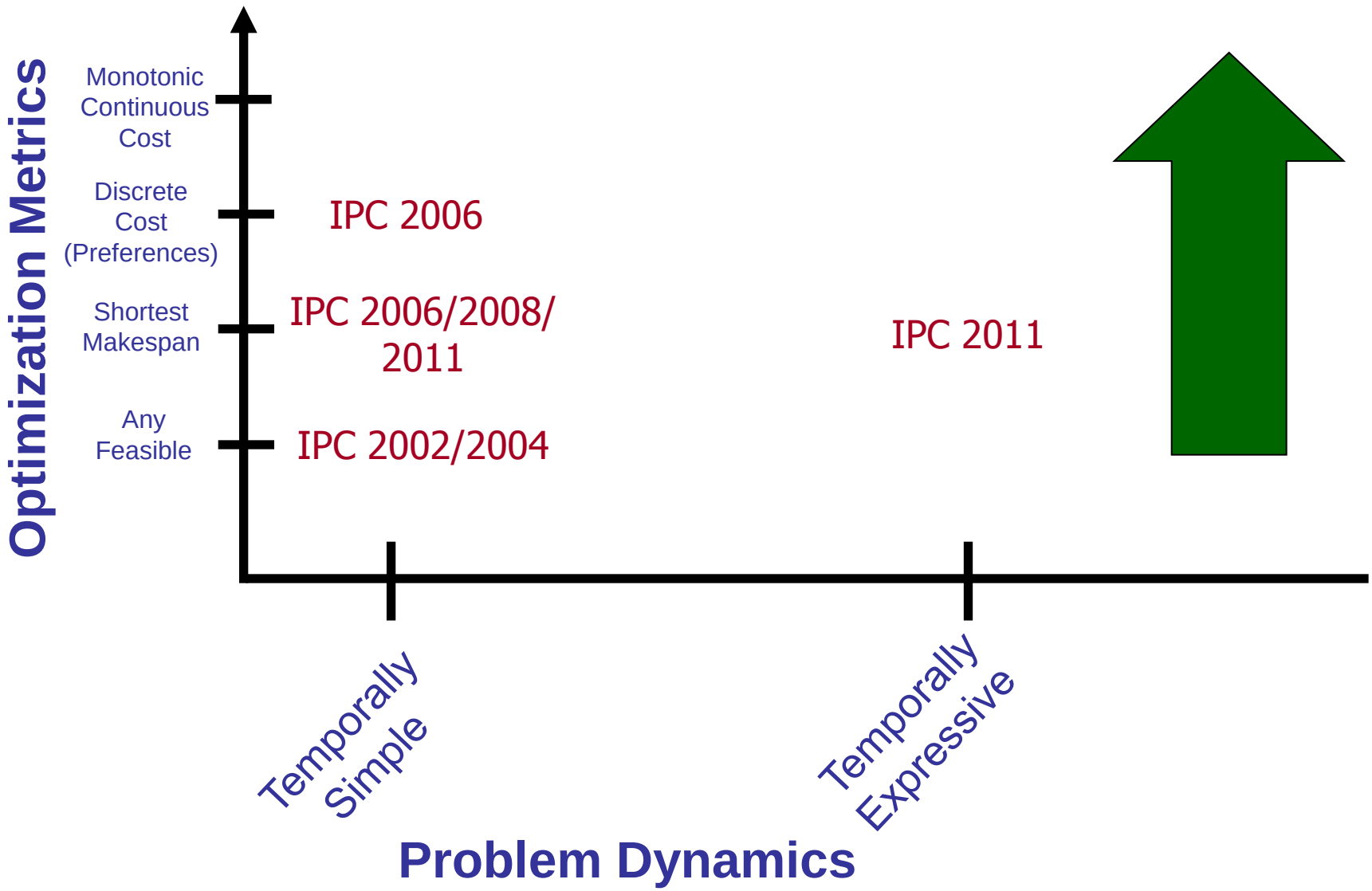




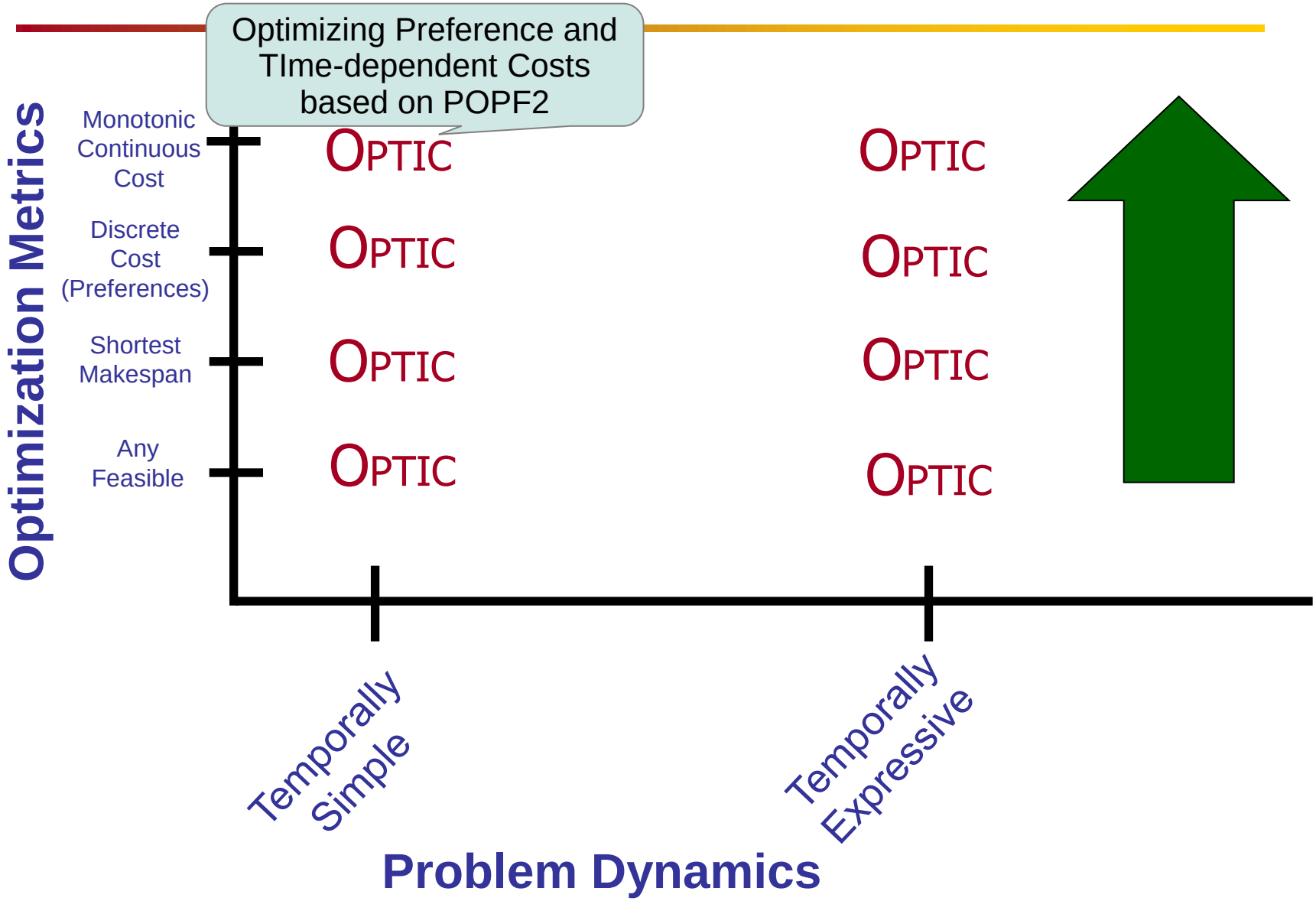
Apples last ~20 days
Oranges last ~15 days
Blueberries last ~10 days

plan	makespan	time-on-shelf
α→β→γ	15	13 + 0 + 0 = 13
β→γ→α	16	4 + 6 + 4 = 14

Satisficing Temporal Planning



Satisficing Temporal Planning



Outline

- Handling Discrete Temporal Costs (PDDL3 Preferences)
- Handling Continuous Costs
- Conclusion

Representation Challenges

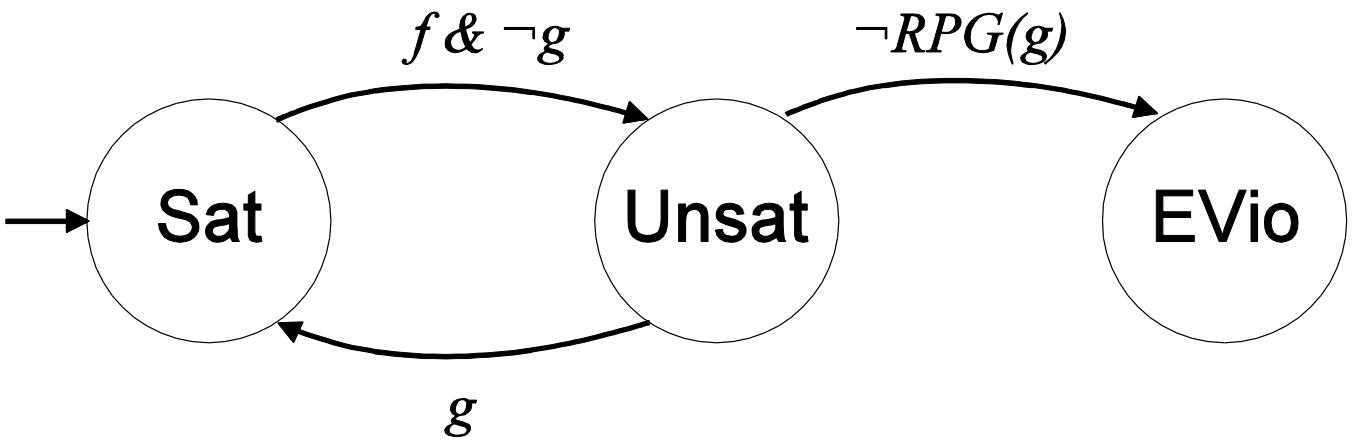
- PDDL 3 -- established language from the IPC 2006
  - Offers discrete-cost using linear temporal logic subset

```
(preference p1 (within 45 (at package1 loc1)))
(preference p2 (sometime-after (at package1 loc1)
                               (at package2 loc1)))

(:metric minimize (+ (* (is-violated p1) 10)
                     (* (is-violated p2) 5)))
```

Automata to Represent PDDL3 Preferences

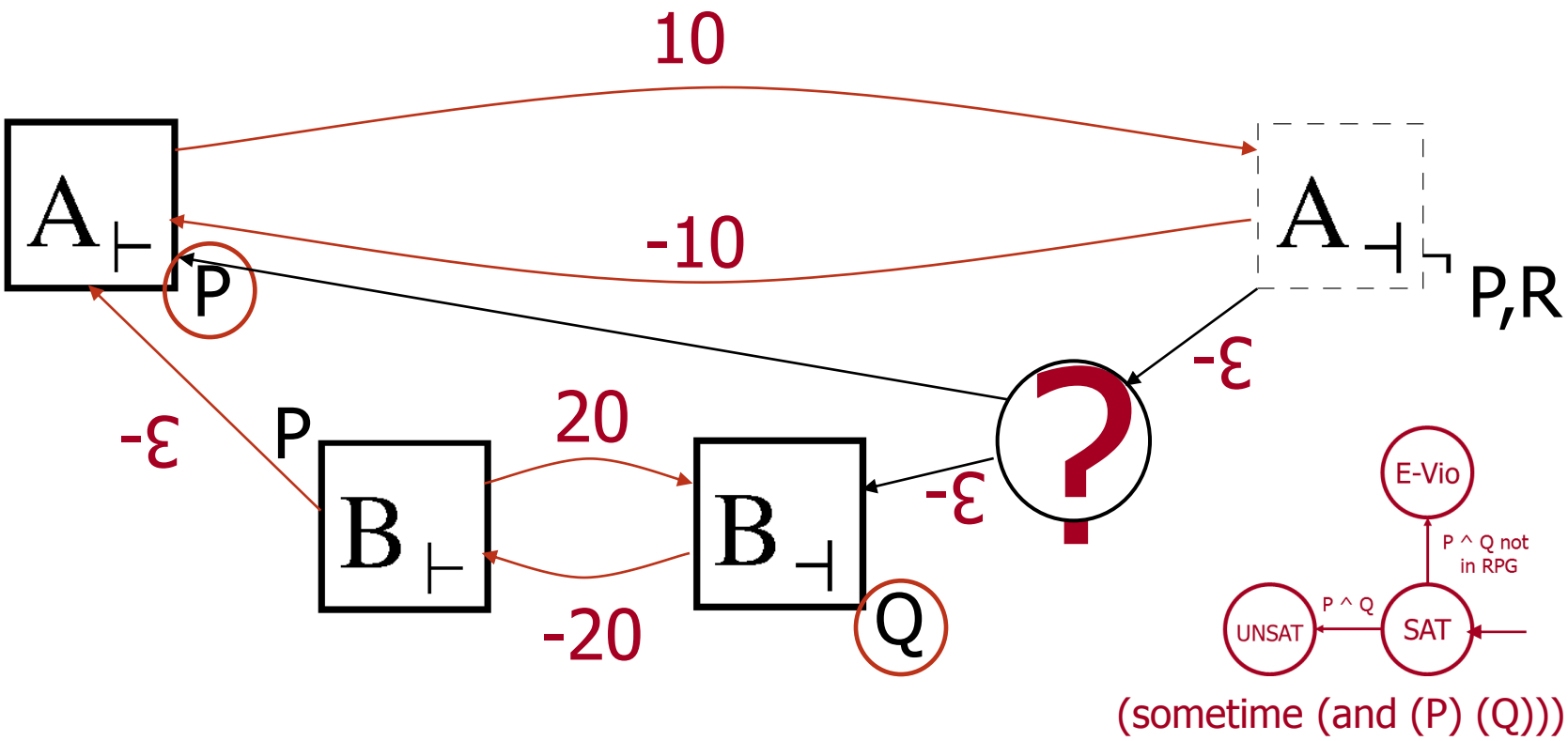
- As in HPlanP, Mips-XXL, LPRPG-P: encode preferences as **automata**
- Update positions as actions are applied



(sometime-after f g)

- Clear interpretation in sequential plans; in temporal plans…?

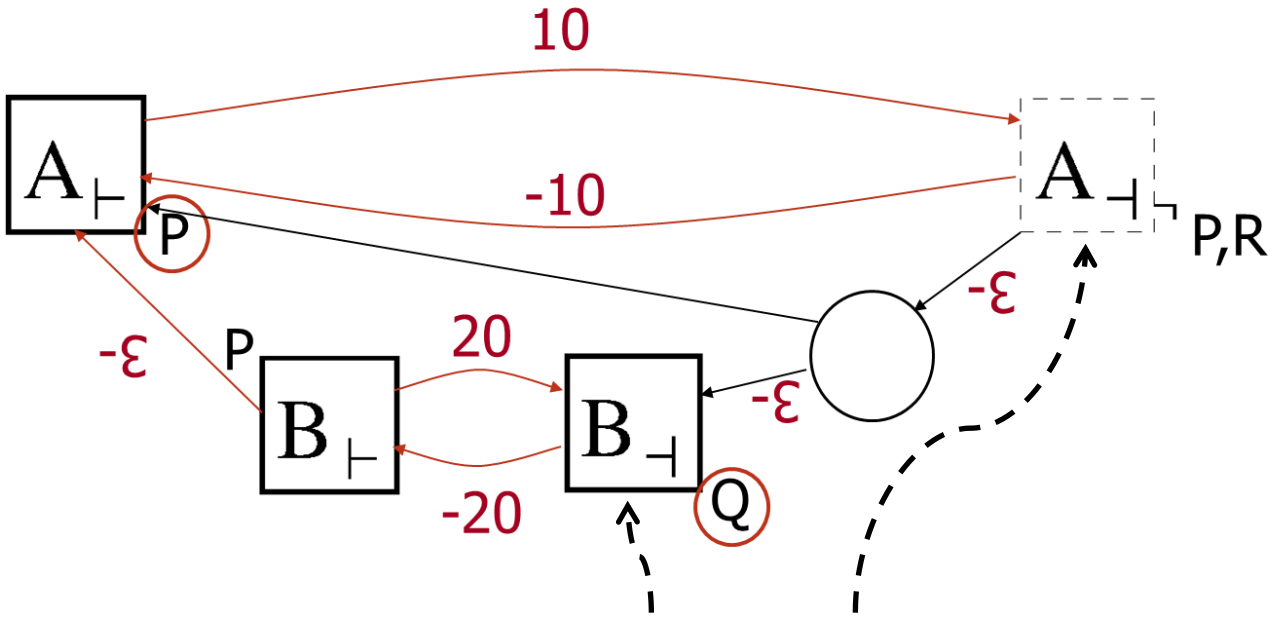
sometime (and (P) (Q))



Simple Temporal Network in each node:
Built as we add actions

Scheduling Challenge

- Challenge: Efficiently handle temporal preferences
  - e.g., ending A before B

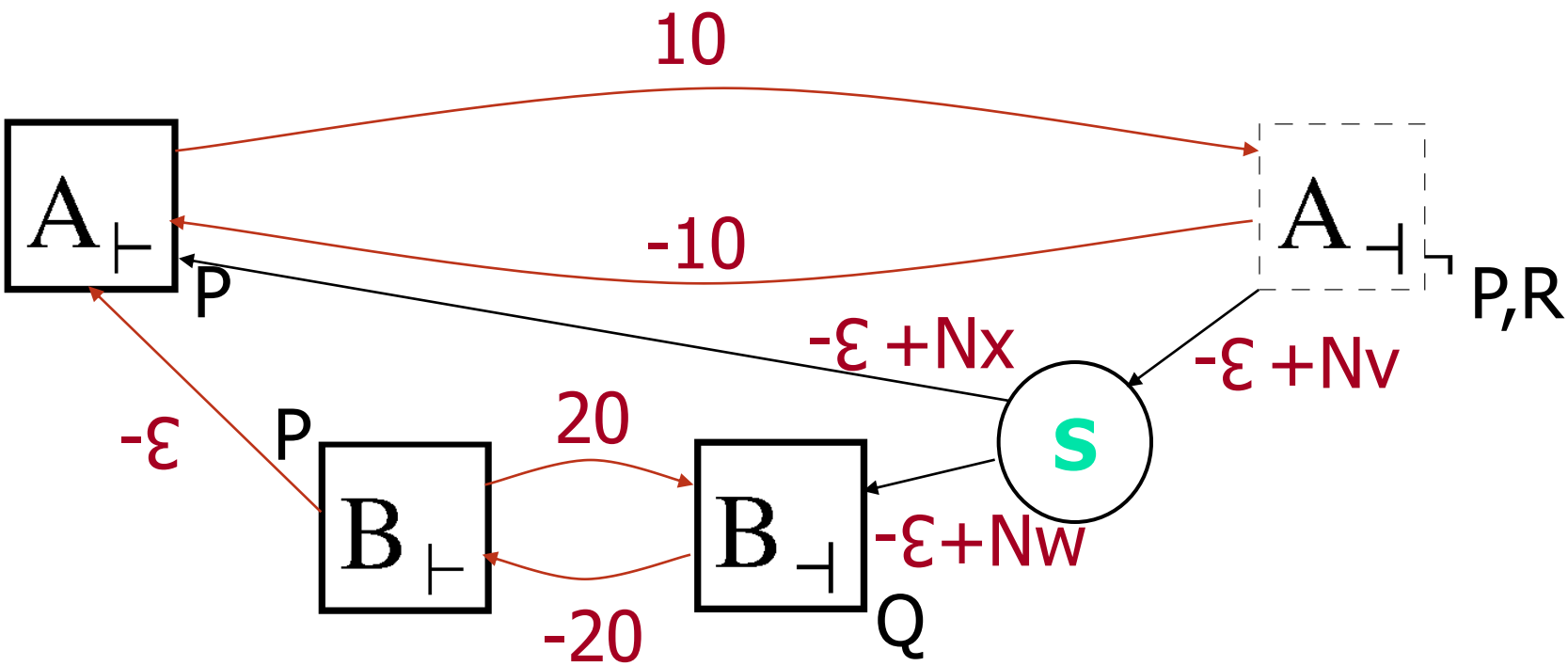


How do we avoid backtracking on this scheduling choice?

Dummy-steps as a MIP

- Solution: Build a mixed integer program (MIP) for the simple temporal network + dummy steps
  - For each dummy step, add "soft constraints" in the MIP
  - Each preference gets a [0,1] integer variable
  - From these variables and constraints, determine the status of the preference. **Objective: minimize cost.**

sometime (and (P) (Q))



v,w,x,s in [0,1], N is a large constant

(v + w + x > 0) <=> (s = 1)

PDDL3 Results

- Compared to existing Temporal Preference Planners: MIPS-XXL, SGPlan
  - Also SGPlan-W: SGPlan with a dummy never-applicable action added to the domain
- PDDL3 domains from IPC 2006: Pipesworld, Storage, Trucks, TPP, Pathways
- 4GB memory limit, 30 minutes CPU

Pipesworld

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OPTIC	**36.4**	**142.5**	**92.8**	**40.3**	**48**	**71.2**	121.7	**75**	**34.2**	**94.6**	-	**70.3**	-	41.5	0
MIPSXXL	0	78.7	22.2	15.2	23.8	-	-	-	-	-	-	-	-	-	-
SGPLAN	**36.4**	105.1	86	35	**48**	63.8	95.8	45.9	**34.2**	-	**75.9**	30.5	**92**	**88.7**	20.3
SGPLAN-W	**36.4**	105.1	86	28.2	34.4	63.8	**171.7**	67	21.4	69.8	53.8	33.3	**92**	71.2	12.1

- OPTIC/SGPlan have similar performance
  - OPTIC struggles on p11, p13
- (Only domain where SGPlan-W is effective)

Storage, Trucks

	Problem	1	2	3	4	5	6	7	8	9	10
Storage	OPTIC	**0**	**1**	**2**	**13**	**127**	**208**	**397**	**477**	**737**	**914**
	MIPSXXL	3	4	38	81	-	-	-	-	-	-
	SGPLAN	18	37	86	113	227	322	505	591	887	1082
	SGPLAN-W	13	-	-	-	-	-				

Out-of-Memory
Grounding Preferences

	Problem	11	12	13	14	15	16				
Storage	OPTIC	**1334**	**1472**	**2063**	**2366**	**3206**	**3383**				
	MIPSXXL	-	-	-	-	-	-	-	-	-	-
	SGPLAN	1532	1670	2248	2622	3290	3571	4556	4813	6332	6378
	SGPLAN-W	-	-	-	-	-	-	-	-	-	-

	Problem	1	2	3	4	5	6	7	8	9	10
Trucks	OPTIC	**0**	**0**	**0**	**0**	**0**	**0**	**0**	6	**4**	**15**
	MIPSXXL	**0**	**0**	-	-	-	-	-	-	-	-
	SGPLAN	4	**0**	2	2	2	1	6	**4**	**4**	24
	SGPLAN-W	-	-	-	-	-	-	-	-	-	-

	Problem	11	12	13	14	15	16	17	18	19	20
Trucks	OPTIC	13	**0**	**6**	21	**0**	36	54	45	**19**	73
	MIPSXXL	-	-	-	-	-	-	-	-	-	-
	SGPLAN	**8**	6	12	**11**	8	**23**	**50**	**29**	32	**46**
	SGPLAN-W	-	-	-	-	-	-	-	-	-	-

TPP, Pathways

	Problem	1	2	3	4	5	6	7	8	9	10
TPP	OPTIC	**0**	**0**	**0**	**0**	5	**0**	**14**	22	-	-
	MIPSXXL	-	-	-	-	-	-	-	-	-	-
	SGPLAN	1	1	**0**	2	**1**	8	16	**7**	**29**	**42**
	SGPLAN-W	-	-	-	-	-	-	-	-	-	-

	Problem	11	12	13	14	15	16	17	18	19	20
TPP	OPTIC	-	**26**	-	-	-	-	-	-	-	-
	MIPSXXL	-	-	-	-	-	-	-	-	-	-
	SGPLAN	**41**	40	**50**	**48**	**36**	**55**	**70**	**65**	**87**	**79**
	SGPLAN-W	-	-	-	-	-	-	-	-	-	-

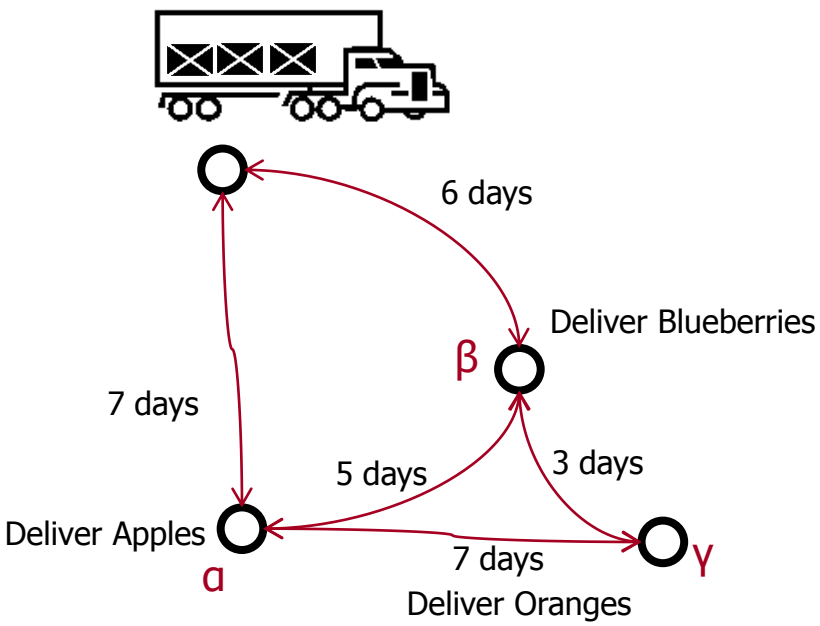
OPTIC has difficulty finding its first solution to p9 onwards

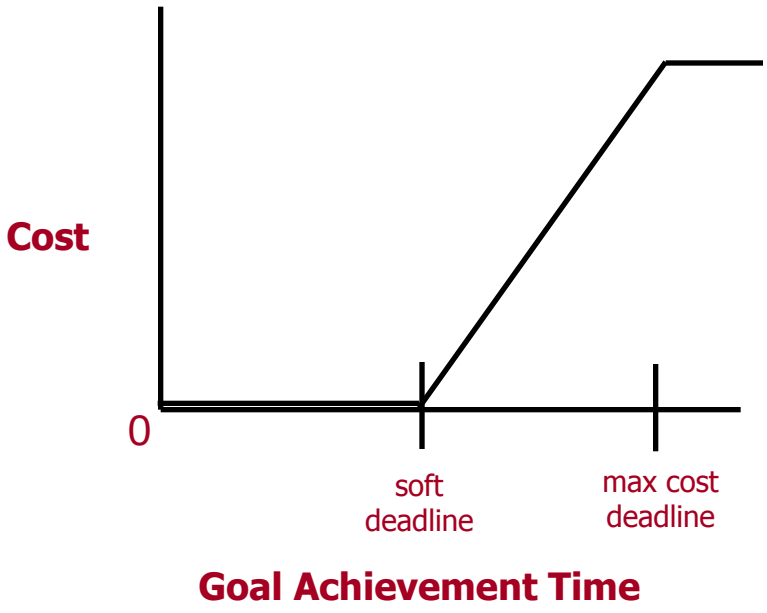
	Problem	1	2	3	4	5	6	7	8	9	10
Pathways-debugged	OPTIC	3.0	13.6	**10**	18	**23.442**	**20**	**27.3**	**33**	**29**	**33.9**
	MIPSXXL	**2**	**8.2**	11	**16.6**	25	21	28.3	34	30	34.9
	SGPLAN	6.0	12.9	-	-	-	-	-	-	-	-
	SGPLAN-W	6.0	12.9	-	-	-	-	-	-	-	-

Consistently the best performer from p5 upwards

Continuous Case

The Dilemma of the Perishable Food





Apples last ~20 days
Oranges last ~15 days
Blueberries last ~10 days

Solving for the Continuous Case

- Handling continuous costs
  - Directly model continuous costs
  - Compile into discretized cost functions (PDDL3 preferences)

Handling Continuous Costs

Model passing time as a PDDL+ process

Use "Collect Cost" Action for Goal

precondition

at(apples, α)

Conditional effects

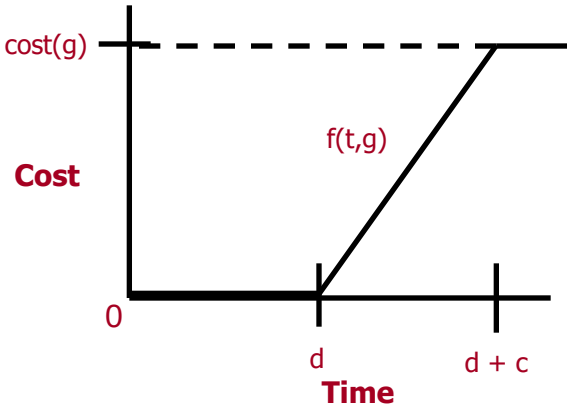
tg < d : 0

d < tg < d + c : f(t,g)

tg ≥ d + c : cost(g)

effect

collected_at(apples, α)



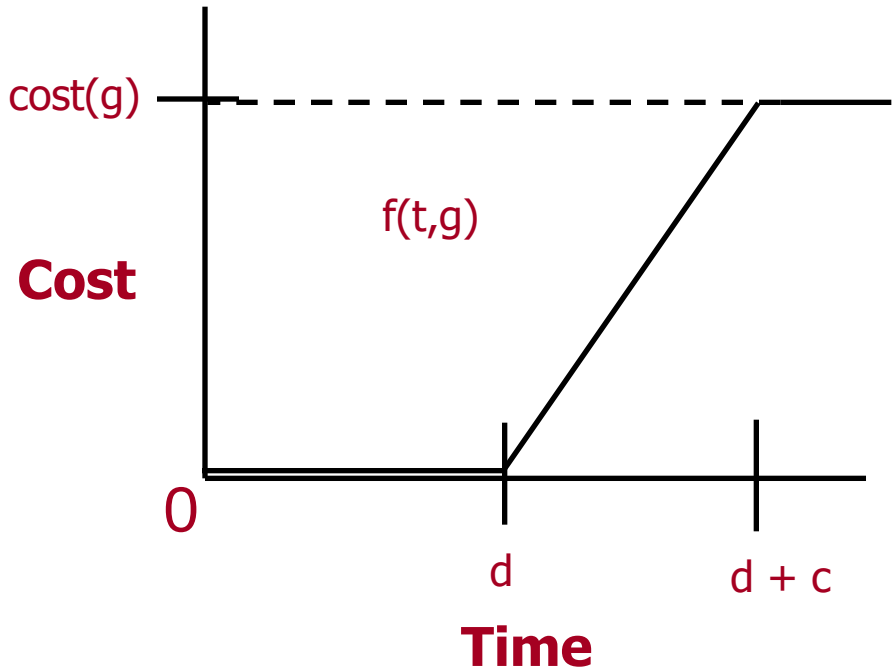
New goal

collected_at(apples, α)

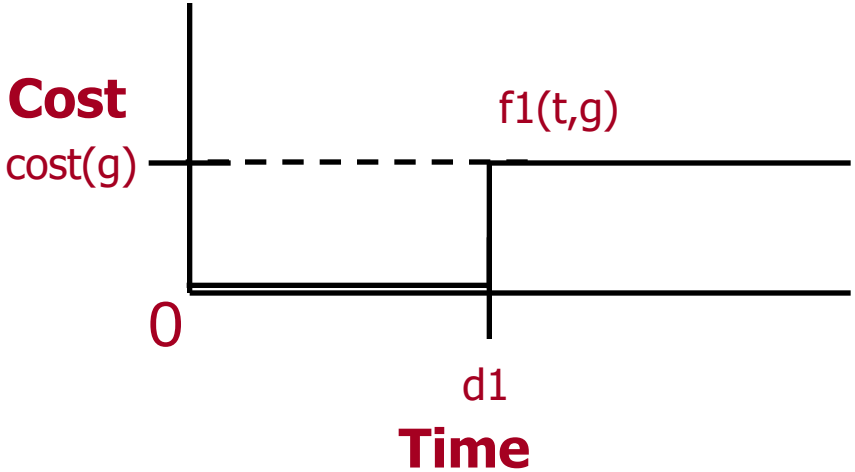
"Anytime" Search Procedure

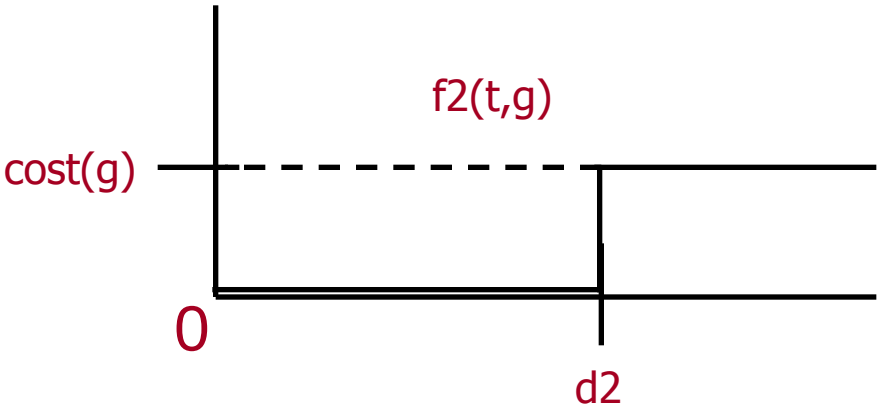
- Enforced hill-climbing search for an incumbent solution *P*
- Restart using best-first branch-and-bound:
  - Prune using cost(*P*)
    - Use admissible heuristic for pruning

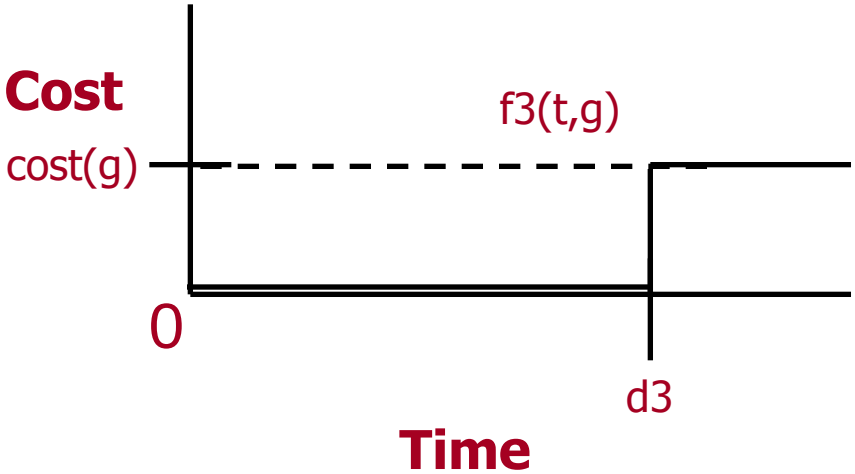
Compile to Discretized Cost



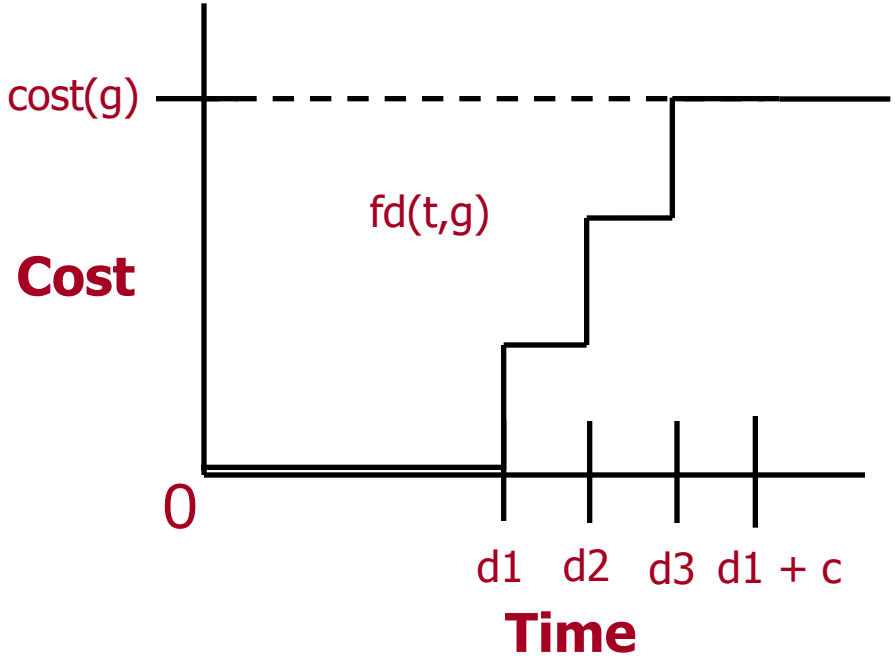
Discretized Compilation







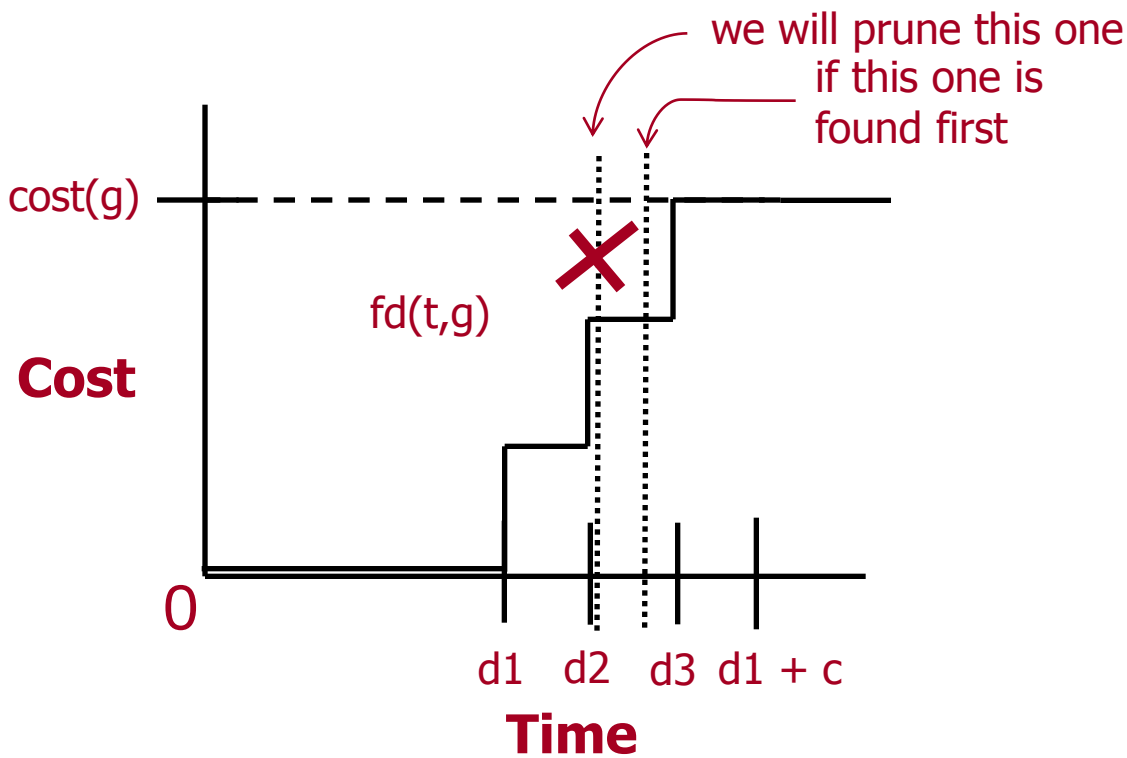
Final Discretized Compilation



$$fd(t,g) = f1(t,g) + f2(t,g) + f3(t,g)$$

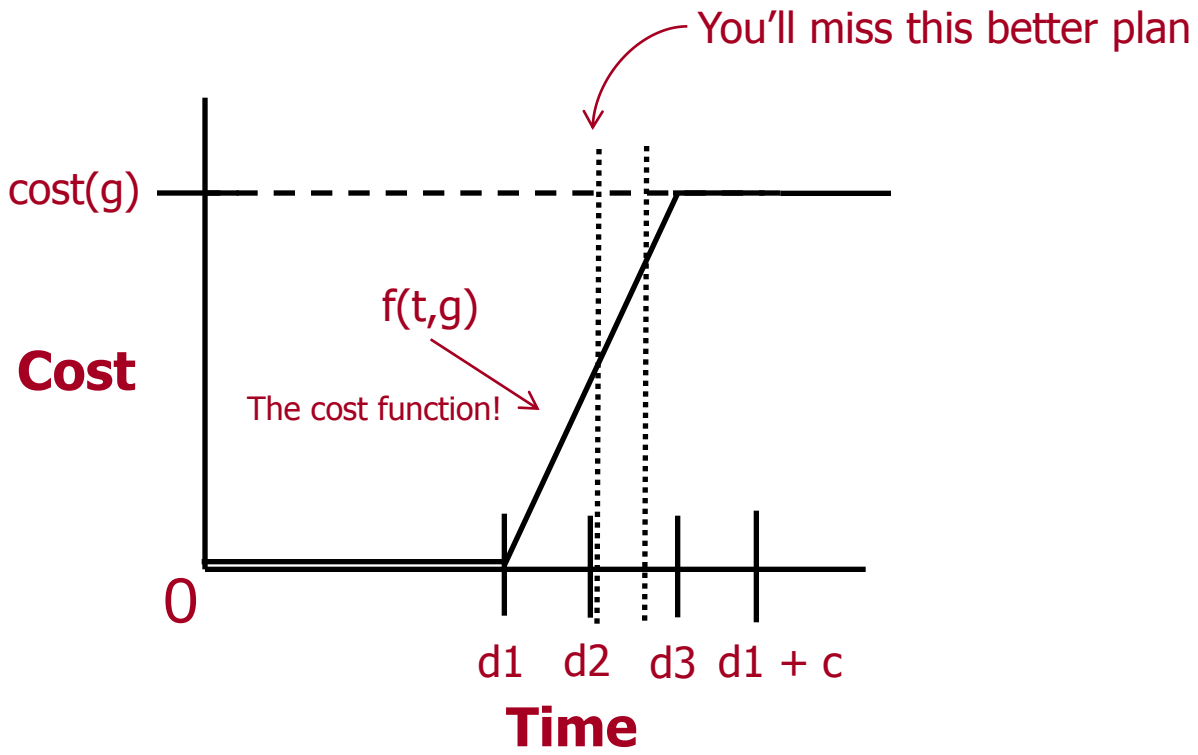
What's the best granularity?

The Secret Discretization Advantage

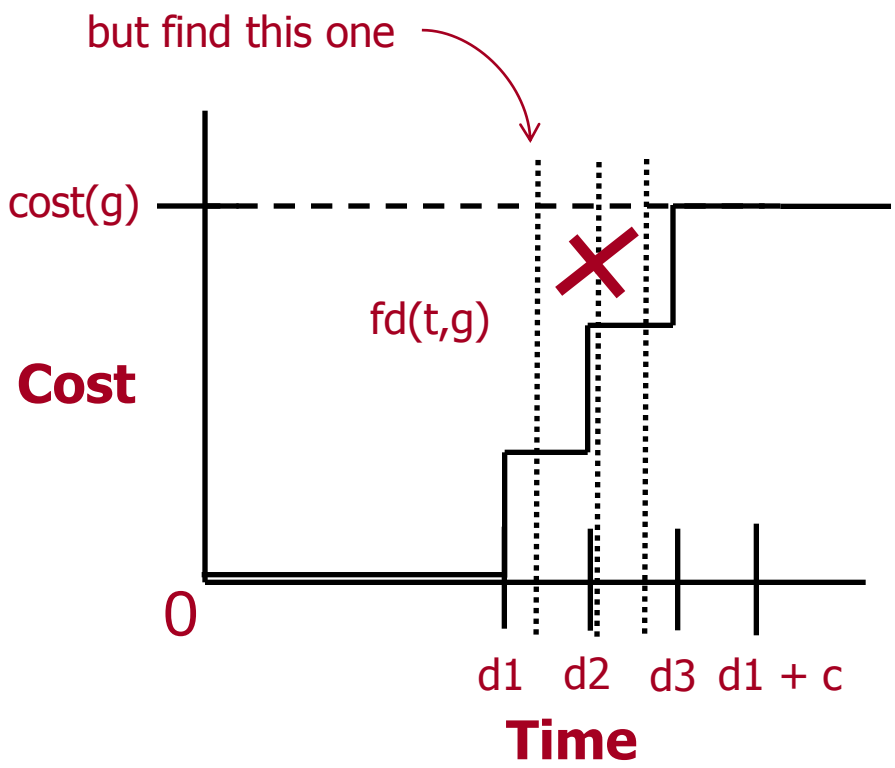


With an admissible heuristic we can do this early enough to reduce the search effort

The Secret Discretization Advantage



The Secret Discretization Advantage

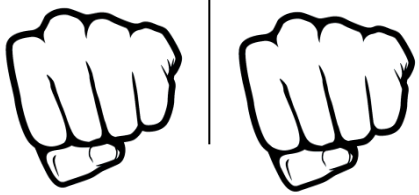


Continuous vs. Discretization

The Contenders

- Continuous Advantage
  - More accurate solutions
  - Represents actual cost functions

- Discretized Advantage
  - "Faster" search
  - Looks for bigger jumps in quality

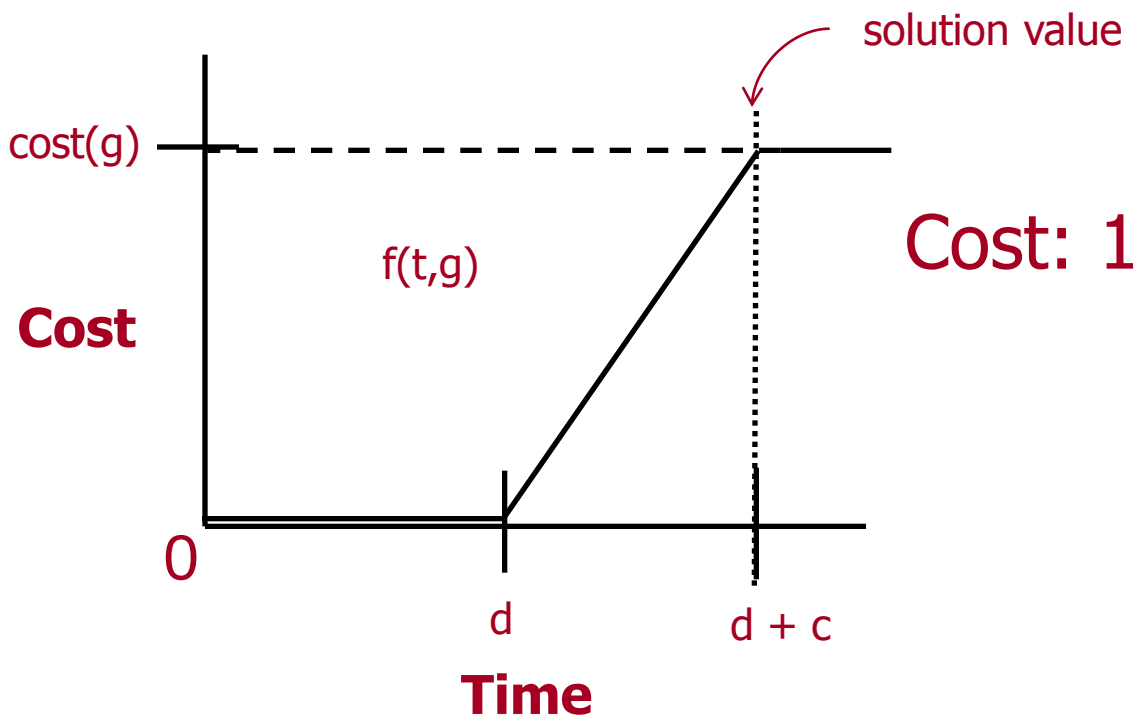


Continuous + Discrete-Mimicking Pruning

Tiered Search

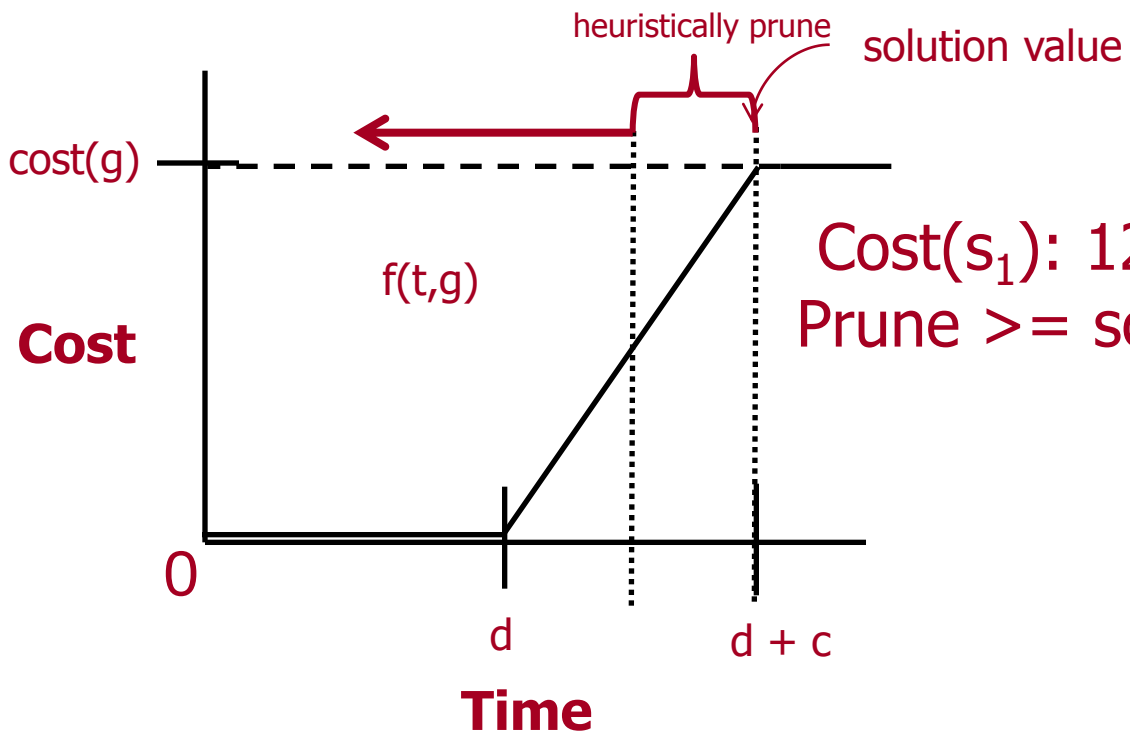
- Continuous Representation
  - More accurate solutions
  - Represents actual cost functions
- Mimicking Discrete Pruning
  - "Faster" search
  - Looks for bigger jumps in quality

Tiered Approach



Cost: 128 (sol)

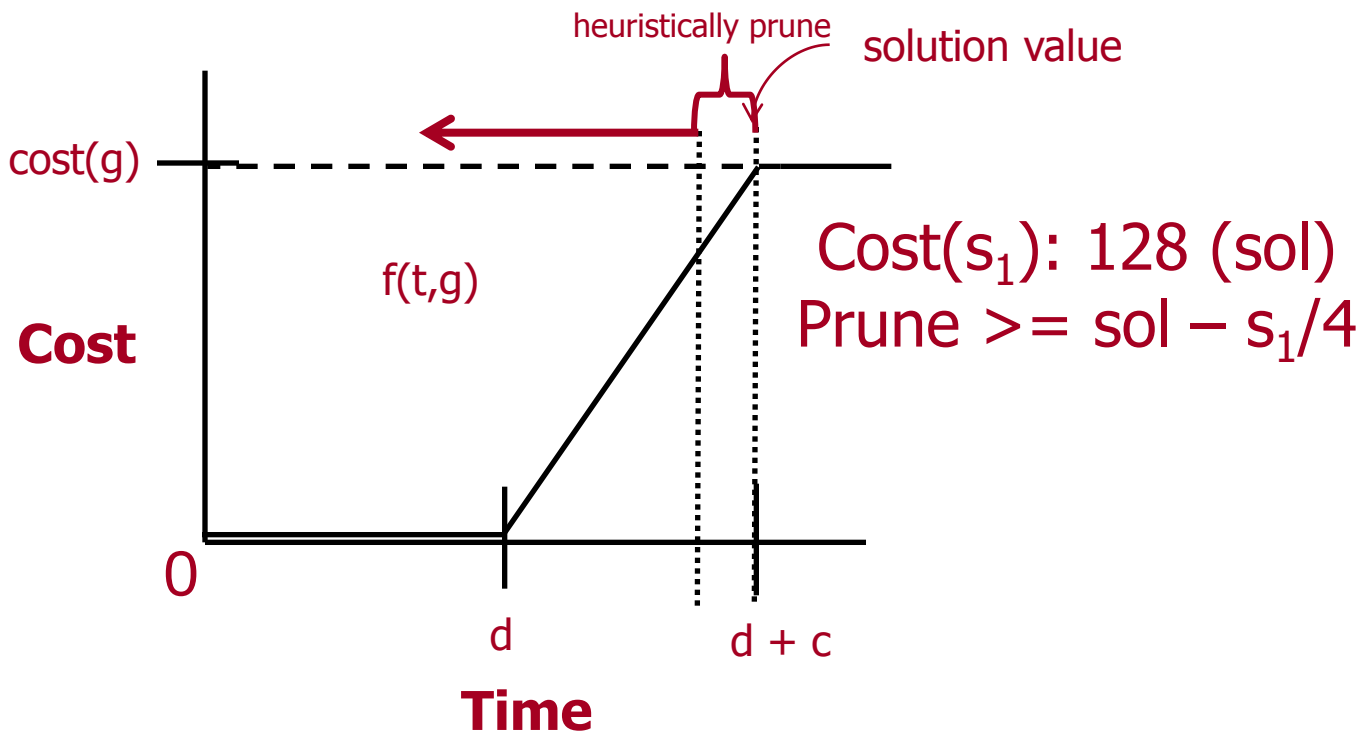
Tiered Approach



Cost(s_1): 128 (sol)
Prune >= sol – s_1/2

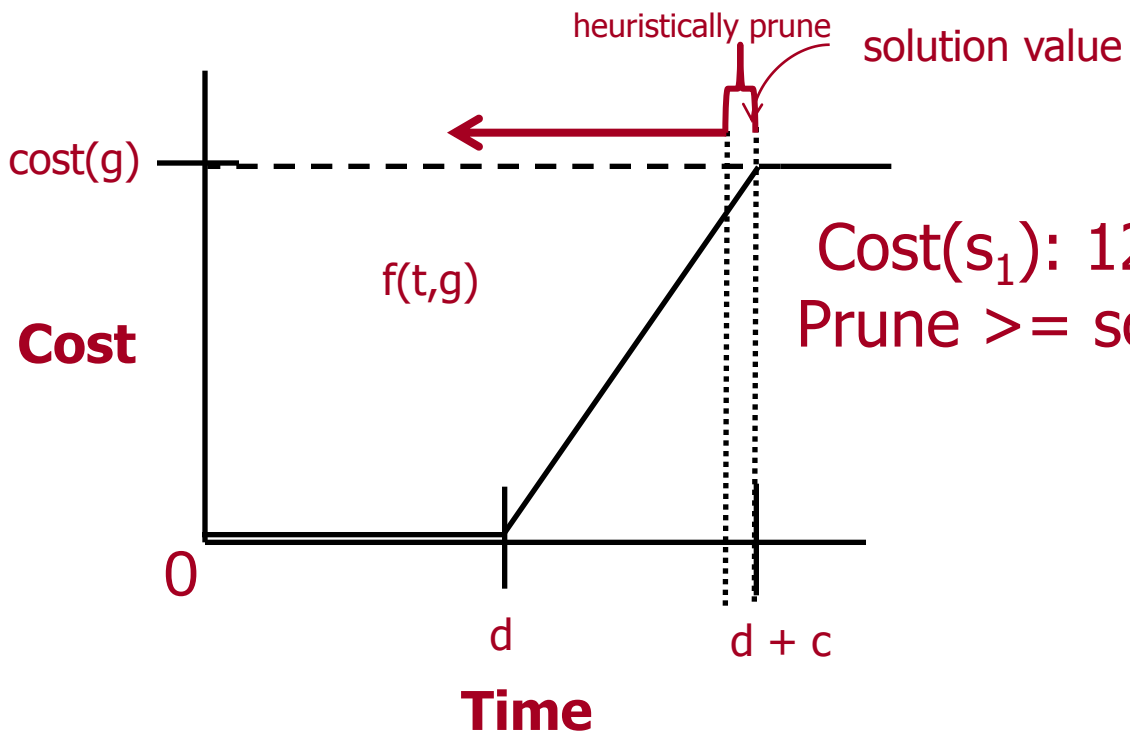
Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

Tiered Approach



Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

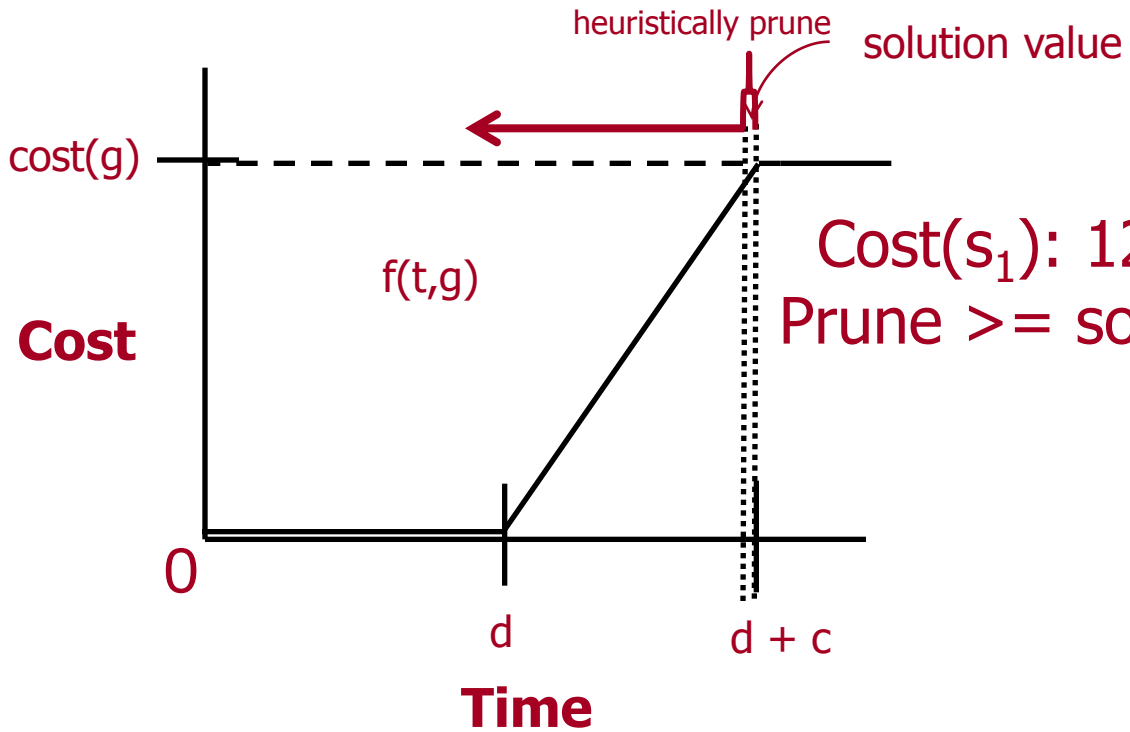
Tiered Approach



Cost(s_1): 128 (sol)
Prune >= sol – s_1/8

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

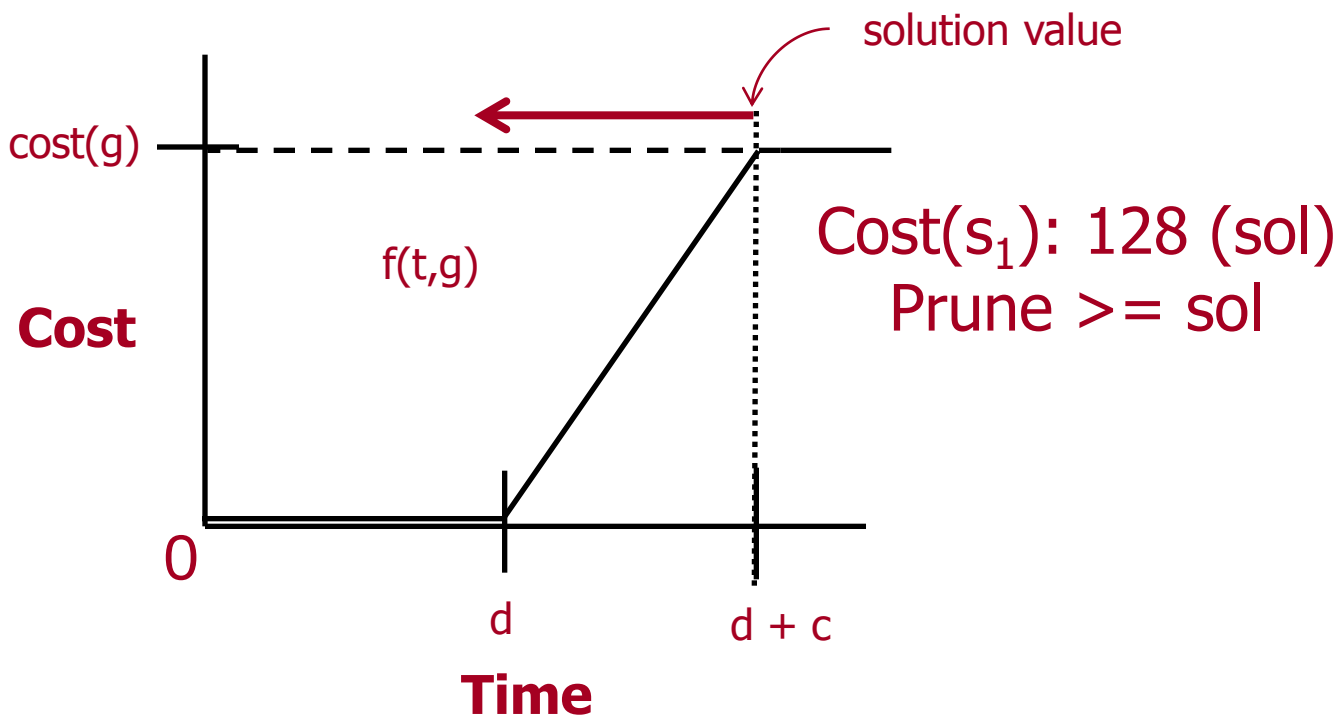
Tiered Approach



Cost(s_1): 128 (sol)
Prune >= sol – s_1/16

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

Tiered Approach



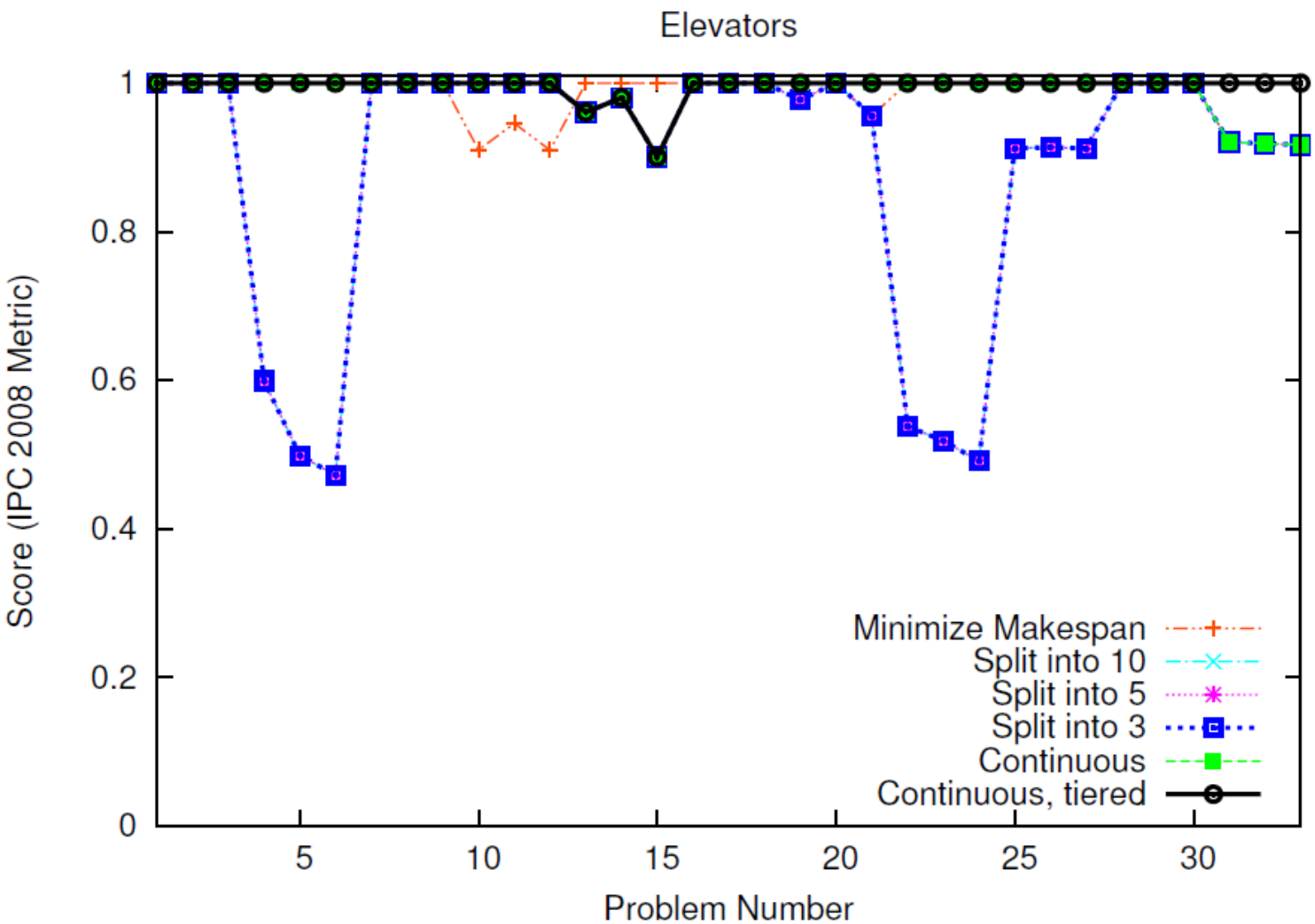
Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

Continuous Cost Results

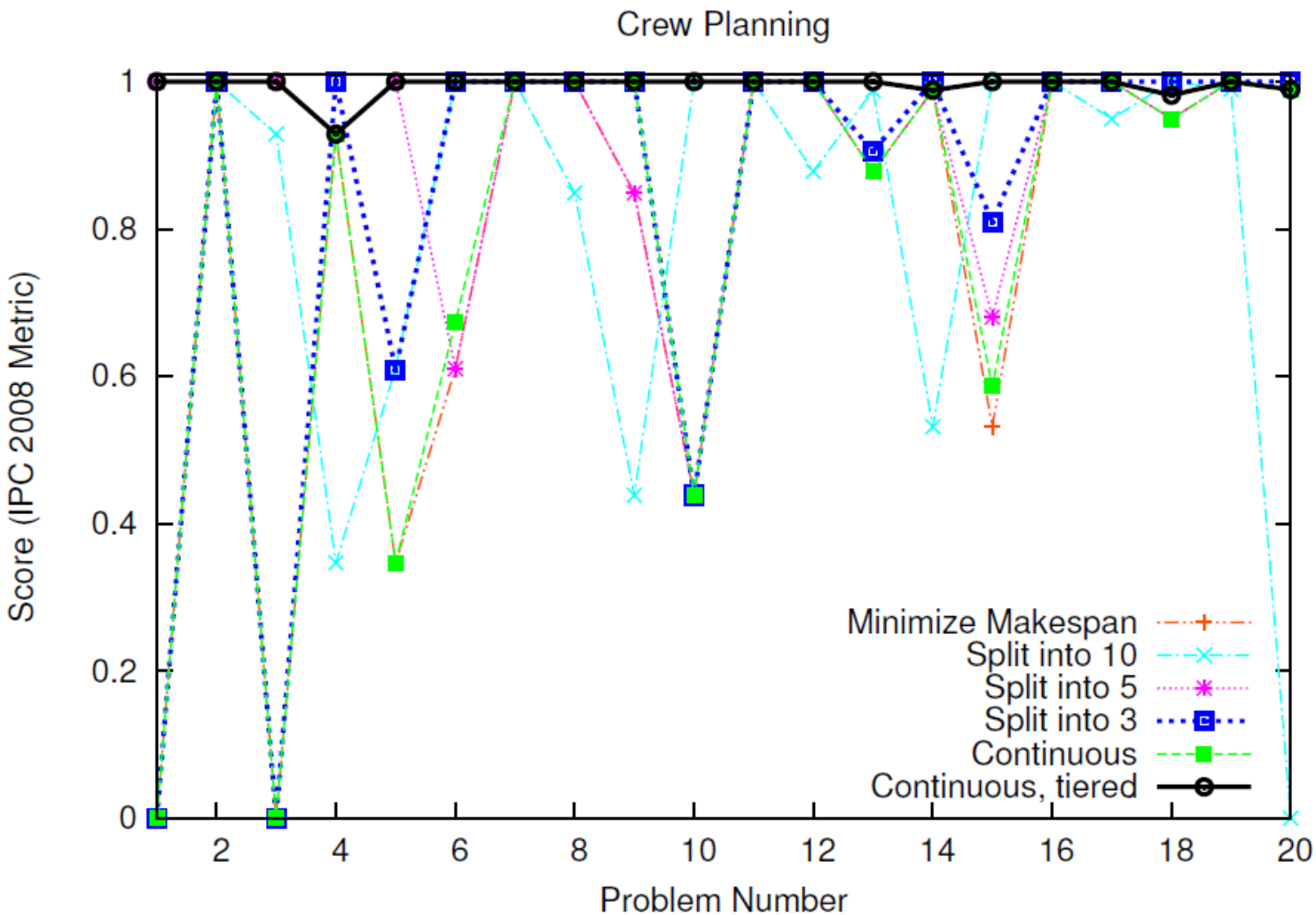
- Comparison:

  Continuous, discretized, tiered search
- Generated three continuous cost domains
  - Elevators: Soft deadlines where each person is solved for independently (w/ random delay)
  - Openstacks: Soft deadlines based on production durations
  - Crew Planning: Soft deadlines on each crew member waking up and payloads
- 4GB memory limit, 30 minutes CPU

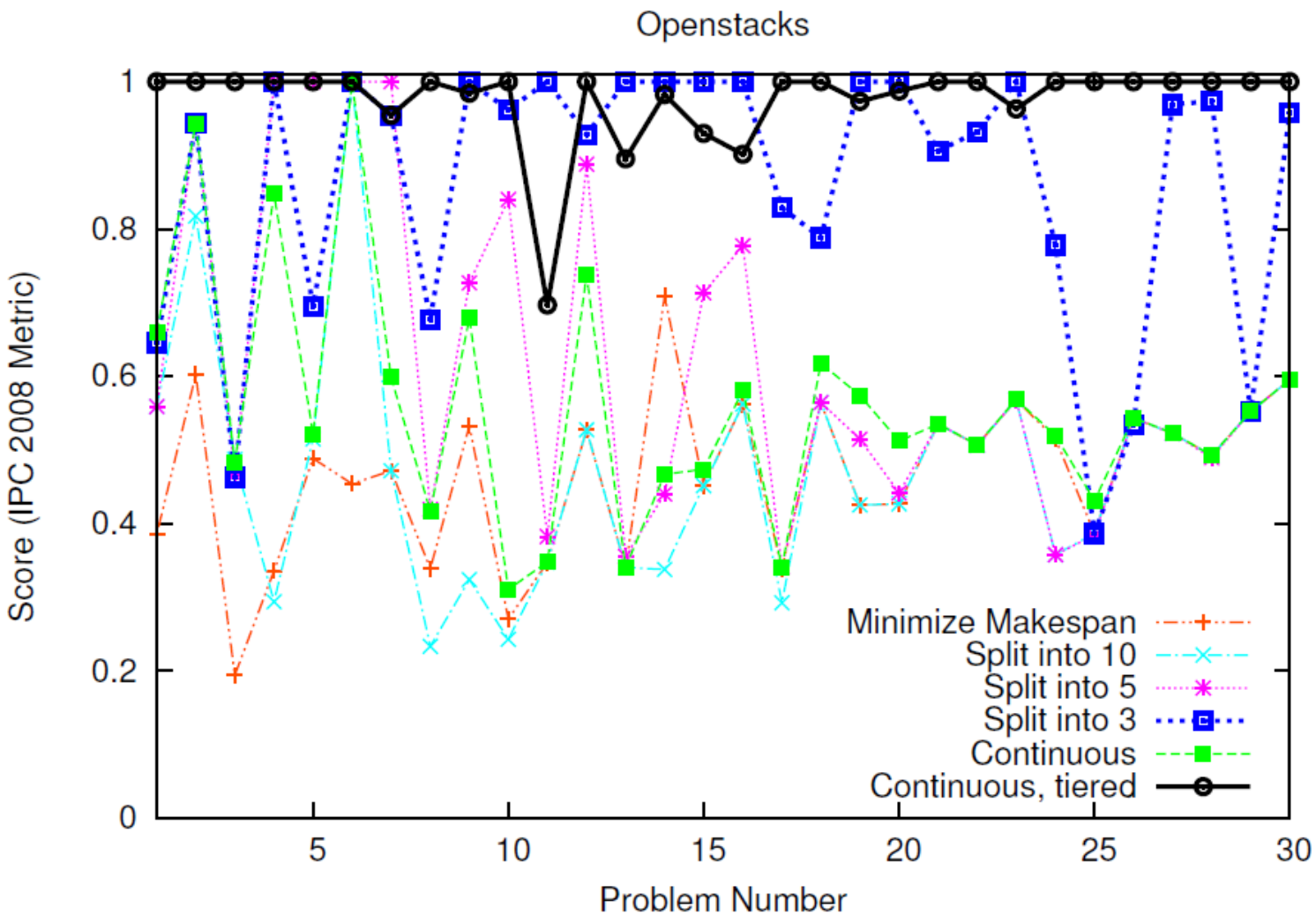
Time-dependent Cost Results



Time-dependent Cost Results



Time-dependent Cost Results



Summary

- State-of-the-art planning for PDDL3 preferences + temporal planning
- Handles (monotonic) continuous cost functions on goal achievement
- Efficient search technique for handling monotonic continuous cost functions

Future Work

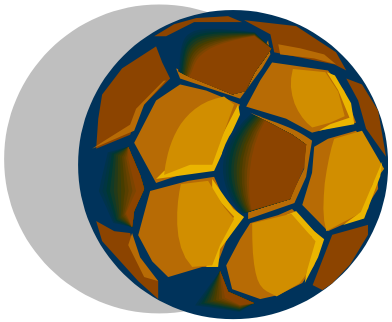
- Handle continuous costs over time windows (analogous to "`always-within`")
- Test search approach on other planning problems (e.g., classical cost-based planning)

Thanks!

OPTIC

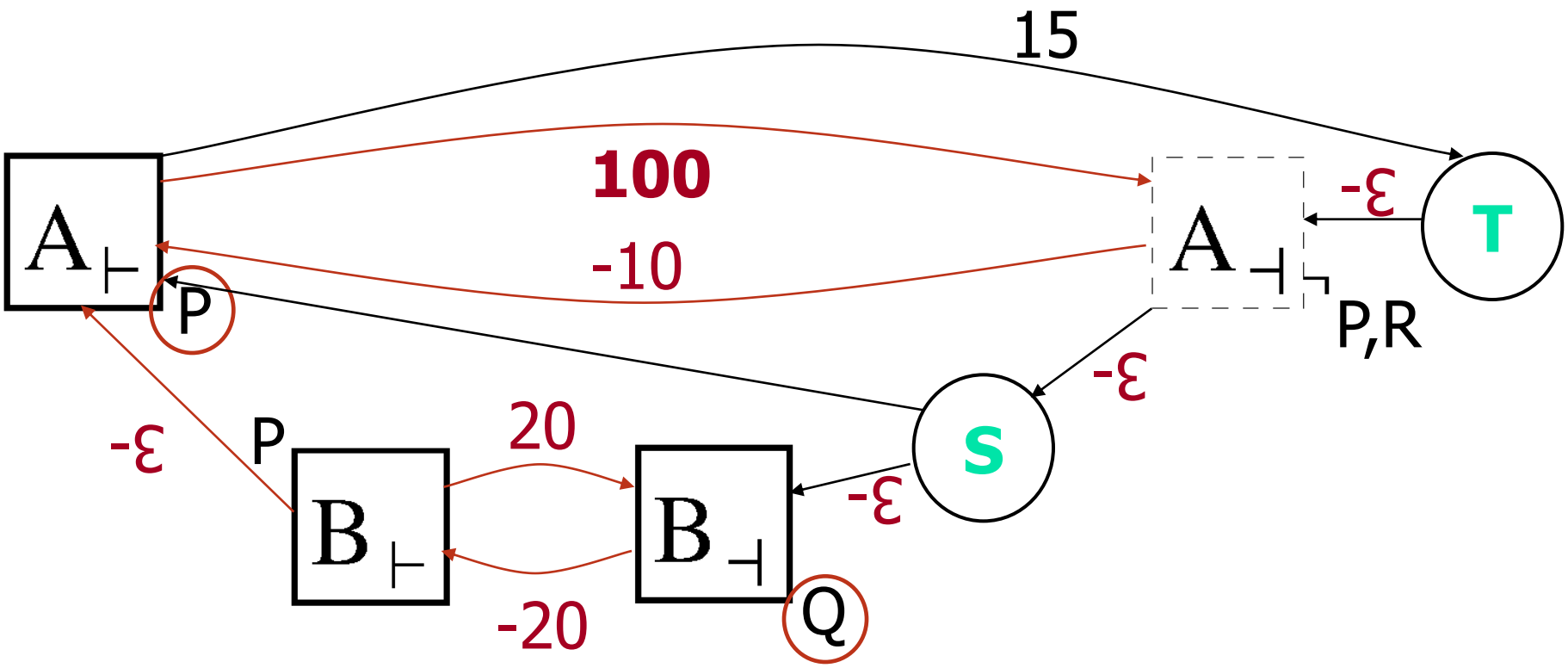
Available from http://tinyurl.com/optic2012

- Built on the code of POPF2
- Temporally expressive problems
- Problems with continuous numeric effects
- PDDL3 preferences
- Net-benefit planning
- Continuous time-dependent costs
- (ADL: Only in preferences)



Enjoy the match later!

Cost trade-off example




```
(preference S (sometime (and (P) (Q))))

(preference T (within 15 (R)))
```

Representation Challenges

- Continuous costs
  - We can express these using a subset of the language PDDL+
  - Should we directly represent continuous costs or discretize them?