Contributions algorithmiques au contrôle optimal stochastique à temps discret et horizon infini

Bruno Scherrer

Soutenance d'HDR - 28 juin 2016

Contrôle optimal stochastique à horizon infini

(Puterman, 1994; Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998)

Système dynamique contrôlé avec récompenses:

$$x_0,\ a_0,\ r_0,\ x_1,\ a_1,\ r_1,\ x_2,\ a_2,\ r_2,\ x_3,\ \ldots$$

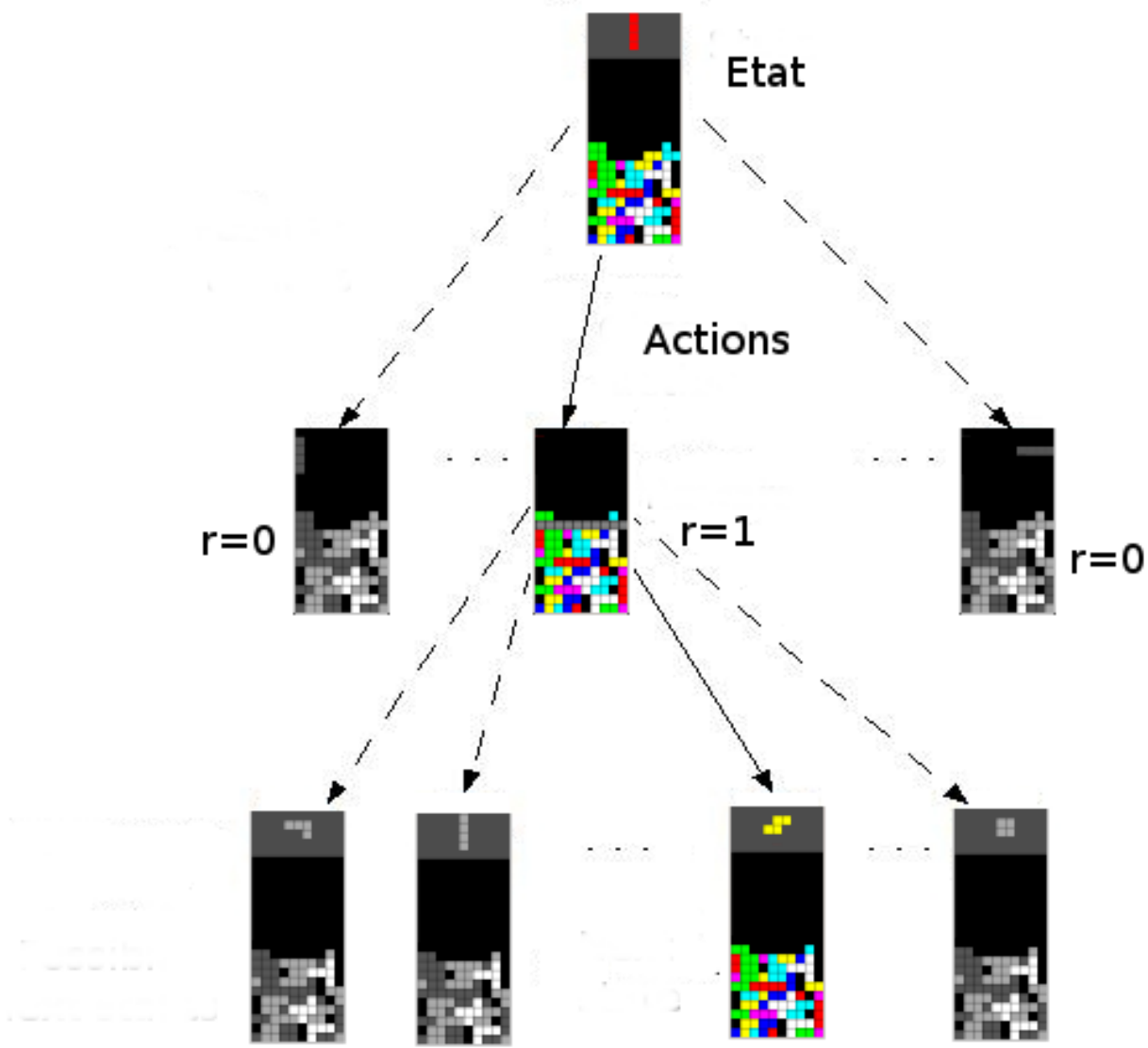
Processus décisionnel de Markov (MDP):

- X, espace d'états (fini ou dénombrable),
- A, espace d'actions (fini ou dénombrable),
- $r : X \times A \to \mathbb{R}$, fonction récompense, $\qquad (r_t = r(x_t, a_t))$
- $p : X \times A \to \Delta_X$, noyau de transition. $\qquad (x_{t+1} \sim p(\cdot|x_t, a_t))$

But: Trouver une politique $\pi : X \to A$ déterministe stationnaire de sorte à maximiser la valeur $v_\pi(x)$ *pour tous* x:

$$v_\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, x_0 = x,\ \{\forall t,\ a_t = \pi(x_t)\}\right]. \qquad (\gamma \in (0,1))$$

Illustration: Tetris



Equations & Operateurs de Bellman

- Pour toute politique π, v_π est solution de **l'équation de Bellman**:

$$\forall x,\ v_\pi(x) = r(x, \pi(x)) + \gamma \sum_{y \in X} p(y|x, \pi(x)) v_\pi(y) \quad \Leftrightarrow \quad v_\pi = T_\pi v_\pi .$$

- La **valeur optimale** v_* est solution de **l'équation d'optimalité de Bellman**:

$$\forall x,\ v_*(x) = \max_{a \in A} \left(r(x, a) + \gamma \sum_{y \in X} p(y|x, a) v_*(y) \right) \quad \Leftrightarrow \quad v_* = T v_* .$$

- $T_\pi : \mathbb{R}^X \to \mathbb{R}^X$ and $T : \mathbb{R}^X \to \mathbb{R}^X$ sont γ-contractants pour la norme infinie $\| \cdot \|_\infty$.
- Pour toute $v \in \mathbb{R}^X$, π est une **politique gloutonne** par rapport à v , noté $\pi = \mathcal{G} v$, ssi

$$\forall x,\ \pi(x) \in \arg\max_{a \in A} \left(r(x, a) + \gamma \sum_{y \in X} p(y|x, a) v(y) \right) \quad \Leftrightarrow \quad T_\pi v = T v .$$

- $\pi_* \in \mathcal{G} v_*$

Algorithmes

Value Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow T v_k = T_{\pi_{k+1}} v_k$

Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow v_{\pi_{k+1}} = (T_{\pi_{k+1}})^{\infty} v_k$

Modified Policy Iteration (Puterman & Shin, 1978)

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow (T_{\pi_{k+1}})^{m} v_k \qquad (1 \leq m \leq \infty)$

Algorithmes

Value Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow T v_k = T_{\pi_{k+1}} v_k$

Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow v_{\pi_{k+1}} = (T_{\pi_{k+1}})^{\infty} v_k$

Modified Policy Iteration (Puterman & Shin, 1978)

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow (T_{\pi_{k+1}})^{m} v_k \qquad (1 \leq m \leq \infty)$

Algorithmes

Value Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow T v_k = T_{\pi_{k+1}} v_k$

Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow v_{\pi_{k+1}} = (T_{\pi_{k+1}})^{\infty} v_k$

Modified Policy Iteration (Puterman & Shin, 1978)

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow (T_{\pi_{k+1}})^{m} v_k \qquad (1 \leq m \leq \infty)$

Plan de l'exposé

1. Complexité de Policy Iteration
2. Classification Based Modified Policy Iteration
3. Politiques non-stationnaires
4. Sur quelques schémas approchés de type PI

Plan de l'exposé

Complexité de Policy Iteration (1)

- $X = \{1, 2, \ldots, n\}$, espace d'états fini,
- $A = \{1, 2, \ldots, m\}$, espace d'actions fini.

Theorème (Scherrer, 2016)

Policy Iteration termine après au plus $O\left(\frac{nm}{1-\gamma} \log\left(\frac{1}{1-\gamma}\right)\right)$ itérations.

Theorème (Scherrer, 2016)

Simplex-PI termine après au plus $O\left(\frac{n^2 m}{1-\gamma} \log\left(\frac{1}{1-\gamma}\right)\right)$ itérations.

- Améliore d'un facteur $O(\log n)$ les précédentes bornes de Policy Iteration (Hansen *et al.* , 2013) et Simplex-PI (Ye, 2011).
- Optimalité: n, m, $\frac{1}{1-\gamma}$? (Fearnley, 2010; Hollanders *et al.* , 2012; Melekopoglou & Condon, 1994).
- Extension aux jeux stochastiques (Akian & Gaubert, 2013)

Complexité de Policy Iteration (2)

Theorème (Scherrer, 2016)

Simplex-PI termine après au plus $O\left(n^3 m^2 \tau_t \tau_r \log^2(n\tau_t\tau_r)\right)$ itérations.

- Généralise le résultat et la preuve de (Post & Ye, 2012) sur les MDPs déterministes (où $\tau_t \leq n$ et $\tau_r \leq n$)

Complexité de Policy Iteration pour un MDP déterministe ?

(entre $\Omega(n^2)$ (Hansen & Zwick, 2010) et $O(\frac{m^n}{n})$ (Hollanders *et al.* , 2015))

Complexité de Policy Iteration (2)

Theorème (Scherrer, 2016)

Simplex-PI termine après au plus $O\left(n^3 m^2 \tau_t \tau_r \log^2(n\tau_t\tau_r)\right)$ itérations.

- Généralise le résultat et la preuve de (Post & Ye, 2012) sur les MDPs déterministes (où $\tau_t \leq n$ et $\tau_r \leq n$)

Complexité de Policy Iteration pour un MDP déterministe ?

(entre $\Omega(n^2)$ (Hansen & Zwick, 2010) et $O(\frac{m^n}{n})$ (Hollanders *et al.* , 2015))

Plan de l'exposé

Programmation dynamique approchée

- $[(T_\pi)^m v](x)$ approché par Monte-Carlo:

$$[(T_\pi)^m v](x) = \mathbb{E}\left[\sum_{t=0}^{m-1} \gamma^t r(x_t, a_t) + \gamma^m v(x_m) \,\middle|\, x_0 = x, \{\forall t, a_t = \pi(x_t)\}\right]$$

- "$v(\cdot) \leftarrow [Au](\cdot)$" approché par régression:

$$\min_{v \in \mathcal{F} \subset \mathbb{R}^X} \sum_x \mu(x) |v(x) - [Au](x)|^p, \quad p = 1, 2, \ldots$$

- $\pi(\cdot) \leftarrow [\mathcal{G}f](\cdot)$" approché par classification (à coût sensitif)

$$\min_{\pi \in \Pi \subset A^X} \sum_x \mu(x) \left(\max_a [T_a f](x) - [T_\pi f](x)\right)$$

Classification-based MPI

(Scherrer, Ghavamzadeh, Gabillon, Lesner, Geist, 2015)

- $''v_k \leftarrow (T_{\pi_k})^m v_{k-1}\ ''$
- $''\pi_{k+1} \leftarrow \mathcal{G}\left[(T_{\pi_k})^m v_{k-1}\right]\ ''$

(v_k) vivant dans $\mathcal{F} \subseteq \mathbb{R}^X$
(π_k) vivant dans $\Pi \subseteq A^X$

Erreur pour l'étape d'évaluation : ϵ_k

$$v_k \leftarrow (T_{\pi_k})^m v_{k-1} + \epsilon_k$$

Erreur pour l'étape gloutonne : ϵ'_k

$$\pi_{k+1} = \mathcal{G}_{\epsilon'_{k+1}}(T_{\pi_k})^m v_{k-1},$$

où pour tout π,

$$T_\pi (T_{\pi_k})^m v_{k-1} \leq T_{\pi_{k+1}} (T_{\pi_k})^m v_{k-1} + \epsilon'_k$$

Propagation des erreurs pour CBMPI

Theorème (Scherrer, Ghavamzadeh, Gabillon, Lesner, Geist, 2015)

Après k itérations, la perte satisfait

$$\|v_* - v_{\pi_k}\|_\infty \leq \frac{2\,\gamma^m\,(\gamma - \gamma^{k-1})}{(1-\gamma)^2} \sup_{1\leq j\leq k-1} \|\epsilon_j\|_\infty + \frac{(1-\gamma^k)}{(1-\gamma)^2} \sup_{1\leq j\leq k} \|\epsilon'_j\|_\infty + O(\gamma^k),$$

- Généralise les bornes pour AVI ($m = 1$) et API ($m = \infty$) de (Bertsekas & Tsitsiklis, 1996).
- m contrôle l'influence de l'erreur sur la fonction valeur.
- Résultats empiriques très bons sur Tetris (20.10^6 lignes)

Plan de l'exposé

Algorithmes

App. Value Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow T v_k + \epsilon_k = T_{\pi_{k+1}} v_k + \epsilon_k$

App. Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow v_{\pi_{k+1}} = (T_{\pi_{k+1}})^{\infty} v_k + \epsilon_k$

App. Modified Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

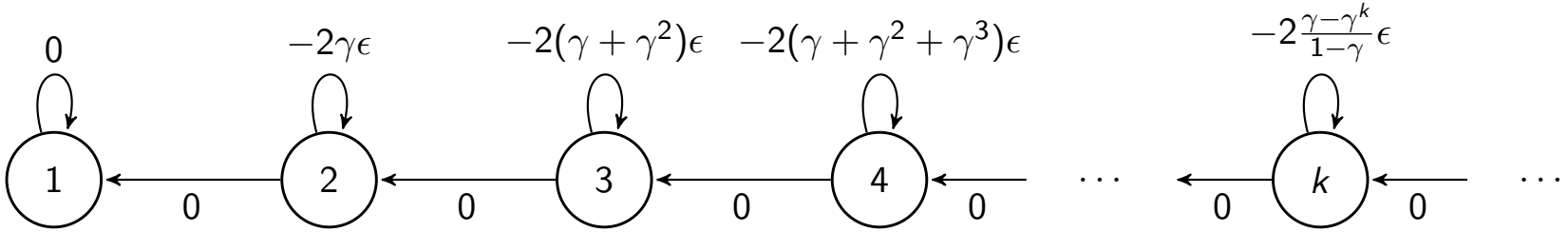
$v_{k+1} \leftarrow (T_{\pi_{k+1}})^{m} v_k + \epsilon_k \qquad (1 \leq m \leq \infty)$

Théorème (Singh & Yee, 1994; Gordon, 1995; Bertsekas & Tsitsiklis, 1996) (Scherrer, Ghavamzadeh, Gabillon, Lesner, Geist, 2015)

Supposons $\|\epsilon_k\|_\infty \leq \epsilon$. La *perte* satisfait

$$\limsup_{k \to \infty} \|v_* - v_{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \epsilon.$$

Optimalité de la borne pour AVI



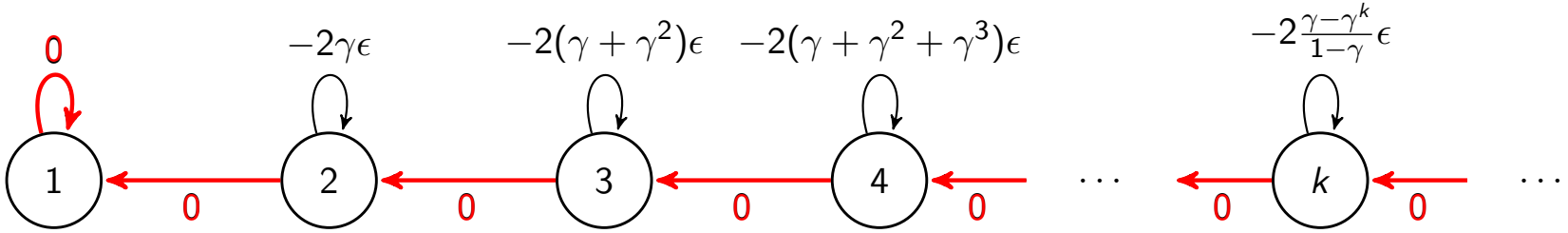
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \overset{k\to\infty}{\longrightarrow} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



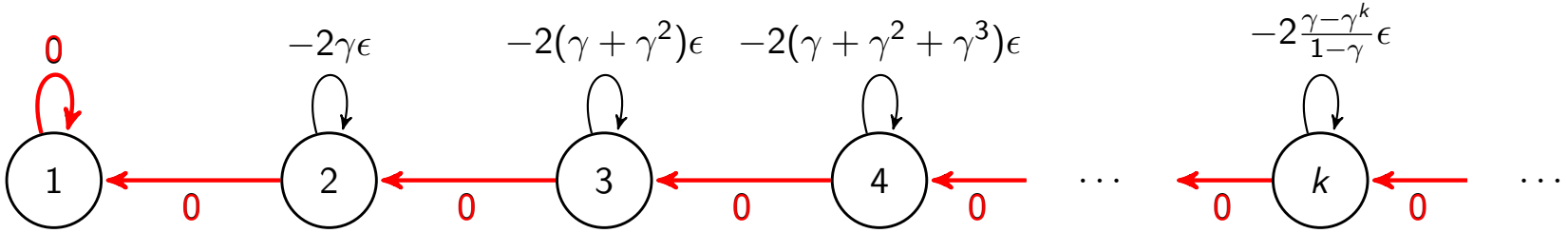
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



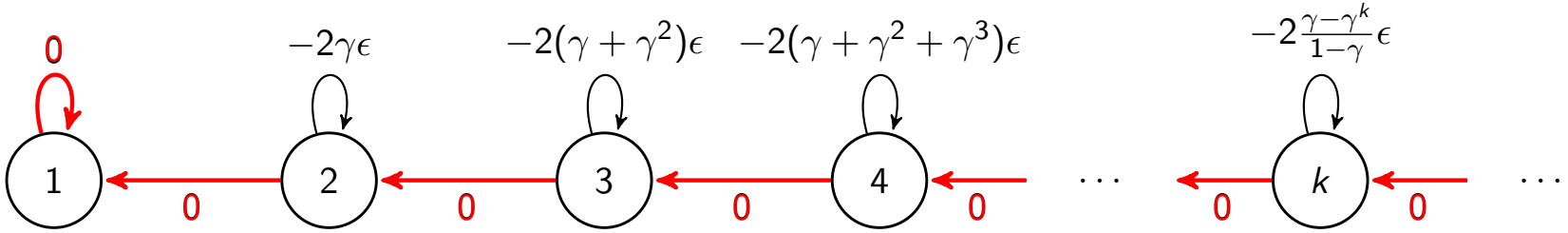
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



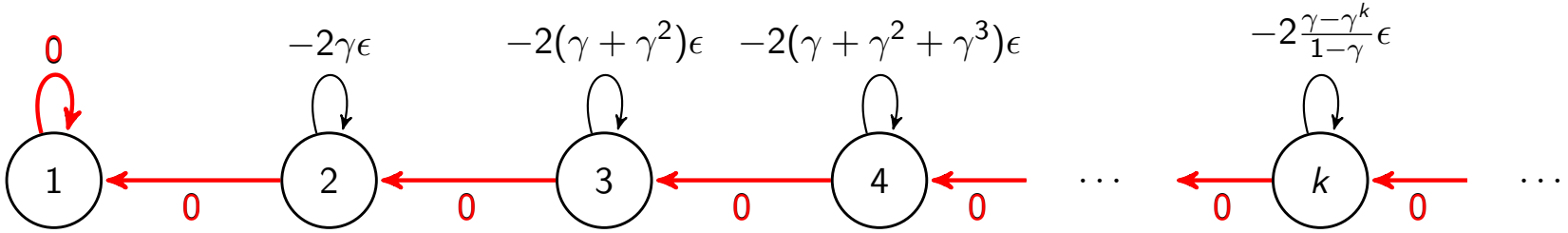
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



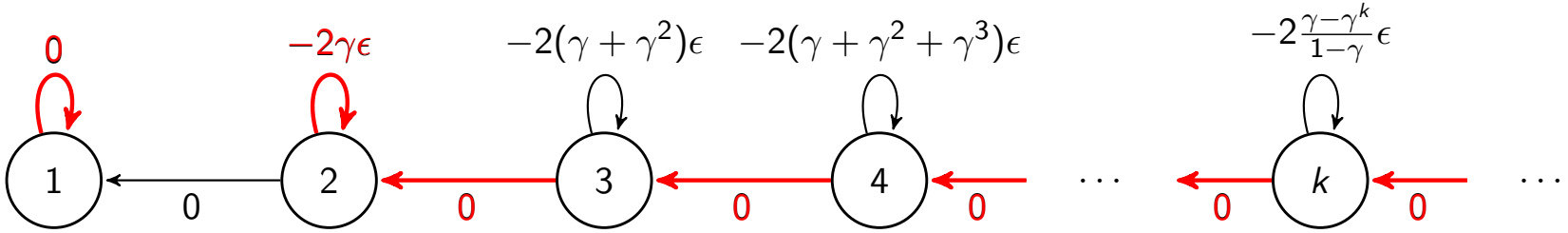
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



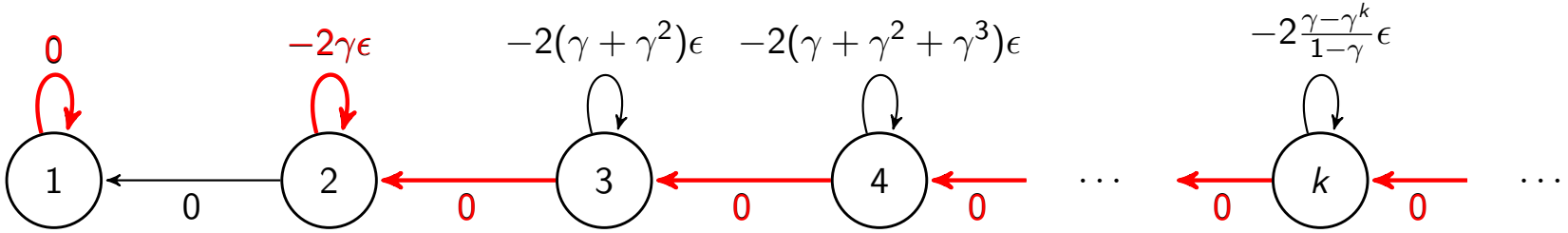
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



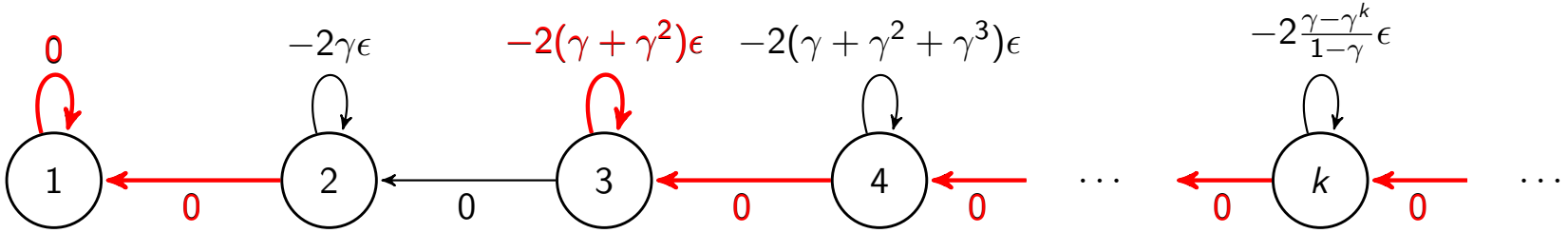
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



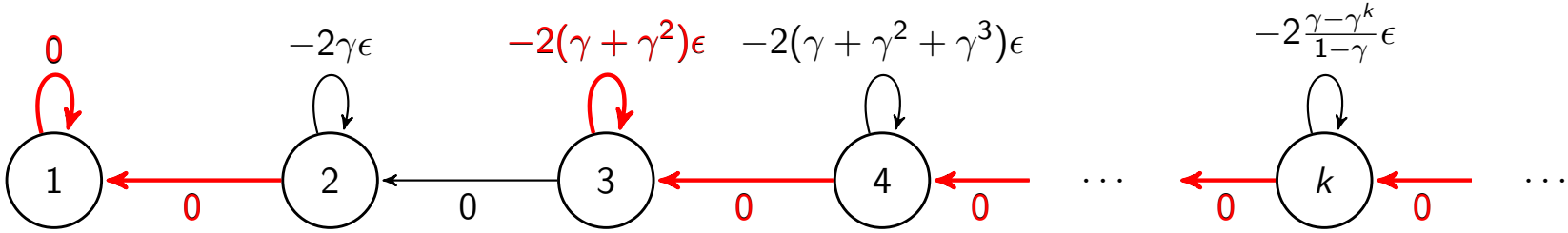
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



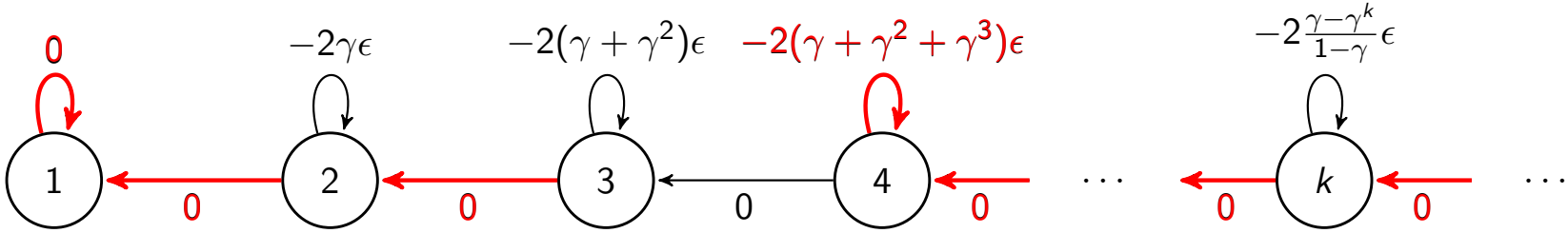
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



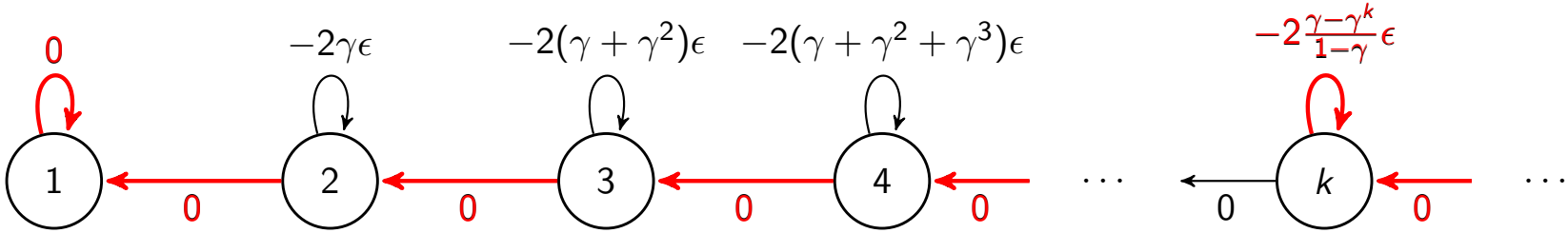
	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Optimalité de la borne pour AVI



	1	2	3	4	...
v_0	0	0	0	0	...
v_1	$-\epsilon$	ϵ	0	0	...
v_2	$-\gamma\epsilon$	$-\epsilon-\gamma\epsilon$	$\epsilon+\gamma\epsilon$	0	...
v_3	$-\gamma^2\epsilon$	$-\gamma^2\epsilon$	$-\epsilon-\gamma\epsilon-\gamma^2\epsilon$	$\epsilon+\gamma\epsilon+\gamma^2\epsilon$	...
...	...	...	...	...	...

Etat 2: $0+\gamma(-\epsilon)=-2\gamma\epsilon+\gamma\epsilon$

Etat 3: $0+\gamma(-\epsilon-\gamma\epsilon)=-2(\gamma+\gamma^2)\epsilon+\gamma(\epsilon+\gamma\epsilon)$

$$v_{\pi_k}(k)=\sum_{t=0}^{\infty}\gamma^t\left(-2\frac{\gamma-\gamma^k}{1-\gamma}\epsilon\right)=-2\frac{\gamma-\gamma^k}{(1-\gamma)^2}\epsilon \xrightarrow{k\to\infty} -\frac{2\gamma}{(1-\gamma)^2}\epsilon$$

Non-Stationary Value Iteration

AVI produit une séquence de valeurs/politiques ($\pi_{i+1} \in \mathcal{G} v_i$)

$$\begin{array}{cccccccc} v_0 & v_1 & v_2 & \cdots & v_{k-\ell} & \cdots & v_{k-2} & v_{k-1} \\ \pi_1 & \pi_2 & \pi_3 & \cdots & \pi_{k-\ell+1} & \cdots & \pi_{k-1} & \pi_k \end{array}$$

Idée: Utiliser la politique non-stationnaire périodique:

$$(\sigma_{k,\ell})^\infty = \underbrace{\pi_k \ \pi_{k-1} \ \cdots \ \pi_{k-\ell+1}}_{\sigma_{k,\ell}\text{: } \ell \text{ dernières politiques}} \ \underbrace{\pi_k \ \pi_{k-1} \ \cdots \pi_{k-\ell+1}}_{\sigma_{k,\ell}\text{: } \ell \text{ dernières politiques}} \ \cdots\cdots$$

Théorème (Scherrer & Lesner, 2012)

Supposons $\|\epsilon_k\|_\infty \leq \epsilon$. Pour tout ℓ, la perte satisfait

$$\limsup_{k\to\infty} \|v_* - v_{(\sigma_{k,\ell})^\infty}\|_\infty \leq \frac{2\gamma}{(1-\gamma^\ell)(1-\gamma)}\epsilon.$$

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow v_{(\sigma_{k+1,\ell})^\infty} + \epsilon_k \quad$ (avec $v_{k+1} \simeq T_{\sigma_{k+1,\ell}} v_{k+1}$)

où $(\sigma_{0,\ell})^\infty = \pi_0\ \pi_{-1}\ \ldots\ \pi_{-\ell+1}\ \pi_0\ \pi_{-1}\ \ldots\ \pi_{-\ell+1}\ \ldots$
est arbitraire et

$$\forall v, \quad T_{\sigma_{k,\ell}} v = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_{k-\ell+1}} v.$$

Sortie en fonction de k:

$$\begin{aligned}
(\sigma_0^\ell)^\infty &= (\pi_0\ \pi_{-1}\ \ldots\ \pi_{-\ell+2}\ \pi_{-\ell+1})^\infty \\
(\sigma_1^\ell)^\infty &= (\pi_1\ \pi_0\ \ldots\ \pi_{-\ell+3}\ \pi_{-\ell+2})^\infty \\
(\sigma_2^\ell)^\infty &= (\pi_2\ \pi_1\ \ldots\ \pi_{-\ell+4}\ \pi_{-\ell+3})^\infty \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
(\sigma_k^\ell)^\infty &= (\pi_k\ \pi_{k-1}\ \ldots\ \pi_{k-\ell+2}\ \pi_{k-\ell+1})^\infty
\end{aligned}$$

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G} v_k$

$v_{k+1} \leftarrow v_{(\sigma_{k+1,\ell})^\infty} + \epsilon_k \qquad (\text{avec } v_{k+1} \simeq T_{\sigma_{k+1,\ell}} v_{k+1})$

où $(\sigma_{0,\ell})^\infty = \pi_0\ \pi_{-1}\ \ldots\ \pi_{-\ell+1}\ \pi_0\ \pi_{-1}\ \ldots\ \pi_{-\ell+1}\ \ldots$
est arbitraire et

$$\forall v, \quad T_{\sigma_{k,\ell}} v = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_{k-\ell+1}} v.$$

Théorème (Scherrer & Lesner, 2012)

Supposons $\|\epsilon_k\|_\infty \leq \epsilon$. La perte satisfait

$$\limsup_{k\to\infty} \|v_* - v_{(\sigma_{k,\ell})^\infty}\|_\infty \leq \frac{2\gamma}{(1-\gamma^\ell)(1-\gamma)}\epsilon.$$

Non Stationary Modified Policy Iteration

NS Value Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$
$v_{k+1} \leftarrow T_{\pi_{k+1}} v_k + \epsilon_k$

NS Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$
$v_{k+1} \leftarrow (T_{\sigma_{k+1,\ell}})^{\infty} T_{\pi_{k+1}} v_k + \epsilon_k$

NS Modified Policy Iteration

$\pi_{k+1} \leftarrow \mathcal{G} v_k$
$v_{k+1} \leftarrow (T_{\sigma_{k+1,\ell}})^{m} T_{\pi_{k+1}} v_k + \epsilon_k \qquad (0 \leq m \leq \infty)$

Théorème (Lesner & Scherrer, 2015)

Supposons $\|\epsilon_k\|_\infty \leq \epsilon$. La perte satisfait

$$\limsup_{k \to \infty} \|v_* - v_{(\sigma_{k,\ell})^\infty}\|_\infty \leq \frac{2\gamma}{(1-\gamma^\ell)(1-\gamma)} \epsilon.$$

Confirmation empirique de l'analyse
Optimalité de la constante $\frac{2\gamma}{(1-\gamma^\ell)(1-\gamma)}$

Plan de l'exposé

Constantes de concentrabilité

- Les bornes de performances sont en fait (Munos, 2003; Munos & Szepesvári, 2008)

$$\limsup_{k\to\infty} \|v_* - v_{(\sigma_{k,\ell})^\infty}\|_{1,\nu} \leq \frac{2C\gamma}{(1-\gamma^\ell)(1-\gamma)} \max_k \|\epsilon_k\|_{1,\mu}.$$

où

$$C = (1-\gamma)(1-\gamma^\ell) \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \gamma^{i+j\ell} c(i+j\ell+k)$$

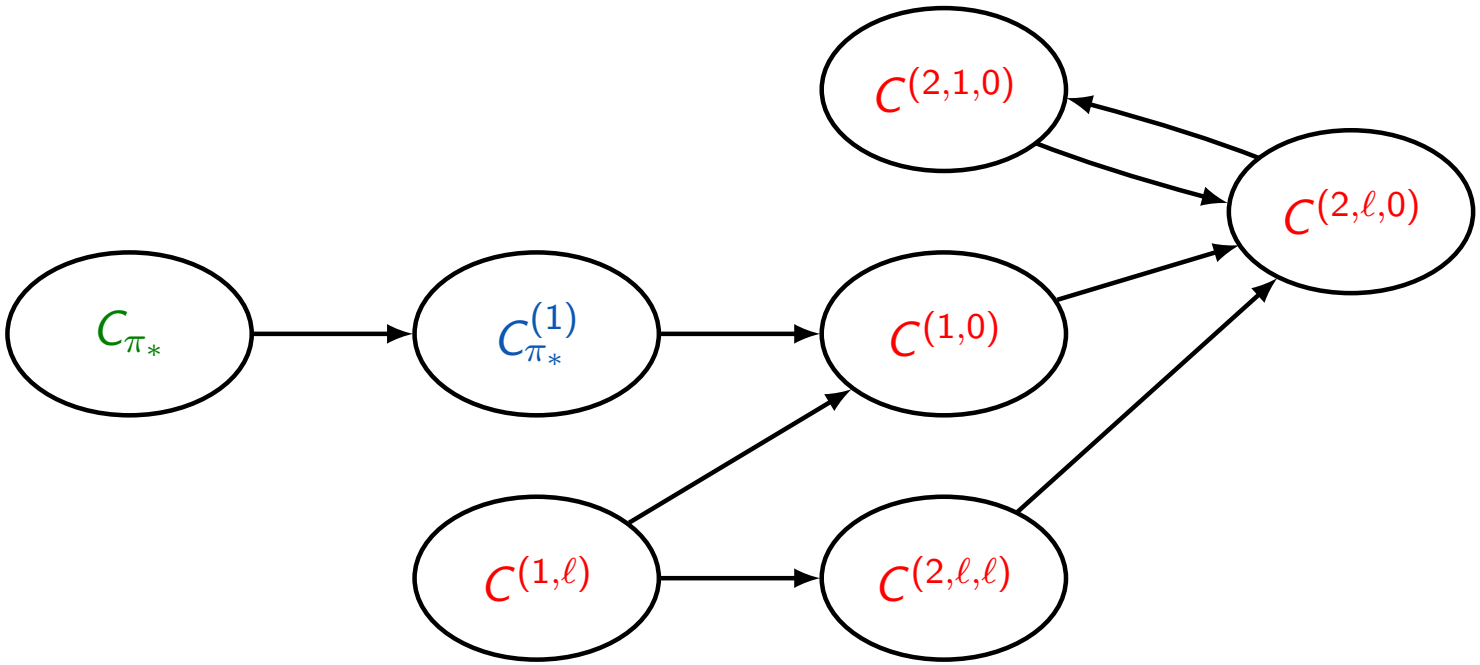
et $c(i) = \max_{\pi_1,\pi_2,\ldots,\pi_i} \left\| \frac{\mu P_{\pi_1} P_{\pi_2} \ldots P_{\pi_i}}{\nu} \right\|_{1,\mu}$.

Analyse (1/2) (Scherrer, 2014)

Algorithme	Perte en norme $l_{1,\mu}$			# Itér.	Mémoire
API	$C^{(2,1,0)}$	$\frac{1}{(1-\gamma)^2}$	ϵ	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	1
	$C^{(1,0)}$	$\frac{1}{(1-\gamma)^2}$	$\epsilon\log\frac{1}{\epsilon}$		
API(α)	$C^{(1,0)}$	$\frac{1}{(1-\gamma)^2}$	ϵ	$\frac{1}{\alpha(1-\gamma)}\log\frac{1}{\epsilon}$	
CPI(α)	$C^{(1,0)}$	$\frac{1}{(1-\gamma)^3}$	ϵ	$\frac{1}{\alpha(1-\gamma)}\log\frac{1}{\epsilon}$	
CPI	$C^{(1,0)}$	$\frac{1}{(1-\gamma)^3}$	$\epsilon\log\frac{1}{\epsilon}$	$\frac{1}{1-\gamma}\frac{1}{\epsilon}\log\frac{1}{\epsilon}$	
	C_{π_*}	$\frac{1}{(1-\gamma)^2}$	ϵ	$\frac{\gamma}{\epsilon^2}$	
PSDP$_\infty$	C_{π_*}	$\frac{1}{(1-\gamma)^2}$	$\epsilon\log\frac{1}{\epsilon}$	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	
	$C^{(1)}_{\pi_*}$	$\frac{1}{1-\gamma}$	ϵ	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	
NSPI(ℓ)	$C^{(2,\ell,0)}$	$\frac{1}{(1-\gamma)(1-\gamma^\ell)}$	ϵ	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	ℓ
	$\frac{C^{(1,0)}}{\ell}$	$\frac{1}{(1-\gamma)^2(1-\gamma^\ell)}$	$\epsilon\log\frac{1}{\epsilon}$	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	
	$C^{(1)}_{\pi_*}+\gamma^\ell\frac{C^{(2,\ell,m)}}{1-\gamma^\ell}$	$\frac{1}{1-\gamma}$	ϵ	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	
	$C_{\pi_*}+\gamma^\ell\frac{C^{(2,\ell,0)}}{\ell(1-\gamma^\ell)}$	$\frac{1}{(1-\gamma)^2}$	$\epsilon\log\frac{1}{\epsilon}$	$\frac{1}{1-\gamma}\log\frac{1}{\epsilon}$	

CPI (Kakade & Langford, 2002), PSDP (Bagnell et al., 2003)

Analyse (2/2): Hiérarchie des constantes



$$A \to B \quad \textit{ssi} \quad \{B < \infty \Rightarrow A < \infty\}$$

Analyse (1'/2)

Algorithme	Perte en norme $l_{1,\mu}$			# Itér.	Mémoire
API	$C^{(1,0)}$	$\frac{1}{(1-\gamma)^2}$	$\epsilon \log \frac{1}{\epsilon}$	$\frac{1}{1-\gamma} \log \frac{1}{\epsilon}$	1
CPI	C_{π_*}	$\frac{1}{(1-\gamma)^2}$	ϵ	$\frac{\gamma}{\epsilon^2}$	
PSDP$_\infty$	C_{π_*}	$\frac{1}{(1-\gamma)^2}$	$\epsilon \log \frac{1}{\epsilon}$	$\frac{1}{1-\gamma} \log \frac{1}{\epsilon}$	
NSPI(ℓ)	$C_{\pi_*} + \gamma^\ell \frac{C^{(2,\ell,0)}}{\ell(1-\gamma^\ell)}$	$\frac{1}{(1-\gamma)^2}$	$\epsilon \log \frac{1}{\epsilon}$	$\frac{1}{1-\gamma} \log \frac{1}{\epsilon}$	ℓ

- CPI arbitrairement meilleur que API, mais avec exponentiellement plus d'itérations
- PSDP$_\infty$ jouit du meilleur des deux mondes
- CPI et PSDP$_\infty$ peuvent requérir beaucoup de mémoire
  $\Rightarrow$ NSPI(ℓ) permet de faire un compromis qualité/mémoire

Confirmation empirique de l'analyse

Existe-t-il un algorithme avec performance $\frac{C_{\pi_*}}{1-\gamma}\epsilon$?

Analyse (1'/2)

Algorithme	Perte en norme $l_{1,\mu}$			# Itér.	Mémoire
API	$C^{(1,0)}$	$\frac{1}{(1-\gamma)^2}$	$\epsilon \log \frac{1}{\epsilon}$	$\frac{1}{1-\gamma} \log \frac{1}{\epsilon}$	1
CPI	C_{π_*}	$\frac{1}{(1-\gamma)^2}$	ϵ	$\frac{\gamma}{\epsilon^2}$	
PSDP_∞	C_{π_*}	$\frac{1}{(1-\gamma)^2}$	$\epsilon \log \frac{1}{\epsilon}$	$\frac{1}{1-\gamma} \log \frac{1}{\epsilon}$	
NSPI(ℓ)	$C_{\pi_*} + \gamma^\ell \frac{C^{(2,\ell,0)}}{\ell(1-\gamma^\ell)}$	$\frac{1}{(1-\gamma)^2}$	$\epsilon \log \frac{1}{\epsilon}$	$\frac{1}{1-\gamma} \log \frac{1}{\epsilon}$	ℓ

- CPI arbitrairement meilleur que API, mais avec exponentiellement plus d'itérations
- PSDP_∞ jouit du meilleur des deux mondes
- CPI et PSDP_∞ peuvent requérir beaucoup de mémoire
  $\Rightarrow$ NSPI(ℓ) permet de faire un compromis qualité/mémoire

Confirmation empirique de l'analyse

Existe-t-il un algorithme avec performance $\frac{C_{\pi_*}}{1-\gamma}\epsilon$?

References I

Akian, M., & Gaubert, S. 2013 (10).
Policy iteration for perfect information stochastic mean payoff games with bounded first return times is strongly polynomial.
Tech. rept. arxiv 1310.4953v1.

Archibald, T., McKinnon, K., & Thomas, L. 1995.
On the generation of Markov decision processes.
Journal of the operational research society, **46**, 354–361.

Bertsekas, D.P., & Tsitsiklis, J.N. 1996.
Neurodynamic Programming.
Athena Scientific.

Fearnley, J. 2010.
Exponential lower bounds for policy iteration.
Pages 551–562 of: 37th international colloquium conference on automata, languages and programming: Part ii.
ICALP'10.
Berlin, Heidelberg: Springer-Verlag.

Gordon, G.J. 1995.
Stable function approximation in dynamic programming.
Pages 261–268 of: International conference on machine learning.

Hansen, T.D., & Zwick, U. 2010.
Lower bounds for Howard's algorithm for finding minimum mean-cost cycles.
Pages 415–426 of: Isaac (1).

References II

Hansen, T.D., Miltersen, P.B., & Zwick, U. 2013.
Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor.
J. acm, **60**(1), 1:1–1:16.

Hollanders, R., Delvenne, J.C., & Jungers, R. 2012.
The complexity of policy iteration is exponential for discounted markov decision processes.
In: Ieee conference on decision and control.

Hollanders, R., Gerenczér, B., Delvenne, J.C., & Jungers, R. 2015.
About upper bounds on the complexity of policy iteration.
Operations research letters.
To appear.

Lesner, B., & Scherrer, B. 2015 (July).
Non-Stationary Approximate Modified Policy Iteration.
In: ICML 2015.

Melekopoglou, M., & Condon, A. 1994.
On the complexity of the policy improvement algorithm for Markov decision processes.
Informs journal on computing, **6**(2), 188–192.

Munos, R. 2003.
Error bounds for approximate policy iteration.
In: International Conference on Machine Learning.

Munos, R., & Szepesvári, Cs. 2008.
Finite-time bounds for fitted value iteration.
Journal of machine learning research, **9**, 815–857.

References III

Post, I., & Ye, Y. 2012.
The simplex method is strongly polynomial for deterministic Markov decision processes.
Tech. rept. arXiv:1208.5083v2.

Puterman, M. 1994.
Markov Decision Processes.
Wiley, New York.

Puterman, M., & Shin, M. 1978.
Modified policy iteration algorithms for discounted Markov decision problems.
Management science, **24**(11).

Saad, Y. 2003.
Iterative methods for sparse linear systems, 2nd edition.
Philadelpha, PA: SIAM.

Scherrer, B. 2014 (June).
Approximate Policy Iteration Schemes: A Comparison.
In: ICML - 31st International Conference on Machine Learning - 2014.

Scherrer, B. 2016.
Improved and Generalized Upper Bounds on the Complexity of Policy Iteration.
Mathematics of operations research.
A paraître.

Scherrer, B., & Lesner, B. 2012 (Dec.).
On the use of non-stationary policies for stationary infinite-horizon Markov decision processes.
In: Neural Information Processing Systems.

References IV

Schoknecht, R. 2002.
Optimality of reinforcement learning algorithms with linear function approximation.
Pages 1555–1562 of: Neural Information Processing Systems.

Singh, S., & Yee, R. 1994.
An Upper Bound on the Loss from Approximate Optimal-Value Functions.
Machine learning, **16-3**, 227–233.

Sutton, R.S., & Barto, A.G. 1998.
Reinforcement learning: An introduction.
MIT Press.

Ye, Y. 2011.
The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate.
Math. oper. res., **36**(4), 593–603.

Yu, H., & Bertsekas, D.P. 2010.
Error bounds for approximations from projected linear equations.
Mathematics of operations research, **35**(2), 306–329.

Illustration of approximation on Tetris

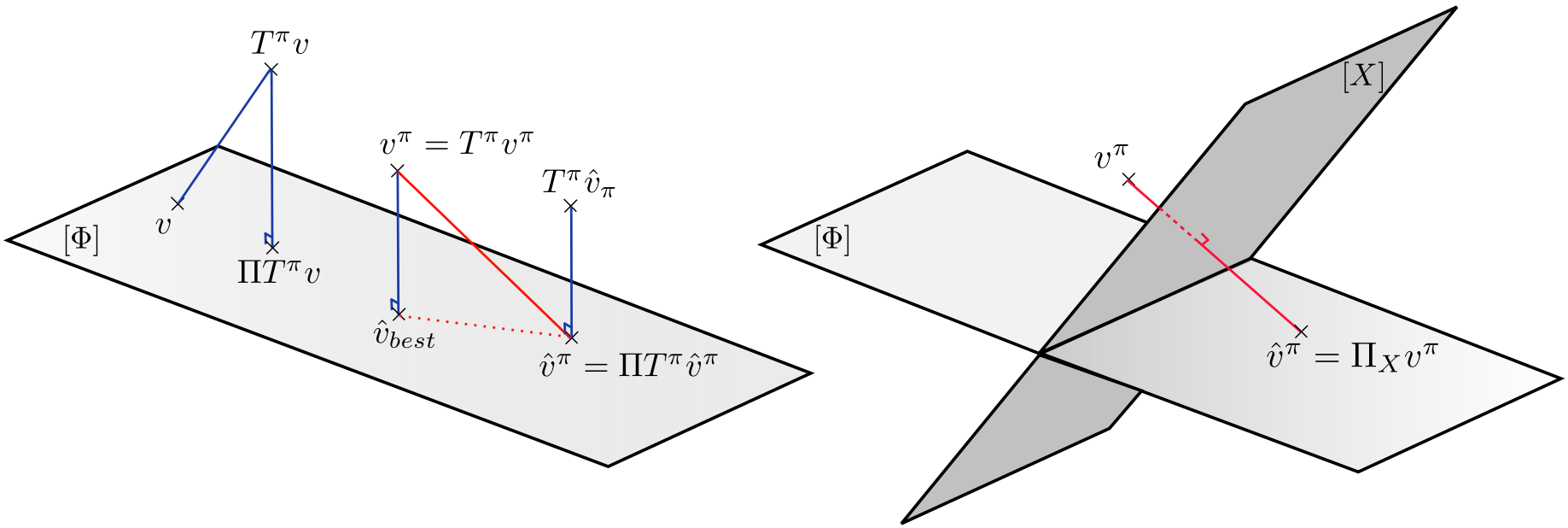
1. Approximation architecture for the value and for the score (on which the policy is based)

$$
\begin{aligned}
f_\theta(x) = \theta_0 & && \text{Constant} \\
& + \theta_1 h_1(x) + \theta_2 h_2(x) + \cdots + \theta_{10} h_{10}(x) && \text{column height} \\
& + \theta_{11} \Delta h_1(x) + \theta_{12} \Delta h_2(x) + \cdots + \theta_{19} \Delta h_9(x) && \text{height variation} \\
& + \theta_{20} \max_k h_k(x) && \text{max height} \\
& + \theta_{21} L(x) && \#\ \text{holes}
\end{aligned}
$$

2. Sampling Scheme: play games

Projected Bellman Equation

- Solve $\hat{v}^\pi = \Pi T^\pi \hat{v}^\pi$ instead of $v^\pi = T^\pi v^\pi$ (?)



$$\|\hat{v}^\pi - v^\pi\| \leq \|\Pi_X\| \|v^\pi - \hat{v}_{best}\| \quad \text{(?)}$$

- Revisit of the analyses of (Schoknecht, 2002) and (Yu & Bertsekas, 2010) in terms of **oblique projection** (Significant simplification)
- Connections with PDE Numerical Analysis (Saad, 2003)

LSTD(λ)

- v is the unique solution of the **Bellman equation**:

$$\forall x,\ v(x) = r(x) + \gamma \sum_y p(y|x)v(y) \quad \Leftrightarrow \quad v_\pi = T v_\pi$$

$$\Leftrightarrow \quad v = r + \gamma P v \quad \Leftrightarrow \quad v = (I - \gamma P)^{-1} r.$$

- Equivalently, for all λ, v is the unique solution of

$$v = T_\lambda v \stackrel{def}{=} (1-\lambda) \sum_{k=0}^{\infty} \lambda^k T^{k+1} v$$

$$= (I - \lambda\gamma P)^{-1}(r + (1-\lambda)\gamma P v)$$

- Look for a linear approximation $\hat{v}(i) = \sum_{j=1}^{d} w_j \phi_j(i)$ or $\hat{v} = \Phi w$

$$\Phi = \begin{pmatrix} \phi(1)' \\ \vdots \\ \phi(N)' \end{pmatrix} = (\underbrace{\phi_1 \quad \dots \quad \phi_d}_{\text{linearly independent}}) \quad \text{and} \quad w = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix}$$

LSTD(λ)

- v is the unique solution of the **Bellman equation**:

$$\forall x,\ v(x) = r(x) + \gamma \sum_y p(y|x)v(y) \quad \Leftrightarrow \quad v_\pi = T v_\pi$$

$$\Leftrightarrow \quad v = r + \gamma P v \quad \Leftrightarrow \quad v = (I - \gamma P)^{-1} r.$$

- Equivalently, for all λ, v is the unique solution of

$$v = T_\lambda v \stackrel{def}{=} (1-\lambda) \sum_{k=0}^{\infty} \lambda^k T^{k+1} v$$

$$= (I - \lambda\gamma P)^{-1}(r + (1-\lambda)\gamma P v)$$

- Look for a linear approximation $\hat{v}(i) = \sum_{j=1}^d w_j \phi_j(i)$ or $\hat{v} = \Phi w$

$$\Phi = \begin{pmatrix} \phi(1)' \\ \vdots \\ \phi(N)' \end{pmatrix} = (\underbrace{\phi_1 \quad \dots \quad \phi_d}_{\text{linearly independent}}) \text{ and } w = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix}$$

LSTD(λ)

- v is the unique solution of the **Bellman equation**:

$$\forall x,\ v(x) = r(x) + \gamma \sum_y p(y|x)v(y) \quad \Leftrightarrow \quad v_\pi = T v_\pi$$

$$\Leftrightarrow \quad v = r + \gamma P v \quad \Leftrightarrow \quad v = (I - \gamma P)^{-1} r.$$

- Equivalently, for all λ, v is the unique solution of

$$v = T_\lambda v \stackrel{def}{=} (1-\lambda) \sum_{k=0}^{\infty} \lambda^k T^{k+1} v$$

$$= (I - \lambda\gamma P)^{-1}(r + (1-\lambda)\gamma P v)$$

- Look for a linear approximation $\hat{v}(i) = \sum_{j=1}^d w_j \phi_j(i)$ or $\hat{v} = \Phi w$

$$\Phi = \begin{pmatrix} \phi(1)' \\ \vdots \\ \phi(N)' \end{pmatrix} = (\underbrace{\phi_1 \quad \dots \quad \phi_d}_{\text{linearly independent}}) \text{ and } w = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix}$$

LSTD(λ) - Main result

Théorème

Let $\|\Phi\|_\infty \leq L$. *Let* ν *be the smallest eigenvalue of* $\Phi' D_\mu \Phi$. *Let* $X_1 \sim \mu$. *Then, for all* $\delta \in (0,1)$, *with probability at least* $1-\delta$, *for all* $n \geq n_0(\delta)$, *A is invertible and:*

$$\|v_\lambda - \hat{v}_\lambda\|_\mu \leq$$

$$\frac{4 V_{max} d L^2}{\sqrt{n-1}(1-\gamma)\nu} \sqrt{\left(1 + \left\lceil \frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)} \right\rceil\right) \sqrt{I(n-1,\delta)} + h(n,\delta)}$$

with $I(n,\delta) = \tilde{O}\left(\log\left(\frac{1}{\delta}\right)\log(n)\right)^\alpha$ *and* $h(n,\delta) = \tilde{O}\left(\frac{\log\left(\frac{1}{\delta}\right)}{n}\right)$.

$\nu > 0$ if the features are linearly independent.

α depends on the β-mixing properties of the process.

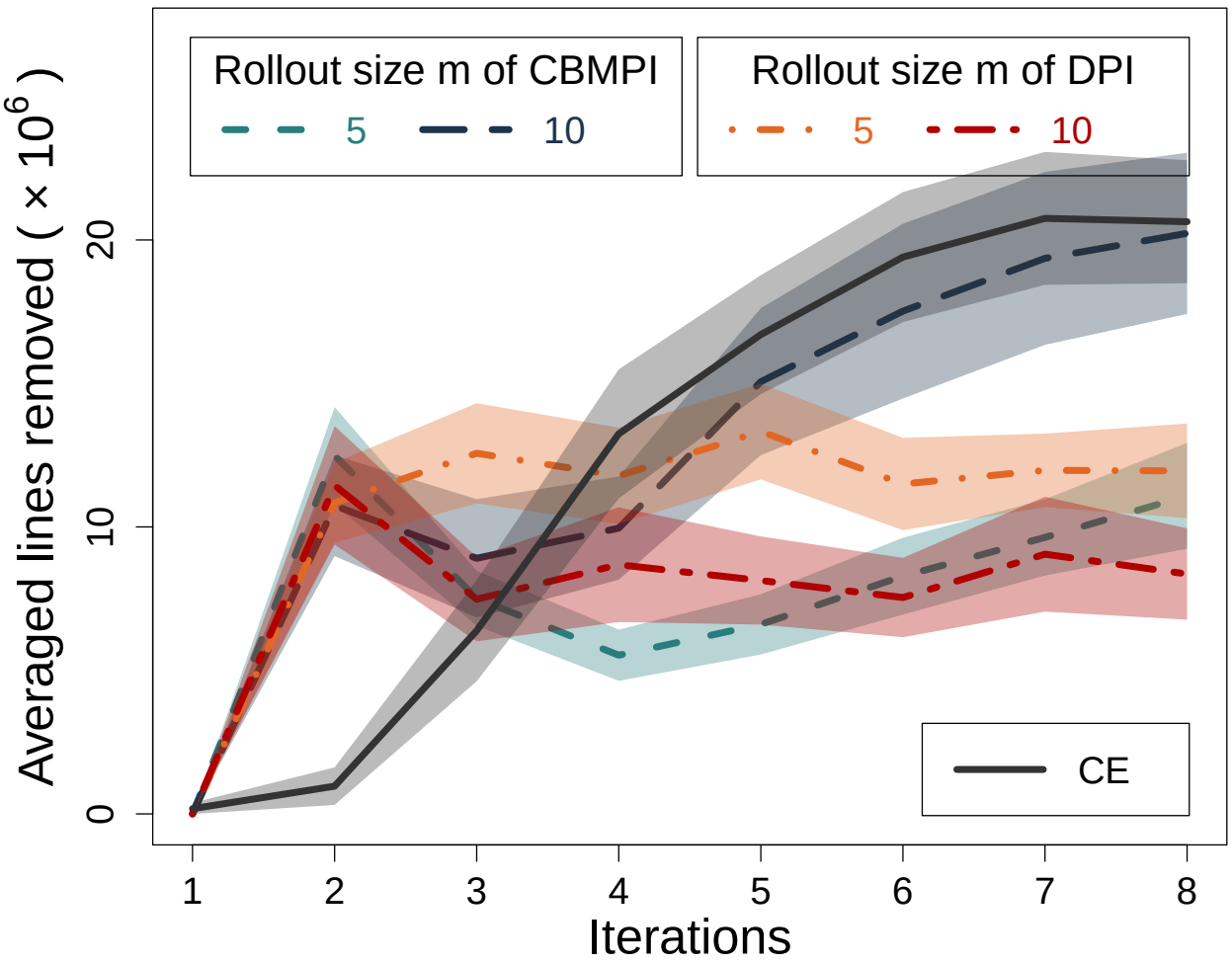
LSTD(λ) - Corollary

- The global error satisfies:

$$\|v - \hat{v}_\lambda\|_\mu \leq \underbrace{\frac{1-\lambda\gamma}{1-\gamma}\|v - \Pi v\|_\mu}_{\text{approximation error}} + \underbrace{\frac{4V_{\max} dL^2}{\sqrt{n-1}(1-\gamma)\nu}\sqrt{\left(1 + \left\lceil \frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)} \right\rceil\right) I(n-1,\delta) + h(n,\delta)}}_{\text{estimation error}}.$$

- $\lambda = 1$ (resp. $\lambda = 0$) minimizes the approximation (resp. estimation) error
- When $n \to \infty$, the best value of λ tends to 1.

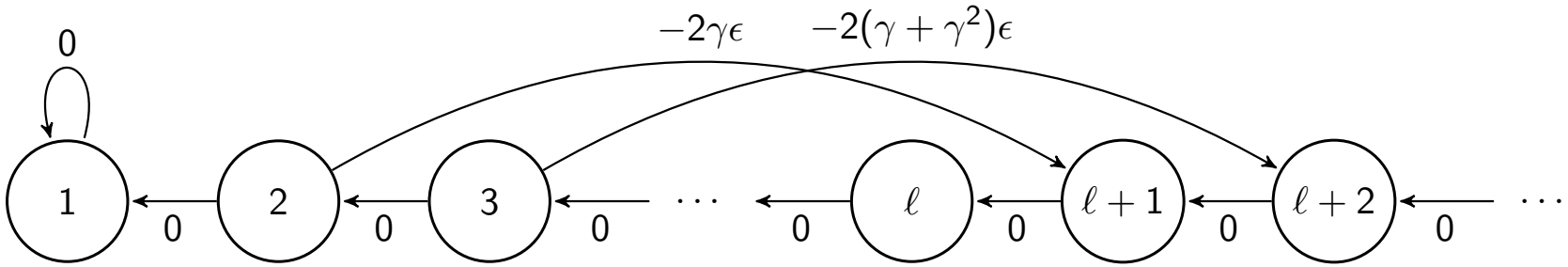
Tetris (10×20)



Courbes d'apprentissage pour Cross Entropy, DPI(=CBMPI avec $v_k = 0$) et CBMPI en utilisant $\simeq 20$ fonctions de base. 100 répétitions des algorithmes.

$B_{DPI/CBMPI} = 32.10^6$ échantillons. $B_{CE} = 1700.10^6$ échantillons.

Optimalité de la borne (Lesner & Scherrer, 2015)



Pour tout m et ℓ, NSMPI produit une séquence de politiques $(\pi_k)_{k\geq 1}$ telles que π_k agit optimalement partout sauf en k. Ainsi, $(\sigma_{k,\ell})^\infty = (\pi_k \pi_{k-1} \dots \pi_{k-\ell+1})^\infty$ reste bloquée dans la boucle

$$k, \quad k+\ell-1, \quad k+\ell-2, \quad k+1, \quad k, \quad \dots$$

et par conséquent

$$v_{(\sigma_{k,\ell})^\infty}(k) = -\frac{2\gamma - \gamma^k}{(1-\gamma^\ell)(1-\gamma)}\epsilon.$$

Illustration empirique

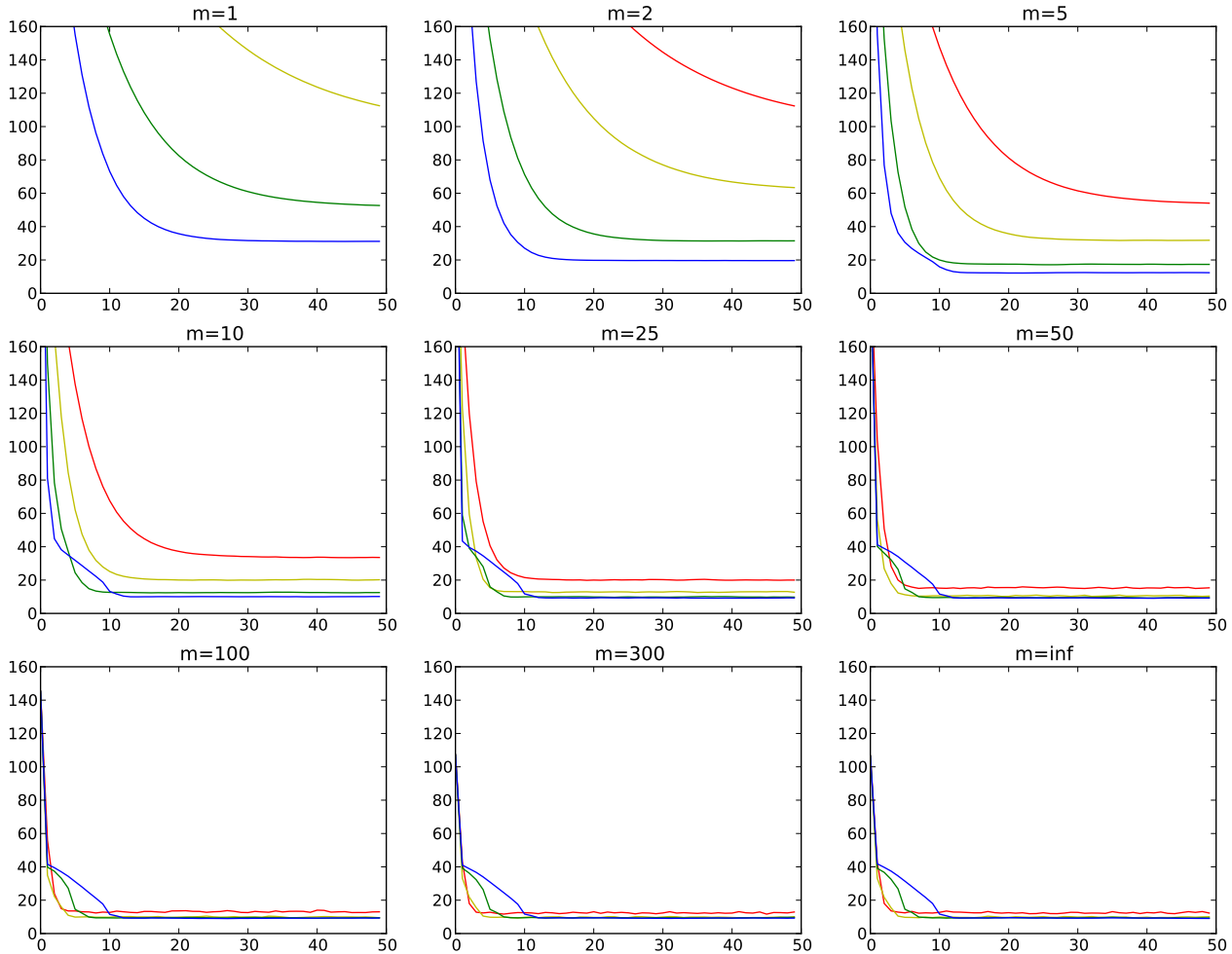


Figure: Erreur moyenne de la politique $(\sigma_{k,\ell})^{\infty}$ en fonction des itérations k. $\ell = 1$, $\ell = 2$, $\ell = 5$, $\ell = 10$.

Approximate/Conservative Policy Iteration

API

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\pi_k})$

CPI/CPI+/CPI(α) (Kakade and Langford, 2002)

$\pi_{k+1} \leftarrow (1-\alpha_{k+1})\pi_k + \alpha_{k+1}\mathcal{G}_{\epsilon_k}(d_{\nu,\pi_k}, v_{\pi_k})$

- $d_{\nu,\pi_k}(x') = (1-\gamma)\mathbb{E}_{x_0\sim\nu}\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{1}_{x_t=x'} \mid a_t = \pi_k(x_t)\right]$

API(α) (Lagoudakis, 2003)

$\pi_{k+1} \leftarrow (1-\alpha)\pi_k + \alpha\mathcal{G}_{\epsilon_k}(\nu, v_{\pi_k})$

Approximate/Conservative Policy Iteration

API

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\pi_k})$

CPI/CPI+/CPI(α) (Kakade and Langford, 2002)

$\pi_{k+1} \leftarrow (1-\alpha_{k+1})\pi_k + \alpha_{k+1}\mathcal{G}_{\epsilon_k}(d_{\nu,\pi_k}, v_{\pi_k})$

- $d_{\nu,\pi_k}(x') = (1-\gamma)\mathbb{E}_{x_0\sim\nu}\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{1}_{x_t=x'} \mid a_t = \pi_k(x_t)\right]$

API(α) (Lagoudakis, 2003)

$\pi_{k+1} \leftarrow (1-\alpha)\pi_k + \alpha\mathcal{G}_{\epsilon_k}(\nu, v_{\pi_k})$

Approximate/Conservative Policy Iteration

API

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\pi_k})$

CPI/CPI+/CPI(α) (Kakade and Langford, 2002)

$\pi_{k+1} \leftarrow (1-\alpha_{k+1})\pi_k + \alpha_{k+1}\mathcal{G}_{\epsilon_k}(d_{\nu,\pi_k}, v_{\pi_k})$

- $d_{\nu,\pi_k}(x') = (1-\gamma)\mathbb{E}_{x_0\sim\nu}\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{1}_{x_t=x'} \mid a_t = \pi_k(x_t)\right]$

API(α) (Lagoudakis, 2003)

$\pi_{k+1} \leftarrow (1-\alpha)\pi_k + \alpha\mathcal{G}_{\epsilon_k}(\nu, v_{\pi_k})$

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma * = \pi_1 * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma_1 * \ = \ \pi_1 * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma_2 * \ = \ \pi_2 \ \pi_1 * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma_3 \ * \ = \ \pi_3 \ \pi_2 \ \pi_1 \ * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma_4 * \ = \ \pi_4 \ \pi_3 \ \pi_2 \ \pi_1 * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma_k * \ = \ \pi_k \ \pi_{k-1} \ \ldots \ \pi_2 \ \pi_1 * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Policy Search by Dynamic Programming à horizon infini

PSDP_∞ (variation de PSDP, Bagnell et al., 2003)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{\sigma_k})$

- $\sigma_k = \pi_k \ \pi_{k-1} \ \ldots \ \pi_1$ est une politique à horizon k $(\sigma_0 = \varnothing)$
- $v_{\sigma_k} = T_{\pi_k} T_{\pi_{k-1}} \ldots T_{\pi_1} 0, \quad (v_{\sigma_0} = 0)$
- **Sortie**: On complète σ_k arbitrairement:

$$\sigma_k * \ = \ \pi_k \ \pi_{k-1} \ \ldots \ \pi_2 \ \pi_1 * \qquad (*=\text{arbitraire})$$

Pour CPI et PSDP_∞, la mémoire augmente linéairement avec le nombre d'itérations!

Non-Stationary Policy Iteration

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{(\sigma_k^\ell)^\infty})$

- $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+1})^\infty$ est une politique de période ℓ
- $v_{(\sigma_k^\ell)^\infty} = T_{\pi_k} T_{\pi_{k-1}} \dots T_{\pi_{k-\ell+1}} v_{(\sigma_k^\ell)^\infty}$
- **Sortie:**

$$
\begin{aligned}
(\sigma_0^\ell)^\infty &= (\pi_0 \ \pi_{-1} \ \dots \ \pi_{-\ell+2} \ )^\infty \\
(\sigma_1^\ell)^\infty &= (\ \pi_0 \ \dots \ \pi_{-\ell+3} \ )^\infty \\
(\sigma_2^\ell)^\infty &= (\ \pi_1 \ \dots \ \pi_{-\ell+4} \ \pi_{-\ell+3})^\infty \\
&\ \ \vdots \qquad\qquad\qquad\qquad \vdots \\
(\sigma_k^\ell)^\infty &= (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+2} \ \pi_{k-\ell+1})^\infty
\end{aligned}
$$

Intermédiaire entre API=NSPI(1) et $\text{PSDP}_\infty \simeq \text{NSPI}(\infty)$.

Non-Stationary Policy Iteration

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{(\sigma_k^\ell)^\infty})$

- $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+1})^\infty$ est une politique de période ℓ
- $v_{(\sigma_k^\ell)^\infty} = T_{\pi_k} T_{\pi_{k-1}} \dots T_{\pi_{k-\ell+1}} v_{(\sigma_k^\ell)^\infty}$
- **Sortie:**
  $(\sigma_0^\ell)^\infty = (\pi_0 \ \pi_{-1} \ \dots \ \pi_{-\ell+2} \ \pi_{-\ell+1})^\infty$
  $(\sigma_1^\ell)^\infty = (\pi_0 \ \dots \ \pi_{-\ell+3} \ \pi_{-\ell+2})^\infty$
  $(\sigma_2^\ell)^\infty = (\pi_2 \ \pi_1 \ \dots \ \pi_{-\ell+4} \ \pi_{-\ell+3})^\infty$
  $\vdots \qquad\qquad \vdots$
  $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+2} \ \pi_{k-\ell+1})^\infty$

Intermédiaire entre API=NSPI(1) et $\text{PSDP}_\infty \simeq \text{NSPI}(\infty)$.

Non-Stationary Policy Iteration

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{(\sigma_k^\ell)^\infty})$

- $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+1})^\infty$ est une politique de période ℓ
- $v_{(\sigma_k^\ell)^\infty} = T_{\pi_k} T_{\pi_{k-1}} \dots T_{\pi_{k-\ell+1}} v_{(\sigma_k^\ell)^\infty}$
- **Sortie:**
  $(\sigma_0^\ell)^\infty = (\pi_0 \ \pi_{-1} \ \dots \ \pi_{-\ell+2} \ \pi_{-\ell+1})^\infty$
  $(\sigma_1^\ell)^\infty = (\pi_1 \ \pi_0 \ \dots \ \pi_{-\ell+3} \ \pi_{-\ell+2})^\infty$
  $(\sigma_2^\ell)^\infty = (\pi_2 \ \pi_1 \ \dots \ \pi_{-\ell+4} \ \pi_{-\ell+3})^\infty$
  $\vdots \qquad\qquad \vdots$
  $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+2} \ \pi_{k-\ell+1})^\infty$

Intermédiaire entre API=NSPI(1) et $\text{PSDP}_\infty \simeq \text{NSPI}(\infty)$.

Non-Stationary Policy Iteration

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{(\sigma_k^\ell)^\infty})$

- $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+1})^\infty$ est une politique de période ℓ
- $v_{(\sigma_k^\ell)^\infty} = T_{\pi_k} T_{\pi_{k-1}} \dots T_{\pi_{k-\ell+1}} v_{(\sigma_k^\ell)^\infty}$
- **Sortie:**

$$(\sigma_0^\ell)^\infty = (\pi_0 \ \pi_{-1} \ \dots \ \pi_{-\ell+2} \ \pi_{-\ell+1})^\infty$$
$$(\sigma_1^\ell)^\infty = (\pi_1 \ \pi_0 \ \dots \ \pi_{-\ell+3} \ \pi_{-\ell+2})^\infty$$
$$(\sigma_2^\ell)^\infty = (\pi_2 \ \pi_1 \ \dots \ \pi_{-\ell+4} \ \pi_{-\ell+3})^\infty$$
$$\vdots \qquad\qquad \vdots$$
$$(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+2} \ \pi_{k-\ell+1})^\infty$$

Intermédiaire entre API=NSPI(1) et $\text{PSDP}_\infty \simeq \text{NSPI}(\infty)$.

Non-Stationary Policy Iteration

NSPI(ℓ)

$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{(\sigma_k^\ell)^\infty})$

- $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+1})^\infty$ est une politique de période ℓ
- $v_{(\sigma_k^\ell)^\infty} = T_{\pi_k} T_{\pi_{k-1}} \dots T_{\pi_{k-\ell+1}} v_{(\sigma_k^\ell)^\infty}$
- **Sortie:**

$$
\begin{aligned}
(\sigma_0^\ell)^\infty &= (\pi_0 \ \pi_{-1} \ \dots \ \pi_{-\ell+2} \ \pi_{-\ell+1})^\infty \\
(\sigma_1^\ell)^\infty &= (\pi_1 \ \pi_0 \ \dots \ \pi_{-\ell+3} \ \pi_{-\ell+2})^\infty \\
(\sigma_2^\ell)^\infty &= (\pi_2 \ \pi_1 \ \dots \ \pi_{-\ell+4} \ \pi_{-\ell+3})^\infty \\
&\ \ \vdots \qquad\qquad\qquad\qquad \vdots \\
(\sigma_k^\ell)^\infty &= (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+2} \ \pi_{k-\ell+1})^\infty
\end{aligned}
$$

Intermédiaire entre API=NSPI(1) et $\text{PSDP}_\infty \simeq \text{NSPI}(\infty)$.

Non-Stationary Policy Iteration

NSPI(ℓ)

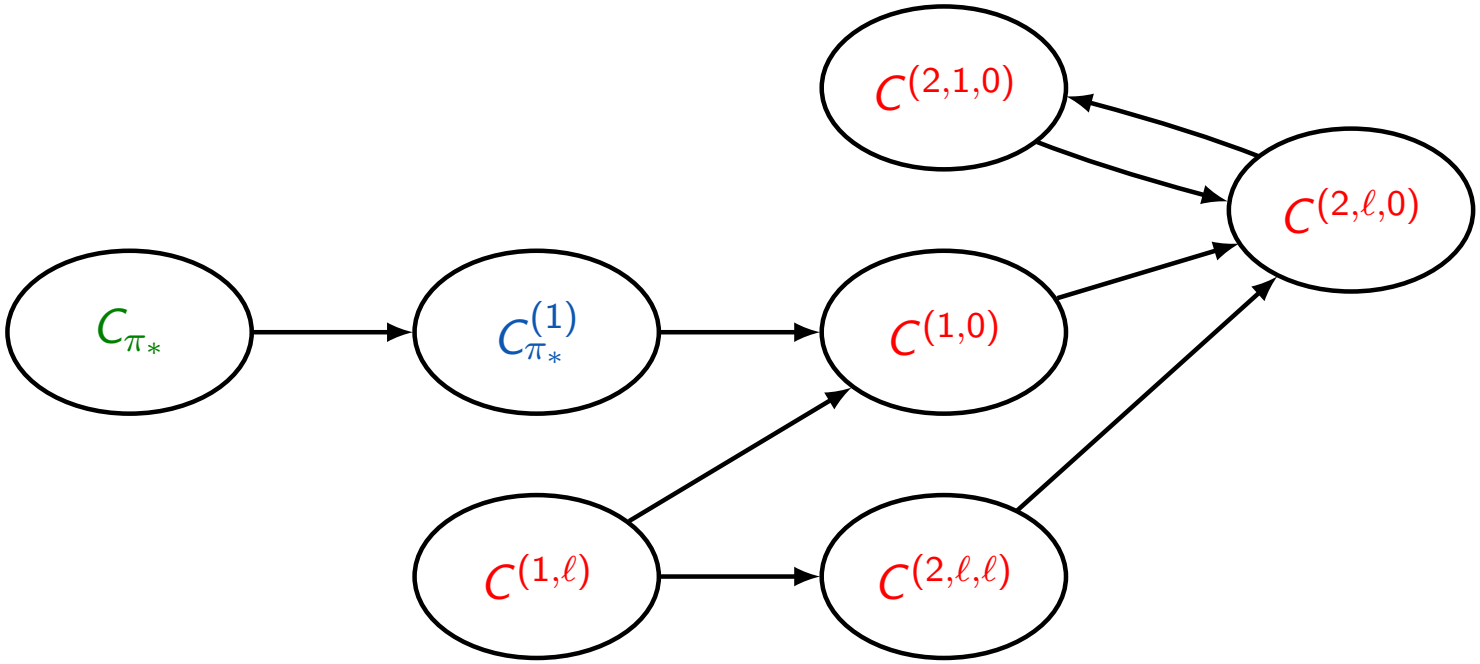
$\pi_{k+1} \leftarrow \mathcal{G}_{\epsilon_k}(\nu, v_{(\sigma_k^\ell)^\infty})$

- $(\sigma_k^\ell)^\infty = (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+1})^\infty$ est une politique de période ℓ
- $v_{(\sigma_k^\ell)^\infty} = T_{\pi_k} T_{\pi_{k-1}} \dots T_{\pi_{k-\ell+1}} v_{(\sigma_k^\ell)^\infty}$
- **Sortie:**

$$
\begin{aligned}
(\sigma_0^\ell)^\infty &= (\pi_0 \ \pi_{-1} \ \dots \ \pi_{-\ell+2} \ \pi_{-\ell+1})^\infty \\
(\sigma_1^\ell)^\infty &= (\pi_1 \ \pi_0 \ \dots \ \pi_{-\ell+3} \ \pi_{-\ell+2})^\infty \\
(\sigma_2^\ell)^\infty &= (\pi_2 \ \pi_1 \ \dots \ \pi_{-\ell+4} \ \pi_{-\ell+3})^\infty \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
(\sigma_k^\ell)^\infty &= (\pi_k \ \pi_{k-1} \ \dots \ \pi_{k-\ell+2} \ \pi_{k-\ell+1})^\infty
\end{aligned}
$$

Intermédiaire entre API=NSPI(1) et $\text{PSDP}_\infty \simeq \text{NSPI}(\infty)$.

Analyse (2/2): Hiérarchie des constantes



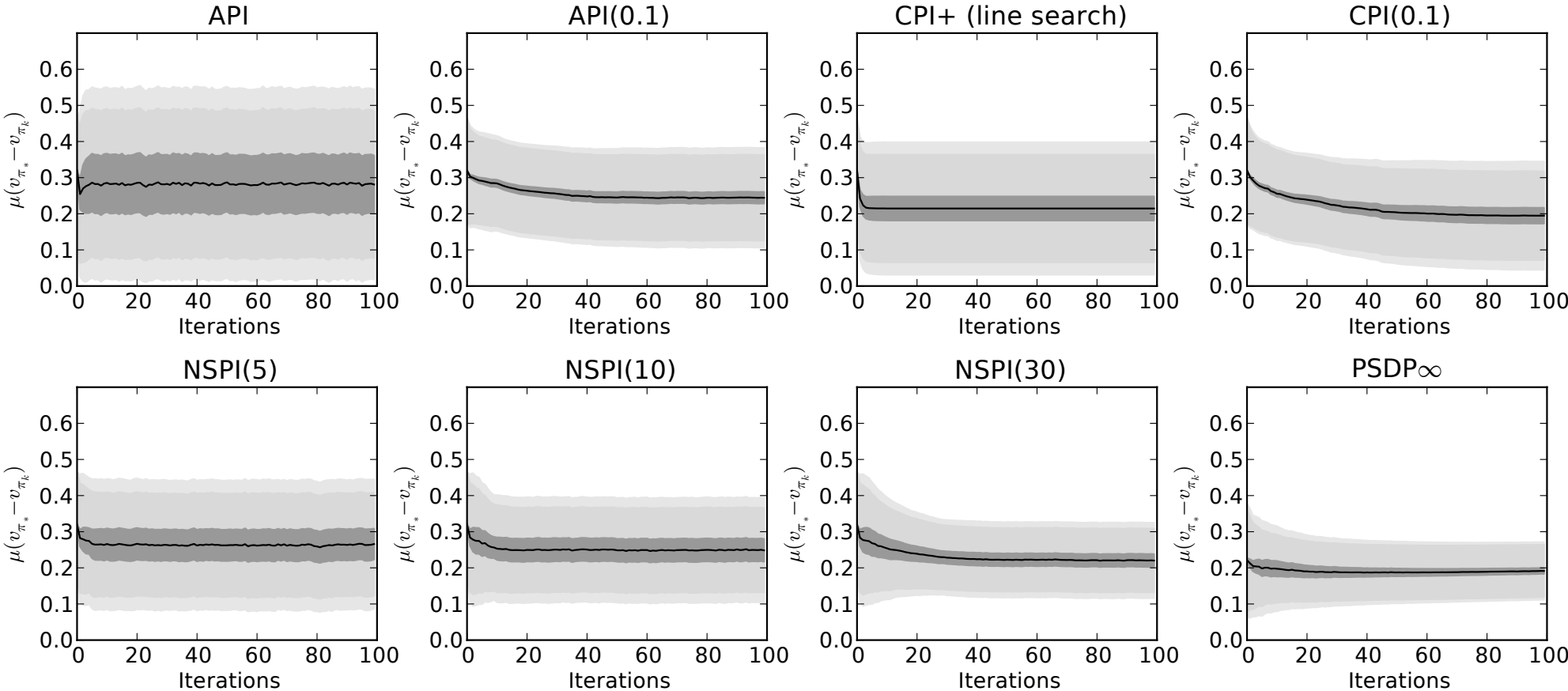
$$A \to B \quad \textit{ssi} \quad \{B < \infty \Rightarrow A < \infty\}$$

$$C^{(1,k)} = (1-\gamma)\sum_{i=0}^{\infty} \gamma^i c(i+k), \qquad C^{(1)}_{\pi_*} = (1-\gamma)\sum_{i=0}^{\infty} \gamma^i c_{\pi_*}(i),$$

$$C^{(2,\ell,k)} = (1-\gamma)(1-\gamma^\ell)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \gamma^{i+j\ell} c(i+j\ell+k), \qquad d_{\pi_*,\mu} \le C_{\pi_*}\nu$$

where $\mu P_{\pi_1} P_{\pi_2} \dots P_{\pi_i} \le c(i)\nu$ and $\mu (P_{\pi_*})^i \le c_{\pi_*}(i)\nu$.

Simulations numériques



$3^3 * 30 \simeq 800$ problèmes Garnet (Archibald *et al.* , 1995).

Pour chaque problème, les algorithmes ont été lancés 30 fois.