Solving Constrained Stochastic Shortest Path Problems with Scalarisation

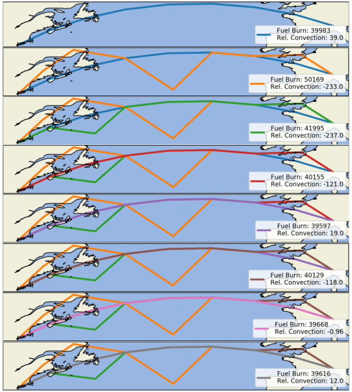
Johannes Schmalz, Felipe Trevizan

UNIVERSITÄT DES SAARLANDES

Australian National University

October 28, 2025

Constrained Stochastic Shortest Path Problems (CSSPs): Posterchild Example



[Geißer et al., 2020]

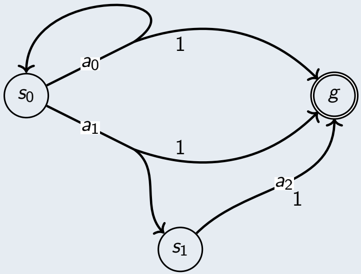
Leal, Sebastian O. (2014, March 29). *Airbus A380 — demo flight at FIDAE 2014* Wikimedia Commons. License: CC BY-SA 3.0. `https://commons.wikimedia.org/wiki/File:Airbus_A380_FIDAE_2014.jpg`

CSSP

- *Shortest Path:* reduce fuel
- *Stochastic:* weather
- *Constrained:* time windows

(Unconstrained) SSPs

Graph



Tuple

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

- S states
- s_0 initial state
- G goals
- A actions
- P probability transition
- C cost

Probabilistic PDDL

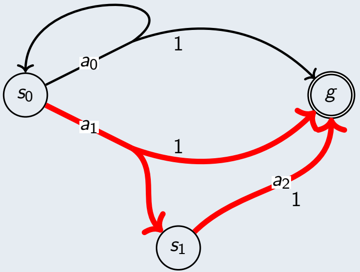
```
(:action move-plane
 :parameters(...) :precondition(...)
 :effect(probabilistic 0.5 (...)
                       0.5 (...)))
```

Probability Transition Matrix for P

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

Policies for (Unconstrained) SSPs

Policy



Optimal Policies

Deterministic $\pi : \mathsf{S} \rightarrow \mathsf{A}$

- lead to goal with probability 1
- minimise cost (over expectation)

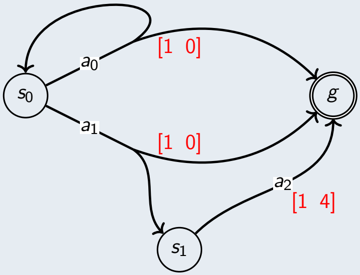
Can be solved efficiently with backup-based heuristic search

- iLAO* [Hansen and Zilberstein, 2001]
- LRTDP [Bonet and Geffner, 2003]
- CG-iLAO* [Schmalz and Trevizan, 2024] ☜ we will use this

Constrained SSPs

Graph

$u_1 = 1$



Tuple

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

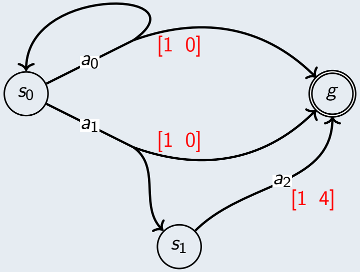
- cost vector $\vec{C} \in \mathbb{R}^{n+1}_{\geq 0}$
- upper bound vector $\vec{u} \in \mathbb{R}^{n}_{\geq 0}$

Policies must satisfy $C_i(\pi) \leq u_i$ for $i = 1, \ldots, n$ (over expectation)

Policies for Constrained SSPs

Graph

$u_1 = 1$



Optimal Policies

- lead to goal with probability 1
- minimise primary cost C_0 (over expectation)
- satisfy $C_i(\pi) \leq u_i$ for secondary costs (over expectation)

May be **stochastic** $\pi : S \to \text{distr}(A)$

Policy

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

Costs:

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

Can be solved efficiently with heuristic search using Linear Programs (LPs)

- i-dual [Trevizan et al., 2016]
- i^2-dual [Trevizan, Thiébaux, and Haslum, 2017]

In this paper: solve CSSPs via a sequence of unconstrained SSPs!

Same approach has been used for similar problems; but not for solving CSSPs optimally

Our algorithm CARL: solves CSSPs via a sequence of SSPs with **scalarisation**

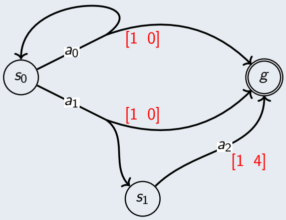
Scalarisation with $\vec{\lambda}$

Takes CSSP $\mathbb{C} = \langle S, s_0, G, A, P, \vec{C}, \vec{u} \rangle$ and $\vec{\lambda} \in \mathbb{R}^n_{\geq 0}$

Gives SSP $\mathbb{S}(\vec{\lambda})$ with scalar cost $C_{\vec{\lambda}} = C_0 + \lambda_1 \big(C_1 - u_1 \big) + \dots$

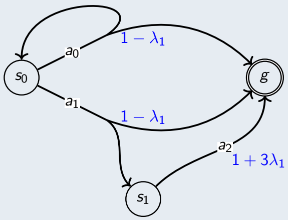
CSSP

$u_1 = 1$



$\longrightarrow$

$\mathbb{S}(\vec{\lambda})$



Our algorithm CARL: solves CSSPs via a sequence of SSPs with **scalarisation**

Scalarisation with $\vec{\lambda}$

Takes CSSP $\mathbb{C} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, \vec{C}, \vec{u} \rangle$ and $\vec{\lambda} \in \mathbb{R}^n_{\geq 0}$

Gives SSP $\mathbb{S}(\vec{\lambda})$ with scalar cost $C_{\vec{\lambda}} = C_0 + \lambda_1 \Big(C_1 - u_1 \Big) + \ldots$

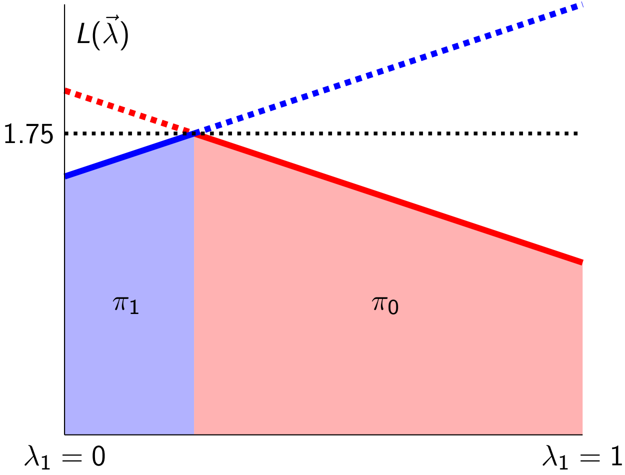
Solving $\mathbb{S}(\vec{\lambda})$

$\mathbb{S}(\vec{\lambda})$ is an SSP

☞ can be solved with SSP algo's, e.g., CG-iLAO* [Schmalz and Trevizan, 2024]

☞ it has an optimal deterministic policy

Let $L(\vec{\lambda}) = \min_\pi C_{\vec{\lambda}}(\pi)$ let's plot $L(\vec{\lambda})$ as $\vec{\lambda}$ varies



For us: $\begin{cases} C_{\vec{\lambda}}(\pi_0) = 2 - \lambda_1 \\ C_{\vec{\lambda}}(\pi_1) = 1.5 + \lambda_1 \end{cases}$

Important Fact

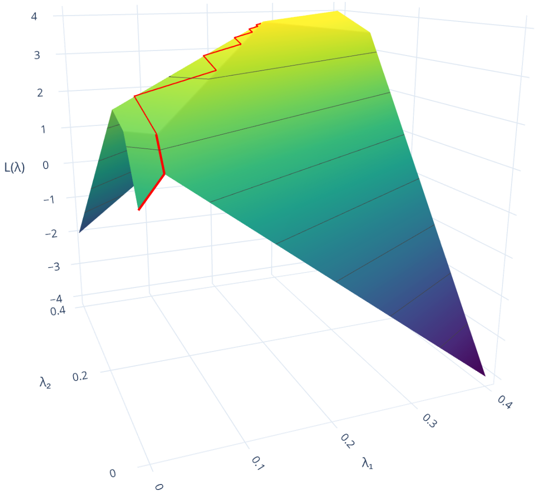
We get the CSSP's opt. policy from $\max_{\vec{\lambda}} L(\vec{\lambda})$ (kind of)

$C_0(\pi^*)$ for CSSP at max

see Lagrangian Duality
e.g., [Hong and Williams, 2023; Lee et al., 2018]

How do we find the max? $C_0(\pi^*) = L(\vec{\lambda}^*) = \max_{\vec{\lambda}} L(\vec{\lambda})$?

Answer: variant of **subgradient descent** (following [Hong and Williams, 2023])



Getting the Subgradient

Solve $\mathbb{S}(\vec{\lambda}) \longrightarrow \pi^*_{\vec{\lambda}}$ is opt. policy

then subgrad. is $[C_1(\pi^*_{\vec{\lambda}}) - u_1 \ \cdots \ C_n(\pi^*_{\vec{\lambda}}) - u_n]$

CARL solves many $\mathbb{S}(\vec{\lambda})$s . . . that needs to be fast!

How can we solve $\mathbb{S}(\vec{\lambda})$s fast?

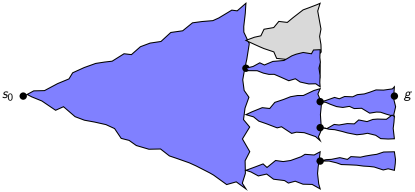
Observation: $\mathbb{S}(\vec{\lambda})$s are similar

Solution: CG-iLAO* with warm starts
(that is, use info. from prev. solns.)

- CG-iLAO* is *old* [Schmalz and Trevizan, 2024]
- warm starts have been considered for iLAO* [Hong and Williams, 2023]
- warm starts for CG-iLAO* are **new**

CG-iLAO* Warm Starts

1. Reuse the search tree



2. Reuse Value Function V

☞ bad: doing this naïvely makes CG-iLAO* non-optimal (V inadmissible)

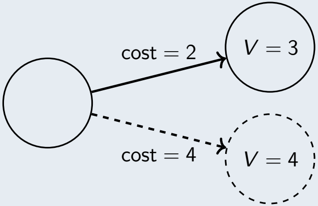
☞ fix: let CG-iLAO* handle it

CG-iLAO*'s Value Function (aka cost-to-go)

Bellman backup

$$V(s) \leftarrow \min_a \underbrace{C(a) + \sum_{s'} P(s'|s,a) \cdot V(s')}_{Q(s,a)}$$

CG-iLAO*'s trick: ignore a if there is a better action $Q_{\min} < Q(s,a)$



$Q(s,a) = 2 + 3 = 5 = Q_{\min}$

$Q(s,a) = 4 + 4 = 8 < Q_{\min}$

Issue: what if V changes and an action becomes important?

Fix: track changes in Γ and check actions whose Q-values affected

Reusing CG-iLAO*'s Value Function

Reuse V

$$V_{\text{old}} = \vec{\lambda}_{\text{old}} \cdot \vec{V} \quad \longrightarrow \quad \vec{\lambda}_{\text{new}} \cdot \vec{V} = V_{\text{new}}$$

Issue: V_{new} might be inadmissible and heuristic search requires admissible V ...

Fix: track changes in Γ and let CG-iLAO* handle it!

Summary of Experiments

CARL is **fast**

- CARL solves 1264 / 1290 problems (next best 808)
- CARL has an average $10\times$ speedup

	CARL		i-dual		i2-dual
	λ-ROC	λ-LMcut	C-ROC	IP-LMcut	
SAR (n, d, r) 4, 3, .75	**30 5.6± 1.4**	**30 4.3± 1.0**	30 104.1± 51.0	30 45.8± 21.1	30 69.2± 56.4
4, 4, .75	**30 41.9± 5.3**	**30 34.1± 4.2**	5 1345.7±534.1	9 979.8±296.6	8 495.5±417.8
5, 3, .50	**30 4.9± 0.9**	**30 4.8± 1.1**	30 25.1± 8.5	30 13.1± 4.6	30 11.1± 4.0
5, 3, .75	**30 12.3± 2.5**	**30 13.3± 2.9**	30 253.8±110.6	30 113.3± 46.0	30 168.0±105.9
5, 4, .50	**30 30.8± 6.8**	**30 36.2± 7.9**	17 695.0±268.5	24 550.6±166.1	27 637.8±213.1
5, 4, .75	**30 126.6±45.2**	**30 142.5±49.3**	6 930.9±369.3	6 499.2±226.4	8 349.1±288.1

PARC (f,u)	CARL		i-dual		i2-dual
	λ-ROC	λ-LMcut	C-ROC	IP-LMcut	
0.0, 1	30 311.2±13.6	30 270.9± 8.3	30 126.2± 9.1	30 103.1±2.4	**30 42.8± 3.1**
0.0, ∞	30 100.7± 0.8	30 141.3± 1.0	30 47.1± 2.5	30 42.1±1.8	**30 29.5± 1.8**
0.2, 1	30 600.7± 3.6	30 863.4± 9.7	0	0	**30 65.1± 11.1**
0.2, ∞	30 121.7± 1.9	30 185.7± 1.2	30 1537.7±72.8	0	**30 41.7± 4.9**
0.4, 1	30 882.8±31.7	**30 564.1±12.8**	0	0	3 801.5±536.2
0.4, ∞	**30 130.9± 2.5**	30 188.1± 1.3	0	0	3 665.9±308.3
0.6, 1	30 318.4±10.3	**30 255.4± 6.9**	0	0	0
0.6, ∞	**30 25.5± 0.5**	30 39.6± 0.3	26 1508.9±68.9	0	0
0.8, 1	30 345.5± 7.4	**30 267.1± 8.0**	0	0	0
0.8, ∞	**30 25.7± 0.6**	30 39.8± 0.3	29 1375.1±89.9	0	0
1.0, 1	**30 208.1± 6.8**	30 226.0± 5.4	0	0	0
1.0, ∞	**30 25.6± 0.6**	30 39.6± 0.3	30 1298.1±55.3	0	0

more domains in paper

Summary

- CARL solves a sequence of SSPs to solve the CSSP via **scalarisation**
- use CG-iLAO* to solve SSPs efficiently with warm starts
- CARL is **fast**: solves 1264 / 1290 (next best 808)

What I didn't cover in the talk

- how do we extract an optimal stochastic policy from the optimal deterministic policies for $\mathbb{S}(\vec{\lambda}^*)$?
- how do we find **all** opt. det. policies for $\mathbb{S}(\vec{\lambda}^*)$?
- how do we get admissible heuristics?
- more details. . .

`schmlz.github.io/carl`

Bonet, B.; and Geffner, H. 2003. Labeled RTDP: Improving the Convergence of Real-Time Dynamic Programming. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 12–21.

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. *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 384–393.

Hansen, E.; and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 35–62.

Hong, S.; and Williams, B. C. 2023. An anytime algorithm for constrained stochastic shortest path problems with deterministic policies. *Artificial Intelligence*, 103846.

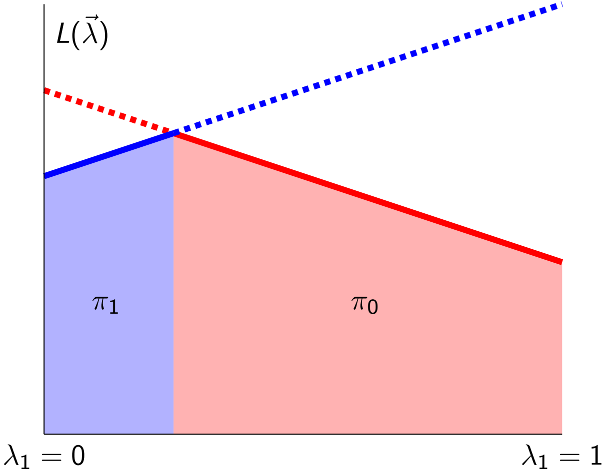
Lee, J.; Kim, G.-h.; Poupart, P.; and Kim, K.-E. 2018. Monte-Carlo Tree Search for Constrained POMDPs. In *Advances in Neural Information Processing Systems (NIPS)*.

Schmalz, J.; and Trevizan, F. 2024. Efficient Constraint Generation for Stochastic Shortest Path Problems. *Proc. of AAAI Conf. on Artificial Intelligence*, 20247–20255.

Trevizan, F.; Thiébaux, S.; Santana, P.; and Williams, B. 2016. Heuristic Search in Dual Space for Constrained Stochastic Shortest Path Problems. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 326–334.

Trevizan, F.; Thiébaux, S.; and Haslum, P. 2017. Occupation Measure Heuristics for Probabilistic Planning. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 306–315.

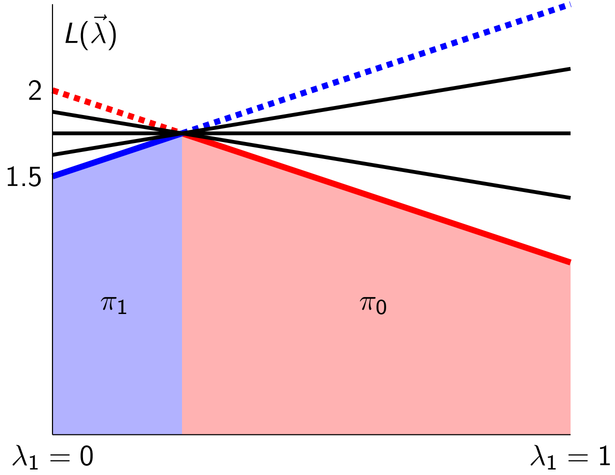
Scalarisation $\vec{\lambda} \longrightarrow$ SSP $\mathbb{S}(\vec{\lambda})$ with $C_{\vec{\lambda}}(\pi) = C_0(\pi) + \lambda_1\Big(C_1(\pi) - u_1\Big) + \ldots$



Facts:

1. gradient $\leq 0 \iff$ constr' ✓
2. $C_0(\pi) =$ intersect vertical axis
3. Stochastic π looks like...

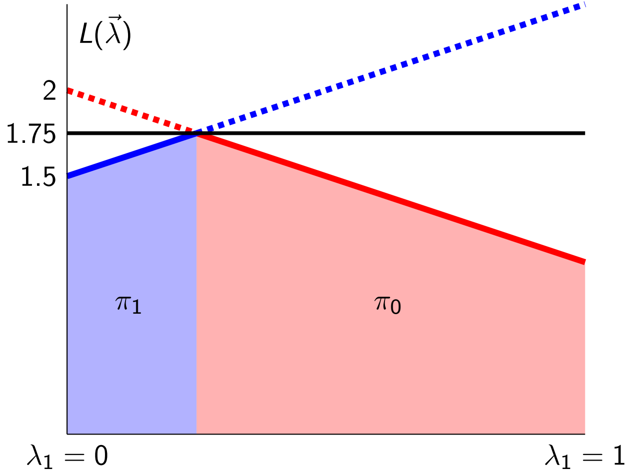
Scalarisation $\vec{\lambda} \longrightarrow$ SSP $\mathbb{S}(\vec{\lambda})$ with $C_{\vec{\lambda}}(\pi) = C_0(\pi) + \lambda_1\Big(C_1(\pi) - u_1\Big) + \ldots$



Facts:

1. gradient $\leq 0 \iff$ constr' ✓
2. $C_0(\pi) =$ intersect vertical axis
3. Stochastic π looks like...

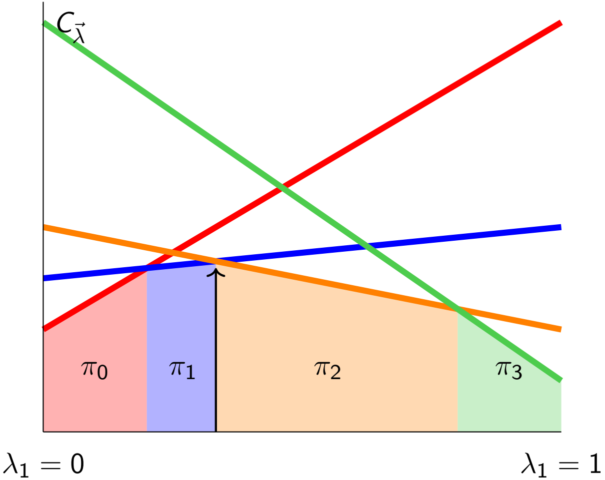
Scalarisation $\vec{\lambda} \longrightarrow$ SSP $\mathbb{S}(\vec{\lambda})$ with $C_{\vec{\lambda}}(\pi) = C_0(\pi) + \lambda_1 \Big(C_1(\pi) - u_1 \Big) + \ldots$

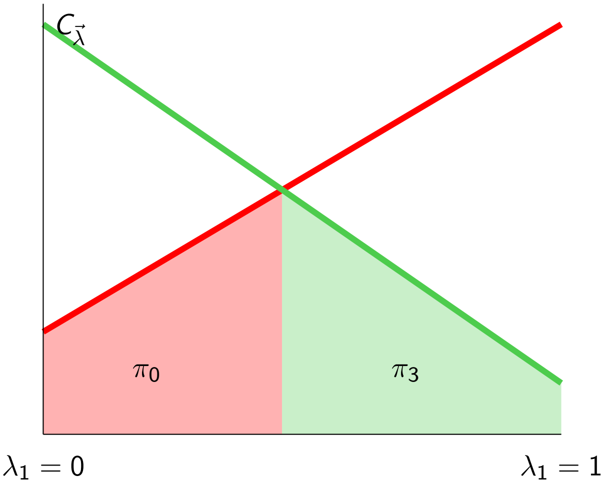


Facts:

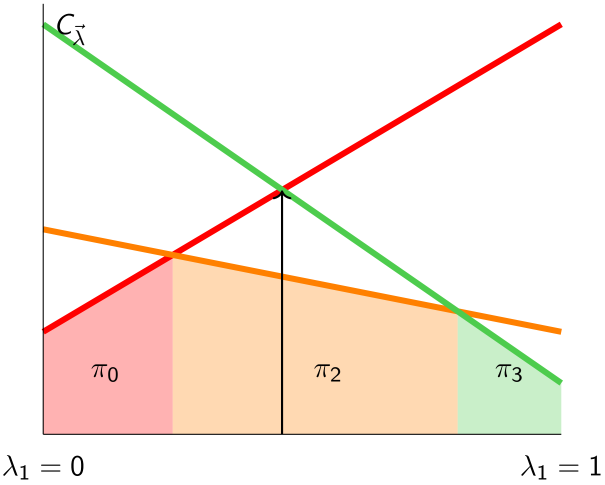
1. gradient $\leq 0 \iff$ constr' ✓
2. $C_0(\pi) =$ intersect vertical axis
3. Stochastic π looks like...

How can we find the max?

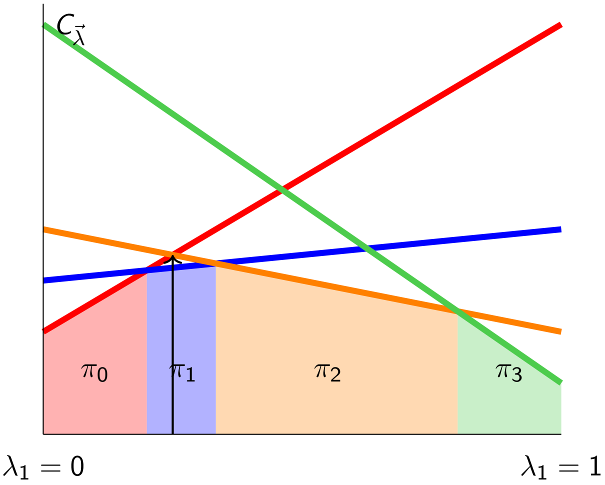


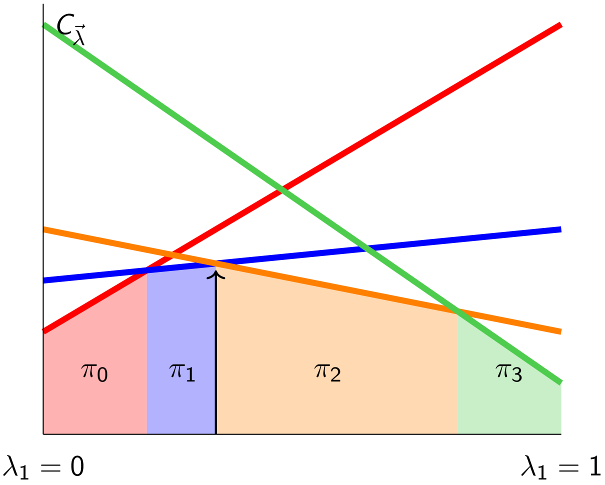


$C_{\vec{\lambda}}$
π_0
π_3
$\lambda_1 = 0$
$\lambda_1 = 1$



$C_{\vec{\lambda}}$
π_0
π_2
π_3
$\lambda_1 = 0$
$\lambda_1 = 1$


$C_{\vec{\lambda}}$
π_0
π_1
π_2
π_3
$\lambda_1 = 0$
$\lambda_1 = 1$


$C_{\vec{\lambda}}$
π_0
π_1
π_2
π_3
$\lambda_1 = 0$
$\lambda_1 = 1$