Apprentissage par renforcement Approche: Programmation Dynamique

Tarik Al ani
A²SI-ESIEE-PARIS

Plan

- I. Introduction à l’apprentissage par renforcement.
- II. Les exemples.

Chapitre 1 : Introduction à l’Apprentissage par Renforcement

(en anglais Reinforcement Learning: RL)

I- Définition de l'apprentissage par renforcement

II- Éléments de l'apprentissage par renforcement

III- Problème de l'apprentissage par renforcement

IV- Programmation dynamique

Définition de l'apprentissage par renforcement

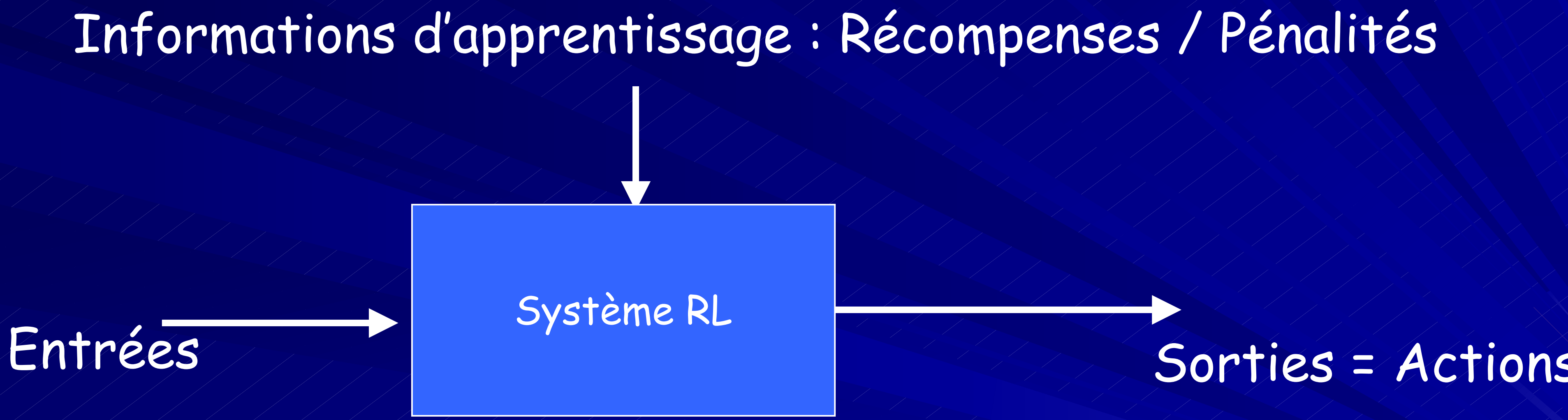
approche programmable qui permet:

- par interaction avec l'environnement

- d'atteindre un but à long terme

- en associant aux situations des actions

Définition de l'apprentissage par renforcement



Eléments de l'apprentissage par renforcement

- Politique P
- Récompense r / Pénalité –r
- Valeur V
- Modèle M de l'environnement

problème de l'apprentissage par renforcement

Interface agent - environnement

problème de l'apprentissage par renforcement

- la politique au pas t : $\pi_t(s,a) = Pr(a_t = a / s_t = s)$
- le revenu R_t
- la valeur d'un état $V^\pi(s)$
- La valeur du choix d'une action a dans un état s sous une politique $Q^\pi(s,a)$

Propriété de Markov

- Une tâche de RL a la propriété de Markov si

- Processus de décision de Markov (MDP)

-Résolution de problèmes de décision

-Programmation d'algorithmes calculant des politiques optimales

- Utilisation des équations d'optimalité de Bellman

équation de Bellman

• Résolution de l'équation d'optimalité de Bellman :

Nécessite :

- une connaissance précise de l'environnement
- une prévision de l'espace et du temps de calcul
- de respecter la propriété de Markov

D'où :

- Des approximations

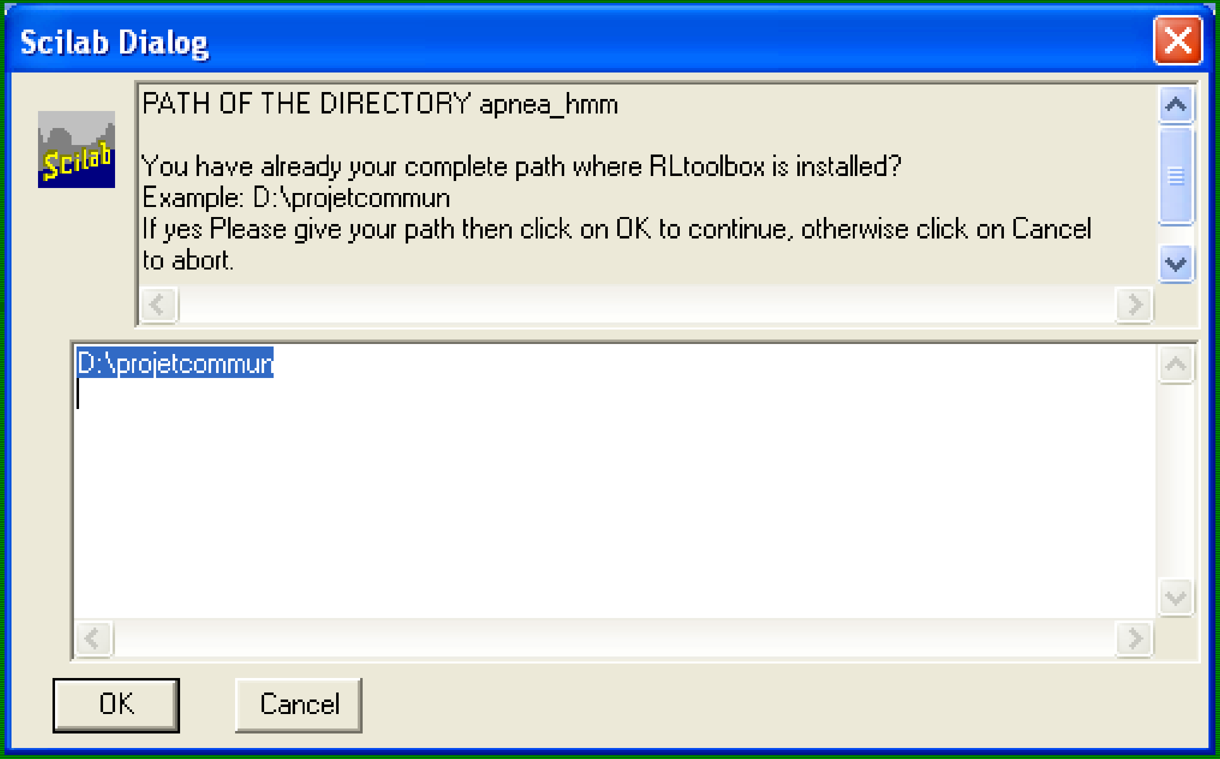
Programmation dynamique

- Les algorithmes de la programmation dynamique partitionnent le problème en sous-problèmes indépendants

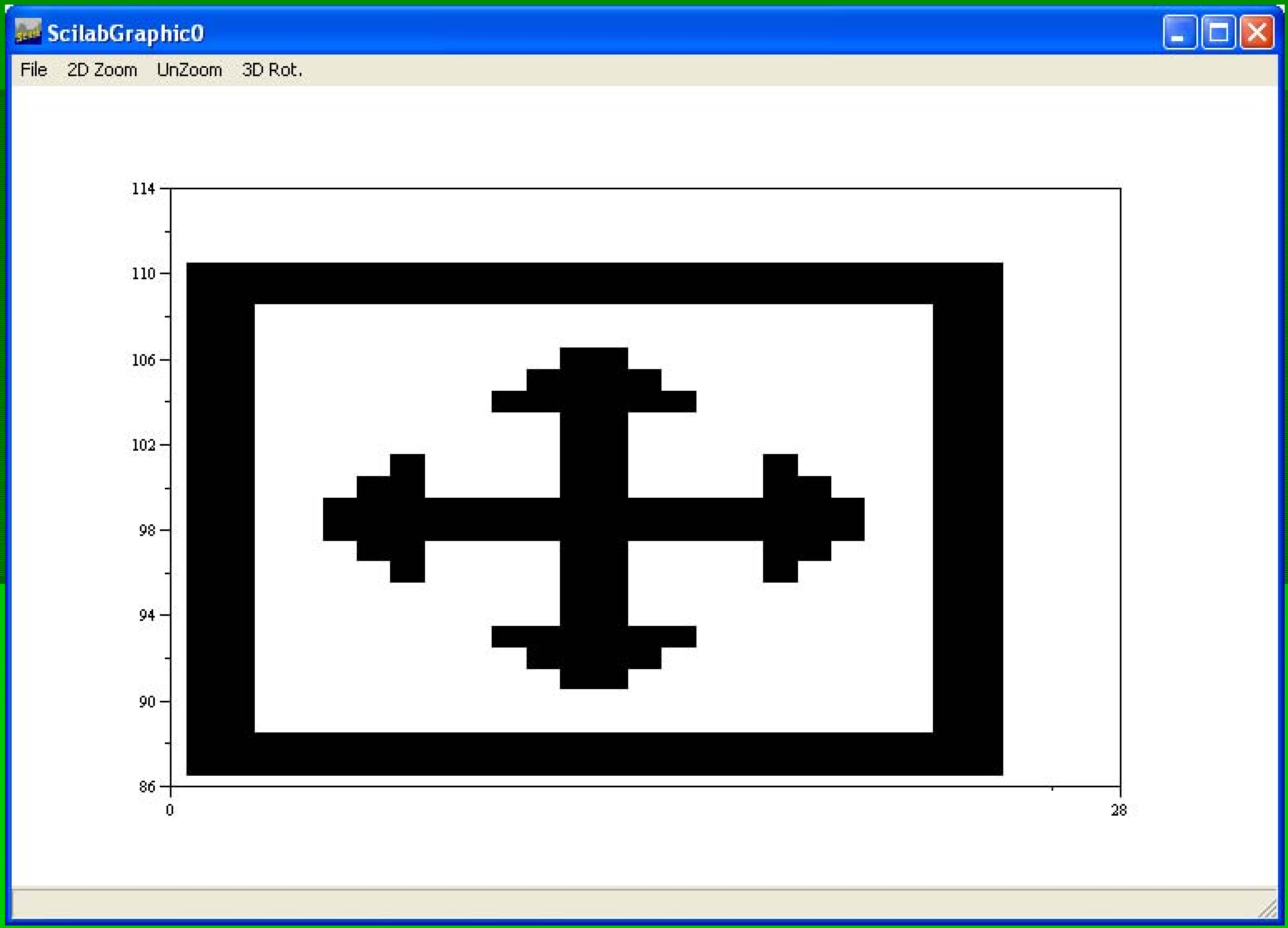
- quatre étapes :
  - ✓ Caractériser la structure de la solution optimale
  - ✓ Définir récursivement ou itérativement
  - ✓ Calculer la valeur d'une solution optimale
  - ✓ Construire une solution optimale

Chapitre 2 :
<u>les exemples</u>

Les fenêtres de messages



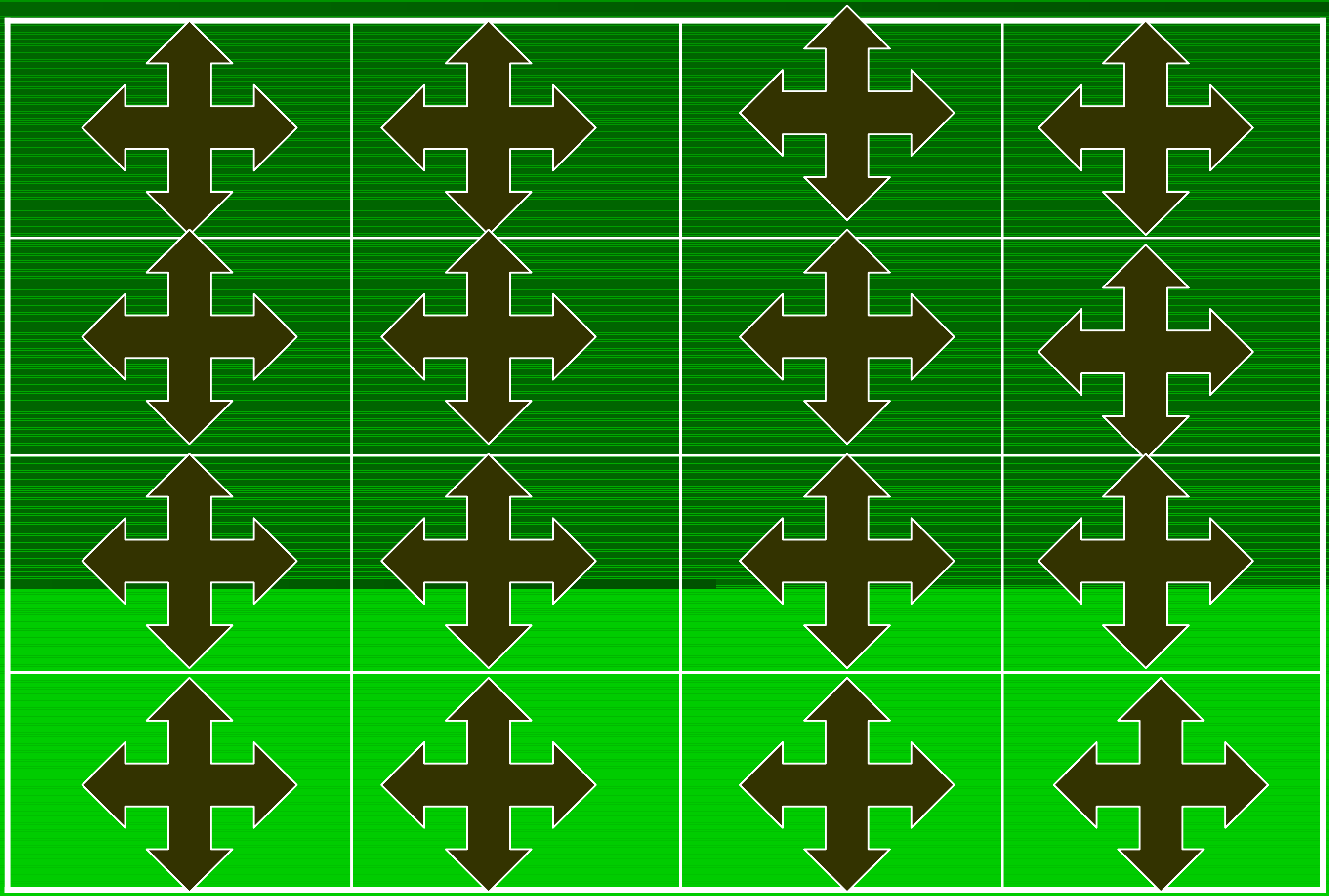
Les fenêtres graphiques



GRILLE

16	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Mouvements initiaux du ROBOT



LA MATRICE TRANSPROB

16	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	0	0
0	0	1	0
0	0	0	0
0	0	0	0

Transprob(3)(5,6)=1
Transprob(3)(6,7)=1
Transprob(3)(5,5)=0
Transprob(3)(6,5)=0

LA MATRICE REWARDS

16	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	-1	0	0
0	0	-1	0
0	0	0	0
0	0	0	0

Rewards(3)(5,6)=-1
Rewards(3)(6,7)=-1
Rewards(3)(14,15)=+1
Rewards(3)(6,5)=0

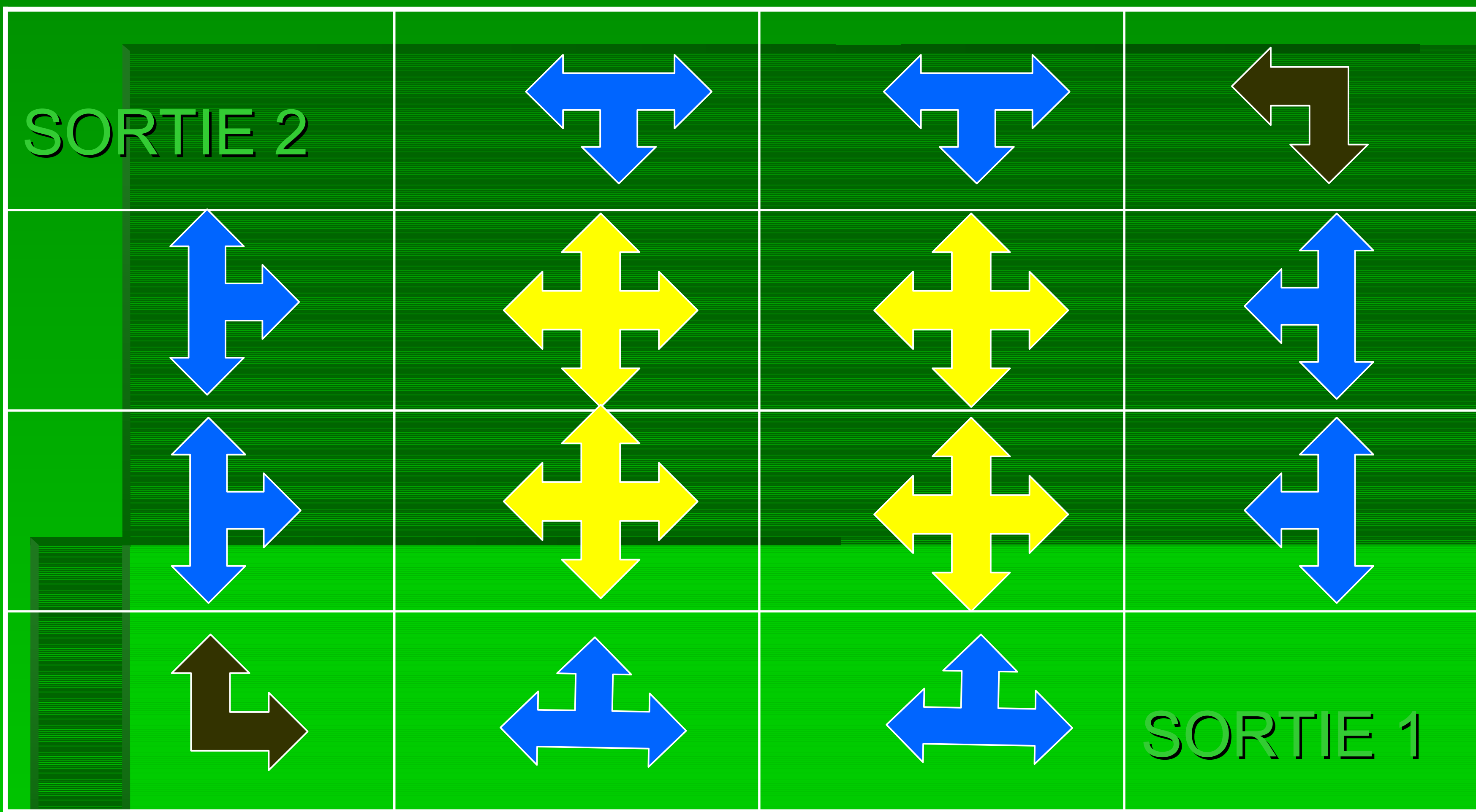
La matrice Actions_states

16	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

F	T	F	F
T	F	T	F
F	T	F	F
F	F	F	F

Action_states(5,6)=T
Action_states(6,7)=T
Action_states(5,5)=F
Action_states (8,5)=F

Mouvements autorisés du ROBOT



Mouvements optimaux du ROBOT

SORTIE 2	←	←	←
↑	←	←	↓
↑	→	→	↓
→	→	→	SORTIE 1

Exemple 2 :
the gambler

- s=states $\in \{0,1,2,\ldots,20\}$, il y a 21 états

- a=actions $\in \{0,1,\ldots,\min(s, 20-s)\}$, il y a au plus 11 actions

- récompense = 0 pour tout couples (états, actions) sauf si le joueur atteint son but, la récompense sera alors 1

- p = 0,4

il y a 11 transprobas (car 11 actions au maximum)

transproba(1) = l'action : le joueur mise 0 $

transproba(2) = l'action : le joueur mise 1 $

transproba(k) = l'action : le joueur mise k-1 $

Transproba(2)

	0	1	2	3	4	…	…	19	20
0	0	0	0	…	…	…	…	…	0
1	**.6**	0	**.4**	0					0
2	0	.	.	.	.				0
3		.	.	.	.	.			.
4			.	.	.	.	.		.
.				.	.	.	.	.	.
.					.	.	.	.	0
19	0	…	…	…	…	0	**.6**	0	**.4**
20	0	…	…	…	…	…	0	0	0

rewards(2)

	0	1	2	3	4	…	…	19	20
0	0	…	…	…	…	…	…	…	0
1	.								.
2	.								.
3	.								.
4	.								.
.	.								.
18	0	…	…	…	…	…	…	…	0
19	0	…	…	…	…	…	…	0	1
20	0	…	…	…	…	…	0	0	0

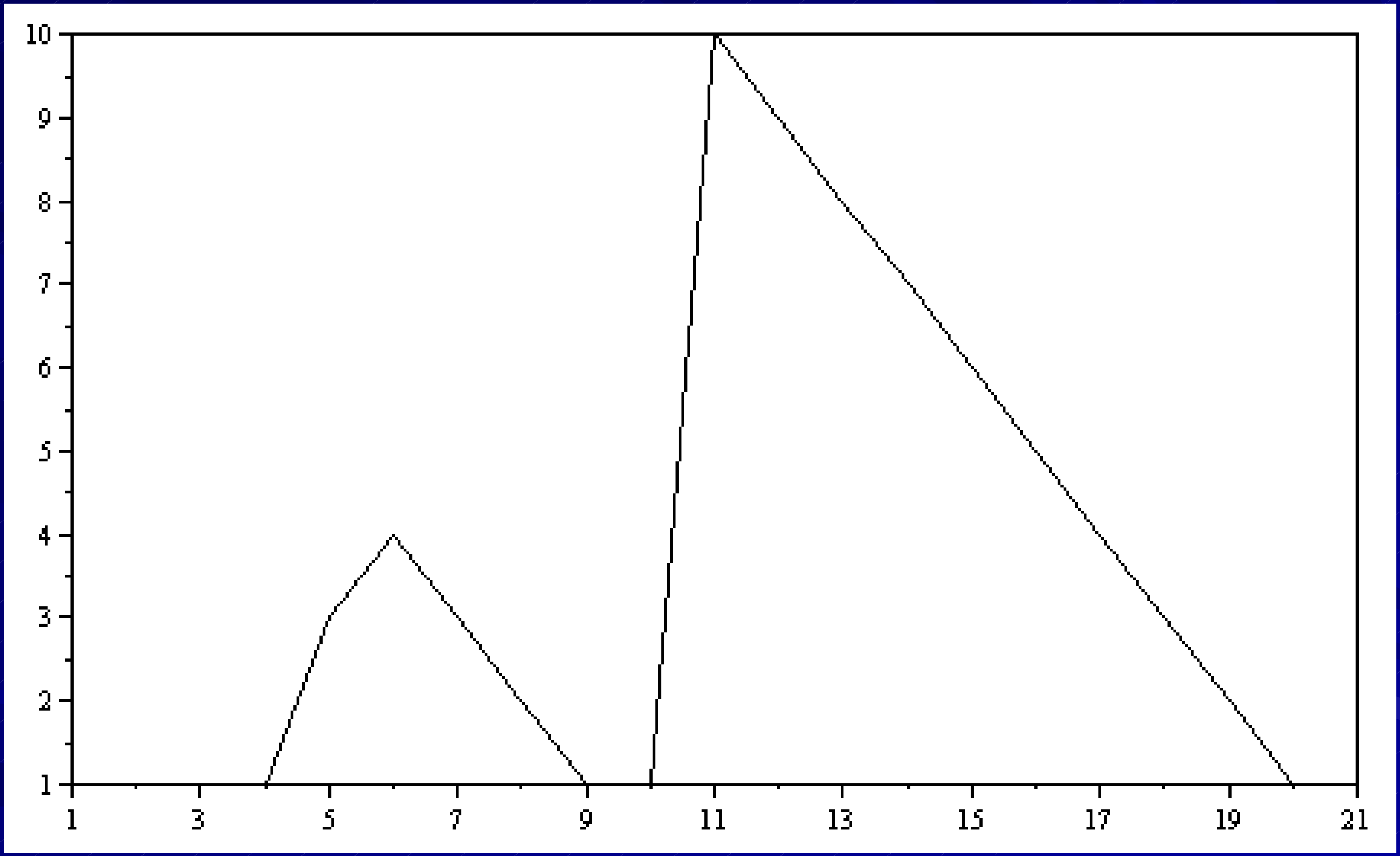
probabilité de choix de chaque action en fonction de l'état est une matrice à paramètre T=true, F=false M(21,11).

actions

états

	0	1	…	…	10
0	T	F	…	…	F
1	T	T	F		.
2	.		.	.	.
3	.			.	F
.	.				T
.	.			.	F
.	.		.	.	.
19	T	T	F		.
20	T	F	.	.	F

Une politique Pi serait une organisation des parties du capital à parier à chaque lancer



Jack's car rental

- Pendant la journée:

Pour le garage1 :

3voitures sortent et

3 voitures entrent.

Pour le garage2 :

4 voitures sortent et

2 voitures entrent.

- Pendant la nuit :

Les voitures peuvent être déplacés d'un garage à l'autre.

Numérotation des états.

- Les différents états sont le nombre de voiture qu'il y a dans les deux garages.
- Dans chaque garage il peut y avoir de 0 à 10 voitures soit 11 possibilités.
- Ainsi il y a 121 états (11 x 11).
- D'où la représentation suivante:

  Un signe négatif signifie que l'on déplace les voitures du garage2 au garage1.

Numéro de l’état	Nombre de voitures dans le garage1	Nombre de voitures dans le garage2
1	0	0
2	0	1
3	0	2
…etc…	…etc…	…etc…
117	10	6
118	10	7
119	10	8
120	10	9
121	10	10

Numérotation des actions.

- Les actions correspondent au nombre de voitures déplacées du garage1 au garage2 **pendant la nuit**.
- Nous avons décidé que nous pouvions déplacés 4 voitures au maximum.

Numéro de l’action	Nombre de voitures déplacées de nuit
1	4
2	3
3	2
4	1
5	0
6	-1
7	-2
8	-3
9	-4