

jeu est suffisamment simple (tel le Morpion, résolu il y a six ans), ou, au contraire, inaccessible en pratique, car nécessitant des puissances de calculs que les ordinateurs n'ont pas : c'est le cas des Échecs et de tous les jeux intéressants. Quand la stratégie optimale d'un jeu de Nim est de calcul trop complexe, on programme les ordinateurs selon une méthode dénommée *minimax*, qui consiste à énumérer jusqu'à une certaine profondeur les échanges de coups possibles en exploitant au mieux l'information obtenue.

Le *minimax* et ses variantes conduisent à de bons programmes qui battent la plupart des humains. G. Kasparov, le champion du monde d'Échecs en titre, en a fait les frais il y a trois ans, perdant contre un ordinateur. Toutefois, au jeu de Go, le niveau des programmes fondés sur le *minimax* est celui d'un débutant moyen, ce qui est décevant... ou réconfortant selon le point de vue.

L'obstacle, pour le jeu *Pair ou impair*, n'est pas celui du calcul et le *minimax* ne donne rien, car les coups sont simultanés (ou, ce qui revient au même, vous ignorez le choix de votre adversaire quand vous jouez). Chaque joueur doit s'inter-

roger sur ce qui, en fonction des coups passés, rend plus probable tel comportement de l'adversaire ou tel autre. Il ne peut jamais être certain d'avoir raison. Bien jouer à ce jeu signifie gagner plus de 50 pour cent des parties lors de séries de quelques centaines de coups.

La théorie mathématique des jeux appliquée à *Pair ou impair* donne l'indication suivante : en jouant au hasard avec une probabilité 1/2 pour pair, 1/2 pour impair (la stratégie *Mathématique*), vous assurez un minimum moyen de succès (égal à 50 pour cent), mais vous ne gagnez pas plus. Contre un adversaire qui joue tout le temps pair, *Mathématique* gagne une fois sur deux, alors que vous pourriez facilement gagner 100 pour cent des parties. De même, face à un joueur qui alternerait pair et impair, la stratégie *Mathématique* est peu judicieuse.

La stratégie *Mathématique* est parfois dénommée (à tort) *Stratégie optimale*, car c'est la meilleure des stratégies qui ne tienne pas compte du passé des échanges (ce qui n'est pas conforme à l'esprit du jeu et lui enlève tout intérêt). Comparée à d'autres, *Mathématique* est médiocre.

L'âme humaine étant rétive à toute réduction, et à toute «mise en règles»,

beaucoup de gens croient impossible que l'on réussisse à programmer un ordinateur pour qu'il gagne à *Pair ou impair*. Ils se trompent. Une série de travaux récents ont conduit Christophe Meyer à réaliser un programme, dénommé *Sagace*, qui applique les conseils du petit garçon de la nouvelle d'Edgar Poe : se mettre dans la peau de son adversaire pour sentir ce qu'il va faire et l'exploiter à ses dépens.

UNE VIEILLE MACHINE PAS SI BÊTE

Au début des années 1950, D. Hagelbarger inventa une machine pour jouer à *Pair ou impair* : elle remporta 53 pour cent des 10 000 parties engagées contre des humains. Claude Shannon, célèbre pour ses travaux sur la théorie de l'information, présenta sa propre machine en 1953, qui atteint 57 pour cent de succès (résultats moyens obtenus avec des étudiants pour des parties de 500 coups). Cette machine était réalisée par des moyens mécaniques et électriques ; aujourd'hui, on la simule par programme. La machine de Shannon, que nous dénommerons *Shannon*, est indiquée sur la figure 3. Il faut environ 40 coups pour que *Shannon* commence à jouer autre chose que des coups au hasard. Lors d'une série suffisamment longue face à un humain (200 coups ou plus), *Shannon* gagne en moyenne 57 pour cent des coups, bien qu'ayant recours au hasard dans 55 pour cent des cas. Bien sûr, si l'humain qui joue contre *Shannon* connaît la méthode de jeu, il peut gagner. Shannon (le mathématicien !) a élaboré une façon optimale de jouer contre *Shannon* (la machine !) avec laquelle on gagne, à la longue, trois coups sur quatre.

Gagner 57 pour cent en moyenne est remarquable pour un mécanisme de jeu élémentaire. Nous reviendrons plus loin sur l'explication de cette prévisibilité (partielle) du comportement humain.

LE PROGRAMME SAGACE

Le programme *Sagace* (acronyme de *Solution algorithmique génétique pour l'anticipation de comportements évolutifs*) mis au point par Ch. Meyer (de l'équipe d'Intelligence artificielle de l'Université de Paris 6, dirigé par J.-G. Ganascia), fonctionne sur une méthode beaucoup plus complexe que *Shannon*. *Sagace* s'applique à toutes sortes d'autres jeux et problèmes. Les lecteurs qui veulent essayer *Sagace* pour le jeu *Pair ou impair* peuvent le télécharger à :

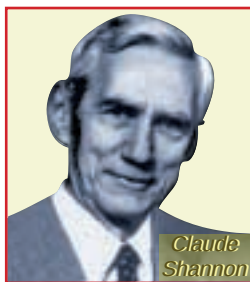
www.animaths.com/smartopponent.htm
Le programme *Sagace* s'organise autour de deux modules : le module *Compréhension de l'adversaire* et le module

2. RÉSULTATS D'UN TOURNOI DE SIX ALGORITHMES DE PAIR OU IMPAIR

	MATHÉMATIQUE	30% - 70%	SHANNON	SAGACE	PÉRIODIQUE	AUTOMATE	RÉUSSITES
MATHÉMATIQUE		50	50	50	50	50	50
30% - 70%	50		45	42	42	50	45,8
SHANNON	50	55		48	50	57	52
SAGACE	50	58	52		76	61	59,4
PÉRIODIQUE	50	58	50	24		49	46,2
AUTOMATE	50	50	43	39	51		46,6

Dans ce tableau, on a indiqué le pourcentage de réussites de six stratégies de jeu confrontées deux à deux.

- *Mathématique* joue au hasard avec une probabilité de 50 pour cent pour pair, 50 pour impair.
- *30 pour cent-70 pour cent* joue au hasard avec une probabilité de 30 pour cent pour pair et 70 pour cent pour impair.
- La stratégie *Shannon* est décrite à la figure 3.
- *Périodique* joue périodiquement la séquence complexe pair, pair, impair, au hasard, pair, pair, pair, au hasard, impair, impair, pair, au hasard, impair, impair.
- *Automate* est une stratégie qui joue selon un principe un peu plus compliqué encore.
- *Sagace* remporte le tournoi avec une moyenne de 59,4 pour cent contre 52 à *Shannon* qui arrive en second. La *Stratégie mathématique*, troisième, assure 50 pour cent (comme prévu par la théorie), mais elle est battue, car contrairement à *Shannon* et à *Sagace* elle ne sait pas exploiter les stratégies ayant une certaine régularité de comportement (pourtant ici aucune régularité n'est évidente). Notons que *Sagace* et *Shannon* sont conçues pour jouer contre des humains et non contre d'autres algorithmes.



3. LE PROGRAMME DE SHANNON

Claude Shannon, en 1953, a défini un joueur électro-mécanique pour le jeu *Pair ou impair*.

Son programme, sur une longue série de parties, gagne environ 57% des coups contre des joueurs humains.

La machine *Shannon* envisage huit situations possibles pour un joueur qui considère ses deux derniers coups :

S1 : j'ai gagné, puis j'ai rejoué la même chose et j'ai encore gagné.

S2 : j'ai gagné, puis j'ai rejoué la même chose et j'ai alors perdu.

S3 : j'ai gagné, puis j'ai changé mon jeu et j'ai encore gagné.

S4 : j'ai gagné, puis j'ai changé mon jeu et j'ai alors perdu.

S5 : j'ai perdu, puis j'ai rejoué la même chose et j'ai alors gagné.

S6 : j'ai perdu, puis j'ai rejoué la même chose et j'ai encore perdu.

S7 : j'ai perdu, puis j'ai changé mon jeu et j'ai alors gagné.

S8 : j'ai perdu, puis j'ai changé mon jeu et j'ai encore perdu.

• Pour chacune des huit situations, *Shannon* mémorise :

– si le joueur a au coup suivant rejoué comme à son dernier coup,

– ou au contraire a changé de choix.

• La machine mémorise encore si ce choix est le même que celui fait lors de la précédente rencontre de la situation.

• Le jeu de Shannon consiste à parier que le joueur qui s'est montré constant vis-à-vis d'une situation le restera encore à la prochaine rencontre de la situation.

• Si le joueur a changé d'attitude confronté à une situation donnée, *Shannon* choisit au hasard son coup, de même qu'en début de série lorsque les informations disponibles sont insuffisantes.

Choix du coup. Dans le module *Compréhension de l'adversaire*, *Sagace* élabore des règles de comportement. Nous en indiquons une, avec une notation où le premier joueur est l'adversaire humain de *Sagace*, et le second joueur (dont le coup est noté en rouge), est *Sagace* :

Si les coups précédents ont été (impair, pair) (pair, impair) (impair, impair), alors jouer impair.

De nombreuses règles de ce type sont créées et comparées au jeu de l'adversaire humain. Le module attribue aux règles qui décrivent correctement le comportement de l'adversaire un coefficient de validité plus fort qu'à celles qui le décrivent mal. Une gestion de ces règles et de leurs coefficients de validité est mise en œuvre à chaque fois qu'un nouveau coup est joué, ce qui, petit à petit, définit «un profil comportemental» de l'adversaire humain. Ce profil, qui évolue au fur et à mesure du jeu, permet, après un certain nombre de coups, de faire une prédiction du type : l'adversaire va jouer pair au prochain coup.

Dans la version actuelle de *Sagace* pour *Pair ou impair*, le nombre de coups passés susceptibles d'intervenir dans une règle est 16. Les derniers coups ont plus d'importance que les plus anciens, qui à la longue sont oubliés.

Le second module, *Choix du coup*, est fondé sur un système de trois règles :

– si, d'après le module *Compréhension de l'adversaire*, il est prévu qu'il va jouer pair, alors jouer pair ;

– si, d'après le module *Compréhension de l'adversaire*, il est prévu qu'il va jouer impair, alors jouer impair ;

– si le module *Compréhension de l'adversaire* ne prédit rien, alors *Sagace* choisit au hasard.

Dans le cas de jeux un peu plus compliqués comme Pierre/Ciseau/Papier, ou nettement plus difficiles comme *Alésia* (voir la figure 4), le second système de règles est plus complexe et fonctionne comme celui du module *Compréhension de l'adversaire*, avec des coefficients de validité évoluant au cours de la partie.

Lorsqu'une règle a été appliquée avec succès, le programme en augmente le coefficient de validité, ce qui rendra plus probable son application future. Quand la règle a conduit à un échec, son coefficient de validité est diminué. Lorsque plusieurs règles sont utilisables simultanément, le programme applique celle dont le coefficient de validité est le plus grand, mais selon un principe probabiliste qui a pour effet de rendre non déterministe le comportement de *Sagace* : face à une même série de coups, *Sagace* ne réagit pas toujours de la même manière.

Un jeu de règles initiales est présent dans le système en début de partie (il a été mis au point soigneusement à la suite de longues séries de tests), mais,

en plus, à chaque étape du jeu, de nouvelles règles sont créées par ce qu'on dénomme la méthode des algorithmes génétiques : on exerce une légère modification sur l'une des règles qui possède un bon coefficient de validité (méthode des mutations), ou on mélange deux bonnes règles entre elles. Un programme simplifie plusieurs règles ayant une zone commune pour n'en garder qu'une.

Des règles nouvelles sont aussi ajoutées par application de ce que Ch. Meyer appelle la méthode du regret : après un coup où il vient de perdre, *Sagace* ajoute une règle indiquant ce qu'il aurait dû faire pour gagner. Le tout déclenche une assez grande quantité de calculs à chaque coup, mais, avec les machines d'aujourd'hui, agit instantanément. Ainsi, appliquant l'idée d'Edgar Poe de se mettre dans la tête de l'adversaire, le programme *Sagace* essaie à la fois de deviner ce qu'il va jouer et s'adapte à ses résultats passés pour tenter de jouer au mieux.

UNE DOMINATION LÉGÈRE, MAIS IMPLACABLE

Opposé à un humain, en début de partie (les 30 premiers coups), *Sagace* ne fait pas mieux que le hasard. Ensuite, *Sagace* remporte environ 62 pour cent des parties, ce qui est sensiblement supérieur au score de *Shannon*. Avec certains joueurs, *Sagace* gagne moins, mais il est bien rare de le battre lors d'une partie de 200 coups.

Il est particulièrement agaçant de jouer contre *Sagace* (mais c'est déjà vrai contre *Shannon*), car malgré les efforts faits pour être imprévisible et pour changer de stratégie, et malgré les ruses mises en œuvre pour piéger votre adversaire de silicium, celui-ci prend de l'avance. Petit à petit, il profite de vos périodes de lassitude où inconsciemment vous vous comportez trop simplement ou adoptez à nouveau un comportement que vous aviez déjà eu (ce qu'il exploite à son profit). Il grignote des points ou regagne l'avance que vous aviez constituée en début de partie, et, dans la grande majorité des cas, il finit par remporter la série de coups.

On constate en particulier que tout comportement périodique de période inférieure à 10 est identifié et utilisé contre vous, amenant vite un gain de 100 pour cent en faveur de *Sagace*. Évidemment, vous pouvez battre *Sagace* en tirant à pile ou face votre coup (pile pour pair, face pour impair) et jouer alors la stratégie *Mathématique*, qui remporte un coup sur deux en moyenne. Cependant, jouer de cette façon est une forme de tricherie, puisque *Sagace* a été conçu pour jouer contre un adversaire humain qui doit simuler le hasard mentalement.