Table des matières

Chapitre 1

Introduction

Le Dilemme Itéré des Prisonniers est un problème classique dans de nombreuses disciplines. Issu de la théorie des jeux, il s'applique aux sciences politiques aussi bien qu'à la biologie de l'évolution, alors que son étude peut se faire avec la même pertinence par des informaticiens, que des zoologues. C'est un des problèmes les plus connus du domaine particulier qu'est la Vie artificielle, dans lequel la simulation occupe une place prépondérante, et que l'on peut voir comme une sorte de *biologie théorique analytique*, dans laquelle au lieu d'étudier les êtres vivants par décomposition, on crée des agents artificiels afin d'étudier leurs comportements collectifs, de façon à pouvoir, par exemple, vérifier certains principes énoncés en théorie de l'évolution.

Le but de mon sujet de DEA était d'approcher le domaine de la Vie Artificielle à travers le Dilemme Itéré des Prisonniers. Cette approche a donc du comporter plusieurs travaux dont une étude approfondie du dilemme par la bibliographie, et par la vérification de certains résultats déjà établis ; mais aussi la mise au point d'un *langage* de description des stratégies pour le dilemme, et l'utilisation d'Algorithmes Génétiques pour étudier ces mêmes stratégies, et plus précisement parcourir de manière *intelligente*, rapide, et automatique l'espace des stratégies défini par ce langage à la recherche de stratégies efficaces.

Le but de ce mémoire est, lui, de faire l'état des différents travaux qui ont été effectués, et des perspectives de poursuite de ceux-ci. Il comporte donc les points suivants :

- une présentation du Dilemme Itéré des Prisonniers, et de l'étude qui en a été faite ;
- une présentation des Algorithmes Genétiques, et de leur application au Dilemme Itéré des Prisonniers ;
- une description des expériences menées et des résultats obtenus.

En annexe à ce document sont consignées la description complète des résultats des différentes expériences ainsi que des références bibliographiques concernant le dilemme.

Cette approche de la Vie artificielle par le Dilemme Itéré des Prisonniers fait suite à des études déjà menées au Laboratoire, consignées dans [DM92], [DM93] et [Lad94].

Chapitre 2

Le Dilemme Itéré des Prisonniers

2.1 Historique

Le dilemme des prisonniers, ainsi que sa version itérée qui nous intéresse plus particulièrement ici, a été introduit par Merrill FLOOD et Melvin DRESHER, en 1952, suite à des travaux sur les points d'équilibres de John NASH dans la théorie des jeux de John VON NEUMANN, et Oskar MORGENSTERN. Le but était d'introduire une certaine dose *d'irrationalité* dans cette théorie, de laquelle sont issus entre autre le théorème du minimax, et l'étude des jeux à somme nulle, qui permettent de déterminer *la* meilleure stratégie rationnelle à adopter dans certains types de jeux.

C'est au sein de la RAND Corporation [1], que ces travaux furent effectués. Cet organisme, crée par Douglas Aircraft, et le gouvernement des États Unis, au sortir de la seconde guerre mondiale, regroupait alors parmi les plus prestigieux des scientifiques de l'époque, dont la plupart avait participé au projet *Manhattan*, conduit par Robert J. OPPENHEIMER, qui fût à l'origine de la création de la bombe atomique.

C'est donc dans un premier temps, pour justifier certaines opinions sur les relations entre nations, que le dilemme fut utilisé. Une opinion particulièrement extrémiste, mais néanmoins trés sérieuse, se proposait d'attaquer l'Union Soviétique, le plus vite possible, afin de prévenir une guerre qui, si elle avait lieu plus tard, aurait pu tourner au désavantage des USA.

Puis, avec le temps les travaux sur le dilemme se firent de plus en plus rares, car on avait su l'énoncer, mais pas le résoudre, ou en tout cas, pas su lui trouver de solutions acceptables.

Les travaux de Robert AXELROD [2], au début des années 1980, le remirent au goût du jour, non seulement parce qu'ils apportaient un éclairage nouveau sur le domaine, mais également parce que la puissance de calcul offerte par l'accroissement de la rapidité des ordinateurs, laissait entrevoir des perspectives d'études nouvelles dans ce domaine. Le renouveau de ce sujet coïncida avec l'apparition d'un autre domaine de recherche, dans lequel de nombreuses spécialités sont représentées : la Vie Artificielle.

Aujourd'hui les travaux sur le dilemme sont nombreux, et portent essentiellement sur des variantes du dilemme initial, pour lequel les résultats obtenus par AXELROD, sont souvent considérés comme définitifs. C'est justement sur ce point que j'ai du travailler ; notre [3] opinion étant différente.

Pour plus de précisions sur la grande et la petite histoire du dilemme des prisonniers, et les applications de la théorie des jeux, entre autre, par les militaires américains, on pourra se reporter à [Pou93], qui est une des références les plus complètes sur la question.

1. Research ANd Development
2. [AH81] et [Axe84] notamment
3. Les travaux sur le dilemme, effectués au sein du Laboratoire, essaient justement de montrer que les résultats d'AXELROD sont inexacts sur certains points.

2.2 Description du Dilemme (Itéré) des Prisonniers

2.2.1 Le dilemme simple

Paradoxes et autres histoires

Deux individus porteurs d'une arme sont arrêtés par la police devant une banque, qui vient d'être dévalisée. Ils sont alors conduits au commissariat, et isolés l'un de l'autre. La police n'ayant aucune preuve de la culpabilité de l'un ou de l'autre, se doit d'obtenir des aveux. Pour cela elle propose un marché à chacun des deux suspects : avouer ou se taire, avec cependant quelques conditions :

- si l'un des deux avoue et l'autre se tait, alors celui qui aura avoué sera libéré, en remerciement de sa collaboration avec la justice, et l'autre fera 5 ans de prison, pour avoir essayer de tromper cette même justice ;

- si les deux avouent, ils feront tous les deux 4 ans de prison, pour avoir attaquer une banque ;

- si les deux se taisent, ils feront alors tous les deux 2 ans de prison, pour port d'armes illicite (ce qui est cher payé).

L'intérêt de la police est clair dans ce marché, puisqu'à tous les coups elle a un coupable, et c'est ce qu'elle recherche, la justice et la vérité n'étant qu'accessoires. Penchons-nous plutôt sur l'intérêt des suspects. Chacun des deux individus, qui sont considérés comme rationnels, raisonne de la manière suivante :

> « Si mon ami se tait, j'ai intérêt à avouer, et donc le trahir, et ainsi être libéré ; mais s'il avoue, j'ai aussi intérêt à avouer, car dans ce cas je ne ferais que 4 ans de prison, au lieu de 5 si je me tais. Donc dans tous les cas je dois avouer, donc j'avoue... »

Les deux suspects vont donc avouer, et faire chacun 4 ans de prison, satisfaisant ainsi la police, alors que s'ils s'étaient tus tous les deux ils n'auraient fait que 2 ans de prison.

Il y a donc un paradoxe entre l'intérêt commun, et l'intérêt personnel.

Un individu a volé le Youkounekoune, le plus gros diamant au monde. Malheureusement il n'y a qu'une personne assez riche, qui puisse le lui acheter. Après avoir contacté ce receleur ils s'entendent sur la manière de procéder pour l'échange, afin d'éviter toute intimidation, ou tricherie. Chacun des deux individus se rend en un endroit connu de lui seul, l'un avec le diamant, l'autre avec l'argent ; chacun dépose son bien à cet endroit, et ils s'appellent au téléphone à un moment bien précis, pour s'indiquer l'un à l'autre le chemin à suivre pour prendre livraison de leur marchandise respective. Les deux chemins sont bien entendu unique, et donc différents.

Le voleur se demande alors pourquoi il devrait partir sans le diamant, il pourrait aller chercher l'argent, et partir avec l'argent et le diamant, ce qui est bien entendu malhonnête, mais bigrement rentable. Il se dit alors que l'acheteur peut tout à fait se tenir le même raisonnement, et dans ce cas il n'a pas intérêt à laisser le diamant, ce qui lui ferait courrir le risque de se trouver les mains vides. Il doit donc aller chercher l'argent avec le diamant. L'acheteur en fait de même et va donc chercher le diamant avec son argent.

Ils ont tous deux suivis la meilleure stratégie pour leur intérêt personnel. Ils ne sont pas pour autant plus avancés.

Une fois de plus l'intérêt commun diffère de l'intérêt personnel, et ce paradoxe nuit finalement aux deux personnages.

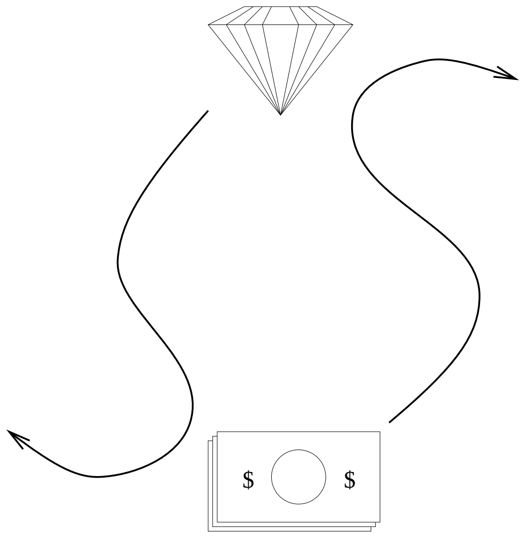

FIG. 2.1 - *Chemins pour le voleur et le receleur*

Formalisation du dilemme

Soit deux individus ayant le choix entre la coopération, ou la trahison. Ils jouent l'un contre l'autre et se voient chacun attribuer un score en fonction de la configuration de jeu dans laquelle ils se mettent. Le but du jeu est d'obtenir le plus gros score, pour gagner.

Les scores sont attribués comme ceci :

- les deux coopèrent, ils obtiennent chacun la récompense de la coopération (*Reward*), soit un total de R points ;
- les deux trahissent, ils obtiennent chacun la punition de l'égoïsme (*Punition*), soit un total de P points ;
- l'un trahit et l'autre coopère, celui qui a trahit obtient la tentation de l'égoïste (*Temptation*), soit T points, et celui qui a coopéré reçoit le salaire du dupe (*Sucker*), soit S points.

Pour conserver le dilemme on s'arrange pour que la tentation rapporte plus que la coopération, qui rapporte plus que la punition, qui rapporte plus que la duperie, soit :

$$T > R > P > S$$

2.2.2 Le dilemme itéré

Itération du dilemme

Dans cette version du dilemme, l'intérêt est limité, car si la partie ne dure qu'un coup, alors il est évident que le coup à jouer est la trahison, car elle assure un minimum de points. L'idée de FLOOD est donc de faire répéter ce jeu un nombre inconnu de fois à des joueurs, et d'essayer de voir la stratégie que chacun va adopter, mais aussi, et surtout le score que chacun, va faire, en cumulant les scores de chaque étape.

Il faut pour éviter de répercuter la solution du dilemme en un coup, éviter de donner trop d'importance à la tentation par rapport à la coopération, et pour ce faire on rajoute la restriction suivante à la formalisation précédente :

$$2R > T + S$$

Cette nouvelle restriction permet de favoriser la coopération mutuelle par rapport au cavalier seul.

Le jeu de scores le plus souvent utilisé est celui présenté dans le tableau 2.1

T	5
R	3
P	1
S	0

TAB. 2.1 - *Scores classiques*

C'est donc simplement en itérant le dilemme des prisonniers, un certain nombre de fois que l'on voit apparaître des propriétés intéressantes.

Propriétés induites par l'itération

Comme le dilemme est maintenant un jeu à part entière, chaque individu se doit d'avoir une stratégie, un comportement à tenir face à ses adversaires, en fonction du comportement de ceux-ci. C'est alors l'étude de ces stratégies qui devient intéressante, car elles peuvent être révélatrices de certains phénomènes naturels ou sociaux.

C'est dans cette optique que les travaux d'AXELROD se sont dirigés, et ont aboutis. Pour cela il a organisé un concours ouvert à tous, où l'on proposait des stratégies, et où toutes les stratégies proposées se rencontrent dans un grand tournoi généralisé, à l'issu duquel elles obtiennent un score, qui permet d'établir un classement. Ce genre de tournoi [4] a également été organisé par Jean-Paul DELAHAYE, et Philippe MATHIEU, au sein du mensuel *Pour la science*, pour reprendre et commenter les résultats d'AXELROD, qui sont présentés ci-dessous ; on pourra trouver les informations concernant ce tournoi dans [DM93].

D'après AXELROD les stratégies pour avoir de bons résultats doivent satisfaire certaines propriétés :

1. *La bienveillance*, la stratégie ne doit pas être la première à trahir ;
2. *La réactivité*, la stratégie doit réagir aux aggressions de l'adversaire, *i.e.* l'aggresser à son tour ;
3. *L'indulgence*, la stratégie doit, certes, punir son adversaire quand il a trahi, mais doit savoir oublier, pour permettre à la coopération de se réinstaller ;
4. *La simplicité*, la stratégie doit avoir un comportement clair, pour être comprise par l'adversaire rapidement, et instaurer ainsi un climat de coopération mutuelle.

Pour lui la stratégie la plus proche de ces objectifs est `tit_for_tat`, dont la description se trouve dans l'annexe A.

Lors de travaux déjà cités DELAHAYE et MATHIEU, ont réussi à trouver une stratégie meilleure que `tit_for_tat`, qui a bien les propriétés énoncées par AXELROD, sauf une : la simplicité. C'est donc ce point de vue que ce mémoire essaie de défendre, en faisant suite à des expériences probantes sur les stratégies ayant une vision limitée du passé.

4. Sur une variante du dilemme avec un choix supplémentaire : le renoncement

Chapitre 3

Mesures de Robustesse

Avant de commencer l'étude des Algorithmes Génétiques, et de l'espace des stratégies du Dilemme Itéré des Prisonniers, nous avons mis au point une technique inédite de mesure de la robustesse du jeu, et des stratégies, pour s'assurer de la pertinence du travail, et de son efficacité. Cette technique utilise les sous-ensembles de stratégies.

3.1 Robustesse du jeu

Un certain nombre de stratégies de base (cf. annexe A), se retrouvent dans les différents travaux sur le Dilemme Itéré des Prisonniers. Nous avons donc étudié, et mis au point un moyen de vérifier que le Dilemme Itéré des Prisonniers, est un jeu stable quelque soit l'environnement de stratégies qu'il utilise.

3.1.1 La méthode

Problème

La méthode utilisée est relativement simple, elle est basée sur l'idée que pour que le jeu soit intéressant à étudier, il faut que les stratégies soient stables quelque soit leur environnement. En effet il est, par exemple, bien clair que dans un environnement de stratégies bienveillantes non réactives, la stratégie `méchante`, est une bonne stratégie, mais elle est nettement moins forte dans un environnement de stratégies réactives. La question qui nous a intéréssé est de savoir si le jeu offre d'autres situations où ce cas se reproduit, et surtout combien de telles situations il peut y avoir.

Si ce nombre est trop élevé, l'intérêt d'étudier le Dilemme Itéré des Prisonniers retombe, car cela revient à étudier un cas beaucoup trop particulier d'évolution de la coopération.

Pour cela nous avons pensé utiliser la méthode suivante pour obtenir un indicateur de la robustesse du jeu.

Début de solution

On utilise un panel de N stratégies, plus ou moins représentatives de l'espace de toutes les stratégies.

On fait un tournoi généralisé entre toutes ces stratégies, ce qui permet d'obtenir un classement général C_i^g de celles-ci.

Ensuite on crée un sous-ensemble de P stratégies sur lequel on réeffectue un tournoi, ce qui permet d'obtenir un nouveau classement. Chaque stratégie est attendue à une certaine position C_i^t dans ce classement en fonction de son classement général. Si la stratégie et le jeu sont complètement stables, le classement attendu C_i^t, et le classement réel C_i^r sur ce sous-ensemble devrait coincider. Ce n'est évidemment pas le cas.

On fait alors la somme des valeurs absolues de ces différences pour chaque stratégie du sous-ensemble, ce qui permet de savoir de combien de positions le jeu a fait évoluer les stratégies.

L'idéal serait alors, pour une taille de sous-ensembles donnée, de faire ce calcul sur tous les sous-ensembles possibles, ce qui n'est malheureusement pas envisageable, car on se trouverait dans ce cas devant une explosion combinatoire.

Par exemple, pour une population de 50 stratégies dans le tournoi de départ, et une taille de sous ensembles de 25, on se retrouve devant C_{50}^{25} sous-ensembles possibles, ce qui représente un nombre de l'ordre de 10^{14}.

La solution adoptée est donc est de générer aléatoirement M sous-tournois.

Un fois ces M sous-tournois effectués, on fait la moyenne des valeurs obtenues, ce qui permet d'obtenir une mesure de la robustesse du jeu :

$$R_{(P/N),M,L} = \frac{\sum_{i=1}^{i=M} \left(\sum_{j=1}^{j=P} \left| C_{i,j}^{t} - C_{i,j}^{r} \right| \right)}{M}$$

avec P le cardinal des sous-ensembles, N le nombre total de stratégies, M le nombre de mesures, L la longueur de chaque partie (nombre de coups dans une partie), et $C_{i,j}^{t}$, $C_{i,j}^{r}$ respectivement le classement attendu de la j^{e} stratégie, dans le i^{e} sous-ensemble, et le classement réel de cette même stratégie, dans ce même sous-ensemble.

Cette mesure de robustesse a un sens. Elle signifie qu'en moyenne une stratégie ne va pas changer son classement de plus de $R_{(P/N),M,L}$ positions.

3.1.2 Les résultats

Des évaluations ont été menées sur une population complète de 50 stratégies, parmi lesquelles se trouvent les 34 stratégies de base, et 16 stratégies générées aléatoirement grâce aux génotypes présentés dans le chapitre 4.

Lors de ces mesures chaque stratégie utilisant la fonction de genération aléatoire est jouée 8 fois, avant de se voir attribuer la moyenne des 8 scores obtenus pour ses matchs, afin d'aplanir les effets indésirables de l'utilisation du random.

Pour ces évaluations j'ai fait varié le nombre de coups par parties, ainsi que la taille des sous-ensembles. Le nombre de mesures étant lui fixé à $M = 1000$. Enfin ces mesures ont été faites pour deux jeux différents de scores.

Les résultats visibles sur les figures 3.1 et 3.2, mettent clairement en évidence que le jeu est assez robuste, puisqu'en moyenne, il ne modifie les positions des stratégies qu'au plus de 2,5 positions

De plus on peut s'apercevoir que plus le nombre de coups dans une partie est grand, plus le jeu est robuste.

Ces résultats bien qu'attendus sont importants, et inédits, et permettent de valider la poursuite des études des stratégies pour le Dilemme Itéré des Prisonniers.

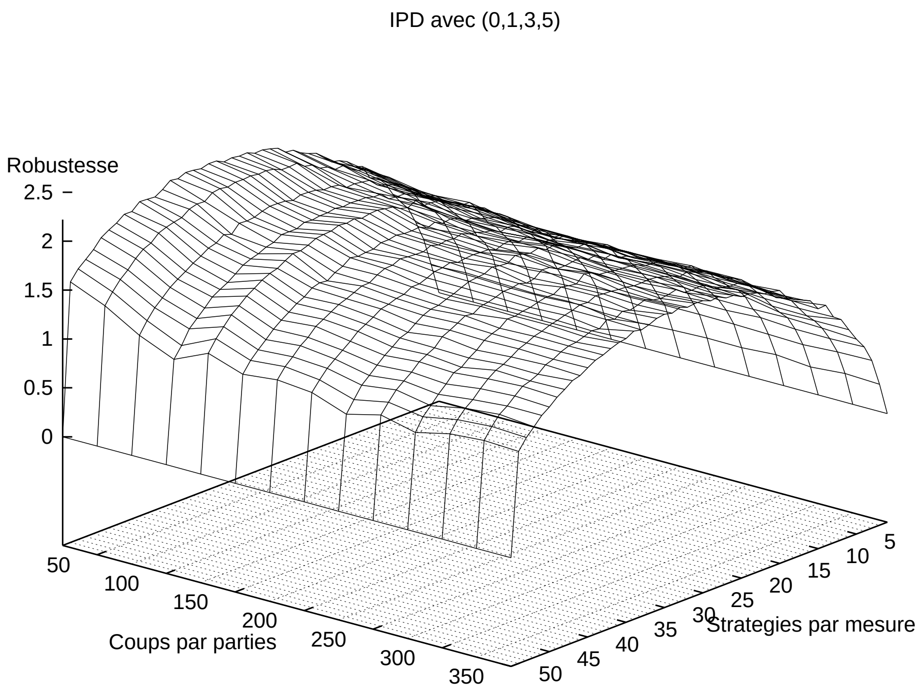

FIG. 3.1 - *Robustesse du DIP pour (0,1,3,5)*

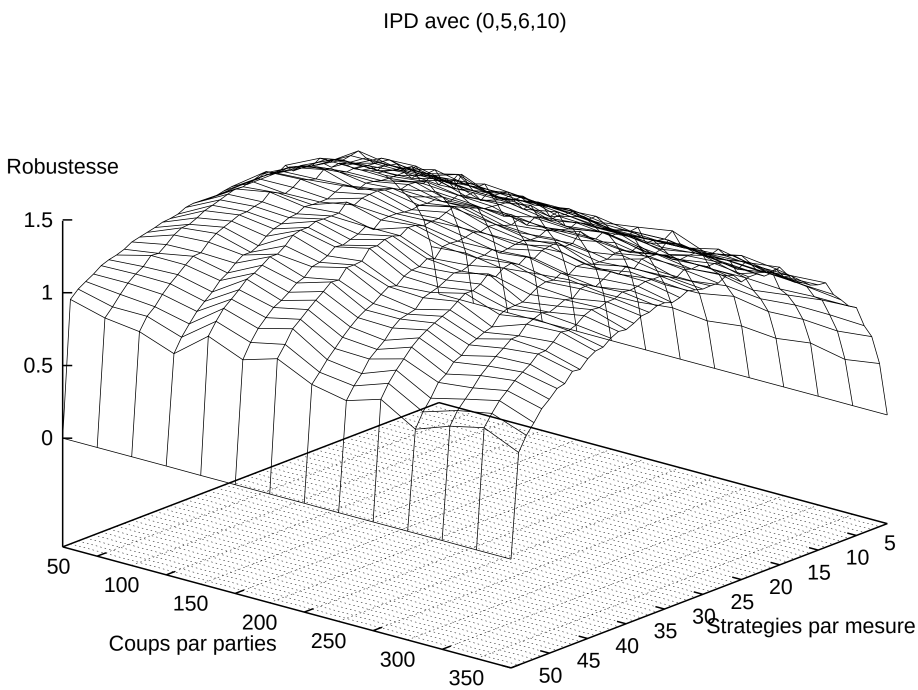

FIG. 3.2 - *Robustesse du DIP pour (0,5,6,10)*

3.2 Robustesse d'une stratégie

3.2.1 Les compétitions écologiques

Jusqu'à présent les deux seuls moyens de déterminer la qualité d'une stratégie sont son classement face à d'autres stratégies, ou son comportement en évolution écologique.

La compétition écologique, est un tournoi généralisé dans lequel, au départ chaque stratégie a une certaine population de représentants, qui va évoluer au fur et à mesure de son comportement dans les tournois, qui se répètent jusqu'à stabilisation de la population, *i.e.* elle ne change plus entre deux tournois.

On peut décrire l'algorithme d'une compétition écologique comme suit :

1. Toutes les stratégies ont la même proportion d'individus dans la population ;
2. Faire un tournoi entre toutes les stratégies de la population ;
3. Remplacer une proportion de la population de la stratégie la plus faible, par des individus de la stratégie la plus forte ;
4. Retourner au point 2 tant que la population ne s'est pas stabilisée.

C'est en outre cette méthode de mesure de la qualité d'une stratégie que l'on retrouve dans de nombreuses études, car elle permet de mettre en évidence la résistance de la stratégie à l'évolution.

3.2.2 Mesure d'une autre propriété

Suite à la mise au point de la mesure de robustesse du jeu, nous avons également mis au point une méthode qui permet de mesurer la robustesse aux sous-ensembles d'une stratégie. Cette mesure ne permet en rien d'affirmer la qualité d'une stratégie, en tant que comportement lors d'un tournoi, mais elle permet en contrepartie de savoir si une stratégie est bonne uniquement à cause de l'environnement dans lequel elle évolue, ou si plus généralement ses concepts lui permettent de se maintenir à une bonne place dans un tournoi contre d'autres stratégies.

La méthode utilisée pour mesurer la robustesse d'une stratégie est très similaire à celle que l'on a développée pour mesurer la robustesse du jeu dans la section précédente.

En fait il n'y a que deux différences entre ces deux méthodes :

- La stratégie évaluée est présente dans tous les sous-ensembles générés ;
- La mesure de la robustesse ne se fait pas sur la moyenne des sommes des valeurs absolues des différences de classement de toutes les stratégies participants aux sous-tournois, mais uniquement sur la moyenne des valeurs absolues des différences de classement de la stratégie étudiée.

L'idée étant la même que pour le cas précédent, je ne précise pas plus la méthode de calcul.

En revanche on peut dire que cette mesure sera utilisée pour vérifier que les stratégies obtenues par l'Algorithme Génetique sont robustes aux sous-ensembles.

3.2.3 Résultats

Afin d'illustrer cette mesure j'ai fait une série d'évaluation des stratégies de base, sur des parties de 250 coups, avec 6 recalculs pour les stratégies utilisant la génération de nombre aléatoire, et en faisant 1000 sous-ensembles. Dans ces évaluations seul la taille des sous-ensembles varie, et ce pour les scores classiques[1].

On peut entre autre noter sur la figure 3.3 l'extraordinaire robustesse de `graduel` parmi les stratégies de base.

La totalité de ces évaluations est consignée dans l'annexe B.

1. cf. table 2.1

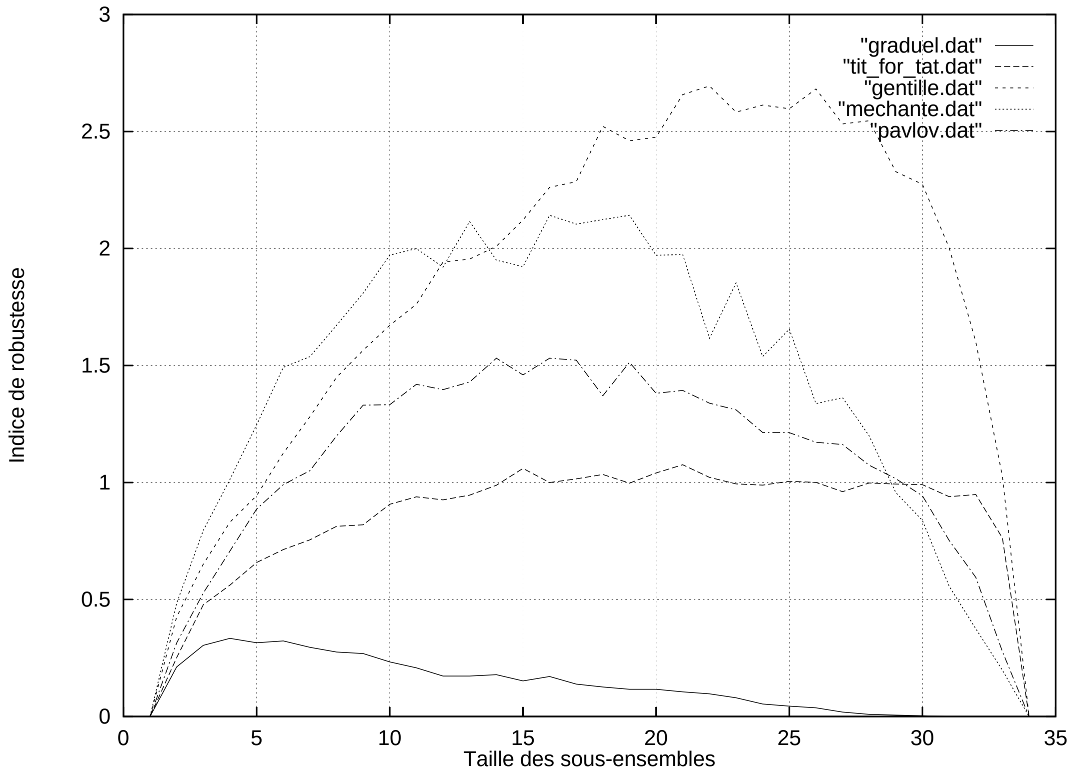

FIG. 3.3 - *Robustesse de quelques stratégies*

Chapitre 4

Les familles de stratégies

Après avoir étudié la bibliographie et testé les résultats établis, j'ai isolé, avec Jean-Paul DELAHAYE et Philippe MATHIEU, certaines classes de stratégies qui semblaient avoir des points communs parmi les stratégies classiques de base, énoncées dans la bibliographie et notamment dans [DM92], [Axe92] et [NS93], et qui sont énumérées dans l'annexe A.

Le but de cette classification est non seulement de pouvoir définir les génotypes de stratégies qui seront utilisés par l'Algorithme Génétique lors des expériences d'évolution, mais aussi de pouvoir définir de manière la plus générale possible l'espace des stratégies que l'on souhaite explorer.

4.1 Les probabilistes

Ce sont des stratégies qui coopèrent, ou trahissent avec une certaine probabilité.

Ces stratégies sont identifiées par trois paramètres, ou gènes [1] :

normal, la réponse normale de la stratégie ;

numerateur, le numérateur de la probabilité ;

denominateur, le dénominateur de la probabilité.

On a bien évidemment **numerateur<denominateur**. Ces stratégies joue le coup **normal** avec la probabilité :

$$p = 1 - \frac{\textbf{numerateur}}{\textbf{denominateur}}$$

.

4.2 Les majoritaires

Ce sont des stratégies *bienveillantes* qui essaient de s'adapter aux comportements de leurs adversaires, les punissant, *i.e.* jouant un certain nombre de fois `TRAHIR`, quand ceux-ci ont plus souvent trahis que coopérés dans le passé.

Ces stratégies sont identifiées par huit gènes :

debut, le premier coup à jouer ;

vision, le nombre de coups du passé que la stratégie retient (0 signifiant que la stratégie peut utiliser les coups joués depuis le début de la partie) ;

1. Dans la suite nous utiliserons plutôt le terme gène

punition, la longueur d'une punition, *i.e.* le nombre de coups pendant lesquels la stratégie joue `TRAHIR` en représailles à un mauvais comportement de l'adversaire ;

acalmie, la longueur d'une acalmie, après une punition, *i.e.* le nombre de coups pendant lesquels la stratégie joue `COOPÉRER` après avoir puni son adversaire ;

$\boldsymbol{A_t}$ et $\boldsymbol{B_t}$, les coefficients du nombre de trahisons de l'adversaire dans la décision du déclenchement d'une punition (cf. description du comportement de la stratégie plus loin) ;

$\boldsymbol{A_c}$ et $\boldsymbol{B_c}$, les coefficients du nombre de coopérations de l'adversaire dans la décision du déclenchement d'une punition (cf. description du comportemet de la stratégie plus loin).

Pour pouvoir jouer, ces stratégies maintiennent, à chaque pas d'une partie, deux indicateurs, N_T et N_C, qui correspondent respectivement aux nombres de trahisons et de coopérations de l'adversaire, au cours du passé visible par la stratégie.

Ces stratégies entament une période punition–acalmie si :

$$A_T N_T + B_T \geq A_C N_C + B_C$$

et coopèrent sinon.

4.3 Les analytiques

Ce sont des stratégies qui essaient d'analyser le comportement de leurs adversaires, afin d'essayer de les exploiter, *i.e.* les trahir quand ils coopèrent, si ceux-ci ne semblent pas trop réagir aux trahisons. Afin de savoir si leurs adversaires sont exploitables ou pas, elles commencent par une série prédéfinie de coups, qui leur permettent d'amorcer leurs indicateurs. Si l'adversaire semble réactif alors elle joue ce que `tit_for_tat` aurait joué.

Ces stratégies sont identifiées par cinq gènes :

longueur_amorce, le nombre de coups que comporte l'amorce ;

amorce, la suite de coups à jouer dans les **longueur_amorce** premiers coups d'une partie ;

vision, le nombre de coups du passé que la stratégie retient (0 signifiant que la stratégie peut utiliser les coups joués depuis le début de la partie) ;

punition, la longueur d'une punition, *i.e.* le nombre de coups pendant lesquels la stratégie joue `TRAHIR` pour exploiter son adversaire peu réactif ;

acalmie, la longueur d'une acalmie, après une punition, *i.e.* le nombre de coups pendant lesquels la stratégie joue `COOPÉRER` après avoir puni son adversaire.

Pour pouvoir jouer ces stratégies maintiennent, à chaque pas d'une partie, quatre indicateurs, N_{TT} N_{TC} N_{CC} et N_{CT}, qui correspondent respectivement au nombre de trahisons de l'adversaire après une trahison de la stratégie, nombre de coopération de l'adversaire après une trahison de la stratégie, nombre de coopération de l'adversaire après une coopération de la stratégie et nombre de trahisons de l'adversaire après une coopération de la stratégie, le tout étant évalué dans le passé visible de la stratégie.

Pendant les **longueur_amorce** premiers coups de la partie ces stratégies jouent les coups de leur **amorce**, tout en maintenant leurs indicateurs. Ensuite, si

$$N_{CT} \geq N_{CC}$$

ce qui signifie que l'adversaire a plus profité de la stratégie que l'inverse, alors elles jouent ce que `tit_for_tat` aurait joué, sinon si

$$N_{TC} \geq N_{TT}$$

ce qui signifie que l'adversaire semble peu réactif, alors elles entament une période punition–acalmie, sinon elles coopèrent dans les autres cas.

4.4 Les super-graduelles

4.4.1 La stratégie `graduel`

La stratégie **`graduel`**, introduite dans [DM92], est une stratégie qui ressemble à **`tit_for_tat`** ormis le fait qu'au lieu de trahir une seule fois en guise de punition, elle trahit de plus en plus [2]. Comme cette stratégie s'avère être la plus efficace en tournoi simple, ainsi qu'en compétition écologique dans toutes les expérimentations faites jusqu'à présent, nous avons essayé de généraliser ses principes en créant une famille de stratégie s'approchant de celle-ci.

Les principales qualités de **`graduel`** sont sa réactivité, sa bienveillance, et son indulgence, mais sa principale force vient du fait qu'elle oblige en quelque sorte son adversaire à coopérer, en l'empêchant de trahir trop souvent, ou en tout état de cause en se laissant de moins en moins exploiter avec le temps.

4.4.2 Les extensions à `graduel`

Nous avons apporté quelques extensions à **`graduel`** afin d'essayer de généraliser ses idées, et de décrire ainsi un grand nombre de stratégies existantes :

- la possibilité de faire varier la taille de son temps de réaction, la longueur de la punition, et la longueur de l'accalmie en fonction du nombre de punitions infligées ;
- la possibilité de prendre ou non en compte les trahisons de l'adversaire pendant une période de réaction–punition–accalmie ;
- la possibilité de déclencher une punition en fonction du nombre de trahisons de l'adversaire ou d'une différence de scores entre la stratégie et l'adversaire, dans le passé visible par la stratégie ;
- la possibilité de pardonner, *i.e.* de ne pas lancer une punition de temps en temps ;
- la possibilité de trahir de temps en temps, *i.e.* de ne plus être bienveillante.

Il est bien évident que tous ces ajouts permettent encore de décrire la stratégie **`graduel`**, qui devient une espèce particulière de super-graduelle.

4.4.3 Description du génotype des super-graduelles

Ces stratégies sont identifiées par dix-neufs gènes :

debut, le premier coup à jouer ;

alea, le nombre utilisé pour déterminer les coups aléatoires où la stratégie trahit ;

type_calcul, le type de détection faite pour le déclenchement d'une punition, *i.e.* en fonction du nombre de trahisons, ou en fonction d'une différence de scores ;

le seuil, le seuil que l'adversaire ne doit pas dépasser avant de se voir infliger une punition ;

generosite, le nombre utilisé pour déterminer les coups aléatoires où la stratégie est clémente, et pardonne, au lieu de déclencher une punition ;

aveugle, la possibilité ou non de compter les défections de l'adversaire pendant un punition ;

vision, le nombre de coups du passé que la stratégie retient (0 signifiant que la stratégie peut utiliser les coups joués depuis le début de la partie) ;

2. Dans sa version de base elle trahit autant de fois que l'adversaire l'a déjà trahi dans le passé

evolution_punition, le type de l'évolution de la durée d'une punition, qui peut être polynomial ou logarithmique ;

$\boldsymbol{A_p}$, $\boldsymbol{B_p}$ et $\boldsymbol{C_p}$, les coefficients de l'évolution de la durée d'une punition ;

evolution_reaction, le type de l'évolution du temps de réaction avant le déclenchement d'une punition, qui peut être polynomial ou logarithmique ;

$\boldsymbol{A_r}$, $\boldsymbol{B_r}$ et $\boldsymbol{C_r}$, les coefficients de l'évolution du temps de réaction ;

evolution_accalmie, le type de l'évolution de la durée d'une accalmie après une punition, qui peut être polynomial ou logarithmique ;

$\boldsymbol{A_a}$, $\boldsymbol{B_a}$ et $\boldsymbol{C_a}$, les coefficients de l'évolution de la durée d'accalmie.

Ces stratégies trahissent avec une probabilité $p = 1/$**alea**. Si elles n'ont pas trahi, alors elles décident si une période réaction–punition–accalmie doit être déclenchée, en fonction de **type_calcul** et de **seuil**, durant le passé visible par la stratégie. Si une telle période doit être déclenchée, elle ne la commence qu'avec une probabilité $p = 1 - 1/$**generosite**. Les paramètres de cette période, *i.e.* temps de réaction, durée de punition, et durée d'accalmie sont calculés en fonction du nombre de punitions déjà effectuées (soit N) d'une des deux manières suivantes en fonction, respectivement, de **evolution_reaction**, **evolution_punition** et **evolution_accalmie** : s'il s'agit d'un calcul polynômial la valeur sera

$$A_i N^2 + B_i N + C_i$$

et s'il s'agit d'un calcul logarithmique la valeur sera

$$A_i \frac{\log(N)}{\log(B_i)} + C_i \quad (B_i > 1)$$

avec $i \in \{r, p, a\}$ et N mis à jour à chaque coup si la stratégie n'est pas aveugle.

4.5 Les parasites

Ce sont des stratégies qui en essaient d'autres pendant une certaine durée, puis au bout de la période d'essai, décide de jouer celle qui a rapportée le plus gros score. Seules les stratégies de base peuvent être parasitées.

Ces stratégies sont identifiées par trois gènes :

nombre, le nombre de stratégies parasitées ;

parasitees, les stratégies parasitées ;

duree_d_essai, la durée d'essai d'une stratégie.

Ces stratégies essaient chacune des stratégies parasitées, pendant une période longue de **duree_d_essai** coups, puis une fois que toutes ont été testées, elles choisissent de devenir la stratégie ayant fait le plus gros score.

4.6 Les automates

4.6.1 Description du génotype des automates

Ce sont des stratégies qui agissent comme un automate d'états, chaque état représentant une des stratégies de base, toutes les transitions entre états étant spécifiées. Quand la stratégie arrive sur un état elle joue le coup que la stratégie de cet état aurait joué.

Ces stratégies sont identifiées par cinq gènes :

nombre_etats, le nombre d'états de l'automate ;

etats, les stratégies composant les états de l'automate ;

transitions, les transitions entre chaucun des états ;

etat_depart, l'état de départ ;

vision, le nombre de coups du passé que la stratégie retient pour l'évaluation des transitions (0 signifiant que la stratégie peut utiliser les coups joués depuis le début de la partie) ;

Elles fonctionnent exactement comme un automate classique, en débutant sur l'état **etat_depart**, et en jouant le coup que la stratégie de l'état jouerait.

4.6.2 Les transitions entre états

Pour décrire les transitions un *micro*-langage logique a été adopté. Il permet de décrire 3 types de transitions en fonction du nombre de trahisons, et de coopérations de l'adversaire dans le passé visible de la stratégie, *i.e.* en se limitant aux **vision** derniers coups.

Ces transitions sont identifiées par 3 trois paramètres :

type, qui spécifie le type de la transition ;

A et ***B***, les coefficients utilisés pour les tests de transitions.

Soit donc N_T le nombre de trahisons de l'adversaire, et N_C le nombre de coopérations de l'adversaire durant les **vision** derniers coups si **vision** n'est pas nul, ou depuis le début de la partie si **vision** est nul. Les trois types de transitions sont alors les suivantes :

transitions ET La transition est acceptée si $N_T > A$ et $N_C > B$;

transitions OU La transition est acceptée si $N_T > A$ ou $N_C > B$;

transitions COMPARAISON La transition est acceptée si $AN_T > BN_C$.

On peut avec ce langage décrire un nombre de transitions qui nous semble suffisament grand.

Voici un exemple de stratégie automate, qui correspond à la stratégie de base `rancuniere`, qui a pour génotype :

2, (`gentille`, `méchante`), ($\emptyset$, (ET, 0, 0), (COMP, 0, 0), $\emptyset$), 0, 0

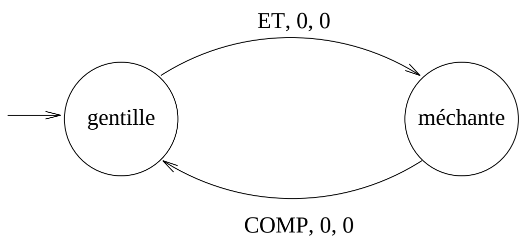

FIG. 4.1 - *Exemple de stratégie automate*

Chapitre 5

Les Algorithmes Génétiques

Les Algorithmes Génétiques sont des algorithmes qui permettent, en se basant sur des idées issues de la biologie, d'optimiser des fonctions, ou plus généralement de faire une recherche rapide, intelligente, et non-exhaustive d'un individu dans un espace de recherche assez vaste. C'est cette approche exploratoire qui est décrite dans ce chapitre, et que j'ai utilisé en relation avec le chapitre 4, pour rechercher des bonnes stratégies pour le Dilemme Itéré des Prisonniers.

5.1 Présentation des Algorithmes Génétiques

Les Algorithmes Génétiques font partie d'une classe d'algorithmes, que l'on appelle les algorithmes évolutifs [1], qui sont trés souvent utilisés en Vie Artificielle. Leur approche de la résolution d'un problème est une approche exploratoire et évolutionniste.

Comme tout algorithme d'optimisation ils ont leurs inconvénients, et leurs avantages. Je n'exposerai ici que la partie des Algorithmes Génétiques qui m'a intéressé dans le cadre de l'exploration de l'espace des stratégies, ce n'est donc qu'un type d'Algorithme Génétique qui est décrit.

5.1.1 Principe d'un Algorithme Génétique

Pour explorer un monde, ou le domaine d'application d'une fonction, il faut pouvoir le décrire, et le parcourir.

Pour décrire l'espace de recherche les Algorithmes Génétiques utilise un codage de la connaissance similaire à celui que l'on trouve en biologie, à savoir une chaîne de bits, ou génotype, dont chaque élément renferme une part de l'information d'un individu.

Pour parcourir cet espace on dispose d'opérations génétiques sur les individus de cet espace. Ces opérations classiques en génétique sont le croisement, ou *crossing-over*, et la mutation.

Lorsque l'on veut explorer un monde c'est que l'on cherche quelque chose de particulier, un individu singulier, qui a certaines caractéristiques, ou une valeur particulière lorsqu'il s'agit d'une fonction. On a donc besoin d'un outil qui permet de nous dire si un individu est bien l'individu recherché, ou si ses caractéristiques se rapprochent ou non, de celles que l'on recherche. Dans les Algorithmes Génétiques cet outil c'est le facteur d'adaptation, ou *fitness*.

1. cf. [Pre94a]

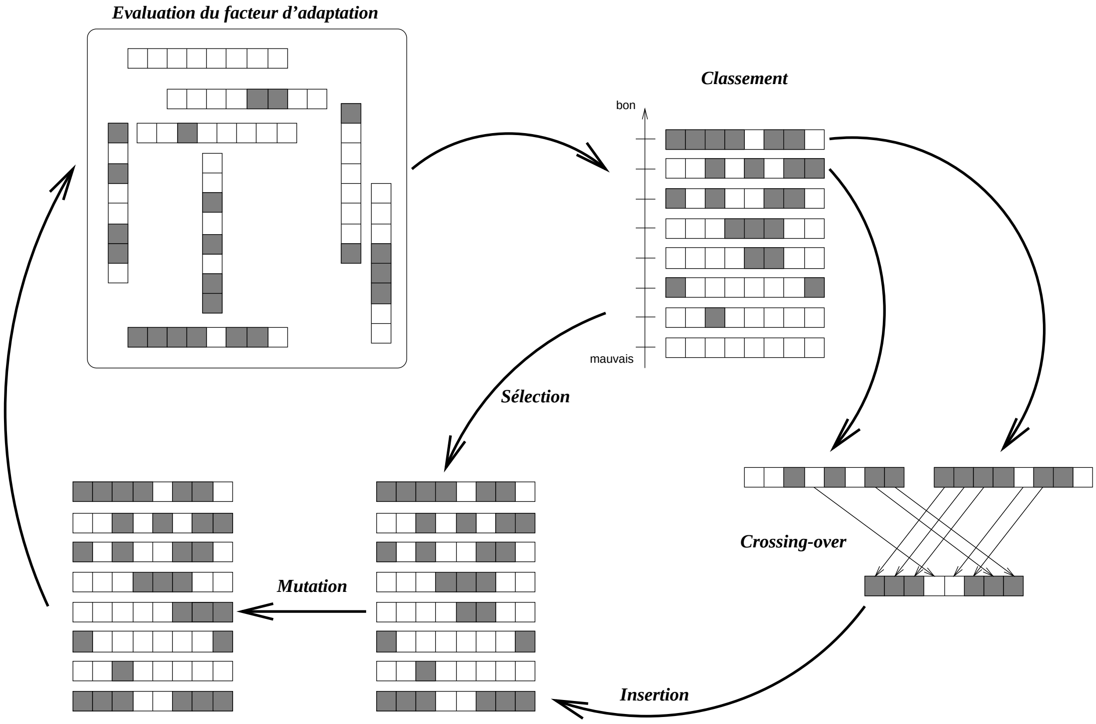

FIG. 5.1 - *Le schéma de fonctionnement d'un Algorithme Génétique*

Pour rechercher un individu il suffit alors de suivre un algorithme qui peut se résoudre à ces quelques lignes :

`Début`

Générer une population aléatoire d'individus

Évaluer le facteur d'adaptation de chaque individu

`Tant que` individu pas trouvé `Faire`

Remplacer les individus au plus faible facteur d'adaptation par des individus meilleurs
Remplacer quelques individus par de nouveaux, générés aléatoirement
Évaluer le facteur d'adaptation de chaque individu

`Fait`

`Fin`

Les Algorithmes Génétiques ont été montré efficaces, et sont basés sur le théorème des schémas, cf. [Pre94b] et [BBM93].

5.1.2 Description d'un Algorithme Génétique

L'idée toute simple est qu'il suffit de se positionner dans l'espace puis de le parcourir. On peut voir sur la figure 5.1 le schéma de fonctionnement d'un algorithme génétique, qui reprend les idées énoncées plus haut.

Voici une explication chronologique des différentes étapes de cet Algorithme Génétique:

1. **Évaluation du facteur d'adaptation**, chaque individu voit son facteur d'adaptation calculé ;

2. **Classement**, on effectue un classement des individus en fonction de leur facteur d'adaptation ;
3. **Sélection**, au cours de cette étape, les plus mauvais individus sont supprimés de la population ;
4. **Crossing-Over**, on remplace les individus par des individus crées par croisement de gènes d'individus bien classés ;
5. **Mutation**, au cours de cette étape tous les indvidus de la nouvelle population sont mutés, certains de leurs gènes sont modifiés, avec une certaine probabilité, classiquement 2%.

Après cette étape le cycle recommence, jusqu'à ce que l'on arrête le processus, parce que tous les individus ont le même génotype, et c'est la convergence, ou parce que l'individu recherché a été trouvé.

5.1.3 Difficultés des Algorithmes Génétiques

Les principales difficultés que l'on rencontre lors de l'utilisation d'un Algorithme Génétique sont le codage d'un génotype, le choix de la fonction d'évaluation du facteur d'adaptation, et enfin le paramétrage de l'évolution, *i.e.* le nombre d'individus à remplacer à chaque cycle.

C'est ce genre de problème qu'il a fallu résoudre pour les appliquer aux stratégies du Dilemme Itéré des Prisonniers.

5.2 Application au Dilemme Itéré des Prisonniers

Parmi les trois problèmes précédents, deux ont pu être résolus facilement, mais d'autres sont apparus. Pour ce qui est du choix du codage, il a été fait dans le chapitre 4, alors que le choix de la fonction d'évaluation s'est tout naturellement porté sur le classement des stratégies dans un tournoi.

5.2.1 Limitations

Pour ce qui est qui du choix du nombre d'individus à remplacer par cycle, les expériences ont été faites avec plusieurs valeurs, chacune apportant un résultat différent.

La plus grosse limitation, est venue de la génération aléatoire des stratégies, et des mutations. En effet une première série d'expériences sans limitations a conduit à la génération d'individus beaucoup trop dispersés pour pouvoir donner des résultats intéressants.

Les générateurs de stratégie ont donc été limités de façon à réduire l'espace de départ, et donc du même coup l'espace de recherche. De plus ces limitations sont arbitraires, et entièrement dépendantes de mon jugement, ce qui peut créer une limitation encore plus forte, mes choix se bornant aux valeurs classiquement utilisées dans la littérature.

5.2.2 Premiers Résultats

La liste complète des résultats se trouve en annexe C.

Je ne cite ici que quelques uns des résultats les plus marquants.

Les expériences menées l'ont été sur chaque famille de stratégies, en adoptant comme fitness, le classement des stratégies dans un tournoi entre tous les individus de la famille, et les stratégies de base, et ce avec différents taux de renouvellement de la population : 5, 10 ou 15 renouvellements par cycle. Le nombre de coups par parties est de 200, et le nombre de recalcul des stratégies utilisant le hasard de 4.

Le premier de ces résultats est une surprise, puisque dans presque tous les cas la famille probabiliste converge vers la stratégie `méchante`, et ce d'une manière fort simple, puisque le numérateur converge en quelques cycles vers 0, alors que la carte normale à jouer devient rapidement **TRAHIR**. On peut expliquer ce phénomène par l'incapacité de la stratégie à faire mieux que **méchante** qui n'est pas forcément une mauvaise stratégie.

Une autre remarque importante est que plus le taux de renouvellement est bas, moins vite l'algorithme converge.

Environ la moitié des stratégies déterminées par ces convergences sont, au bout du compte, meilleures que **tit_for_tat** dans un tournoi contre les stratégies de base, mais aucune n'arrive pour l'instant à battre la stratégie **graduel**, le plus troublant étant le fait que même la famille de stratégie **super_raduelles** ne converge pas vers celle-là. On peut cependant sans doute expliquer ce fait par l'introduction du paramètre **alea**, qui fait perdre une des qualités de **graduel**, à savoir la bienveillance.

La plupart des bonnes stratégies découvertes, ont non seulement une robustesse très forte, mais également un excellent comportement en compétition écologique. Ce dernier point étant facilement compréhensible, puisque ces stratégies ont été *conditionnées* à jouer contre elle même, du fait du choix du fitness.

Voici une des meilleures stratégies crée, c'est une automate dont le génotype est :

(3, (**graduel**, **sondeur4**, **mefiant**), (∅, (ET, 4, 28), (COMP, 49, 50), (COMP, 76, 2), ∅, (COMP, 44, 19), (ET, 31, 60), (OU, 37, 23), ∅), 0, 20)

Elle se classe seconde, derrière **graduel** dans un tournoi comprenant les stratégies de base, et elle, a une robustesse de 0,46, pour 250 sous-ensembles de 17 stratégies sur 35, et est apparue après convergence de l'algorithme pour des taux de renouvellement de 10 et 15 individus, en 25 cycles.

De plus en compétition écologique elle arrive seconde, une fois que la population s'est stabilisée.

5.2.3 Extensions et modifications

Globalement les résultats restent pour l'instant plutôt décevant, mais cette déception est quasiment à chaque fois explicable.

De plus des améliorations sont en cours pour atteindre des stratégies meilleures que **graduel**, comme par exemple la modification du fitness pour prendre en compte une mesure de robustesse de l'individu, et le classement de l'individu dans un tournoi contre uniquement les stratégies de base.

Cette dernière idée est particulièrement intéressante, car elle ne peut donner naissance qu'à des stratégies avec de multiples qualités, chaque élément du facteur d'adaptation en favorisant une, la robustesse écologique, la robustesse aux sous-ensembles, le classement face aux stratégies de base, ...

Le seul inconvénient de cette idée étant de compliquer le calcul de l'adaptation, et donc de réduire la vitesse de fonctionnement de l'algorithme, en demandant beaucoup plus de temps de calcul.

Chapitre 6

Conclusions et Perspectives

La plus grande partie des objectifs du sujet a été atteinte, même si les principaux, restent encore décevants, notamment la convergence de l'algorithme vers des stratégies certes meilleures que **tit_for_tat** mais moins bonnes que **graduel**.

Cependant ces différents travaux tendent à conforter l'idée que la simplicité n'est pas une bonne qualité pour les stratégies, contrairement aux idées d'AXELROD. Seul la poursuite de telles expérimentations pourra montrer que cette idée est fondée.

Les outils de mesure des stratégies, et les résultats sur la robustesse du Dilemme Itéré des Prisonniers sont à eux seuls une nouveauté importante, que l'on envisage d'utiliser pour créer un test permanent d'évaluation (benchmark publique) de stratégie, dont le protocole reste encore à définir, et à étayer de quelques résultats et mesures plus nombreux, qui pourront être fait dans un travail de début de thèse.

L'ouverture des expérimentations à des compétences différentes en matière de stratégies ne pouvant qu'être bénéfique à l'évolution des expérimentations.

Ce projet pourra de surcroît faire connaître les travaux effectués au Laboratoire sur le Dilemme Itéré des Prisonniers et suscitera peut-être l'intérêt de certains pour le domaine de la Vie artificielle.

D'autres directions de poursuite de ce travail sont d'ores et déjà en cours, comme par exemple la mise au point d'un facteur d'adaptation plus performant. Cette amélioration entraînerait à coup sûr des modifications du programme utilisé afin de pouvoir optimiser les calculs de ce facteur d'adaptation, en tentant de les paralléliser par exemple.

Une direction à ne pas négliger pourrait être la programmation génétique, qui bien que complexe peut sans doute apporter une nouvelle manière de rechercher de bonnes stratégies.

La Vie Artificielle a enfin bien été approchée, à travers le Dilemme Itéré des Prisonniers, qui semble être un sujet majeur du domaine.

Annexe A

Les stratégies de base

Les stratégies de base utilisées aussi bien dans les expériences de robustesse, que d'évolution avec les Algorithmes Génétiques sont décrites ici. Elles sont issues des différents articles, livres et autres rapports d'études cités dans la bibliographie.

1. **gentille** : toujours coopérer ;
2. **mechante** : toujours trahir ;
3. **lunatique** : COOPÉRER ou TRAHIR avec $p(\text{COOPÉRER}) = 1/2$;
4. **tit_for_tat** : commence par coopérer puis recopie les coups de l'adversaire ;
5. **rancuniere** : coopère jusqu'à ce que l'adversaire trahisse une fois après quoi elle trahit tout le temps ;
6. **per_mechante** : joue périodiquement **ttcttc**. . . ;
7. **per_gentille** : joue périodiquement **cctcct**. . . ;
8. **majo_mou** : coopérer puis jouer le coup majoritaire de l'adversaire, si égal alors coopérer ;
9. **mefiant** : comme **tit_for_tat**, mais commence par TRAHIR ;
10. **majo_dur** : coopérer puis jouer le coup majoritaire de l'adversaire, si égal alors trahir ;
11. **sondeur** : joue **tcc**, si aux coups 2 et 3 l'adversaire répond **cc** je trahis toujours, sinon je joue comme **tit_for_tat** ;
12. **tft_dur** : coopérer sauf si l'adversaire a trahi l'un des deux derniers coups ;
13. **graduel** : commence par coopérer, puis après le 1^er^ TRAHIR trahi une fois, . . . , puis apès le n^{e} trahir trahir n fois ;
14. **tf2t_dur** : toujours coopérer sauf si dans les 3 coups précédents l'adversaire a trahi 2 fois de suite ;
15. **tf2t_mou** : si l'adversaire trahi 2 fois consécutives, trahir ;
16. **joss_dur** : **tit_for_tat** avec en plus une trahison avec $p(\text{TRAHIR}) = 1/10$;
17. **joss_mou** : **tit_for_tat** avec en plus une coopération à la place d'une trahison avec $p(\text{COOPÉRER}) = 1/10$;
18. **coop_puis_tc** : coopérer jusqu'à une trahison, puis $(tc)\star$;
19. **binaire** : $(ct)\star$;

20. `ccctct` : $(ccctct)\star$;
21. `c_4_sur_5` : coopérer ou trahir avec $p(\texttt{COOPÉRER}) = 4/5$;
22. `quatre_c_un_t` : $(cccct)\star$;
23. `calculateur` : `joss_dur` les 20 premiers coups, puis si période trahir toujours, sinon `tit_for_tat`;
24. `sondeur2` : jouer `tcc`, si aux coups 2 et 3 l'adversaire répond `tc` je coopère toujours, sinon `tit_for_tat`;
25. `sondeur3` : trahir et coopérer puis si l'adversaire a coopéré en 2 toujours trahir sinon `tit_for_tat`;
26. `sondeur4` : `cctctttcctctcctcttct` puis ... ;
27. `sondeur_dur` : trahir les 2 premiers coups puis coopérer 2 coups, si l'adversaire a coopéré aux coups 2 et 3 toujours trahir, sinon `tit_for_tat`;
28. `mieux_en_mieux` : trahir avec une probabilité de $(1000-n)/1000$, donc de moins en moins ;
29. `pire_en_pire` : trahir avec une probabilité de $n/1000$, donc de plus en plus ;
30. `pire_en_pire2` : `tit_for_tat` pendant 20 coups puis trahir avec $p(\texttt{TRAHIR}) = ((n-20)/n)$;
31. `pire_en_pire3` : coopérer puis plus l'adversaire trahit, plus je trahis (au coup $n+1$ avec une probabilité de $nb(\texttt{TRAHIR})/n$;
32. `doubleur` : toujours coopérer sauf après une trahison et que le nombre des coopérations de l'adversaire est inférieur au double de ses trahisons ;
33. `ranc_mou` : coopérer sauf après chaque trahison où on trahit 4 fois coopérer 2 fois avant de reprendre ;
34. `pavlov` : commence par coopérer, puis si au coup précédent les deux joueurs ont joué le même coup continuer à jouer ce coup, sinon joue le coup contraire (*win-stay/lose-shift*).

Ce qui fait un total de 34 stratégies de base.

Annexe B

Robustesse des stratégies de base

Les mesures suivantes ont été effectuées sur des DEC-ALPHA, avec 1000 mesures, 6 recalculs des stratégies utilisant le générateur aléatoire, et sur les 34 stratégies de base présentées dans l'annexe A.

Les résultats sont présentés sous forme de 7 graphiques, comportant 5 courbes chacun, dans un souci de lisibilité de ces mêmes courbes.

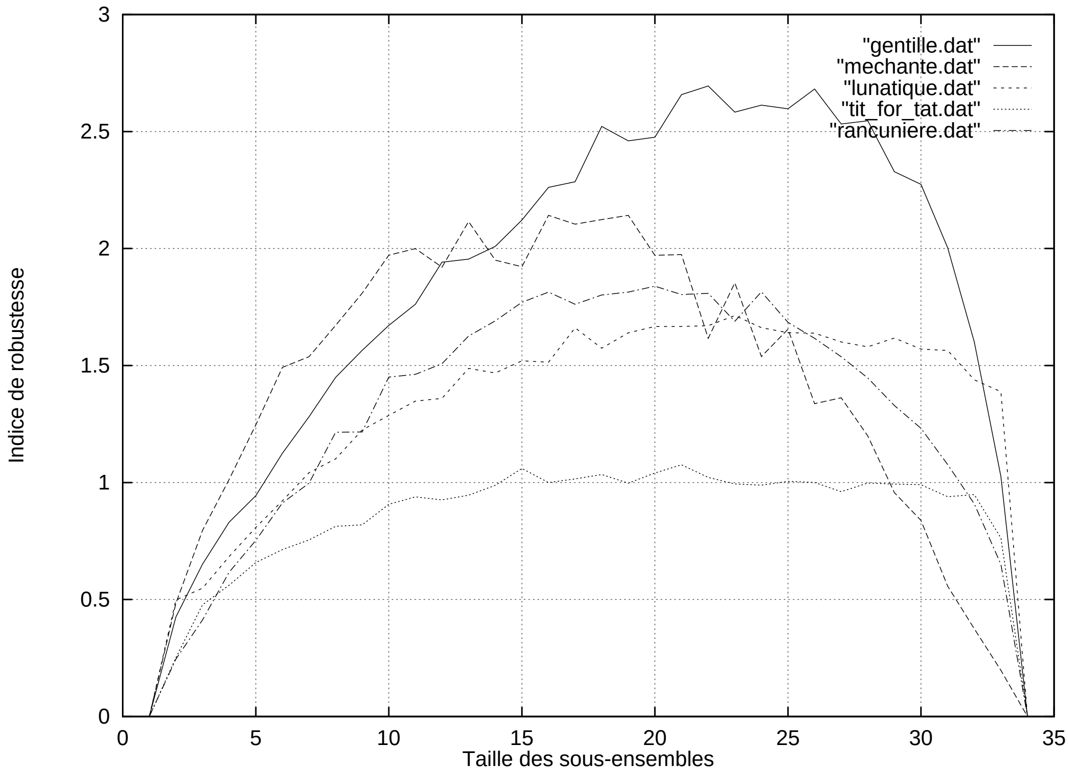

FIG. B.1 - *Robustesse des 5 premières stratégies de base*

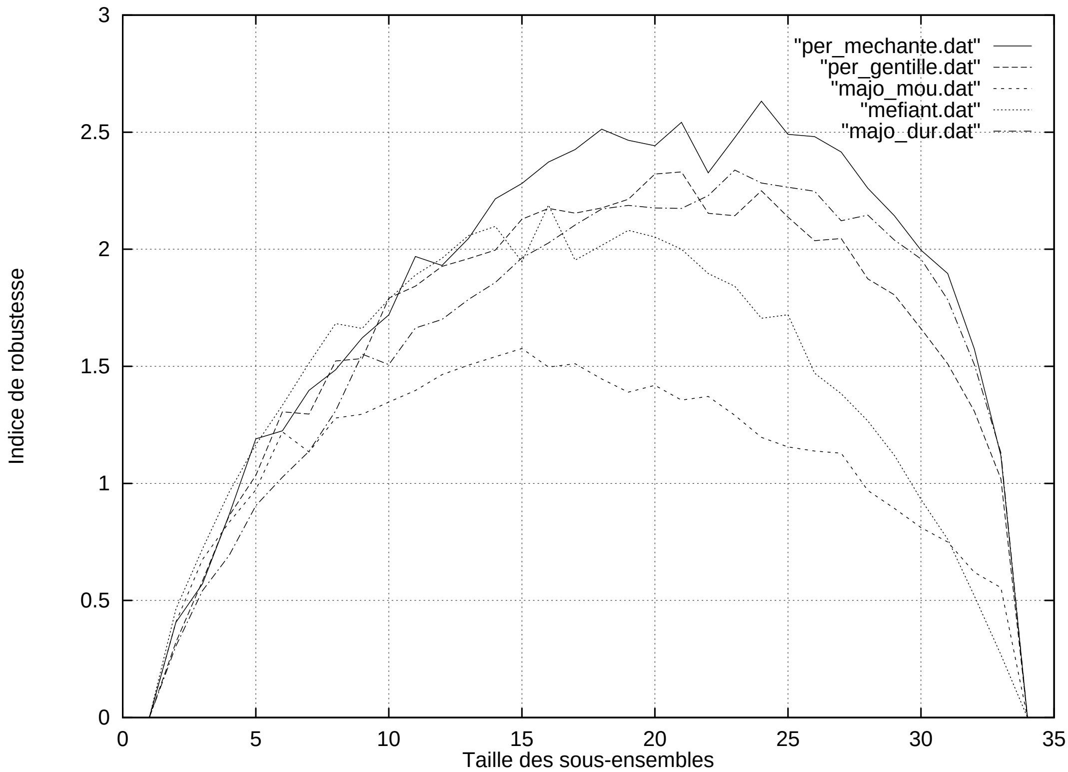

FIG. B.2 - *Robustesse des 6 à 10es stratégies de base*

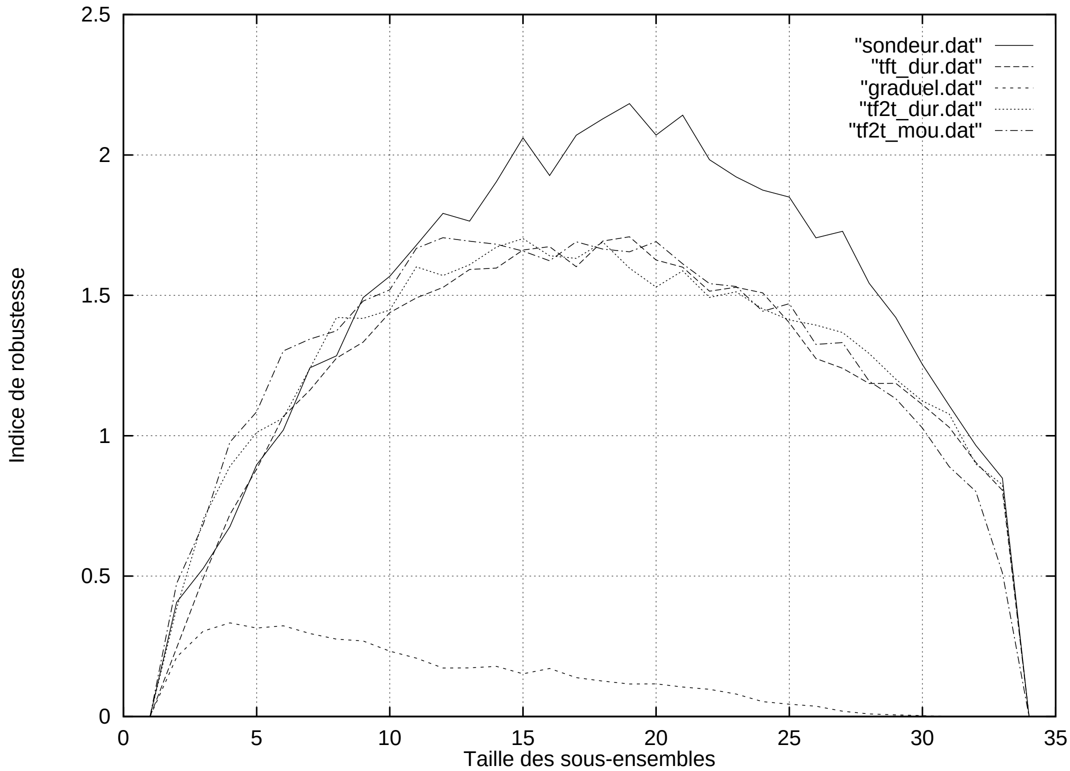

FIG. B.3 - *Robustesse des 11 à 15es stratégies de base*

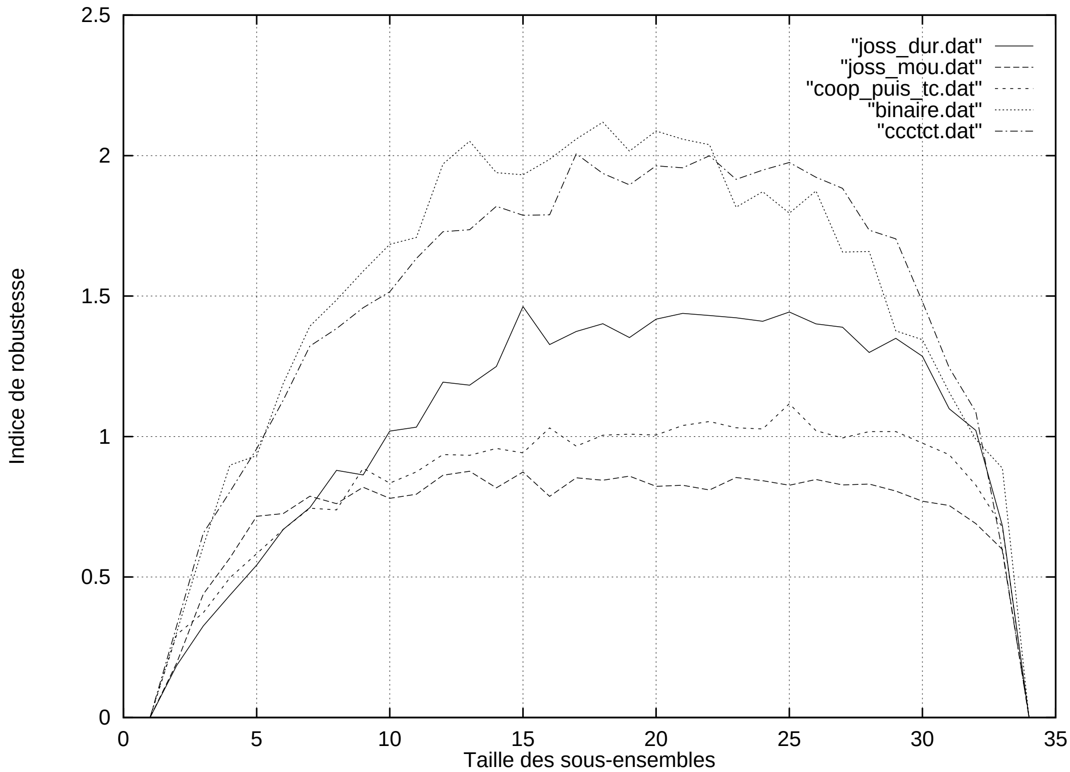

FIG. B.4 - *Robustesse des 16 à 20es stratégies de base*

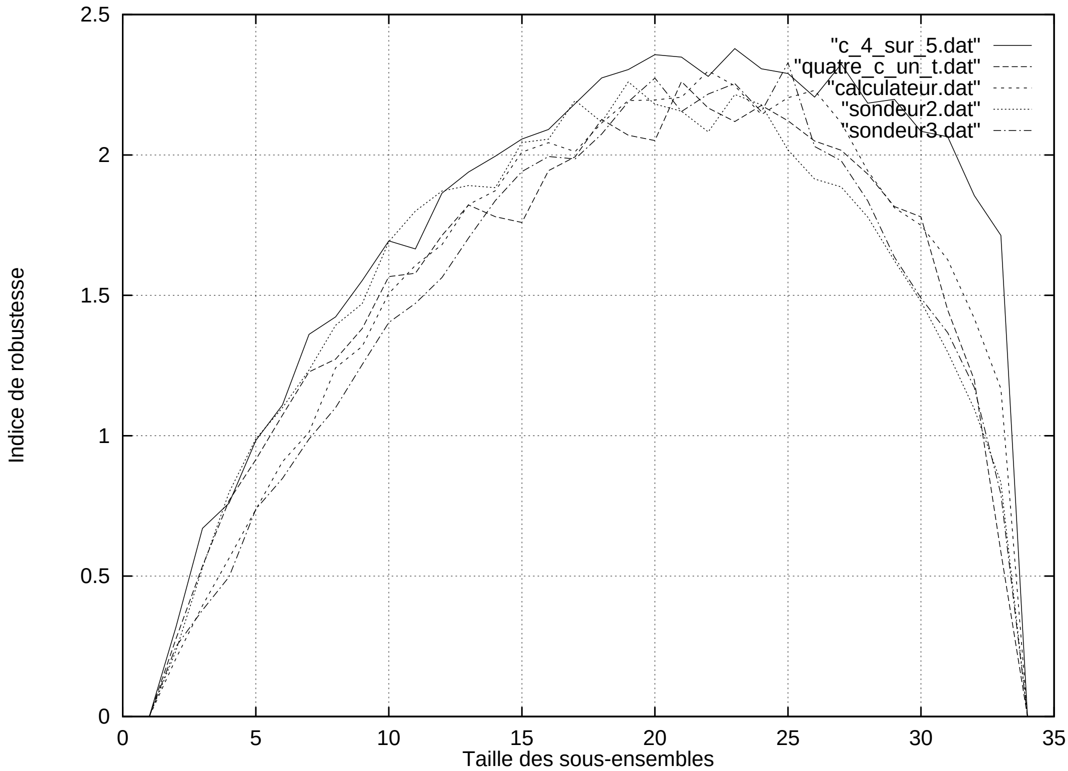

FIG. B.5 - *Robustesse des 21 à 25es stratégies de base*

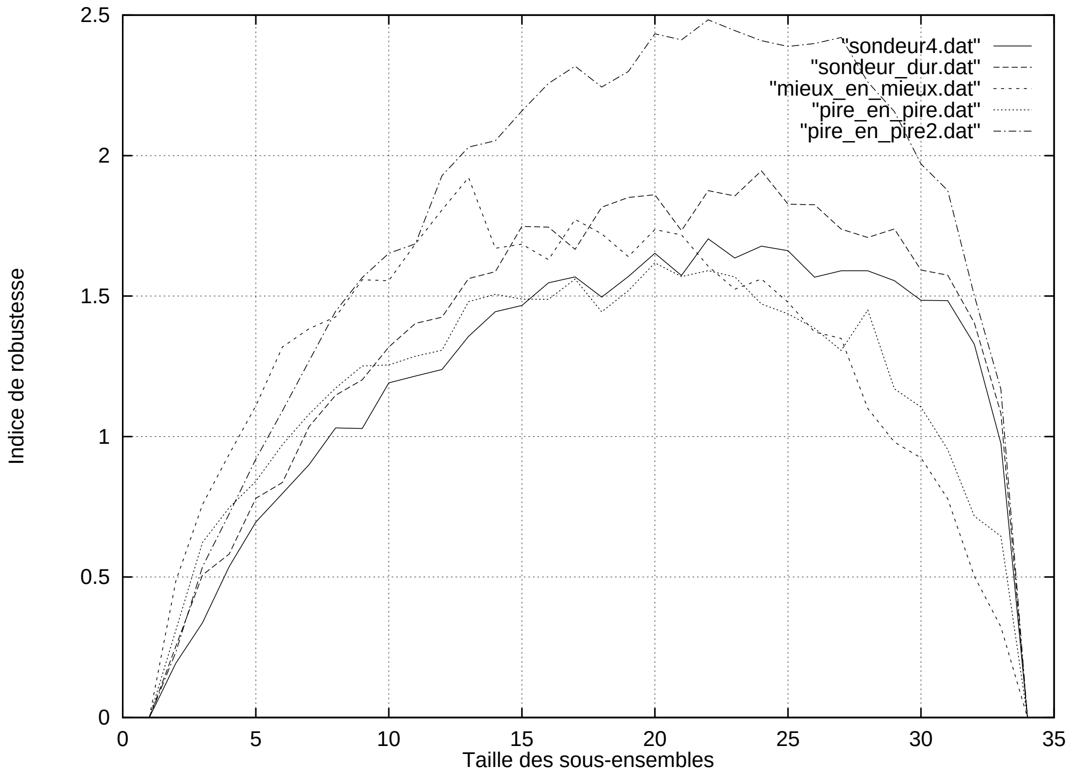

FIG. B.6 - *Robustesse des 26 à 30es stratégies de base*

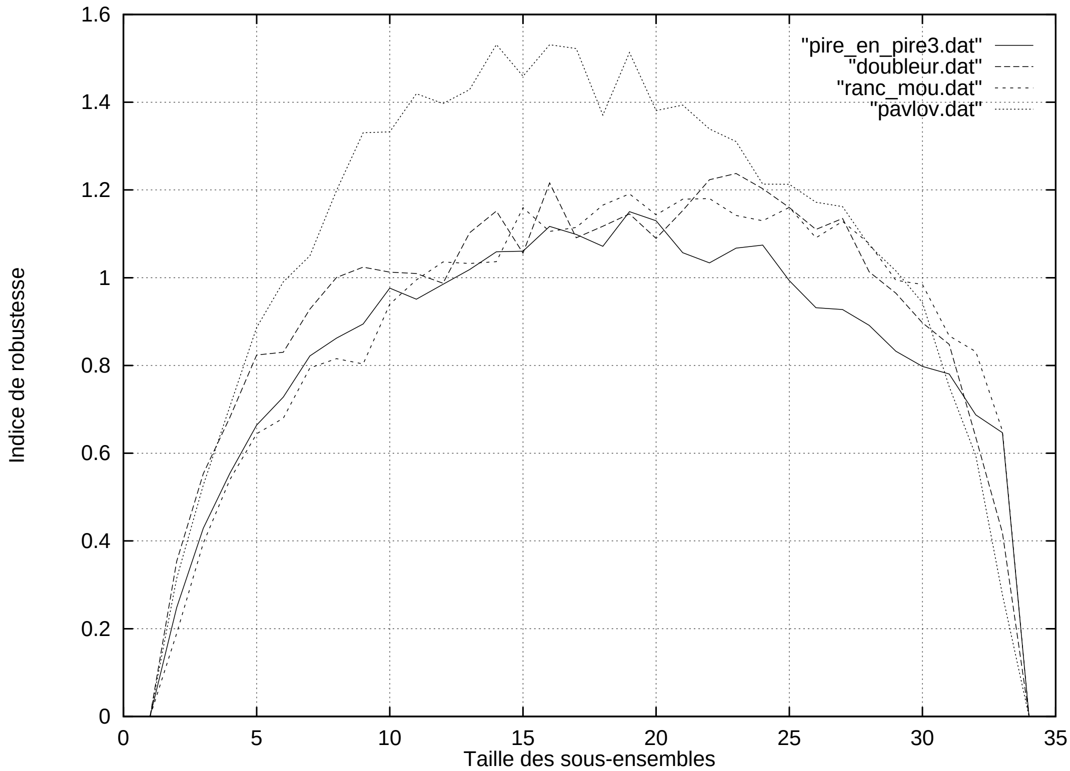

FIG. B.7 - *Robustesse des 31 à 34es stratégies de base*

Annexe C

Résultats des expériences génétiques

Les expériences ont été menées sur chaque famille de stratégies, en adoptant comme fitness, le classement des stratégies dans un tournoi entre tous les individus de la famille, et les stratégies de base, et ce avec différents taux de renouvellement de la population : 5, 10 ou 15 renouvellements par cycle. Le nombre de coups par parties est de 200, et le nombre de recalcul des stratégies utilisant le hasard de 4.

Les populations de chaque famille comportent 30 individus, générés aléatoirement au début de chaque utilisation du programme.

Le taux de mutation est de 2%.

Pour chaque stratégie obtenue, je donne le nombres de cycles nécessaires à la convergence, le classement sur les 64 stratégies à la convergence, le classement de la stratégie contre les stratégies de base, et une mesure de la robustesse pour des sous-ensembles de 17 parmi 35, avec 250 mesures.

Enfin pour chaque classe d'expériences je donne l'évolution de la population en compétition écologique. Pour ces compétitions le nombre de coups par partie est porté à 1000, et le nombre de recalcul à 8.

C.1 15 renouvellements par cycle

Les calculs ont été faits sur un i486 (chti).

C.1.1 Les probabilistes

- Cycle 25 ;
- convergé vers `mechante` ;
- 23e sur les 64 de l'algo ;
- 34e sur les 35 de la base plus elle-même ;
- Robustesse de 2,292.

C.1.2 Les majoritaires

- Cycle 25 ;
- convergé vers (`C`, 1, 1, 1, 6, 1, 1, 2) ;

- 8e sur les 64 de l'algo ;
- 9e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 1,038.

C.1.3 Les analytiques

- Cycle 50 ;
- convergé vers (7, (`CTCTTCC`), 0, 10, 1) ;
- 10e sur les 64 de l'algo ;
- 7e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 1,966.

C.1.4 Les super graduelles

- Cycle 25 ;
- convergé vers (`C`, 7, TRAHISONS, 6, 9, OUI, 10, POLYNOME, 8, 6, 7, LOGARITHME, 0, 3, 0, LOGARITHME, 4, 3, 8) ;
- 9e sur les 64 de l'algo ;
- 18e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 1,574.

C.1.5 Les parasites

- Cycle 25 ;
- convergé vers (10, (`binaire, doubleur, joss_mou, mefiant, tf2t_mou, tit_for_tat, graduel, pavlov, mieux_en_mieux, tft_dur`), 5) ;
- 2e sur les 64 de l'algo ;
- 9e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 1,838.

C.1.6 Les automates

- Cycle 25 ;
- convergé vers (3, (`graduel, sondeur4, mefiant`), (∅, (ET, 4, 28), (COMP, 49, 50), (COMP, 76, 2), ∅, (COMP, 44, 19), (ET, 31, 60), (OU, 37, 23), ∅), 0, 10) ;
- 2e sur les 64 de l'algo ;
- 2e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 0,6.

C.2 10 renouvellements par cycle

Les calculs ont été faits sur un i486 (labatt).

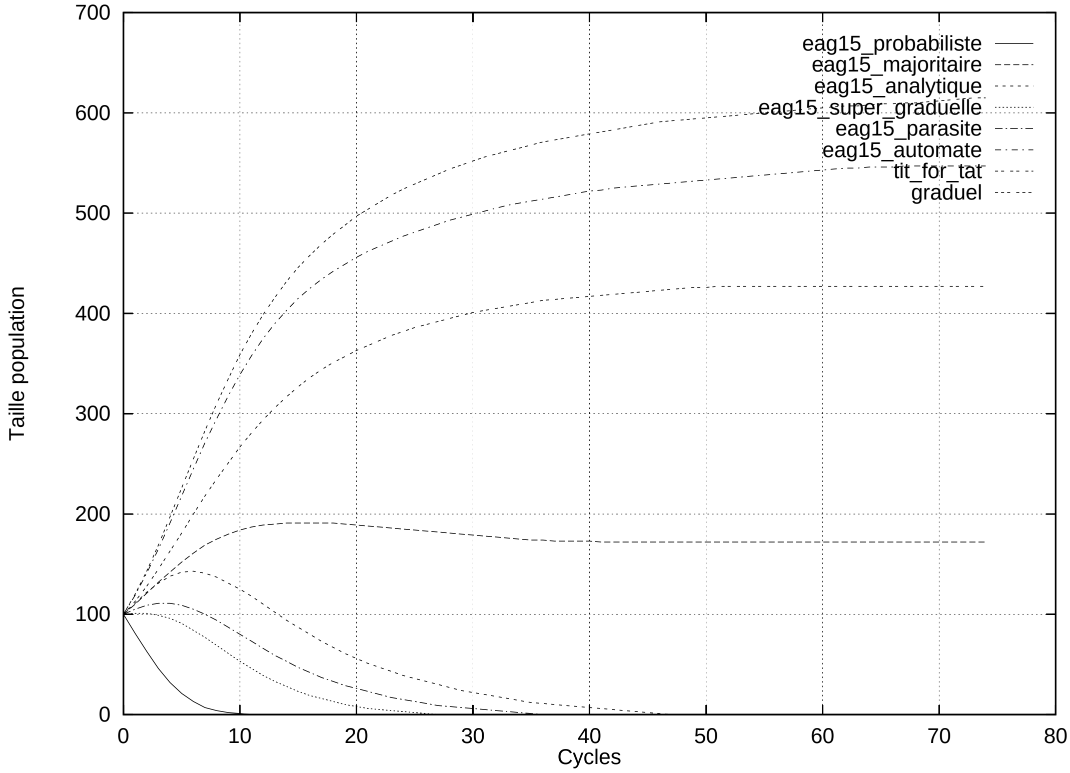

FIG. C.1 - *Stratégies issues de 15 renouvellements*

C.2.1 Les probabilistes

- Cycle 25 ;
- convergé vers `mechante` ;
- 24e sur les 64 de l'algo ;
- 34e sur les 35 de la base plus elle-même ;
- Robustesse de 1,75.

C.2.2 Les majoritaires

- Cycle 75 ;
- convergé vers (`C`, 1, 1, 2, 7, 6, 0, 7) ;
- 3e sur les 64 de l'algo ;
- 7e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 1,35.

C.2.3 Les analytiques

- Cycle 100 ;
- convergé vers (1, (`C`), 1, 1, 20) ;

- 4e sur les 64 de l'algo ;
- 8e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 1,172.

C.2.4 Les super graduelles

- Cycle 50 ;
- convergé vers (`C`, 3, SCORES, 6, 9, NON, 10, POLYNOME, 7, 0, 1, LOGARITHME, 2, 10, 9, POLYNOME, 2, 2, 0) ;
- 17e sur les 64 de l'algo ;
- 15e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 1,55.

C.2.5 Les parasites

- Cycle 25 ;
- convergé vers (10, (`binaire, doubleur, joss_mou, mefiant, tf2t_mou, tit_for_tat, graduel, pavlov, mieux_en_mieux, tft_dur`), 5) ;
- 2e sur les 64 de l'algo ;
- 8e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 2,04.

C.2.6 Les automates

- Cycle 25 ;
- convergé vers (3, (`graduel, sondeur4, mefiant`), (∅, (ET, 4, 28), (COMP, 49, 50), (COMP, 76, 2), ∅, (COMP, 44, 19), (ET, 31, 60), (OU, 37, 23), ∅), 0, 10) ;
- 2e sur les 64 de l'algo ;
- 2e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 0,46.

C.3 5 renouvellements par cycle

Les calculs ont été faits sur un i486 (bush).

C.3.1 Les probabilistes

- Cycle 50 ;
- convergé vers `mechante` ;
- 24e sur les 64 de l'algo ;
- 34e sur les 35 de la base plus elle-même ;
- Robustesse de 2,362.

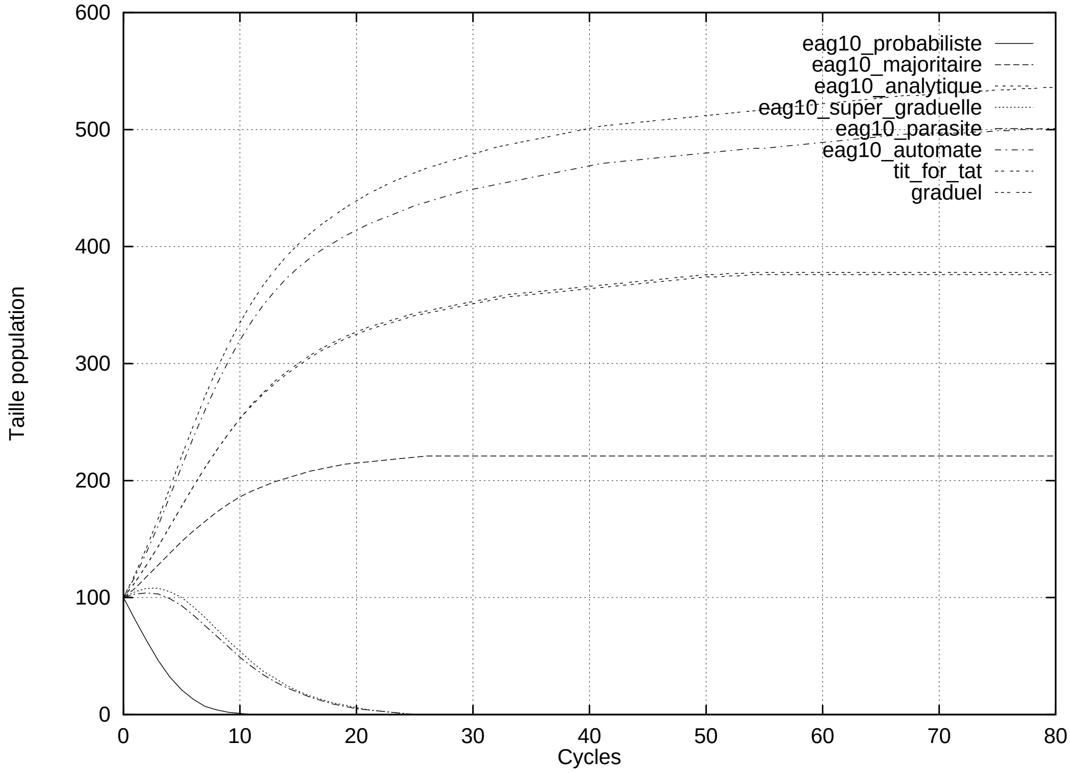

Fig. C.2 - *Stratégies issues de 10 renouvellements*

C.3.2 Les majoritaires

- Cycle 175 ;
- convergé vers (`C`, 20, 1, 2, 6, 0, 0, 4) ;
- 4e sur les 64 de l'algo ;
- 6e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 1,38.

C.3.3 Les analytiques

- Cycle 300 ;
- convergence relative vers (1, (`C`), 1, 10, 10) ;
- 4e sur les 64 de l'algo ;
- 7e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 1,016.

C.3.4 Les super graduelles

- Cycle 100 ;
- convergé vers (`C`, 0, SCORES, 7, 6, NON, 10, POLYNOME, 8, 6, 7, LOGARITHME, 0, 3, 0, LOGARITHME, 1, 10, 1) ;

- 3e sur les 64 de l'algo ;
- 4e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 1,312.

C.3.5 Les parasites

- Cycle 25 ;
- convergé vers (10, (`binaire, doubleur, joss_mou, mefiant, tf2t_mou, tit_for_tat, graduel, pavlov, mieux_en_mieux, tft_dur`), 5) ;
- 2e sur les 64 de l'algo ;
- 6e sur les 35 de la base plus elle-même (derrière `tit_for_tat` et `graduel`) ;
- Robustesse de 2,266.

C.3.6 Les automates

- Cycle 135 ;
- convergé vers (2, (`graduel, tf2t_mou`), ($\emptyset$, (ET, 65, 61), (COMP, 51, 14), $\emptyset$), 0, 200) ;
- 2e sur les 64 de l'algo ;
- 2e sur les 35 de la base plus elle-même (derrière `graduel`) ;
- Robustesse de 0,6.

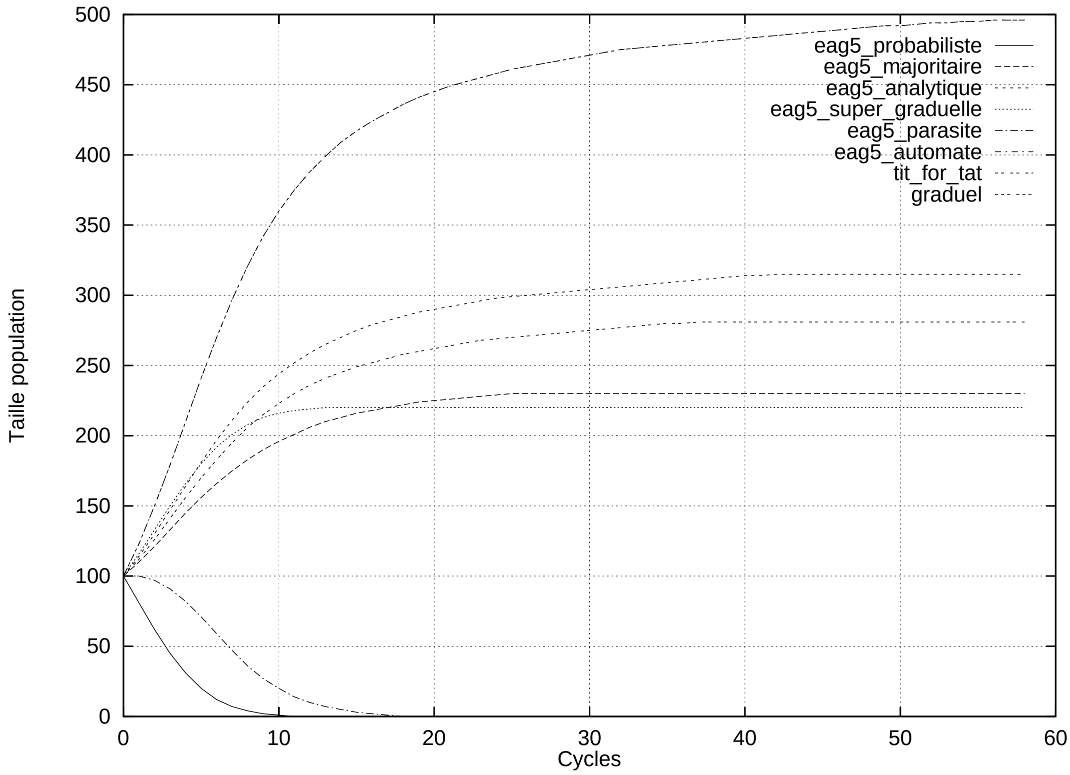

FIG. C.3 - *Stratégies issues de 10 renouvellements*

Bibliographie

[AD88] Robert Axelrod and Douglas Dion. The further evolution of cooperation. *Science*, 242:1385–1390, December 1988.

[AH81] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, March 1981.

[Ang94] Peter J. Angeline. An alternate interpretation of the iterated prisoner's dilemma and the evolution of non-mutual cooperation. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life, Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, volume 4, pages 350–352. MIT Press, 1994.

[ASST94] Dan Ashlock, Mark D. Smucker, E. Ann Stanley, and Leigh Tesfatsion. Preferential partner selection in an evolutionnary study of prisoner's dilemma. Economics Report No 35, Submitted for publication, September 1994.

[Axe84] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, New York, 1984.

[Axe92] Robert Axelrod. *Donnant Donnant :* Théorie du comportement coopératif. Number 2-7381-0145-3. Éditions Odile Jacob, Paris, 1992. Traduction française de [Axe84].

[Ban94] Steve Bankes. Exploring the foundations of artificial societies: Experiments in evolving solutions to iterated n-player prisoner's dilemma. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life, Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, volume 4, pages 337–342. MIT Press, 1994.

[BBM93] David Beasley, David R. Bull, and Ralph R Martin. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.

[Ben87] Jonathan Bendor. In good times and bad: Reciprocity in an uncertain world. *American Journal of Political Science*, 31:531–558, 1987.

[BK94] John Batali and Philip Kitcher. Evolutionary dynamics of altruistic behavior in optional and compulsory versions of the iterated prisoner's dilemma. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life, Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, volume 4, pages 344–348. MIT Press, 1994.

[BL87] Robert Boyd and Jeffrey P. Lorberbaum. No pure strategy is evolutionarily stable in the repeated prisoner's dilemma game. *Nature*, 327:58–59, May 1987.

[DM92] J.P. Delahaye and P. Mathieu. Expériences sur le dilemme itéré des prisonniers. Publication 233, Laboratoire d'Informatique Fondamentale de Lille CNRS (URA 369), Juin 1992.

[DM93] J.P. Delahaye and P. Mathieu. L'altruisme perfectionné. Publication 249, Laboratoire d'Informatique Fondamentale de Lille (CNRS URA 369), Mai 1993.

[DM95] J.P. Delahaye and P. Mathieu. Complex strategies in the iterated prisoner's dilemma. In Alain Albert, editor, *Chaos and Society*, Amsterdam, 1995. IOS Press.

[Fre94] Marcus R. Frean. The prisoner's dilemma without synchrony. *Proceedings Royal Society London*, 257(B):75–79, 1994.

[God92] H. C. J. Godfray. The evolution of forgiveness. *Nature*, 355:206–207, January 1992.

[Gol94] David E. Goldberg. *Algorithmes Génétiques*. Number 2-87908-0054-1. Addison-Wesley, Paris, 1994. Traduction Française.

[Jos87] N. V. Joshi. Evolution of cooperation by reciprocation within structured demes. *Journal of Genetics*, 66(1):69–84, April 1987.

[Lad94] Laurent Ladrière. Algorithmes génétiques et dilemme itéré des prisonniers. Mémoire de dea, Université des Sciences et Technologies de Lille, Laboratoire d'Informatique Fondamentale de Lille, juin 1994.

[Lin92] Kristian Lindgren. Evolutionary phenomena in simple dynamics. In Christopher Langton, Charles Taylor, J. Doyne Farmer, and Steen Rasmissen, editors, *Artificial Life, Proceedings of the Workshop on Artificial Life*, volume 2, pages 295–312. Santa Fe Institute, Addisson Wesley, 1992.

[Mar95] Carlo Martino. Emergent nastiness in iterated prisoner's dilemme games. 2.725: Design and Automation, 1995.

[May87] Robert M. May. More evolution of cooperation. *Nature*, 327:15–17, May 1987.

[Mol85] Per Molander. The optimal level of generosity in a selfish, uncertain environment. *Journal of Conflict Resolution*, 29(4):611–618, December 1985.

[Now90] Martin Nowak. Stochastic strategies in the prisoner's dilemma. *Theoretical Population Biology*, 38:93–112, 1990.

[NS89] Martin Nowak and Karl Sigmund. Oscillations in the evolution of reciprocity. *Journal of Theoretical Biology*, 137:21–26, 1989.

[NS90] Martin Nowak and Karl Sigmund. The evolution of stochastic strategies n the prisoner's dilemma. *Acta ApplicandæMathematicæ*, 20:247–265, 1990.

[NS92] Martin Nowak and Karl Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355:250–253, January 1992.

[NS93] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisonner's dilemma game. *Nature*, 364:56–58, July 1993.

[Oli94] Michael Oliphant. Evolving cooperation in the non-iterated prisoner's dilemma: The importance of spatial organization. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life, Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, volume 4, pages 350–352. MIT Press, 1994.

[Poo95] Robert Pool. Putting game theory to the test. *Science*, 267:1591–1593, March 1995.

[Pou93] William Poundstone. *Prisoner's Dilemma :* John von Neumann, Game Theory, and the Puzzle of the Bomb. Number 0-19-286162-X. Oxford University Press, Oxford, 1993.

[Pre94a] Philippe Preux. Algorithmes évolutifs, 1994. Séminaire de DEA.

[Pre94b] Philippe Preux. Les algorithmes génétiques. Publication AS-145, Laboratoire d'Informatique Fondamentale de Lille (CNRS URA 369), février 1994.

[SSA94] Mark D. Smucker, E. Ann Stanley, and Dan Ashlock. Analyzing social network structures in the iterated prisoner's dilemma with choice and refusal. Technical Report CS-TR-94-1259, University of Wisconsin-Madison, Departement of Computer-Sciences, December 1994.

[TK94] Masaru Tomita and Takashi Kido. Sacrificial acts in single round prisoner's dilemma. *Informatica*, 18:411–416, 1994.

[YD94] Xin Yao and Paul J. Darwen. An experimental study of n-person iterated prisonner's dilemma game. *Informatica*, 18:435–450, 1994.