Choix de la fonction de renforcement et des valeurs initiales pour accélérer les problèmes d'Apprentissage par Renforcement de plus court chemin stochastique.

Laëtitia Matignon, Guillaume J. Laurent et Nadine Le Fort-Piat
Laboratoire d'Automatique de Besançon UMR CNRS 6596,
24 rue Alain Savary, 25000 Besançon, France
E-mails :{laetitia.matignon, guillaume.laurent, nadine.piat}@ens2m.fr

Résumé

Un point important en apprentissage par renforcement (AR) est l'amélioration de la vitesse de convergence du processus d'apprentissage. Nous proposons dans cet article d'étudier l'influence de certains paramètres de l'AR sur la vitesse d'apprentissage. En effet, bien que les propriétés de convergence de l'AR ont été largement étudiées, peu de règles précises existent pour choisir correctement la fonction de renforcement et les valeurs initiales de la table Q. Notre méthode aide au choix de ces paramètres dans le cadre de problèmes de type *goal-directed*, c'est-à-dire dont l'objectif est d'atteindre un but en un minimum de temps. Nous développons une étude théorique et proposons ensuite des justifications expérimentales pour choisir d'une part la fonction de renforcement et d'autre part des valeurs initiales particulières de la table Q, basées sur une fonction d'influence.

Mots clés : apprentissage par renforcement de type *goal-directed*, fonction de renforcement, initialisation de la table Q, fonction d'influence, *reward shaping*

1 Introduction

L'apprentissage par renforcement (AR) [10] est une technique permettant à un agent, interagissant avec un environnement, de résoudre de manière autonome des tâches grâce à un système de récompenses. La plupart des algorithmes d'AR se placent dans le cadre des processus décisionnels de Markov. L'agent apprend par *essais et erreurs* à sélectionner, pour chaque couple d'état-action, les actions qui vont lui permettre de maximiser la somme de ses récompenses futures, ou *espérance de gain*. L'algorithme du Q-learning [11] est l'un des plus usuel en AR. La stratégie optimale y est apprise de manière implicite sous la forme d'une fonction de valeur Q. La convergence de cet algorithme a été démontrée [12].

Une des principales limitations des algorithmes d'AR est *la lenteur de convergence*. Ainsi, plusieurs méthodes proposent d'accélérer l'AR. Elles nécessitent l'incorporation de connaissance *a priori* ou de conseils dans l'AR. Dans cette optique, Garcia [3] présente un état de l'art des différentes techniques d'*exploration guidée* et propose deux méthodes permettant d'ajouter de la connaissance *a priori* afin de guider l'agent dans son exploration. Pour cela, il s'inspire des méthodes introduisant les macro-actions pour contraindre le nombre d'actions à considérer dans chaque état et propose des mécanismes de relâchement progressif de ces contraintes. L'intérêt est de diminuer l'espace de recherche et donc d'espérer résoudre le problème plus rapidement. D'autres techniques pour guider l'exploration consistent à *incorporer un conseil* directement dans la fonction de renforcement ou la fonction de valeur Q. Les méthodes par estimateurs de progrès [7], par fonctions de potentiel [13], par *reward shaping* [8,9], par imitation [1] ou l'initialisation de la table Q [4–6] en font ainsi partie.

Mataric [7] propose une méthode pour *choisir des fonctions de récompenses* utilisant les connaissances implicites sur l'environnement. Elle implique l'utilisation de fonctions de récompenses continues et d'*estimateurs de progrès*. De même, dans la méthode par *reward shaping*, on ajoute des récompenses additionnelles basées sur des fonctions de potentiels aux récompenses reçues de l'environnement [8]. Un exemple classique du *reward shaping* concerne le problème de l'apprentissage de la conduite d'un vélo [9]. Cependant, ces méthodes peuvent amener l'agent à apprendre des stratégies sous-optimales et ainsi, à piéger le système. Wiewiora [13] complète l'étude du *reward shaping* et de plus, démontre certaines similarités entre les méthodes basées sur les fonctions de potentiel et *l'initialisation de la table Q*. En effet, la technique la plus élémentaire pour influencer l'apprentissage est le choix des valeurs initiales de la table Q [4, 6]. Koenig et Simmons [5] étudient diverses représentations des fonctions de récompenses

et analysent la complexité des algorithmes basés sur le Q-learning selon ces représentations. Finalement, concernant *l'apprentissage par imitation*, Behnke et Bennewitz [1] proposent de donner accès à l'agent aux valeurs de la table Q d'un agent expérimenté.

Ainsi, la fonction de renforcement et l'initialisation de la table Q jouent un rôle important dans l'AR. Néanmoins, bien que l'AR a été largement étudié et ses propriétés de convergence bien connues, en pratique, on choisit souvent la fonction de renforcement de manière intuitive et les valeurs initiales de la table Q de manière arbitraire [10]. Dans cet article, les effets des paramètres de l'AR sur la stratégie sont discutés afin de suggérer une analyse générique. Nous validons notre étude avec l'algorithme du Q-learning. L'objectif principal est de *proposer une méthode pour correctement initialiser les paramètres de l'AR en vue d'obtenir le comportement optimal désiré en un minimum de temps, dans le cadre de problèmes de plus court chemin stochastique.*

2 Apprentissage par renforcement

Le cadre de la plupart des algorithmes d'AR est celui des *processus décisionnels de Markov* (PDM), défini comme un ensemble fini d'états, S, un ensemble fini d'actions, A, et une fonction de transition $T : S \times A \times S \rightarrow [0; 1]$ donnant pour chaque état et action la probabilité de transition entre états. $R : S \times A \times S \rightarrow \mathbb{R}$ est la fonction de renforcement calculant la récompense immédiate ou *renforcement* obtenue pour chaque transition effectuée. Le but est d'apprendre une table d'état-action, appelée *stratégie* et notée π, qui maximise la somme pondérée des récompenses futures [1] : $\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}, s_{t+k+1})$.

Nous avons validé notre étude avec l'algorithme du *Q-learning* [11]. Dans cet algorithme, une *fonction de valeur* $Q^\pi(s, a)$ est estimée durant le processus d'apprentissage et mémorisée dans un tableau. La *fonction de valeur* représente la somme des récompenses futures espérées que l'agent espère recevoir en exécutant l'action a depuis l'état s et en suivant la stratégie π. La fonction de valeur de l'action optimale Q^* est l'unique solution de l'équation de Bellman,

$$Q^*(s, a) = \sum_{\forall s' \in S} T(s, a, s') \Big[R(s, a, s') + \gamma \max_{a'} Q^*(s', a') \Big] \quad (1)$$

Le Q-learning est une méthode hors-ligne dont l'équation de mise à jour est :

$$Q(s, a) \leftarrow Q(s, a) + \alpha \Big[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \Big] \quad (2)$$

où r est la récompense reçue pour la transition de l'état s au nouvel état s' après l'exécution de l'action a. $\alpha \in]0; 1]$ est le coefficient d'apprentissage et $\gamma \in [0; 1[$ le coefficient d'actualisation.
Sous certaines conditions [2], l'algorithme du Q-learning est garanti de converger vers la fonction de valeur optimale [12]. La sélection de l'action se fait selon un critère d'exploration/exploitation. Nous avons utilisé *la méthode ϵ-glouton* dans laquelle la probabilité de choisir une action aléatoire est ϵ, et sinon, l'action sélectionnée est celle ayant, pour l'état courant, la plus grande valeur de Q [3].

3 Choix des valeurs initiales avec des récompenses binaires

Nous supposons que deux tendances se démarquent lors de l'apprentissage : *une stratégie globale* et *un comportement spécifique en début d'apprentissage*. Dans cette partie, nous allons préciser ces deux tendances qui dépendent du choix de la fonction de renforcement et de l'initialisation de la table Q. Tout d'abord, nous avons étudié le cas d'une *fonction de renforcement binaire*, ce qui a pour avantage d'inclure un grand nombre de cas et de permettre d'extrapoler nos résultats.

3.1 Stratégie optimale

La fonction de renforcement binaire est telle que la récompense reçue est toujours r_∞ excepté si le nouvel état est l'état cible à atteindre. Dans ce cas, la récompense est r_g. On a ainsi :

$$\forall s \in S \ \forall a \in A, \ R(s, a, s') = \begin{cases} r_g \text{ si } s' = s_g \\ r_\infty \text{ sinon} \end{cases} \quad (3)$$

où s' est l'état obtenu en effectuant l'action a depuis l'état s, et s_g l'état cible. Dans le cas où toutes les récompenses sont identiques ($r_g = r_\infty$), la solution de l'équation de Bellman (1) est une constante notée Q_∞,

$$\forall s \ \forall a \ Q^*(s, a) = Q_\infty = \frac{r_\infty}{1 - \gamma} \quad (4)$$

En d'autres termes, pendant le processus d'apprentissage, les valeurs de la table Q pour tous les couples d'état-action convergent vers Q_∞. Par contre, si $r_g \neq r_\infty$, Q_∞ est la limite de la fonction de valeur de l'action $Q^*(s, a)$ dans le cas où la distance entre s et s_g tend vers l'infini. Donc, selon les valeurs de r_g et

[1] Les récompenses sont actualisées par un coefficient d'atténuation γ qui contrôle la balance entre l'importance des récompenses immédiates et futures.

[2] Les conditions sur α sont $\sum_{t=0}^{\infty} \alpha_t = \infty$ et $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$. Usuellement, on utilise un facteur constant $\alpha \in]0; 1]$.

[3] Si plusieurs valeurs de Q sont identiques, le choix est aléatoire parmi les actions gloutonnes.

de Q_∞, les états sont *de plus en plus* ou *de moins en moins attirants* lorsque l'on se rapproche de l'état cible. Dans un cas, si $r_g > Q_\infty$, la valeur de Q pour les couples d'état-action menant à l'état cible est de plus en plus attirante que Q_∞. Donc la stratégie optimale globale est *le plus court chemin vers* s_g. D'un autre côté, si $r_g < Q_\infty$, la stratégie optimale est un comportement aléatoire partout excepté *une répulsion locale de* s_g.

Évidemment, le plus court chemin vers l'état cible est la stratégie optimale recherchée dans le cas de problèmes de plus court chemin stochastique. Donc r_g *doit toujours être supérieur à* Q_∞.

3.2 Comportement en début d'apprentissage

De même que la fonction de renforcement, les valeurs initiales Q_i de la fonction de valeur influencent la stratégie, mais seulement au début du processus d'apprentissage. Nous pensons qu'une tendance générale se distingue pendant les premiers épisodes de l'apprentissage.

Étudions les valeurs de la table Q en début d'apprentissage. Si nous calculons la première mise à jour d'un couple état-action (s, a) avec l'équation (2), tel que l'état suivant s' ne soit pas l'état cible et n'ayant jamais été mis à jour, on obtient :

$$\begin{aligned} Q(s,a) &\leftarrow Q_i + \alpha\left[r_\infty + (\gamma - 1)Q_i\right] \\ &\leftarrow Q_i + \alpha(1-\gamma)(Q_\infty - Q_i)\ . \end{aligned} \tag{5}$$

Donc la valeur discriminante de Q_i est aussi Q_∞. En fonction des valeurs de Q_i par rapport à Q_∞, *les états déjà visités seront plus ou moins attractifs* tant que l'agent n'a pas atteint un grand nombre de fois l'état cible.

- **Si** $Q_i > Q_\infty$ **:** les états déjà visités auront une valeur inférieure à celle des états non visités ($Q(s,a) < Q_i$). En d'autres termes, les états non visités seront donc plus attirants, ce qui induit l'agent à explorer. Cette *exploration est plus systématique* en début d'apprentissage que dans le cas de l'exploration aléatoire. Nous nommons ce comportement l'*exploration systématique*.
- **Si** $Q_i < Q_\infty$ **:** les états déjà visités auront une valeur supérieure à celle des états non visités ($Q(s,a) > Q_i$). C'est-à-dire que les états déjà visités seront plus attirants. Ceci induit une moins grande exploration de l'agent en début d'apprentissage . Ce comportement, que nous nommons *piétinement*, ralentit considérablement l'apprentissage. Il est donc préférable de l'éviter.
- **Si** $Q_i = Q_\infty$ **:** les états déjà visités auront une valeur identique à celle des états non visités ($Q(s,a) = Q_i$).

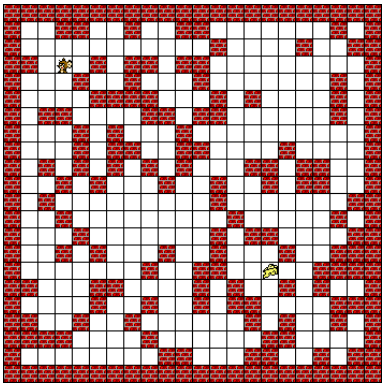

FIG. 1 – Labyrinthe non-déterministe de 20×20 cases avec un état initial (état $[2, 2]$) et un état cible (le fromage) (état $[14, 14]$).

Le comportement sera *purement aléatoire* en début d'apprentissage.

3.3 Expériences sur le labyrinthe

Nous avons discuté précédemment de la manière dont les fonctions de récompenses et des valeurs initiales arbitraires de la table Q peuvent ralentir l'apprentissage d'une stratégie intéressante. Nous allons maintenant valider cette analyse et, pour des raisons de simplicité et de clarté, nous choisissons tout d'abord un labyrinthe non-déterministe pour démontrer comment le comportement de l'agent est influencé. Nous utilisons en premier lieu des récompenses binaires et différentes valeurs initiales de Q.

Benchmark. Le système est représenté par une souris évoluant dans un damier (Fig. 1). A chaque position de la souris correspond un état discret. Quand la souris atteint l'état cible, l'épisode se termine. La souris a le choix entre quatre actions, selon ses intentions de se déplacer dans l'une des quatre directions cardinales (N,E,S,O). Si une action entraîne la souris dans un mur, celle-ci reste dans son état courant. La souris atteint l'état désigné avec une probabilité de 0.6, et sinon, elle se retrouve de façon aléatoire dans un des quatre états voisins de l'état désigné. Pour tous les épisodes, l'algorithme du Q-learning est paramétré avec un coefficient d'apprentissage α de 0.1, un coefficient d'actualisation γ de 0.9, une table de Q initialisée uniformément à la valeur Q_i et l'agent suit une stratégie où l'action gloutonne est choisit avec une probabilité de 0.9 ($\epsilon = 0.1$).

Fonction de renforcement binaire. La fonction de renforcement est identique à celle de (3), avec $r_g = 1$ et $r_\infty = 0$. Avec une telle fonction de renforcement, la stratégie optimale est *le plus court chemin vers l'état cible*. La valeur discriminante de Q_i est 0 (Q_∞=0).

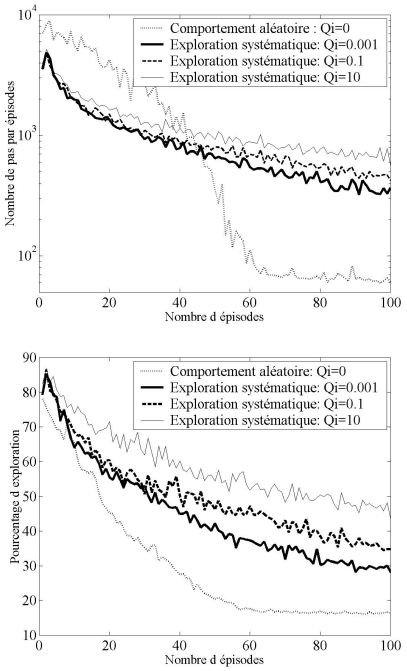

FIG. 2 – En haut, nombre de pas par épisodes pour atteindre l'état cible en fonction du nombre d'épisodes. En bas, pourcentage d'exploration de l'espace d'états induit par différentes valeurs de Q_i. Expériences sur le labyrinthe moyennées sur 50 essais indépendants, avec différentes valeurs de Q_i. Ces deux méthodes illustrent les comportements aléatoires et d'exploration systématique en tant que tendance de début d'apprentissage.

La figure 2 illustre notre étude précédente. En effet, on constate que l'utilisation du *comportement par exploration systématique* favorise l'accélération de l'apprentissage pendant les premiers épisodes. Avec une exploration plus systématique, le pourcentage d'états visités est supérieur à celui d'une exploration aléatoire. En effet, l'agent a visité chaque recoin du labyrinthe et l'état cible a ainsi été découvert plus rapidement.
Néanmoins, dans le cas d'un comportement purement aléatoire, la courbe représentant le nombre de pas par épisodes converge vers une limite qui est le nombre minimum d'états à visiter si l'agent suit le plus court chemin vers l'état cible. Dans le cas de l'exploration systématique, l'agent est constamment incité à explorer l'environnement et c'est pourquoi il effectue un plus grand nombre de pas pour atteindre l'état cible. De plus, nous avons utilisé différentes valeurs de Q_i pour expérimenter l'exploration systématique. Nous remarquons que plus Q_i est supérieur à Q_∞, plus l'agent explore. Donc *plus la différence entre Q_i et Q_∞ est importante, plus le comportement spécifique de début d'apprentissage s'accentue.*

Concernant le *comportement de piétinement* ($Q_i < 0$), nous ne soumettons pas d'expérimentations. En effet, dans notre cas, les épisodes ne se terminent pas car l'agent tourne en rond dans une zone de l'environnement où les mises à jour de la table Q sont fonction de γ^n, avec n le nombre d'épisodes. Ainsi, la fonction de valeur Q dans la zone de piétinement converge vers zéro. Or, zéro est une valeur très bien approximée [4]. Il est évident que *le comportement de piétinement doit être évité.*

3.4 Conclusion

Ceci met en évidence *l'importance des valeurs initiales de la table Q.* Le choix de Q_i n'est pas trivial et doit être fait en accord avec le comportement désiré. Dans les cas où le système s'éloigne naturellement de l'état cible, l'exploration systématique peut être préférée afin d'accélérer l'apprentissage au début. En effet, l'exploration systématique force le système à explorer des états nouveaux, et ainsi à se rapprocher de l'état cible.

4 Choix d'une fonction de renforcement continue et de valeurs initiales hétérogènes

Afin d'élargir le cadre de notre étude, nous proposons maintenant d'utiliser tout d'abord des *fonctions de récompenses continues* avec une initialisation uniforme de la table Q. Nous nous intéresserons ensuite à une initialisation particulière de la fonction de valeur avec *une fonction d'influence.*

4.1 Fonction de renforcement utilisant des estimateurs de progrès

Nous proposons tout d'abord d'étudier le cas d'une fonction de renforcement continue au lieu des récompenses binaires. Certains auteurs conçoivent ainsi les fonctions de récompenses grâce aux **estimateurs de progrès** [7] ou aux fonctions de potentiel [13]. Les estimateurs de progrès fournissent une mesure de progrès liée à un objectif. Ils ne procurent pas une information complète mais seulement un *conseil* partiel. Par exemple, dans le cas du labyrinthe, un estimateur de progrès peut être une moyenne du nombre de pas nécessaire pour atteindre l'état final depuis le nouvel état s', définie comme $\varphi(s', a) = d(s', s_g)$. d est la distance manhattan entre le nouvel état s' et s_g. L'objectif de l'agent dans le labyrinthe est alors de minimiser cette fonction et les paramètres pourraient

[4] Le point de rupture est autour de $-1.7e^{-308}$. Si la fonction de renforcement avait été différente, telle que $Q_\infty \neq 0$ par exemple, les valeurs de la table Q dans la zone de piétinement seraient devenues homogènes beaucoup plus rapidement, l'approximation de valeurs différentes de zéro étant moins efficace. L'agent aurait donc exploré petit à petit son environnement proche et le comportement de piétinement aurait duré moins longtemps.

être :

$$\begin{cases} \mathrm{r}(s,a,s') = -\varphi^2 = -d^2(s',s_g) \\ Q_i = 0 \end{cases} \quad (6)$$

Ainsi, l'agent est de moins en moins punit lorsqu'il s'approche de l'état cible. La stratégie globale est le plus court chemin vers l'état cible. Néanmoins, étant donné notre labyrinthe, cette forme de récompenses est trompeuse pour l'agent car il y a de nombreux murs entre l'état initial et l'état cible. En particulier, cela engendre un *phénomène de désapprentissage* après quelques épisodes : l'agent se bloque dans une impasse. En effet, si l'agent s'aventure dans une impasse (qui le rapproche de l'état cible au sens de la distance manhattan), il ne pourra en sortir qu'en explorant car les états sont de plus en plus attirants vers l'état cible. Au début de l'apprentissage, les valeurs de Q sont proches de 0 donc l'exploration systématique est forte : il est possible pour l'agent de sortir d'une impasse. Mais après quelques épisodes, revenir en arrière est équivalent à choisir une valeur de Q moins attirante. Cela est possible seulement si plusieurs actions d'exploration se succèdent, c'est-à-dire *rarement*.

Les méthodes par estimateurs de progrès et fonctions de potentiel sont donc risquées. Il est donc préférable d'utiliser ces approches avec circonspection dans la mesure où elles peuvent conduire à un comportement pernicieux.

4.2 Fonction de renforcement continue inspirée d'une fonction gaussienne

Par conséquent, nous proposons un fonction de renforcement continue telle que d'une part, *r soit uniforme* pour un certain nombre d'états loin de l'état cible afin d'éviter le phénomène de désapprentissage, et d'autre part, qu'il y ait un *gradient de récompenses* dans une zone autour de l'état cible.
Nous suggérons la fonction de renforcement inspirée de *la fonction gaussienne* :

$$\mathrm{r}(s,a,s') = \beta e^{-\frac{d(s',s_g)^2}{2\sigma^2}} \ . \quad (7)$$

Les valeurs de Q_i sont uniformes, β ajuste l'amplitude de la fonction et σ, l'écart type, caractérise la *zone d'influence du gradient de récompenses*. Bien entendu, le comportement de piétinement devra être évité, c'est-à-dire $Q_i \geq \frac{\beta}{1-\gamma}$.
Pour nos expériences avec le labyrinthe, nous avons choisi $Q_i = 100$ et $\beta = 10$. Sur la fig. 3, le phénomène de désapprentissage est mis en évidence dès 80 épisodes avec $\sigma = 3.5$ [5]. En effet, la zone d'influence du gradient de récompenses est trop large et quelques impasses y sont inclus. A l'opposé, si la zone d'influence du gradient n'est activée que pour les états distants de 6 pas de s_g ($\sigma = 2$), il n'y aura aucun phénomène de désapprentissage et le processus d'apprentissage sera alors accéléré.

[5] c'est-à-dire que tous les états éloignés de plus de 10 pas de l'état cible ont une récompense r identique.

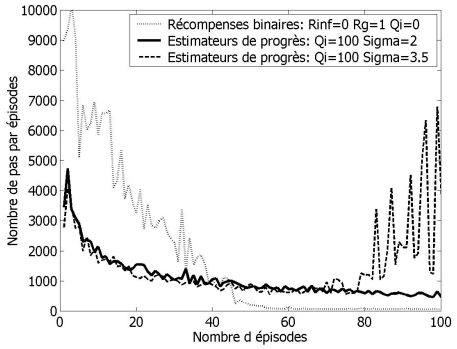

FIG. 3 – Nombre de pas pour atteindre l'état cible en fonction du nombre d'épisodes. Expériences sur le labyrinthe moyennées sur 20 essais indépendants, avec une fonction de renforcement inspirée d'une fonction gaussienne. $Q_i = 100$; $\mathrm{r}(s,a,s') = 10e^{-\frac{d(s',s_g)^2}{2\sigma^2}}$.

Une telle fonction de renforcement continue est *ajustable* afin d'éviter un comportement nuisible. Somme toute, la meilleure approche serait de pouvoir influencer de manière éphémère l'apprentissage.

4.3 Fonction d'influence

Étant donné l'importance de l'initialisation de la fonction de valeur de l'action, nous nous inspirons dans cette partie des *estimateurs de progrès* afin d'*initialiser la table Q* avec des informations précises. Dans cette section, la fonction de renforcement est binaire (3) avec $r_\infty = 0$ et $r_g = 1$.

Essayons de concevoir une *fonction d'influence dirigée vers l'état cible* grâce à notre analyse précédente. Une influence intéressante doit instaurer *un gradient ajustable sur les valeurs des états*. De plus, le comportement de piétinement doit être évité. Nous suggérons par exemple la *fonction d'influence gaussienne* suivante :

$$Q_i(s,a) = \beta e^{-\frac{d(s,s_g)^2}{2\sigma^2}} + \delta + Q_\infty \ . \quad (8)$$

δ fixe le niveau d'exploration systématique loin de l'état cible, β l'amplitude de la fonction d'influence et σ la zone d'influence.
Concernant le labyrinthe, la fonction d'influence est telle que les états près de l'état cible sont de plus en plus attirants *a priori* que les états loin de l'état cible. Donc δ et β doivent être choisis très petit par rapport à 1 (afin d'éviter trop d'exploration systématique). La figure 4 expose les résultats obtenus avec la fonction

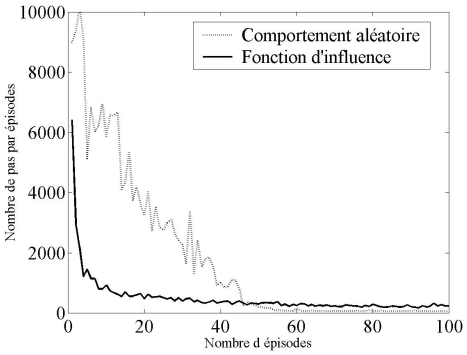

FIG. 4 – Nombre de pas pour atteindre l'état cible en fonction du nombre d'épisodes. Expériences sur le labyrinthe moyennées sur 20 essais indépendants avec une fonction d'influence. Récompenses binaires avec $r_\infty = 0$ et $r_g = 1$. Le comportement aléatoire correspond à $Q_i = 0$. La fonction d'influence est $Q_i(s,a) = 0.001e^{-\frac{d(s,s_g)^2}{2\times 13^2}}$.

d'influence sur le labyrinthe. Ceux-ci sont manifestes. *La fonction d'influence conduit à un processus d'apprentissage plus rapide.* Dès le dixième épisode, le nombre de pas nécessaire pour atteindre l'état cible est divisé par 6 par rapport à un apprentissage classique.

Il est important de remarquer qu'il n'y a avec cette méthode *aucun problème de désapprentissage concernant les impasses, même si notre fonction d'influence est fausse.* Contrairement à la section 4.2, l'effet du la fonction d'influence est éphémère. Elle conseille l'agent *seulement en début d'apprentissage.*

4.4 Conclusion

Les méthodes par estimateurs de progrès et fonctions de potentiel doivent être appliquées avec prudence pour choisir une fonction de renforcement continue. C'est pourquoi nous proposons une fonction de renforcement continue inspirée d'une fonction gaussienne et dont la zone d'influence du gradient est ajustable de manière à pouvoir gérer les risques. *Toutefois, la meilleure solution est d'opter pour une fonction d'influence adéquate qui n'engendre aucun problème. Notre étude aide au choix d'une fonction d'influence correcte.*

5 Expériences sur le problème du pendule inversé

Pour finir, nous validons nos résultats sur le problème en espace continu du *contrôle d'un pendule inversé à couple limité* [2] (Fig. 5). Le contrôle de ce système à un degré de liberté est non-trivial si le couple maximal u^{max} est plus petit que le couple maximal mgl

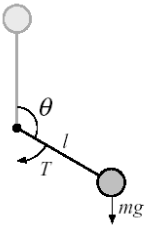

FIG. 5 – Pendule avec un couple limité. La dynamique est donnée par $\dot{\theta} = \omega$ et $ml^2\dot{\omega} = -\mu\omega + mgl\sin\theta + u$. Les paramètres physiques sont $m = l = 1$, $g = 9.8$, $\mu = 0.01$, et $u^{max} = 5.0$. Les paramètres d'apprentissage sont $\gamma = 0.97$, $\alpha = 0.1$.

engendré par le poids. Le contrôleur doit balancer le pendule plusieurs fois pour acquérir un moment permettant de remonter le pendule en position verticale mais doit aussi décélérer le pendule pour éviter qu'il ne retombe.
Nous avons choisi un espace d'états à deux dimensions $\mathbf{x} = (\theta, \omega)$ et une discrétisation de $30 \times 30 \times 9$ a été utilisé pour l'espace d'état-action (θ, ω, u). Chaque épisode débute depuis un état initial $\mathbf{x}(0) = (\pi, 0.1)$ et dure 20 secondes. Le temps d'échantillonnage est de 0.03 secondes. Pour la mesure de performance, nous avons défini t_{up} comme le temps pendant lequel le pendule reste en position haute ($|\theta| < \pi/4$). Un épisode est considéré comme réussi si t_{up} est supérieur à la moyenne des t_{up} sur les 1000 derniers épisodes.

Nous testons la performance de l'algorithme du Q-learning selon la forme de la fonction de renforcement et des valeurs initiales de la table Q (Fig. 6). Dans tous les cas, la moyenne des t_{up} après 1000 épisodes est d'environ 14 secondes.

Dans notre premier essai, la fonction de renforcement est binaire :

$$R(\mathbf{x}, u, \mathbf{x}') = \begin{cases} 1 \text{ si } |\theta'| < \pi/4 \\ 0 \text{ otherwise} \end{cases} \quad (9)$$

et les valeurs de Q_i sont uniformes. Avec $Q_i = 0$, la tâche est réellement difficile à apprendre pour le contrôleur (`bar1`) car le comportement loin du but final est aléatoire. Une meilleure performance avec une fonction de renforcement binaire et des valeurs de Q_i uniformes est observée avec $Q_i > 0$ (`bar2`), c'est-à-dire que le comportement engendré quand $|\theta| > \pi/4$ est l'*exploration systématique.* La stratégie conduit l'agent dans des zones inexplorées de l'environnement qui ont des valeurs attractives. Autrement dit, le contrôleur est incité à remonter le pendule. L'exploration systématique est donc la meilleure stratégie dans ce cas particulier.

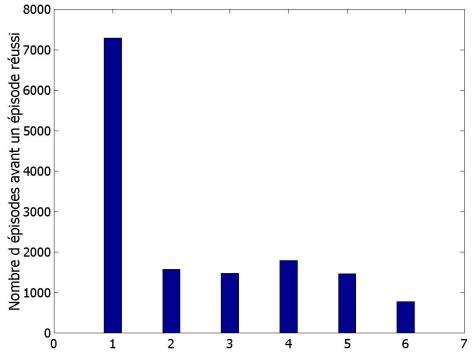

FIG. 6 – Comparaison du nombre d'épisodes effectués avant un épisode réussi. La simulation dure 10000 épisodes moyennés sur 10 essais indépendants.
`bar1` : {fonction de renforcement binaire ; $Q_i(\mathbf{x}) = 0$}
`bar2` : {fonction de renforcement binaire ; $Q_i(\mathbf{x}) = 0.1$}
`bar3` : {fonction de renforcement binaire ; $Q_i(\mathbf{x}) = e^{-\frac{\theta^2}{2\times 0.25^2}} + 0.1$ } `bar4` : $\{R(\mathbf{x}, u, \mathbf{x}') = \cos(\theta')$; $Q_i(\mathbf{x}) = 0\}$
`bar5` : $\{R(\mathbf{x}, u, \mathbf{x}') = e^{-\frac{\theta'^2}{2\times 0.25^2}}$; $Q_i(\mathbf{x}) = 10$ }
`bar6` : $\{R(\mathbf{x}, u, \mathbf{x}') = e^{-\frac{\theta'^2}{2\times 0.25^2}}$;
$Q_i(\mathbf{x}) = 11\ e^{-\frac{\theta^2}{2\times 0.25^2}} + 0.1\}$

Nous testons maintenant la fonction d'influence avec des récompenses binaires : $Q_i(\mathbf{x}) = \beta e^{-\frac{\theta^2}{2\sigma^2}} + \delta$. D'après nos résultats précédents, il est évident que la fonction d'influence doit favoriser l'exploration systématique quand $|\theta| > \pi/4$, donc nous posons $\delta = 0.1$, $\beta = 1$ et $\sigma = 0.25$ (`bar3`). La fonction d'influence n'améliore pas de manière significative l'apprentissage.

La récompense classique pour ce problème de contrôle sur un espace d'états continu est fonction de la hauteur de l'extrémité du pendule [2] , c'est-à-dire $R(\mathbf{x}, u, \mathbf{x}') = \cos(\theta')$. La valeur arbitraire de Q_i est généralement 0 (`bar4`). Ainsi, en début d'apprentissage, le pendule piétine quand $|\theta| < \pi/2$ et explore systématiquement quand $|\theta| > \pi/2$. Le résultat est décevant.

Nous appliquons maintenant la fonction de renforcement gaussienne (7) avec $Q_i = 10$ et $\beta = 1$. La distance est définie comme $d(\mathbf{x'}, \mathbf{x}_{up}) = \theta'$ avec $\mathbf{x'}$ le nouvel état et $\mathbf{x}_{up}$ l'état cible. La fonction de renforcement continue est

$$R(\mathbf{x}, u, \mathbf{x}') = e^{-\frac{\theta'^2}{2\sigma^2}} \ . \qquad (10)$$

$\sigma = 0.25$ de sorte que la zone d'influence du gradient de récompenses soit seulement active autour de $|\theta| < \pi/4$. Les résultats (`bar5`) sont très proches du cas de récompenses binaires avec exploration systématique (`bar2`).

Pour finir, nous avons essayé la fonction d'influence avec une fonction de renforcement continue. La fonction de renforcement est continue, il en est de même pour Q_i qui doit être supérieur à $\frac{R(\mathbf{x})}{1-\gamma}$. Donc la fonction d'influence est $Q_i(\mathbf{x}) = \beta(1+\frac{1}{1-\gamma})e^{-\frac{d^2}{2\sigma^2}} + \delta$. Nous avons conservé nos choix précédents : $\delta = 0.1$, $\beta = 1$ et $\sigma = 0.25$. Cette dernière simulation est la meilleure performance obtenue concernant le pendule inversé.

6 Conclusion

Ainsi, le choix de la fonction de renforcement et des valeurs initiales de la table Q a un impact majeur sur la performance des algorithmes d'AR. Le choix de ces paramètres doit donc être judicieux.

Notre étude a abouti à l'élaboration de *règles pour évaluer correctement les récompenses et valeurs initiales de Q* selon le comportement désiré. Notamment, certaines valeurs de Q_i peuvent amener l'agent à avoir un comportement néfaste qui doit être évité. Grâce à nos expériences, nous avons confirmé *la présence de limites qui démarquent différents comportements.* Il est important de noter que plus Q_i s'éloigne de ces bornes, plus le comportement caractéristique est marqué.

De plus, nous conseillons de rester prudent vis-à-vis des méthodes par estimateurs de progrès et fonctions de potentiel, qui peuvent entraîner des comportements néfastes. Une fonction de renforcement continue ajustable et moins risquée est ainsi suggérée. Enfin, avec l'aide de nos conditions sur les valeurs initiales de Q, nous avons développé une *fonction d'influence générique*, dont la principale caractéristique est d'être éphémère. Cette méthode se révèle être une manière efficace *d'améliorer les performances du processus d'apprentissage de problèmes de plus court chemin stochastique.* La table 1 récapitule les choix que nous préconisons concernant la fonction de renforcement et l'initialisation de la fonction de valeur de l'action pour des tâches de type *goal-directed.*

Références

[1] S. Behnke and M. Bennewitz. Learning to play soccer using imitative reinforcement. In *Proc. of the ICRA Workshop on Social Aspects of Robot Programming through Demonstration*, Barcelona, April 2005.

[2] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1) :219–245, 2000.

[3] P. Garcia. *Exploration guidée et induction de comportements génériques en apprentissage par*

TAB. 1 – Choix préconisés pour la fonction de renforcement et les valeurs initiales de Q dans le cadre de problèmes de plus court chemin stochastique.

FONCTION DE RENFORCEMENT BINAIRE POUR UN ESPACE D'ÉTATS DISCRET	FONCTION DE RENFORCEMENT CONTINUE POUR UN ESPACE D'ÉTATS CONTINU
$r(s,a,s') = \begin{cases} r_g \text{ si } s' = s_g \\ r_\infty \text{ else} \end{cases}$ Choix de r_g et r_∞ : $r_g \geq \frac{r_\infty}{1-\gamma}$	$r(s,a,s') = \beta e^{-\frac{d(s',s_g)^2}{2\sigma^2}}$
Choix des valeurs initiales uniformes de Q : $Q_i = \frac{r_\infty}{1-\gamma} + \delta$	Choix des valeurs initiales de Q : $Q_i = \frac{\beta}{1-\gamma}$
Choix des valeurs initiales influencées de Q : $Q_i(s) = \beta e^{-\frac{d(s,s_g)^2}{2\sigma^2}} + \delta + \frac{r_\infty}{1-\gamma}$	Choix des valeurs initiales influencées de Q : $Q_i(s) = \beta(1 + \frac{1}{1-\gamma}) e^{-\frac{d(s,s_g)^2}{2\sigma^2}} + \delta$
$\delta \geq 0$ fixe le niveau d'exploration systématique loin de l'état cible et $\beta > 0$ ajuste l'amplitude du gradient σ fixe la zone d'influence du gradient et γ est le coefficient d'actualisation, s est l'ancien état, s' le nouveau, s_g l'état cible	

renforcement. PhD thesis, INSA de Rennes, July 2004.

[4] G. Hailu and G. Sommer. On amount and quality of bias in reinforcement learning. In *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1491–1495, Tokyo, Oct. 1999.

[5] S. Koenig and R. G. Simmons. The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *Machine Learning*, 22(1-3) :227–250, 1996.

[6] G.J. Laurent and E. Piat. Learning mixed behaviours with parallel q-learning. In *Proc. of IROS*, pages 1002–1007, Lausanne, Oct. 2002.

[7] M. J. Mataric. Reward functions for accelerated learning. In *Proc. of the 11th ICML*, pages 181–189, 1994.

[8] A.Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations : theory and application to reward shaping. In *Proc. of the 16th ICML*, pages 278–287, 1999.

[9] J. Randlov and P. Alstrom. Learning to drive a bicycle using reinforcement learning and shaping. In *Proc. of the 16th ICML*, pages 463–471, 1998.

[10] R. S. Sutton and A. G. Barto. *Reinforcement Learning : An Introduction*. The MIT Press, Cambridge, 1998.

[11] C.J.C.H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England, 1989.

[12] C.J.C.H. Watkins and P. Dayan. Technical note : Q-learning. *Machine Learning*, 8 :279–292, 1992.

[13] E. Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19 :205–208, 2003.