

pas celle de l'autre action. L signifie « linear », R signifie « reward », P « penalty », I « inaction ».

### Incrémentalité

Les méthodes de valeurs d'action définies avant estiment des valeurs d'action avec la technique de l'échantillonnage moyen. Pour cela on peut utiliser la formule 1, mais on peut aussi utiliser une formule incrémentale :

$$Q_{k+1} = Q_k + (r_{k+1} - Q_k)/(k+1) \quad (5)$$

Les formules 4 et 5 sont caractéristiques de la mise à jour utilisée en AR. En général la mise à jour en AR utilise une formule de la forme suivante :

$$NelleEstim = AncEstim + Pas \times (Cible - AncEstim) \quad (6)$$

$(Cible - AncEstim)$  correspond à une erreur de l'estimation. Pas est le pas de l'apprentissage. Dans la formule 5, il valait  $1/k$ . On le note souvent  $\alpha$ .

## Le problème de l'apprentissage par renforcement

Cette partie définit le problème de l'AR, cœur de ce chapitre. Il le définit dans un sens très large.

### L'interface Agent-Environnement

Le problème de l'AR est censé être une caricature directe du problème de l'apprentissage d'un agent à atteindre des buts à partir de l'interaction. L'apprenant et le décideur constituent une même entité appelée *l'agent*. La chose avec laquelle l'agent interagit, tout ce qui est extérieur à l'agent, s'appelle *l'environnement*. L'agent et l'environnement interagissent continuellement, l'agent sélectionnant des actions et l'environnement répondant à ces actions et présentant de nouvelles situations à l'agent. L'environnement donne des *récompenses* à l'agent, sous forme numérique, que l'agent essaie de maximiser au fur et à mesure que le temps passe.

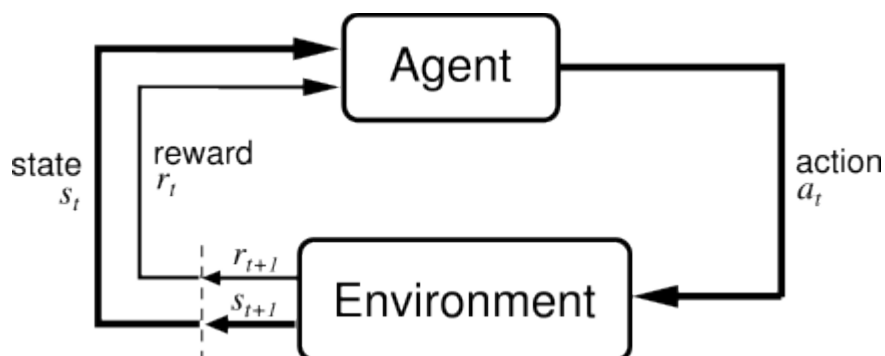


Figure 4 : L'interaction agent-environnement.

L'agent et l'environnement interagissent à des temps discrets  $t = 0, 1, 2, 3, \dots$ . A chaque pas de temps  $t$ , l'agent reçoit une représentation de l'état de l'environnement, appelé *état* et noté  $s_t \in S$ .

S est l'ensemble de tous les états possibles. Sur cette base, l'agent sélectionne une *action*, notée  $a_t \in A(s_t)$ .  $A(s_t)$  est l'ensemble de toutes les actions possibles à effectuer dans l'état  $s_t$ . Un pas de temps plus tard, l'agent reçoit un *retour*  $r_{t+1} \in R$  et se retrouve dans un état  $s_{t+1}$ . La figure 4 montre cela sous forme de diagramme. A chaque pas de temps, l'agent fait une correspondance entre les états et les probabilités de choisir chaque action possible. Cette correspondance s'appelle la *politique* de l'agent et elle est notée  $\pi_t$  où  $\pi_t(s, a)$  est la probabilité de choisir l'action  $a$  quand l'agent est dans l'état  $s$ . Les méthodes d'AR spécifient comment l'agent change de politique en fonction de son expérience. Le but de l'agent est de maximiser ses retours au fur et à mesure que le temps passe. Ce cadre est souple et flexible et peut être appliqué à différents problèmes de différentes manières.

## Buts et récompenses

L'usage d'une récompense pour formaliser l'idée de but est l'une des principales caractéristiques de l'AR. Par exemple, pour un robot qui doit sortir d'un labyrinthe, on peut pénaliser le robot avec  $-1$  à chaque pas de temps tant qu'il est dans le labyrinthe et lui donner  $+100$  lorsqu'il est sorti du labyrinthe. La récompense n'est pas donnée pour indiquer à l'agent comment il doit atteindre son but, mais seulement pour indiquer que le but est atteint. La récompense telle que définie ici est considérée comme donnée par l'environnement et extérieure à l'agent. En revanche, cela n'empêche pas l'agent d'avoir sa propre façon de se récompenser de manière interne, et c'est d'ailleurs sur ce principe que se basent les méthodes d'AR.

## Retours

Que signifie maximiser les retours au fur et à mesure du temps ? Si la séquence de retours reçus à partir du pas de temps  $t$  est notée,  $r_{t+1}, r_{t+2}, r_{t+3}, \dots$ , l'agent cherche à maximiser la somme des retours  $R$  :

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots + r_T \quad (7)$$

où  $T$  est le pas de temps final. Ce qui n'a de sens que si l'interaction agent-environnement se découpe en sous-séquences ou *épisodes*. Chaque épisode se termine par un *état final*. A la fin d'un épisode, l'agent repart sur un état de départ, etc. Dans ce cas, la tâche de l'agent est appelée une tâche épisodique.

A l'opposé, la tâche de l'agent peut être *continue*. Dans ce cas, la formule 7 est problématique. En introduisant le concept de diminution, elle est remplacée par :

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{k=\infty} \gamma^k r_{t+k+1} \quad (8)$$

$\gamma$  est un paramètre tel que  $0 \leq \gamma \leq 1$  appelé le taux de diminution<sup>4</sup>. Ainsi un retour dans  $k$  pas de temps dans le futur vaut  $\gamma^{k-1}$  fois ce qu'il aurait valu s'il était reçu tout de suite.

<sup>4</sup> « discount rate » en anglais

## Propriété de Markov

Dans le cas le plus général, quand l'agent est dans l'état  $s = s_t$  au pas de temps  $t$ , la probabilité d'être dans l'état  $s' = s_{t+1}$ , en recevant le retour  $r = r_{t+1}$ , dépend de tout ce qui s'est passé avant, ce que l'on écrit :

$$\Pr (s_{t+1}=s', r_{t+1}=r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0) \quad (9)$$

Si la réponse de l'environnement à l'instant  $t+1$  ne dépend que de l'état et l'action prise à l'instant  $t$ , alors on dit que l'on a la *propriété de Markov* et l'on écrit :

$$\Pr (s_{t+1}=s', r_{t+1}=r \mid s_t, a_t) \quad (10)$$

La propriété de Markov est importante en AR car les décisions et valeurs ne dépendent que de l'état courant et pas du passé. Ce qui entraîne que la représentation de l'état doit être informative. Cette propriété n'est pas caractéristique de l'AR, elle est utilisée aussi dans d'autres domaines de l'IA.

## Processus Décisionnels de Markov (MDP)

Un système d'AR qui vérifie la propriété de Markov est appelé un Processus Décisionnel de Markov (PDM). Si le nombre d'états est fini, il s'agit d'un « PDM fini ». Un PDM fini est défini par les probabilités de transitions vers un état  $s'$  sachant que l'état est  $s$  et l'action choisie  $a$  :

$$P_{ss'}^a = \Pr (s_{t+1} = s' \mid s_t = s, a_t = a) \quad (11)$$

Et par l'espérance de gain sachant que l'état précédent est  $s$ , l'action choisie  $a$  et l'état nouveau  $s'$ :

$$R_{ss'}^a = E (r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s') \quad (12)$$

Les quantités  $P_{ss'}^a$  et  $R_{ss'}^a$  spécifie complètement le MDP fini. La suite du document s'intéresse aux MDP finis.

## EXEMPLE DU ROBOT RECYCLANT

A FAIRE (pages 66-68)

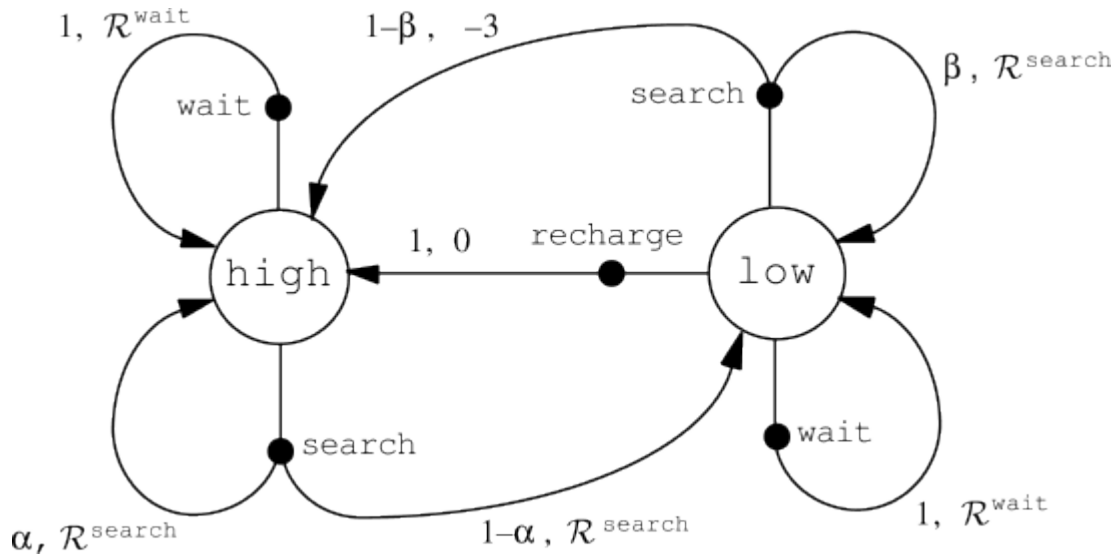


Figure 5 : Le graphe de transitions pour le robot recyclant.

## Fonctions de valeur

La plupart des algorithmes d'AR sont basés sur l'estimation de *fonction de valeur* (valeur d'état ou valeur de couple état-action). Ces fonctions estiment de combien il est bon d'être dans un état ou de combien il est bon d'effectuer une action dans un état. La notion « de combien » est définie par les récompenses futures attendues par l'agent, autrement dit par le retour attendu par l'agent qui utilise sa politique  $\pi$ .  $\pi$  est une application qui relie un couple état  $s$  et action  $a$  à une probabilité d'effectuer cette action  $a$  dans l'état  $s$  ( $\pi(s, a)$ ). La valeur d'un état  $s$  suivant une politique  $\pi$  est notée  $V^\pi(s)$ , est le retour espéré quand on part de  $s$  et que l'on suit la politique  $\pi$ , ce qui s'écrit :

$$V^\pi(s) = E_\pi \{ R_t \mid s = s_t \} = E_\pi \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s = s_t \} \quad (13)$$

$V^\pi$  s'appelle *la fonction de valeur d'état* pour la politique  $\pi$ . De manière similaire, on peut définir  $Q^\pi(s, a)$ , la valeur d'une action  $a$  quand on est dans l'état  $s$  et quand on suit la politique  $\pi$ , comme étant le retour espéré si on effectue  $a$  à partir de  $s$  en suivant ensuite  $\pi$ . Ce que l'on écrit de la manière suivante :

$$Q^\pi(s, a) = E_\pi \{ R_t \mid s = s_t, a = a_t \} = E_\pi \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s = s_t, a = a_t \} \quad (14)$$

$Q^\pi$  s'appelle *la fonction de valeur d'action* pour la politique  $\pi$ . Les fonctions  $V^\pi$  et  $Q^\pi$  peuvent être estimées par l'expérience. Par exemple, si l'agent maintient une moyenne, pour chaque état rencontré  $s$ , des retours effectifs qui ont suivi cet état, alors la moyenne converge vers la valeur de l'état  $V^\pi(s)$ . Si de manière similaire il maintient une moyenne, pour chaque action  $a$  effectuée dans un état rencontré  $s$ , des retours effectifs qui ont suivi, alors cette moyenne converge vers  $Q^\pi(s, a)$ . Ce type d'estimation s'appelle des *estimations Monte Carlo* car elle calcule des moyennes sur des échantillons aléatoires de retours effectifs.

La particularité des fonctions de valeurs est qu'elle satisfait à des équations récursives. On peut d'abord écrire.

$$\begin{aligned} V^\pi(s) &= E_\pi \{ \sum_{k=0}^{k=\infty} \gamma^k r_{t+k+1} \mid s = s_t \} \\ &= E_\pi \{ r_{t+1} + \gamma \sum_{k=0}^{k=\infty} \gamma^k r_{t+k+2} \mid s = s_t \} \end{aligned}$$

Ensuite en développant sur toutes les actions a possibles à partir de s, et sur tous les états suivants s', on a :

$$\begin{aligned} V^\pi(s) &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma E_\pi \{ \sum_{k=0}^{k=\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \} ] \\ &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')] \end{aligned} \quad (15)$$

La formule 15 s'appelle l'équation de Bellman de  $V^\pi$ .

On peut faire pareil pour  $Q^\pi$  :

$$\begin{aligned} Q^\pi(s, a) &= E_\pi \{ \sum_{k=0}^{k=\infty} \gamma^k r_{t+k+1} \mid s = s_t, a = a_t \} \\ &= E_\pi \{ r_{t+1} + \gamma \sum_{k=0}^{k=\infty} \gamma^k r_{t+k+2} \mid s = s_t, a = a_t \} \\ &= \sum_{s'} P_{ss'}^a \sum_{a'} \pi(s', a') [R_{ss'}^a + \gamma Q^\pi(s', a')] \end{aligned} \quad (16)$$

La formule 16 s'appelle l'équation de Bellman de  $Q^\pi$ . On peut représenter ces équations sous forme de diagrammes de « back-up » comme le montre la figure 6.

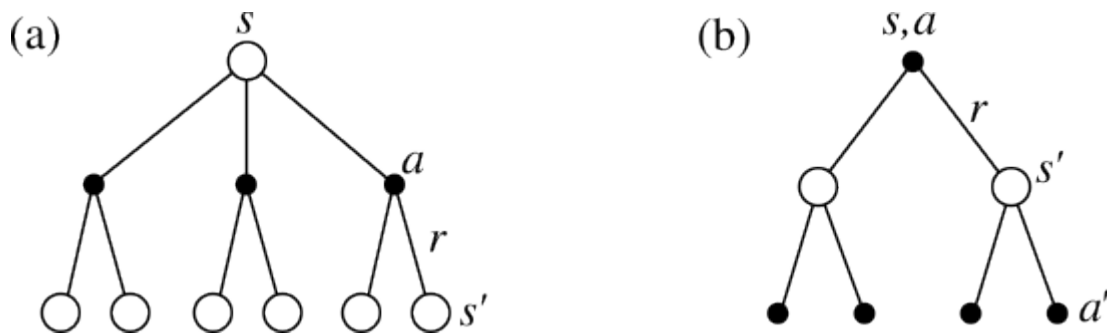


Figure 6 : Les diagrammes de back-up des équations de Bellman pour (a)  $V^\pi$  et (b)  $Q^\pi$ .

### Exemple de la grille

Pour illustrer les MDP, l'exemple de la grille de la figure 7a est adapté. Un agent se déplace sur une grille. L'état de l'agent correspond à une case de la grille. A chaque pas de temps, l'agent effectue une des 4 actions suivantes: nord, est, sud ou ouest. L'environnement est déterministe au sens où l'agent se retrouve sur la case correspondant à l'action qu'il a choisie. Si l'agent est situé sur une case du bord et qu'il choisit une action le faisant sortir de la grille, alors il reste sur la même case et reçoit une pénalité -1. S'il est sur la case A (respectivement B), quelle que soit son action, il va sur la case A' (respectivement B') et reçoit une récompense +10 (respectivement +5). Dans les autres cas, il va sur la case correspondant à son choix et ne reçoit aucune récompense. Dans le cas simple considéré ici, la politique suivie par l'agent est aléatoire complètement: il sélectionne de manière équiprobable une action parmi les 4 possibles. La

figure 7b montre la valeur de  $V$ , pour cette politique aléatoire, solution des équations de Bellman 15. Sur cet exemple 15 est un système de 25 équations linéaires, donc résolubles en pratique.

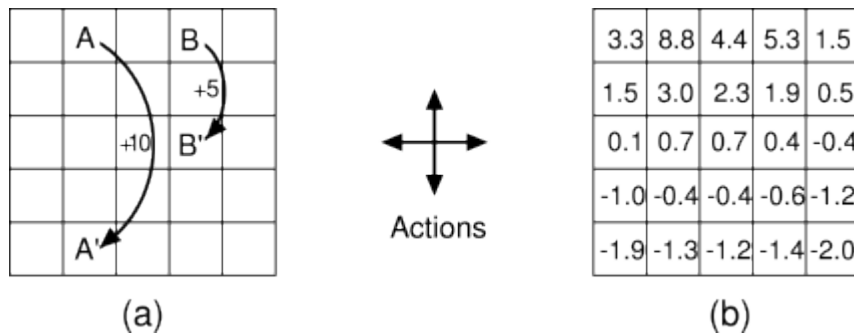


Figure 7 : L'exemple de la grille : (a) dynamique des récompenses exceptionnelles ; (b) fonction de valeur d'état pour la politique aléatoire équiprobable, solution des équations de Bellman.

### Exercice a

La valeur d'un état dépend des valeurs des actions possibles dans cet état et de l'urgence de ces actions suivant la politique de l'agent. On peut penser cela sous la forme d'un diagramme de back-up :

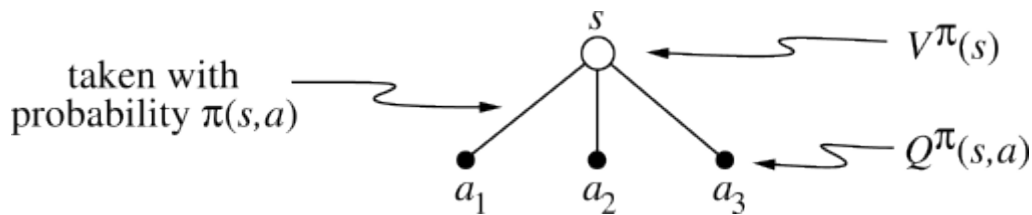


Figure 8

Donnez l'équation correspondante pour la valeur du nœud racine  $V^{\pi}(s)$  en fonction des termes des nœuds feuilles,  $Q^{\pi}(s, a)$ , sachant que  $s_t = s$ . Cette espérance dépend de la politique  $\pi$ . Puis donnez un autre équation dans laquelle l'espérance est explicitement écrite en fonction de  $\pi(s, a)$  de façon à ce que qu'aucune espérance n'apparaisse dans la formule.

### Exercice b

La valeur d'une action  $Q^{\pi}(s, a)$  peut être divisée en deux parties : la récompense prochaine, ne dépendant pas de la politique  $\pi$ , et la somme attendue des récompenses suivantes qui dépendent du prochain état et de la politique  $\pi$ . A nouveau si on considère le diagramme de back-up suivant :

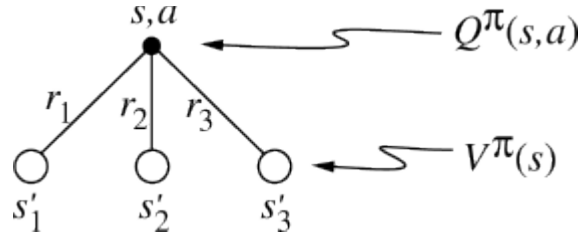


Figure 9

Donnez l'équation correspondante pour la valeur d'action  $Q^\pi(s, a)$  en fonction de la récompense suivante  $r_{t+1}$ , et la valeur de l'état suivant  $V^\pi(s_{t+1})$ , sachant que  $s_t = s$  et  $a_t = a$ . puis donnez une seconde équation donnant la valeur attendue explicitement en fonction de  $P^{a_{ss'}}$  et  $R^{a_{ss'}}$ , définis par les formules 11 et 12, de manière à ce qu'aucune espérance n'apparaisse dans la formule.

### Fonctions de valeur optimales

Résoudre un problème d'AR consiste à trouver une politique qui donne un maximum de récompenses à l'agent. Les fonction valeurs  $V^\pi$  définissent un ordre partiel sur les politiques  $\pi$ . Une politique  $\pi$  est supérieure à une politique  $\pi'$  si  $V^{\pi'}(s) \leq V^\pi(s)$  pour les états possibles. On note cela  $\pi' \leq \pi$ . Il existe au moins une politique qui est meilleure que toutes les autres, appelées politiques optimales et notée  $\pi^*$ . Ces politiques partagent la même fonction valeur d'état, notée  $V^*$ , appelée *fonction de valeur d'état optimale*. On a :

$$V^*(s) = \max_{\pi} V^\pi(s) \quad (17)$$

Les politiques optimales partagent aussi une même *fonction de valeur d'action optimale*, notée  $Q^*$  et définie par :

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) \quad (18)$$

On a l'équation suivante :

$$Q^*(s, a) = E \{ r_{t+1} + \gamma V^*(s_{t+1}) \mid s = s_t, a = a_t \} \quad (19)$$

Parce que  $V^*$  est la fonction de valeur pour une politique  $\pi^*$ , elle doit satisfaire la condition de l'équation de Bellman pour les valeurs d'état (formule 15) :

$$\begin{aligned} V^*(s) &= \max_a Q^{\pi^*}(s, a) \\ &= \max_a E_{\pi^*} \{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s = s_t, a = a_t \} \\ &= \max_a E_{\pi^*} \{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s = s_t, a = a_t \} \\ &= \max_a E_{\pi^*} \{ r_{t+1} + \gamma V^*(s_{t+1}) \mid s = s_t, a = a_t \} \end{aligned} \quad (20)$$

$$= \max_a \sum_{s'} P^{a_{ss'}} [R^{a_{ss'}} + \gamma V^*(s')] \quad (21)$$

Les équations 20 et 21 sont les deux formes de l'équation d'optimalité de Bellman pour  $V^*$ . On peut écrire de manière similaire l'équation d'optimalité de Bellman pour  $Q^*$  :

$$\begin{aligned} Q^*(s, a) &= E_{\pi^*} \{ r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s = s_t, a = a_t \} \\ &= \sum_{s'} P^{a_{ss'}} [R^{a_{ss'}} + \gamma \max_{a'} Q^*(s', a')] \end{aligned} \quad (22)$$

Enfin on peut dessiner les diagrammes de back-up pour  $V^*$  et  $Q^*$ .

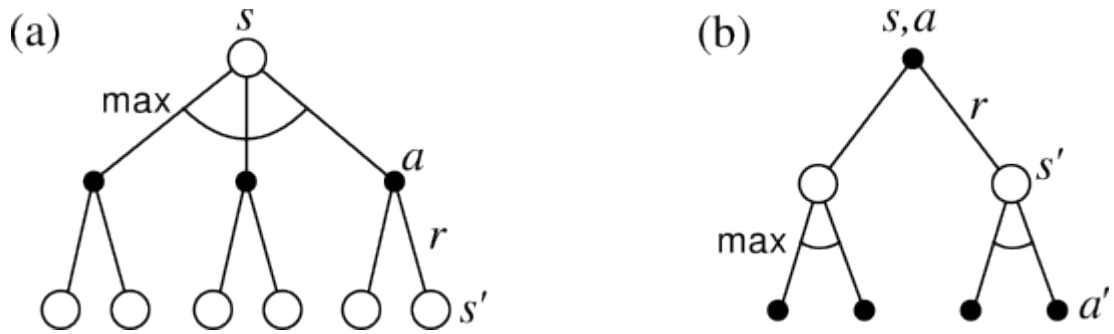


Figure 10 : Les diagrammes de back-up des équations de Bellman pour (a)  $V^*$  et (b)  $Q^*$ .

### Exemple de la grille (suite)

Si l'on résout l'équation d'optimalité de Bellman pour  $V^*$  pour l'exemple de la grille, la figure 11 donne la fonction de valeur d'état optimale et les politiques optimales correspondantes.

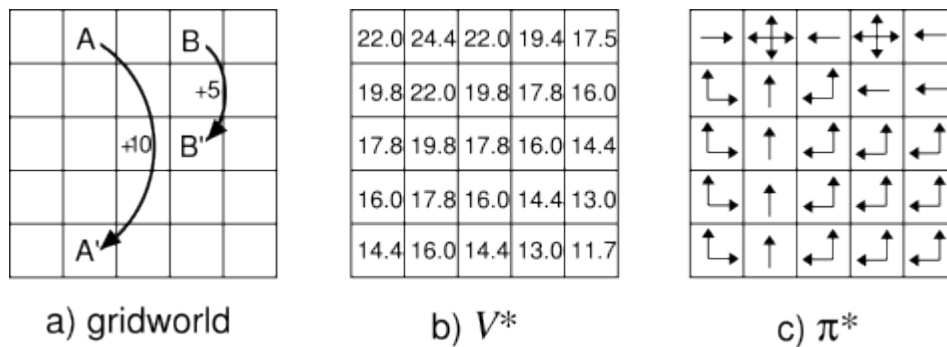


Figure 11 : les solutions optimales pour l'exemple de la grille.

Résoudre explicitement les équations de Bellman est un moyen de trouver une politique optimale, et donc de résoudre la problème de l'AR. Cependant, cette solution est rarement directement utile. Elle dépend d'une recherche exhaustive, regardant toutes les possibilités. Pour les tâches qui nous intéressent, ce n'est pas possible de faire comme cela. Même pour le jeu de Backgammon, qui n'a que  $10^{20}$  états, cela demanderait des millions d'années à nos ordinateurs les plus rapides pour résoudre les équations de Bellman pour  $V^*$ . Beaucoup de méthodes d'AR peuvent être comprises comme des résolutions approximatives de l'équation d'optimalité de Bellman, utilisant des transitions correspondant à l'expérience, plutôt que des transitions attendues. La deuxième partie du document présente ces méthodes « approximatives ».