

# Classes complètes de stratégies pour le Dilemme Itéré des Prisonniers Classique\*

Bruno BEAUFILS

beaufils@lifl.fr

Laboratoire d'Informatique Fondamentale de Lille

U.R.A. 369 C.N.R.S. – Université des Sciences et Technologies de Lille

U.F.R. d'I.E.E.A. Bât. M3

59655 Villeneuve d'Ascq Cédex – FRANCE

## Résumé

Dans cet article après un bref rappel de la description du Dilemme Itéré des Prisonniers Classique (CIPD), et de la manière dont il est utilisé dans l'étude de l'évolution de la coopération entre agents, nous montrons comment des idées simples peuvent être utilisées, avec une approche génétique, pour générer automatiquement un grand nombre de stratégies. Ensuite nous montrons quelques résultats d'évolution écologique sur ces stratégies avec la description des expériences que nous avons menées. Notre but principal étant de trouver une méthode objective d'évaluation des stratégies de coopération dans le CIPD. Finalement nous utilisons les résultats précédents pour ajouter un argument en faveur de notre idée affirmant que pour être *bonne* une stratégie n'a pas être simple, et qu'au contraire il existe un gradient de complexité infinie dans la structure de telles stratégies.

## 1 Le Dilemme Itéré des Prisonniers Classique

Introduit par Merill M. FLOOD et Melvin DRESHER à la RAND Corporation en 1952, voir [5], le Dilemme Itéré des Prisonniers Classique (*Classical Iterated Prisoner's Dilemma (CIPD)*), est le modèle formel d'étude de la coopération et de l'évolution de la coopération le plus utilisé.

Il est basé sur cette petite question proposé par Albert TUCKER par exemple dans [6, pages 117–118] :

*Deux hommes, accusés d'avoir violé la loi, sont arrêtés par la police et se voit simultanément, mais séparément, proposer un marché. On leur dit que*

*(1) si l'un avoue et pas l'autre, alors le premier aura une récompense et le second une amende*

*(2) si les deux avouent les deux auront une amende*

*Mais dans le même temps, chacun a une bonne raison de penser que*

*(3) si aucun des deux n'avoue, les deux seront libérés sans aucune poursuite.*

Clairement le choix le plus raisonnable est de trahir son partenaire.

Plus formellement le CIPD est représenté en théorie des jeux comme un jeu simultané à deux joueurs et à somme non-nulle où chaque joueur a le choix entre deux coups :

– COOPÉRER, on notera C, et on dira être gentil

– TRAHIR, on notera D, et on dira être méchant

Le gain de chaque joueur dépend des coups joués par les deux agents. La table 1 nomme ces scores.

Pour qu'il y ait dilemme, l'inéquation suivante doit être respectée :

$$S < P < R < T \tag{1}$$

---

\*Par manque de place les résultats complets des mes expériences ont été omis, mais peuvent être trouvés dans [4]

TAB. 1 – Matrice de gain du CIPD. *Le score du joueur de la ligne est donné en premier.*

	Cooperate	Defect
Cooperate	$R = 3, R = 3$ <i>Reward</i> for mutual cooperation	$S = 0, T = 5$ <i>Sucker's payoff</i> <i>Temptation to defect</i>
Defect	$T = 5, S = 0$ <i>Temptation to defect</i> <i>Sucker's payoff</i>	$P = 1, P = 1$ <i>Punishment</i> for mutual defection

Comme la version en un coup est résolu par l'équilibre de NASH, le modèle est étendu : dans une version itéré, les joueurs se rencontrent régulièrement, sans savoir exactement combien de fois. Le gain est alors simplement la somme des gains reçus lors de chaque rencontre. Pour favoriser la coopération, et aussi garder la différence entre l'intérêt individuel et l'intérêt collectif l'inéquation suivante doit également être respectée :

$$S + T < 2R \tag{2}$$

Un choix de valeurs classiques est donnée dans la table 1.

## 2 Tournois et compétitions écologiques

Deux types de méthodes peuvent être utilisées pour évaluer les stratégies pour le CIPD.

La première consiste en un tournoi entre différentes stratégies. Le gain de chaque stratégie étant la somme de ses scores dans chaque jeu itéré. Un classement peut alors être établi, en fonction du score de chacune. Plus une stratégie est bien classée, meilleure elle sera considérée.

Le second type d'expérimentations est une sorte d'imitation des processus d'évolution et de sélection naturelle, et est très proche des problèmes de dynamique de population. Considérons une population de  $N$  joueurs, chacun choisissant une stratégie dans un ensemble. Au départ on considère que chaque stratégie de cet ensemble est également représentée dans la population. Ensuite un tournoi est effectué et les bonnes stratégies sont favorisées par rapport aux moins bonnes. Cette redistribution se fait de manière proportionnelle. Ce schéma, aussi appelé une génération, est répété jusqu'à une éventuelle stabilisation de la population, *i.e.* pas de changements dans la population entre deux cycles.

Une bonne stratégie est alors une stratégie qui reste dans la population le plus longtemps possible, et dans la plus grande proportion possible.

Les résultats classiques sur le sujet, qui ont été mis en valeur par AXELROD dans [1, 2], montre que pour être efficace une stratégie doit :

- être gentille, *i.e.* ne jamais être la première à trahir
- être réactive
- savoir pardonner
- être simple, *i.e.* pouvoir être comprise facilement

La célèbre stratégie `tit_for_tat`<sup>1</sup>, qui satisfait ces critères a, depuis la large diffusion des travaux d'*Axelrod*, été considérée comme la meilleure stratégie, non seulement pour la coopération, mais également pour l'évolution de la coopération.

Nous pensons que la *simplicité* n'est pas un bon critère, et avons à cet effet introduis dans [3] une stratégie appelée `graduelle`<sup>2</sup>, qui illustre nos idées.

Nos directions de recherches principales pour conforter notre idée sur la *complexité* sont :

- d'essayer le plus de stratégies différentes possibles, d'une manière automatique et objective
- de créer une méthode d'évaluation objective des stratégies

<sup>1</sup>`tit_for_tat` coopère le premier coup, puis joue ce que son adversaire a joué au coup précédent

<sup>2</sup>`graduelle` coopère le premier coup, puis après la première défection de son adversaire trahis une fois et coopère deux fois, après la deuxième trahison, trahis deux fois et coopère deux fois, ..., après la  $n^e$  trahison trahis  $n$  fois et coopère deux fois

### 3 Les classes complètes de stratégies

Dans le but de nous aider dans cette recherche, nous devons trouver une méthode *descriptive* de définir des stratégies, qui est plus fiable qu'une méthode exhaustive, ne pouvant être ni objective ni complète.

Une des solutions que nous avons choisi est d'utiliser une approche génétique. Nous devons donc décrire une structure (on dira un génotype) qui peut être décodé en un comportement (on dira un phénotype).

Une méthode pour avoir une stratégie est alors simplement de remplir cette structure. Une manière d'avoir beaucoup de stratégies est de considérer toutes les manières de remplir cette structure comme autant d'individus. On appelle alors l'ensemble de toutes ces stratégies décrite par un génotype particulier, la *classe complète* des stratégies de ce génotype.

Nous avons décrit trois génotypes basés sur la même idée simple, dans le but de rester objectif. Cette idée est de considérer la longueur de l'historique du jeu visible par les joueurs. De telles idées ont déjà été étudiées dans [8].

Ces trois stratégies sont :

`memory` : Chaque stratégie voit seulement  $M_{ml}$  coups de son passé, et  $O_{ml}$  coups du passé de son adversaire. Le jeu est amorcé par  $\max(M_{ml}, O_{ml})$  coups prédéfinis dans le génotype. Tous les autres coups sont codés dans le génotype en fonction de la configuration du passé visible. La longueur du génotype est alors  $\max(M_{ml}, O_{ml}) + 2^{(M_{ml}+O_{ml})}$ .

`binary_memory` : Même chose que pour la précédente, à l'exception du fait que la réponse à l'adversaire ne dépend plus seulement du passé visible, mais aussi du fait que l'adversaire a plus souvent trahis que coopéré, ou non. La longueur du génotype est alors  $\max(M_{ml}, O_{ml}) + 2^{(M_{ml}+O_{ml}+1)}$ .

`memory_automata` : On représente ici des automates classiques à deux états, qui débutent en l'état 0. Chaque stratégie voit seulement  $M_{ml}$  coups de son passé, et  $O_{ml}$  coups du passé de son adversaire. Le jeu est amorcé par  $\max(M_{ml}, O_{ml})$  coups prédéfinis dans le génotype. Tous les autres coups sont codés dans le génotype en fonction de la configuration du passé visible et de l'état courant. Les transitions d'états sont aussi codées dans le génotype. La longueur du génotype est alors  $\max(M_{ml}, O_{ml}) + 2^{(M_{ml}+O_{ml}+2)}$ . Des stratégies de ce genre ont déjà été étudiées dans [7].

Il est à noter que malgré l'apparente simplicité des stratégies décrites par ces génotypes, un grand nombre de stratégies classiques, comme `tit_for_tat` sont incluses dans ces classes complètes.

### 4 Les expériences

Nous avons fait quelques expériences sur ces classes complètes. Le but principal étant d'évaluer d'autres stratégies dans de grandes compétitions écologiques.

Une évolution écologique est faite entre toutes les stratégies d'une classe, et la stratégie à évaluer.

Les figures 1 et 2 représentent les évolutions des populations de deux classes complètes ainsi que la même évolution dans laquelle `graduelle` a été ajoutée.

D'autres résultats sont donnés dans le tableau 2.

### 5 Conclusions

La méthode génétique utilisée pour définir des stratégies pour le CIPD offre deux gros avantages. Tout d'abord il est aisé de définir un grand nombre de stratégies, ensuite la manière dont elles sont définies est objective, *i.e* sans choix biaisé, de telle sorte qu'elle peut être utilisée comme évaluation de stratégies.

Avec les classes complètes de stratégies, nous avons fait certaines évaluations qui confirment nos idées à propos de la complexité dans les stratégies de coopération entre agents.

Nous espérons pouvoir inclure ce genre d'évaluation dans une méthode plus globale d'évaluation de stratégies.

Un logiciel de simulation avec de nombreuses stratégies est disponible pour les systèmes UNIX, DOS et Windows sur le World Wide Web à l'adresse <http://www.lifl.fr/~mathieu/ipd> ou par ftp anonyme sur le site <ftp.lifl.fr> dans `pub/users/mathieu/soft`.

TAB. 2 – Quelques résultats d'évaluation de *graduelle* dans des classes complètes. *taille* est le nombre de stratégies décrites, alors qu' *évaluation* correspond au rang de la stratégie à la fin de l'évolution de la classe complète.

	$M_{ml}$	$O_{ml}$	taille	évaluation		
				gradual	tit_for_tat	spiteful
memory	0	1	8	1	1	1
	0	2	64	5	2	21
	1	1	32	2	3	1
	1	2	1024	6	13	37
binary_memory	0	1	32	1	2	1
	1	1	512	1	7	13
memory_automata	0	1	512	1	31	32

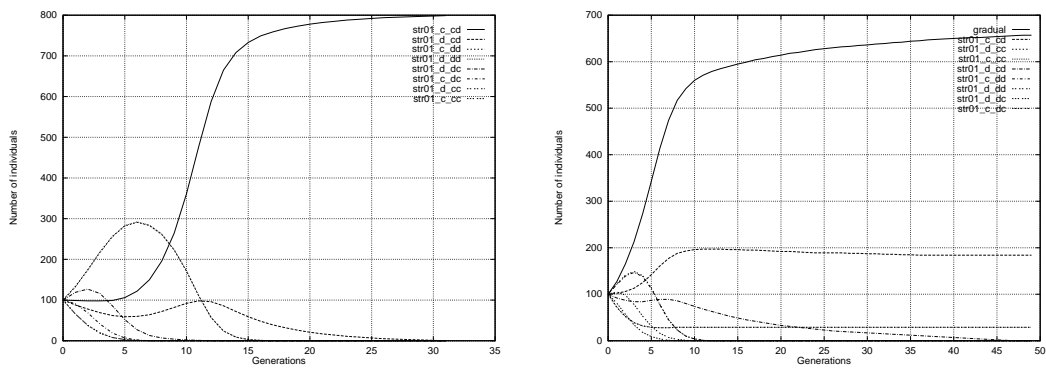


FIG. 1 – Évolution de la classe *memory* ( $M_{ml} = 0$  and  $O_{ml} = 1$ )

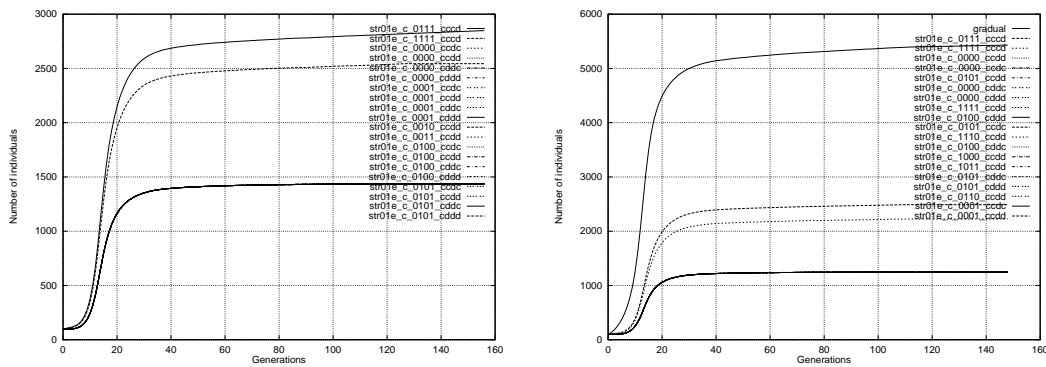


FIG. 2 – Évolution de la classe *memory\_automata* ( $M_{ml} = 0$  and  $O_{ml} = 1$ )

## Références

- [1] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, USA, 1984.
- [2] R. Axelrod. *Donnant donnant : théorie du comportement coopératif*. Éditions Odile Jacob, Paris, France, 1992. ISBN 2-7381-0145-3. Traduction française de [1].
- [3] B. Beaufils, J.P. Delahaye, and P. Mathieu. Our meeting with gradual, a good strategy for the iterated prisoner's dilemma. In Christopher G. Langton and Katsunori Shimohara, editors, *Artificial Life V : Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 202–209, Cambridge, MA, USA, 1996. The MIT Press/Bradford Books.
- [4] Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Complete classes of strategies for the classical iterated prisoner's dilemma. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *EVOLUTIONNARY PROGRAMMING VII*, volume 1447 of *Lecture Notes in Computer Science*, pages 33–41, Berlin, 1998. Springer-Verlag. ISBN 3-540-64891-7.
- [5] Merrill M. Flood. Some experimental games. Research memorandum RM-789-1-PR, RAND Corporation, Santa-Monica, CA, USA, June 1952.
- [6] W. Poundstone. *Prisoner's Dilemma : John von Neumann, Game Theory, and the Puzzle of the Bomb*. Oxford University Press, Oxford, UK, 1993.
- [7] Abdellah Salhi, Hugh Glaser, David De Roure, and John Putney. The Prisoner's Dilemma Revisited. Technical Report DSSE-TR-96-2, University of Southampton, Department of Electronics and Computer Science, Declarative Systems and Software Engineering Group, Southampton, UK, March 1996.
- [8] Tuomas W. Sandholm and Robert H. Crites. Multiagent reinforcement learning in the Iterated Prisoner's Dilemma. *BioSystems*, 37(1,2) :147–166, 1996. Special Issue, Prisoner's Dilemma.