Apprentissage par Renforcement Batch fondé sur la Reconstruction de Trajectoires Artificielles

Raphael Fonteneau[1], Susan A. Murphy[2], Louis Wehenkel[1], Damien Ernst[1]

[1] Départment d'Electricité, Electronique et Informatique, Université de Liège, Belgique
{raphael.fonteneau, l.wehenkel, dernst}@ulg.ac.be

[2] Department of Statistics, University of Michigan, USA
samurphy@umich.edu

Résumé :
Cet article se situe dans le cadre de l'apprentissage par renforcement en mode batch, dont le problème central est d'apprendre, à partir d'un ensemble de trajectoires, une politique de décision optimisant un critère donné. On considère plus spécifiquement les problèmes pour lesquels l'espace d'état est continu, problèmes pour lesquels les schémas de résolution classiques se fondent sur l'utilisation d'approximateurs de fonctions. Cet article propose une alternative fondée sur la reconstruction de "trajectoires artificielles" permettant d'aborder sous un angle nouveau les problèmes classiques de l'apprentissage par renforcement batch.
Mots-clés : Apprentissage par renforcement batch.

1 Introduction

La thématique du contrôle optimal intervient dans de nombreuses applications, que ce soit dans le domaine de l'ingénierie [38], en médecine [32, 33, 39] ou en intelligence artificielle [42]. Les techniques issues de l'apprentissage par renforcement (RL) sont de plus en plus sollicitées pour résoudre de tels problèmes. Initialement dédiées à la construction d'agents capables d'interagir de manière optimale dans un environnement [42], les techniques RL se sont diversifiées, et depuis la fin des années 90, une communauté de chercheurs s'est concentrée sur la mise au point de politiques de décision lorsque la seule information disponible sur l'environnement est un ensemble (un "batch") de trajectoires. Ce problème est usuellement nommé apprentissage par renforcement en mode "batch"[10, 15].

La majorité des techniques permettant de résoudre les problèmes de RL batch dans le cas d'espaces de grande taille / continus se fondent sur la combinaison de schémas de type itération sur les valeurs / politiques issus de la théorie de la programmation dynamique [2] avec des approximateurs de fonctions (e.g., réseaux de neurones, arbres de regression, etc) représentant les fonctions de valeur. Ces approximateurs ont deux missions principales : (i) offrir une représentation compacte des fonctions de valeur définies sur de grands espaces, et (ii) généraliser l'information contenue dans un ensemble fini de données. Une autre famille d'algorithmes, moins étudiés jusqu'à présent, adoptent une approche en deux phases : un modèle est d'abord appris à partir d'approximateurs de fonctions, puis ce modèle est résolu à partir de schémas classiques (recherche directe de politique, programmation dynamique) afin d'extraire une politique quasi-optimale.

Bien qu'ayant débouché sur de nombreux succès, l'utilisation d'approximateurs de fonction pour résoudre les problème de RL batch a également des inconvénients. En particulier, leur nature souvent "boite noire" complexifie leur analyse, et freine la mise au point d'algorithmes avec garanties de performance. D'autre part, les politiques inférées dans ce cadre ont parfois des propriétés contre-intuitives : par exemple, dans un contexte déterministe, pour un état initial fixé, et disposant dans les données d'une trajectoire générée à partir d'une politique optimale partant de cet état, il n'y a aucune garantie qu'un algorithme utilisant des approximateurs reproduira le comportement optimal.

Ces constatations nous ont amenés à explorer une nouvelle direction de recherche fondée sur la reconstruction de "trajectoires artificielles" pour le RL batch. De telles trajectoires artificielles sont reconstruites à partir de quadruplets (état, action, récompense, état suivant) extraits du batch de trajectoires dans le but d'atteindre un objectif donné. Dans cet article, on revisite les travaux récents [17, 18, 19, 20, 21] dans l'optique de faire la synthèses des problèmes que la reconstruction de trajectoires artificielles permet d'aborder.

En particulier, quatre algorithmes utilisant les trajectoires artificielles sont présentés : le premier permet d'estimer les performances d'une politique donnée [20]. Le second présente une approche pour calculer des garanties de performance dans le cas déterministe [17]. Le troisième mène au calcul de politiques prudentes [18] tandis que le quatrième décrit une stratégie d'échantillonnage pour générer des données supplémentaires [19]. Enfin, on met en lumière une connexion possible entre la reconstruction de trajectoires artificielles et le fonctionnement des algorithmes RL batch classiques. On précise enfin qu'une version étendue de cet article incorporant notamment des validations expérimentales est disponible (voir [22]).

La suite de cet article est organisée de la manière suivante : la section 2 donne un bref exposé de la littérature RL batch. La section 3 décrit le formalisme utilisé dans cet article, ainsi que plusieurs sous-problèmes classiques. La section 4 décrit la notion de trajectoires artificielles et différentes approches permettant de les manipuler en fonction des problèmes abordés. Enfin, la section 5 établit quelques connections entre le paradigme des trajectoires artificielles et d'autres techniques classiques du RL batch, et la section 6 conclut cet article.

2 Travaux connexes

Les prémices de l'apprentissage par renforcement batch remontent très probablement aux travaux de Bradtke et Barto [5] ainsi que Boyan [4] relatifs à l'utilisation de techniques des moindres carrés dans le contexte de l'apprentissage des différences temporelles (LSTD) pour estimer le retour de politiques de décision. Ces travaux ont par la suite été étendus pour aborder des problèmes de contrôle optimal par Lagoudakis et Parr [25] introduisant l'algorithme LSPI (Least-Square Policy Iteration), lui-même inspiré du célèbre algorithme d'itération sur les politiques issu de la programmation dynamique. De nombreux papiers se sont penchés sur les questions d'ordre théorique liés à ces approches [27, 28, 34, 41].

Un autre algorithme classique de la programmation dynamique, "itération sur les valeurs", a aussi beaucoup inspiré la communauté RL batch. En 2002, Ormoneit et Sen ont développé un algorithme de cette famille en utilisant des noyaux [35]. Le principe de l'itération sur les valeurs a également été combiné avec des arbres de regression dans le cadre de l'algorithme Fitted Q Iteration (FQI) [10].

Les travaux présentés dans [26, 38] et [43] s'intéressent aux performances de FQI utilisés avec des réseaux de neurones (profonds) et des CMACs (Cerebella Model Articulator Controllers). L'algorithme FQI régularisé utilise une regression des moindres carrés pénalisée pour limiter la complexité de l'approximateur de fonction [14]. Des extensions de FQI lorsque l'espace de décision est continu ont également été proposées [1]. Des travaux plus théoriques sur le paradigme FQI ont également été publiés [8, 31].

Le RL batch a permis d'obtenir des résultats expérimentaux prometteurs dans les domaines de la robotique [3, 36, 44], des réseaux électriques [11], du traitement des images [12], de l'optimisation de réservoirs d'eau [6, 7], de la médecine [13, 24, 32] et de la conduite automobile autonome [37].

3 RL batch : formalisation et problèmes typiques

On considère un système dynamique à temps discret régi par l'équation :

$$x_{t+1} = f\left(x_t, u_t, w_t\right), \forall t \in \{0, \ldots, T-1\}$$

où x_t appartient à un espace d'état $\mathcal{X} \subset \mathbb{R}^d$, où $\mathbb{R}^d$ est un espace Euclidien de dimension d et $T \in \mathbb{N} \setminus \{0\}$ l'horizon d'optimisation supposé fini. A chaque pas de temps $t \in \{0, \ldots, T-1\}$, le système est contrôlé via une décision (ou action) $u_t \in \mathcal{U}$, et soumis à une perturbation $w_t \in \mathcal{W}$ tirée selon une distribution de probabilité $p_{\mathcal{W}}(\cdot)$ [1]. A chaque transition du système du temps t au temps $t+1$ est associé un signal de récompense : $r_t = \rho\left(x_t, u_t, w_t\right), \forall t \in \{0, \ldots, T-1\}$. Soit $h : \{0, \ldots, T-1\} \times \mathcal{X} \to \mathcal{U}$ une politique de décision. Partant d'un état initial donné x_0 et suivant la politique h, un agent collectera une somme de récompenses $R^h(x_0, w_0, \ldots, w_{T-1})$:

$$R^h(x_0, w_0, \ldots, w_{T-1}) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t), w_t) \text{ avec } x_{t+1} = f(x_t, h(t, x_t), w_t), w_t \sim p_{\mathcal{W}}(\cdot).$$

1. On fait ici l'hypothèse que w_t ne dépend pas de $w_{t-1}, w_{t-2}, \ldots, w_0$ étant donné x_t et u_t ; par soucis de simplification, on fait également l'hypothèse que le processus ne dépend ni du temps, ni de la paire état-action x_t, u_t.

En RL, le critère de performance classiquement utilisé pour évaluer une politique h est son retour espéré sur T pas de temps :

$$J^h(x_0) = \mathbb{E}\big[R^h(x_0, w_0, \dots, w_{T-1})\big] \,,$$

mais lorsqu'on souhaite incorporer la notion de risque, il est préférable de recourir à un autre critère, comme par exemple le critère suivant : soit $b \in \mathbb{R}$ et $c \in [0, 1[$.

$$J_{RS}^{h,(b,c)}(x_0) = \begin{cases} -\infty & \text{si } P\left(R^h(x_0, w_0, \dots, w_{T-1}) < b\right) > c \,, \\ J^h(x_0) & \text{sinon.} \end{cases}$$

Le principal problème en RL batch est de déterminer une bonne approximation d'une politique h^* optimisant l'un des critères précédents, étant donné le fait que *les fonctions f, ρ et $p_{\mathcal{W}}(\cdot)$ sont inconnues*, et donc non disponibles à la simulation. Elles sont en revanche "remplacées" par le batch de $n \in \mathbb{N} \setminus \{0\}$ *transitions du système*, définies selon le procédé suivant.

Soit $\mathcal{P}_n = \left\{\left(x^l, u^l\right)\right\}_{l=1}^n \in (\mathcal{X} \times \mathcal{U})^n$ un ensemble fixé de paires état-action. Considérons l'ensemble des batch de transitions de taille n qui pourraient être générés en générant, pour chaque paire (x^l, u^l) de $\mathcal{P}_n$, une perturbation w^l selon la distribution $p_{\mathcal{W}}(.)$ afin d'obtenir des valeurs $\rho(x^l, u^l, w^l)$ et $f(x^l, u^l, w^l)$. On nomme $\tilde{\mathcal{F}}_n\left(\mathcal{P}_n, w^1, \dots, w^n\right)$ un tel ensemble "aléatoire" de transitions défini par le tirage de n perturbations i.i.d. $w^l, \quad l = 1 \dots n$. On fait ainsi l'hypothèse que l'on connaît une réalisation de l'ensemble aléatoire $\tilde{\mathcal{F}}_n\left(\mathcal{P}_n, w^1, \dots, w^n\right)$ que l'on nomme $\mathcal{F}_n$:

$$\mathcal{F}_n = \left\{\left(x^l, u^l, r^l, y^l\right)\right\}_{l=1}^n$$

où, pour chaque $l \in \{1, \dots, n\}, r^l = \rho\left(x^l, u^l, w^l\right), y^l = f\left(x^l, u^l, w^l\right)$, pour des réalisations du processus de perturbations $w^l \sim p_{\mathcal{W}}(\cdot)$.

Remarquons à ce stade que la résolution du problème central, celui consistant à trouver une politique de décision proche de l'optimalité, est tout à fait corrélé au problème consistant à évaluer précisément les performances d'une politique donnée. Lorsqu'un tel problème est résolu, la recherche d'une politique optimale peut être théoriquement réduite à un problème d'optimisation sur un ensemble de politiques candidates. Cet article abordera donc dans un premier temps le problème de la caractérisation des performances d'une politique donnée. Il n'est pas rare de rencontrer des problèmes pour lesquels il est primordial que les politiques aient des garanties de performance. Certains problèmes définissent même un niveau de performance minimal à atteindre, plutôt qu'un objectif d'optimalité. Ce problème est abordé dans cet article. Enfin, pour certains types d'applications, il peut y avoir la possibilité de générer des données supplémentaires afin d'enrichir le batch de transitions précédemment acquis. Ce problème est également abordé dans cet article.

4 Reconstruction de Trajectoires Artificielles

On formalise la notion de "trajectoires artificielles" dans la section 4.1. Dans la section 4.2, on montre comment les trajectoires artificielles peuvent servir à estimer les performances d'une politique de décision. On se focalise ensuite sur le cas déterministe en section 4.3 pour lequel on montre comment les trajectoires artificielles peuvent servir à calculer des bornes sur les performances des politiques. Par la suite, ces bornes sont exploitées dans deux cas de figures : calculer des politiques de décision avec garanties de performance (section 4.4) et mettre au point des stratégies d'échantillonnage permettant de générer des données supplémentaires (section 4.5).

4.1 Trajectoires Artificielles

Les trajectoires artificielles sont constituées de transitions prises dans l'ensemble $\mathcal{F}_n$. Formellement, une trajectoire artificielle se définit comme suit :

Définition 1 (Trajectoire artificielle)
Une trajectoire artificielle est une suite finie (ordonnée) de T transitions :

$$\left[\left(x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0}\right), \dots, \left(x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}}\right)\right] \in \mathcal{F}_n^T$$

où $l_t \in \{1, \dots, n\}, \forall t \in \{0, \dots, T-1\}$.

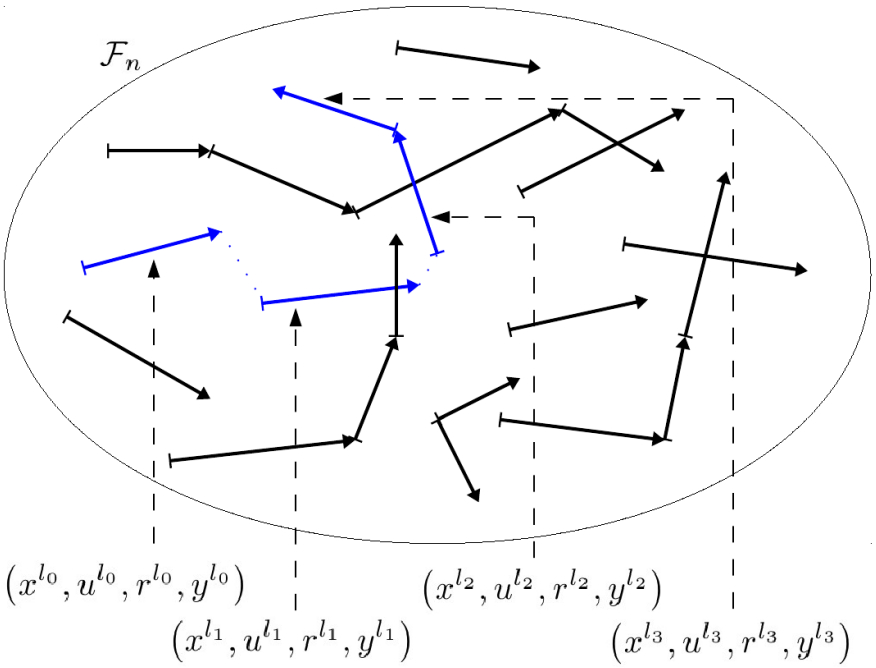


FIGURE 1 – Un exemple de trajectoire artificielle reconstruite à partir de 4 transitions prises dans l'ensemble $\mathcal{F}_n$.

On donne à la figure 1 une illustration d'une trajectoire artificielle.

Remarquons qu'un maximum de n^T trajectoires artificielles différentes sont constructibles à partir de l'ensemble de transitions $\mathcal{F}_n$. Dans la suite de cet article, différentes techniques de construction d'ensembles de trajectoires artificielles adaptés à la résolution de problèmes particuliers sont présentées.

4.2 Evaluer le retour espéré d'une politique

Un problème fondamental en RL batch est d'estimer le retour espéré $J^h(x_0)$ pour une politique de décision donnée h. Lorsqu'un tel oracle est disponible, la recherche d'une politique optimale se réduit à un problème d'optimisation dans un espace de politiques candidates. Si on connaît un modèle de la dynamique du système, de la fonction de récompense et de la distribution de probabilité à partir de laquelle les perturbations sont générées, alors les techniques d'estimation Monte Carlo peuvent être mises en oeuvre pour estimer les performances de politiques. Ceci n'est pas possible dans le contexte batch. Dans cette section, on détaille précisément une approche permettant d'estimer les performances d'une politique de décision fondée sur la reconstruction de trajectoires artificielles en imitant - en quelque sorte - la comportement d'un estimateur Monte Carlo. On fait l'hypothèse dans cette section (et également en section 4.3) que l'espace de décisions $\mathcal{U}$ est continu et normé.

4.2.1 Estimation Monte Carlo

L'estimateur de Monte Carlo (MC) fonctionne dans un contexte "model-based" (c'est à dire pour lequel f, ρ et $p_{\mathcal{W}}(.)$ sont connus). Il permet d'estimer $J^h(x_0)$ en moyennant les retours de plusieurs ($p \in \mathbb{N} \setminus \{0\}$) trajectoires générées en simulant le système à partir de x_0 et en utilisant la politique h. Formellement, l'estimateur MC s'écrit :

$$\mathbb{M}_p^h(x_0) = \frac{1}{p} \sum_{i=1}^{p} \sum_{t=0}^{T-1} \rho\left(x_t^i, h(t, x_t^i), w_t^i\right)$$

avec $\forall t \in \{0,\dots,T-1\}, \forall i \in \{1,\dots,p\}$, $w_t^i \sim p_{\mathcal{W}}(.), x_0^i = x_0, x_{t+1}^i = f\left(x_t^i, h(t,x_t^i), w_t^i\right)$. Le biais et la variance de l'estimateur MC valent respectivement :

$$\underset{w_t^i \sim p_{\mathcal{W}}(.), i=1\dots p, t=0\dots T-1}{\mathbb{E}} \left[\mathbb{M}_p^h(x_0) - J^h(x_0)\right] = 0 \,,$$

$$\underset{w_t^i \sim p_{\mathcal{W}}(.), i=1\dots p, t=0\dots T-1}{\mathbb{E}} \left[\left(\mathbb{M}_p^h(x_0) - J^h(x_0)\right)^2\right] = \frac{\sigma^2_{R^h}(x_0)}{p} \,.$$

où $\sigma^2_{R^h}(x_0)$ désigne la variance (supposée finie) de $R^h(x_0, w_0, \dots, w_{T-1})$:

$$\sigma^2_{R^h}(x_0) = \underset{w_0,\dots,w_{T-1} \sim p_{\mathcal{W}}(.)}{Var}\left[R^h(x_0, w_0, \dots, w_{T-1})\right] < +\infty.$$

4.2.2 Estimation Monte Carlo sans modèle

A partir d'un ensemble de transitions $\mathcal{F}_n$, l'estimateur MC "model-free" (MFMC) fonctionne en construisant $p \in \mathbb{N}\backslash\{0\}$ trajectoires artificielles. Ces trajectoires artificielles servent d'"imitations" des p trajectoires "réelles" qui pourraient être obtenues en simulant la politique de décision h sur le système. L'estimateur MFMC moyenne les retours cumulés des trajectoires artificielles afin d'obtenir une estimation de $J^h(x_0)$. L'idée sous-jacente derrière cette approche consiste à construire des trajectoires artificielles "proches" des trajectoires qu'on pourrait obtenir avec un estimateur MC classique en simulant h.

Pour reconstruire un ensemble de p trajectoires artificielles de longueur T partant de x_0 proche des trajectoires qui seraient obtenues avec un estimateur MC, l'algorithme MFMC utilise chaque transition au maximum une seule fois. On fait donc l'hypothèse que $pT \leq n$. Les p trajectoires sont construites successivement. Chaque trajectoires est construite itérativement en sélectionnant, parmi les transitions non utilisées, une transition dont les deux premiers éléments minimisent la distance - en utilisant une distance Δ in $\mathcal{X} \times \mathcal{U}$ définie plus loin - avec la paire formée du dernier élément de la transition précédemment sélectionnée associée à l'action induite par la politique h en cet état. Etant donné que toutes les perturbations $w^l \quad l = 1 \dots n$ sont i.i.d. selon $p_{\mathcal{W}}(\cdot)$ et que les transitions ne sont jamais réutilisées, les perturbations associées aux trajectoires artificielles sont également i.i.d., ce qui donne à l'estimateur MFMC de bonnes propriétés statistiques (voir section 4.2.3). Par ailleurs, ce mécanisme permet de s'assurer que les p trajectoires artificielles sont distinctes.

Algorithm 1 L'algorithme MFMC reconstruisant p trajectoires artificielles de longueur T à partir de n transitions.

Input : $\mathcal{F}_n, h(.,.), x_0, \Delta(.,.), T, p$
Soit $\mathcal{G}$ l'ensemble des transitions non encore utilisées ; Initiallement,
$\mathcal{G} \leftarrow \mathcal{F}_n$;
for $i = 1$ to p (construire une trajectoire artificielle) **do**
  $t \leftarrow 0$;
  $x_t^i \leftarrow x_0$;
  while $t < T$ **do**
    $u_t^i \leftarrow h\left(t, x_t^i\right)$;
    $\mathcal{H} \leftarrow \underset{(x,u,r,y)\in\mathcal{G}}{\arg\min} \quad \Delta\left((x,u),\left(x_t^i, u_t^i\right)\right)$;
    $l_t^i \leftarrow$ plus petit indice dans $\mathcal{F}_n$ des transitions appartenant à $\mathcal{H}$;
    $t \leftarrow t+1$;
    $x_t^i \leftarrow y^{l_t^i}$;
    $\mathcal{G} \leftarrow \mathcal{G} \setminus \left\{\left(x^{l_t^i}, u^{l_t^i}, r^{l_t^i}, y^{l_t^i}\right)\right\}$; \\ *ne pas réutiliser les transitions*
  end while
end for
Return l'ensemble des indices $\left\{l_t^i\right\}_{i=1,t=0}^{i=p,t=T-1}$.

Un pseudo-code de l'algorithme MFMC est donné dans le tableau 1. L'algorithme renvoie un ensemble d'indices de transitions $\left\{l_t^i\right\}_{i=1,t=0}^{i=p,t=T-1}$ de $\mathcal{F}_n$ en utilisant h, x_0, la distance Δ et le paramètre p. Partant

de cet ensemble d'indices, on définit formellement l'estimateur MFMC du retour d'une politique h partant d'un état initial x_0 :

$$\mathfrak{M}_p^h\left(\mathcal{F}_n, x_0\right) \quad = \quad \frac{1}{p}\sum_{i=1}^{p}\sum_{t=0}^{T-1} r^{l_t^i}.$$

Une illustration de l'estimateur MFMC est donnée en figure 2. Remarquons ici que le calcul de l'estimation MFMC est linéaire en fonction du nombre de transitions n, du nombre de trajectoires artificielles p et de l'horizon d'optimisation T.

4.2.3 Résultats théoriques associés à l'estimateur MFMC

Cette section vise à caractériser les propriétés théoriques principales de l'estimateur MFMC. Dans cette optique, on s'intéresse à la distribution de l'estimateur $\mathfrak{M}_p^h\left(\tilde{\mathcal{F}}_n\left(\mathcal{P}_n, w^1, \ldots, w^n\right), x_0\right)$, vu sous l'angle d'une fonction de l'ensemble aléatoire $\tilde{\mathcal{F}}_n\left(\mathcal{P}_n, w^1, \ldots, w^n\right)$; pour caractériser cette distribution, on borne le biais et la variance de l'estimateur MFMC en fonction d'une mesure de la dispersion de l'ensemble $\mathcal{P}_n$, nommée "$k-$dispersion" ; il s'agit du plus petit rayon tel que toutes les Δ-boules de l'espace $\mathcal{X}\times\mathcal{U}$ possédant un tel rayon contiennent au moins k elements de $\mathcal{P}_n$. L'usage de cette notion implique que l'espace $\mathcal{X}\times\mathcal{U}$ soit borné.

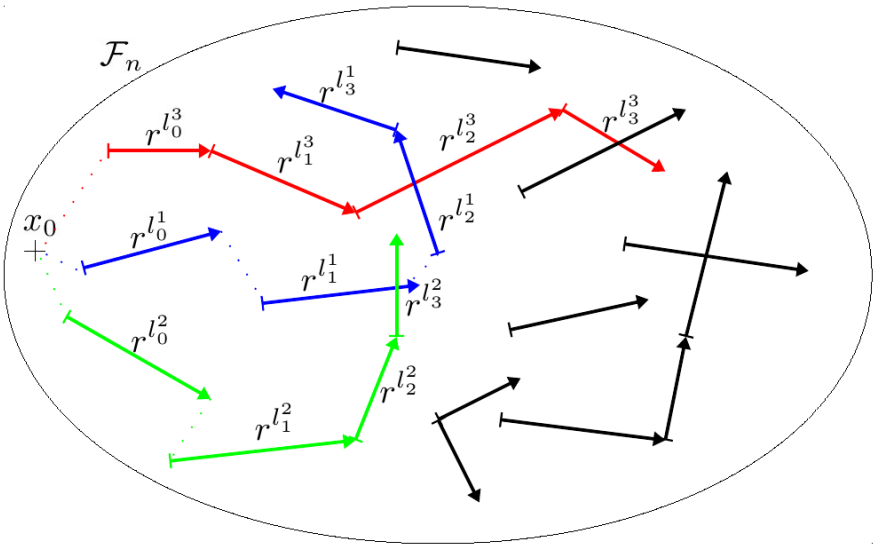


FIGURE 2 – Reconstruction de 3 trajectoires artificielles pour estimer le retour d'une politique.

La caractérisation du biais et de la variance se font sous les hypothèses données ci-dessous. On donne juste après les théorèmes décrivant ces caractérisations. Les preuves des résultats sont données dans [20].
Hypothèse : continuité lipschitzienne des fonctions f, ρ et h. On suppose que la dynamique f, la fonction de récompense ρ et la politique h sont lipschitziennes, c'est à dire qu'on fait l'hypothèse qu'il existe des constantes L_f, L_ρ et $L_h \in \mathbb{R}^+$ telles que :
$\forall\ (x, x', u, u', w) \in \mathcal{X}^2 \times \mathcal{U}^2 \times \mathcal{W}$,

$$\begin{aligned}
\|f(x,u,w) - f(x',u',w)\|_{\mathcal{X}} &\leq L_f(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}),\\
|\rho(x,u,w) - \rho(x',u',w)| &\leq L_\rho(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}),\\
\|h(t,x) - h(t,x')\|_{\mathcal{U}} &\leq L_h\|x-x'\|_{\mathcal{X}}\ ,\forall t \in \{0,\ldots,T-1\}\ ,
\end{aligned}$$

où $\|.\|_{\mathcal{X}}$ et $\|.\|_{\mathcal{U}}$ sont les normes (ici euclidiennes) dans les espaces $\mathcal{X}$ et $\mathcal{U}$, respectivement.
Hypothèse : $\mathcal{X} \times \mathcal{U}$ est borné.

$$\forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2, \quad \Delta((x, u), (x', u')) = \|x - x'\|_{\mathcal{X}} + \|u - u'\|_{\mathcal{U}}.$$

Etant donné $k \in \mathbb{N} \setminus \{0\}$ avec $k \leq n$, on introduit la notion de $k-$*dispersion*, $\alpha_k(\mathcal{P}_n)$ de l'ensemble $\mathcal{P}_n$:

$$\alpha_k(\mathcal{P}_n) = \sup_{(x,u) \in \mathcal{X} \times \mathcal{U}} \Delta_k^{\mathcal{P}_n}(x, u) \, ,$$

où $\Delta_k^{\mathcal{P}_n}(x, u)$ désigne le distance entre (x, u) et le $k-$ème plus proche voisin (selon la distance Δ) dans l'ensemble $\mathcal{P}_n$. La $k-$dispersion est le plus petit rayon telle que toute les Δ-boules dans $\mathcal{X} \times \mathcal{U}$ possédant un tel rayon contiennent au moins k éléments de $\mathcal{P}_n$; cette notion peut être interprétée comme une mesure "pire cas" de la façon dont l'ensemble $\mathcal{P}_n$ couvre l'espace $\mathcal{X} \times \mathcal{U}$ en utilisant les k plus proches voisins. On note $E^h_{p,\mathcal{P}_n}(x_0)$ la valeur espérée de l'estimateur MFMC :

$$E^h_{p,\mathcal{P}_n}(x_0) = \mathop{\mathbb{E}}_{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)} \left[\mathfrak{M}^h_p \left(\tilde{\mathcal{F}}_n \left(\mathcal{P}_n, w^1, \ldots, w^n \right), x_0 \right) \right] \, .$$

On a le résultat suivant :

Théorème 1 (Borne sur le biais de $\mathfrak{M}^h_p \left(\tilde{\mathcal{F}}_n \left(\mathcal{P}_n, w^1, \ldots, w^n \right), x_0 \right)$)

$$\left| J^h(x_0) - E^h_{p,\mathcal{P}_n}(x_0) \right| \leq C \alpha_{pT}(\mathcal{P}_n) \text{ avec } C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left(L_f (1 + L_h) \right)^i .$$

La preuve de ce résultat est donnée dans [20]. La borne montre que le biais diminue avec la dispersion. Notons que la dispersion ne dépend que de l'ensemble $\mathcal{P}_n$ et de la valeur de p (qui augmente avec le nombre de trajectoires artificielles utilisées). On définit ensuite $V^h_{p,\mathcal{P}_n}(x_0)$ la variance de l'estimateur MFMC :

$$V^h_{p,\mathcal{P}_n}(x_0) = \mathop{\mathbb{E}}_{w^1,\ldots,w^n \sim p_{\mathcal{W}}(.)} \left[\left(\mathfrak{M}^h_p \left(\tilde{\mathcal{F}}_n \left(\mathcal{P}_n, w^1, \ldots, w^n \right), x_0 \right) - E^h_{p,\mathcal{P}_n}(x_0) \right)^2 \right] .$$

On a le résultat :

Théorème 2 (Borne sur la variance de $\mathfrak{M}^h_p \left(\tilde{\mathcal{F}}_n \left(\mathcal{P}_n, w^1, \ldots, w^n \right), x_0 \right)$)

$$V^h_{p,\mathcal{P}_n}(x_0) \quad \leq \quad \left(\frac{\sigma_{R^h}(x_0)}{\sqrt{p}} + 2C \alpha_{pT}(\mathcal{P}_n) \right)^2 \text{ avec } C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left(L_f (1 + L_h) \right)^i .$$

La preuve de ce résultat est donnée dans [20]. La borne indique que la variance de l'estimateur MFMC se rapproche de celle d'un estimateur MC classique quand la dispersion tend vers 0.

4.2.4 Estimation MFMC sensible au risque

Dans l'optique de prendre en compte le caractère risqué des politiques, et pas uniquement leurs bonnes performance en moyenne, il est préférable de recourir à un critère de performance sensible au risque. Ce type de critères a reçu une attention croissante au cours des dernières années dans la communauté RL [9, 29, 30, 40].

Considérant les p trajectoires artificielles construites par l'estimateur MFMC, le retour sensible au risque $J^{h,(b,c)}_{RS}(x_0)$ peut-être approché par l'estimation $\tilde{J}^{h,(b,c)}_{RS}(x_0)$ définie ci-dessous : Soit $b \in \mathbb{R}$ et $c \in [0, 1[$.

$$\tilde{J}^{h,(b,c)}_{RS}(x_0) = \begin{cases} -\infty & \text{si } \frac{1}{p} \sum_{i=1}^{p} \mathbb{I}_{\{\mathbf{r}^i < b\}} > c \, , \\ \mathfrak{M}^h(\mathcal{F}_n, x_0) & \text{sinon} \end{cases}$$

où $\mathbf{r}^i$ désigne le retour de la $i-$ème trajectoire artificielle : $\mathbf{r}^i = \sum_{t=0}^{T-1} r^{l^i_t}$.

On encourage le lecteur curieux à se reporter à l'article [22] dans lequel des résultats expérimentaux engendrés via l'approche MFMC sont proposés.

4.3 Cas déterministe : calcul de bornes à partir de trajectoires artificielles

A partir de cette sous-section et jusqu'à la fin de la section 4, on fait l'hypothèse que l'environnement est déterministe. Formellement, on fait l'hypothèse que l'espace de perturbation est réduit à un élément unique $\mathcal{W} = \{0\}$ tel que $p_{\mathcal{W}}(0) = 1$. On utilise la convention suivante :

$$\begin{aligned} \forall (x,u) \in \mathcal{X} \times \mathcal{U}, \qquad f(x,u) &= f(x,u,0)\,, \\ \rho(x,u) &= \rho(x,u,0)\,. \end{aligned}$$

On fait toujours l'hypothèse que les fonctions f, ρ et h sont lipschitzienne. Observons ici que, dans le cas déterministe, une seule trajectoire suffit pour calculer $J^h(x_0)$ par simulation MC. On a alors le résultat suivant :

Proposition 1 (Borne inférieure calculée à partir de l'estimateur MFMC)
Soit $\left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1}$ *une trajectoire artificielle reconstruite par l'algorithme MFMC en utilisant la distance* Δ*. On a alors :*

$$\left|\mathfrak{M}_1^h(\mathcal{F}_n, x_0) - J^h(x_0)\right| \leq \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right)$$

avec $L_{Q_{T-t}} = L_\rho \sum_{i=0}^{T-t-1} \left(L_f\left(1 + L_h\right)\right)^i$ *et* $y^{l-1} = x_0$.

La preuve de ce résultat est donnée dans [17]. Le résultat précédent étant valable pour n'importe quelle trajectoire artificielle, on a :

Corollaire 1 (Borne inférieure gobale)
Soit $\left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1}$ *une trajectoire artificielle. Alors,*

$$J^h(x_0) \geq \sum_{t=0}^{T-1} r^{l_t} - \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right).$$

Cela suggère d'identifier une trajectoire artificielle maximisant la borne précédente :

$$L^h(\mathcal{F}_n, x_0) = \max_{[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})]_{t=0}^{T-1} \in \mathcal{F}_n^T} \sum_{t=0}^{T-1} r^{l_t} - \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right)\,.$$

De la même manière, une borne supérieure minimale peut être calculée :

$$U^h(\mathcal{F}_n, x_0) = \min_{[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})]_{t=0}^{T-1} \in \mathcal{F}_n^T} \sum_{t=0}^{T-1} r^{l_t} + \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right)\,.$$

On peut alors montrer que ces bornes sont fines en ce sens qu'elle convergent vers $J^h(x_0)$ lorsque la dispersion de l'ensemble de transitions $\mathcal{F}_n$ tend vers zéro.

Proposition 2 (Précision des bornes)

$$\begin{aligned} \exists C_b > 0 : \quad J^h(x_0) - L^h(\mathcal{F}_n, x_0) &\leq C_b \alpha_1(\mathcal{P}_n) \\ U^h(\mathcal{F}_n, x_0) - J^h(x_0) &\leq C_b \alpha_1(\mathcal{P}_n) \end{aligned}$$

où $\alpha_1(\mathcal{P}_n)$ *désigne la* $1-$*dispersion de l'échantillon* $\mathcal{F}_n$.

La preuve de ce résultat est donnée dans [17]. Remarquons que le calcul des bornes optimales peut se reformuler comme un problème de plus court chemin dans un graphe, dont la résolution a une complexité linéaire en fonction de l'horizon T et quadratique en fonction de n.

4.3.1 Extension au cas d'un espace de décision fini

Les résultats précédents peuvent être étendus au cas où l'espace de décision $\mathcal{U}$ est fini en considérant des politiques de décision en boucle ouverte, chaque politique étant entièrement spécifiée par une séquence de décisions. Soit Π l'ensemble de ces politiques :

$$\Pi = \{\pi : \{0, \dots, T-1\} \to \mathcal{U}\}$$

Etant donnée une politique en boucle ouverte π, le retour (déterministe) de la politique π s'écrit :

$$J^\pi(x_0) = \sum_{t=0}^{T-1} \rho(x_t, \pi(t))$$

avec $x_{t+1} = f(x_t, \pi(t)), \quad \forall t \in \{0, \dots, T-1\}$. Dans le cas d'un espace de décision fini, la continuité lipschitzienne de f et ρ s'écrit : $\forall\ (x, x', u) \in \mathcal{X}^2 \times \mathcal{U}$,

$$\begin{aligned} \|f(x,u) - f(x',u)\|_\mathcal{X} &\leq L_f \|x - x'\|_\mathcal{X}, \\ |\rho(x,u) - \rho(x',u)| &\leq L_\rho \|x - x'\|_\mathcal{X}\ . \end{aligned}$$

Etant donné que l'espace de décision n'est plus "normé", il est nécessaire de redéfinir la notion de dispersion $\alpha^*(\mathcal{P}_n)$:

$$\alpha^*(\mathcal{P}_n) = \sup_{x \in \mathcal{X}} \min_{l \in \{1,\dots,n\}} \left\|x^l - x\right\|_\mathcal{X}\ .$$

Soit $\pi \in \Pi$ une politique en boucle ouverte. On a le résultat suivant :

Proposition 3 (Borne inférieure - Politique π)

Soit $\left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1}$ une trajectoire artificielle telle que : $u^{l_t} = \pi(t), \forall t \in \{0, \dots, T-1\}$. Alors,

$$J^\pi(x_0) \geq \sum_{t=0}^{T-1} r^{l_t} - \sum_{t=0}^{T-1} L'_{Q_{T-t}} \left\|y^{l_{t-1}} - x^{l_t}\right\|_\mathcal{X}$$

avec $L'_{Q_{T-t}} = L_\rho \sum_{i=0}^{T-t-1} (L_f)^i$.

Une borne inférieure maximale peut être calculée en maximisant la borne précédente sur l'ensemble de toutes les trajectoires artificielles satisfaisant à la condition $u^{l_t} = \pi(t) \quad \forall t \in \{0, \dots, T-1\}$. Par la suite, on note $\mathcal{F}_{n,\pi}^T$ un tel ensemble de trajectoires artificielles :

$$\mathcal{F}_{n,\pi}^T = \left\{ \left[\left(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t}\right)\right]_{t=0}^{T-1} \in \mathcal{F}_n^T | u^{l_t} = \pi(t) \quad \forall t \in 0, \dots, T-1 \right\}$$

On a alors :

$$L^\pi(\mathcal{F}_n, x_0) = \max_{[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})]_{t=0}^{T-1} \in \mathcal{F}_{n,\pi}^T} \sum_{t=0}^{T-1} r^{l_t} - \sum_{t=0}^{T-1} L'_{Q_{T-t}} \left\|y^{l_{t-1}} - x^{l_t}\right\|_\mathcal{X}\ .$$

De manière analogue, une borne supérieure minimale $U^\pi(\mathcal{F}_n, x_0)$ peut être calculée :

$$U^\pi(\mathcal{F}_n, x_0) = \min_{[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})]_{t=0}^{T-1} \in \mathcal{F}_{n,\pi}^T} \sum_{t=0}^{T-1} r^{l_t} + \sum_{t=0}^{T-1} L'_{Q_{T-t}} \left\|y^{l_{t-1}} - x^{l_t}\right\|_\mathcal{X}\ .$$

Ces deux bornes sont "fines" au sens de la caractérisation suivante :

Proposition 4 (Précision des bornes - Politique π)

$$\begin{aligned} \exists C'_b > 0 : \quad J^\pi(x_0) - L^\pi(\mathcal{F}_n, x_0) &\leq C'_b \alpha^*(\mathcal{P}_n)\ , \\ U^\pi(\mathcal{F}_n, x_0) - J^\pi(x_0) &\leq C'_b \alpha^*(\mathcal{P}_n)\ . \end{aligned}$$

Les preuves des résultats ci-dessus sont données dans [18].

4.4 Calcul de politiques prudentes à partir de trajectoires artificielles

Comme en section 4.3.1, on fait ici l'hypothèse que l'espace de décision $\mathcal{U}$ est fini, et on considère des politiques en boucle ouverte. On cherche à déterminer une politique ayant de bonnes garanties de performance, et pour cela, on propose de chercher une politique $\hat{\pi}^*_{\mathcal{F}_n,x_0} \in \Pi$ telle que :

$$\hat{\pi}^*_{\mathcal{F}_n,x_0} \in \underset{\pi\in\Pi}{\arg\max} \quad L^\pi(\mathcal{F}_n, x_0)\ .$$

On rappelle ici qu'une telle politique est optimisée pour un état initial fixé x_0. La résolution du problème formulé ci-dessus peut être vue comme la reconstruction d'une trajectoire artificielle optimale $\left[\left(x^{l_t^*}, u^{l_t^*}, r^{l_t^*}, y^{l_t^*}\right)\right]_{t=0}^{T-1}$ à partir de laquelle une séquence de décisions est extraite :

$$\forall t \in \{0,\ldots,T-1\}, \qquad \hat{\pi}^*_{\mathcal{F}_n,x_0}(t) = u^{l_t^*}\ .$$

Déterminer une politique de ce type peut à nouveau être réalisé de manière efficace en reformulant le problème comme un problème de plus court chemin dans un graphe. L'article [18] propose un algorithme nommé CGRL (de l'anglais "Cautious approach to Generalization in RL") de complexité $\mathcal{O}\left(n^2T\right)$ pour trouver une telle politique. On donne ci-dessous un théorème montrant la convergence de la politique $\hat{\pi}^*_{\mathcal{F}_n,x_0}$ vers une politique optimale lorsque la dispersion $\alpha^*(\mathcal{P}_n)$ de l'ensemble de transitions converge vers zero.

Théorème 3 (Convergence de $\hat{\pi}^*_{\mathcal{F}_n,x_0}$)

Soit $\mathfrak{J}^*(x_0)$ l'ensemble de politiques en boucle ouverte optimales :

$$\mathfrak{J}^*(x_0) = \underset{\pi\in\Pi}{\arg\max} \qquad J^\pi(x_0).$$

Supposons que $\mathfrak{J}^*(x_0) \neq \Pi$ (si $\mathfrak{J}^*(x_0) = \Pi$, la recherche d'une politique optimale est triviale). On définit :

$$\epsilon(x_0) = \min_{\pi\in\Pi\setminus\mathfrak{J}^*(x_0)} \left\{ \left(\max_{\pi'\in\Pi} J^{\pi'}(x_0) \right) - J^\pi(x_0) \right\}\ .$$

Alors,

$$\left(C_b'\alpha^*(\mathcal{P}_n) < \epsilon(x_0)\right) \implies \hat{\pi}^*_{\mathcal{F}_n,x_0} \in \mathfrak{J}^*(x_0)\ .$$

Ce résultat est prouvé dans [18]. L'algorithme CGRL est expérimentalement illustré dans l'article [22].

4.4.1 Tirer parti de trajectoires optimales

On énonce ci-dessous un résultat montrant que, dans l'éventualité où une trajectoire optimale serait présente dans l'ensemble de transitions, la politique $\hat{\pi}^*_{\mathcal{F}_n,x_0}$ calculée par l'algorithme CGRL est également optimale.

Théorème 4 (Politique optimale calculée à partir d'une trajectoire optimale)

Soit $\pi^*_{x_0} \in \mathfrak{J}^*(x_0)$ une politique en boucle ouverte optimale. Supposons que $\mathcal{F}_n$ contienne une séquence de T transitions :

$$\left[\left(x^{l_0}, u^{l_0}, r^{l_0}, x^{l_1}\right), \left(x^{l_1}, u^{l_1}, r^{l_1}, x^{l_2}\right), \ldots, \left(x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, x^{l_T}\right)\right] \in \mathcal{F}_n^T$$

telle que $x^{l_0} = x_0$ and $u^{l_t} = \pi^*_{x_0}(t) \forall t \in \{0,\ldots,T-1\}$. Soit $\hat{\pi}^*_{\mathcal{F}_n,x_0}$ telle que $\hat{\pi}^*_{\mathcal{F}_n,x_0} \in \underset{\pi\in\Pi}{\arg\max} L^\pi(\mathcal{F}_n, x_0)$. Alors,

$$\hat{\pi}^*_{\mathcal{F}_n,x_0} \in \mathfrak{J}^*(x_0).$$

La preuve de ce résultat est donnée dans [22]. Mentionnons finalement aussi que des travaux récents ont approfondi la question du calcul de politiques prudentes en généralisation, et finalement montré qu'il est possible de calculer des bornes inférieures plus précises que l'algorithme CGRL par la résolution d'une relaxation Lagrangienne des contraintes induites par la continuité lipschitzienne [16].

4.5 Stratégies d'échantillonnage à partir de trajectoires artificielles

On fait maintenant l'hypothèse que des transitions supplémentaires peuvent être générées, et on détaille ci-dessous une stratégie visant à sélectionner des paires état-action pour lesquelles générer les informations $f(x,u)$ et $\rho(x,u)$ permet de distinguer rapidement - au fur et à mesure que de nouvelles transitions sont générées - les politiques optimales des autres. Cette stratégie se fonde directement sur le calcul des bornes définies précédemment.

On donne préalablement quelques définitions. Notons tout d'abord qu'une politique est candidate à l'optimalité étant donné un ensemble de transitions $\mathcal{F}$ si sa borne supérieure n'est pas inférieure à la borne inférieure d'une autre politique (les bornes étant calculées à partir de $\mathcal{F}$). On qualifie les politiques vérifiant cette propriété de "politiques candidates étant donné $\mathcal{F}$", et on note $\Pi(\mathcal{F}, x_0)$ un tel ensemble :

$$\Pi(\mathcal{F}, x_0) = \left\{ \pi \in \Pi \quad | \quad \forall \pi' \in \Pi, U^{\pi}(\mathcal{F}, x_0) \geq L^{\pi'}(\mathcal{F}, x_0) \right\}.$$

On définit également l'ensemble de "transitions compatibles avec $\mathcal{F}$" : Une transition $(x,u,r,y) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{X}$ est dite compatible avec l'ensemble de transitions $\mathcal{F}$ si

$$\forall (x^l, u^l, r^l, y^l) \in \mathcal{F}, \quad \left(u^l = u\right) \implies \begin{cases} \left|r - r^l\right| & \leq & L_\rho \|x - x^l\|_{\mathcal{X}}, \\ \left\|y - y^l\right\|_{\mathcal{X}} & \leq & L_f \|x - x^l\|_{\mathcal{X}}. \end{cases}$$

On note $\mathcal{C}(\mathcal{F}) \subset \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{U}$ l'ensemble collectant toutes les transitions compatibles avec l'ensemble de transitions $\mathcal{F}$.

Notre stratégie d'échantillonnage génère des transitions de manière itérative. Etant donné un ensemble $\mathcal{F}_m$ de $m \in \mathbb{N} \setminus \{0\}$ transitions, composé des transitions de l'ensemble initial $\mathcal{F}_n$ et des $m-n$ transitions générées pendant les $m-n$ premières iterations de la stratégie, la paire suivante $(x^{m+1}, u^{m+1}) \in \mathcal{X} \times \mathcal{U}$ est sélectionnée en minimisant, dans les pires conditions, la plus grande différence entre borne supérieure et borne inférieure des politiques candidates :

$$(x^{m+1}, u^{m+1}) \in \underset{(x,u) \in \mathcal{X} \times \mathcal{U}}{\arg\min} \left\{ \underset{\substack{(r,y) \in \mathbb{R} \times \mathcal{X} \; s.t. (x,u,r,y) \in \mathcal{C}(\mathcal{F}_m) \\ \pi \in \Pi(\mathcal{F}_m \cup \{(x,u,r,y)\}, x_0)}}{\max} \delta^{\pi}\left(\mathcal{F}_m \cup \{(x,u,r,y)\}, x_0\right) \right\}$$

avec $\delta^{\pi}(\mathcal{F}, x_0) = U^{\pi}(\mathcal{F}, x_0) - L^{\pi}(\mathcal{F}, x_0)$.

Etant donné les propriétés de convergence des bornes, on conjecture que la séquence $\left(\Pi\left(\mathcal{F}_m, x_0\right)\right)_{m \in \mathbb{N}}$ converge vers l'ensemble des politiques optimales en un nombre fini d'itérations :

$$\exists m_0 \in \mathbb{N} \setminus \{0\} : \forall m \in \mathbb{N}, \left(m \geq m_0\right) \implies \Pi\left(\mathcal{F}_m, x_0\right) = \mathfrak{J}^*(x_0).$$

Des résultats expérimentaux générés à partir de cette stratégie sont donnés dans [22].

5 Vers un nouveau paradigme ?

Dans cette section, on met en lumière quelques connexions entre les approches présentées dans cet article et quelques algorithmes classiques du RL batch dérivés du principe de l'algorithme Fitted Q Iteration (FQI, voir [10]) lorsque ce dernier est utilisé en mode "évaluation de politique". Du point de vue du formalisme, on reprend dans cette section le formalisme stochastique donné en section 3. L'espace de décision est supposé continu et normé, et on considère une politique h en boucle fermée, dépendante du temps, et lipschitzienne $h : \{0, \ldots, T-1\} \times \mathcal{X} \to \mathcal{U}$.

5.1 Fitted Q Iteration en mode évaluation de politique

L'algorithme FQI en mode "évaluation de politique" (FQI-PE) fonctionne en calculant récursivement une suite finie de fonctions $\left(\hat{Q}^h_{T-t}(.\,,.)\right)_{t=0}^{T-1}$ de la manière suivante :

— $\forall (x,u) \in \mathcal{X} \times \mathcal{U}, \hat{Q}_0^h(x,u) = 0$
— Pour $t = T-1 \ldots 0$, construire l'ensemble $D = \left\{\left(i^l, o^l\right)\right\}_{l=1}^n$:

$$i^l = \left(x^l, u^l\right), o^l = r^l + \hat{Q}_{T-t-1}^h\left(y^l, h(t+1, y^l)\right)$$

et utiliser un algorithme de regression $\mathcal{RA}$ pour inférer à partir de D la fonction $\hat{Q}_{T-t}^h$: $\hat{Q}_{T-t}^h = \mathcal{RA}(D)$.

L'estimateur FQI -PE de la politique h est donné par : $\hat{J}_{FQI}^h(\mathcal{F}_n, x_0) = \hat{Q}_T^h(x_0, h(0, x_0))$.

5.2 FQI avec une méthode de type $k-$ plus proches voisins : point de vue "trajectoires artificielles"

On propose dans cette section d'utiliser une méthode des k plus proches voisins ($k-$NN) comme algorithme de régression $\mathcal{RA}$. Dans la suite, pour une paire état-décision $(x,u) \in \mathcal{X} \times \mathcal{U}$, on note $l_i(x,u)$ le plus petit indice dans $\mathcal{F}_n$ de la i-ème plus proche transition depuis la paire (x,u) en utilisant la distance Δ. En utilisant cette notation, l'algorithme $k-$NN FQI-PE s'écrit :

— $\forall (x,u) \in \mathcal{X} \times \mathcal{U}, \hat{Q}_0^h(x,u) = 0$,
— Pour $t = T-1 \ldots 0$, $\forall (x,u) \in \mathcal{X} \times \mathcal{U}$,

$$\hat{Q}_{T-t}^h(x,u) = \frac{1}{k} \sum_{i=1}^{k} \left(r^{l_i(x,u)} + \hat{Q}_{T-t-1}^h\left(y^{l_i(x,u)}, h\left(t+1, y^{l_i(x,u)}\right)\right)\right) .$$

L'estimateur $k-$NN FQI-PE des performances de h est donné par : $\hat{J}_{FQI}^h(\mathcal{F}_n, x_0) = \hat{Q}_T^h\left(x_0, h(0,x_0)\right)$.

On peut observer que, pour un état initial x_0 fixé, le calcul de l'estimateur $k-$NN FQI-PE identifie implicitement $(k + k^2 + \ldots + k^T)$ transitions non-distinctes pour construire l'estimation. Ces transitions sont non-distinctes dans le sens que certaines peuvent être sont sélectionnées plusieurs fois par l'algorithme $k-$NN. Dans un soucis de concision, on introduit la notation $l^{i_0,i_1,\ldots,i_t}$ pour faire référence à la transition $l_{i_t}(y^{l^{i_0,\ldots,i_{t-1}}}, h(t, y^{l^{i_0,\ldots,i_{t-1}}}))$ pour $i_0, \ldots, i_t \in \{1, \ldots, k\}$, $t \geq 1$ avec $l^{i_0} = l_{i_0}(x_0, h(0,x_0))$. A partir de ces notations, on illustre le calcul de l'estimateur $k-$NN FQI-PE en figure 3. On a alors le résultat

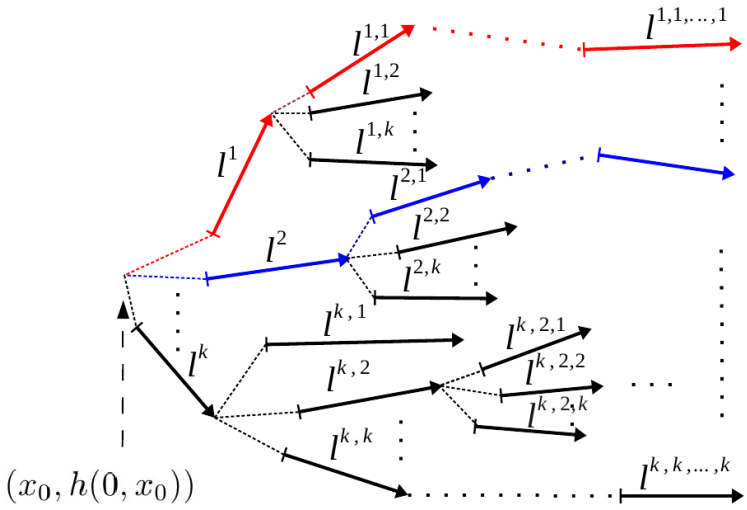


FIGURE 3 – Illustration du calcul de l'estimateur $k-$NN FQI-PE.

suivant :

Proposition 5 ($k-$NN FQI-PE via Trajectoires Artificielles)

$$\hat{J}^h_{FQI}(\mathcal{F}_n, x_0) = \frac{1}{k^T} \sum_{i_0=1}^{k} \dots \sum_{i_{T-1}=1}^{k} \left(r^{l^{i_0}} + r^{l^{i_0,i_1}} + \dots + r^{l^{i_0,i_1,\dots,i_{T-1}}} \right) .$$

où l'ensemble de trajectoires artificielles

$$\left\{ \left[\left(x^{l^{i_0}}, u^{l^{i_0}}, r^{l^{i_0}}, y^{l^{i_0}} \right), \dots, \left(x^{l^{i_0,\dots,i_{T-1}}}, u^{l^{i_0,\dots,i_{T-1}}}, r^{l^{i_0,\dots,i_{T-1}}}, y^{l^{i_0,\dots,i_{T-1}}} \right) \right] \right\}$$

est tel que $\forall t \in \{0, \dots, T-1\}, \forall (i_0, \dots, i_t) \in \{1, \dots, k\}^{t+1}$,

$$\Delta \left(\left(y^{l^{i_0,\dots,i_{t-1}}}, h\left(t, y^{l^{i_0,\dots,i_{t-1}}}\right) \right), \left(x^{l^{i_0,\dots,i_t}}, u^{l^{i_0,\dots,i_t}} \right) \right) \leq \alpha_k(\mathcal{P}_n) .$$

La preuve de ce résultat est donnée en [22]. Ce résultat montre que l'estimation $k-$NN FQI-PE est obtenue en moyennant le retour de k^T trajectoires artificielles. Ces trajectoires sont reconstruites à partir de $(k + k^2 + \dots + k^T)$ transitions non-distinctes tirées de $\mathcal{F}_n$ et choisies en minimisant la distance entre deux transitions successives (sous le contrôle de la politique h).

5.3 Approximateurs à base de noyaux

Le résultat donné en section 5.2 peut être étendu au cas où l'algorithme FQI-PE est combiné à un approximateur à base de noyaux. Dans ce contexte, la suite de fonctions $\left(\hat{Q}^h_{T-t}(.)\right)_{t=0}^{T}$ se calcule de la manière suivante :

— $\forall (x,u) \in \mathcal{X} \times \mathcal{U}$,

$$\hat{Q}^h_0(x,u) = 0 ,$$

— Pour $t = T-1 \dots 0$, $\forall (x,u) \in \mathcal{X} \times \mathcal{U}$,

$$\hat{Q}^h_{T-t}(x,u) = \sum_{l=1}^{n} k_{\mathcal{F}_n}\left((x,u),(x^l,u^l)\right) \left(r^l + \hat{Q}^h_{T-t-1}(y^l, h(t+1, y^l)) \right) ,$$

avec $k_{\mathcal{F}_n}((x,u),(x^l,u^l)) = \dfrac{\Phi\left(\frac{\Delta\left((x,u),(x^l,u^l)\right)}{b_n}\right)}{\sum_{i=1}^{n} \Phi\left(\frac{\Delta\left((x,u),(x^i,u^i)\right)}{b_n}\right)}$ où $\Phi : [0,1] \to \mathbb{R}^+$ est un noyau, $b_n > 0$ le paramètre de bande passante. On fait l'hypothèse que $\int_0^1 \Phi(z)dz = 1$ and $\Phi(x) = 0 \; \forall x > 1$.

L'estimateur KB FQI-PE du retour espéré de h s'écrit : $\hat{J}^h_{FQI}(\mathcal{F}_n, x_0) = \hat{Q}^h_T(x_0, h(0, x_0))$.

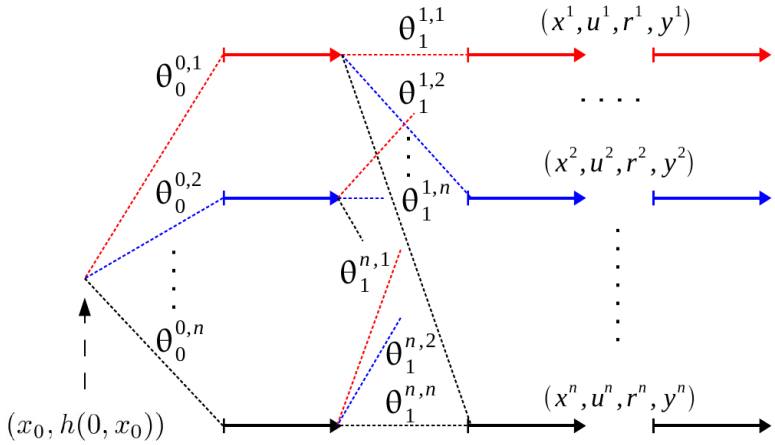
Etant donné un état initial $x_0 \in \mathcal{X}$, le calcul de l'estimation KB FQI-PE peut également être interprété comme l'identification d'un ensemble de transitions tirées de $\mathcal{F}_n$. A chaque pas de temps t, toutes les transitions (x^l, u^l, r^l, y^l) dans un rayon inférieur à b_n du point $(x_t, h(t, x_t))$ sont sélectionnées et pondérées en fonction de la distance. Ce procédé est opéré récursivement en sortie de chaque transition sélectionnée. Une illustration de l'estimateur KB-FQI-PE est donnée à la figure 4. La valeur retournée par l'estimateur KB FQI-PE peut s'exprimer ainsi :

Proposition 6

$$\hat{J}^h_{FQI}(\mathcal{F}_n, x_0) = \sum_{i_0=1}^{n} \dots \sum_{i_{T-1}=1}^{n} \theta_0^{0,i_0} \theta_1^{i_0,i_1} \dots \theta_{T-1}^{i_{T-2},i_{T-1}} \left(r^{i_0} + \dots + r^{i_{T-1}} \right)$$

avec

$$\theta_0^{0,i_0} = k_{\mathcal{F}_n}\left((x_0, h(0,x_0)), (x^{i_0}, u^{i_0}) \right) ,$$
$$\theta_{t+1}^{i_t,i_{t+1}} = k_{\mathcal{F}_n}\left((y^{i_t}, h(t+1, y^{i_t})), (x^{i_{t+1}}, u^{i_{t+1}}) \right), \forall t \in \{0, \dots, T-2\} .$$



$$\theta_0^{0,i_0} = k_{\mathcal{F}_n}\Big((x_0, h(0,x_0)), (x^{i_0}, u^{i_0})\Big) \ ,$$
$$\theta_{t+1}^{i_t,i_{t+1}} = k_{\mathcal{F}_n}\Big((y^{i_t}, h(t+1,y^{i_t})), (x^{i_{t+1}}, u^{i_{t+1}})\Big), \forall t \in \{0,\dots,T-2\} \ .$$

FIGURE 4 – Illustration du calcul de l'estimateur KB FQI-PE.

La preuve de ce résultat est donnée dans [22].

La proposition 6 montre que l'estimation KB FQI-PE s'exprime sous forme d'une somme pondérée des retours de n^T trajectoires artificielles. Chaque trajectoire

$$[(x^{i_0}, u^{i_0}, r^{i_0}, y^{i_0}), (x^{i_1}, u^{i_1}, r^{i_1}, y^{i_1}), \dots, (x^{i_{T-1}}, u^{i_{T-1}}, r^{i_{T-1}}, y^{i_{T-1}})]$$

est pondérée par un facteur $\theta_0^{0,i_0}\theta_1^{i_0,i_1}\dots\theta_{T-1}^{i_{T-2},i_{T-1}}$ (certains de ces coefficients pouvant éventuellement être nuls). De manière analogue à l'estimateur $k-$NN FQI-PE, ces trajectoires artificielles sont reconstruites à partir de $T \times n^T$ transitions non-distinctes.

Plus généralement, la notion de trajectoire artificielle pourrait être utilisée pour caractériser d'autres techniques de RL batch fondées sur l'utilisation d'approximateurs de fonctions utilisant des schémas de "moyennage" [23].

On oriente le lecteur vers l'article [22] dans lequel une analyse empirique de l'impact de la réutilisation de transitions dans la construction des trajectoires artificielles est proposée. On y décrit un exemple pour lequel l'approche MFMC présente un meilleur compromis biais-variance que l'approche $k-$NN FQI-PE.

6 Conclusion

Ce papier revisite des travaux récents de l'apprentissage par renforcement batch sous l'angle de la reconstruction de trajectoires artificielles. On montre que ce nouveau paradigme est prometteur, autant dans l'analyse d'algorithmes existants, que dans la création de nouvelles approches.

Remerciements

Raphael Fonteneau est Chargé de Recherches F.R.S.-FNRS. Les auteurs remercient également le réseau d'excellence PASCAL2 et le PAI Belge DYSCO, ainsi que le support financier de la NIH, grants P50 DA10075 et R01 MH080015.

Références

[1] ANTOS A., MUNOS R. & SZEPESVÁRI C. (2007). Fitted Q-iteration in continuous action space MDPs. In *Advances in Neural Information Processing Systems 20 (NIPS)*.

[2] BELLMAN R. (1957). *Dynamic Programming*. Princeton University Press.

[3] BONARINI A., CACCIA C., LAZARIC A. & RESTELLI M. (2008). Batch reinforcement learning for controlling a mobile wheeled pendulum robot. *Artificial Intelligence in Theory and Practice II*, p. 151–160.

[4] BOYAN J. (2005). Technical update : Least-squares temporal difference learning. *Machine Learning*, **49**, 233–246.

[5] BRADTKE S. & BARTO A. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, **22**, 33–57.

[6] CASTELLETTI A., DE RIGO D., RIZZOLI A., SONCINI-SESSA R. & WEBER E. (2007). Neuro-dynamic programming for designing water reservoir network management policies. *Control Engineering Practice*, **15**(8), 1031–1038.

[7] CASTELLETTI A., GALELLI S., RESTELLI M. & SONCINI-SESSA R. (2010). Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, **46**.

[8] CHAKRABORTY B., STRECHER V. & MURPHY S. (2008). Bias correction and confidence intervals for fitted Q-iteration. In *Workshop on Model Uncertainty and Risk in Reinforcement Learning, NIPS, Whistler, Canada*.

[9] DEFOURNY B., ERNST D. & WEHENKEL L. (2008). Risk-aware decision making and dynamic programming. In *Workshop on Model Uncertainty and Risk in Reinforcement Learning, NIPS, Whistler, Canada*.

[10] ERNST D., GEURTS P. & WEHENKEL L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, **6**, 503–556.

[11] ERNST D., GLAVIC M., CAPITANESCU F. & WEHENKEL L. (2009). Reinforcement learning versus model predictive control : a comparison on a power system problem. *IEEE Transactions on Systems, Man, and Cybernetics - Part B : Cybernetics*, **39**, 517–529.

[12] ERNST D., MARÉE R. & WEHENKEL L. (2006a). Reinforcement learning with raw image pixels as state input. In *International Workshop on Intelligent Computing in Pattern Analysis/Synthesis (IWICPAS). Proceedings series : Lecture Notes in Computer Science*, volume 4153, p. 446–454.

[13] ERNST D., STAN G., GONCALVES J. & WEHENKEL L. (2006b). Clinical data based optimal STI strategies for HIV : a reinforcement learning approach. In *Machine Learning Conference of Belgium and The Netherlands (BeNeLearn)*, p. page 65–72.

[14] FARAHMAND A., GHAVAMZADEH M., SZEPESVÁRI C. & MANNOR S. (2008). Regularized fitted q-iteration : Application to planning. In S. GIRGIN, M. LOTH, R. MUNOS, P. PREUX & D. RYABKO, Eds., *Recent Advances in Reinforcement Learning*, volume 5323 of *Lecture Notes in Computer Science*, p. 55–68. Springer Berlin / Heidelberg.

[15] FONTENEAU R. (2011). *Contributions to Batch Mode Reinforcement Learning*. PhD thesis, University of Liège.

[16] FONTENEAU R., ERNST D., BOIGELOT B. & LOUVEAUX Q. (2013a). Min max generalization for deterministic batch mode reinforcement learning : relaxation schemes. *SIAM Journal on Control and Optimization*, **51**(5), 3355–3385.

[17] FONTENEAU R., MURPHY S., WEHENKEL L. & ERNST D. (2009). Inferring bounds on the performance of a control policy from a sample of trajectories. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, Nashville, TN, USA.

[18] FONTENEAU R., MURPHY S., WEHENKEL L. & ERNST D. (2010a). A cautious approach to generalization in reinforcement learning. In *Second International Conference on Agents and Artificial Intelligence (ICAART)*, Valencia, Spain.

[19] FONTENEAU R., MURPHY S., WEHENKEL L. & ERNST D. (2010b). Generating informative trajectories by using bounds on the return of control policies. In *Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010)*.

[20] FONTENEAU R., MURPHY S., WEHENKEL L. & ERNST D. (2010c). Model-free Monte Carlo–like policy evaluation. In *Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR : W&CP 9*, p. 217–224, Chia Laguna, Sardinia, Italy.

[21] FONTENEAU R., MURPHY S. A., WEHENKEL L. & ERNST D. (2011). Towards min max generalization in reinforcement learning. In *Agents and Artificial Intelligence : International Conference, ICAART 2010, Valencia, Spain, January 2010, Revised Selected Papers. Series : Communications in Computer and Information Science (CCIS)*, volume 129, p. 61–77 : Springer, Heidelberg.

[22] FONTENEAU R., MURPHY S. A., WEHENKEL L. & ERNST D. (2013b). Batch mode reinforcement learning based on the synthesis of artificial trajectories. *Annals of Operations Research*, **208**(1), 383–416.

[23] GORDON G. (1995). Stable function approximation in dynamic programming. In *Twelfth International Conference on Machine Learning (ICML)*, p. 261–268.

[24] GUEZ A., VINCENT R., AVOLI M. & PINEAU J. (2008). Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *Innovative Applications of Artificial Intelligence (IAAI)*.

[25] LAGOUDAKIS M. & PARR R. (2003). Least-squares policy iteration. *Jounal of Machine Learning Research*, **4**, 1107–1149.

[26] LANGE S. & RIEDMILLER M. (2010). Deep learning of visual control policies. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Brugge, Belgium*.

[27] LAZARIC A., GHAVAMZADEH M. & MUNOS R. (2010a). *Finite-sample analysis of least-squares policy iteration*. Rapport interne, SEQUEL (INRIA) Lille - Nord Europe.

[28] LAZARIC A., GHAVAMZADEH M. & MUNOS R. (2010b). Finite-sample analysis of LSTD. In *International Conference on Machine Learning (ICML)*, p. 615–622.

[29] MORIMURA T., SUGIYAMA M., KASHIMA H., HACHIYA H. & TANAKA T. (2010a). Nonparametric return density estimation for reinforcement learning. In *27th International Conference on Machine Learning (ICML), Haifa, Israel, Jun. 21-25*.

[30] MORIMURA T., SUGIYAMA M., KASHIMA H., HACHIYA H. & TANAKA T. (2010b). Parametric return density estimation for reinforcement learning. In *26th Conference on Uncertainty in Artificial Intelligence (UAI), Catalina Island, California, USA Jul. 8-11*, p. 368–375.

[31] MUNOS R. & SZEPESVÁRI C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, p. 815–857.

[32] MURPHY S. (2003). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society, Series B*, **65(2)**, 331–366.

[33] MURPHY S., VAN DER LAAN M. & ROBINS J. (2001). Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, **96**(456), 1410–1423.

[34] NEDIC A. & BERTSEKAS D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, **13**, 79–110. 10.1023/A :1022192903948.

[35] ORMONEIT D. & SEN S. (2002). Kernel-based reinforcement learning. *Machine Learning*, **49**(2-3), 161–178.

[36] PETERS J., VIJAYAKUMAR S. & SCHAAL S. (2003). Reinforcement learning for humanoid robotics. In *Third IEEE-RAS International Conference on Humanoid Robots*, p. 1–20 : Citeseer.

[37] PIETQUIN O., TANGO F. & ARAS R. (2011). Batch reinforcement learning for optimizing longitudinal driving assistance strategies. In *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2011 IEEE Symposium on*, p. 73–79 : IEEE.

[38] RIEDMILLER M. (2005). Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *Sixteenth European Conference on Machine Learning (ECML)*, p. 317–328, Porto, Portugal.

[39] ROBINS J. (1986). A new approach to causal inference in mortality studies with a sustained exposure period–application to control of the healthy worker survivor effect. *Mathematical Modelling*, **7**(9-12), 1393–1512.

[40] SANI A., LAZARIC A. & MUNOS R. (2012). Risk-aversion in multi-armed bandits. In *Advances in Neural Information Processing Systems 25 (NIPS)*, p. 3284–3292.

[41] SCHERRER B. (2013). Improved and generalized upper bounds on the complexity of policy iteration. In C. BURGES, L. BOTTOU, M. WELLING, Z. GHAHRAMANI & K. WEINBERGER, Eds., *Advances in Neural Information Processing Systems 26*, p. 386–394.

[42] SUTTON R. & BARTO A. (1998). *Reinforcement Learning*. MIT Press.

[43] TIMMER S. & RIEDMILLER M. (2007). Fitted Q iteration with cmacs. In *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, p. 1–8 : IEEE.

[44] TOGNETTI S., SAVARESI S., SPELTA C. & RESTELLI M. (2009). Batch reinforcement learning for semi-active suspension control. In *Control Applications (CCA) & Intelligent Control (ISIC), 2009 IEEE*, p. 582–587.