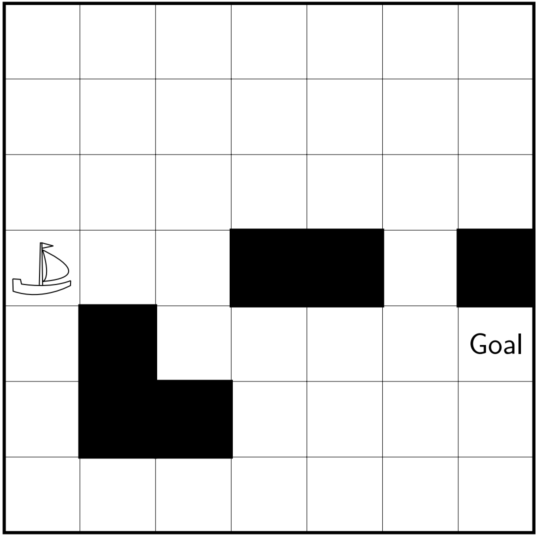
Finding Optimal Deterministic Policies for Constrained Stochastic Shortest Path Problems

Johannes Schmalz, Felipe Trevizan

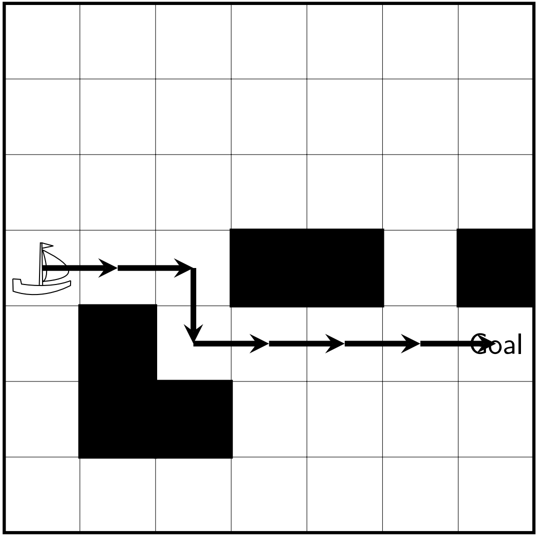
Australian National University

October 30, 2024

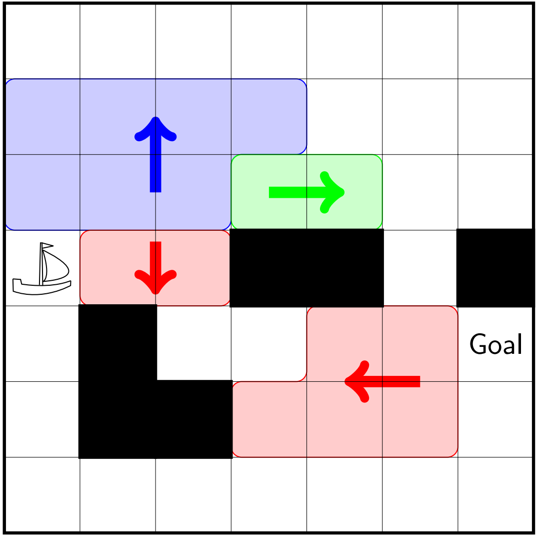
Deterministic Shortest Path Problem



Deterministic Shortest Path Problem

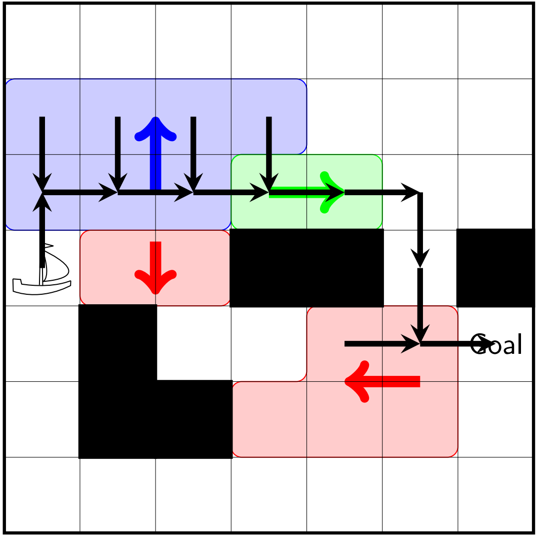


Stochastic Shortest Path Problem (SSP)

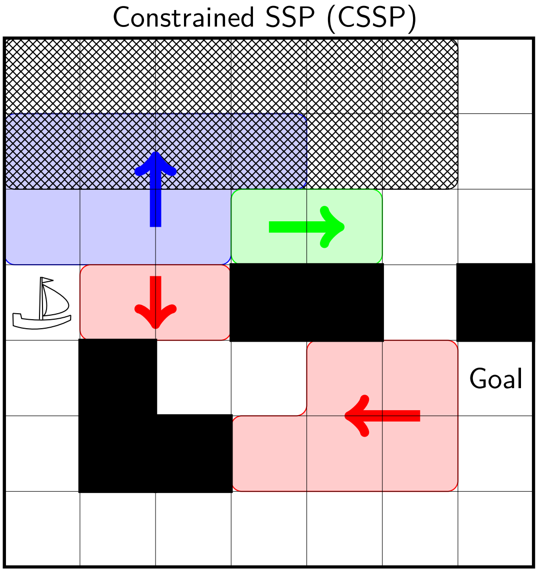


Windy zones have 50% chance to push ship in direction of arrow.

Stochastic Shortest Path Problem (SSP)



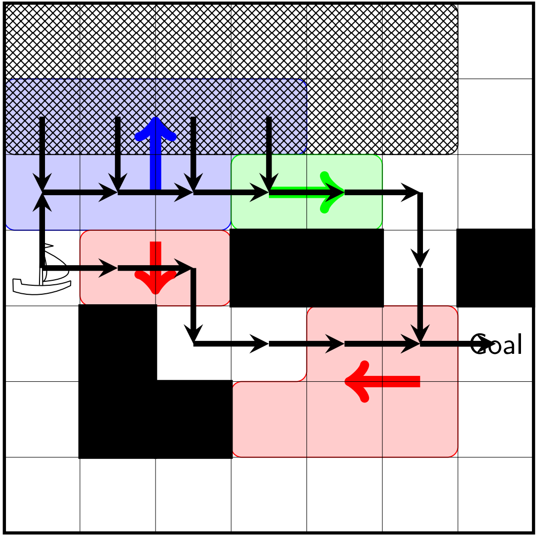
Windy zones have 50% chance to push ship in direction of arrow.



Windy zones have 50% chance to push ship in direction of arrow.

Rough waters damage the ship over time, enter these no more than once (over expectation)

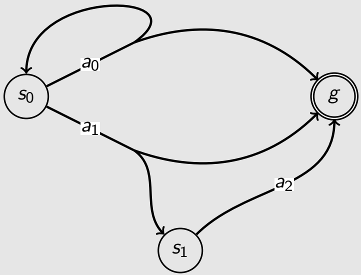
Constrained SSP (CSSP)



Windy zones have 50% chance to push ship in direction of arrow.

Rough waters damage the ship over time, enter these no more than once (over expectation)

Graph



Probabilistic PDDL

```
(:action move-ship
:parameters(...) :precondition(...)
:effect(probabilistic 0.5 (...)
                      0.5 (...)))
```

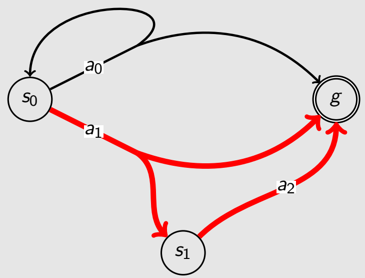
Tuple

SSP $\mathbb{S} = \langle \mathrm{S}, s_0, \mathrm{G}, \mathrm{A}, P, C \rangle$

Probability Transition Matrix

$$\begin{array}{c c}
 & \begin{array}{ccc} s_0 & s_1 & g \end{array} \\
\begin{array}{c} s_0, a_0 \\ s_0, a_1 \\ s_1, a_2 \end{array} & \begin{bmatrix} 0.5 & & 0.5 \\ & 0.5 & 0.5 \\ & & 1 \end{bmatrix}
\end{array}$$

Policy



We want Proper Policies

1. defined for all reachable states (closed)
2. leads to goal with probability 1

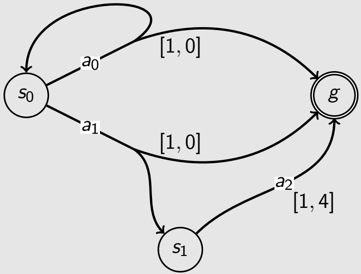
Cost of a Proper Policy

$C(\pi)$ is the cost incurred to reach the goal (over expectation)

Optimal Policy

Should be proper and a minimiser of C.

Graph



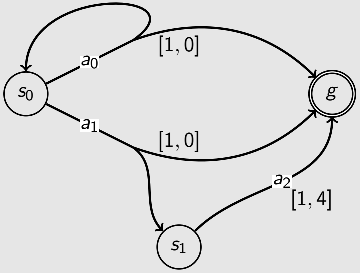
Tuple

CSSP $\mathbb{C} = \langle \mathrm{S}, s_0, \mathrm{G}, \mathrm{A}, P, \vec{C}, \vec{u} \rangle$

- C_0 is the **primary cost**
- C_i are **secondary costs** for $i = 1, 2, \ldots, n$
- u_i is an upper bound on cost function C_i for $i = 1, 2, \ldots, n$

Policy

$u_1 = 1$



$$\pi^*(s_0, a_0) = 0.5 = \pi^*(s_0, a_1)$$

Costs

$$C_0(\pi^*) = 0.5 \cdot 2 + 0.5 \cdot 1.5 = 1.75$$
$$C_1(\pi^*) = 1$$

Costs of a Proper Policy

$C_i(\pi)$ is the incurred cost over C_i to reach the goal (over expectation)

Feasible Proper Policy

π is feasible if $C_i(\pi) \leq u_i$ for all **secondary costs** $i \in \{1, \ldots, n\}$

Optimal Policy

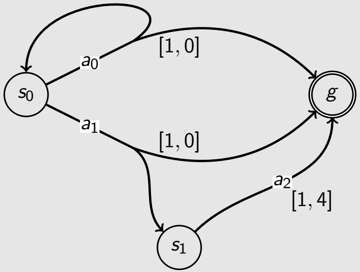
Should be proper and feasible and a minimiser of **primary cost** C_0.

Why stochastic policies?

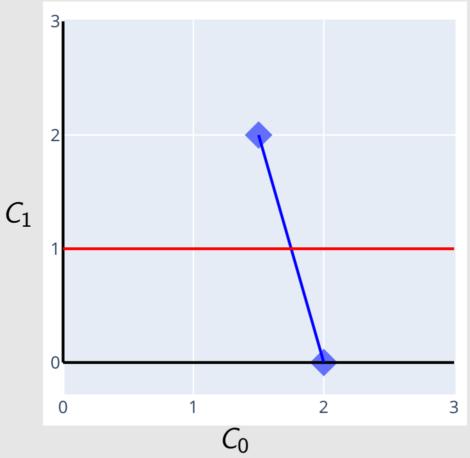
Deterministic Policy Costs

- $C(\pi_0) = [2, 0]$
- $C(\pi_1) = [1.5, 2]$

CSSP



Cost Plot



We can solve CSSPs with Linear Programs (LPs)

Occupation Measure LP (shorthand)

$$\min_{\vec{x}} C_0(\vec{x}) \text{ s.t.}$$
$$out(s) - in(s) = [s = s_0] \qquad \forall s \in \mathsf{S} \setminus \mathsf{G}$$
$$\sum_{g \in \mathsf{G}} in(g) = 1$$
$$x_{s,a} \geq 0 \qquad \forall s \in \mathsf{S}, a \in \mathsf{A}(s)$$
$$C_i(\vec{x}) \leq u_i \qquad \forall i \in \{1, \ldots, n\}$$

Macros

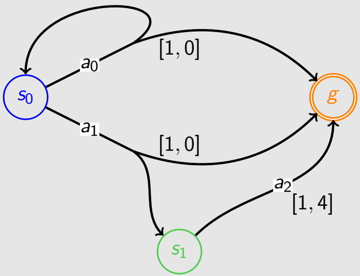
- $C_i(\vec{x}) = \sum_{s \in \mathsf{S}, a \in \mathsf{A}(s)} x_{s,a} C_i(a)$
- $out(s) = \sum_{a \in \mathsf{A}(s)} x_{s,a}$
- $in(s) = \sum_{s' \in \mathsf{S}, a' \in \mathsf{A}(s')} x_{s,a} P(s'|s, a)$

Induced Policy

Given $\vec{x}$, $\pi(s, a) = \frac{x_{s,a}}{in(s)}$

Example CSSP

$u_1 = 1$



Optimal Solution

$$x_{s_0,a_1} = 0.5, \quad x_{s_1,a_2} = 0.25, \quad x_{s_0,a_0} = 1$$

Occupation Measure LP for our Example

$$\min_{\vec{x}} x_{s_0,a_0} + x_{s_0,a_1} + x_{s_1,a_2} \text{ s.t.}$$

$$\vec{x} \geq 0$$

$$\underbrace{x_{s_0,a_0} + x_{s_0,a_1}}_{out(s_0)} - \underbrace{0.5 x_{s_0,a_0}}_{in(s_0)} = 1$$

$$\underbrace{x_{s_1,a_2}}_{out(s_1)} - \underbrace{0.5 x_{s_0,a_1}}_{in(s_1)} = 0$$

$$\underbrace{0.5 x_{s_0,a_0} + 0.5 x_{s_0,a_1} + x_{s_1,a_2}}_{in(g)} = 1$$

$$4 x_{s_1,a_2} \leq 1$$

The story so far

- SSPs
- CSSPs
- Deterministic and Stochastic Policies
- LP for solving CSSPs

Stochastic Policies for CSSPs

- CSSPs always have an optimal stochastic policy
- finding the cheapest is **P**

Deterministic Policies for CSSPs

- CSSP may **not** have an optimal deterministic policy
- finding the cheapest is **NP**

So, who cares about deterministic policies for CSSPs?!

Why do people want deterministic policies for CSSPs?

- ethical issues in medical contexts Roijers et al. (2013)
- aviation regulations Geißer et al. (2020)
- coordination in multi-agent systems Dolgov and Durfee (2005)
- accountability and explainability Hong and Williams (2023)

Occupation Measure MIP (shorthand)

$$\min_{\vec{x},\vec{\Delta}} C_0(\vec{x}) \text{ s.t.}$$

$$out(s) - in(s) = [s = s_0] \qquad \forall s \in \mathrm{S} \setminus \mathrm{G}$$

$$\sum_{g \in \mathrm{G}} in(g) = 1$$

$$x_{s,a} \geq 0 \qquad \forall s \in \mathrm{S}, a \in \mathrm{A}(s)$$

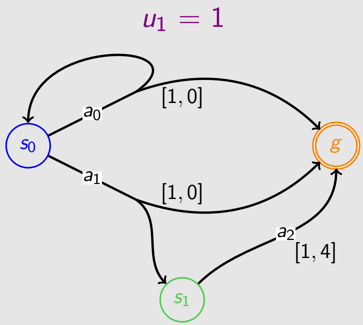
$$C_i(\vec{x}) \leq u_i \qquad \forall i \in \{1, \ldots, n\}$$

$$x_{s,a} \leq M\Delta_{s,a} \qquad \forall s \in \mathrm{S}, a \in \mathrm{A}(s)$$

$$\sum_{a \in \mathrm{A}(s)} \Delta_{s,a} \leq 1 \qquad \forall s \in \mathrm{S}$$

$$\Delta_{s,a} \in \{0, 1\} \qquad \forall s \in \mathrm{S}, a \in \mathrm{A}(s)$$

Example SSP



Optimal Solution

$$x_{s_0,a_1} = 0, \quad x_{s_1,a_2} = 0, \quad x_{s_0,a_0} = 2$$
$$\Delta_{s_0,a_1} = 0, \quad \Delta_{s_1,a_2} = 0, \quad \Delta_{s_0,a_0} = 1$$

OM MIP for our Example

$$\min_{\vec{x}} x_{s_0,a_0} + x_{s_0,a_1} + x_{s_1,a_2} \text{ s.t.}$$

$$\vec{x} \geq 0$$

$$\underbrace{x_{s_0,a_0} + x_{s_0,a_1}}_{out(s_0)} - \underbrace{0.5x_{s_0,a_0}}_{in(s_0)} = 1$$

$$\underbrace{x_{s_1,a_2}}_{out(s_1)} - \underbrace{0.5x_{s_0,a_1}}_{in(s_1)} = 0$$

$$\underbrace{0.5x_{s_0,a_0} + 0.5x_{s_0,a_1} + x_{s_1,a_2}}_{in(g)} = 1$$

$$4x_{s_1,a_2} \leq 1$$

$$x_{s_0,a_0} \leq M\Delta_{s_0,a_0}$$

other indicator constraints

A badly chosen M ruins everything!

If M is too small

- can make whole MIP infeasible
- can make optimal policy infeasible

☞ we had $x_{s_0,a_0} = 2$ and $\Delta_{s_0,a_0} = 1$ need $x_{s_0,a_0} \leq M\Delta_{s_0,a_0}$

- **"Fun" game:** *Is my MIP infeasible because the model is infeasible or M is too small?*

If M is too big

- less efficient
- can introduce numerical instability
- might allow non-integer solutions (trickle flow) ☜ catastrophic!

Good news! There's an automatic way to pick M for Constrained MDPs!

LP to find M

$$\max_{\vec{x}} \sum_{s \in S, a \in A(s)} x_{s,a} \text{ s.t.}$$

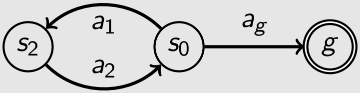
$$out(s) - in(s) = [s = s_0] \qquad \forall s \in S \setminus G$$

$$\sum_{g \in G} in(g) = 1$$

$$x_{s,a} \geq 0 \qquad \forall s \in S, a \in A(s)$$

Bad news, it doesn't work for Constrained SSPs...

Counterexample



A solution:	$x_{s_0,a_g} = 1$	$x_{s_0,a_1} = 1$	$x_{s_2,a_2} = 1$
Another solution:	$x_{s_0,a_g} = 1$	$x_{s_0,a_1} = 10$	$x_{s_2,a_2} = 10$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

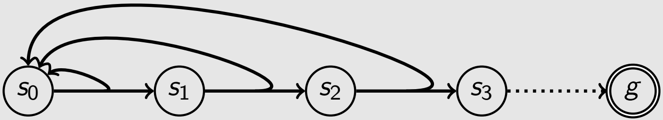
New contribution: $p_{\text{min}}^{-|S \setminus G|} \geq x_{\text{max}}$

Proof Idea

Consider a trajectory from s to a goal. Its probability must be $\geq p_{\text{min}}^{|S \setminus G|}$. So, if there was $x_{\text{max}} > p_{\text{min}}^{-|S \setminus G|}$, then more than a unit of flow reaches the goals!

It's tight

Backwards chaining SSP has $x_{\text{max}} = p_{\text{min}}^{-|S \setminus G|}$



But $|S \setminus G|$ can get very big. . .

New contribution: algorithm for finding M

Insight

- Suppose we have some feasible policy for $\mathbb{C}$.
- Its cost must be $\geq$ optimal cost.
- Relation between x_{max} and cost: $\text{obj}(\vec{x}) \cdot \underline{g}^{-1} \geq x_{\text{max}}$

Algorithm for finding M

1. select some M
2. try to solve MIP with M
   - if infeasible: increase M_0 and repeat step 2
   - if feasible: set $M \leftarrow \text{obj}(\vec{x}) \cdot \underline{g}^{-1}$
3. solve MIP with M

Careful: only works if CSSP is feasible.

New contribution: avoid M completely!

SOS1 Constraints

Specify a set of **continuous** variables $\{x_0, \ldots, x_k\}$.

☞ At most one of these is allowed to be non-zero.

So, we want for each $s \in S$ the SOS1 constraint $\{x_{s,a} | a \in A(s)\}$.

SOS1 are a standard feature, but there are some caveats.

The story so far

- Deterministic policies are worse than stochastic ones for CSSPs
- ... but we want them anyways
- we can use MIPs with M
- New: we can automatically derive M for CSSPs
- New: we can avoid M and use SOS1 constraints

We want heuristics!

Blind Search

The LP and MIP consider whole state space.

Let's move away from the dark age of blind search!

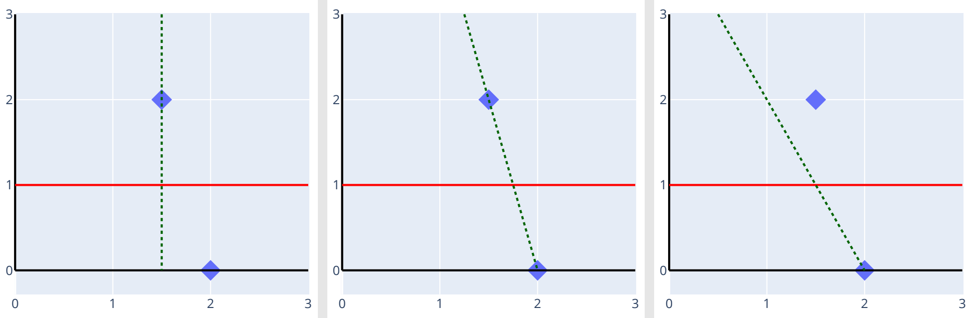
Algorithms

- LAny Hong and Williams (2023) (for deterministic policies)
- i^2-dual Trevizan, Thiébaux, and Haslum (2017) (for stochastic policies)
- new: i^2-dual-det (for deterministic policies)

LAny Hong and Williams (2023)

Step 1

Solve the MIP's Lagrangian dual by solving a sequence of linearised SSPs.

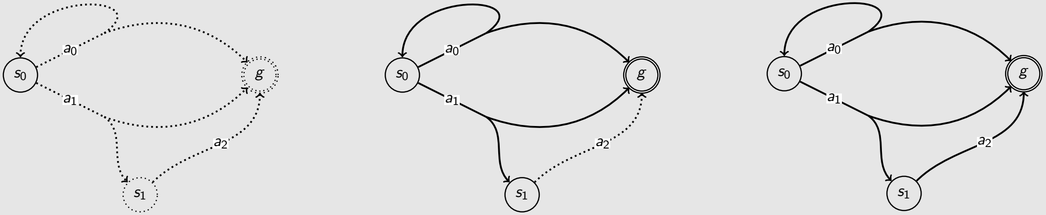


Step 2

Brute force enumerate deterministic policies until you provably find the optimal one. . .

i^2-dual Trevizan, Thiébaux, and Haslum (2017)

Technique 1: Partial SSPs



Technique 2: Atomic Projections

Embedded in the CSSP LP! See original paper for details.

New: i^2-dual-det

☞ drag and drop results for the MIP for deterministic policies into i^2-dual

Teaser: we improve performance by solving intermediate problems partially!

How interesting are the constraints of a CSSP?

New contribution: we present a simple categorisation!

- trivial
- linearisable
- interesting

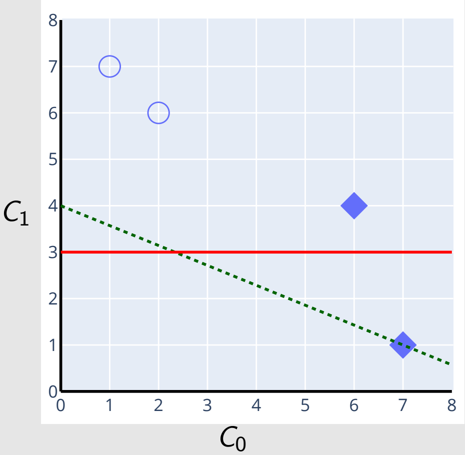
Trivial

The constraints can be ignored.

☞ SSP-relaxation gives feasible solution.

Linearisable

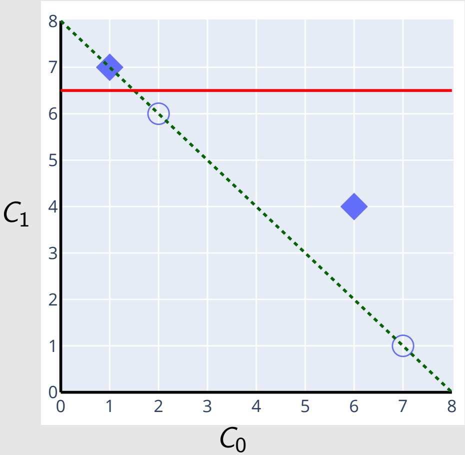
There is some λ so that π^* for λ-SSP is feasible.



$\blacklozenge / \circ$ = feasible / infeasible w.r.t. C_2

Interesting

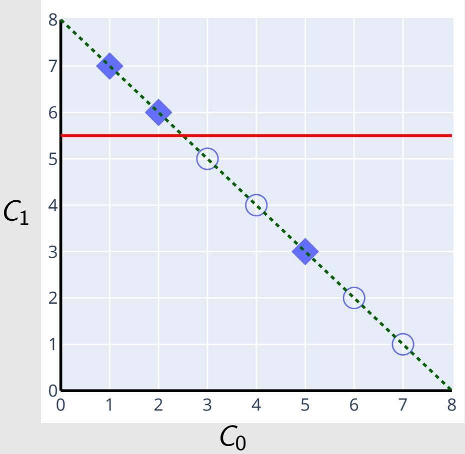
Not linearisable.



♦/○ = feasible / infeasible w.r.t. C_2

Practically Interesting (Unofficial)

Linearisable, but often behaves like interesting.



$\blacklozenge / \circ$ = feasible / infeasible w.r.t. C_2

Results for optimal planners.

Search And Rescue

	MIP 1		LAny		i^2-dual-det-M		i^2-dual-det-SOS1	
problem	cov.	runtime	cov.	runtime	cov.	runtime	cov.	runtime
(4, F)	0		26	142.3±53.9	15	753.2±229.0	25	605.8±235.5
(4, T)	0		26	179.0±99.7	1	494.3	1	309.0
(5, F)	0		30	229.2±71.1	5	264.9±276.4	0	
(5, T)	0		27	249.8±73.8	2	564.4±633.4	0	

Exploding Blocks World

	MIP 1		LAny		i^2-dual-det-M		i^2-dual-det-SOS1	
problem	cov.	runtime	cov.	runtime	cov.	runtime	cov.	runtime
(4, 5, 0.17)	0		30	838.5±42.3	30	77.9±20.2	30	16.6±0.8
(4, 5, 0.2)	0		30	0.2	30	10.6±1.1	30	1.6
(5, 7, 0.1)	0		30	10.9±0.1	25	717.2±139.3	30	19.7±0.2
(7, 8, 0.48)	0		30	1.8±0.1	27	559.6±144.5	30	35.3±2.6

Looks bad. . . but these are all linearisable!

Elevators

	MIP 1		LAny		i^2-dual-det-M		i^2-dual-det-SOS1	
problem	cov.	runtime	cov.	runtime	cov.	runtime	cov.	runtime
(1, 2, 2)	30	9.9±3.5	10	13.6±3.2	30	247.6±84.7	30	68.8±42.8
(2, 1, 1)	30	5.2±0.2	30	9.7±2.3	24	667.2±199.9	30	118.8±34.5
(2, 1, 2)	30	275.9±103.9	29	125.4±17.9	0		0	
(2, 2, 1)	30	196.2±118.3	28	64.5±12.8	1	1260.3	8	527.7±192.1

PARC Printer

	MIP 1		LAny		i^2-dual-det-M		i^2-dual-det-SOS1	
problem	cov.	runtime	cov.	runtime	cov.	runtime	cov.	runtime
(1, 0.0, 1)	0		0		30	32.9±5.9	30	17.0±0.2
(1, 0.0, ∞)	0		30	53.7±0.7	30	17.8±0.1	30	16.7±0.1
(1, 0.1, 1)	0		0		30	69.9±24.2	25	121.6±123.1
(1, 1.0, ∞)	0		30	20.7±0.4	0		0	
(2, 0.0, 1)	0		0		30	32.8±6.2	30	16.7±0.1
(2, 0.0, ∞)	0		30	151.6±1.7	30	17.3±0.1	30	16.1±0.2
(2, 1.0, ∞)	0		30	44.7±0.5	0		0	

Looks mixed. . . we have a mix of problem difficulties!

Constrained Tireworld

	MIP 1		LAny		i^2-dual-det-M		i^2-dual-det-SOS1	
problem	cov.	runtime	cov.	runtime	cov.	runtime	cov.	runtime
(4, 4, 2)	30	371.9±58.0	30	349.0±1.4	30	110.6±16.3	30	29.6±2.1
(4, 4, 4)	30	583.4±63.5	0		29	330.4±101.2	30	44.9±2.3
(4, 4, 6)	18	949.4±91.6	0		29	513.9±79.7	30	57.0±2.1
(4, 5, 2)	14	1621.7±60.6	0		30	385.9±26.7	30	141.8±14.7
(4, 5, 4)	0		0		27	850.2±108.6	30	205.7±11.7
(4, 5, 6)	0		0		25	1021.1±88.2	30	246.1±15.6
(4, 6, 2)	0		0		0		30	1064.3±102.2
(4, 6, 4)	0		0		0		25	1376.7±96.9
(4, 6, 6)	0		0		0		28	1281.5±81.7

Looks much better... this domain is practically interesting!

Summary

- Deterministic policies for CSSPs are required in some cases, even if they are not optimal, and they are harder to find
- We can use MIPs to find deterministic policies, **but there was no automatic way to get** M
- **New:** we present ways to automatically derive M!
- **New:** i^2-dual-det (adapts i^2-dual to find deterministic policies)
- **New:** are CSSP constraints interesting? (trivial, linearisable, interesting)
- i^2-dual-det is the state-of-the-art for interesting problems!

More at `schmlz.github.io/det-pi-for-cssp`

Bonus Facts

- LAny is very bad on infeasible problems, i^2-dual-det is ok

Dolgov, D. A.; and Durfee, E. H. 2005. Stationary Deterministic Policies for Constrained MDPs with Multiple Rewards, Costs, and Discount Factors. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*.

Geißer, F.; Povéda, G.; Trevizan, F.; Bondouy, M.; Teichteil-Königsbuch, F.; and Thiébaux, S. 2020. Optimal and Heuristic Approaches for Constrained Flight Planning under Weather Uncertainty. *Proceedings of the International Conference on Automated Planning and Scheduling*.

Hong, S.; and Williams, B. C. 2023. An anytime algorithm for constrained stochastic shortest path problems with deterministic policies. *Artificial Intelligence*.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of Artificial Intelligence Research*.

Trevizan, F.; Thiébaux, S.; and Haslum, P. 2017. Occupation Measure Heuristics for Probabilistic Planning. In *Proc. of 27th Int. Conf. on Automated Planning and Scheduling (ICAPS)*.