



3. LE PROGRAMME DE SHANNON

Claude Shannon, en 1953, a défini un joueur électromécanique 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.

On peut dire que *Shannon* et encore plus *Sagace* lisent dans nos pensées, non pas au point de nous écraser à chaque rencontre (ce qui serait de la magie), mais suffisamment pour gagner la plupart des longues séries jouées sans tricher.

TOURNOI DES MACHINES

Conçue pour jouer contre des adversaires humains, *Sagace* a cependant été testée contre un ensemble d'adversaires algorithmiques. Les résultats obtenus sont riches d'enseignements.

Contre la stratégie *Mathématique* (au hasard 50 pour cent de pair, 50 pour cent d'impair), *Sagace*, nous le savons, gagne 50 pour cent des coups, et donc 50 pour cent des parties.

Lorsque *Sagace* joue contre *Shannon*, il gagne 52 pour cent des coups, ce qui est bien, mais loin des résultats de la stratégie spécialement adaptée que *Cl. Shannon* avait imaginée et qui, elle, réussit à 75 pour cent contre *Shannon* ; *Sagace* ne converge pas vers le meilleur jeu possible face à *Shannon*, ce qu'on aurait pu espérer, et cette imperfection ouvre la porte à de futurs perfectionnements.

Contre une stratégie qui joue au hasard pair ou impair dans des proportions 30 pour cent-70 pour cent, *Sagace* gagne 58 pour cent des parties. Là encore, on est en dessous de ce que donnerait le jeu le mieux adapté (qui consisterait à jouer toujours pair et ferait gagner 70 pour cent des coups). La stratégie mathématique contre la stratégie 30 pour cent-70 pour cent ne gagne que 50 pour cent des coups, ce qui est moins bien que *Sagace*.

Contre la stratégie appelée *Périodique* qui joue cycliquement : pair, pair, impair, au hasard, pair, pair, pair, au hasard, impair, impair, pair, au hasard, impair, impair (ce qui n'est pas très facile à deviner, vous en conviendrez !), *Sagace* fait un score de 76 pour cent.

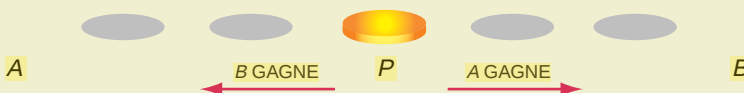
Contre une stratégie algorithmique réactive baptisée *Automate*, la stratégie *Sagace* réussit à 61 pour cent, ce qui ne signifie pas que *Sagace* réussirait aussi bien contre toute stratégie algorithmique réactive.

Au total, en imaginant que chacune des six stratégies (*Shannon*, *Sagace*, *Stratégie mathématique*, *Hasard 30 pour cent-70 pour cent*, *Périodique*, *Automate*) affronte chacune des cinq autres et qu'on compare leurs résultats globaux, *Sagace* est première, suivie de *Shannon* (voir la figure 2).

GÉNÉRALISATION

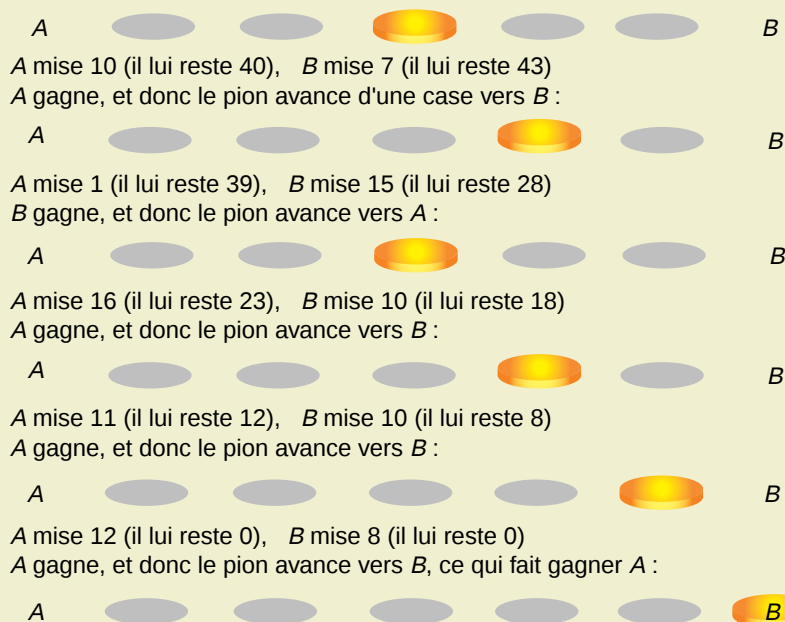
Les idées mises en œuvre dans *Sagace* peuvent être appliquées à d'autres jeux du même type comme le fameux

4. LE JEU D'ALÉSIA



Un pion est placé au départ en *P*, et chaque joueur dispose d'une cagnotte de 50 points. À chaque coup, les joueurs choisissent secrètement un certain nombre de points qui sont soustraits à leur cagnotte. Ce choix fait, les joueurs dévoilent leurs mises. Celui qui a proposé le plus avance le pion d'un cran vers le camp de son adversaire. Lorsqu'un joueur arrive dans le camp adverse, il a gagné.

EXEMPLE DE PARTIE



La partie peut être nulle, auquel cas nul ne marque de point. Le programme *Sagace* a appris à jouer à ce jeu, ce qui lui a conféré une connaissance tactique (la théorie du jeu est maintenant classique). Plus original, lors d'une série de parties que vous jouez contre lui, *Sagace* élabore votre profil psychologique, qu'il exploite ensuite contre vous. Même le concepteur de *Sagace* ne réussit plus aujourd'hui à battre son propre programme.