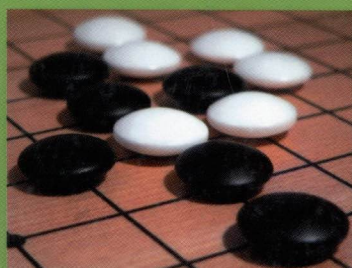




INTELLIGENCE ARTIFICIELLE

Une approche ludique

Tristan Cazenave



ellipses



Intelligence artificielle

Une approche ludique

Tristan CAZENAVE

Professeur au LAMSADE, université Paris-Dauphine

جامعة أمحمد بوقرة - بومرداس
Université M'hamed Bougara - Boumerdes

المكتبة الجامعية

رقم: 004.8/CAZ

جامعة أمحمد بوقرة - بومرداس
Université M'hamed Bougara - Boumerdes

المكتبة الجامعية

رقم الجرد: 0104146

1exp-

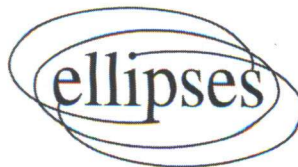


Table des matières

1 Fonctions d'évaluation	1
1.1 Influence de la représentation du problème	1
1.2 Existence de stratégies toujours gagnantes	3
1.3 Une fonction d'évaluation aux Échecs	4
1.4 Le jeu du virus	5
1.5 Recherche contre connaissances	7
1.6 Corrigés des exercices	7
1.6.1 Coups gagnants à Nim	7
1.6.2 Joueur parfait pour Nim	8
1.6.3 Fonction d'évaluation pour le jeu du virus	9
2 Minimax, Alpha-Bêta et heuristiques associées	13
2.1 Le Minimax	13
2.2 Le Negamax	16
2.3 L'Alpha-Bêta	16
2.4 Mémorisation de la variation principale	18
2.5 L'Effet d'horizon et la recherche de quiescence	18
2.6 L'approfondissement itératif	19
2.7 L'heuristique des coups qui tuent	20
2.8 L'heuristique de l'historique	21
2.9 La recherche aspirante	22
2.10 La recherche avec fenêtre nulle	22
2.11 La recherche avec variation principale	23
2.12 L'heuristique du coup nul	23
2.13 L'approfondissement sélectif	25
2.14 Corrigés des exercices	26
2.14.1 Arbre Minimax	26
2.14.2 Classe Virus incrémentale	26

2.14.3	Minimax	30
2.14.4	Negamax	31
2.14.5	Coupe bêta	32
2.14.6	Développement de l'arbre avec l'Alpha-Bêta	32
2.14.7	Alpha-Bêta	34
2.14.8	Variation principale	35
2.14.9	Quiescence	36
2.14.10	Approfondissement itératif	37
2.14.11	Coups qui tuent	40
2.14.12	Heuristique de l'historique	41
2.14.13	Recherche aspirante	44
2.14.14	Recherche avec variation principale	44
2.14.15	Heuristique du coup nul	45
2.14.16	Heuristique du coup nul vérifié	46
3	Tables de Transposition	49
3.1	Introduction	49
3.2	Hachage d'une position	50
3.3	Probabilité d'erreur	51
3.4	Stratégies de remplacement	52
3.5	Entrées de la table de transposition	52
3.6	Utilisation de la table de transposition	53
3.7	Coupes de transposition améliorées	54
3.8	La recherche avec partition	54
3.9	Corrigés des exercices	55
3.9.1	Hachage d'une position de Tic-Tac-Toe	55
3.9.2	Hachage d'une position aux Échecs et au jeu du virus	56
3.9.3	Hachage incrémental au Tic-Tac-Toe	56
3.9.4	Probabilité d'erreur	56
3.9.5	Classes génériques pour les tables de transposition	57
3.9.6	Alpha-Bêta avec tables de transposition	58
3.9.7	Coupes de transposition améliorées	63
4	Recherche avec menaces	65
4.1	Introduction	65
4.2	Le Phutball	66
4.3	Les menaces directes	68
4.4	La recherche λ	68

4.5	La réduction aux coups prometteurs	69
4.6	L'élargissement itératif	70
4.7	La limitation du nombre de coups de chaque ordre	70
4.8	Les coups qui tuent	71
4.9	Les zones pertinentes	71
4.10	Corrigés des exercices	71
4.10.1	Phutball	71
4.10.2	Positions <i>gagnant</i> ₀ à <i>gagnant</i> ₃ au Go-Moku	77
4.10.3	Menaces au Phutball	77
4.10.4	Recherche λ au Phutball	79
4.10.5	Réduction aux coups prometteurs	79
4.10.6	Elargissement Itératif	83
4.10.7	Lambda limitée	84
4.10.8	Coups qui tuent	87
5	Recherche arborescente Monte-Carlo	91
5.1	Introduction	91
5.2	Le jeu de Go	91
5.3	Algorithme basique de Monte-Carlo	92
5.4	UCB	93
5.5	UCT	93
5.6	UCT avec transpositions	94
5.7	RAVE	94
5.8	Développements	96
5.9	Corrigés des exercices	97
5.9.1	Parties aléatoires de Go	97
5.9.2	Monte-Carlo basique	108
5.9.3	UCB	109
5.9.4	UCT	111
5.9.5	UCT avec transpositions	113
5.9.6	RAVE	118
6	Recherche en meilleur d'abord pour les jeux à deux joueurs	123
6.1	L'algorithme Proof Number Search	123
6.2	L'algorithme PN ²	126
6.3	L'algorithme PN*	126
6.4	Df-pn	127
6.5	Les nombres conspirants	129

6.6	L'algorithme B*	131
6.7	Corrigés des exercices	133
6.7.1	La classe Connect	133
6.7.2	L'algorithme proof number search	135
6.7.3	L'algorithme PN ²	138
6.7.4	Nombres conspirants	140
7	Bases de données de finales	141
7.1	Les Échecs	142
7.1.1	Principe de l'algorithme	142
7.1.2	Résultats dûs aux bases de données de finales	143
7.2	Les Dames anglaises	144
7.3	L'Awele	145
7.4	WoodPush	146
7.5	Autres jeux de pions	146
7.6	Corrigés des exercices	147
7.6.1	Nombre de positions à 6 pièces aux Échecs	147
7.6.2	Indice d'une position	148
8	Recherche de plus court chemin sur une carte	149
8.1	L'algorithme de Dijkstra	149
8.2	L'algorithme A*	152
8.3	L'heuristique triangulaire	155
8.4	Recherche de plus court chemin multi-agents	157
8.4.1	Algorithme optimal	158
8.4.2	Algorithme avec replanification	158
8.4.3	Recherche coopérative	159
8.4.4	Recherche temps-réel	160
8.4.5	Recherche avec cible mouvante	161
8.5	Corrigés des exercices	161
8.5.1	Dijkstra	161
8.5.2	A*	164
8.5.3	L'heuristique triangulaire	166
9	Recherche de la solution la plus courte pour les puzzles	169
9.1	Introduction	169
9.2	Le voyageur de commerce	169
9.3	L'espace du problème	170
9.4	Le Taquin	170

9.5	Les heuristiques admissibles	171
9.6	L'algorithme A*	172
9.7	L'algorithme IDA*	173
9.8	Le Rubik's cube	173
9.9	Corrigés des exercices	174
9.9.1	Le voyageur de commerce	174
9.9.2	L'algorithme A* pour le Taquin	175
9.9.3	L'algorithme IDA* pour le Taquin	179
9.9.4	Le Rubik's cube	179
10	Bases de patterns	181
10.1	Le Taquin	181
10.2	Le Rubik's cube	182
10.3	Les bases de patterns additives	183
10.4	La compression de bases de patterns	183
10.5	Sokoban	183
10.6	La vie et la mort au jeu de Go	183
10.7	Corrigés des exercices	184
10.7.1	Le Taquin	184
10.7.2	Le Rubik's cube	186
10.7.3	Bases additives	186
11	Méthodes de Monte-Carlo pour les jeux à un joueur	187
11.1	Introduction	187
11.2	Recherche Monte-Carlo imbriquée	187
11.3	Le problème du choix du coup à gauche	188
11.3.1	Le nombre de coups sur le chemin le plus à gauche	188
11.3.2	Le nombre de coups à gauche	189
11.4	SameGame	190
11.5	Corrigés des exercices	190
11.5.1	Complexité de la recherche Monte-Carlo imbriquée	190
11.5.2	Le nombre de coups sur le chemin le plus à gauche	190
11.5.3	Le nombre de coups à gauche	191
11.5.4	SameGame	195
12	Problèmes de satisfaction de contraintes	203
12.1	Introduction	203
12.2	Exemples de Problèmes	204
12.3	Le Backtrack	204

12.4	Le Sudoku	205
12.5	Le Forward Checking	205
12.6	L'ordre d'instanciation des variables	205
12.7	La recherche avec déviations limitées	206
12.8	Les contraintes globales	206
12.9	La recherche locale	207
12.10	La recherche Monte-Carlo imbriquée	207
12.11	Corrigés des exercices	207
12.11.1	Les N reines	207
12.11.2	Le Backtrack	208
12.11.3	Le Sudoku	210
12.11.4	Le Forward Checking	212
12.11.5	L'ordre d'instanciation des variables	214
12.11.6	La recherche avec déviations limitées	214
12.11.7	La recherche Monte-Carlo imbriquée	215
13	Jeux à information incomplète	219
13.1	Introduction	219
13.2	Le Go fantôme	219
13.3	Le Bridge	219
13.4	Le Poker	220
14	Théorie combinatoire des jeux	221
14.1	Domineering	221
14.2	Les nombres surréels	222
14.3	Corrigés des exercices	223
14.3.1	↑ à Domineering	223
15	Théorie des jeux	225
15.1	Classification des jeux	225
15.2	Les jeux simultanés à information complète	226
15.2.1	La matrice de gain	226
15.2.2	Équilibre de Nash en stratégie pure	226
15.2.3	Le dilemme du prisonnier	227
15.2.4	Comment trouver un équilibre de Nash	227
15.2.5	Jeux de coordination	227
15.2.6	Jeux de la poule mouillée	227
15.2.7	Jeux sans équilibre en stratégie pure	228
15.2.8	Jeu de stratégie temps réel	228

15.3	Équilibre de Nash en stratégie mixte	228
15.4	Jeux répétés	229
15.5	Corrigés des exercices	229
15.5.1	Le dilemme du prisonnier	229
15.5.2	Équilibre de Nash	230
15.5.3	Pierre feuille ciseaux	230
15.5.4	Jeu de stratégie temps réel	230
15.5.5	Stratégie mixte	231
15.5.6	Jeu répété	231
16	Jeux généraux	233
16.1	Introduction	233
16.2	Le langage de description de jeux	233
16.3	Méthode de Monte-Carlo	234
	Bibliographie	234

Ce livre traite d'intelligence artificielle pour les jeux. Il s'adresse à des personnes ayant des rudiments de programmation.

Il aborde aussi bien les jeux de réflexion que les puzzles ou la recherche du plus court chemin dans les jeux vidéo. Chaque algorithme fait l'objet d'un chapitre. Chaque chapitre comporte un cours, des exercices et leur correction en C++.

Le contenu du livre fait l'objet de cours en Master et en école d'ingénieurs depuis plus de dix ans. Les algorithmes abordés pour les jeux à deux joueurs sont l'Alpha-Bêta et ses optimisations pour le jeu du virus, la recherche arborescente Monte-Carlo pour le jeu de Go, la recherche en meilleur d'abord pour le Gomoku et l'analyse rétrograde pour les Échecs et pour d'autres jeux. Pour les jeux à un joueur, l'algorithme A* est appliqué à la recherche du plus court chemin sur une carte ainsi qu'à la résolution de puzzles comme le Rubik's cube. L'analyse rétrograde, les méthodes de Monte-Carlo et la satisfaction de contraintes sont aussi abordées pour des problèmes à un joueur comme SameGame ou le Sudoku. Enfin les jeux à information incomplète, la théorie combinatoire des jeux, la théorie des jeux et les jeux généraux sont présentés.

Tristan Cazenave

est professeur
au laboratoire LAMSADE
à l'université Paris-Dauphine
où il fait de la recherche
en intelligence artificielle
dans le domaine des jeux.

Crédits photographiques :

© Beboy - Fotolia.com, © mweber67 - Fotolia.com,

© Dawn - Fotolia.com, © GJS - Fotolia.com



www.editions-ellipses.fr