Apprentissage de stratégies coopératives dans un contexte de jeu multi-opérateurs de télécommunications : l'IA coopérative au service des dilemmes sociaux

Tangui Le Gléau

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Par
Tangui Le Gléau

Apprentissage de stratégies coopératives dans un contexte de jeu multi-opérateurs de Télécom

L'intelligence artificielle coopérative au service des dilemmes sociaux

Thèse présentée et soutenue à Rennes, le 3 février 2022
Unité de recherche : IRISA – UMR 6074

Rapporteurs avant soutenance :

Latifa Oukhellou	Directrice de recherche, Université Gustave Eiffel
Philippe Mathieu	Professeur, Université de Lille

Composition du Jury :

Examinateurs :	Abderrahim Benslimane	Professeur, Université d'Avignon
	Latifa Oukhellou	Directrice de recherche, Université Gustave Eiffel
	Philippe Mathieu	Professeur, Université de Lille
	Tanguy Urvoy	Ingénieur de recherche, Orange Labs Lannion
Dir. de thèse :	Tayeb Lemlouma	Maître de Conférences HDR, Université de Rennes 1, IRISA, CNRS
Co-dir. de thèse :	Xavier Marjou	Ingénieur de recherche, Orange Labs Lannion

Remerciements

Je tiens tout d'abord à remercier mes encadrants de thèse : Tayeb, pour sa réactivité, sa disponibilité et son efficacité dans les relectures et dans ses retours constructifs, Benoit pour son partage de connaissance sur les réseaux de télécommunications et pour sa patience pour m'expliquer la signification de tous ces ATL [1], et enfin Xavier, pour son humanité, sa confiance et pour les échanges créatifs que l'on a pu avoir ensemble.

Je remercie mes rapporteurs Latifa Oukhellou et Philippe Mathieu qui ont accepté d'évaluer mon travail de thèse. Je remercie également les membres du jury, Abderrahim Benslimane et Tanguy Urvoy, d'accorder du temps pour participer à la soutenance de cette thèse. Je remercie enfin les membres de mon CSI, Abderrezak Rachedi et Cédric Gueguen pour les conseils donnés lors des réunions annuelles.

Je tiens ensuite à adresser toute ma reconnaissance à l'équipe ARC d'Orange Labs de m'avoir accueilli de la meilleure des manières. D'abord à Gaël pour son efficacité administrative et ses encouragements très motivants, et enfin bien entendu au reste de l'équipe pour cet environnement familial et chaleureux, ainsi que pour l'organisation des différents pique-niques et sorties. Et naturellement, un grand merci aussi aux collègues de l'équipe du fond du couloir ;) Je n'oublierai pas ces trois années passées dans l'inégalable ambiance positive et bienveillante du couloir WD100, vous allez me manquer !

Et puisque la thèse ne s'arrête pas aux portes du laboratoire, je tiens à remercier toutes les personnes rencontrées qui ont fait de ces années passées dans le Trégor un excellent moment. Une pensée d'abord bien évidemment à mes colocataires (qu'ils aient été thésards ou non [2]) que j'ai pu côtoyés plus ou moins longuement durant ces trois ans. Dans l'ordre d'apparition : Pierre,

1. ATL : Acronyme à Trois Lettres
2. personne n'est parfait :)

Solenne, Rémy, David et Enora. Merci pour ces moments passés au X rue de Bourgogne ! C'était bien sympathique. Un petit bonjour à tous les autres doctorants et stagiaires du site d'Orange Lannion, pour les rencontres, les repas, (les TP), les échanges, les plages, les restos etc. Une grosse pensée à mes amis, en particulier du lycée ou anciens de promo qui m'ont soutenu et sont venus me rendre visite en Bretagne, et aux copains de rando avec qui j'ai passé des moments très sympathiques ;) J'adresse également un immense merci à toute l'équipe d'Ultimate Frisbee de Lannion pour leur bonne humeur et la super ambiance des entraînements du soir : "Allez les Krampouz !". Enfin, un petit coucou à tous les copains de windsurf pour toutes ces sessions partagées. Aller se vider la tête dans les coups de vents hivernaux bretons ou aller profiter de la brise thermique des soirs d'été, aura été incontestablement une chance !

Enfin, merci à ma famille, en particulier mes parents, pour m'avoir accueilli dans les plus belles conditions au cours d'une partie des confinements dûs au [3] COVID-19, de m'avoir soutenu dans les moments de doute tout en sachant rester plus distants dans les périodes un peu chargées. Une pensée toute particulière à ma nièce qui est venue au monde au cours de la pandémie, pour redonner le sourire à un moment où tout le monde en avait besoin, une grosse pensée bien entendu aussi à ses parents :) J'adresse pour finir toute ma reconnaissance à mon frère, pour m'avoir emmener décompresser et m'aérer lors de ces longs treks en raquettes, seuls dans les montagnes enneigées, merci infiniment pour ces moments d'évasion, même si tu ne regardais pas toujours d'un bon oeil les papiers que je glissais dans le sac pour la lecture du soir [4].

Pour finir, merci à tous ceux que je n'ai pas mentionnés, que j'ai croisés, dans un cadre professionnel ou personnel, qui ont contribué à faire de ces trois années, un très bon moment, riche et épanouissant.

3. à la
4. tu reconnaîtras que ça nous a bien servi...

Table des matières

Liste des acronymes **11**

Liste des figures **16**

Liste des tables **18**

Introduction **19**

Contexte et motivations de la thèse 19

L'intelligence artificielle coopérative 21

L'intérêt de la collaboration dans l'univers des Télécom 23

Contributions 25

Contributions algorithmiques 25

Contributions logicielles 26

Publications et brevets 26

Organisation du manuscrit 27

I Des dilemmes sociaux à l'apprentissage multi-agents **29**

1 Jeux non-coopératifs et dilemmes sociaux **31**

1.1 Introduction 31

1.1.1 De l'économie aux mathématiques 31

1.1.2 Des applications omniprésentes 32

1.1.3 Notions abordées dans ce chapitre 34

1.2 Notions générales de la théorie des jeux 35

1.2.1 Définition d'un jeu sous forme normale 35

1.2.2 Stratégies dominantes et dominées 37

1.2.3 Equilibre de Nash . . . 37
1.2.4 Extension mixte d'un jeu sous forme normale . . . 38
1.2.5 Optimum de Pareto . . . 39
1.2.6 Jeux à information incomplète . . . 40
1.2.7 Jeux répétés . . . 41
1.3 Les dilemmes sociaux . . . 42
1.3.1 Définition de base . . . 42
1.3.2 Les trois catégories de dilemme social . . . 44
1.3.3 Dilemme du prisonnier itéré . . . 46
1.3.4 Dilemme du prisonnier continu . . . 46
1.3.5 Le dilemme du prisonnier à N joueurs . . . 47
1.4 Stratégies gagnantes pour le dilemme du prisonnier itéré . . . 48
1.4.1 Les "bonnes" stratégies . . . 48
1.4.2 Exemples de stratégies . . . 48
1.4.3 Simulations de tournois . . . 50
1.4.4 Tit-for-Tat continu . . . 51
1.5 Synthèse . . . 51

2 Apprentissage par renforcement et dilemmes sociaux complexes 53
2.1 Le Machine Learning : une brève topologie . . . 53
2.1.1 Les catégories d'apprentissage . . . 54
2.1.2 Les réseaux de neurones . . . 56
2.2 Apprentissage par renforcement . . . 58
2.2.1 Processus de décision markovien . . . 58
2.2.2 Principe général et dilemme "exploitation/exploration" . . . 59
2.2.3 Catégories d'algorithmes . . . 60
2.2.4 Des méthodes tabulaires au *Deep Reinforcement Learning* (DRL) . . . 64
2.2.5 Bandits manchots : du *Reinforcement Learning* (RL) sans observation . . . 66
2.2.6 Apprentissage par Renforcement Multi-Agents (MARL) . . . 67
2.3 Dilemmes sociaux séquentiels . . . 68
2.3.1 Exemples de jeux . . . 68
2.3.2 Dilemme social séquentiel . . . 69
2.3.3 Approches proposées pour jouer dans les *Sequential Social Dilemma* (SSD) 73
2.4 Conclusion et perspectives . . . 75

II Vers l'apprentissage de la coopération 77

3 Apprentissage de stratégies dans un dilemme du prisonnier itéré simple 79
3.1 Introduction . . . 79
3.2 Aspects méthodologiques . . . 80
3.2.1 Tournoi de dilemme du prisonnier itéré à deux joueurs . . . 80
3.2.2 Métriques sociales . . . 80
3.3 Apprentissage par renforcement et dilemmes du prisonnier itérés . . . 83
3.3.1 Modèle . . . 83
3.3.2 Scénarios étudiés . . . 83
3.3.3 Étude de l'algorithme *Q-learning* . . . 84
3.3.4 Impact des valeurs des gains (S, P, R, T) sur la coopération . . . 85
3.4 Propositions d'une stratégie de Tit-for-Tat continu à deux joueurs . . . 88
3.4.1 Contexte . . . 88
3.4.2 Proposition d'un Tit-for-Tat (TFT) continu . . . 88
3.4.3 Bilan . . . 92
3.4.4 Simulations empiriques . . . 94
3.4.5 Évaluation des paramètres . . . 97
3.4.6 Quelques aspects théoriques . . . 98
3.5 Conclusions et perspectives . . . 101

4 Proposition d'un dilemme du prisonnier à N joueurs continu et non-réciproque 103
4.1 Introduction et motivations . . . 103
4.1.1 Les coopérations réciproques non systématiques . . . 103
4.1.2 Un Tit-for-Tat classique inadapté . . . 104
4.2 Formalisme du dilemme du prisonnier basé sur des graphes . . . 105
4.2.1 Comparaison des modèles et algorithmes existant . . . 105
4.2.2 Dilemme du prisonnier à N joueurs . . . 106
4.2.3 Dilemme du prisonnier multi-joueurs basé sur des graphes . . . 108
4.2.4 Exemples de *Graph-based Iterated Prisoner's Dilemma* (GIPD) . . . 109
4.3 Une extension de la politique de Tit-for-Tat classique . . . 110
4.3.1 Rappels sur le Tit-for-Tat et notions de réseaux de flots . . . 110
4.3.2 Principe et architecture de notre extension (GTFT) . . . 112
4.4 Méthode comparative des algorithmes de Tit-for-Tat . . . 115
4.4.1 Métriques utilisées . . . 115
4.4.2 Agents étudiés et leurs caractéristiques . . . 118
4.4.3 Tournois de simulation . . . 118

4.5 Résultats et discussion . . . 119
4.5.1 Impact du choix d'algorithme pour le traitement de graphe . . . 119
4.5.2 Impact du choix de la fonction de Tit-for-Tat . . . 121
4.5.3 Synthèse des observations . . . 123
4.6 Limitations . . . 123
4.6.1 Observation partielle . . . 123
4.6.2 Ambiguïté du choix commun du cycle . . . 123
4.7 Conclusion et perspectives . . . 125

5 Dilemmes sociaux séquentiels asymétriques et méthodes hybrides 127
5.1 Introduction et motivations . . . 127
5.2 Dilemmes sociaux séquentiels dans la littérature . . . 131
5.3 Modèle de dilemme social séquentiel non réciproque . . . 132
5.3.1 Quelques rappels sur les dilemmes sociaux séquentiels . . . 132
5.3.2 Politiques de coopération continue . . . 133
5.3.3 Dilemmes sociaux séquentiels circulaires . . . 134
5.4 Jeux et bancs de test . . . 136
5.4.1 Présentation des jeux . . . 136
5.4.2 Structure de la coopération configurable . . . 139
5.5 Une approche hybride . . . 140
5.5.1 Rappels sur les notions de TFT et de RL . . . 141
5.5.2 Architecture de notre approche . . . 142
5.5.3 Génération de politiques de RL pré-entraînées . . . 143
5.5.4 Détection de la coopération . . . 144
5.5.5 TFT à structure de graphe . . . 145
5.6 Simulations et résultats . . . 146
5.6.1 Métriques sociales . . . 146
5.6.2 Performances du Graph-based TFT . . . 146
5.6.3 Impact des paramètres de la fonction de TFT . . . 147
5.6.4 Impact de la détection de la coopération . . . 148
5.6.5 Limitations . . . 149
5.7 Conclusion et perspectives . . . 150

III Vers une collaboration entre opérateurs de Télécoms 153

6 Un framework multi-agents pour la collaboration multi-opérateurs 155
6.1 Introduction et motivations . . . 155

6.2 Notions et vocabulaire liés aux réseaux mobiles . . . 157
6.3 Notions de jeux et environnements OpenAI Gym . . . 160
6.3.1 Rappels des processus de décision markoviens POMDP . . . 160
6.3.2 Environnements au format Gym de OpenAI . . . 160
6.4 Proposition d'un environnement multi-opérateurs . . . 161
6.4.1 Caractéristiques de l'environnement . . . 162
6.4.2 Caractéristiques liées au framework Gym . . . 163
6.5 Exemples d'applications . . . 167
6.5.1 Jeux prédéfinis simples . . . 167
6.5.2 Jeux de situations réelles avec les données de l'ANFR . . . 169
6.6 Quelques aspects liés à la théorie des jeux . . . 171
6.6.1 Jeux simples . . . 171
6.6.2 Jeux de situation réelle . . . 172
6.7 Conclusion et perspectives . . . 174

7 Collaborations inter-opérateurs : algorithmes et simulations 177
7.1 Introduction . . . 177
7.2 Problèmes de collaboration . . . 178
7.2.1 Extension de couverture . . . 179
7.2.2 Extension de capacité . . . 182
7.3 Un modèle de partage par TFT . . . 183
7.3.1 Architecture . . . 183
7.3.2 Offres et demandes . . . 184
7.3.3 Détection du degré de coopération . . . 184
7.3.4 Politique de réponse de coopération . . . 185
7.3.5 Allocation . . . 185
7.4 Évaluation de l'extension de couverture . . . 186
7.4.1 Modèle . . . 186
7.4.2 Expérimentations . . . 186
7.5 Évaluation de l'extension de capacité . . . 188
7.5.1 Exemple de jeu pour les simulations . . . 188
7.5.2 Évaluations . . . 189
7.6 Conclusion et perspectives . . . 191

Conclusions et perspectives 193
Rappels du positionnement de la thèse et des contributions . . . 193
Perspectives . . . 195

Liste des publications **196**

Bibliographie **197**

A Simulations additionnelles et détails des métriques sociales **213**

A.1 Simulations additionnelles 213

A.2 Intuition des métriques sociales 215

Explications/Intuitions sur les dilemmes sociaux 216

Liste des acronymes

ANFR Agence Nationale des Fréquences. 26, 162, 167, 169, 174, 195

CNN *Convolutionnal Neural Network*. 163

CSSD *Circular Sequential Social Dilemma*. 28, 129–132, 150

DDPG *Deep Deterministic Policy Gradient*. 66

DP Dilemme du Prisonnier. 13, 45, 47, 48, 51, 103, 105–107, 119, 125, 130, 133

DQN *Double Deep Q-Network*. 142

DQN *Deep Q-Network*. 13, 65, 66, 142, 146

DRL *Deep Reinforcement Learning*. 6, 51, 58, 64, 68, 75, 163

DTFT *Damped Tit-for-Tat*. 51, 88, 89, 92

GIPD *Graph-based Iterated Prisoner's Dilemma*. 7, 105, 108–110, 116, 119, 123, 125, 127, 130, 150, 167, 194

GTFT *Graph-based Tit-for-Tat*. 7, 14, 112, 113, 118–123, 125, 126, 129–131, 136, 141–143, 145–148, 150, 174, 194

IPD Dilemme du Prisonnier Itéré. 25, 46, 48, 83–85, 94, 105, 125, 131, 141

LSA *Licensed Shared Access*. 177

MAB *Multi-Armed Bandit*. 83

MARL *Multi-Agent Reinforcement Learning*. 6, 58, 67, 131, 132, 156

MDP *Markov Decision Process*. 58, 59, 61, 62, 160

ML *Machine Learning*. 22, 53, 56

MNO *Mobile Network Operator*. 155–159, 162–164, 167, 174

MVNO *Mobile Virtual Network Operator*. 157

NIPD *N-Iterated Prisoner Dilemma*. 105–107, 123, 125

POMDP *Partially Observable Markov Decision Processes*. 9, 58, 66, 160, 163

PRB *Physical Ressource Block*. 159

RAN *Radio Access Network*. 155, 159

RL *Reinforcement Learning*. 6, 8, 26, 28, 53, 56, 58, 60, 66–70, 73–75, 79, 83, 85, 87, 101, 105, 106, 126–133, 137, 140–143, 146, 149, 150, 155, 156, 160, 161, 164, 174, 194–196

SSD *Sequential Social Dilemma*. 6, 28, 53, 68–73, 75, 126–134, 139, 150, 194

TFT Tit-for-Tat. 7–9, 14–18, 28, 49–51, 74, 75, 79, 80, 84–89, 94, 95, 97, 98, 101–105, 110–114, 118–123, 125, 126, 129–131, 140, 141, 145–148, 150, 151, 156, 174, 178, 181, 183–188, 190, 191, 194, 195, 213, 214

TS *Thompson Sampling*. 66

UCB *Upper Confidence Bound*. 66

VCG *Vickrey-Clarke-Groves*. 177

WSLS *Win–Stay, Lose–Shift*. 50

Table des figures

1 Exemples de dilemmes sociaux dans des situations réelles 20
2 Robots qui doivent apprendre à coopérer dans les entrepôts pour éviter les situations de conflit. Travaux de Mitsubishi sur l'apprentissage coopératif[1]. 22
3 Apprentissage fédéré en situation non-coopérative 22
4 Exemples de collaboration inter-opérateurs 23
5 Approches possibles pour la coopération 24

1.1 Jeu du morpion et algorithme *Minimax* 32
1.2 Paradoxe de Braess 33
1.3 Gains du Dilemme du Prisonnier (DP) continu 47

2.1 Trois types d'apprentissage 55
2.2 Principe du RL 56
2.3 Principe d'un neurone artificiel à 3 entrées 56
2.4 Perceptron basique 57
2.5 Processus de Décision Markovien (MDP) 59
2.6 Le dilemme "exploitation/exploration" 60
2.7 Principe des méthodes *Value-based* 64
2.8 Principe des méthodes *policy-based* 64
2.9 Un *Deep Q-Network* (DQN) qui étend le *Q-learning* 65
2.10 Méthodes neuronales pour les modèles de type *policy-based* 65
2.11 Exemple de jeux de type SSD 68
2.12 Diagrammes de Schelling pour les trois dilemmes sociaux classiques 71
2.13 Diagrammes de Schelling pour les jeux de dilemmes sociaux séquentiels à plus de deux joueurs proposés par [HLP+18] 73

3.1 Apprentissage d'un *Q-learning* avec modèles différents 84

3.2 Apprentissage d'un *Q-learning* avec modèle identique . . . 85
3.3 Apprentissage d'un *Q-learning* contre un TFT . . . 85
3.4 Impact de (S, P, R, T) sur les taux de coopération dans un entraînement de bandits EXP3 . . . 86
3.5 Impact de (S, P, R, T) sur les taux de coopération dans un entraînement de *Q-learning* 86
3.6 Cas du Lift Dilemma, un Q-learning face à un TFT . . . 87
3.7 Intérêt du paramètre d'inertie . . . 90
3.8 Intuition de l'intérêt du paramètre adaptatif β . . . 91
3.9 Intuition de l'intérêt du paramètre stochastique γ . . . 93
3.10 Simulation de deux TFT aux paramètres identiques : TFT($\alpha = 0.5, \beta = 0.3, \gamma = 0, r_0 = 0.2, c_0 = 0.0$) . . . 94
3.11 Paramètre d'inertie différent $\alpha \in \{0.1, 0.5\}$ . . . 95
3.12 Degré de coopération initial différent : $c_0 \in \{0.0, 1.0\}$ . . . 95
3.13 Taux d'incitation initial différent : $r_0 \in \{0.0, 0.5\}$ . . . 96
3.14 Coefficients r_0 et β différents : $(r_0, \beta) \in \{(0.5, 0.1), (0.0, 0.5)\}$ . . . 96

4.1 Tit-for-Tat dans des situations circulaires . . . 104
4.2 Dilemme du prisonnier à 3 joueurs . . . 107
4.3 Exemples de Dilemme du Prisonnier sous forme graphique . . . 109
4.4 Exemple de jeux concrets présentant des patterns de coopération réciproques, homogènes, ou bien circulaires . . . 110
4.5 Exemple d'un réseau de flot à gauche, et d'un de ses flots maximaux à droite[1] . . . 112
4.6 Architecture de notre approche *Graph-based Tit-for-Tat* (GTFT) . . . 113
4.7 Étapes pour trouver un sous-graphe cyclique de flot maximal partant et revenant au joueur 1 dans le graphe (4.7a) . . . 114
4.8 Tournois Circ(N) et DoubleCirc(N) . . . 119
4.9 Impact du choix de l'algorithme de traitement de graphe . . . 120
4.10 Impact du choix de la fonction de f_{TFT} sur le jeu DoubleCirc(6) avec GTFT et l'approche *min cost* . . . 122
4.11 Ambiguïté du choix de cycles avec quatre joueurs . . . 124
4.12 Ambiguïté du choix de cycles avec six joueurs . . . 124
4.13 Ambiguïté du choix des cycles avec cinq joueurs . . . 124

5.1 Quelques exemples d'agents intelligents dans des situations de coopération asymétrique . . . 128
5.2 Partage de ressources dont l'utilité marginale est décroissante . . . 135
5.3 Deux exemples de structure de coopération du jeu Collect . . . 137
5.4 Jeux Sharing et Traffic . . . 139

5.5 Jeu COLLECT avec des structures différentes de coopération 140
5.6 Architecture de notre approche GRTFTRL 142
5.7 Impact de la détection du potentiel de coopération sur U 148

6.1 Principe et exemple d'une coopération multi-MNOs 156
6.2 Stations de base et cellules 157
6.3 Partitions de Voronoï 158
6.4 Exemple de partage de ressources radio 159
6.5 Jeux Atari en OpenAI Gym 161
6.6 Quatre instances du jeu simple nommé `env_3A_3S_9U` 165
6.7 Observation totale et partielle du jeu prédéfini `env_3A_3S_9U-v1` 165
6.8 Quelques exemples de jeux prédéfinis à deux ou trois joueurs 167
6.9 Antennes du réseau 4G (LTE) dans la région du Trégor 170
6.10 Le concept d'un dilemme à deux opérateurs de Télécom 171
6.11 Une situation de dilemme social à trois joueurs représenté par un diagramme de Schelling 172
6.12 Rayon de voisinage pour la densification 172
6.13 Graphes de potentiel de coopération des situations présentées sur la Figure 6.9 . 173

7.1 Exemples de collaboration inter-opérateurs 178
7.2 Extension de la couverture réseau 179
7.3 Jeu d'échange de ressources discret à trois joueurs 182
7.4 Exemple de jeux avec quatre opérateurs 182
7.5 Architecture de notre agent à base de TFT 183
7.6 Modèle de simulation 186
7.7 Exemple de partage de ressources entre trois agents avec trois items d'utilité concave 187
7.8 Simulation dans le cadre de l'exemple de la Figure 7.7 avec trois agents TFT avec ($\alpha = 0.5$, $r_0 = 0.2$, $\beta = 0$). 187
7.9 Simulation avec deux agents TFT($\alpha = 0.5$, $r_0 = 0.2$, $\beta = 0$) 188
7.10 Simulation avec deux agents TFT au coefficient adaptatif β non nul : TFT($\alpha = 0.5$, $r_0 = 0.2$, $\beta = 0.3$) 188
7.11 Exemple de jeux avec quatre opérateurs. 189
7.12 Exemple de jeux avec quatre opérateurs joués par quatre agents de TFT 190
7.13 Exemple de jeux avec quatre opérateurs impliquant trois agents de TFT et un agent égoïste 190

A.1 Exemple de situation avec trois opérateurs impliquant trois agents de TFT dans le jeu `env_3A_3S_9U-v0` 213

A.2 Exemple de situation avec trois opérateurs impliquant trois agents de TFT dans le jeu `env_3A_3S_18U-v0` 214
A.3 Exemple de situation avec trois opérateurs impliquant deux agents de TFT et un agent égoïste dans le jeu `env_3A_3S_18U-v0` 214
A.4 Intuition des métriques sociales sous forme graphique 215

Liste des tableaux

1.1 Bataille des sexes 35
1.2 Coupure téléphonique 35
1.3 Volume sonore entre voisins aux goûts musicaux différents 36
1.4 Jeu du penalty 36
1.5 Jeu du penalty 39
1.6 Le jeu Pierre-Feuille-Ciseaux 39
1.7 Jeu du shérif 41
1.8 Gains dans un dilemme social à deux joueurs 43
1.9 Le jeu *Stag Hunt* 44
1.10 Respect du confinement 44
1.11 Le jeu *Chicken Game* 45
1.12 Le problème du candidat unique 45
1.13 Le Dilemme du Prisonnier d'origine 46
1.14 Le Dilemme du Prisonnier de la littérature 46
1.15 TFT vs Égoïste 50
1.16 Naïf vs Égoïste 50
1.17 TFT vs Lunatique 50
1.18 Rancuniervs Lunatique 50

2.1 Gains dans un dilemme social à deux joueurs 70

3.1 Rappels des gains du dilemme du prisonnier 80
3.2 Métriques sociales selon les paramètres du TFT 97
3.3 Métriques sociales selon le paramètre stochastique γ du TFT 98

4.1 Les agents étudiés lors des simulations 118
4.2 Comparaison des trois types de TFT (alpha, beta, gamma). 121

4.3 Impact du choix de la fonction de TFT 122

5.1 Positionnement des contributions des chapitres 4 et 5 130
5.2 Résultats avec les environnements Bilateral et Circular 147
5.3 Évaluation de l'impact du choix du GTFT 148

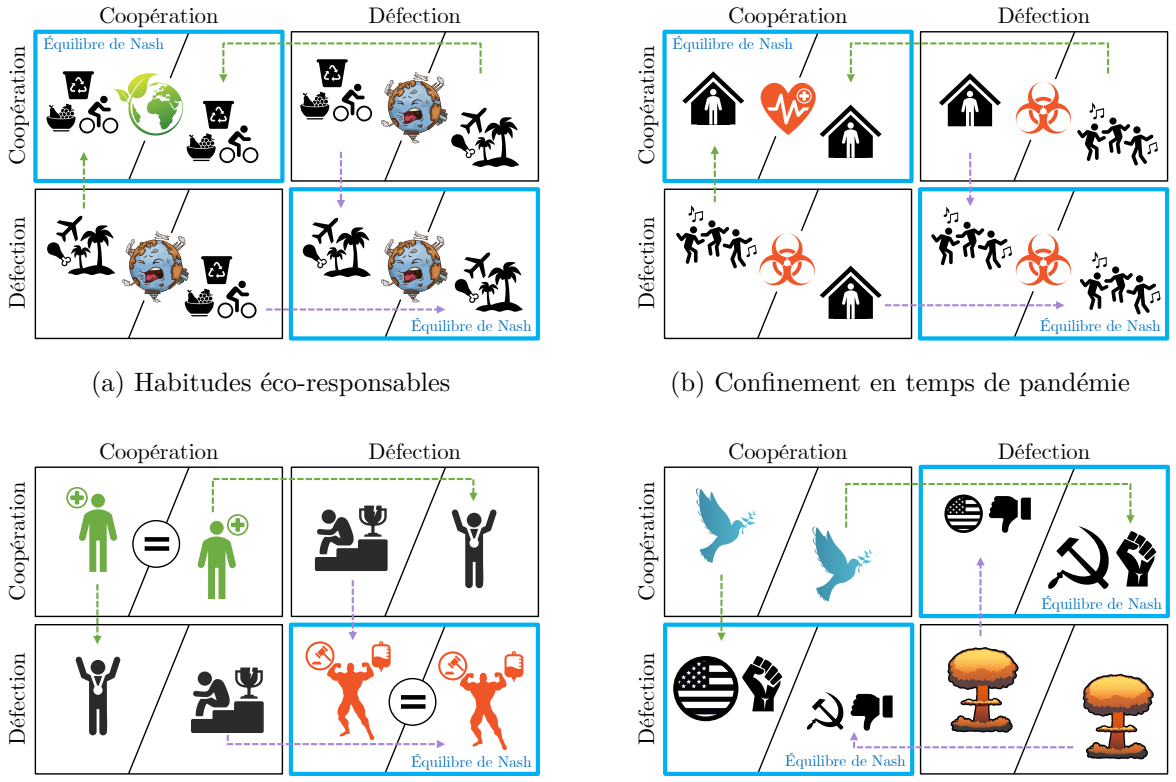
Introduction

Les travaux présentés dans ce manuscrit concerne la théorie des jeux et l'apprentissage automatique dans un contexte de télécommunications. Dans cette thèse, nous étudions les enjeux de la coopération d'agents apprenants (notamment des politiques d'apprentissage par renforcement) au sein de situations appelées dilemmes sociaux. Nous chercherons à appliquer ce type de modèle et d'agents intelligents dans un contexte de collaborations entre opérateurs de télécommunications.

Dans cette introduction, nous développons deux aspects. Nous commençons par illustrer à l'aide de quelques exemples, l'importance et les enjeux de l'intelligence artificielle coopérative. Puis, nous montrons brièvement comment le domaine des télécommunications offre un contexte intéressant pour ce paradigme.

Contexte et motivations de la thèse

Quotidiennement, nous faisons tous face à des situations de dilemme. Il s'agit de situations dans lesquelles nous interagissons avec un ou plusieurs autres acteurs et où notre intérêt personnel est souvent choisi au détriment d'une vision collective, ce qui conduit parfois à des situations regrettables. Dans certaines situations, nous avons la possibilité de choisir un comportement vertueux, dit de coopération, dont le choix mutuel garantit à tous un résultat optimal. Cependant, lorsque l'on adopte un comportement rationnel et égo-centré, choisir seul une décision dite coopérative est risqué et peut être perdant. Il arrive fréquemment que nos choix respectifs s'orientent alors vers une défection mutuelle qui est plus néfaste. Modélisé sous le nom de dilemme social (et en particulier le dilemme du prisonnier), ce concept modélise tout type de situation où au moins un des équilibres de décision (appelés équilibres de Nash) n'est pas optimal pour le bien commun. Nous pouvons retrouver ce concept dans de nombreuses situations de la vie quotidienne : en conduisant, en investissant notre argent, en jouant à certains jeux de société, en déclarant ses impôts, etc.



(a) Habitudes éco-responsables

(b) Confinement en temps de pandémie

(c) Tentation du dopage

(d) Crise des missiles (Cuba, 1962)

FIGURE 1 – Exemples de dilemmes sociaux dans des situations réelles. Les encadrés bleus désignent les équilibres de Nash. Les flèches désignent les préférences des joueurs d'une situation à une autre. Les flèches vertes (quand elles sont dirigées vers le bas) désignent l'avidité (préférence d'exploiter un coopérateur plutôt que la coopération mutuelle). Les flèches violettes (quand dirigées vers la droite) désignent la crainte (préférence de la défection mutuelle plutôt que de se faire exploiter par un défecteur).

Les dilemmes se retrouvent également dans l'actualité récente. En effet, les enjeux climatiques et le respect des confinements imposés par la pandémie de COVID-19 illustrent bien que les situations d'effort collectif conduisent parfois à des craintes d'être seul à s'impliquer par exemple dans des comportements éco-responsables (Figure 1a) ou dans des efforts de distanciation sociale (Figure 1b). Ces craintes peuvent alors conduire à un abandon collectif d'effort. Mais à l'inverse, dans certains cas, le fait qu'un grand nombre de "joueurs" s'implique peut inciter le reste à faire des efforts à leur tour pour atteindre ensemble une meilleure issue. Nous verrons dans le chapitre 1 que cette situation de confiance collective s'appelle le jeu de la chasse au cerf (*Stag hunt*). Par ailleurs, un deuxième type de dilemme, le dilemme du prisonnier, est également très présent dans nos comportements. On peut le retrouver dans les tentations d'exploiter la coopération des autres acteurs comme les cas de triche (par exemple la Figure 1c). On peut également rencontrer ces situations

dans les relations internationales ou dans la politique. L'exemple historique de la crise des missiles de Cuba en 1962 est un bel exemple du troisième type de dilemme. Ce dilemme est appelé *Chicken Game*, ou le jeu de la "poule mouillée" et symbolise les situations où l'un des acteurs doit se dévouer ou céder en premier au détriment de son intérêt personnel mais au profit du bien commun.

Si ces dilemmes sont omniprésents dans notre quotidien, on peut également les rencontrer dans des situations impliquant des agents intelligents. Dans la suite, nous allons évoquer les enjeux de l'intelligence artificielle coopérative, puis nous intéresser aux situations de potentielles collaborations dans des contextes de télécommunications.

Les enjeux de l'intelligence artificielle coopérative

Dans l'industrie ou dans notre quotidien, l'implémentation et le déploiement d'agents intelligents sont de plus en plus présents. Dès lors que ces agents sont indépendants et *a priori* intéressés uniquement par leurs intérêts personnels, il peut survenir des situations où l'intérêt collectif peut être plus profitable. Dans la suite, nous proposons quelques exemples pour illustrer les enjeux de la coopération entre agents artificiels.

- **Internet des objets**

L'explosion de l'utilisation d'objets connectés indépendants peut naturellement impliquer des situations de coopération. On peut en particulier envisager que des appareils puissent coopérer pour s'échanger des données de capteurs sans fil [YKAD12] ou collaborer pour mieux effectuer une tâche. Ceci peut être réalisé notamment grâce à un partage d'informations, comme cela a été proposé avec des appareils du domaine de la santé [RPP11]. Enfin, on peut imaginer les situations où les objets connectés ont à leur disposition de l'énergie pour leur batterie ou de la connectivité à partager, et qu'ils doivent coopérer pour ne pas aboutir à des situations non optimales [BEBS+19].

- **Véhicules autonomes**

Dans le cadre des véhicules autonomes qui se développent sur les routes [HCS+16], dans les entrepôts [WDM08] ou bien dans les airs [WYZL20], il devient nécessaire de s'intéresser aux enjeux des comportements ego-centrés. En effet, parfois, des conflits dûs à des choix égoïstes sans vision collective peuvent s'avérer contre-productifs voire très pénalisants, comme des collisions ou des impasses.

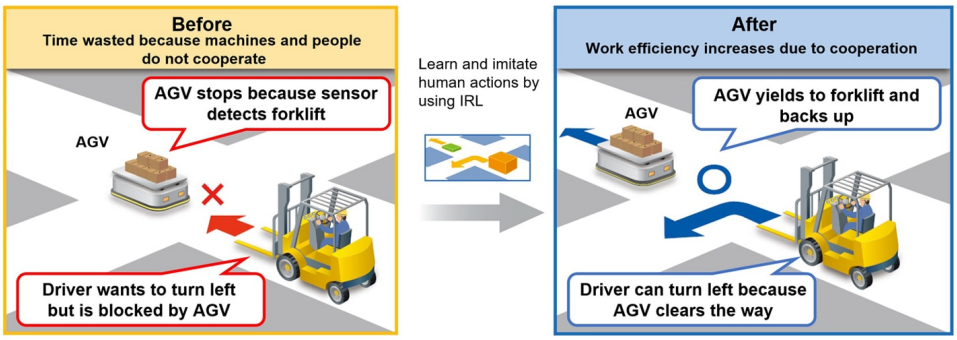


FIGURE 2 – Robots qui doivent apprendre à coopérer dans les entrepôts pour éviter les situations de conflit. Travaux de Mitsubishi sur l'apprentissage coopératif[1].

• **Apprentissage fédéré**

L'apprentissage fédéré (*federated learning*) consiste à entraîner des modèles de *Machine Learning* (ML) de manière décentralisée sur des machines différentes [SS15, MMR+17]. L'objectif des machines est d'entraîner localement ces modèles sur des données locales (souvent privées, ce qui fait l'intérêt du *federated learning*) puis de partager les mises à jour de modèle aux partenaires. Les agents décentralisés peuvent être considérés comme indépendants. Dans ce cas, ils peuvent faire preuve d'égoïsme comme par exemple, continuer de recevoir les mises à jour des paramètres des modèles sans consommer personnellement de l'énergie pour l'entraînement local, ou bien volontairement de ne pas partager les mises à jour issues d'entraînement sur des données précieuses. Dans ce contexte, il est alors intéressant d'introduire des mécanismes pour inciter la coopération et se prémunir des agents égoïstes [KXN+19, ZLQ+20].

FIGURE 3 – Apprentissage fédéré en situation non-coopérative

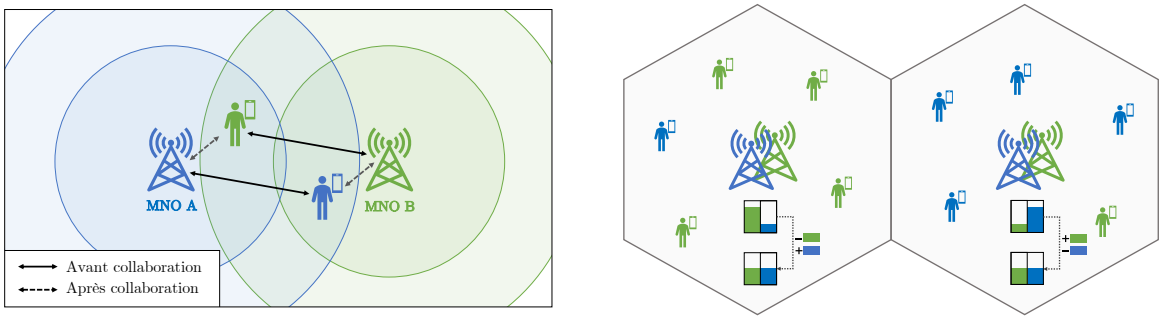
• **Partage et gestion des ressources**

La rareté des ressources naturelles et les besoins croissants en énergie doivent faire prendre conscience qu'il serait idéal que des agents intelligents coopérent ensemble pour mieux gérer collectivement les ressources. Ils peuvent alors apprendre à exploiter de manière collective et altruiste les ressources naturelles [OGW+94, PLZ+17] ou bien homogénéiser leurs ressources de manière optimale afin d'éviter le gaspillage ou les sur-coûts de stockage [SF89, LH19].

1. Figure issue de `https://be.mitsubishielectric.com/en/news/releases/global/2020/0603-a/index.html`

L'intérêt de la collaboration dans l'univers des Télécom

Le trafic des données mobiles est en constante augmentation. Ceci est dû en particulier à l'explosion des usages d'internet, comme la vidéo en streaming, en 4K ou bien la multiplication des objets connectés et le développement de nouvelles technologies comme les véhicules autonomes ou la télémedecine. Par conséquent, la gestion de ce trafic devient un enjeu fondamental pour les prochaines générations de téléphonie mobile (telles que la 5G). Pour adresser cette augmentation de demande, il est nécessaire de déployer de nouvelles infrastructures. Cependant dans un soucis environnemental et financier, il pourrait être intéressant de considérer des approches consistant à mutualiser certaines infrastructures entre opérateurs de télécommunications par le biais de transactions de ressources radio.



(a) Extension de capacité

(b) Extension de couverture

FIGURE 4 – Exemples de collaboration inter-opérateurs. Une intuition de jeux représentatifs avec deux opérateurs (représentés en vert et en bleu). Ils sont concurrents mais peuvent faire face à des situations où une coopération mutuelle est optimale pour tous.

Le partage de ressources peut être envisagé dans deux principaux cas de figure. Dans un premier temps, ces échanges peuvent contribuer à augmenter les capacités de réseau (amélioration de la qualité de service par l'augmentation du débit). Cette situation illustrée par la Figure 4a implique des clients que l'on suppose en bordure de réseau de leur antenne de rattachement. Ils pourraient être idéalement pris en charge par l'antenne d'un opérateur concurrent plus proche afin d'offrir une meilleure connectivité. Si le service est rendu de manière réciproque, alors la coopération entre opérateurs aura été gagnante pour les deux acteurs. Le deuxième cas de figure concerne l'augmentation de couverture réseau (Figure 4b). L'idée est ici d'adresser les zones à faible couverture voire les zones sans couverture (zones blanches). Un opérateur rendrait alors service à un autre en lui allouant de la ressource dans une cellule où il en manquerait.

Choix de l'approche et cheminement de pensée

Avant de détailler plus précisément les différentes contributions de la thèse, explicitons le cheminement de pensée qui nous a amené à étudier les aspects algorithmiques d'agents apprenant à coopérer. Pour commencer, nous avons cherché à formaliser de manière relativement simple les environnements impliquant divers opérateurs de télécommunication qui souhaitent coopérer par le partage d'infrastructures et de ressources radio. Il existe plusieurs approches pour tendre vers une coopération mutuelle. Pour les illustrer, nous faisons le parallèle avec la situation impliquant des voitures dans un carrefour (Figure 5).

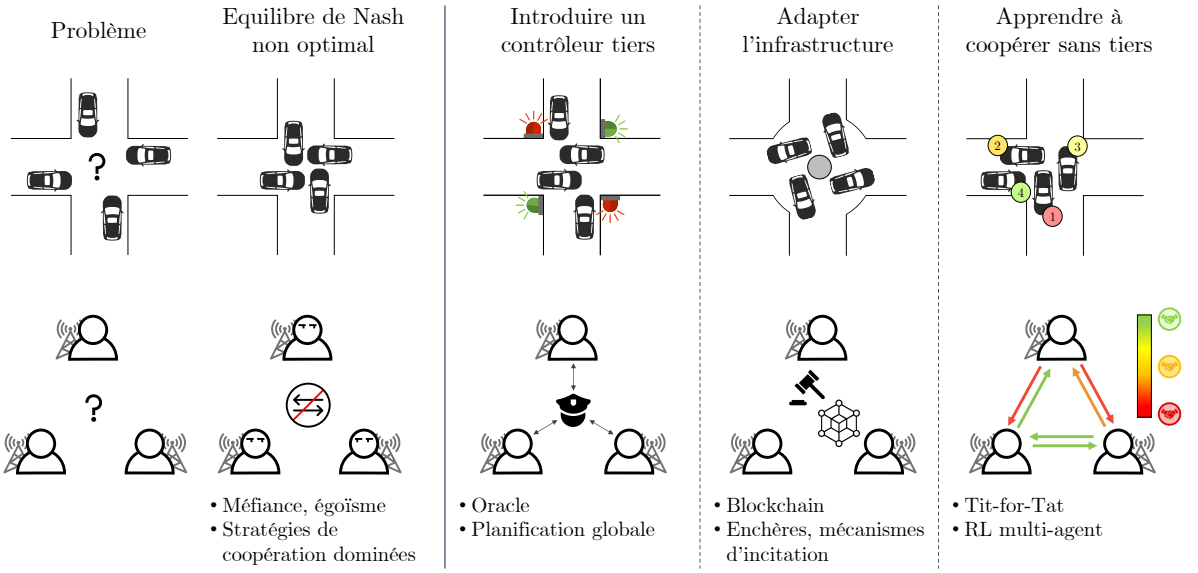


FIGURE 5 – Approches possibles pour la coopération entre opérateurs, comparaison avec la situation non-coopérative d'un carrefour

La situation de la Figure 5 est un dilemme au sens où les voitures ont toutes intérêt à traverser sans perdre de temps, ce qui conduit à un équilibre (équilibre de Nash) non optimal. Il en est de même pour les opérateurs ne souhaitant pas s'engager dans une coopération sans garanties. On peut donc envisager plusieurs solutions pour les inciter. La première consiste en la contrainte : il s'agit d'introduire un contrôleur qui va accéder à l'ensemble des données, et puis décider des actions (transactions) à effectuer par les différents joueurs. La présence d'un tiers ne fait pas partie des hypothèses envisagées, c'est pourquoi dans une deuxième idée, nous pourrions nous diriger vers des solutions de mécanismes d'incitation [CBA+19]. Cela revient à utiliser des mécanismes financiers sans tiers conçus pour inciter à être le plus honnête possible. Ces mécanismes pourraient être inscrits dans des structures de type blockchain *Distributed Ledger Technology* (DLT) [MSGR18] de sorte à être indépendant. Cependant, bien que cette idée

soit intéressante et s'approche davantage de nos hypothèses, elle nécessite une infrastructure commune dont l'accord peut être contraignant. Pour ces raisons, nous nous sommes dirigés vers une approche qui se base sur l'apprentissage de la coopération par la simple observation du jeu à l'aide de politiques apprenant à réagir face à l'égoïsme tout en maintenant une incitation à coopérer.

Contributions

La thèse s'articule dans un premier temps autour de contributions algorithmiques liées à l'apprentissage de politiques coopératives au sein de dilemmes sociaux. Une deuxième catégorie de contributions se concentre sur l'application de ces thématiques et algorithmes dans des cas de figure relatifs aux télécommunications. Nous commençons par résumer les contributions algorithmiques puis les contributions liées aux cas d'usages, et enfin nous listons les publications et brevets.

Contributions algorithmiques

Les travaux de la thèse se sont concentrés dans une importante proportion sur des aspects algorithmiques dans le domaine de l'apprentissage de politiques coopératives au sein de dilemmes sociaux (discrets, continus et complexes).

(i) **Étude de diverses politiques (TFT, RL) dans un IPD continu :** des études ont été conduites dans des tournois de dilemme du prisonnier itéré (IPD) entre plusieurs politiques d'apprentissage par renforcement (*Reinforcement Learning*) notamment pour motiver le fait que des politiques plus incitatives comme des Tit-for-Tat étaient nécéssaires.

(ii) **Contributions sur les stratégies de Tit-for-Tat continues :** nous avons apporté de nouvelles propositions sur le Tit-for-Tat continu, notamment en apportant une inertie, en le rendant plus robuste à la défection, et en le garantissant résilient. Quelques simulations empiriques ainsi que quelques preuves théoriques sont apportées.

(iii) **Formalisation des dilemmes sociaux multi-joueurs complexes et asymétriques :** nous avons introduit un formalisme permettant de modéliser des dilemmes du prisonnier itérés et continus à N joueurs et non nécessairement symétriques. En particulier, des dilemmes pouvant faire intervenir des chemins circulaires de coopération à plus de deux joueurs.

(iv) **Création d'un algorithme de Tit-for-Tat permettant d'interagir dans un dilemme du prisonnier multi-agents complexe et asymétrique (et/ou circulaire) :** pour s'adapter au problème plus général mentionné précédemment, nous apportons à l'algorithme Tit-for-Tat multi-joueurs une structure de graphe. Cette structure permet, à

l'aide de recherche de flot maximal, de déterminer un chemin permettant de coopérer (à deux joueurs ou plus) dans un dilemme du prisonnier non nécessairement symétrique et en particulier circulaire.

(v) **Formalisation des dilemmes sociaux *séquentiels* multi-agents asymétriques :** à l'image du dilemme social séquentiel [LZL+17] qui étend le dilemme social classique à des politiques de coopération plus complexes (de type RL), nous proposons la formalisation d'un dilemme social séquentiel asymétrique et en particulier circulaire qui reprend et étend l'idée présentée en (iii).

(vi) **Amélioration d'un algorithmique hybride (RL/TFT) :** nous proposons une amélioration d'un algorithme existant [LP17] qui combine des stratégies de Tit-for-Tat et des politiques de RL qui permet de s'adapter à la formalisation du point précédent.

Contributions logicielles et cas opérationnels

Parallèlement aux aspects algorithmiques, nous nous sommes penchés sur une modélisation de collaboration dans un univers de Télécom. Dans ce contexte, des opérateurs de télécommunication concurrents sont amenés à coopérer pour atteindre et obtenir de meilleurs gains de qualité de service pour leurs clients. Cette coopération va également engendrer des économies d'énergie liées à une meilleure utilisation des infrastructures.

(viii) **Création d'environnements de simulation multi-opérateurs :** nous avons implémenté un framework permettant de générer des jeux multi-opérateurs en Python (> 1000 lignes de code). Des jeux prédéfinis simples sont fournis et il est également possible de modéliser des situations réelles à l'aide des données de l'Agence Nationale des Fréquences (ANFR)

(ix) **Partage de ressources par Tit-for-Tat :** nous avons proposé une première approche de partage de ressources à base d'un algorithme de Tit-for-Tat continu que nous avons proposé dans les contributions mentionnées plus haut. L'idée revient à modéliser les ressources des opérateurs de Télécom sous la forme d'items à s'échanger.

(x) **Quelques simulations d'algorithmes :** nous avons utilisé nos algorithmes et évalué des politiques existantes sur certains environnements issus du framework développé dans le point (viii).

Publications et brevets

Articles acceptés

- *Multi-agents Ultimatum Game with Reinforcement Learning*, **T. Le Gléau**, X. Marjou, T. Lemlouma et B. Radier, PAAMS 2020 (Workshop)

- *Game theory approach in multi-agent resources sharing*, **T. Le Gléau**, X. Marjou, T. Lemlouma et B. Radier, IEEE ISCC 2020
- *A Multi-agent OpenAI Gym Environment for Telecom Providers Cooperation*, **T. Le Gléau**, X. Marjou, T. Lemlouma et B. Radier, ICIN 2021
- *Towards circular and asymmetric cooperation in a multi-player Graph-based Iterated Prisoner's Dilemma*, **T. Le Gléau**, X. Marjou, T. Lemlouma et B. Radier, ICAART 2022

Articles soumis

- *Tackling Asymmetric and Circular Sequential Social Dilemmas with Reinforcement Learning and Graph-based Tit-for-Tat*, **T. Le Gléau**, X. Marjou, T. Lemlouma et B. Radier
- *Evaluating Inter-Operator Cooperation Scenarios to Save Radio Access Network Energy*, X. Marjou, **T. Le Gléau**, V. Messié, B. Radier, T. Lemlouma et G. Fromentoux

Brevets

- Brevet 201841FR01 : *Procédé de gestion de ressources de télécommunications allouées dynamiquement à une pluralité d'opérateurs de télécommunications, produit programme d'ordinateur et dispositifs correspondants*, B. Radier, **T. Le Gléau**, X. Marjou et G. Fromentoux, Mai 2019
- Brevet 202360FR01 : *Procédé et dispositif de mise à disposition d'une ressource de communication*, **T. Le Gléau**, X. Marjou et B. Radier, Mars 2021
- Brevet 202444FR01 : *Procédé et dispositif de substituion d'une pluralité de fournisseurs de services par un fournisseur de services*, B. Radier, **T. Le Gléau** et X. Marjou, Septembre 2021

Organisation du manuscrit

Le manuscrit est organisé comme suit :

■ **Partie A : Des dilemmes sociaux à l'apprentissage multi-agents**
Nous détaillons dans cette partie l'état de l'art et les notions nécessaires à la compréhension du manuscrit.

- **Chapitre 1 :** Nous y introduisons les notions principales de théorie des jeux, et en particulier les jeux non-coopératifs et les dilemmes sociaux.
- **Chapitre 2 :** Nous présentons le principe de l'apprentissage par renforcement qui sert notamment à définir les dilemmes sociaux séquentiels que nous détaillons également.

■ **Partie B : Vers l'apprentissage de la coopération**

Cette partie traite de l'ensemble des contributions algorithmiques proposées pendant la thèse, des améliorations du Tit-for-Tat continu et la proposition d'un algorithme hybride TFT/RL.

- **Chapitre 3 :** Nous menons quelques études de politiques dans un dilemme du prisonnier classique à deux joueurs, discret ou continu. Nous étudions en particulier l'apprentissage de la coopération au sein de politiques de RL, de bandits et de TFT. Nous proposons des améliorations de TFT continus.
- **Chapitre 4 :** Nous introduisons dans ce chapitre notre modèle du dilemme du prisonnier continu, itéré à N joueurs dont les coopérations possibles sont basées sur un graphe orienté et pondéré. Nous proposons alors un algorithme de Tit-for-Tat adapté.
- **Chapitre 5 :** Nous étendons dans ce chapitre les dilemmes sociaux séquentiels (SSD) en leur version non réciproque CSSD (même paradigme que le chapitre 4). Nous proposons alors un algorithme hybride TFT/RL qui permet d'adresser ces jeux.

■ **Partie C : Vers la collaboration entre opérateurs de Télécom**

Dans cette partie, nous nous penchons sur des scénarios de collaboration entre les opérateurs de télécommunications, en utilisant notamment les algorithmes évoqués ou détaillés dans les parties précédentes.

- **Chapitre 6 :** Nous présentons l'implémentation d'un framework permettant de générer des environnements de coopération multi-opérateurs. Ces environnements au format OpenAI Gym sont modulables et permettent la simulation d'agents.
- **Chapitre 7 :** Nous présentons un algorithme de partage de ressources par TFT et nous menons quelques simulations de scénarios de coopération entre opérateurs de télécommunication sur des jeux implémentés au chapitre 6.

Partie I

Des dilemmes sociaux à l'apprentissage multi-agents

Chapitre 1

Jeux non-coopératifs et dilemmes sociaux

La théorie des jeux est un domaine à l'intersection des mathématiques, de l'économie et de l'intelligence artificielle. Elle est souvent désignée comme la science de la décision, elle a pour objectif d'étudier les situations impliquant des choix de plusieurs acteurs rationnels. Les situations étudiées dans cette thèse sont celles de la catégorie des jeux non-coopératifs, *i.e.* où les acteurs agissent dans leur seul intérêt personnel et de manière indépendante. Nous nous penchons en particulier sur les dilemmes sociaux, où l'intérêt personnel est préféré à l'intérêt collectif. Ce chapitre a pour objectif de définir les concepts nécessaires à la compréhension et l'étude des dilemmes sociaux et en particulier du dilemme du prisonnier et de ses variantes.

1.1 Introduction

La théorie des jeux est une branche scientifique relativement récente dont les fondations ne remontent qu'au siècle dernier. Pourtant, ses concepts nous touchent quotidiennement. Avant d'aborder des concepts de manière plus formelle dans les prochaines sections, nous présentons dans cette introduction quelques aspects historiques et proposons d'illustrer ce domaine à travers quelques exemples simples du quotidien.

1.1.1 De l'économie aux mathématiques

Les tous premiers concepts de la théorie des jeux tirent leurs origines au lendemain de la seconde guerre mondiale. Le mathématicien John von Neumann et l'économiste Oskar Morgenstern partagent alors la volonté de refonder l'économie sur des bases mathématiques. Ils proposent alors dans l'ouvrage fondateur *Theory of Games and Economic Behavior* [MVN53], une formalisation mathématique pour modéliser les comportements stratégiques d'acteurs économiques. Depuis ces fondations, de nombreux scientifiques ont contribué aux concepts liés à la théorie des jeux et pas moins de onze économistes se sont vus décerner depuis 1944, le prix Nobel d'économie pour leurs

recherches sur la théorie des jeux. On peut citer John Forbes Nash, connu notamment pour la formalisation des équilibres du même nom [Nas51], et Roger Myerson pour ses contributions sur la théorie des mécanismes d'incitation [Mye79].

L'économie est donc la discipline reine où l'on retrouve par essence les concepts de la théorie des jeux. Le gain que les acteurs ou les agents rationnels cherchent à maximiser est naturellement l'argent. Par exemple, le commerçant cherche à vendre ses articles le plus cher possible tandis que le client souhaite l'inverse. La présence de concurrents et le principe de l'offre et la demande permettent alors de régir les prix de manière rationnelle [Cou38].

1.1.2 Des applications omniprésentes

Bien que la théorie des jeux soit originellement dédiée à l'économie, on peut l'appliquer dans de nombreux autres domaines. Pour commencer, on retrouve ces concepts bien entendu dans les jeux au sens plus commun, telles que les échecs, le poker, le morpion ou tout autre jeu de plateau. Chaque joueur souhaite maximiser un gain représentant le but recherché. Par exemple, un gain positif en cas de victoire, négatif pour une défaite et éventuellement nul en cas d'égalité. Certains jeux sont dits compétitifs et à somme nulle. Les joueurs agissent alors de manière antagoniste avec intérêts personnels et la somme des gains des joueurs à l'issue du jeu est nulle : ce que l'un gagne, l'autre le perd ; il y a un gagnant et un perdant, éventuellement une égalité (gains nuls). D'autres jeux peuvent être à somme non-nulle ce qui peut prêter à des stratégies gagnant-gagnant. Enfin certains jeux de plateau sont dits collaboratifs auquel cas les joueurs partagent le même gain qu'ils cherchent à maximiser collectivement. Dans le jeu du morpion (voir la Figure 1.1), deux joueurs s'affrontent, en choisissant des stratégies (placer une croix ou un cercle) et obtiennent à l'issue du jeu, un gain selon les objectifs visés (par exemple, +10 en cas de victoire , −10 en cas de défaite). Il s'agit donc d'un jeu à somme nulle. Cette catégorie de jeu peut être résolue en suivant notamment le principe du *minimax* [Fan53] : on déroule des simulations de jeu et on choisit à

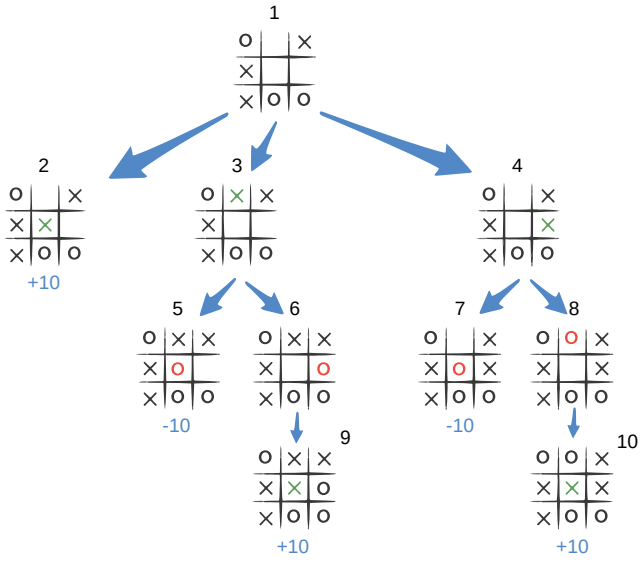


FIGURE 1.1 – Jeu du morpion, un exemple de déroulement de l'algorithme *Minimax* [Fan53] utilisé pour résoudre des jeux compétitifs [1]

1. Figure adaptée de `https://www.neverstopbuilding.com/blog/minimax`

chaque tour les stratégies qui minimisent les pertes en supposant que l'adversaire cherche lui, à les maximiser pour son propre adversaire.

Dans le monde du sport, on peut imaginer également de nombreuses problématiques liées à la théorie des jeux. Hormis les aspects directement liés à la pratique (par exemple de quel côté plonger pour le gardien de but lors d'un penalty (Table 1.4)), on peut penser aux choix stratégiques tels que la simulation d'une faute ou la tentation du dopage. Ces deux derniers cas s'apparentent d'ailleurs à la catégorie des dilemmes sociaux. En effet, il peut être risqué d'être seul à jouer honnêtement (risque de carton rouge à la place ou à cause d'un joueur qui simule ou un déclassement dans le cas du dopage car entourés de sportifs meilleurs). Il est aussi tentant d'être malhonnête (obtention d'un penalty ou gain de performance grâce au dopage). Cela peut facilement conduire à un comportement collectif malhonnête, qui est évidement regrettable (beauté du jeu, risques pour la santé, etc).

Les problèmes de circulation routière sont également sujets à des problèmes de choix entre intérêts personnels et collectifs. Hormis les questions de courtoisie au volant (laisser passer un véhicule qui ne peut s'engager, etc), on peut naturellement envisager les situations liées à la congestion qui sont très bien illustrées par le paradoxe de Braess [Fra81, TW00]. Dans la Figure 1.2, un aéroport est relié au centre-ville par deux routes. La première passe par le parc et comporte une portion courte mais sujette aux embouteillages (le temps dépend donc du trafic n de voitures, disons par exemple une durée égale à $n/100$ minutes) puis d'une portion plus longue mais à durée fixe de 35 minutes. La seconde route passe par l'hôpital, elle est similaire mais inversement disposée. Il se trouve que 3000 voitures font le trajet. Comme les deux alternatives sont équivalentes, les voitures choisissent uniformément l'un ou l'autre des trajets qui durent donc pour chaque automobiliste : $1500/100 + 35 = 50$ minutes.

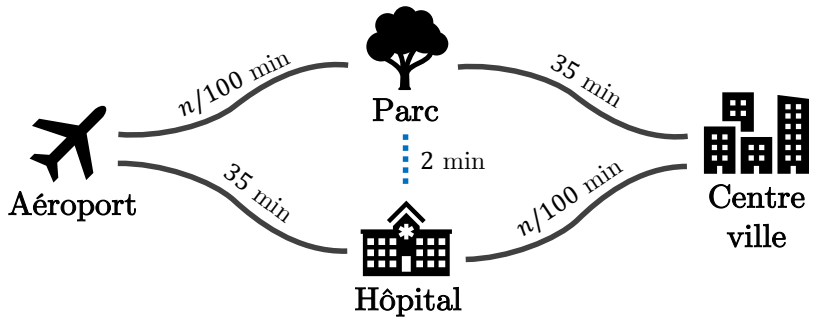


FIGURE 1.2 – Paradoxe de Braess

Cependant, le maire de la ville souhaitant fluidifier le trafic, décide de faire construire une route pour relier le parc et l'hôpital. Cette route dont le trajet dure inconditionnellement 2

minutes permet de relier les deux portions dans lesquelles chaque voiture mettait initialement 15 minutes. Chaque conducteur établit alors que cela reviendrait à une durée globale de 32 minutes au lieu de 50 et choisit alors cette solution, ce qui conduit à augmenter le trafic et porte pour tous la durée totale à $3000/100 + 2 + 3000/100 = 62$ minutes. Cette pénalité regrettable est appelée le prix de l'anarchie [CK05] et montre qu'il est parfois nécessaire d'imposer le choix de la coopération (ici, en condamnant la route de 2 minutes).

Dans un tout autre contexte, on peut envisager l'univers diplomatique. Ce n'est pas un hasard si la théorie des jeux a été développée notamment au cours de la guerre froide. En effet, les relations internationales sont un exemple crucial en ce qui concerne les décisions et les dilemmes. Par exemple, dans un contexte de course à l'armement, le choix de consacrer un budget à l'arsenal nucléaire ou bien à l'éducation et la culture est un vrai dilemme. D'un côté, une situation pacifique et vertueuse, de l'autre, des tensions explosives, et au milieu la crainte d'être vulnérable face à un pays armé, ce qui pousse alors à la surenchère.

Dans la politique, on rencontre également des situations de dilemmes. D'abord, le faible poids d'un vote individuel pousse certains citoyens à l'abstention : si cette stratégie est choisie par une majorité, c'est alors une minorité qui est en charge du choix démocratique. Mais, on peut également envisager le problème de rapprochement de candidats pour optimiser les chances de victoire dans un bord politique (voir Table 1.12).

Enfin, les problèmes de société liés aux efforts collectifs comme évoqué en introduction à propos des problèmes environnementaux ou de pandémies. L'effort individuel est pour la majorité perçu comme une punition ou peu motivant, alors que l'effort collectif peut parfois mener à des issues plus vertueuses. Ces problématiques s'apparentent à ce qu'on appelle la tragédie des biens communs [Har68]. Ces situations sont basées sur la confiance : les premiers à faire des efforts exigent d'être suivis, tandis que les réticents attendent des autres qu'ils se dévouent en premier pour les suivre ; une communication bien choisie est donc fondamentale dans ces situations.

1.1.3 Notions abordées dans ce chapitre

L'objectif de ce chapitre n'est pas d'être exhaustif sur les concepts de théorie des jeux, mais d'introduire formellement le concept des dilemmes sociaux. Il convient donc de définir quelques notions fondamentales de formalisme pour bien définir le concept de dilemme social ainsi que ses variantes.

1.2 Notions générales de la théorie des jeux

Comme précisé en introduction, la théorie des jeux a pour objectif principal de formaliser les comportements d'acteurs rationnels qui cherchent à maximiser leurs gains. Ce formalisme s'accompagne donc de définitions de concepts que nous apportons dans cette section. Nous allons donc définir un jeu sous forme normale, aborder le concept de dominance de stratégie ainsi que le concept qui y est lié : l'équilibre de Nash, une des clés de voûte de la théorie des jeux. Par ailleurs, nous verrons ensuite comment sont adaptées les variantes des jeux, l'extension en stratégies mixtes, avec information partielle ou non, et en version répétée.

1.2.1 Définition d'un jeu sous forme normale

L'une des manières de définir un jeu est la forme normale. Ce formalisme est décrit dans la définition 1.1.

Jeu sous forme normale

Définition 1.1 *Un jeu sous forme normale est défini par :*

- *Un nombre de joueurs N*
- *Un ensemble de stratégies S_i pour chaque joueur $i \in [\![1, N]\!]$*
- *Une fonction de valuation $v_i : \prod_{j=1}^{N} S_j \to \mathbb{R}$ pour chaque joueur $i \in [\![1, N]\!]$*

Exemples de jeux : Nous proposons d'illustrer ce formalisme avec les jeux suivants. Ils impliquent deux joueurs qui ont chacun deux actions (ou stratégies) possibles. Les deux joueurs choisissent une stratégie de manière simultanée et reçoivent un gain selon la fonction de valuation. Les quatre gains possibles selon les choix des joueurs sont définis dans les tableaux suivants.

	Opéra	Football
Opéra	$(3, 2)$	$(0, 0)$
Football	$(0, 0)$	$(2, 3)$

Table 1.1 – Bataille des sexes

	Rappeler	Attendre
Rappeler	$(-1, -1)$	$(1, 1)$
Attendre	$(1, 1)$	$(0, 0)$

Table 1.2 – Coupure d'un appel téléphonique

La bataille des sexes (Table 1.1) est un jeu impliquant un couple qui doit décider d'une activité pour la soirée. Ils n'ont pas exactement les mêmes goûts mais n'ont aucune envie de se séparer. Il faut donc faire des concessions. Dans ce jeu, l'un préfère le football et l'autre l'opéra. Les deux stratégies possibles sont le choix de l'une ou l'autre des activités à choisir simultanément. Un choix différent implique un gain mutuel nul car ils ne passent pas la soirée ensemble. Dans le cas d'un même choix, ils gagneront un gain de 3 ou 2 selon que leur activité favorite ait été choisie

	Raisonnable	Fort
Raisonnable	$(1, 1)$	$(-2, 2)$
Fort	$(2, -2)$	$(0, 0)$

TABLE 1.3 – Volume sonore entre voisins aux goûts musicaux différents

	Gauche	Droite
Gauche	$(1, -1)$	$(-1, 1)$
Droite	$(-1, 1)$	$(1, -1)$

TABLE 1.4 – Jeu du penalty : choix du gardien (à gauche) et du tireur (en haut)

ou non. Le second jeu (Table 1.2) modélise une situation assez agaçante (du moins à l'époque des traditionnels appels téléphoniques), où lors d'une coupure, il faut choisir entre rappeler immédiatement ou compter sur l'autre au risque d'attendre bêtement pour rien (gain mutuel de 0). Ou bien pire, perdre encore plus de temps en tombant mutuellement sur le répondeur (gain mutuel de -1). Le jeu des voisins (Table 1.3) désigne une variante du dilemme du prisonnier que nous verrons plus loin : deux voisins ont des goûts musicaux radicalement différents et peuvent régler le volume sonore de leur musique. Le savoir-vivre leur dicterait normalement de choisir chacun un volume raisonnable pour apprécier séparément leur musique (1 chacun). Cependant, il est tellement plus tentant de profiter en augmentant le volume (obtenir 2 au lieu de 1). Le voisin au faible volume va subir la situation (gain de -2) et donc préférer à son tour augmenter le volume, ce qui conduit à une situation un peu cacophonique sans réel plaisir (0 chacun). Enfin, le dernier jeu (Table 1.4) illustre une situation de compétition avec un gagnant et un perdant : lors d'un penalty, le tireur et le gardien doivent choisir quasi-simultanément de quel côté tirer ou plonger.

On peut catégoriser les jeux selon la définition des gains :

- **Jeu à intérêt commun** : les gains obtenus par les joueurs sont identiques ($\forall\ (i, j) \in [\![1, N]\!]^2\ ,\ v_i = v_j$). Par exemple, le jeu de la coupure téléphonique.
- **Jeu à somme nulle** : comme son nom l'indique, la somme des gains des joueurs est nulle. En d'autres termes, les joueurs sont en pure opposition : s'il y a un gagnant, il y a un ou plusieurs perdants. A deux joueurs, ce que l'un gagne, l'autre le perd. Le jeu du penalty est un jeu à somme nulle.
- **Jeu à somme non nulle** : il peut exister des jeux où la somme des gains est non nulle pour au moins une des issues. Par exemple, dans le jeu des voisins, la situation (Raisonnable, Raisonnable) est à somme positive (une stratégie gagnant-gagnant). A l'inverse, (Fort, Fort) est à somme négative (une issue préjudiciable).

1.2.2 Stratégies dominantes et dominées

Il convient de s'intéresser aux stratégies que chaque joueur peut choisir. On rappelle que chaque joueur est purement rationnel et qu'il agit indépendamment des autres au sens où il décide seul de sa stratégie et uniquement dans son propre intérêt. Dans la suite, nous dénotons un jeu sous forme normale $\mathcal{G} = (N, S, v)$ avec les notations de la défition 1.1 *i.e.* où N est le nombre de joueurs, S l'ensemble des stratégies des joueurs $S = (S_1, ..., S_N)$ et v les fonctions de gains des joueurs $v = (v_1, ..., v_N)$. Par ailleurs, nous notons l'ensemble des stratégies jointes des joueurs opposant le joueur i :

$$S_{-i} = \prod_{j \neq i} S_j$$

On définit alors le concept de stratégies dominées et dominantes comme suit.

Stratégies dominées et dominantes

Définition 1.2 *: Une stratégie $\hat{s}_i \in S_i$ du joueur i est dite dominée s'il existe une stratégie $t_i \in S_i$:*

$$\forall s_{-i} \in S_{-i}, \quad v_i(\hat{s}_i, s_{-i}) \leq v_i(t_i, s_{-i})$$

Définition 1.3 *: Une stratégie $s_i^* \in S_i$ du joueur i est dite dominante si :*

$$\forall s_i \neq s_i^* \in S_i, \quad \forall s_{-i} \in S_{-i}, \quad v_i(s_i^*, s_{-i}) \geq v_i(s_i, s_{-i})$$

Exemple : Dans le jeu des voisins (Table 1.3), la stratégie de mettre le son plus fort est dominante, car peu importe quelle stratégie choisit le voisin, augmenter le volume rapporte plus.

1.2.3 Equilibre de Nash

Nous considérons un jeu $\mathcal{G} = (N, S, v)$ avec les notations de 1.5. Si chaque joueur choisit une stratégie dominante, alors on dit que le profil de stratégies constitue un équilibre de Nash. Nous pouvons le définir formellement comme suit.

Equilibre de Nash

Définition 1.4 *: Un profil de stratégies $s^* = (s_1^*, ..., s_N^*) \in S$ est un équilibre de Nash si :*

$$\forall i \in [\![1, N]\!], \quad \forall s_i \in S_i, \quad v_i(s_i^*, s_{-i}^*) \geq v_i(s_i, s_{-i}^*)$$

Un équilibre de Nash n'étant pas nécessairement unique, on note NE($\mathcal{G}$) l'ensemble des équilibres de Nash du jeu $\mathcal{G}$.

Exemple : Dans l'exemple des voisins (Table 1.3), il y a un équilibre de Nash qui est de choisir mutuellement un fort volume sonore. En effet, changer sa décision seul, serait perdant. Dans le jeu de la coupure téléphonique (Table 1.2), il y a deux équilibres de Nash qui sont les deux situations où l'un des joueurs a rappelé tout de suite et l'autre a attendu.

1.2.4 Extension mixte d'un jeu sous forme normale

Les jeux précédemment définis sont à stratégies déterministes. Il est possible d'étendre le concept de manière stochastique. C'est ce qu'on appelle les stratégies mixtes.

Stratégies mixtes

Définition 1.5 *Soit un jeu sous forme normale $\mathcal{G} = (N, S, v)$. On définit une stratégie mixte pour le joueur i comme une distribution de probabilités sur S_i (l'ensemble des stratégie de i).*

On note $\Delta(S_i)$ l'ensemble des stratégies mixtes du joueur i. Les stratégies classiques $s_i \in S_i$ vues plus haut sont appelées stratégies pures, et sont simplement les stratégies mixtes jouant s_i avec la probabilité 1.
Pour étendre la définition d'un jeu sous forme normale en stratégies mixtes, il convient d'apporter quelques définitions et de modifier quelques notations.
Pour commencer, on note : $\Sigma_i = \Delta(S_i)$ l'ensemble des stratégies mixtes du joueurs i, $\Sigma = \prod_{j=1}^{N} \Sigma_j$ l'ensemble des stratégies mixtes du jeu et $\Sigma_{-i} = \prod_{j \neq i} \Sigma_j$ l'ensemble des stratégies mixtes des joueurs différents de i. On désigne par $\sigma_i \in \Delta(S_i)$ une stratégie mixte de i que l'on décrit avec un vecteur de probabilités sur S_i. Par exemple si $S_i = \{s_{i,1}, s_{i,2}, s_{i,3}\}$, le joueur i peut jouer $\sigma_i = (0, \frac{1}{2}, \frac{1}{2})$ qui consiste à jouer $s_{i,2}$ avec probabilité $\frac{1}{2}$ ou $s_{i,3}$ avec probabilité $\frac{1}{2}$ (et ne jamais jouer $s_{i,1}$).

Si chaque joueur j joue la stratégie mixte σ_j, alors la stratégie $s = (s_1, ..., s_N)$ survient avec probabilité $\prod_{j=1}^{N} \sigma_j(s_j)$. Par conséquent, on peut donc définir la nouvelle fonction de valuation (de gains) d'un profil de stratégies mixtes :

$$\mu_i(\sigma) = \sum_{s \in S} \left[\prod_{j=1}^{N} \sigma_j(s_j) \right] v_i(s)$$

On définit alors l'extension mixte d'un jeu sous forme normale :

Jeu sous forme normale en stratégies mixtes

Définition 1.6 *: l'extension en stratégies mixtes du jeu sous forme normale* (N, S, v) *est le jeu sous forme normale* (N, Σ, μ) *avec :*

- $\Sigma = \prod_{j=1}^{N} \Delta(S_j)$
- $\mu_i(\sigma) = \sum_{s \in S} \left[\prod_{j=1}^{N} \sigma_j(s_j) \right] v_i(s)$

Exemples de jeux :

	Gauche	Droite
Gauche	$(1, -1)$	$(-1, 1)$
Droite	$(-1, 1)$	$(1, -1)$

Table 1.5 – Jeu du penalty, choix du gardien (à gauche) et du tireur (en haut)

	Pierre	Feuille	Ciseaux
Pierre	$(0, 0)$	$(-1, 1)$	$(1, -1)$
Feuille	$(1, -1)$	$(0, 0)$	$(-1, 1)$
Ciseaux	$(-1, 1)$	$(1, -1)$	$(0, 0)$

Table 1.6 – Le jeu Pierre-Feuille-Ciseaux

On admet que les équilibres de Nash en stratégies mixtes sont $(\frac{1}{2}, \frac{1}{2})$ pour le jeu du penalty et $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ pour le jeu du Pierre-Feuille-Ciseaux *i.e.* jouer, avec égale probabilité, un des choix possibles.

1.2.5 Optimum de Pareto

L'optimum de Pareto peut être en certains termes considéré comme un optimal en matière d'intérêt commun. Il est défini comme une situation dans lequel aucune stratégie ne peut être changée sans diminuer le gain d'un autre joueur.

Optimum de Pareto

Définition 1.7 *: On définit la préférence de Pareto* $>_P$ *:*

$$(s_1, ..., s_N) >_P (s'_1, ..., s'_N)$$
$$ssi \quad \forall i \in [\![1, N]\!], \quad v_i((s_1, ..., s_N)) \geq v_i((s'_1, ..., s'_N))$$

Exemple : Dans le jeu des voisins (Table 1.3), l'issue (Raisonnable, Raisonnable) est un optimum de Pareto car si un des acteurs change sa décision, ce sera au détriment de l'autre.

1.2.6 Jeux à information incomplète

Dans les parties précédentes, l'hypothèse d'information complète était faite : chaque joueur connaît les actions possibles de chacun des autres joueurs ainsi que les fonctions de valuation associées pour chaque joueur. Or, il peut arriver qu'un ou plusieurs joueurs ne connaissent pas une information sur un ou plusieurs joueurs, leur empêchant alors d'en déduire les fonctions de gains de chacun. Il est possible de formaliser ce genre de situations sous la forme d'un jeu bayésien dont la définition est apportée par [Har67, KM97] :

Jeu bayésien

Définition 1.8 *: Un jeu bayésien est défini comme un jeu* $\mathcal{G} = (N, \Omega, p, \langle A_i, u_i, T_i, \tau_i \rangle_{i \in [\![1,N]\!]})$ *avec :*

- N *le nombre de joueurs*
- Ω *un ensemble d'états (avec* p *la distribution de probabilité sur* Ω*)*
- A_i *l'ensemble des actions du joueur* i *(on note* $A = A_1 \times ... \times A_N$*)*
- T_i *l'ensemble des types du joueur* i *(qui est donné selon le contexte par la fonction* $\tau_i : \Omega \to T_i$ *; on note* $T = T_1 \times ... \times T_N$*)*
- $u_i : T \times A \to \mathbb{R}$ *la fonction de gain pour chaque joueur* i

Une stratégie pure pour un joueur i dans un jeu bayésien est alors une fonction $s_i : T_i \to A_i$ et une stratégie mixte est une fonction $\sigma_i : T_i \to \Delta(A_i)$.

Exemple : Le jeu du shérif est un exemple simple de jeu bayésien. Le jeu se décrit comme suit. Un shérif fait face à un homme armé, dont il ne connaît pas ses intentions. Il peut s'agir d'un simple civil "innocent" ou bien d'un criminel. Tout comme l'homme qui lui fait face, il peut alors décider de tirer ou non. Les gains obtenus par chacun dépendront des choix respectifs ainsi que de la nature de l'homme armé. Le shérif ne connaît pas la nature du suspect et sa priorité est la légitime défense. Il tirera si le suspect tire et ne tirera pas si le suspect ne tire pas (même s'il est criminel). Quant au suspect, s'il est criminel, il tirera quoiqu'il arrive (même si le shérif ne tire pas) ; et s'il est innocent, il ne tirera pas (même si le shérif tire). Deux tables des gains peuvent alors résumer cette situation selon les deux possibilités : que le suspect soit innocent (Table 1.7a) ou criminel (Table 1.7b).

En aparté, un jeu très intéressant assez similaire, est celui du truel de Simon Singh [SR97] : M. Blanc, M. Gris et M. Noir se provoquent en truel. Ils tirent à tour de rôle sur l'un des autres opposants jusqu'à ce qu'il n'en reste plus qu'un. M. Noir atteint sa cible avec la probabilité $\frac{1}{3}$, M. Gris avec la probabilité $\frac{2}{3}$ tandis que M. Blanc ne manque jamais sa cible. Pour plus d'équité, M.

	Tirer	Pas Tirer
Tirer	$(-3,-1)$	$(-1,-2)$
Pas Tirer	$(-2,-1)$	$(0,0)$

(a) Suspect innocent

	Tirer	Pas Tirer
Tirer	$(0,0)$	$(2,-2)$
Pas Tirer	$(-2,-1)$	$(-1,1)$

(b) Suspect criminel

TABLE 1.7 – Jeu du shérif : selon le cas où le suspect est innocent ou criminel, les gains changent. Choix du shérif en haut et celui du suspect à gauche.

Noir a l'honneur de commencer, puis M. Gris suivi par M. Blanc et ainsi de suite jusqu'à la mort de deux truellistes. La question qui se pose est : Que doit commencer par faire M. Noir pour optimiser ses chances de survie ? Viser M. Gris ou bien M. Blanc ? [2]

1.2.7 Jeux répétés

Jusqu'à présent, nous considérions les jeux comme des situations à une seule étape. Il est possible de les étendre en version itérée. Les joueurs jouent alors plusieurs étapes de jeu au cours desquelles ils choisissent simultanément une des stratégies disponibles de la version en un coup simple. Ils obtiennent alors à chaque étape les gains correspondant à la version simultanée.

Prenons un jeu sous forme normale que l'on souhaite répéter : $\mathcal{G} = (N, (A_i)_{i\in[\![1,N]\!]}, (v_i)_{i\in[\![1,N]\!]})$. Les ensembles de stratégies S_i ont été remplacés par des ensembles d'actions A_i finis (avec $A = A_1 \times ... \times A_N$) afin de faire la distinction entre jeu simultané et jeu répété. Le principe est qu'à chaque étape t, chaque joueur i choisit indépendemment et simultanément une action a_i^t pour former l'action jointe $a^t = (a_1^t, ..., a_N^t)$. Ils reçoivent alors un gain égal à $v_i(a^t)$.

Pour définir le jeu répété sous forme normale (en stratégies mixtes car c'est le cas plus général), nous commençons par définir une stratégie mixte $\sigma_i \in \Sigma_i$ du joueur i par une fonction stochastique qui associe à chaque étape t et l'historique des actions jointes $(a^1, ..., a^{t-1})$, un choix dans A_i : $\sigma_i : \mathbb{N} \times A^{\mathbb{N}} \to \Delta(A_i)$. On a donc l'ensemble des stratégies $\Sigma = \prod_{i=1}^{N} \Sigma_i$. Il reste à définir la fonction de gain en version répétée. Il existe principalement deux manières de calculer le gain total. On peut premièrement le définir comme le gain moyen au cours du jeu (Définition 1.9). Cependant, dans de nombreuses situations, on peut accorder plus de valeur au gain actuel qu'aux gains futurs (par exemple, on préfère 1 euro maintenant que demain). C'est pourquoi il est courant dans la littérature de proposer une formulation de gain actualisé grâce un coefficient $\delta \in [0, 1[$ qui atténue le gain futur (Définition 1.10).

On peut alors formaliser la version répétée du jeu $\mathcal{G}$ par les définitions 1.9 ou 1.10 suivantes.

2. Réponse : tirer en l'air... Explications : `https://www.soler7.com/IFAQ/TruelA.html`

Jeu répété

Soit le jeu $\mathcal{G} = (N, (A_i)_{i\in[\![1,N]\!]}, (v_i)_{i\in[\![1,N]\!]})$. Avec les notations détaillées plus haut, on note $\mathbb{E}_\sigma$ l'espérance associée à stratégie mixte σ.

Définition 1.9 *: On définit le jeu répété de $\mathcal{G}$ en T étapes par le jeu sous forme normale $\mathcal{G}_T = (N, (\Sigma_i)_{i\in[\![1,N]\!]}, (g_i^T)_{i\in[\![1,N]\!]})$ avec :*

$$g_i^T(\sigma) = \mathbb{E}_\sigma\left[\frac{1}{T}\sum_{t=1}^{T} v_i(a^t)\right] \tag{1.2}$$

Définition 1.10 *: On définit le jeu escompté au taux $\delta \in [0,1[$ le jeu sous forme normale $\mathcal{G}_\delta = (N, (\Sigma_i)_{i\in[\![1,N]\!]}, (g_i^\delta)_{i\in[\![1,N]\!]})$ avec :*

$$g_i^\delta(\sigma) = \mathbb{E}_\sigma\left[(1-\delta)\sum_{t=1}^{\infty} \delta^{t-1} v_i(a^t)\right] \tag{1.3}$$

Remarques et exemples : Dans les définitions 1.9 et 1.10, $\frac{1}{T}$ et $(1-\delta)$ jouent le rôle de coefficient de normalisation de sorte que la somme des poids accordés à chaque étape soit de 1. Remarquons que dans le jeu $\mathcal{G}_\delta$, plus δ est faible, moins on accorde d'importance au futur. Ainsi, dans le jeu des voisins (Table 1.3), avec la version escomptée au taux δ (Définition 1.10), si $\delta < 1$, alors un voisin préférera un fort volume mutuel tous les soirs c'est-à-dire avec des gains successifs $(0, 0, 0, 0, ...)$ plutôt qu'une entente en alternant Raisonnable/Fort les jours où il se dévoue $(-2, +2, -2, +2, ...)$.

1.3 Les dilemmes sociaux

Dans cette section, nous allons introduire la notion des dilemmes sociaux : un certain type de jeu qui est très majoritairement traité dans ce manuscrit. Nous commençons par définir le concept général avant de se pencher sur la version la plus connue : le dilemme du prisonnier. Ensuite, nous définissons ses variantes, à savoir, la version itérée, ou bien continue ainsi que ses modèles à plus de deux joueurs.

1.3.1 Définition de base

Bien que le dilemme du prisonnier ait été introduit et formalisé bien avant [RCO65], une définition plus élargie des dilemmes sociaux est proposée par [MF02] . Dans cette définition plus

générale, on suppose qu'il existe deux types d'incitation poussant un joueur à choisir la défection : la crainte d'être exploité et la tentation d'exploiter.

Un dilemme social se définit par une situation impliquant deux joueurs qui peuvent choisir simultanément une des deux stratégies possibles entre la coopération ou la défection. Les quatre gains (symétriques) des quatre issues sont donnés par la Table 1.8.

	Coopération	Défection
Coopération	(R, R)	(S, T)
Défection	(T, S)	(P, P)

TABLE 1.8 – Gains dans un dilemme social à deux joueurs

R désigne la récompense de la coopération mutuelle (R pour *Reward*), P est la punition d'avoir choisi la défection mutuelle (P pour *Punishment*), S représente le risque de se faire exploité (S pour *Sucker*) et T désigne la tentation d'exploiter un coopérateur (T pour *Temptation*).

Compte tenu de ces notations, on apporte la définition suivante d'un dilemme social.

Dilemme social à deux joueurs

Définition 1.11 *: Un jeu $\mathcal{G} = (2, S, v)$ avec $S = \{Coopération, Défection\}$ et la fonction de gain v définie par les valeurs R, P, S, T indiquées par la table 1.8 est un dilemme social si les conditions suivantes sont vérifiées :*

1. *$R > P$: La coopération mutuelle est préférée à la défection mutuelle*
2. *$R > S$: La coopération mutuelle est préférée à l'exploitation*
3. *(a) Soit l'avidité (greed) i.e. $T > R$: L'exploitation d'un coopérateur est préférée à la coopération mutuelle*
   (b) Soit la crainte (fear) i.e. $P > S$: La défection mutuelle est préférée à l'exploitation par un défecteur
4. *$2R > S+T$: La coopération mutuelle est préférée à l'alternance Coopération/Défection*

Un dilemme social peut être défini par un jeu dans lequel il existe au moins un équilibre de Nash qui est Pareto-dominé, en d'autres termes quand l'intérêt personnel est préféré au détriment de l'intérêt commun.

1.3.2 Les trois catégories de dilemme social

D'après la définition 1.11, nous sommes en présence d'un dilemme social dès lors qu'au moins une des inégalités 3.a et/ou 3.b est vérifiée. Nous rencontrons alors trois types de dilemme social, selon si l'une, l'autre ou les deux inégalités sont vérifiées. Nous décrivons brièvement dans la suite ces trois cas de la littérature avec des exemples de jeux.

Stag Hunt

Le jeu *Stag Hunt* (chasse au cerf) est un dilemme social dans lequel les joueurs sont incités à la défection par la crainte (*fear*) d'être exploité (soit $P > S$). Le principe est le suivant : deux chasseurs choisissent un équipement adapté soit pour la traque du cerf (Coopération), soit du lièvre (Défection). Il faut être deux pour traquer le cerf mais seul pour chasser un lièvre. Comme le cerf rapporte plus qu'un lièvre, il y a présence d'un dilemme.

	Cerf	Lièvre
Cerf	$(4, 4)$	$(0, 3)$
Lièvre	$(3, 0)$	$(1, 1)$

TABLE 1.9 – Gains du jeu *Stag Hunt*

	Respect	Non respect
Respect	$(1, 1)$	$(-2, 0)$
Non respect	$(0, -2)$	$(-1, -1)$

TABLE 1.10 – Respect du confinement

EXEMPLES DANS LA VIE QUOTIDIENNE : Des exemples et analogies sont omniprésents dans les situations quotidiennes. Dès lors que nous faisons face à des situations dans lesquelles un effort collectif permet d'atteindre une issue plus bénéfique, nous sommes pris d'un doute et une crainte d'être seul à coopérer. En revanche, il n'y a pas d'avidité dans le sens où si l'effort collectif semble majoritairement adopté, il devient alors idéal d'y participer également. En d'autres termes : "Si mon partenaire ne fait pas d'effort, je n'en fais pas. S'il en fait, j'en ferai". L'illustration similaire la plus marquante du jeu *Stag Hunt* est probablement celle du confinement quasi-mondial du printemps 2020 (Table 1.10). En effet, bien que le respect des règles soit en grande partie dû à des fortes contraintes, on a pu assister néanmoins à une acceptation globale des règles de confinement puisqu'un grand nombre s'y soumettait. A l'inverse, les confinements qui ont suivi ont probablement été moins respectés en raison cette fois-ci de l'observation d'un moindre effort collectif et donc la crainte d'être seul à subir l'isolement. Dans ce genre de situations, la communication joue alors un rôle très important.

Chicken Game

A l'inverse du jeu *Stag Hunt*, le jeu *Chicken Game* modélise un dilemme social dans lequel cette fois-ci les joueurs préférent la défection par avidité et non par crainte (*i.e.* $T > R$). Ce jeu,

que l'on pourrait désigner par le "jeu de la poule mouillée", met en scène deux conducteurs qui se font face à haute vitesse (Table 1.11). Ils peuvent choisir de braquer (Coopération) ou tenir leur cap (Défection). Malgré une issue qui peut être fatale, aucun des deux n'a envie d'être le premier à céder.

	Braquer	Tenir
Braquer	$(3,3)$	$(1,4)$
Tenir	$(4,1)$	$(0,0)$

TABLE 1.11 – Gains du jeu *Chicken Game*, couramment utilisés dans la littérature

	Désistement	Maintien
Désistement	$(0,0)$	$(-1,5)$
Maintien	$(5,-1)$	$(-2,-2)$

TABLE 1.12 – Le problème du candidat unique dans un bord politique

EXEMPLES DANS LA VIE QUOTIDIENNE : On retrouve des situations similaires dès lors qu'il y a un intérêt à ce qu'un partenaire cède en premier ou se porte volontaire au détriment de son propre gain mais au profit d'un intérêt collectif. Ainsi, lors de chaque élection au scrutin majoritaire, se pose à chaque fois la question de possibles rapprochements de candidats d'un bord politique similaire. Aucun n'a envie de céder puisqu'il perd la possibilité (même infime) de remporter l'élection ou bien simplement l'opportunité de représenter son parti. Cependant, l'issue la pire est celle où aucun candidat du même bord n'est élu (Table 1.12).

Le dilemme du prisonnier

Le Dilemme du Prisonnier (DP) est le dilemme social qui fait intervenir à la fois l'avidité (*greed*) *i.e.* $T > R$ et la crainte (*fear*) *i.e.* $P > S$.

Véritable cas d'école de la théorie des jeux et des sciences économiques, le problème est introduit comme suit. Deux complices sont arrêtés pour un méfait et sont interrogés séparément. Ils ont la possibilité soit de nier les faits, soit d'avouer. En cas d'aveu mutuel, les deux coupables seront condamnés à une peine de 5 ans de prison. En revanche, dans le cas où les deux complices nient les faits, leur peine commune ne sera que de 3 ans (Table 1.13). Il convient alors de se taire pour éviter une peine plus forte. Cependant, les enquêteurs, astucieux, leur précisent individuellement, que dans le cas où ils avoueraient alors que leur complice continuerait à nier les faits, ils seraient alors libres mais que dans le cas inverse, ils se verraient infliger une peine bien plus forte de 10 ans. La communication étant impossible, la crainte d'une très forte peine ainsi que la tentation d'être libre incitent alors les deux complices à choisir mutuellement l'aveu. Cette issue du jeu, connue pour en être l'équilibre de Nash, n'est pas optimale d'où la situation de dilemme.

	Nier	Avouer
Nier	$(-3,-3)$	$(-10,0)$
Avouer	$(0,-10)$	$(-5,-5)$

TABLE 1.13 – Gains du Dilemme du Prisonnier d'origine : *i.e.* l'opposé des années de prison.

	Coopération	Défection
Coopération	$(3,3)$	$(0,5)$
Défection	$(5,0)$	$(1,1)$

TABLE 1.14 – Gains du Dilemme du Prisonnier, couramment utilisés dans la littérature : $S=0$, $P=1$, $R=3$, $T=5$

EXEMPLES DANS LA VIE QUOTIDIENNE : Comme déjà évoqués en introduction, les exemples sont nombreux : le devoir civique, le dopage dans le sport, le respect entre voisins etc.

1.3.3 Dilemme du prisonnier itéré

Comme évoqué en section 1.2.7, chaque jeu peut être étendu en version répétée. La version itérée du dilemme du prisonnier (*iterated prisoner's dilemma*, IPD) apporte de nombreuses propriétés intéressantes. En effet, par exemple la possibilité de représailles ou d'incitation à se faire confiance change radicalement les stratégies optimales. Pour étudier les stratégies possibles dans ce jeu, Robert Axelrod organise en 1981 un grand tournoi de Dilemme du Prisonnier Itéré (IPD) dont il résume les résultats et conclusions dans son ouvrage : *The Evolution of Cooperation* [AH81]. Compte tenu de l'approfondissement possible des stratégies dans ce type de jeu, la version itérée du dilemme du prisonnier est principalement celle qui est la plus étudiée [Fog93, DM95, BDM+01].

1.3.4 Dilemme du prisonnier continu

Le dilemme du prisonnier a été étendu en un dilemme du prisonnier continu dans lequel les deux joueurs, au lieu de choisir une action atomique entre *Coopération* et *Défection*, définissent leur choix par un degré de coopération continue $x \in [0,1]$. Avec 0 pour la défection totale et 1 pour la coopération totale. Dans le modèle de [Ver93, Ver98], les valeurs S, P, R, T sont interpolées en deux fonctions de gain continues à deux variables (Définition 1.12).

Dilemme du prisonnier continu

Définition 1.12 *Les gains v_A et v_B obtenus par des joueurs A et B qui choisissent respectivement des degrés de coopération $a \in [0,1]$ et $b \in [0,1]$ sont définis par :*

$$\begin{aligned} v_A(a,b) &= abR + a\bar{b}S + \bar{a}bT + \bar{a}\bar{b}P \\ v_B(a,b) &= baR + b\bar{a}S + \bar{b}aT + \bar{b}\bar{a}P \\ &\text{avec } \bar{x} = 1-x \end{aligned} \tag{1.4}$$

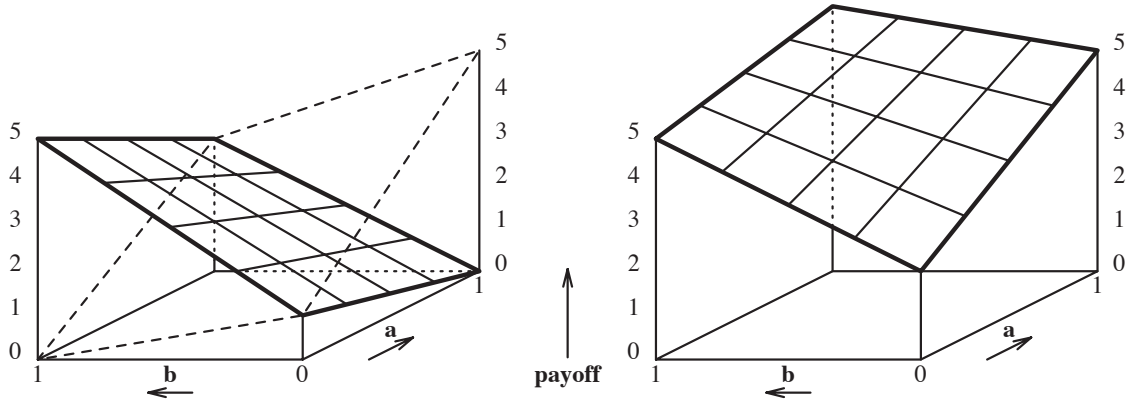


FIGURE 1.3 – Interpolation des gains du dilemme du prisonnier continu. Dans la figure de gauche, sont représentées les fonctions de gain indivivuel qui dépendent des choix de coopération a et b : en trait plein celle du joueur A et en pointillés celle (symétrique) du joueur B. La figure de droite correspond à la somme des deux fonctions de gains (Figure issue de [Ver98])

La figure 1.3 illustre l'interpolation des valeurs S, P, R et T du DP discret. Notons plusieurs choses : premièrement nous retrouvons les valeurs discrètes aux quatre coins du graphe $((0,0),(0,1),(1,0),(1,1))$. La somme des gains illustre également l'issue la plus vertueuse qui est un équilibre de Pareto (coordonnées $(1,1)$). Nous pouvons également vérifier qu'à toute valeur de b (resp. a), la fonction de gain du joueur A (resp. B) est décroissante en fonction de a (resp. b), ce qui correspond à l'incitation à défecter. Notons que dans le cas continu, il y a moins de sens à distinguer la crainte de A (quand b proche de 0) et son avidité (quand b proche de 1) puisqu'il s'agit d'une fonction continue sur $[0,1] \times [0,1]$.
Bien que la courbure soit peu visible sur la figure 1.3, la surface correspondante aux fonctions des gains est un paraboloïde hyperbolique qui se dégénère sur un plan uniquement dans le cas $R + P = S + T$. Plus précisément, il s'agit d'une surface réglée : si l'on fixe une des valeurs a ou b, la fonction restante à une variable est affine (donc représentée par une droite).

Ce type de jeu qui a fait l'objet d'études [WN99, KD02, LB07] permet de représenter des situations de dilemme de manière plus réaliste, comme nous l'avons fait dans notre méthode de partage de ressources [LGMLR20].

1.3.5 Le dilemme du prisonnier à N joueurs

Dans ce qui précède, le DP était considéré avec deux joueurs. Certaines versions du DP proposent d'étendre le concept à $N \geq 2$ joueurs. Ainsi, dans le modèle de [Ham73], les joueurs ont toujours deux actions possibles mais les récompenses de chaque joueur dépendent du nombre de joueurs qui ont coopéré. Certains modèles ont également proposé d'inclure au sein des agents

une structure de graphe [Ash07, LCS10]. Chaque arrête du graphe définit alors un DP classique. Nous y reviendrons dans le chapitre 4.

1.4 Stratégies gagnantes pour le dilemme du prisonnier itéré

Malgré la simplicité apparente de ce cas d'école, adopter une bonne stratégie face à un partenaire inconnu est loin d'être trivial. Nous nous intéressons ici à la version itérée du jeu (IPD). Dans ce contexte, un choix ponctuel a donc un impact sur la stratégie de l'autre joueur au cours des futures étapes. Brièvement, il est évident que la stratégie optimale sera différente selon que l'on soit face à un "lâche" qui avoue toujours ou bien un "dur" qui n'avoue jamais et qui cherchera à se venger. En effet, le caractère itératif change radicalement les issues du jeu en cela qu'il apporte la possibilité de représailles ou bien d'incitation à la confiance.

1.4.1 Les "bonnes" stratégies

Lors de son tournoi de IPD, Robert Axelrod a trié la soixantaine de stratégies qui lui avaient été soumises et les a triées à l'aide de méthodes évolutionnistes. En quelques mots, des individus issus des populations de stratégies jouent entre eux, les moins bons scores sont supprimés. Il résume ses conclusions dans son son ouvrage *The evolution of cooperation* [AH81]. Il déduit des stratégies restantes que quelques critères semblent utiles et nécessaires pour être une bonne stratégie dans un IPD. Il en liste quatre :

1. *Niceness* : Une bonne politique doit être gentille, c'est-à-dire, avoir une propension à coopérer *a priori*, au début par exemple.
2. *Forgiveness* : La politique doit être indulgente, non rancunière. Par exemple, après une défection d'un partenaire, il faut être capable de revenir à la coopération si le partenaire est de nouveau volontaire.
3. *Retaliation* : Elle doit être capable de représailles, pour pénaliser une défection adverse afin de ne pas se faire exploiter.
4. *Clarity* : Enfin, elle doit être idéalement simple, facilement compréhensible par un adversaire afin de garantir un accord mutuel.

1.4.2 Exemples de stratégies

Dans le tournoi d'Axelrod, certaines stratégies simples, que nous citerons dans la suite, sont introduites.

• Stratégies sans prise en compte de l'adversaire

Dans un premier temps, on peut imaginer des stratégies qui ne dépendent pas du comportement de l'adversaire. Parmi lesquelles on peut citer les suivantes :

Stratégie gentille : Appelée *Nice* en anglais, et que l'on pourrait désigner aussi par naïve, elle consiste simplement à toujours choisir la coopération.

Stratégie égoïste : Elle consiste à toujours choisir la défection.

Stratégie indécise : Elle consiste à alterner coopération et défection en commençant par choisir la coopération.

Stratégie indécise méfiante : Elle consiste à alterner coopération et défection en commençant par choisir la défection.

Stratégie lunatique : Appelée *Random* en anglais, elle consiste à agir de manière aléatoire.

• Stratégies avec prise en compte de l'adversaire

Il est cependant intuitif qu'il est important de prendre en compte le comportement de l'adversaire. En effet, on ne joue pas de la même manière face à un égoïste que face à un coopérateur. C'est pourquoi quelques stratégies ont été proposées :

Stratégie rancunière : Appelée *Grim* en anglais, elle consiste à coopérer jusqu'au moment où le partenaire choisit la défection, auquel cas on choisit pour le reste du jeu la défection.

Stratégie "Donnant-Donnant" : C'est la stratégie gagnante du tournoi d'Axelrod [AH81]. Cette stratégie s'est démarquée par sa simplicité et son efficacité. Il s'agit du Tit-for-Tat (TFT), introduit par Anatol Rapoport [RCO65].

Tit-for-Tat

Définition 1.13 *Dans un Dilemme du Prisonnier Itéré, la stratégie du Tit-for-Tat consiste à choisir à chaque étape :*

- *Au début du jeu : la coopération*
- *Puis : le choix précédent de l'opposant*

Stratégie du *Win-stay, lose-shift (WSLS)* : C'est une autre stratégie assez simple qui a été introduite plus tard, *Win–Stay, Lose–Shift* (WSLS) et qui s'inspire d'un apprentissage de type pavlovien [NS93].

Win-stay, lose-shift

Définition 1.14 *Dans un Dilemme du Prisonnier Itéré, la stratégie du Win–Stay, Lose–Shift (WSLS) consiste à chaque étape à :*

- *Au début du jeu, choisir la coopération*
- *Après, garder son choix si le partenaire choisit la coopération, le changer sinon.*

1.4.3 Simulations de tournois

Pour illustrer les stratégies définies plus haut, simulons quelques tournois sur quelques étapes (par exemple $T_{max} = 5$) en y opposant deux stratégies, notamment pour montrer l'intérêt d'un certain compromis de propriétés telles que la gentillesse, l'indulgence et la rancune dont fait preuve par exemple le TFT. Les gains des simulations sont basés sur la Table 1.8.

G(TFT)	0	1	1	1	1	Tot = 4
TFT	C	D	D	D	D	
Égoïste	D	D	D	D	D	
G(Égoïste)	5	1	1	1	1	Tot = 9

TABLE 1.15 – Gains dans un tournoi Tit-for-Tat (TFT) vs Égoïste

G(Naïf)	0	0	0	0	0	Tot = 0
Naïf	C	C	C	C	C	
Égoïste	D	D	D	D	D	
G(Égoïste)	5	5	5	5	5	Tot = 25

TABLE 1.16 – Gains dans un tournoi Naïf vs Égoïste

G(TFT)	3	0	5	3	0	Tot = 11
TFT	C	C	D	C	C	
Lunatique	C	D	C	C	D	
G(Lunatique)	3	5	0	3	5	Tot = 16

TABLE 1.17 – Gains dans un tournoi Tit-for-Tat (TFT) vs Lunatique

G(Rancunier)	3	0	5	5	1	Tot = 14
Rancunier	C	C	D	D	D	
Lunatique	C	D	C	C	D	
G(Lunatique)	3	5	0	0	1	Tot = 9

TABLE 1.18 – Gains dans un tournoi Rancunier vs Lunatique

Une bonne stratégie, qui gagne beaucoup en moyenne n'est pas nécessairement une stratégie qui gagne beaucoup de parties. En effet, par exemple la stratégie Égoïste gagne toutes ses parties dans le tournoi d'Axelrod mais va peu gagner en moyenne notamment contre la stratégie rancunière ou le TFT à la fin du tournoi. Cette dernière, gagnante du tournois n'a d'ailleurs strictement remporté aucune partie (tout au plus des parties nulles).

1.4.4 Tit-for-Tat continu

Pour adresser son formalisme de DP continu, [Ver98] introduit une première extension de Tit-for-Tat continu appelée *Damped Tit-for-Tat* (DTFT). Cette version de TFT est donc une fonction adaptée pour prendre en compte des degrés de coopération continus.

r-Damped Tit-for-Tat

Définition 1.15 *Soit un dilemme du prisonnier itéré et continu opposant les deux joueurs A et B jouant respectivement a_t et b_t à chaque étape t du jeu. Alors si A joue le DTFT, cela consiste à choisir la valeur a_t comme suit :*

$$a_t = \begin{cases} 1.0 & si\ t = 0 \\ r.1 + (1-r).b_{t-1} & si\ t > 0. \end{cases} \tag{1.5}$$

avec $r \in [0, 1]$

En quelques mots, ce TFT commence par une coopération pure puis reproduit le choix (continu) du partenaire (b_{t-1}) auquel est appliqué une pondération vers la coopération grâce au coefficient d'incitation $r \in [0, 1]$. On remarque en effet que si $b_{t-1} = 0$, alors $a_t = r$. Cela permet d'être plus incitatif (si $r \neq 0$).

1.5 Synthèse

La théorie des jeux, branche relativement récente des mathématiques, a pour objectif de formaliser les comportements d'agents rationnels. Nous avons défini les concepts les plus importants dans l'objectif de définir formellement la catégorie de jeux sur lesquels s'articule cette thèse, à savoir les dilemmes sociaux. Ceux-ci se déclinent en trois catégories qui sont les jeux de type *Stag Hunt*, *Chicken game* et le dilemme du prisonnier (DP). C'est ce dernier qui retient notre attention et en particulier sa version itérée. En effet ce modèle est très adapté pour l'étude de stratégies dans les situations non-coopératives qui nous intéressent. Nous avons abordé également la version continue du DP qui a notamment pour objectif d'adresser des situations plus réalistes. Le prochain chapitre sera l'occasion de définir un autre type de dilemme social offrant plus de complexité et de réalisme. Il s'agit des dilemmes sociaux séquentiels dont les stratégies sont des politiques d'apprentissage par renforcement profond ou *Deep Reinforcement Learning* (DRL), ce qui offre de nombreuses possibilités de situations plus réalistes.

Chapitre 2

Apprentissage par renforcement et dilemmes sociaux complexes

Dans ce chapitre, nous nous intéressons à l'ajout de politiques d'apprentissage par renforcement (*Reinforcement Learning* (RL)) au sein des dilemmes sociaux. Depuis quelques années, avec l'essor du *Deep Reinforcement Learning*, le concept de dilemme social a en effet été étendu en dilemmes sociaux séquentiels (SSD). Les stratégies possibles entre les joueurs ne sont plus limitées à des actions simples (coopérer ou trahir). Elles peuvent prendre la forme de véritables politiques complexes dites de coopération ou de défection. Comme certaines de nos contributions concernent ces concepts, nous proposons dans ce chapitre de les définir. Pour commencer, après une courte introduction de l'apprentissage machine, nous formaliserons et détaillerons quelques techniques de RL qui seront par ailleurs utiles pour le chapitre 3. Enfin, nous finirons le chapitre en nous focalisant sur les définitions des SSD.

2.1 Le Machine Learning : une brève topologie

Le *Machine Learning* (ML) ou Apprentissage Machine en français désigne diverses techniques permettant grâce à des données ou de l'expérience, de déduire des modèles de connaissances ou bien apprendre à réaliser des taches automatisables. Parmi ces tâches, on peut citer la reconnaissance d'objets, la prédiction d'une certaine valeur temporelle, la détection d'émotion, ou bien la prise de décision en temps réel. Il est courant de distinguer les techniques de ML en trois grandes catégories que l'on détaille brièvement dans la section suivante. Nous détaillerons également le principe des méthodes neuronales qui est une technique d'approximation de modèle universel.

2.1.1 Les catégories d'apprentissage

Il est commun de distinguer trois grands types d'apprentissage : l'apprentissage supervisé, l'apprentissage non supervisé et pour finir l'apprentissage par renforcement.

Apprentissage supervisé

Nous pouvons formaliser le problème de l'apprentissage supervisé comme suit [Aze19]. Soit $\mathcal{X}$ un espace d'observations, et $\mathcal{Y}$ un espace d'étiquettes (aussi appelées cibles, ou *labels*). On dispose d'un ensemble de données $\mathcal{D} = \{(x^i, y^i)\}_{i=1,...,n} \subset \mathcal{X} \times \mathcal{Y}$. En d'autres termes, nous disposons de n observations x^i dont on connaît l'étiquette y^i. L'objectif principal de l'apprentissage supervisé est de déterminer une fonction $f : \mathcal{X} \to \mathcal{Y}$ qui permet de relier "au mieux" les données de $\mathcal{D}$, c'est à dire que pour toute donnée $(x^i, y^i) \in \mathcal{D}$, $f(x^i) \approx y^i$. Pour cela, il est commun d'introduire une fonction dite de coût (*loss function* en anglais) $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ telle que la valeur $L(y, f(x))$ est d'autant plus élevée que l'étiquette prédite $f(x)$ est éloignée de sa vraie valeur y. Le prédicteur recherché peut être alors communément mis sous la forme d'une fonction paramétrée f_θ dont on cherchera à déterminer les paramètres optimaux θ^* définis alors par :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{|\mathcal{D}|} \sum_{(x^i,y^i)\in\mathcal{D}} L(y^i, f_\theta(x^i)) \tag{2.1}$$

L'espace $\mathcal{X}$ des observations peut être continu (*e.g.* $\mathbb{R}^p$) ou bien discret. Il peut s'agir également d'un signal, d'un texte, ou bien même de structures complexes telles que des graphes. Quant à l'espace des étiquettes, il peut être :

— Continu (*i.e.* $\mathbb{R}^k$), on parle de régression

— Discret, on parle alors de classification qui peut être :

- Binaire : $\mathcal{Y} \equiv \{0, 1\}$
- Multi-classes : $\mathcal{Y} \equiv \{1, ..., C\}$ avec C le nombre de classes
- Multi-labels : $\mathcal{Y} \equiv \mathcal{P}(\{1, ..., C\})$ avec $\mathcal{P}(E)$ l'ensemble des parties de l'ensemble E

Exemple : Pour la reconnaissance d'images, l'ensemble des entrées est généralement $\mathcal{X} = [\![0, 255]\!]^{3\times L\times H}$ pour une image de taille $L \times H$, en couleurs RGB dont chacun des trois canaux est codée sur 8 bits. L'espace des cibles peut alors être soit binaire, par exemple $\mathcal{Y} =$ {Présence d'un chat, Absence de chat}. Il peut s'agir également d'un ensemble multi-classes, *e.g.* $\mathcal{Y} =$ {Chat, Chien, Lapin} ou bien d'un ensemble multi-labels $\mathcal{P}$({Chat, Chien, Lapin}) où $\mathcal{P}$ désigne l'ensemble des parties. Dans ce dernier cas, $y =$ {Chien} ou encore $y =$ {Chat, Chien} sont des exemples de cible.

Apprentissage non supervisé

L'apprentissage non supervisé diffère de l'apprentissage supervisé par l'absence d'étiquettes. Ainsi, on dispose seulement d'observations $x^i \in \mathcal{X}$. L'objectif devient la recherche d'une fonction sur $\mathcal{X}$ dont la nature et le rôle dépendent de l'objectif recherché.

Exemples : Il peut s'agir d'effectuer du *clustering* (*i.e.* le regroupement de données proches ou similaires) [JMF99], de la réduction de dimensions (*i.e.* diminuer le nombre de variables des données par rapprochement vectoriel)[Jol05]. Plus récemment, les techniques de traitement du langage naturel ont conduit auprès de corpus de texte massifs et non annotés à entraîner des modèles de langue, c'est-à-dire apprendre à s'exprimer comme un corpus donné, ce qui permet entre autres la modélisation, la prédiction et la génération de texte [DCLT18].

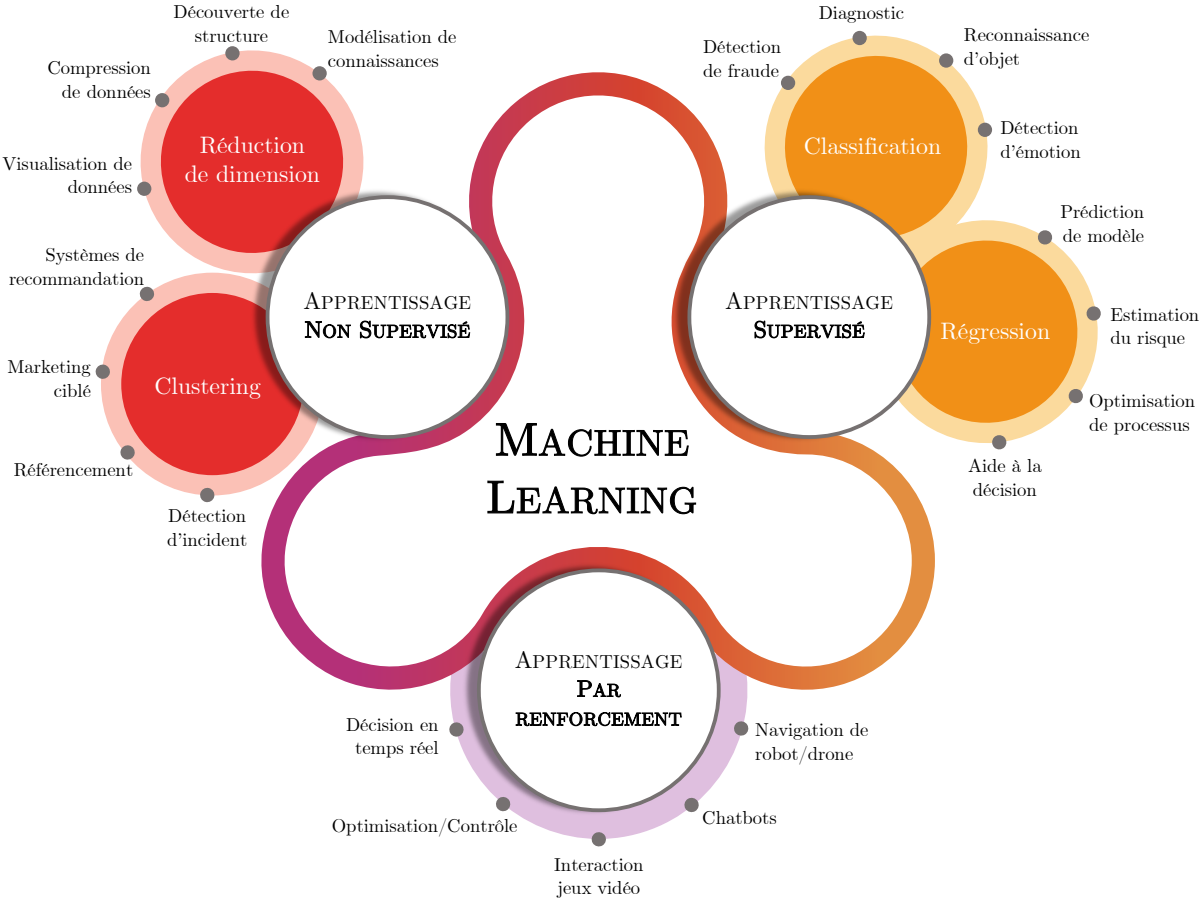


Figure 2.1 – Les trois grands types d'apprentissage : non supervisé, supervisé et par renforcement, avec quelques exemples d'utilisation [1].

1. Figure inspirée de `https://www.ironhack.com/en/data-analytics/what-is-machine-learning`, adaptée et traduite.

Apprentissage par renforcement (RL)

Cette catégorie diffère largement des deux autres types de ML. Dans ce nouveau paradigme, disposer de données n'est plus nécessaire. Un agent de RL interagit avec un environnement par essais et erreurs, il apprend à mieux se comporter et réagir avec pour objectif de maximiser une récompense moyenne. Nous reviendrons plus en détails dans la section 2.2.

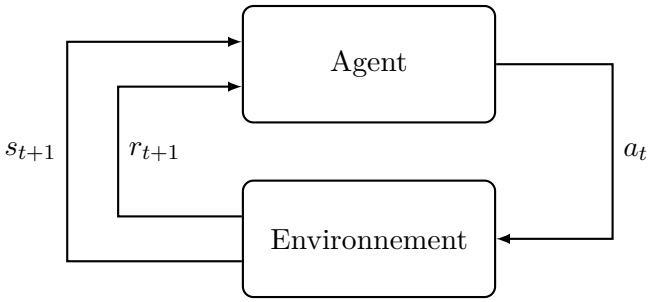


FIGURE 2.2 – Principe du RL : au cours du temps, un agent effectue des actions a_t face à un environnement, il observe alors un nouvel état s_{t+1} dans lequel il se trouve alors et reçoit une récompense r_{t+1} caractérisant l'optimalité de son action. Par étapes successives, l'agent apprend à améliorer son modèle de décision.

EXEMPLES : Le RL est par exemple utile dans la robotique [KBP13] notamment pour permettre à des mobiles à se déplacer de manière optimale [SK02], ou effectuer des mouvements complexes impossible à programmer manuellement [PS08]. Enfin, l'aide à la décision [Bot12] ou les jeux vidéos [Lev96, MKS+13] sont des bons exemples de l'utilité et la puissance du RL.

2.1.2 Les réseaux de neurones

Popularisés récemment grâce aux gains de capacités de calcul et par les réussites de taches complexes [GBC16], les réseaux de neurones artificiels tirent leurs origines du perceptron [Ros57] : inspirés du fonctionnement des neurones et des synapses du cerveau, ils permettent de modéliser un prédicteur universel à l'aide d'une composition de fonctions linéaires et non-linéaires.

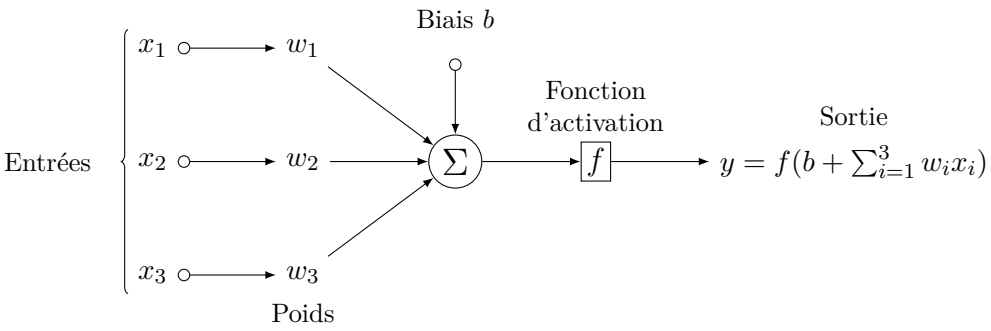


FIGURE 2.3 – Principe d'un neurone artificiel à 3 entrées

Brièvement, un perceptron est une succession de couches de neurones artificiels, dont les entrées de l'une sont les sorties de la précédente. Un réseau de neurones comporte une couche d'entrée qui accueille un vecteur d'entrée, une couche de sortie qui génère la sortie du réseau global, et enfin au moins une couche cachée de neurones artificiels.

Chaque valeur de neurone est calculée par la composition d'une fonction non linéaire et d'une fonction affine dont les variables sont les valeurs des neurones de la couche précédente (voir Figure 2.3). Les paramètres de chaque fonction affine sont les poids des synapses ainsi qu'un biais additionnel.

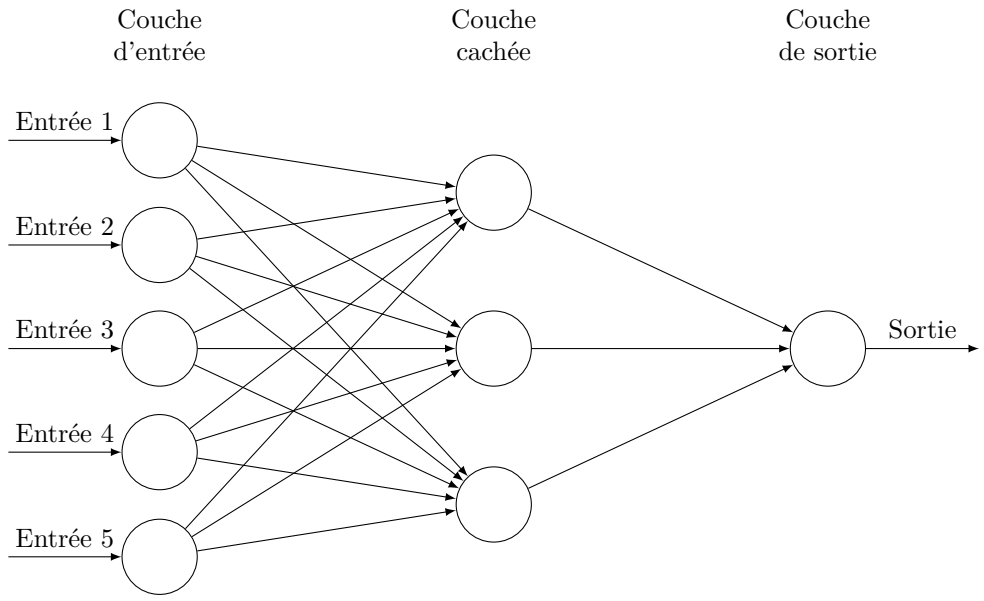


FIGURE 2.4 – Perceptron basique avec un vecteur d'entrée de dimension 5, une couche de sortie de dimension 1 et une couche cachée de 3 neurones. Il comporte donc 5×3 poids de synapses et 3 valeurs de biais pour la couche cachée, ainsi que 3 poids et 1 biais pour la couche de sortie. Au total : 22 paramètres.

Ainsi, un réseau de neurones peut être vu comme une fonction $f_\theta : x \to y$ dont les paramètres θ sont les poids et les biais des différentes couches. L'apprentissage de ces poids est effectué sur un ensemble de données disponibles, par des descentes de gradient au cours desquelles une succession de phases dite de *forward propagation* (pour obtenir la sortie prédite) et de *backward propagation* (pour obtenir le gradient de la fonction de coût) permet de minimiser l'erreur entre les valeurs prédites et les vraies valeurs.

2.2 Apprentissage par renforcement

Dans cette section, nous formalisons plus précisément la technique d'apprentissage par renforcement (*Reinforcement Learning* (RL)). Nous commençons par formaliser la tâche à accomplir notamment par la définition des processus de décision de Markov (*Markov Decision Process* (MDP)). Puis nous détaillerons le principe général du RL avant d'en expliciter quelques méthodes. Enfin, nous aborderons les extensions et variantes du RL, à savoir le *Deep Reinforcement Learning* (DRL) qui utilise des approximations de modèle à base de réseau de neurones ainsi que l'apprentissage multi-agents (*Multi-Agent Reinforcement Learning* (MARL)).

2.2.1 Processus de décision markovien

Commençons par formaliser les processus de décision markovien [Bel57].

Processus de décision markovien

Définition 2.1 *Un Markov Decision Process (MDP) est défini par le quadruplet* $(\mathcal{S}, \mathcal{A}, T, r)$ *où :*

- $\mathcal{S}$ *est un ensemble (dénombrable) d'états*
- $\mathcal{A}$ *est un ensemble (fini) d'actions*
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ *est une fonction stochastique de transition*
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *est une fonction de récompense*

Notons qu'une extension existe dans le cas où les états ne sont que partiellement observables. Il convient alors de rajouter :

- Ω : un ensemble d'observations
- $O : \Omega \times S \to [0, 1]$: une fonction stochastique d'observation

Cette extension, désignée par le sextuplet $(\mathcal{S}, \mathcal{A}, T, r, \Omega, O)$ est appelée un processus de décision markovien partiellement observable (*Partially Observable Markov Decision Processes* (POMDP)). Dans la suite, pour plus de clarté, nous considérons la version totalement observable, en parlant d'état observé $s_t \in S$, mais en principe on devrait observer o_t avec la probabilité $O(o_t|s_t)$.

Durant un ensemble d'itérations (ou étapes) qu'on appelle un épisode, un agent interagit alors avec ce processus markovien (l'environnement qui lui est inconnu). A chaque étape t, il observe un état $s_t \in \mathcal{S}$, il effectue une action $a_t \in \mathcal{A}$. Après cette action, il reçoit une récompense $r_{t+1} = r(s_t, a_t)$, et l'environnement passe à l'état s_{t+1} selon la fonction T. La probabilité que l'environnement passe à l'état s' depuis un état s en choisissant l'action a est égale $T(s'|s, a)$.

L'objectif de l'agent est alors de déterminer comment choisir de manière optimale des actions de sorte à maximiser une récompense totale R_t (appelée *return*) qui est généralement considérée cumulée et atténuée par un facteur $\gamma \in [0, 1[$, définie par :

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... \tag{2.2}$$

Comme $\gamma \in [0, 1[$ et les récompenses sont supposées bornées, la somme R_t est finie. Cela permet de s'affranchir notamment de la durée d'un épisode, voire de considérer des épisodes infinis.

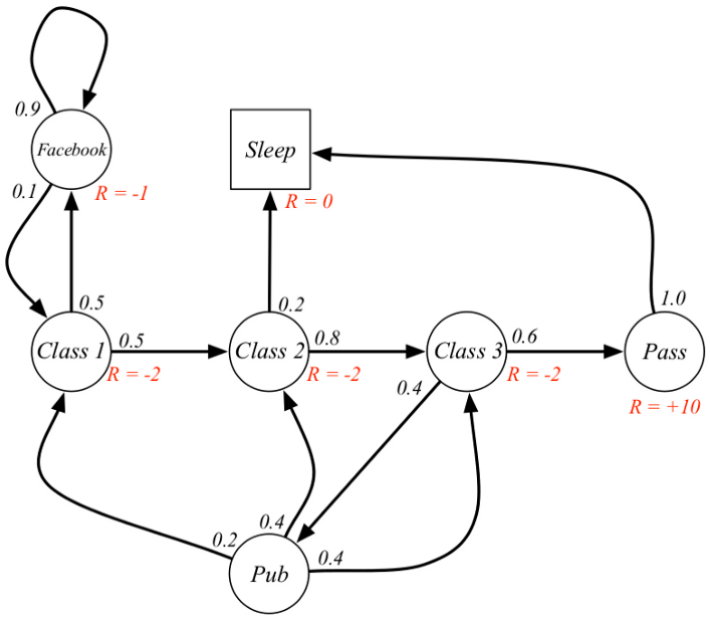


FIGURE 2.5 – Processus de Décision Markovien (MDP)[1]

2.2.2 Principe général et dilemme "exploitation/exploration"

Comme mentionné précédemment, le principe général d'un agent d'apprentissage par renforcement est d'apprendre à maximiser son gain total dans un MDP qui lui est inconnu. Son objectif est d'aboutir à ce qu'on appelle une politique, à savoir une fonction (notée usuellement π) qui permet de relier une observation à une action : $\pi : s \mapsto a$. Une politique notée π peut être :

— déterministe : $\pi : \mathcal{S} \to \mathcal{A}$, lorsque l'action optimale a est directement donnée par $a = \pi(s)$

— stochastique : $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, la probabilité de choisir l'action a est donnée par $\pi(a|s)$

1. Figure issue de `https://miro.medium.com/max/724/1*p5KQnP1rwTcXFooMFOn-TA.png`

L'agent de RL agit alors par essai et erreur jusqu'à obtenir une bonne connaissance de l'environnement et donc une politique de plus en plus optimale. Tout le long de l'apprentissage, un des enjeux de l'agent est de choisir efficacement entre tester de nouvelles combinaisons (exploration) et continuer à suivre la politique qu'il a jusqu'à là apprise (exploitation). C'est ce qu'on appelle le dilemme "exploitation/exploration".

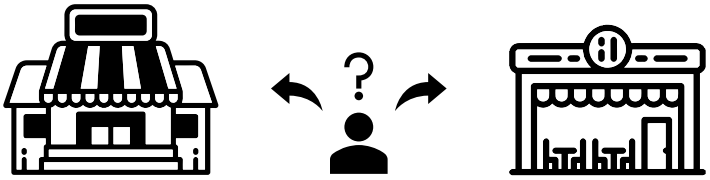


FIGURE 2.6 – Le dilemme "exploitation/exploration" : Faut-il expérimenter le nouveau restaurant qui vient d'ouvrir (exploration) au risque d'être déçu mais avec la possibilité d'être comblé, ou bien continuer sur une valeur sûre qui a fait ses preuves jusqu'à là (exploitation) ?

Une des méthodes simples pour traiter ce compromis est la méthode ϵ-*greedy* [SB+98]. Elle consiste à chaque itération à faire le choix de l'exploration avec une probabilité ϵ et le choix de l'exploitation avec une probabilité $1 - \epsilon$. Au cours de l'entraînement, on fait décroître ϵ de manière à prioriser davantage l'exploitation. D'autres manières plus sophistiquées existent comme l'utilisation de distributions de Boltzmann [BBS91] ou de marches aléatoires [NW89].

2.2.3 Catégories d'algorithmes

Il existe de nombreux algorithmes permettant d'entraîner les politiques de type RL. Il est courant de les catégoriser en trois grands types. Pour commencer, les méthodes dites *Value-based* consistent à utiliser des fonctions intermédiaires dites fonctions de valeur qui estiment une espérance de gain à partir d'un état ou d'un couple état/action. Cela permet ensuite d'en déduire les actions optimales à jouer selon l'état. Un deuxième type de méthodes est ce qu'on appelle les algorithmes de type *Policy-based* : plutôt que de passer par l'estimation de fonctions de valeur, la politique est directement apprise. Enfin, la troisième catégorie mélange les deux approches précédentes, elle est appelée *Actor-Critic*. Il s'agit donc d'une méthode hybride utilisant une fonction de valeur (appelée *Critic*) et une politique (*Actor*) qui sont entraînées simultanément par méthodes *Value-based* et *Policy-based* jusqu'à convergence.

Selon la nature des espaces d'observations (discret à faible cardinal, discret très vaste ou bien même continu) ainsi que de la nature des espaces d'actions (discret ou continu), certains modèles peuvent être privilégiés et des fonctions d'approximations complexes telles que les réseaux de neurones peuvent être nécessaires. Dans cette section, nous n'abordons pas ces aspects qui seront évoqués dans la section 2.2.4.

Dans toute la suite, on considère qu'un agent fait face à un processus de type MDP $(\mathcal{S}, \mathcal{A}, T, r)$ dont la fonction de *return* associée est notée R_t (voir 2.2.1).

Méthodes Value-based

Une catégorie populaire d'algorithmes d'apprentissage sont les méthodes dites *value-based* [SB+98]. Elles consistent à estimer par apprentissage des fonctions qui associent à chaque état s (resp. à chaque couple état/action (s, a)) une espérance maximale de gain que l'on obtient en étant à l'état s (resp. en effectuant l'action a à l'état s).

Dans le cas d'une fonction de $\mathcal{S}$ dans $\mathbb{R}$, on note communément V_π la fonction qui associe à tout état s, l'espérance du gain R_t (défini en section 2.2.1) que l'agent obtient en suivant la politique π à partir de l'état s : $V_\pi(s) = \mathbb{E}_\pi[R_t | s_t = s]$.

Un exemple d'algorithme pour entraîner et estimer cette fonction est l'algorithme du *Temporal Difference learning* (ou *TD-learning*). Son principe est le suivant : à chaque itération, l'agent observe l'état s, il évalue la politique $a \leftarrow \pi(s)$, puis joue a pour observer alors un nouvel état s' tout en recevant r. On effectue alors la mise à jour suivante :

$$V_\pi(s) \leftarrow V_\pi(s) + \alpha(r + \gamma V_\pi(s') - V_\pi(s)) \tag{2.3}$$

où $\alpha \in]0, 1[$ est un taux d'apprentissage et $\gamma \in [0, 1[$ est un coefficient qui atténue l'importance des récompenses futures. En quelques mots, en tenant compte de la récompense r que l'agent vient de recevoir et du gain espéré à l'état suivant $V_\pi(s')$, la nouvelle valeur du gain espéré à l'état s est de $r + \gamma V_\pi(s')$, et donc $(r + \gamma V_\pi(s') - V_\pi(s))$ correspond à l'erreur ce qui permet ensuite de procéder à la mise à jour au taux α.

En ce qui concerne le cas d'une fonction de $\mathcal{S} \times \mathcal{A}$ dans $\mathbb{R}$, on note Q_π la fonction qui associe à tout couple (s, a) l'espérance du gain R_t que l'on obtient en suivant la politique π à partir de l'état s en jouant l'action a est $Q_\pi(s, a) = \mathbb{E}_\pi[R_t | s_t = s, a_t = a]$.

Un algorithme de ce type d'apprentissage est le *Q-learning* [WD92], le principe de son entraînement est similaire à celui du *TD-learning*. A chaque itération, on observe un état s, on évalue la politique actuelle[1] pour obtenir une action a, en la jouant on observe un nouvel état s'

et on reçoit une récompense r. On procède alors à la mise à jour suivante :

$$Q_\pi(s,a) \leftarrow Q_\pi(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q_\pi(s,a)] \tag{2.4}$$

Dans cette formulation, le nouveau gain espéré pour l'état s est $r+\gamma \max_{a'} Q(s',a')$ avec $\max_{a'} Q(s',a')$ le gain espéré à l'état suivant.

Pour finir, si l'on suppose que ces fonctions de valeur convergent vers des fonctions optimales (Un MDP fini est suffisant pour leur existence) que l'on note V^* et Q^*, alors on peut en déduire une politique π^* à savoir :

— *V-value* : $\pi^*(s) = \underset{a}{\text{argmax}} \ \sum_{s'} T(s'|s,a)(r(s,a) + \gamma V^*(s'))$

— *Q-value* : $\pi^*(s) = \underset{a}{\text{argmax}} \ Q^*(s,a)$

En général, la méthode avec les fonctions de *Q-value* sont plus adaptées quand on ne connaît pas les modèles de transition T et de récompense r (*Model-free*). Ajoutons pour finir, qu'un troisième type de fonction de valeur est proposé dans les algorithmes de la littérature. Il s'agit de la fonction Avantage (*Advantage*) ; elle est notée A, se base sur les fonctions V et Q, et est définie comme suit :

$$\begin{aligned} A &: \mathcal{S} \times \mathcal{A} \to \mathbb{R} \\ A &: (s,a) \mapsto Q(s,a) - V(s) \end{aligned} \tag{2.5}$$

En quelques mots, cette fonction donne le gain (l'avantage) en espérance de choisir l'action a depuis l'état s par rapport au gain espéré étant à l'état s.

Méthodes *Policy-based*

L'objectif ici est qu'un agent apprenne directement la politique $\pi(a|s)$ [SMSM00]. On suppose que l'on modélise cette politique qu'on la note π_θ par des paramètres θ qui peuvent être des coefficients d'une fonction linéaire, ou polynomiale, voire les poids d'un réseau de neurones.
Pour déterminer les poids optimaux de la politique, on se donne pour objectif de chercher à maximiser une certaine fonction de gain $J(\theta)$ donnée par :

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)} R(\tau) = \sum_\tau \pi_\theta(\tau) R(\tau) \tag{2.6}$$

où τ est une trajectoire, c'est-à-dire une succession d'états et d'actions qui ont été respectivement observés et jouées : $\tau = (s_1, a_1, s_2, a_2, ..., s_T, a_T)$. $R(\tau)$ représente la récompense cumulée liée à cette trajectoire. Par ailleurs, $\pi_\theta(\tau)$ désigne, par abus, la probabilité de la trajectoire τ sachant

1. Évaluer la politique ici ne signifie pas simplement choisir $a \leftarrow \pi(s)$ mais plutôt de suivre la politique de compromis exploitation/exploration. Par exemple dans le cas du ϵ-*greedy*, il s'agit de choisir a au hasard avec une probabilité ϵ et $\pi(s)$ avec une probabilité $1-\epsilon$

les paramètres θ de la politique ayant généré τ. Les poids θ^* de la politique optimale π_{θ^*} sont ceux qui maximisent J :

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\ J(\theta) \tag{2.7}$$

Le principe des méthodes dites de *Policy-gradients* [SMSM00] est d'obtenir les paramètres optimaux par montée de gradient :

$$\begin{aligned} &\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \\ &\text{avec } \nabla_\theta J(\theta) \text{ le gradient de } J \end{aligned} \tag{2.8}$$

L'objectif revient alors à déterminer le gradient de $J(\theta)$. Une première idée est proposée par l'algorithme REINFORCE [SBW92]. En utilisant une astuce de dérivée ($\nabla_x f(x) = f(x)\nabla_x \log(f(x))$), on peut transformer le gradient de J comme une espérance sur les trajectoires générées par la politique entraînée jusque-là :

$$\begin{aligned} \nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\Big[R(\tau)\nabla_\theta \log \pi_\theta(\tau)\Big] \\ &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_{t=1}^{|\tau|-1} \nabla_\theta \log \pi(a_t|s_t)\right] \end{aligned} \tag{2.9}$$

Pour estimer alors cette espérance, on procède par échantillonnage de Monte-Carlo en jouant la politique jusqu'à là entraînée, afin de générer un échantillon de trajectoires $\{\tau_i\}_{i\in[\![1,M]\!]}$. On calcule alors une estimation du gradient :

$$\nabla_\theta J(\theta) \approx \frac{1}{M}\sum_{i=1}^{M} \nabla_\theta \log(\pi_\theta(\tau_i))R(\tau_i) \tag{2.10}$$

Pour finir, après l'obtention de $\nabla_\theta J(\theta)$, on peut procéder à la mise à jour des poids θ par montée de gradient selon l'équation 2.8.
Notons que ce type de méthode est générique et s'adapte à toute forme de modèle de politiques qui est différentiable en θ. Bien que ces méthodes peuvent adresser tout type d'états et d'actions, elles souffrent néanmoins d'une lente convergence et d'une forte variance. Des améliorations ont été proposées pour palier ces problèmes comme l'utilisation d'un recalage en soustrayant la moyenne des $R(\tau)$ [WT13].

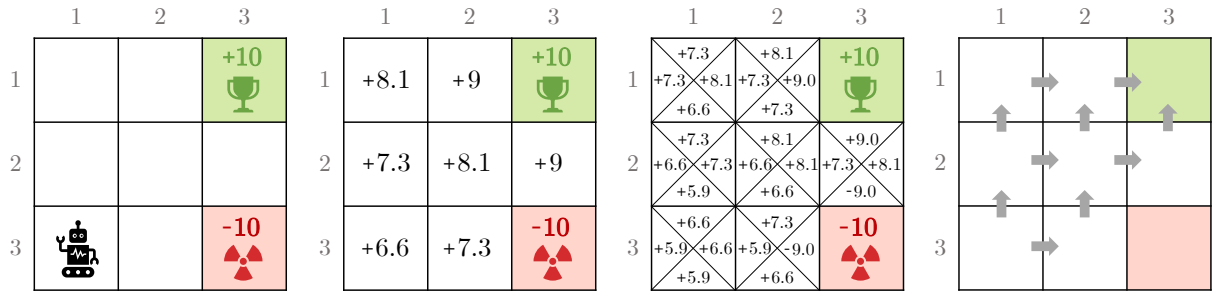
Actor-Critic

Un autre type d'algorithme regroupe les deux premiers, il s'agit des méthodes dites *Actor-Critic* [KT00]. L'algorithme est composé d'abord d'un modèle de type *policy-based* $\pi(a|s)$ qui est une politique entraînée à générer directement des actions compte tenu des observations, cette politique est appelée l'*Actor*. Un deuxième modèle vient compléter l'algorithme avec une fonction

de valeur (modèle de type *value-based*), le *Critic*, que l'on note $C(s, a)$ qui va être entraîné à juger les actions décidées par l'*Actor* selon les états. Les deux modèles sont entraînés simultanément jusqu'à convergence.

2.2.4 Des méthodes tabulaires au *Deep Reinforcement Learning* (DRL)

Les méthodes précédemment décrites brièvement ont été volontairement agnostiques sur le format des fonctions de valeur et sur les modèles de politiques. En effet, il est possible selon les situations d'adapter le choix du modèle. Lorsque l'ensemble des états est discret à faible cardinal, on peut alors se contenter de méthodes tabulaires. Ces méthodes sont basées sur le stockage des valeurs dans une table lors de la mise à jour, pour ensuite en extraire une politique. On peut également envisager des réseaux de neurones si l'espace des états est très vaste, voire continu. Pour illustrer ces différences, considérons un jeu très simple (Figure 2.7a) : un robot évolue sur un petit plateau de jeu de 3×3 cases. Il y a deux cases terminales : une case objectif avec une récompense de $+10$ et une case piège avec une pénalité de -10.



(a) Problème (b) $V^*(s)$ (c) $Q^*(s, a)$ (d) Politique

FIGURE 2.7 – Principe des méthodes *Value-based*. Soit le jeu à résoudre montré en Fig 2.7a, il y a 9 états, dont 2 terminaux. Pour chaque état non terminal, on peut alors déterminer la fonction optimale V^* (Fig 2.7b) ainsi que Q^* (Fig 2.7c) qui donne pour chaque état, le gain espéré pour les quatre actions possibles. Notons qu'on utilise un taux d'amortissement $\gamma = 0.9$. Enfin la politique optimale peut être extraite à l'aide notamment de Q^* (Fig 2.7d).

Les tables de fonction de valeurs de V et Q sont entraînées et actualisées jusqu'à convergence selon les mises à jour décrites dans la section 2.2.3. Enfin une politique en est extraite.

Sur ce même exemple, un algorithme de type *policy-based* consisterait à encoder l'état, à savoir la position du robot, par exemple sous forme vectorielle (dans l'exemple, la position du robot est $(1, 3)$). Ensuite, à l'aide de la politique, on obtient un

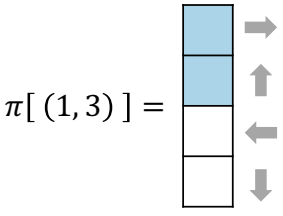


FIGURE 2.8 – Principe des méthodes *policy-based*

vecteur définissant les probabilités de choisir chacune des quatre actions possibles. Notons alors qu'à la différence des méthodes tabulaires de type *value-based*, il est possible de traiter des cas continus. Enfin, dans le cas d'ensembles d'états complexes ou très vastes, on peut utiliser des réseaux de neurones comme modèle de fonction de valeur et de politique. Par exemple, la Figure 2.9 représente une extension du *Q-learning* sous la forme d'un réseau de neurones qui prend, en entrée, un état s et une action a sous forme vectorielle et génère en sortie un scalaire qui estime le gain espéré à l'état s en jouant l'action a. Ce modèle s'appelle un *Deep Q-Network* (DQN). Le principe de l'entraînement est le même que le *Q-learning* à la différence que la mise à jour des valeurs est remplacée par une mise à jour des poids neuronaux par descente de gradient [MKS+15].

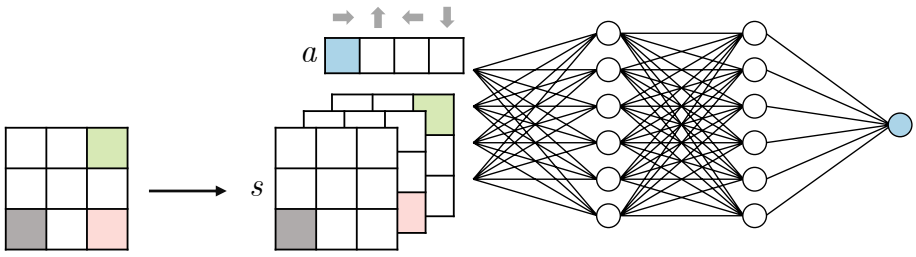


FIGURE 2.9 – Un *Deep Q-Network* (DQN) qui étend le *Q-learning* pour modéliser une fonction complexe : $DQN : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Dans le même esprit, pour le cas *Policy-based*, il est tout à fait possible de modéliser une politique par un réseau de neurones qui cette fois-ci prendra en entrée un état s (sous forme vectorielle) et génère en sortie les probabilités de choisir chaque action (sous forme vectorielle de même dimension que l'espace des actions).

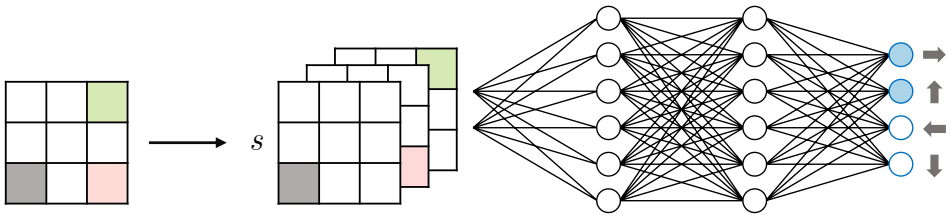


FIGURE 2.10 – Un réseau neuronal pour modéliser la politique $\pi : \mathcal{S} \to \Delta(\mathcal{A})$

L'intérêt des méthodes neuronales s'illustre dans cet exemple par sa capacité à entraîner des modèles qui peuvent couvrir de nombreuses combinaisons d'états, selon la position du robot (case noire), de l'objectif (case verte) et du piège (case rouge). En entraînant ces modèles sur divers environnements, l'agent sera capable de réagir même si les cases spécifiques de l'environnement sont différentes. Ce qui est impossible avec les méthodes tabulaires.

En ce qui concerne les algorithmes de *Deep* RL, l'état de l'art est relativement récent et est en développement constant. Dans ce qui suit, nous allons citer quelques modèles et méthodes d'entraînement couramment utilisés dans la littérature. Pour commencer, l'extension neuronale au RL remonte probablement à l'algorithme REINFORCE [SBW92]. Avec la nouvelle émergence du *Deep learning*, le *Deep Q-network* a ensuite proposé d'étendre le *Q-learning* avec un réseau neuronal [MKS+15]. Une extension du DQN a été proposée à l'aide de réseaux de neurones récurrents [HS15] notamment pour palier les situations partiellement observables (de type *Partially Observable Markov Decision Processes* (POMDP)). D'autres approches ont étudié également le cas partiellement observable [IZL+18, LVC18]. L'algorithme *Deep Deterministic Policy Gradient* (DDPG) a été proposé pour s'adapter à des ensembles d'actions continu [LHP+15]. Il a été suivi ensuite par d'autres approches de type *Actor-Critic* telles que les algorithmes A3C [MBM+16], TRPO [SLA+15] et PPO [SWD+17]. Cette dernière approche est pour le moment un des meilleurs algorithmes de l'état de l'art et permet d'adresser des ensembles d'états et d'actions continus ou discrets.

2.2.5 Bandits manchots : du RL sans observation

Le bandit manchot est l'ancien nom pour désigner une machine à sou. Ce modèle est le père de l'apprentissage par renforcement [Rob52, Sli19]. Il est formalisé comme suit. On suppose qu'un bandit possède plusieurs bras sur lesquels un agent peut tirer, auquel cas il se voit recevoir un gain qui dépend du bras tiré et d'une distribution stochastique fixée associée à ce bras. L'agent ne connaît pas les distributions mais il souhaite maximiser son gain cumulé. Il doit donc à la fois affiner sa connaissance quant aux bras les plus profitables (exploration) tout en continuant à choisir les bras jusque-là considérés comme rentables (exploitation). Le bandit manchot peut donc être vu comme un environnement de RL sans observation d'état dans lequel les actions sont modélisées par les bras sur lesquels on peut tirer. La fonction de récompense ne dépend que de l'action choisie et est régie par les distributions. Remarquons cependant que certains modèles de bandits peuvent permettre l'observation d'un état. On les appelle les bandits contextuels [LZ07]. Le formalisme des bandits présente l'avantage d'être propice à de solides résultats théoriques. Parmi les nombreux algorithmes qui ont été proposés pour jouer efficacement face aux bandits, on peut citer la méthode du *Thompson Sampling* (TS) [Tho33] initialement proposée pour palier le dilemme "exploitation/exploitation". Il y a également l'algorithme *Upper Confidence Bound* (UCB) [ACBF02] qui se base sur des intervalle de confiance qui diminuent au cours de l'apprentissage. Enfin, rappelons que les distributions de probabilité de gain sont fixées. Certains modèles de bandits considèrent néanmoins le cas non-stochastique, ils ont alors appelés les bandits adverses (*Adversarial bandits*) [ACBFS02] avec des algorithmes proposés pour adresser ce cas de figure comme le EXP3 [SSAAY13].

2.2.6 Apprentissage par Renforcement Multi-Agents (MARL)

L'apprentissage par renforcement classique résumé plus haut devient plus complexe dès lors que plusieurs agents interviennent dans l'environnement qui perd alors certaines propriétés de stationnarité du fait du comportement des autres joueurs. La propriété de Markov n'est plus respectée [LMFP+11] *i.e.* l'évolution de l'environnement ne dépend plus seulement de l'état et de l'action d'un agent. L'environnement multi-agents est alors appelé un jeu. Dans cette section, nous détaillons quelques concepts liés à l'apprentissage par renforcement multi-agents ou *Multi-Agent Reinforcement Learning* (MARL), notamment en ce qui concerne les aspects coopératifs.

Jeux de Markov à N joueurs

Pour modéliser ce type d'environnement multi-agents, il est commun d'utiliser le formalisme des jeux de Markov, qui sont définis comme des processus de décision markovien partiellement observable (*Partially Observable Markov Decision Processes* (POMDP)) à N joueurs [Sha53, Lit94]. Un jeu de Markov $\mathcal{M}$ à N joueurs est défini dans la littérature par un sextuplet $(\mathcal{I}, \mathcal{S}, \mathcal{A}, O, \mathcal{T}, R)$ où $\mathcal{I} = \{1, ..., N\}$ est un ensemble de joueurs, $\mathcal{S}$ est un ensemble d'états et $O : \mathcal{S} \times \mathcal{I} \to \mathcal{S}$ est une fonction d'observation. $\mathcal{A} = \mathcal{A}^1 \times ... \times \mathcal{A}^N$ est l'ensemble des actions jointes. Une action jointe notée $\vec{a} = (a^1, ..., a^N)$ transforme un état s du jeu selon la fonction de transition stochastique $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \to \Delta(\mathcal{S})$. Enfin, une fonction individuelle de récompense $R : \mathcal{I} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ donne une récompense r^i à chaque joueur i. La récompense jointe est notée $\vec{r} = (r^1, ..., r^N)$. L'objectif de chaque joueur ou agent i est de trouver une politique optimale $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ ($\vec{\pi}$ étant la politique jointe $(\pi^1, ..., \pi^N)$) de manière à maximiser l'espérance de la récompense totale atténuée au taux γ :

$$V^i_{\vec{\pi}}(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^i(s_t, \vec{a}_t) | \vec{a}_t \sim \vec{\pi}, s_{t+1} \sim \mathcal{T}(s_t, \vec{a}_t)]$$

Deep RL dans les jeux

En ce qui concerne le *Deep* RL dans les jeux, il a été connu d'abord pour la résolution de jeux à somme nulle, telles que des jeux de type Arcade [MKS+13] ou plus complexes comme le Go [SHM+16]. Pour les jeux à somme non nulle, le MARL a été également étudié sur ses aspects coopératifs. Dans un premier temps, quand la coopération n'est pas une stratégie dominée, c'est-à-dire quand il n'y a pas d'intérêt particulier à être égoïste [TMK+17, FCAS+17, LWT+17, PL17]. Puis dans un second temps, quand la coopération n'est pas la stratégie dominante et qu'il existe des incitations à être égoïste malgré une coopération mutuelle optimale (Dilemmes sociaux) [LZL+17, HLP+18, LP17]. Nous reviendrons plus en détails sur cette dernière catégorie de jeux dans la section 2.3. Enfin, d'autres approches ont abordé des situations où plusieurs agents

apprenaient à communiquer ou à négocier [FADFW16, LYD+17].

2.3 Dilemmes sociaux séquentiels

Dans la section précédente, nous avons évoqué les jeux à somme non-nulle. Nous nous intéressons dans cette section à un type de jeux à somme non-nulle qui sont les dilemmes sociaux séquentiels ou *Sequential Social Dilemma* (SSD). Il s'agit d'une extension des dilemmes sociaux définis dans le chapitre 1 (tels que le dilemme du prisonnier). Dans les dilemmes sociaux classiques, les actions possibles des joueurs sont simples, soit la coopération soit la défection, ou dans une moindre mesure un choix continu entre les deux. Les dilemmes sociaux séquentiels sont, eux, une extension qui a pour but d'étendre les actions atomiques des joueurs des dilemmes classiques en des politiques complexes. Avec l'essor du *Reinforcement Learning* et du *Deep Reinforcement Learning* [MKS+15, MKS+13], des études ont été conduites sur l'apprentissage de politiques complexes dans des situations de dilemme social séquentiel.

2.3.1 Exemples de jeux

Les travaux de [LZL+17] et [HLP+18] proposent d'adresser des jeux en situation de dilemme social séquentiel (SSD). Avant de définir plus formellement la modélisation des SSD, nous présentons et détaillons quelques exemples de jeux proposés dans les travaux cités.

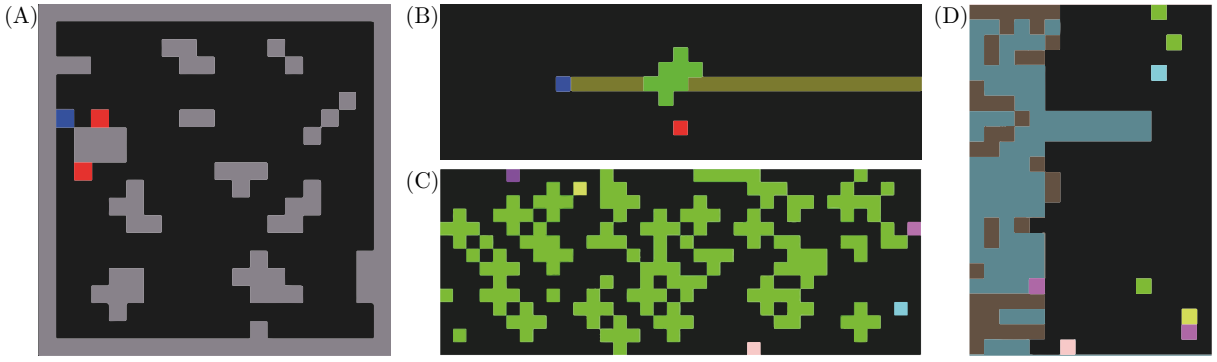


FIGURE 2.11 – Exemples de jeux de type SSD proposés dans [LZL+17] et [HLP+18]. (A) : WOLFPACK, (B) : GATHERING, (C) : HARVEST, (D) : CLEANUP

Les visuels de ces jeux sont représentés en Figure 2.11. La meute de loups (WOLFPACK) (Figure 2.11A) [LZL+17] implique deux loups (pixels rouges) qui chassent collectivement une proie (pixel bleu). Lors d'une capture, ils reçoivent une récompense proportionnelle au nombre de loups présents dans un certain rayon. Cette récompense est justifiée notamment par le fait que la supériorité numérique permet à la fois d'atteindre une proie plus massive et de la protéger

davantage des charognards une fois attrapée. Les deux loups obtiennent donc une récompense de r_{lone} et r_{team} en fonction du nombre de loups présents dans un rayon $radius$. En jouant sur le ratio $\frac{r_{team}}{r_{lone}}$ et le paramètre $radius$, on peut alors plus ou moins inciter les agents à la coopération.

Le jeu GATHERING [LZL+17] implique deux joueurs (pixels rouge et bleu) qui ont pour objectif de collecter des pommes (pixels verts) qui leur rapportent une récompense de +1 chacune (Figure 2.11B). Après sa collecte, une pomme ne réapparaît que N_{apples} itérations plus tard. Chaque joueur peut tirer sur l'autre avec un rayon qui a pour effet de le neutraliser pendant N_{tagged} itérations. Un tir ne rapporte aucune récompense mais il peut être tentant d'être seul pour récolter plus de pommes. Cependant, si un comportement agressif mutuel émerge, les joueurs seront tous deux perdants. Dans les études de [LZL+17], l'agressivité a été plus ou moins incitée en réglant les valeurs de N_{apples} et N_{tagged}. Selon, ces valeurs, la situation se retrouve en situation de dilemme social.

Les jeux de moisson HARVEST (Figure 2.11C) et de nettoyage CLEANUP (Figure 2.11D) impliquent cette fois-ci plus de deux joueurs [HLP+18]. Dans le jeu HARVEST, les joueurs (pixels de couleur autre que le vert) collectent des pommes (pixels verts). L'idée est d'observer comment les joueurs se répartissent dans l'environnement pour ne pas convoiter les mêmes zones. Dans le jeu CLEANUP, les joueurs doivent également collecter des pommes mais la fréquence d'apparition de pommes est conditionnée à la propreté de la rivière (pixels bleus) attenante aux pommiers. Les joueurs doivent donc se dévouer pour aller éliminer les déchets (pixels marrons) au risque donc de perdre du temps et l'occasion de collecter les pommes.

Ces jeux sont formalisés comme des dilemmes sociaux séquentiels (SSD) dont on apporte les définitions dans la section suivante.

2.3.2 Dilemme social séquentiel

Le concept de dilemme social séquentiel a été étudié dans plusieurs travaux antérieurs [CAK96, BEKN09] qui proposent une variante du dilemme social répété dans laquelle les joueurs reçoivent une récompense conditionnée par une succession de choix et non plus une récompense à chaque itération. Un concept similaire à l'aide de politiques de RL est proposé par [LZL+17]. Bien que le terme *Sequential Social Dilemma* est également utilisé, l'approche est assez différente. Il s'agit d'une extension du dilemme social classique vu dans le chapitre 1. Cette extension est à la fois temporelle au sens où des actions effectuées précocement peuvent avoir un impact plus tardivement avec des gains qui confèrent au jeu une situation de dilemme. Cette extension est également plus complexe au sens où les actions ne sont plus limitées à la coopération/défection (ou au mieux à un degré de coopération) mais sont étendues à des politiques plus complexes (par

exemple de RL). Nous définissons dans cette section, le modèle proposé dans [LZL+17] ainsi que l'extension à N joueurs définie par [LP17].

Rappels sur les dilemmes sociaux simples

Rappelons qu'un dilemme social discret à deux joueurs est défini par un jeu dans lequel deux joueurs peuvent choisir entre la coopération et la défection et que leurs gains respectifs explicités dans la table 2.1 vérifient les inégalités données par 2.11.

	Coopération	Défection
Coopération	(R, R)	(S, T)
Défection	(T, S)	(P, P)

TABLE 2.1 – Gains dans un dilemme social à deux joueurs

$$\begin{aligned} &R > P \\ &R > S \\ &T > R \text{ (avidité) et/ou } P > S \text{ (crainte)} \\ &2R > T + S \end{aligned} \tag{2.11}$$

Dilemme social séquentiel (SSD) classique

L'extension proposée dans [LZL+17] commence par définir empiriquement des ensembles de politiques de coopération et de défection, notées Π^C et Π^D. En utilisant des politiques $\pi^C \in \Pi^C$ et $\pi^D \in \Pi^D$ dans un jeu de Markov (voir section 2.2.6), les espérances de gains empiriques suivantes sont introduites pour chaque état s du jeu où $V^i_{\pi^X,\pi^Y}(s)$ est le gain espéré (à l'état s) par l'agent i si les agents 1 et 2 jouent les politiques X et Y.

$$\begin{aligned} R(s) &:= V^1_{\pi^C,\pi^C}(s) = V^2_{\pi^C,\pi^C}(s) \\ P(s) &:= V^1_{\pi^D,\pi^D}(s) = V^2_{\pi^D,\pi^D}(s) \\ S(s) &:= V^1_{\pi^C,\pi^D}(s) = V^2_{\pi^D,\pi^C}(s) \\ T(s) &:= V^1_{\pi^D,\pi^C}(s) = V^2_{\pi^C,\pi^D}(s) \end{aligned} \tag{2.12}$$

Dans les travaux de [LZL+17], les politiques dites de coopération et de défection, sont entraînées pour chaque jeu évoqué plus haut grâce à une manipulation des paramètres du jeu, à savoir les taux de durées de disparition N_{apples} et N_{tagged} dans le jeu GATHERING et le ratio r_{team}/r_{lone} et la taille du rayon $radius$ dans le jeu WOLFPACK.

Compte tenu de ces notations et concepts, un dilemme social séquentiel (SSD) est défini comme suit (Définition 2.2).

Dilemme Social Séquentiel (SSD)

Définition 2.2 *Soit un jeu de Markov $\mathcal{M}$ avec un ensemble d'état $\mathcal{S}$ et des ensembles de politiques Π^C et Π^D modélisant des politiques dites coopératives et défectives. Alors un Dilemme Social Séquentiel (SSD) est défini par le tuple $(\mathcal{M}, \Pi^C, \Pi^D)$ tel qu'il existe des états $s \in \mathcal{S}$ de sorte que les gains empiriques $S(s), P(s), R(s), T(s)$ définis en 2.12 vérifient les inégalités classiques de dilemme social (2.11).*

Extension à N joueurs

La formalisation précédente des SSD [LZL+17] n'explicite pas le cas à plus de deux joueurs. Une proposition d'extension à plus de deux joueurs est donnée dans [HLP+18]. Elle s'inspire de l'idée des diagrammes de Schelling [Sch73]. Ce type de diagramme est une représentation très utile des gains pour des jeux à plus de deux joueurs auquel cas les tables de gains ne sont plus appropriées. Ici, on suppose que les gains sont symétriques et homogènes. En d'autres termes, le gain reçu par un joueur ne dépend que :

1. de son choix de stratégie (coopérer ou non)
2. du nombre l d'autres coopérateurs dans le jeu $(0 \leq l \leq N-1)$

Un diagramme de Schelling représente alors une courbe pour chaque choix de coopération en fonction du nombre d'autres coopérateurs. Pour mieux comprendre ces diagrammes, les trois dilemmes canoniques de la littérature sont représentés en Figure 2.12.

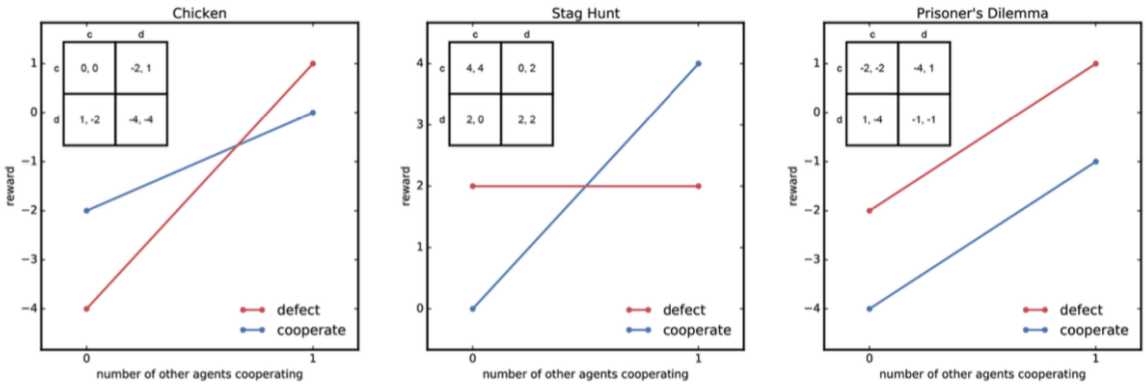


FIGURE 2.12 – Diagrammes de Schelling pour les trois dilemmes sociaux classiques issus de [LZL+17], davantage d'explications sont données plus bas.

L'extension proposée dans [LZL+17] à N joueurs proposée est définie comme suit. On suppose que Π^C et Π^D définissent des ensembles de politiques de coopération et de défection. Commençons alors par considérer le profil de stratégies $\pi = (\pi_1^C, ..., \pi_l^C, \pi_1^D, ..., \pi_m^D) \in (\Pi^C)^l \times (\Pi^D)^m$ avec $l + m = N$. On dénote alors respectivement $R_C(l)$ (resp. $R_D(l)$) les gains moyens obtenus par chaque joueur qui choisit la politique de coopération (resp. de défection) quand il fait face à l autres coopérateurs. Avec ce formalisme, les SSD à N joueurs sont alors définis comme suit (Définition 2.3) :

Dilemme Social Séquentiel (SSD) à N joueurs

Définition 2.3 *On dit que $(\mathcal{M}, \Pi^C, \Pi^D)$ est un dilemme social séquentiel à N joueurs si et seulement si les fonctions de gains $R_C(l)$ et $R_D(l)$ définies plus haut vérifient les conditions suivantes :*

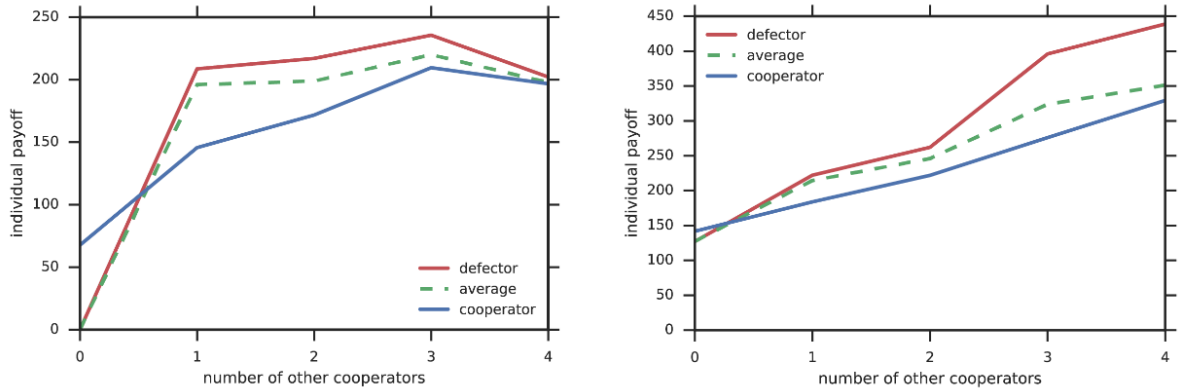
1. *$R_C(N) > R_D(0)$: la coopération mutuelle préférée à la défection mutuelle*
2. *$R_C(N) > R_C(0)$: la coopération mutuelle préférée à l'exploitation par des défecteurs*
3. *Au moins l'une des deux conditions :*
   (a) *Pour des valeurs de i assez petites, $R_D(i) > R_C(i)$: Crainte, la défection mutuelle préférée à l'exploitation par des défecteurs*
   (b) *Pour des valeurs de i assez grandes, $R_D(i) > R_C(i)$: Avidité, l'exploitation de coopérateurs préférée à la coopération mutuelle*

Pour une bonne intuition, les diagrammes de Schelling représentent les deux courbes $R_C(l)$ et $R_D(l)$. Autrement dit, les gains moyens obtenus si un joueur coopère ou défecte en présence de l autres coopérateurs. Les conditions de la définition 2.3 peuvent être graphiquement visibles. Étant donné que les courbes rouge et bleue (Figures 2.12 et 2.13) représentent respectivement les choix de défection et de coopération, nous avons alors que les conditions peuvent être montrées par :

1. Le point le plus à droite de la courbe bleue est au-dessus du point le plus à gauche de la rouge
2. La courbe bleue est croissante (*a minima* le point de droite au-dessus du point de gauche)
3. Au moins une partie de la courbe rouge est au-dessus de la courbe bleue :
   (a) Crainte : partie de gauche
   (b) Avidité : partie de droite

Nous pouvons vérifier que ces conditions sont vérifiées dans les diagrammes de Schelling des trois dilemmes sociaux canoniques à deux joueurs de la Figure 2.12. On visualise bien l'avidité

dans le jeu du *Chicken*, la crainte dans le jeu *Stag Hunt* et enfin les deux dans le Dilemme du Prisonnier. On peut retrouver les mêmes conditions graphiques sur les diagrammes de Schelling des jeux CLEANUP et HARVEST (Figures 2.13a et 2.13b).



(a) jeu CLEANUP (b) jeu HARVEST

FIGURE 2.13 – Diagrammes de Schelling pour les jeux de dilemmes sociaux séquentiels à plus de deux joueurs proposés par [HLP+18]

2.3.3 Approches proposées pour jouer dans les SSD

Les dilemmes sociaux séquentiels ont fait l'objet d'études dans lesquelles quelques approches ont été proposées pour l'entraînement des politiques dans de telles situations. Dans la suite, nous en résumons les principales.

Aversion à l'inéquité

Les travaux introduisant l'extension des SSD à N joueurs s'accompagne d'une proposition d'algorithme pour adresser ce type de jeu [HLP+18]. Elle se base sur une modification de la récompense d'entraînement des politiques de RL en tenant compte d'une aversion à l'inéquité. On suppose que les N joueurs reçoivent les récompenses $r_1, ..., r_N$. Chaque agent (de RL) modifie sa récompense en la transformant par la formule suivante :

$$U_i(r_1, ..., r_N) = r_i + \frac{\alpha_i}{N-1} \sum_{j \neq i} \max(r_j - r_i, 0) - \frac{\beta_i}{N-1} \sum_{j \neq i} \max(r_i - r_j, 0) \tag{2.13}$$

avec α_i (resp. β_i) le coefficient d'aversion à l'inéquité désavantageuse (resp. avantageuse) pour le joueur i.

Les résultats d'évaluation montrent que le bien commun (à savoir la somme des récompenses)

est supérieur à celui obtenu avec des agents simples suivant une politique de type A3C [MBM+16].

Motivation intrinsèque

Une approche similaire à celle de la modification de la récompense a été proposée dans [JLH+18]. La récompense est également modifiée mais en tenant compte d'une estimation de l'influence qu'une action a sur les actions des autres. La récompense r_t^k d'entraînement de politique RL pour le joueur k est calculée par la formule suivante :

$$r_t^k = \alpha e_t^k + \beta c_t^k \tag{2.14}$$

avec e_t^k qui est la récompense extrinsèque (celle de l'environnement) et c_t^k une récompense intrinsèque qui mesure l'influence causale qu'une action de l'agent k a sur les actions des autres joueurs. Pour calculer c_t^k, sans rentrer dans les détails, on suppose que l'agent k est capable de calculer les probabilités que chaque autre agent j choisisse l'action a_t^j en observant l'état s_t^j ainsi que la probabilité de choisir cette même action sachant que k joue a_t^k. La récompense intrinsèque est alors calculée avec la divergence de Kullback-Leibler D_{KL} [KL51] pour chaque autre joueur j comme suit :

$$c_t^k = \sum_{j \neq k} D_{KL}\Big[p(a_t^j | a_t^k, s_t^j) \;\Big|\Big|\; p(a_t^j | s_t^j)\Big] \tag{2.15}$$

D'une manière similaire à la modification de la récompense par aversion à l'inéquité, cette méthode permet de conduire à des gains collectifs meilleurs que ceux obtenus par entraînement de politiques à la récompense simple.

Approche hybride Tit-for-Tat/RL

Pour palier la possibilité d'exploitation par des défecteurs, le modèle proposé dans [LP17] s'inspire des performances du Tit-for-Tat (TFT). Il propose une approche hybride mêlant des politiques pré-entraînées en *selfplay* et des stratégies de TFT pour décider de quelle politique suivre. Cette méthode est proposée pour le cas à deux joueurs. Dans un premier temps, deux politiques sont entraînées en *selfplay* : ainsi l'agent 1 entraîne une politique de coopération $\hat{\pi}_C$ en utilisant une récompense collective $r_t^1 + r_t^2$ avec r_t^1 et r_t^2 les récompenses des joueurs 1 et 2. Ensuite il entraîne une autre politique de défection $\hat{\pi}_D$ égoïste qui utilise la récompense personnelle simple r_t^1. On suppose que le modèle des politiques entraînées contiennent une fonction de valeur $\hat{Q}$ associée. L'algorithme fonctionne comme suit. Nous sommes à l'état s, et le joueur 2 vient de jouer l'action a^2, on suit alors les étapes suivantes :

1. On cumule une différence de gain afin d'estimer s'il y a eu coopération ou non :
   $W \leftarrow W + (Q_{CC}^2(s, a^2) - Q_{CC}^2(s, \hat{\pi}_C^2(s)))$. En d'autres termes, on estime à quel point le

fait de jouer l'action a^2 a fait gagner plus au joueur 2 par rapport à s'il avait joué l'action dictée par sa politique de coopération $\hat{\pi}_C^2(s)$.

2. Lorsque cette différence cumulée atteint un certain plafond, on considère alors qu'il y a une défection, et en vertu du Tit-for-Tat, on choisit de jouer la politique de défection (pendant K itérations pour stabiliser le comportement).

Cette méthode hybride est très efficace pour inciter la coopération et prévenir l'exploitation par des défecteurs. Elle présente cependant le problème de la difficulté de détecter efficacement la volonté de coopérer chez l'adversaire.

2.4 Conclusion et perspectives

Dans ce chapitre, nous avons abordé un concept important des dilemmes sociaux qui est relativement récent. Il propose d'étendre les actions simples des dilemmes classiques en des politiques de type *Deep Reinforcement Learning* (DRL) ce qui permet de modéliser des situations plus réalistes. Des définitions formelles des SSD ont été apportées et étendues à des versions à plus de deux joueurs. Plusieurs jeux intéressants ont été implémentés dans la littérature pour modéliser des situations de dilemmes et permettre d'évaluer des solutions. En ce qui concerne les algorithmes proposés pour adresser ces jeux, des études ont été conduites notamment pour observer l'égoïsme des agents indépendants. En outre, des algorithmes ont été introduits en modifiant notamment les fonctions de récompense pour rendre les comportements plus pro-sociaux. Cependant, si les politiques deviennent alors moins égoïstes, elles restent vulnérables à la défection. Pour adresser ce problème, des approches hybrides à base de politiques de RL et de stratégies de TFT ont été proposées et sont très prometteuses.

Cependant, les modèles de jeu de dilemme social séquentiel pourraient être sensiblement améliorés en apportant dans un premier temps une distinction des joueurs. En effet, les modèles définis dans ce chapitre abordent le problème sous l'angle "un contre un" ou bien "un contre les autres" (compte tenu uniquement du nombre d'autres coopérateurs). Ils ne permettent donc pas d'identifier précisément des égoïstes et de continuer ainsi à coopérer sans eux (dans le cas à plus de deux joueurs). Dans un second temps, la coopération vue dans ces modèles est vue comme équitablement réciproque d'un joueur à l'autre. Nous verrons, dans les chapitres 4 et 5, que ce n'est pas toujours le cas et nous proposerons des extensions à ces modèles. Concernant les potentielles limites des approches proposées pour adresser ces jeux, on peut commencer par rappeler que certaines d'entre elles sont très vulnérables à l'exploitation par des défecteurs. Par ailleurs, pour considérer l'impossibilité de distinguer précisément des joueurs au comportement néfaste, il s'avère que certaines approches, même robustes à la défection, ne permettent pas d'isoler des joueurs égoïstes en continuant à jouer avec les autres. Nous proposerons alors des

extensions à ces méthodes dans les chapitres 4 et 5.

Pour conclure, l'extension des dilemmes classiques en des dilemmes plus complexes et réalistes est très prometteuse et les approches pour les adresser offrent de nombreuses perspectives. Par ailleurs, compte tenu des enjeux grandissants tels que la diminution de ressources et la raréfaction de l'énergie ainsi que l'explosion des agents intelligents qui nous entourent (objets connectés, etc), il devient fondamental de s'intéresser au paradigme d'agents apprenant à coopérer dans de telles situations.

Partie II

Vers l'apprentissage de la coopération

Chapitre 3

Apprentissage de stratégies dans un dilemme du prisonnier itéré simple

Dans ce chapitre, nous nous intéressons à l'étude de politiques interagissant dans un dilemme du prisonnier itéré à deux joueurs. Nous étudions le cas discret dans un premier temps puis le cas continu. Nous conduisons quelques simulations de tournoi dans l'objectif d'opposer plusieurs types de politiques qui sont apprises telles que des politiques d'apprentissage par renforcement ou *Reinforcement Learning* (RL) ou des bandits manchots. Dans un second temps, nous présentons une des contributions les plus importantes pour le reste de la thèse qui est une amélioration d'une stratégie de Tit-for-Tat (TFT) continu.

3.1 Introduction

Un des principaux objectifs de la thèse est d'étudier le comportement de politiques indépendantes de RL dans des environnements multi-opérateurs qui s'avèrent être des situations de dilemme. L'étude préliminaire d'un dilemme social plus simple est donc fondamentale. Nous nous penchons donc sur le dilemme du prisonnier dans lequel nous allons étudier rapidement quelques politiques de RL. De nombreuses études ont déjà été conduites à ce sujet dans la littérature [HK04, AF11, HKJ+17, LGMLR22]. L'idée dans ce chapitre est de procéder à quelques simulations pour rappeler que dès qu'il y a au moins une incitation à ne pas coopérer, les politiques de RL s'écartent du choix mutuel de la coopération, bien qu'il représente l'issue optimale. Devant ce constat, nous faisons le choix de nous intéresser ensuite à des politiques de type TFT. Ainsi, dans un second temps, nous présenterons nos principales contributions sur des fonctions de TFT continu. Pour évaluer ces nouveaux algorithmes, nous introduisons des métriques sociales qui seront d'ailleurs étendues à plus de deux joueurs dans les prochains chapitres. Notons que la fonction de TFT que nous introduisons dans ce chapitre sera un algorithme qui servira de

brique de base pour divers algorithmes dans toute la suite du manuscrit. En particulier, dans l'introduction d'un agent permettant de traiter les dilemmes à N joueurs dans le chapitre 4 ou dans un algorithme de partage de ressources dans le chapitre 7.

3.2 Aspects méthodologiques

Dans cette section, nous rappelons le cadre de notre modèle de dilemme du prisonnier itéré. Par ailleurs, nous introduisons également des métriques sociales qui seront utiles dans la section 3.4 pour évaluer nos propositions en ce qui concerne les stratégies de TFT continus.

3.2.1 Tournoi de dilemme du prisonnier itéré à deux joueurs

Soit un dilemme du prisonnier itéré à deux joueurs A et B. Dans ce chapitre, il sera décliné dans les versions discrète et continue. Dans la version discrète, les actions possibles sont C (coopération) et D (défection). Dans la version continue, il s'agit d'un degré de coopération $c \in [0, 1]$. On suppose que le jeu dure T étapes et qu'à chaque étape $t \in [\![0, T-1]\!]$, les joueurs A et B jouent simultanément une action. Ils reçoivent alors une récompense $V_A(t)$ et $V_B(t)$ selon les choix de chacun (voir Table 3.1 pour le cas discret ou l'interpolation présentée dans la définition 1.4 pour le cas continu). On suppose également qu'à chaque étape, les joueurs ont connaissance de l'ensemble des choix précédents. Enfin, le nombre d'étapes T du jeu leur est inconnu, sinon au dernier tour, chaque joueur choisirait la défection, et donc également au tour précédent, et ainsi de suite. Dans ce cas, l'équilibre de Nash du tournoi consisterait en une défection totale.

	Coopération	Défection
Coopération	(R, R)	(S, T)
Défection	(T, S)	(P, P)

TABLE 3.1 – Rappels des gains du dilemme du prisonnier, les valeurs couramment utilisées dans la littérature sont $S = 0$, $P = 1$, $R = 3$, $T = 5$

3.2.2 Métriques sociales

Pour évaluer les performances des politiques des joueurs, nous utilisons les métriques suivantes. Pour commencer, apportons quelques notations : $V_A^{\pi_A,\pi_B}(t)$ et $V_B^{\pi_A,\pi_B}(t)$ désignent les gains obtenus respectivement par A et B à l'étape t si ces derniers jouent les politiques π_A et π_B. Par ailleurs, on note $\overline{V_X^{\pi_A,\pi_B}} = \sum_{t=0}^{T-1} V_X^{\pi_A,\pi_B}(t)$ la somme des gains du joueur X au cours du tournoi. Enfin, on note respectivement, par abus, la politique de coopération C (consistant à jouer C à chaque étape) et la politique de défection D (consistant à jouer D à chaque étape).
En se basant sur ces notations, nous définissons les métriques sociales suivantes permettant

d'évaluer nos politiques. Une intuition graphique de ces métriques est accessible en annexe (Figure A.4).

Le bien commun

Cette métrique est aussi appelé *Social Welfare*, et notée SW. Elle représente la somme des gains des joueurs à chaque instant t :

$$SW(t) = V_A(t) + V_B(t) \tag{3.1}$$

Remarquons que compte tenu de l'expression des gains (eq 1.4), on a que $\forall t,\ 2P \leq SW(t) \leq 2R$.

L'efficacité

L'efficacité $E(t)$ est le bien commun normalisé.

$$E(t) = \frac{SW(t) - 2P}{2R - 2P} \tag{3.2}$$

Notons que puisque $2P \leq SW(t) \leq 2R$, nous avons systématiquement à chaque étape t : $E(t) \in [0, 1]$. Enfin, l'efficacité finale peut être définie soit par la moyenne $E_m = \frac{1}{T}\sum_{t=0}^{T-1} E(t)$ ou bien par la valeur de la dernière étape $E_{max} = E(T-1)$.

La rapidité

La rapidité Sp représente la vitesse à laquelle l'efficacité rejoint sa valeur finale.

$$\begin{aligned} &Sp(\mathcal{T}, \pi) = \frac{1}{\tau E_{max}} \int_0^{\tau} E(\mathcal{T}, \pi, t)dt \\ &\text{avec } \tau \in [1, T] \end{aligned} \tag{3.3}$$

L'*Incentive-Compatibility*

L'*Incentive-Compatibility* IC ou bien l'incitation à coopérer d'une certaine politique π mesure la différence de gains d'un partenaire entre deux alternatives : celle où il choisit la coopération et celle où il choisit la défection lorsqu'il fait face à π :

$$\begin{aligned} &IC(\pi) = \frac{1}{I_{norm}} \left[\overline{V_B^{\pi,C}} - \overline{V_B^{\pi,D}} \right] \\ &\text{avec } I_{norm} = \overline{V_B^{C,C}} - \overline{V_B^{D,D}} = T(R - P) \end{aligned} \tag{3.4}$$

La *Safety*

La *Safety* ou la sûreté mesure le risque que prend un joueur en jouant une certaine politique π face à un défecteur. On définit la *Safety* d'une politique π, notée $Sf(\pi)$, par :

$$\begin{aligned} Sf(\pi) &= \frac{1}{I_{norm}}\left[\overline{V_A^{\pi,D}} - \overline{V_A^{D,D}}\right] \\ \text{avec } I_{norm} &= \overline{V_A^{D,D}} - \overline{V_A^{C,D}} = T(P-S) \end{aligned} \tag{3.5}$$

Notons que puisque la défection est une stratégie dominante (par définition du dilemme), cette métrique est alors toujours négative ou nulle. Par conséquent, plus la métrique est élevée (donc proche de 0), plus la politique est sûre. En particulier $Sf \in [-1, 0]$ avec une valeur à 0 pour la politique la plus sûre (correspondant à la défection pure).

Indulgence

Supposons que l'on appelle un "défecteur repentant" L un agent qui défecte pour $t \in [0, \tau_0]$ puis se met à coopérer. Alors l'indulgence (ou *forgiveness*) Fg mesure à quelle vitesse le gain de l'agent L va augmenter lorsqu'il coopère à nouveau :

$$\begin{aligned} Fg(\pi) &= \frac{1}{I_{norm}}\left[\overline{V_A^{L,\pi}} - \overline{V_A^{C,D}}\right] \\ \text{avec } I_{norm} &= \overline{V_A^{C,C}} - \overline{V_A^{C,D}} = (T-\tau)(R-S) \end{aligned} \tag{3.6}$$

En quelques mots, on suppose que le joueur B est joué par la politique π (que l'on étudie), et qu'il fait face au joueur A pour lequel on compare les deux cas selon qu'il soit joué par l'agent L ou par la politique de coopération.

Résilience

La résilience Rl peut être vue comme une autre forme d'indulgence. Elle permet d'évaluer la capacité des joueurs à "sortir" d'une situation de défection mutuelle durable. Pour ce faire, l'idée est de forcer un choix de défection jusqu'à stabilité, puis de relaxer cette contrainte. La résilience est alors calculée comme l'efficacité après cet instant.

Pour formaliser cette métrique, soit deux politiques π^1 et π^2.

$$\begin{aligned} &\forall t < \tau,\ \pi_1(t) \leftarrow 0.0 \text{ et } \pi_2(t) \leftarrow 0.0 \\ &Rl(\pi^1, \pi^2) = \lim_{t\to\infty} E(t) = E_{max} = E(T-1) \end{aligned} \tag{3.7}$$

Brièvement, une résilience nulle survient après défecté longuement, les agents ne peuvent plus revenir à une coopération mutuelle. Dans le cas d'une résilience égale à 1, ce serait si après

une défection mutuelle, les agents seraient parvenus très vite à converger vers la coopération mutuelle à nouveau. S'il est plus pertinent de dénoter la résilience d'un couple de politiques $Rl(\pi^1, \pi^2)$, dans la suite nous ne considérerons la résilience que d'une politique π qui correspond à la résilience d'un couple de politiques identiques $Rl(\pi, \pi)$ que nous noterons alors par abus $Rl(\pi)$.

3.3 Apprentissage par renforcement et dilemmes du prisonnier itérés

Dans cette section, nous procédons à quelques rapides simulations de politiques d'apprentissage par renforcement (RL) appliquées dans un Dilemme du Prisonnier Itéré (IPD). L'objectif n'est pas d'être exhaustif dans les techniques et algorithmes de RL, mais plutôt de se faire une intuition du problème de convergence des politiques de RL dans un IPD.

3.3.1 Modèle

Pour l'étude des politiques de RL, nous adoptons le modèle du IPD en version discrète. Il y a donc deux actions possibles qui sont la coopération ou la défection. En ce qui concerne les observations, on considère qu'il s'agit des dernières actions personnelles et celles de l'adversaire. Les récompenses associées à chaque action sont les gains du dilemme du prisonnier classique : (S, R, P, T) rappelées en section 3.4.1. Notons que pour l'étude des bandits, il conviendra de normaliser les gains obtenus par la différence maximale [RCO65] comme suit :

$$r \leftarrow \frac{r - S}{T - S}$$

3.3.2 Scénarios étudiés

Pour chaque type de politique de RL notée π, nous étudions trois principaux scénarios d'entraînement :

- Avec des modèles différents : π_1 vs π_2
- En *self-play*, avec un modèle identique : π vs π
- Face à un Tit-for-Tat discret : π_1 vs TFT

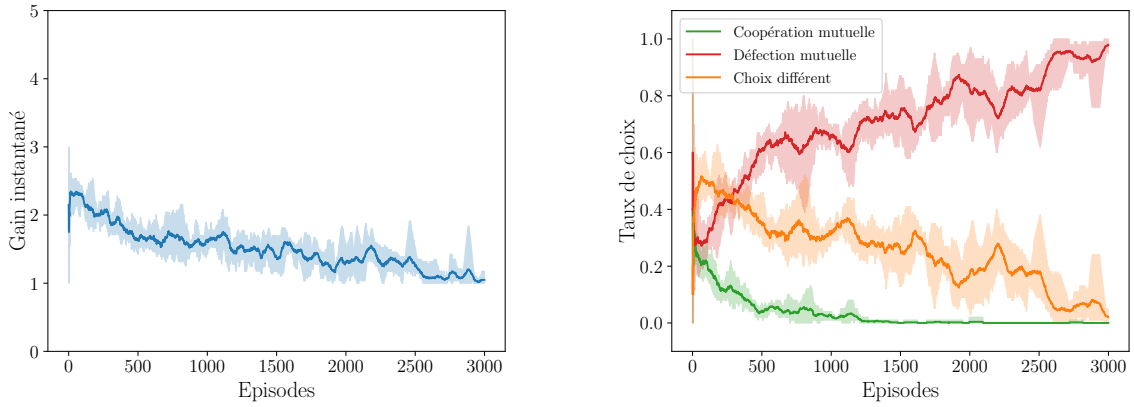
Les politiques de RL que nous étudions sont de plusieurs types :

- Bandits manchots ou *Multi-Armed Bandit* (MAB), *e.g.* algorithme EXP3 [ACBFS02]
- Méthodes tabulaires, avec l'algorithme *Q-learning* [WD92]

3.3.3 Étude de l'algorithme *Q-learning*

Dans notre étude, nous choisissons un des algorithmes les plus basiques qui est le *Q-learning* en méthode tabulaire, c'est-à-dire en observation discrète. Les observations considérées sont les K dernières actions effectuées par les deux agents. Comme il y a seulement deux actions discrètes possibles, cela fait $2^{2\times K}$ possibles états d'observation. Nos simulations ayant conclu que la longueur de l'historique K ne variait pas les conclusions, nous avons donc simplement choisi de fixer K à 1 (c'est à dire uniquement les actions du coup précédent).

Commençons alors à entraîner deux agents indépendants de *Q-learning* (*i.e.* π_1 et π_2) dans 5 différents tournois de IPD de $T = 3000$ épisodes (5 exécutions ou *runs*). On représente les courbes d'entraînement dans la Figure 3.1.

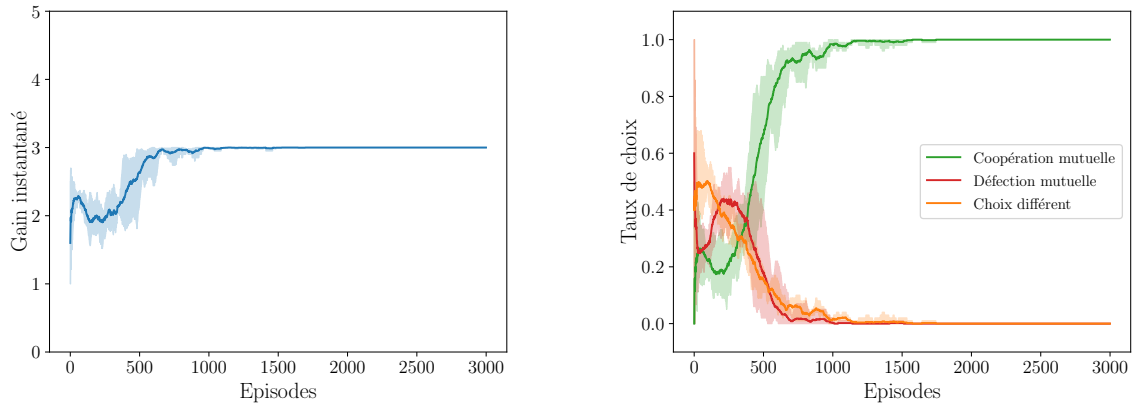


(a) (b)

FIGURE 3.1 – Apprentissage d'un *Q-learning* avec modèles différents. *Return* calculé au taux d'atténuation $\gamma = 0.9$ et exploration avec un ϵ-greedy où le ϵ d'exploration initialisé à 1.0 est multiplié à chaque étape par $\epsilon_{decay} = 0.998$. Le *learning rate* est choisi égal à $\alpha = 0.01$.

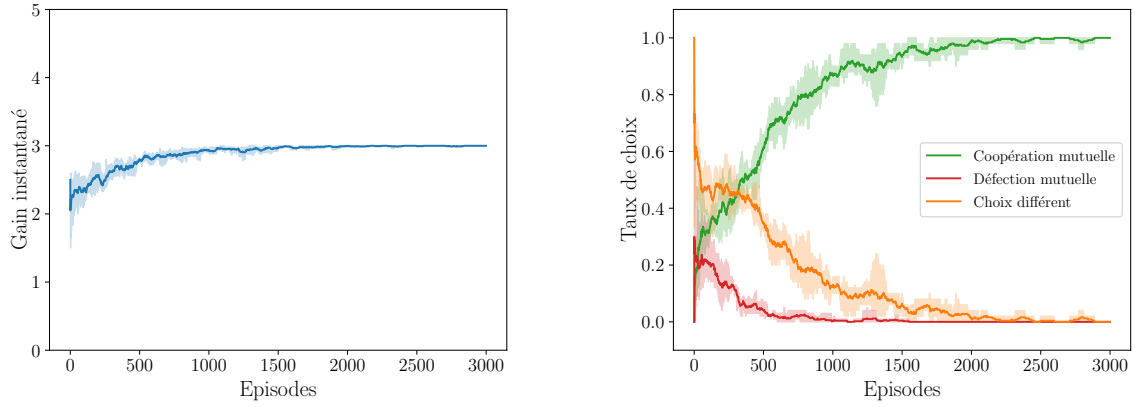
On peut observer sur la Figure 3.1b que deux agents de *Q-learning* indépendants peinent à converger vers une coopération mutuelle. Au mieux, au début de l'entraînement, ils hésitent entre la défection et l'alternance coopération/défection. La coopération est vite écartée des choix intéressants. Cependant, on peut remarquer que si les agents utilisent un modèle identique (π vs π) (Figure 3.2), alors les agents tendent plutôt vers la coopération mutuelle. Ceci peut s'expliquer par le fait qu'ils ne sont plus indépendants puisqu'ils sont régis en quelque sorte par le même agent décisionnel.

Enfin, remarquons que face à une politique de TFT (Figure 3.3), notre agent de *Q-learning* va être incité à coopérer. Ceci n'est pas surprenant puisque face à un TFT, la stratégie rationnelle et ego-centrée qui est optimale est celle consistant à rejoindre la coopération.



(a) (b)

FIGURE 3.2 – Apprentissage d'un *Q-learning* avec modèle identique (mêmes hyperparamètres que dans la Figure 3.1).



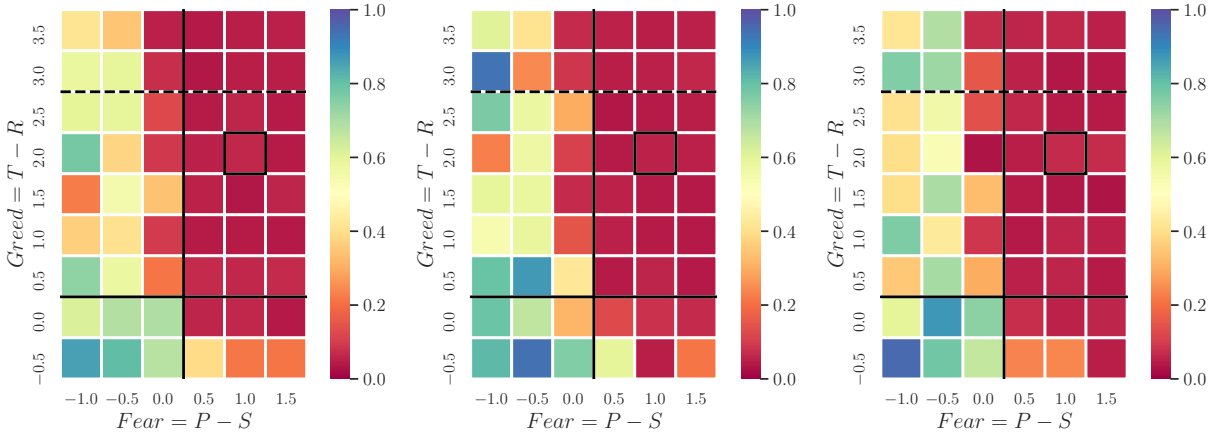
(a) (b)

FIGURE 3.3 – Apprentissage d'un *Q-learning* contre un TFT (mêmes hyperparamètres que dans la Figure 3.1).

3.3.4 Impact des valeurs des gains (S, P, R, T) sur la coopération

Il est intéressant de noter que les valeurs (S, P, R, T) de la littérature ont été définies de sorte qu'il existe deux types d'incitation à la défection : la crainte de se faire exploiter (*Fear* égale à $P - S$) et l'avidité (*Greed* égale à $T - R$). Pour insister sur le fait que ces deux types d'incitation à la défection sont néfastes pour la convergence des politiques de RL, nous proposons d'entraîner de telles politiques dans des IPD dans lesquels d'autres valeurs de (S, P, R, T) sont utilisées. Ainsi, dans la Figure 3.4, on représente les taux de coopération à l'issue d'un entraînement pour une politique de bandits (on a choisi l'algorithme EXP3 connu pour adresser des environnements non-stationnaires [ACBFS02]). Les entraînements sont effectués pour chacune des valeurs de la

crainte (resp. de l'avidité) dans $[-1, +1.5]$ (resp. dans $[-0.5, +3.5]$). On représente également les trois cas (indépendants, même modèle et contre un TFT). Nous représentons les résultats dans la Figure 3.4 et nous les discuterons un peu plus bas.



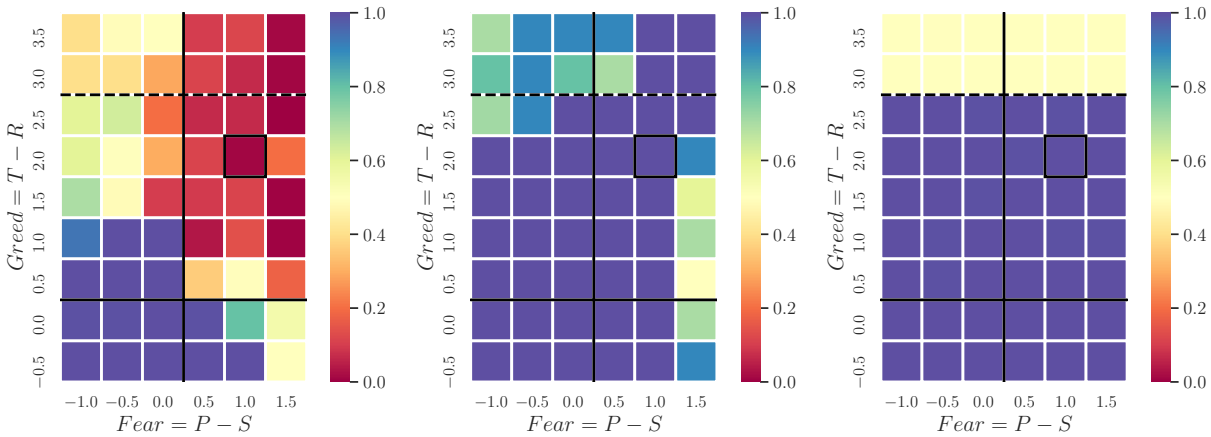
(a) Deux modèles différents : π_1 vs π_2

(b) Modèle partagé : π_1 vs π_1

(c) Contre un Tit-for-Tat : π_1 vs TFT

FIGURE 3.4 – Impact des valeurs (S, P, R, T) sur les taux de coopération dans un entraînement de bandits EXP3 (de paramètre $\gamma = 0.1$). Un tournoi de 500 épisodes. La case correspondant au dilemme du prisonnier classique est celle entourée par un carré en noir.

On procède exactement au même type d'entraînements groupés pour l'agent de *Q-learning* (Figure 3.4) avec modèles différents, identiques puis contre un TFT.



(a) Deux modèles différents : π_1 vs π_2

(b) Modèle partagé : π_1 vs π_1

(c) Contre un Tit-for-Tat : π_1 vs TFT

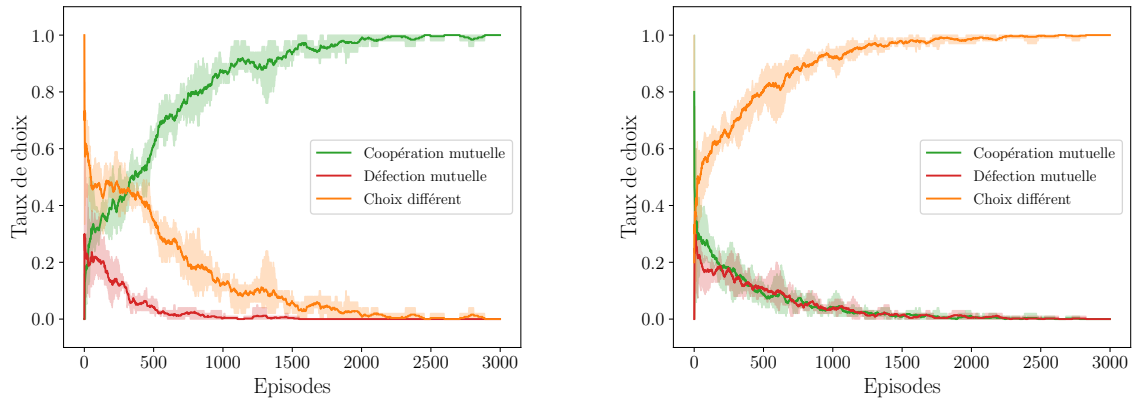
FIGURE 3.5 – Impact de (S, P, R, T) sur les taux de coopération dans un entraînement de *Q-learning*. Un tournoi de 3000 épisodes avec les mêmes hyperparamètres que dans la Figure 3.1

On peut observer dans la Figure 3.4 que les bandits EXP3 peinent à converger dès lors qu'il y a au moins une des deux incitations à défecter. En effet, les taux finaux sont faibles dans les parties du haut ou de droite, et sont tous quasi nuls (en rouge) dans la partie en haut à droite (quand la crainte et l'avidité sont positives).

En ce qui concerne les entraînements du *Q-learning*, les taux finaux de coopération dépendent davantage des valeurs (S, P, R, T) choisies. Elles sont globalement plus élevées dans le modèle partagé (identique). Avec des modèles indépendants, il y a une convergence vers la défection quand la crainte et l'avidité sont positives. Concernant le cas contre un TFT, cela dépend de la valeur de l'avidité que nous discutons plus bas.

Le cas $T + S > 2R$

Sur les Figures 3.4 et 3.5, les cas où $T + S > 2R$ sont représentés par les cases au-dessus de la ligne en pointillés (c'est-à-dire quand l'avidité $T - R$ est supérieure à $R - S = 3$). Ces cas correspondent aux situations ou l'alternance entre coopération et défection est préférée à la coopération mutuelle. Ce cas a déjà été étudié sous le nom du *Lift Dilemma* [DM96, DMB00]. On retrouve que face à un TFT, un agent de RL semble préférer cette alternance dès que la condition $T + S < 2R$ n'est plus respectée. Ainsi sur la Figure 3.6, on représente les deux cas limite $T + S < 2R$ (Figure 3.6a) et $T + S > 2R$ (Figure 3.6b).



(a) $2R > T + S$
avec $(S, P, R, T) = (0, 1, 3, 5)$

(b) $2R < T + S$
avec $(S, P, R, T) = (0, 1, 3, 6.5)$

FIGURE 3.6 – Cas du Lift Dilemma, un Q-learning face à un TFT

Précisons que pour le cas limite $(S, P, R, T) = (0, 1, 3, 6)$ donc avec $2R = T+S$, l'apprentissage se comporte comme le cas avec inégalité stricte $2R > T + S$ (Figure 3.6b).

3.4 Propositions d'une stratégie de Tit-for-Tat continu à deux joueurs

Comme motivé en introduction, une des contributions les plus importantes pour la suite de la thèse est celle qui apporte des améliorations au TFT continu à deux joueurs. Dans cette section, nous commençons par détailler le cadre de l'étude avec notamment l'introduction de métriques sociales permettant d'évaluer certains aspects des politiques que nous allons construire. Nous présentons ensuite nos améliorations avant de conduire quelques évaluations.

3.4.1 Contexte

Dans cette section, nous travaillons dans le contexte d'un dilemme du prisonnier continu (voir section 1.3.4 du chapitre 1) en version itérée. Pour rappel, deux joueurs A et B s'opposent : à chaque itération t, ils jouent simultanément les degrés de coopération $a_t \in [0, 1]$ et $b_t \in [0, 1]$. Notons que 0 représente une défection pure et 1 une coopération pure. Les deux joueurs obtiennent alors à chaque itération une récompense basée sur une interpolation des valeurs S, P, R, T (voir Eq 1.4).

Nous nous basons sur la baseline du *Damped Tit-for-Tat* (DTFT) introduit par [Ver98], sur laquelle nous apportons diverses modifications que nous détaillons puis évaluons dans la suite. Pour rappel, ce *Damped Tit-for-Tat* (DTFT) comporte un paramètre $r_0 \in]0, 1]$ d'incitation à coopérer qui est constant : si A joue cette stratégie face à B, alors à l'itération t, il jouera une action avec le degré de coopération suivant :

$$a_t = \begin{cases} 1.0 & \text{si } t = 0 \\ r_0.1 + (1 - r_0).b_{t-1} & \text{si } t > 0. \end{cases} \tag{3.8}$$

avec $r_0 \in]0, 1]$

En quelques mots, cette stratégie est "gentille" au sens où elle commence à coopérer et elle coopère même avec un taux $r_0 > 0$ quand l'opposant adopte une défection pure ($b_t = 0$). Elle est suffisamment rancunière et indulgente puisqu'elle se base avec un poids $1 - r_0$ sur la décision de l'opposant : donc si l'opposant arrête de coopérer, le degré de DTFT va descendre mais si l'opposant revient sur une coopération, le DTFT remontera son degré.

3.4.2 Proposition d'un TFT continu

Nous présentons ici nos propositions d'amélioration du DTFT vu plus haut.

Un taux d'incitation dynamique

Pour commencer, nous pouvons faire le constat qu'il est regrettable que le DTFT offre un taux de coopération constant r_0 si le comportement de l'opposant est résolument égoïste. C'est pourquoi nous proposons de rendre ce taux d'incitation dynamique afin de s'adapter au comportement de l'opposant. Ainsi, quand le joueur B défecte à l'étape précédente ($b_{t-1} = 0$), le joueur A offrira un taux dynamique r_t. Pour varier ce taux, considérons la différence entre ce que A vient d'offrir à B (a_{t-1}) et ce que B vient de lui offrir (b_{t-1}) : $\Delta = b_{t-1} - a_{t-1}$. Cette différence permet d'évaluer si l'opposant répond favorablement ou non à l'incitation à coopérer. Cette différence Δ permet de modifier r_t qui sera décrémenté (ou incrémenté selon le signe de Δ) avec un poids β : $r_t \leftarrow r_{t-1} + \beta\Delta$. Ainsi ce taux r_t peut rapidement monter ou chuter selon le comportement de l'opposant. Notons qu'il est nécessaire de projeter cette valeur sur $[0, 1]$ en appliquant un opérateur noté $[.]^+$ et décrit par : $[.]^+ : x \mapsto [x]^+ = \min(1, \max(0, x))$.

Un paramètre d'inertie

Si l'on fait face à un opposant avec des stratégies différentes, il peut arriver que les réponses de notre TFT soient assez instables. Nous proposons alors d'introduire un coefficient d'inertie α. Il sert à pondérer la réponse $\tilde{a_t}$ de notre TFT (sous-section précédente) par le poids $1 - \alpha$ et l'ancienne réponse que l'on a effectuée à l'étape précédente a_{t-1} par le poids α. Nous avons alors : $a_t = \alpha a_{t-1} + (1 - \alpha)\tilde{a_t}$.

Premier bilan

Dans un premier bilan, nous allons résumer nos deux premières contributions. On note $\text{TFT}_{\alpha,\beta,r_0,c_0}$ notre algorithme aux paramètres α et β. Il comporte également les paramètres c_0 et r_0 qui sont les valeurs initiales (pour $t = 0$) choisies pour le degré de coopération (a_t) et le taux d'incitation (r_t). On peut donc résumer nos deux premières contributions dans l'équation 3.9 :

$$\text{TFT}_{\alpha,\beta,r_0,c_0}(t, a_{t-1}, b_{t-1}), r_t = \begin{cases} c_0, r_0 & \text{if } t = 0 \\ \alpha a_{t-1} + (1-\alpha)(r_t + (1 - r_t)b_{t-1}), & \\ [r_{t-1} + \beta(b_{t-1} - a_{t-1})]^+ & \text{if } t > 0. \end{cases} \tag{3.9}$$

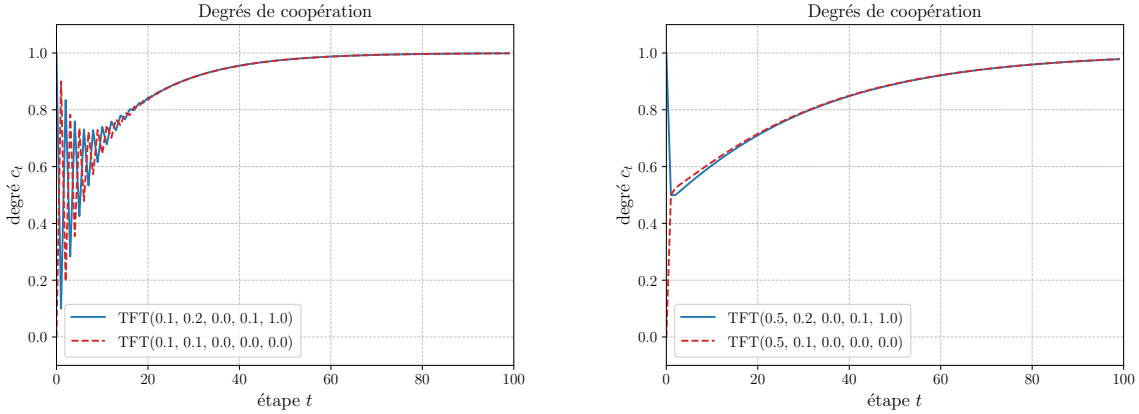
avec $[x]^+ = \min(1, \max(0, x))$

et $\alpha, \beta, r_0, c_0 \in [0, 1]^4$

Intéressons-nous à l'intérêt de ces paramètres. Dans la Figure 3.8, on représente un tournoi de 100 itérations opposant le TFT que nous avons développé à un Traître, *i.e.* un agent qui coopère totalement, puis subitement se met à défecter totalement à partir de $t > 20$. On compare alors

deux TFT avec ou sans paramètre β. Sur la partie gauche (Figures 3.8a,c,e), on représente les résultats des évaluations avec $\beta = 0$ (donc un taux d'incitation r_t constant) et à droite (Figures 3.8b,d,f), un TFT avec un paramètre β non nul. Les taux d'incitation sont représentés en haut, les degrés de coopération choisis sont représentés au milieu et les gains obtenus en bas lors du tournoi de dilemme du prisonnier itéré. On peut observer dans la Figure 3.8a que le taux r_t reste constant égal à 0.2 ce qui a pour conséquence de générer un degré de coopération aussi égal à 0.2 (Figure 3.8c) et donc de se faire un peu exploiter quand on regarde la courbe des gains (Figure 3.8e). En revanche, on peut observer qu'avec le paramètre β, le taux r_t chute très rapidement à 0.0 (Figure 3.8b) ainsi que le degré de coopération (Figure 3.8d) ce qui a pour conséquence de rejoindre rapidement le gain du Traître (Figure 3.8f), et donc notre agent est moins exploité.

L'intérêt du coefficient d'inertie α peut s'observer (Figure 3.7) quand les joueurs proposent des choix éloignés et donc sont sujets à un va et vient fréquent. Un coefficient d'inertie va alors lisser le comportement pour le rendre ainsi moins instable.



(a) $\alpha = 0.1$ pour les deux joueurs

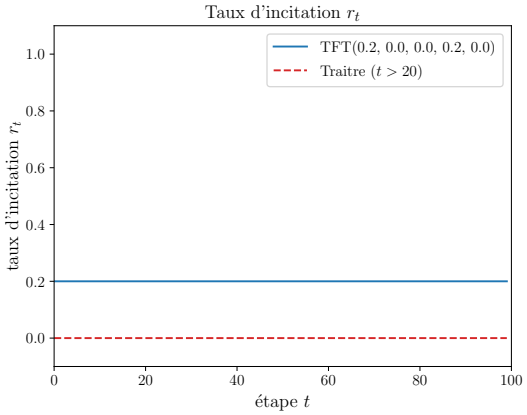
(b) $\alpha = 0.5$ pour les deux joueurs

FIGURE 3.7 – Intérêt du paramètre d'inertie α. Deux TFT différents s'opposent : TFT$(\alpha, \beta = 0.2, r_0 = 0.1, c_0 = 1.0)$ contre TFT$(\alpha, \beta = 0.1, r_0 = 0.0, c_0 = 0.0)$. Les deux figures montrent la simulation avec deux valeurs différentes de α (0.1 ou 0.5).

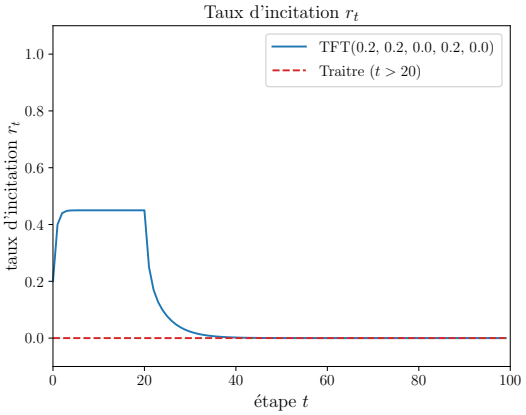
On peut remarquer dans la Figure 3.7 qu'avec un coefficient d'inertie plus important ($\alpha = 0.5$), les réactions sont rendues plus fluides mais la convergence se fait moins rapidement.

Un paramètre stochastique d'incitation

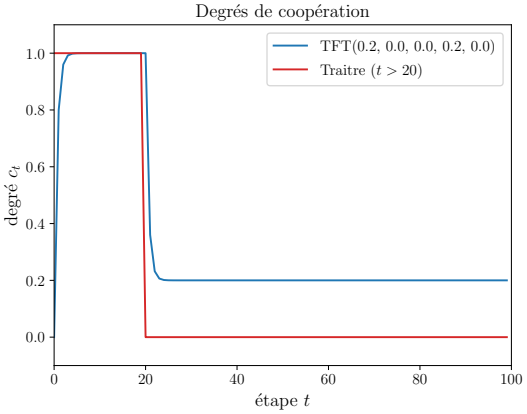
Dans les précédentes versions du Tit-for-Tat, un problème peut survenir quand les choix des deux joueurs ont convergé mutuellement vers un taux de coopération strictement nul $a_t = b_t = 0.0$ ainsi que leurs taux d'incitation respectifs $r_t^a = r_t^b = 0.0$. Dans ce cas, il devient donc impossible



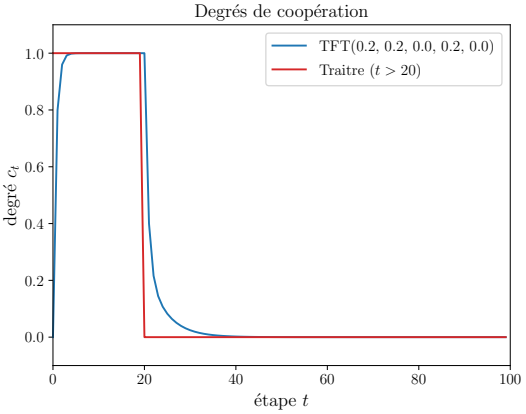
(a) Taux d'incitation r_t, quand $\beta = 0$



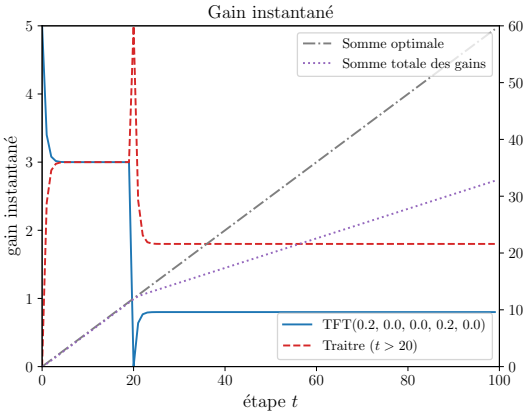
(b) Taux d'incitation r_t, quand $\beta > 0$



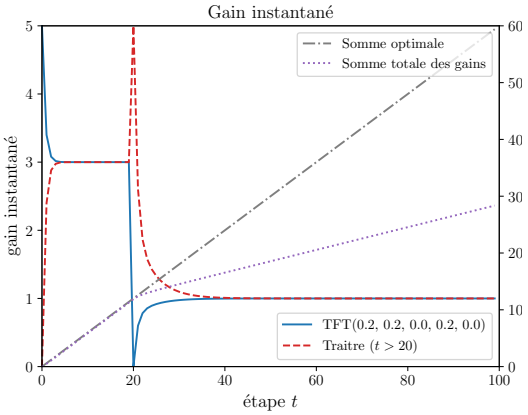
(c) Choix de degrés, quand $\beta = 0$



(d) Choix de degrés, quand $\beta > 0$



(e) Gains obtenus, quand $\beta = 0$



(f) Gains obtenus, quand $\beta > 0$

FIGURE 3.8 – Intuition de l'intérêt du paramètre adaptatif β

en l'état de décoincer la situation. C'est pourquoi, nous introduisons un coefficient stochastique qui permet avec une probabilité γ de réinitialiser le taux d'incitation à r_0 s'il est devenu nul. Cette proposition peut être formalisée par l'addition du terme $r_0\mathcal{B}(1,\gamma)\mathbb{1}_{r_{t-1}=0}$ avec $\mathcal{B}(1,\gamma)$ une variable de Bernoulli qui vaut 1 avec une probabilité γ et 0 sinon. L'indicatrice $\mathbb{1}_{r_{t-1}=0}$ ou $\mathbb{1}_{\{0\}}(r_{t-1})$ vaut 1 si $r_{t-1}=0$ et 0 sinon.

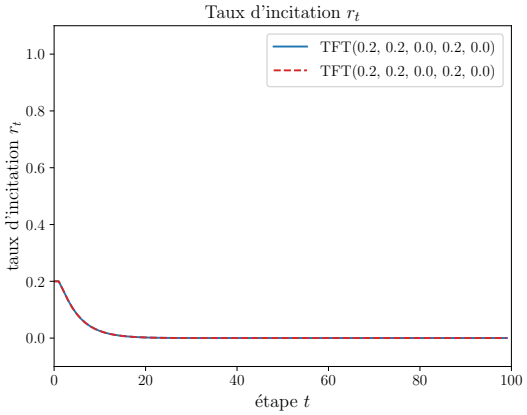
La Figure 3.9 illustre l'intérêt de ce coefficient stochastique γ. On considère deux TFT qui se retrouvent dans une situation de défection mutuelle depuis un certain moment. Le choix de leurs degrés est donc nul ($a_t = b_t = 0.0$) ainsi que leurs taux d'incitation ($r_t^A = r_t^B = 0.0$). Cette situation est problématique puisque les deux joueurs sont coincés : chacun des deux joueurs n'a plus la possibilité d'inciter l'autre. Dans les Figures 3.9a et 3.9b, on peut observer que le taux d'incitation des deux joueurs a convergé vers 0.0 aux alentours de l'étape $t = 20$. Dans le cas $\gamma = 0.0$, leurs taux d'incitation restent donc nuls ainsi que leurs degrés de coopération (Figure 3.9c). Avec un coefficient stochastique égal à $\gamma = 0.05$, on peut observer que l'agent rouge a l'occasion de réinitialiser son taux d'incitation à 0.2 ce qui a pour conséquence immédiate de relancer le taux r_t de son partenaire (Figure 3.9b). Ainsi leurs degrés de coopération tendent à leur tour vers 1.0 (Figure 3.9d) ce qui permet une croissance des gains vers la valeur 3 ce qui est un optimal de Pareto (Figure 3.9f).

3.4.3 Bilan

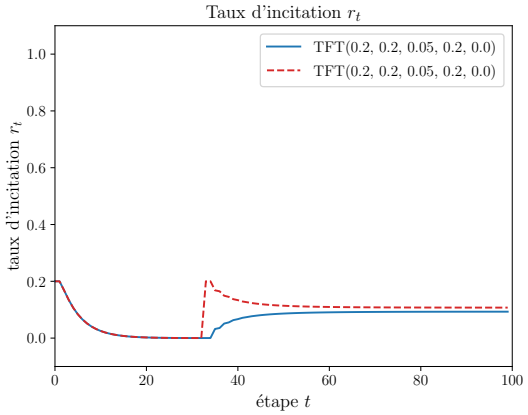
Rappelons que nous pouvons résumer nos contributions du Tit-for-Tat continu par cette formulation :

$$\mathrm{TFT}_{\alpha,\beta,\gamma,r_0,c_0}(t,a_{t-1},b_{t-1}),r_t = \begin{cases} c_0, r_0 & \text{si } t=0 \\ \alpha a_{t-1} + (1-\alpha)(r_t + (1-r_t)b_{t-1}), & \\ [r_{t-1} + \beta(b_{t-1}-a_{t-1})]^+ + r_0\mathcal{B}(1,\gamma)\mathbb{1}_{r_{t-1}=0} & \text{si } t>0. \end{cases} \tag{3.10}$$

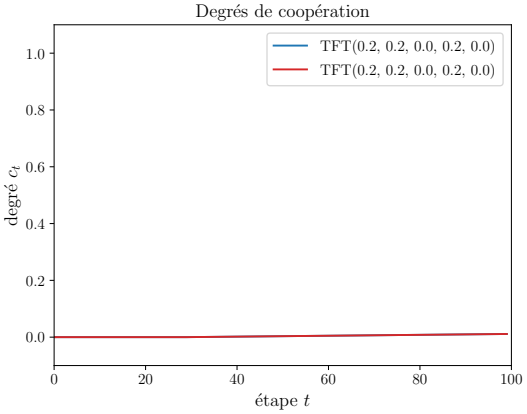
Dans la suite, cette fonction sera notée $\mathrm{TFT}_{\alpha,\beta,\gamma,r_0,c_0}$ ou $\mathrm{TFT}(\alpha,\beta,\gamma,r_0,c_0)$ pour plus de lisibilité dans les figures. Précisons également que lorsque deux joueurs sont impliqués avec deux fonctions de TFT différentes, nous noterons respectivement $(\alpha^A,\beta^A,\gamma^A,r_0^A,c_0^A)$ et $(\alpha^B,\beta^B,\gamma^B,r_0^B,c_0^B)$ les paramètres des joueurs A et B. Par ailleurs, les taux d'incitation sont notés r_t^A et r_t^B quand les degrés de coopération sont notés a_t et b_t. Notons également, que le Tit-for-Tat continu DTFT proposé dans [Ver98] (section 1.4.4) peut être noté $\mathrm{TFT}(\alpha=0,\beta=0,\gamma=0,r_0,c_0=1)$. Il a ainsi pour seul paramètre, le taux de d'incitation (constant) r_0.



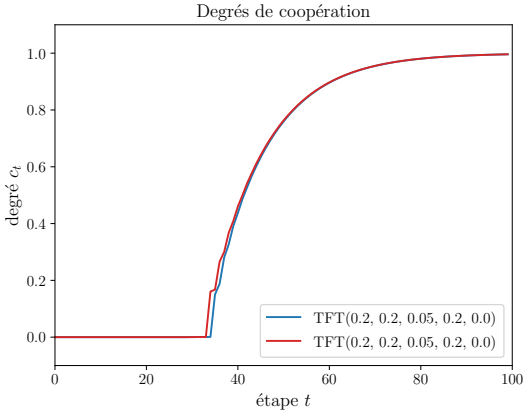
(a) Taux d'incitation r_t, quand $\gamma = 0$



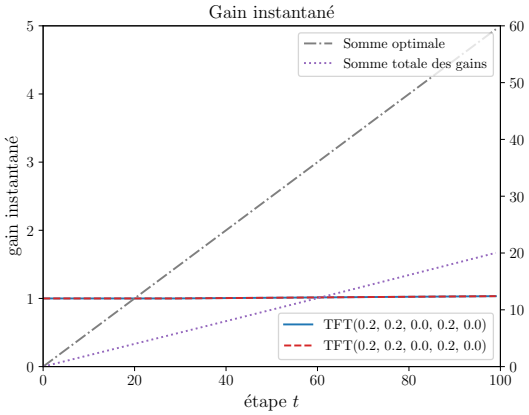
(b) Taux d'incitation r_t, quand $\gamma > 0$



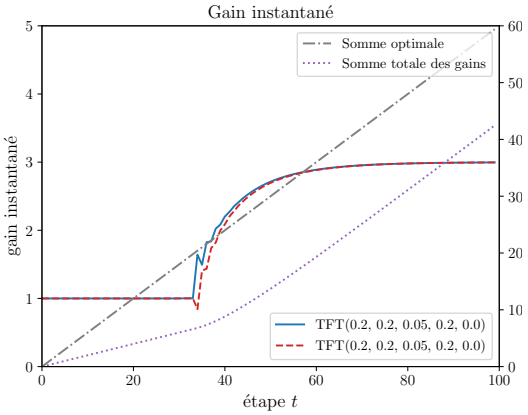
(c) Choix de degrés, quand $\gamma = 0$



(d) Choix de degrés, quand $\gamma > 0$



(e) Gains obtenus, quand $\gamma = 0$

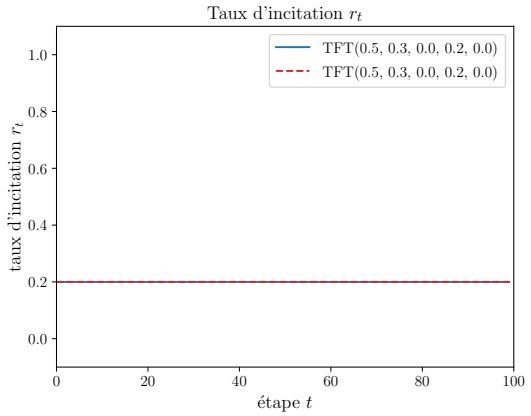


(f) Gains obtenus, quand $\gamma > 0$

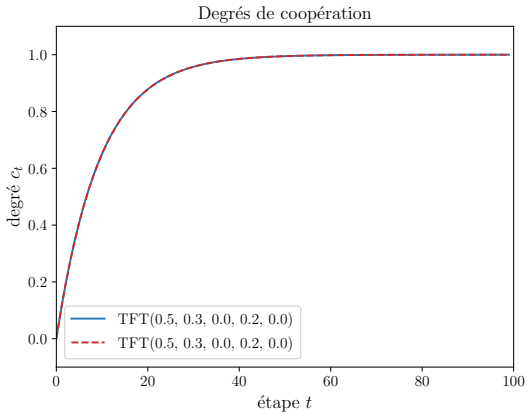
FIGURE 3.9 – Intuition de l'intérêt du paramètre stochastique γ

3.4.4 Simulations empiriques

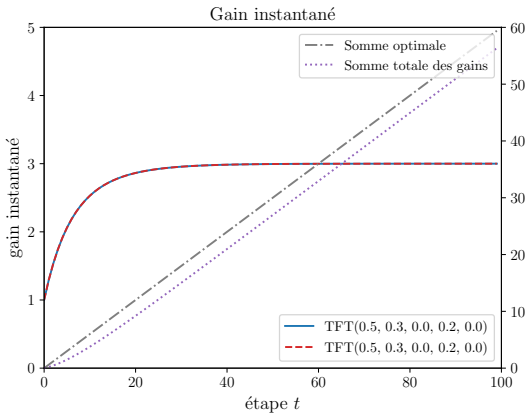
Avant d'évaluer certains paramètres de notre fonction de TFT dans la prochaine section, commençons par observer empiriquement la convergence de nos fonctions. On reprend le cadre méthodologique de notre IPD continu opposant deux joueurs.
Dans un premier temps, observons le comportement des stratégies de TFT quand ses paramètres sont égaux d'un agent à l'autre.



(a) Taux d'incitation



(b) Degrés de coopération

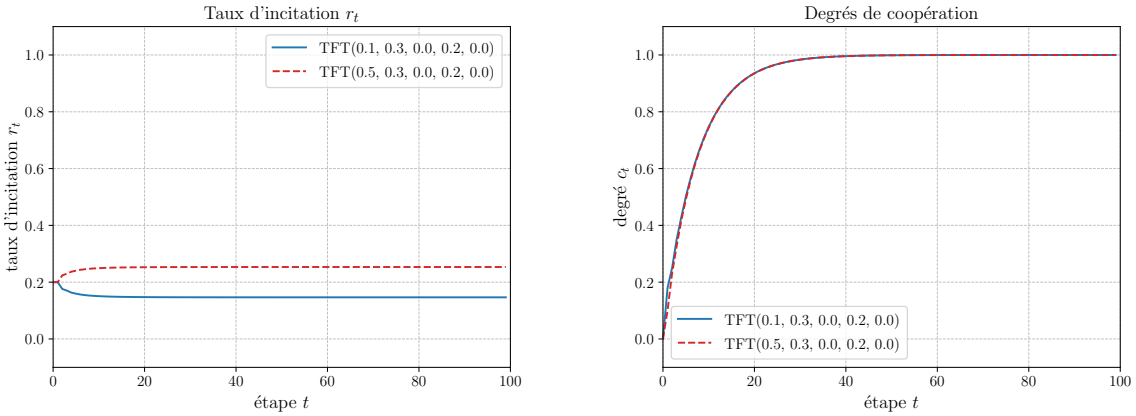


(c) Gains

FIGURE 3.10 – Simulation de deux TFT aux paramètres identiques : TFT($\alpha = 0.5, \beta = 0.3, \gamma = 0, r_0 = 0.2, c_0 = 0.0$)

Nous pouvons dans un premier temps observer que les degrés de coopération (*i.e.* le choix des joueurs) tendent mutuellement vers 1.0 (Figure 3.10b). Il n'est pas utile de procéder à davantage de simulations puisque nous le démontrerons dans la section 3.4.6.

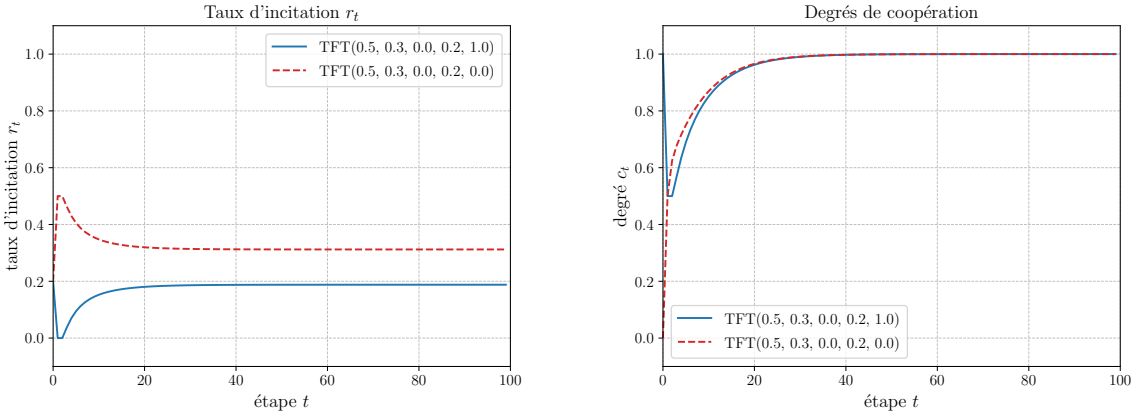
Considérons maintenant le cas où les paramètres sont différents. Soient deux TFT de paramètres $(\alpha^A, \beta^A, \gamma^A, r_0^A, c_0^A)$ et $(\alpha^B, \beta^B, \gamma^B, r_0^B, c_0^B)$. Il est intéressant d'étudier ce cas, ce qui reviendrait dans une situation un peu pratique au cas où les joueurs ont choisi des politiques différentes. Nous étudions ici quelques cas, notamment quand les agents choisissent : un degré initial c_0 radicalement différent, un taux d'incitation initial r_0 différent ou bien des paramètres d'inertie α et adaptatif β différents.



(a) Taux d'incitation (b) Degrés de coopération

FIGURE 3.11 – Paramètre d'inertie différent $\alpha \in \{0.1, 0.5\}$

On peut remarquer, dans la Figure 3.11, qu'un coefficient α assez différent joue légèrement sur les valeurs de convergence des taux d'incitation qui restent non nuls, ce qui est l'essentiel pour la convergence des degrés de coopération.

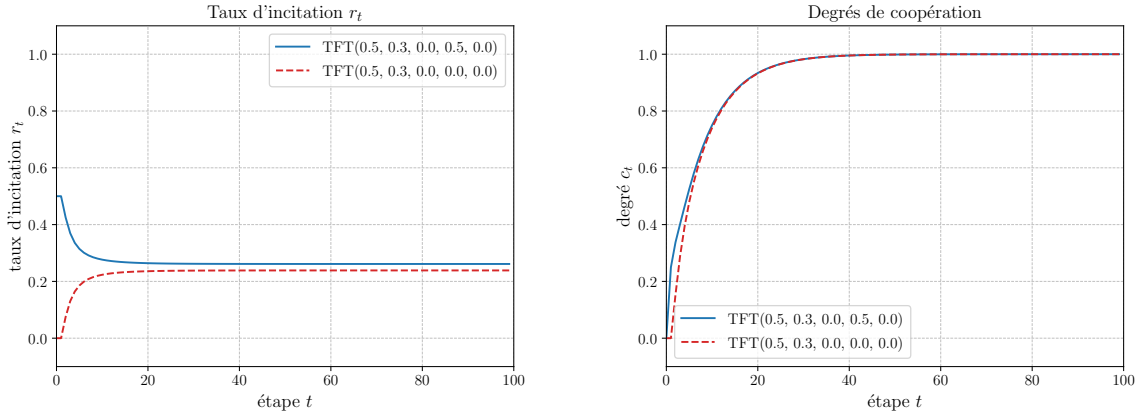


(a) Taux d'incitation (b) Degrés de coopération

FIGURE 3.12 – Degré de coopération initial différent : $c_0 \in \{0.0, 1.0\}$

Nous observons que même dans le cas où l'un des agents commence par coopérer totalement

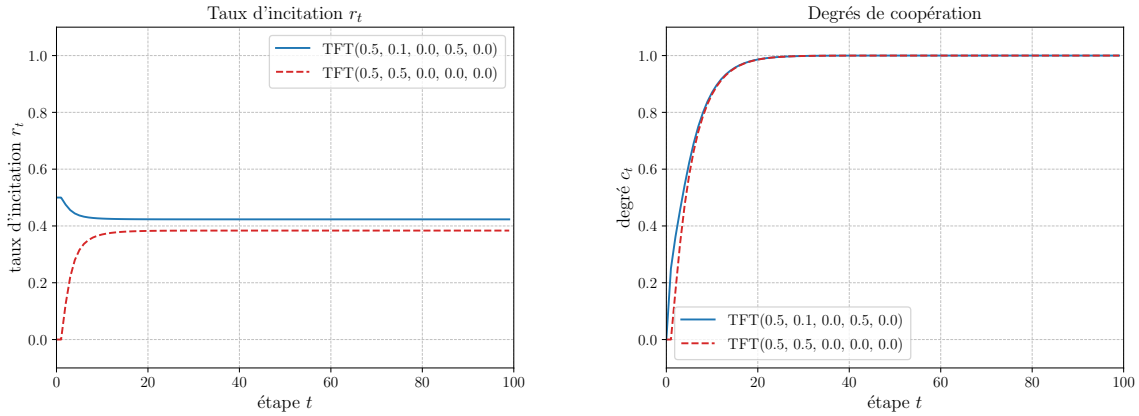
($c_0 = 1.0$) et l'autre pas du tout ($c_0 = 0.0$), les taux d'incitation dynamique permettent de faire tendre le jeu vers un accord commun (Figure 3.12).



(a) Taux d'incitation (b) Degrés de coopération

FIGURE 3.13 – Taux d'incitation initial différent : $r_0 \in \{0.0, 0.5\}$

Comme pour le cas où les degrés de coopération initiaux c_0 sont différents, si les agents choisissent un taux d'incitation différent r_0, on peut observer, dans la Figure 3.13, que le taux d'incitation dynamique r_t est construit de telle sorte qu'il tende vers une valeur fixe non nulle et ainsi qu'il permette de faire de tendre les degrés de coopération vers 1.0.



(a) Taux d'incitation (b) Degrés de coopération

FIGURE 3.14 – Coefficients r_0 et β différents : $(r_0, \beta) \in \{(0.5, 0.1), (0.0, 0.5)\}$

Dans le cas où le paramètre β est choisi différent (avec un taux d'incitation r_0 choisi aussi différent car sinon ça ne change pas du cas où les paramètres sont tous identiques), on peut observer dans la Figure 3.14 que les taux d'incitation dynamiques tendent chacun vers une valeur fixe non nulle et qu'ils sont pondérées par leur valeur respective de β.

Empiriquement, on peut conclure que nos fonctions de TFT continues semblent converger vers la valeur 1.0 dès lors que que les taux d'incitation dynamiques sont non nuls. Ceci arrive dans les cas suivants : soit les deux agents ont choisi un taux d'incitation initial r_0 non nul, soit au moins un des deux agents a choisi un paramètre r_0 non nul et que les deux agents ont un paramètre β non nul. Ce choix leur permet de modifier leur taux d'incitation dynamique en particulier pour l'agent qui l'avait choisi initialement nul.

3.4.5 Évaluation des paramètres

Après avoir simulé les convergences des stratégies de TFT, il est intéressant d'évaluer les paramètres de notre fonction. Pour ce faire, nous évaluons le comportement des stratégies en utilisant les métriques sociales introduites en section 3.2.2, à savoir l'efficacité E, la rapidité Sp, l'incitation IC, la sûreté Sf, l'indulgence Fg et la résilience Rl. Nous évaluerons quelques paramètres et discuterons des résultats.

Paramètres					Métriques (×100)					
α	β	γ	r_0	c_0	E	Sp	IC	Sf	Fg	Rl
0.0	0.0	0.0	0.1	0.0	92.4	62.8	78.7	−9.9	98.7	90.5
0.0	0.0	0.0	0.2	0.0	96.1	78.2	58.9	−19.8	98.9	95.9
0.0	0.0	0.0	0.3	0.0	97.4	84.3	39.1	−29.7	99.0	97.6
0.0	0.0	0.0	0.4	0.0	98.0	87.3	19.3	−39.6	99.1	98.5
0.0	0.0	0.0	0.3	0.0	97.4	84.3	39.1	−29.7	99.0	97.6
0.0	0.05	0.0	0.3	0.0	97.4	84.3	86.6	−6.0	98.7	85.5
0.0	0.1	0.0	0.3	0.0	97.4	84.3	92.5	−3.0	98.6	47.2
0.0	0.2	0.0	0.3	0.0	97.4	84.3	95.5	−1.5	98.6	2.4
0.0	0.3	0.0	0.3	0.0	97.4	84.3	96.5	−1.0	98.6	0.1
0.0	0.0	0.0	0.3	0.0	97.4	84.3	39.1	−29.7	99.0	97.6
0.2	0.0	0.0	0.3	0.0	96.7	81.2	38.9	−29.6	98.8	97.1
0.4	0.0	0.0	0.3	0.0	95.7	76.3	38.5	−29.5	98.3	96.2
0.6	0.0	0.0	0.3	0.0	93.6	67.4	37.8	−29.2	97.5	94.3
0.4	0.0	0.0	0.3	0.0	95.7	76.3	38.5	−29.5	98.3	96.2
0.4	0.0	0.0	0.3	0.5	98.2	87.2	38.1	−30.3	98.3	96.2
0.4	0.0	0.0	0.3	1.0	100.0	95.0	37.7	−31.2	98.3	96.2
0.0	0.0	0.0	0.3	0.0	97.4	84.3	39.1	−29.7	99.0	97.6
0.0	0.0	0.0	0.3	0.5	98.9	90.5	38.9	−30.2	99.0	97.6
0.0	0.0	0.0	0.3	1.0	100.0	95.0	38.6	−30.7	99.0	97.6

TABLE 3.2 – Métriques sociales selon les paramètres du TFT. Ici, on évalue séparément l'impact des paramètres r_0, puis β, et enfin α et c_0.

Pour commencer, nous étudions l'impact du taux d'incitation initial r_0 (quatrième colonne, premier groupe de la Table 3.2), en fixant les autres paramètres à 0.0. On remarque alors que si

r_0 augmente, l'efficacité E et la rapidité Sp augmentent, mais au détriment de la sûreté (*safety*) Sf et de l'incitation à coopérer (*Incentive-comptatibility*) IC. Il est donc intéressant d'évaluer l'intérêt de notre coefficient adaptatif β (deuxième colonne, deuxième groupe de la table). On observe que dès lors que le paramètre a une valeur non-nulle, il a un effet immédiat puisqu'il permet de remonter significativement la sûreté et l'incitation. Lorsque sa valeur augmente, ces deux dernières métriques augmentent sans diminuer l'efficacité ni la vitesse. En revanche, cela se fait au détriment de la résilience puisque les agents ont alors tendance à être plus méfiants quand β augmente. Nous verrons dans la Table 3.3 que le paramètre stochastique γ permet d'adresser ce problème. Nous allons maintenant analyser l'impact du paramètre d'inertie α (première colonne, troisième groupe de la table). On remarque qu'un coefficient d'inertie ne fait que diminuer la vitesse de convergence vers l'équilibre. Cela n'a pas d'importance particulière si ce n'est de stabiliser les comportements comme on l'a vu précédemment. Enfin, en ce qui concerne la valeur c_0 (le choix initial de coopération), on peut voir les résultats dans le quatrième groupe (cinquième colonne) de la table, on peut observer que cela n'a pour effet que d'augmenter légèrement la vitesse de convergence.

En ce qui concerne l'analyse du coefficient stochastique γ, nous procédons à des simulations similaires en plusieurs épisodes pour étudier la variabilité due à la stochasticité.

Paramètres					Métriques ($\times 100$)					
α	β	γ	r_0	c_0	E	Sp	IC	Sf	Fg	Rl
0.5	0.3	0.0	0.3	0	94.9 ± 0	72.5 ± 0	95.0 ± 0.0	-1.0 ± 0.0	97.1 ± 0.0	0.0 ± 0.0
0.5	0.3	0.01	0.3	0	94.9 ± 0	72.5 ± 0	92.9 ± 2.4	-2.1 ± 1.2	97.1 ± 0.0	38.6 ± 34.1
0.5	0.3	0.03	0.3	0	94.9 ± 0	72.5 ± 0	91.5 ± 1.9	-2.7 ± 1.0	97.2 ± 0.2	58.9 ± 36.2
0.5	0.3	0.05	0.3	0	94.9 ± 0	72.5 ± 0	89.5 ± 2.6	-3.8 ± 1.3	97.2 ± 0.2	70.4 ± 19.6
0.5	0.3	0.08	0.3	0	94.9 ± 0	72.5 ± 0	88.1 ± 2.8	-4.4 ± 1.4	97.2 ± 0.2	75.2 ± 20.0

TABLE 3.3 – Métriques sociales selon le paramètre stochastique γ du TFT. Evalutions sur 10 épisodes pour calculer les écarts-types.

On peut observer dans la Table 3.3 que le coefficient stochastique γ permet d'adresser les problèmes de résilience Rl. On rappelle que γ est la probabilité de réinitialiser le taux r_t quand il est nul. Les évaluations montrent que quand on augmente la valeur de γ, on gagne de la résilience, mais inévitablement on perd en sûreté et incitation.

3.4.6 Quelques aspects théoriques

Nous proposons ici de montrer quelques résultats sous un angle théorique. Pour commencer, supposons simplement que deux TFT de même paramètre jouent l'un face à l'autre. Montrons alors qu'ils tendent vers une coopération mutuelle ce qui implique des degrés de coopération égaux à 1.0.

Convergence de la coopération de $\text{TFT}_{\alpha,\beta,0,r_0,c_0}$

Proposition 3.1 *: Soient deux joueurs A et B, jouant la même politique $TFT_{\alpha,\beta,\gamma,r_0,c_0}$. En notant a_t et b_t les choix de A et B à l'étape t, nous avons alors que :*

$$\begin{aligned} &\forall t, \quad a_t = b_t \\ &a_t = 1 - Q^t(1 - c_0) \\ &\textit{avec } Q = \alpha + (1-\alpha)(1-r_0) \end{aligned} \tag{3.11}$$

Preuve

Supposons dans un premier temps que les joueurs A et B utilisent la même politique $\text{TFT}_{\alpha,\beta,\gamma,r_0,c_0}$, nous avons donc $\forall t, a_t = b_t$. Commençons alors par nous concentrer sur l'expression (commune) de r_t. On peut supposer que $\gamma = 0.0$ car γ n'a d'effet que si $r_t = 0$. Nous avons donc $r_t = r_0$ (car $\beta(b_{t-1} - a_{t-1}) = 0$). Grâce à l'équation 3.9, nous pouvons donc réécrire l'expression de a_t (ou b_t) au cours du temps :

$$\begin{aligned} a_t &= \alpha a_{t-1} + (1-\alpha)(r_0 + (1-r_0)a_{t-1}) \\ &= [\alpha + (1-\alpha)(1-r_0)]a_{t-1} + (1-\alpha)r_0 \\ &= Qa_{t-1} + R \\ \text{avec} \quad & Q = \alpha + (1-\alpha)(1-r_0) \text{ et } R = (1-\alpha)r_0 \end{aligned} \tag{3.12}$$

On remarque que la suite de terme a_t est une suite arithmético-géométrique de raisons Q et R. En notant $W = \frac{R}{1-Q}$ et en rappelant que $a_0 = c_0$, la solution de l'expression de a_t est donnée par :

$$a_t = Q^t(c_0 - W) + W \tag{3.13}$$

$$\begin{aligned} \text{Or } W &= \frac{R}{1-Q} \\ &= \frac{(1-\alpha)r_0}{1-(\alpha+(1-\alpha)(1-r_0))} \\ &= \frac{(1-\alpha)r_0}{(1-\alpha)(1-(1-r_0))} \\ &= \frac{(1-\alpha)r_0}{(1-\alpha)r_0} \\ &= 1 \end{aligned} \tag{3.14}$$

L'expression finale de a_t peut alors s'écrire :

$$\begin{aligned} &a_t = 1 - Q^t(1 - c_0) \\ &\text{avec } Q = \alpha + (1-\alpha)(1 - r_0) \end{aligned} \tag{3.15}$$

□

Il est intéressant également de montrer que notre TFT est robuste à la défection, notamment que face un défecteur pur, la stratégie TFT tend vers un degré de coopération égal à 0.0 et obtenir un majorant de cette décroissance vers 0.0.

Sûreté de TFT$_{\alpha,\beta,0,r_0,c_0}$

Proposition 3.2 *: Soit un joueur A jouant a_t en suivant la politique $TFT_{\alpha,\beta,0,r_0,c_0}$ face à un joueur B défecteur pur ($\forall t,\ b_t = 0.0$).*
Si le coefficient $\beta \geq \frac{r_0(1-\alpha)}{c_0}$, alors il existe une étape τ tel que $\forall t \geq \tau,\ r_t = 0.0$ et par conséquent $\lim_{t\to+\infty} a_t = 0$

Preuve

Pour commencer, si l'on suppose $\forall t,\ b_t = 0.0$, nous avons grâce à l'équation 3.9 :

$$\begin{aligned} a_t &= \alpha a_{t-1} + (1-\alpha)(r_t + (1 - r_t)b_{t-1}) \\ &= \alpha a_{t-1} + (1-\alpha)r_t \\ &\geq \alpha a_{t-1} \text{ car } \alpha \in [0,1] \end{aligned} \tag{3.16}$$

Avec cette suite géométrique, en rappelant $a_0 = c_0$, nous pouvons obtenir un minorant pour la suite a_t :

$$\forall t,\ a_t \geq c_0\alpha^t \tag{3.17}$$

Cherchons maintenant un majorant pour r_t, toujours en supposant que $\forall t,\ b_t = 0.0$, et en utilisant l'expression de r_t (équation 3.9), nous avons :

$$\begin{aligned} r_t &= r_{t-1} + \beta(b_{t-1} - a_{t-1}) \\ &= r_{t-1} - \beta a_{t-1} \\ &\leq r_{t-1} - \beta c_0 \alpha^{t-1} \end{aligned} \tag{3.18}$$

D'où,

$$\forall u \geq 1, \quad r_u - r_{u-1} \leq -\beta c_0 \alpha^{u-1} \tag{3.19}$$

En additionnant les différents termes de l'inégalité 3.19 pour faire apparaître un majorant de

r_t, et on utilisant la somme des termes de la suite géométrique de raison α, on obtient :

$$\begin{aligned} \forall t \geq 1, \quad & \sum_{u=1}^{t} r_u - r_{u-1} \leq -\beta c_0 \sum_{u=1}^{t} \alpha^{u-1} \\ & \iff r_t - r_0 \leq -\beta c_0 \frac{1-\alpha^t}{1-\alpha} \\ & \iff r_t \leq r_0 - \beta c_0 \frac{1-\alpha^t}{1-\alpha} \end{aligned} \tag{3.20}$$

Par conséquent, la condition $\beta \geq \frac{r_0(1-\alpha)}{c_0}$ est une condition suffisante (non nécessaire) pour qu'il existe τ pour laquelle l'expression $r_0 - \beta c_0 \frac{1-\alpha^\tau}{1-\alpha}$ est devenue négative et donc, comme le minimum de r_t est fixé à 0 que : $\forall t \geq \tau,\ r_t = 0$.

Enfin, $\forall t \geq \tau, \quad a_t = a_\tau \alpha^{(t-\tau)}$ et ainsi $\lim_{t \to +\infty} a_t = 0$

□

Remarquons que nous n'avons pas démontré la convergence dans les cas où les paramètres seraient différents. Nous nous sommes contentés de quelques simulations en faisant varier les paramètres. Il pourra s'agir en perspectives de vérifier si cette conjecture est vraie et de le démontrer le cas échéant.

3.5 Conclusions et perspectives

Dans ce chapitre, nous avons conduit une étude sur un dilemme du prisonnier itéré continu ou discret à deux joueurs. Un premier objectif a été de déterminer empiriquement si des politiques d'apprentissage par renforcement (RL) parvenaient à apprendre à choisir des stratégies coopératives. Comme le consensus de la littérature s'accorde sur le fait que les politiques de RL tendent vers l'équilibre de Nash qui est la défection mutuelle, nous nous sommes penchés vers l'amélioration des fonctions de TFT continues. Nous avons abouti à une fonction à deux variables (degrés de coopération du tour précédent) qui comporte une variable interne r_t qui est un taux d'incitation dynamique. Notre fonction est paramétrée par cinq coefficients ou valeurs initiales : un coefficient d'inertie α, un taux d'incitation initial r_0, un degré de coopération initial c_0, un coefficient adaptatif β qui modifie le taux dynamique r_t et enfin un coefficient stochastique γ qui permet de réinitialiser de manière aléatoire le taux d'incitation dynamique lorsque celui-ci s'est retrouvé nul.

Les observations et les évaluations menées avec des métriques sociales ont montré qu'un taux β non nul permettait à la politique d'être plus sûre face à la défection et donc plus incitative.

Cependant, il faudrait idéalement le compléter par un coefficient γ pour pallier les situations où une défection mutuelle et durable s'est installée. Enfin, un coefficient d'inertie α permet de lisser les comportements. Ce paramètre doit être choisi assez faible pour ne pas limiter la vitesse de convergence vers la coopération mutuelle.

Dans le prochain chapitre, nous nous servirons de cette fonction de TFT continu pour concevoir un algorithme permettant d'adresser des jeux de dilemmes du prisonnier asymétriques et circulaires à plus de deux joueurs.

Chapitre 4

Proposition d'un dilemme du prisonnier à N joueurs continu et non-réciproque

Dans ce chapitre, nous présentons nos contributions sur un nouveau formalisme de dilemme du prisonnier (DP) multi-joueurs basé sur des graphes orientés. Nous proposons également une extension de l'algorithme du Tit-for-Tat (TFT) à base d'une approche de réseau de flot maximal.

4.1 Introduction et motivations

Face à un dilemme social, la plus simple et la plus efficace [AH81] des stratégies à adopter reste la stratégie du Tit-for-Tat introduite par Anatole Rappoport [RCO65]. Aussi appelée "donnant-donnant" ou "prêté-rendu", cette stratégie consiste simplement à commencer à coopérer puis à reproduire le choix précédent de l'adversaire. Cependant, dès lors qu'une coopération n'est plus bilatérale, cette stratégie appliquée comme telle n'est plus efficace puisqu'elle se base sur une réponse réciproque.

4.1.1 Les coopérations réciproques non systématiques

Dans notre quotidien, on considère souvent l'argent comme notre seule monnaie d'échange. Or, parfois, on peut donner, aider, rendre service sans pouvoir recevoir directement par la personne qu'on a aidée. Cependant, on peut supposer que l'on recevra en retour ce que l'on a "donné" via d'autres personnes ou moyens [1].

1. Avec ce beau plan séquence de cinq minutes `https://youtu.be/nwAYpLVyeFU`, ce clip issu d'une association caritative met en scène plusieurs personnes se rendant service successivement de manière unilatérale. La chaîne de coopération s'achève à la fin de la vidéo quand la dernière personne finit par rendre un service au premier protagoniste, formant ainsi un cycle de coopération et donnant alors tout son sens au titre de la vidéo : *Kindness Boomerang*

Plus formellement, il peut arriver dans certaines situations qu'une coopération réciproque et équitable n'est pas systématiquement possible. En effet, on peut parfois coopérer via un ou plusieurs acteurs tiers et ainsi envisager des coopérations mutuelles le long de cycles d'acteurs dans lesquels la stratégie qui consiste à rendre service à un autre acteur est perdante mais devient gagnante si ce service est rendu d'une façon ou d'une autre. Par exemple, on peut imaginer une situation à trois joueurs A, B et C, dans laquelle une coopération optimale s'effectue sous la forme du cycle suivant : A a la possibilité d'aider B, qui peut aider C et enfin ce dernier peut aider A. Les services réciproques (par exemple "B aide A", "A aide C" et "C aide B") sont moins optimaux voire impossibles ou contre-productifs. Par ailleurs, chaque service rendu entraîne une perte d'utilité pour l'acteur altruiste mais inférieure au gain du bénéficiaire du service. Nous sommes donc en présence d'un dilemme social circulaire.

4.1.2 Un Tit-for-Tat classique inadapté

Nous rappelons que le principe de base du TFT est d'adapter la coopération qu'un agent A choisit face à un partenaire B selon la coopération que ce même partenaire a précédemment choisie. En version discrète, A débute par la coopération puis reproduit la décision précédente de l'opposant B. De ce fait, cette stratégie vise à la fois à inciter la coopération ainsi qu'à empêcher d'être exploité par un défecteur. En version continue, les actions sont remplacées par un degré de coopération $c \in [0, 1]$ et la réponse est définie par une fonction dont les variables sont les degrés précédents. Dans le chapitre 3, nous avons mené une étude de différents types de TFT continus. L'équilibre vertueux est alors atteint quand une confiance réciproque s'est stabilisée et que la coopération est devenue mutuelle. Dans la Figure 4.1, dans la situation du haut à deux joueurs A et B, il y a un potentiel de coopération purement réciproque. A et B ont la possibilité de coopérer de manière réciproque, ainsi le TFT est adapté et converge vers une coopération mutuelle. Cependant, dans le cas où la coopération réciproque directe n'est pas possible, mais peut néanmoins être rendue via un ou plusieurs joueurs, le TFT n'est plus adapté. En effet, dans la Figure 4.1, dans le cas du bas, A peut aider B qui peut aider C qui a son tour peut finir le cycle en aidant A. Les joueurs n'observeront pas de coopération directe en retour et par conséquent en vertu du principe du TFT vont arrêter toute coopération. Le TFT classique n'est donc en l'état pas adapté à ce genre de situations.

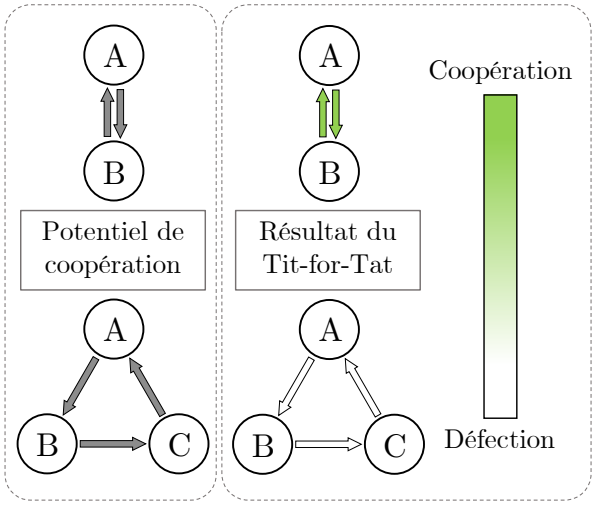


FIGURE 4.1 – En situation circulaire, sans réponse directe, le TFT classique ne converge pas.

4.2 Formalisme du dilemme du prisonnier basé sur des graphes

Pour pallier la spécificité de coopérations potentielles maximales non réciproques, nous introduisons un nouveau formalisme pour généraliser les dilemmes sociaux à N joueurs. L'idée principale de cette extension est que la coopération maximale possible pour chaque couple de joueurs est donnée par un graphe orienté et pondéré où les nœuds représentent les joueurs qui sont reliés par un arc valué. Ainsi, on peut envisager qu'une coopération puisse n'être possible que dans une direction donnée et le jeu peut se voir conférer une structure de coopération avec des cycles.

Nous commençons par détailler dans la section 4.2.1 les différents formalismes de dilemmes du prisonnier à N joueurs qui sont proposés dans la littérature en détaillant brièvement en quoi ils diffèrent de notre approche. Nous y présentons également quelques approches utilisées pour adresser ces jeux. Nous détaillerons ensuite plus précisément notre formalisme : d'abord le modèle que nous avons choisi pour modéliser le Dilemme du Prisonnier (DP) simple à N joueurs puis nous introduirons notre dilemme du prisonnier itéré à structure de graphe aussi appelé *Graph-based Iterated Prisoner's Dilemma* (GIPD).

4.2.1 Comparaison des modèles et algorithmes existant

Les dilemmes sociaux et en particulier le dilemme du prisonnier (DP) ont fait l'objet d'études au cours des précédentes décennies. Ces dilemmes tirent leurs origines des jeux non-coopératifs développés par John F. Nash [Nas51]. Le DP a été suggéré par Albert W. Tucker et formalisé par Merrill Flood [Flo58]. Sa version itérée (le IPD) a été étudiée à l'occasion notamment des tournois d'Axelrod [AH81] en version discrète. L'algorithme sortant victorieux de ce concours est la stratégie du Tit-for-Tat (TFT) [RCO65] : une stratégie simple qui consiste à commencer à coopérer puis à reproduire le choix de l'adversaire. Par la suite, de nombreuses propositions et extensions ont été apportées, à base de TFT bruité [WA95], ou bien un TFT à mémoire rancunière [BDM+01]. D'autres stratégies similaires au TFT ont été également introduites *e.g. win-stay and lose-shift* [NS93]. Des politiques d'apprentissage par renforcement (RL) ont également été utilisées pour entraîner des agents à faire face aux IPD. Les travaux de [MLLFP12] se concentrent sur un algorithme de *Q-learning* multi-agents décentralisé dans des jeux coopératifs et non-coopératifs. Les travaux l'étendent ensuite à un *Q-learning hysteretic* pour résoudre les problèmes de non-stationnarité des agents [MLLFP07].

En ce qui concerne les modèles du IPD à N joueurs, dans le modèle de *N-Iterated Prisoner Dilemma* (NIPD) de [Ham73], les gains des joueurs sont définis selon le nombre de coopérateurs dans le jeu sans aucune distinction de coopération entre les paires de joueurs. Des études ont été

conduites sur les modèles de NIPD [Yao96], et en particulier des expériences avec des politiques de RL [AF11].

Certains modèles NIPD proposent, comme notre formalisme, une approche basée sur des graphes [Ash07, LCS10]. Leurs modèles simulent des situations de joueurs placés dans un graphe non-orienté dans lequel ils peuvent jouer un Dilemme du Prisonnier (DP) uniquement s'ils sont voisins, *i.e.* s'ils sont reliés par une arête. Notre formalisme diffère de cette approche sur deux principaux points : le premier est que le graphe régissant notre jeu est orienté : la coopération possible n'est pas nécessairement réciproque. Le deuxième aspect qui se distingue est la pondération des arcs du graphe qui permet de quantifier une coopération maximale qui est graduelle et non binaire. Ces différences permettent d'abord de prendre en compte des coopérations non réciproques circulaires, et deuxièmement de modéliser des coopérations moins optimales de degré inférieur.

Pour finir, certaines formes de DP existent également sous forme non-symétrique mais au sens des gains et non au sens de la non-réciprocité comme nous l'entendons avec notre formalisme. Ainsi, dans ces dilemmes du prisonnier, les gains des joueurs sont différents même en cas de choix commun. Des études ont été menées dans ces dilemmes [SGJ73, BHSMR07, Daw10] ainsi qu'en version itérée [ALRW07]. Un cas particulier de ce genre de jeux est le jeu de l'Alibi [RG04] qui est un dilemme du prisonnier dans lequel un des partenaires gagne davantage que l'autre.

4.2.2 Dilemme du prisonnier à N joueurs

Dans cette section, nous définissons le modèle choisi pour formaliser le jeu du dilemme du prisonnier à N joueurs dans un premier temps sans structure de graphe. Nous considérons un modèle à N joueurs dans lequel chaque paire de joueurs joue un DP continu à deux joueurs indépendemment des autres paires de joueurs. Chaque joueur $i \in [\![1, N]\!]$ choisit un degré de coopération $c_{ij} \in [0, 1]$ vers chacun des autres joueurs j pour ainsi définir un niveau continu de coopération (entre 0 pour la défection totale et 1 pour la coopération complète). Plus formellement, chaque joueur i choisit un vecteur de coopération $\overrightarrow{C_i} = (c_{ij})_{j \in [\![1,N]\!]}$. L'ensemble des décisions des joueurs définit alors la matrice $C = (c_{ij}) \in [0, 1]^{N \times N}$. Notons que l'on considère qu'un joueur ne coopère pas avec lui-même, les coefficients diagonaux de cette matrice ne sont donc pas utiles et on les définit nuls pas convention ($\forall i, c_{ii} = 0$). Une fois que les degrés continus de coopération ont été choisis par les N joueurs, le tour de jeu peut avoir lieu. Selon notre modèle, les $N(N-1)/2$ DP continus sont joués simultanément au sein de chaque paire de joueurs (i, j). Les gains de chaque DP sont calculés selon l'interpolation vue en définition 1.12.

Nous résumons formellement notre modèle pour le dilemme du prisonnier à N joueurs :

Dilemme du prisonnier continu à N joueurs

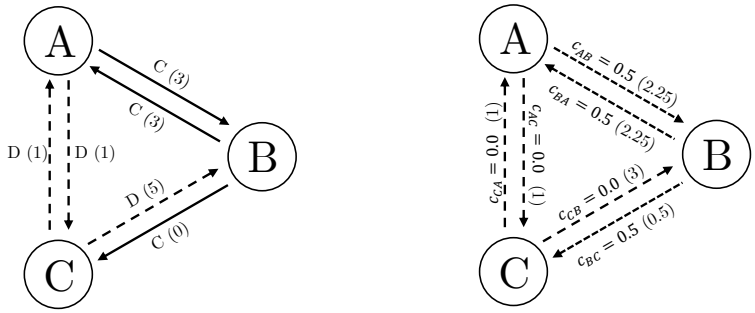
Définition 4.1 *: Soit N joueurs, ils jouent les degrés formant la matrice $C = (c_{ij}) \in [0,1]^{N\times N}$. Chaque joueur i reçoit une récompense V_i définie par :*

$$V_i = \sum_{j\neq i} c_{ij}c_{ji}R + (1-c_{ij})(1-c_{ji})P + c_{ij}(1-c_{ji})S + (1-c_{ij})c_{ji}T \tag{4.1}$$

Comme vu dans les chapitres 1 et 3, les valeurs de (S, P, R, T) généralement utilisées dans la littérature sont $(0, 1, 3, 5)$.

Exemples

Introduisons un exemple de dilemme du prisonnier à 3 joueurs selon notre formalisme avec un graphe complet à 3 nœuds. Dans la Figure 4.2a, les joueurs A, B et C jouent respectivement les choix $\{C, D\}$, $\{C, C\}$ et $\{D, D\}$. Ils gagnent les gains respectifs : $3 + 1$, $3 + 0$ et $1 + 5$. Dans notre version continue du NIPD, sur la Figure 4.2b, les joueurs A, B et C jouent respectivement les vecteurs $[0.0, 0.5, 0.0]$, $[0.5, 0.0, 0.5]$ et $[0.0, 0.0, 0.0]$ pour obtenir les gains définis suivant : $2.25 + 1.0$, $2.25 + 0.5$ et $1.0 + 3.0$.



(a) Exemple de situation discrète (b) Exemple de situation continue

FIGURE 4.2 – Exemples de dilemmes du prisonnier à 3 joueurs où une distinction des coopérations entre les joueurs est faite. La figure de gauche montre une situation discrète (C pour une totale coopération en trait plein, D pour une totale défection en pointillés). La figure de droite montre une situation continue où les joueurs choisissent un degré de coopération orienté $c_{ij} \in [0, 1]$. Les arcs sont plus ou moins pointillés selon le degré de coopération. Dans les deux figures, les gains obtenus de chaque DP sont entre parenthèses.

4.2.3 Dilemme du prisonnier multi-joueurs basé sur des graphes

Nous introduisons notre formalisme de dilemme du prisonnier multi-joueurs basé sur des graphes ou *Graph-based Iterated Prisoner's Dilemma* (GIPD) [LGMLR22] que nous définissons par :

Dilemme du prisonnier multi-joueurs basé sur des graphes

Définition 4.2 *: Un Graph-based Iterated Prisoner's Dilemma (GIPD) est défini par :*

- *un nombre N de joueurs*
- *un graphe orienté et pondéré $\mathcal{G}_{max}$ défini par la matrice d'adjacence pondérée $\mathcal{C}_{max} \in [0,1]^{N\times N}$*
- *un vecteur $D_{max} \in (\mathbb{R}^+)^N$ de débit de coopération maximal*
- *un nombre d'étapes T_{max}*

Dans un GIPD, à chaque étape t, chacun agent i choisit $N-1$ degrés de coopération c_{ij}^t destinés à quantifier la coopération qu'il dédit à chaque agent j. Ces degrés de coopération comportent deux contraintes définies par le jeu. La première est que la valeur maximale pour chaque degré c_{ij}^t est fixée par le graphe orienté pondéré $\mathcal{G}_{max}$; la deuxième contrainte est que l'ensemble du flux sortant (*i.e.* la somme des degrés sortant) de chaque agent i est limité par une valeur $D_{max}[i] : C_i^{t+} = \sum_{j\neq i} c_{ij}^t \leq D_{max}[i]$. En pratique, pour s'assurer que les contraintes sont vérifiées, il convient d'écrêter la matrice $C^t = (c_{ij}^t)$ des $N(N-1)$ choix des N agents par la matrice d'adjacence du graphe de coopération maximale $\mathcal{G}_{max}$: $C^t \leftarrow \min(C^t, \mathcal{C}_{max})$. Cela signifie, par exemple, que si $\mathcal{C}_{max}[a,b] = 0$, le joueur a ne peut coopérer avec b (dans la direction a "aide" b) donc même si a choisit un degré $c_{ab}^t = 1$, le degré effectif sera en réalité égal à 0. En ce qui concerne la deuxième contrainte, elle est assurée en normalisant chaque vecteur colonne $\overrightarrow{C_i^t}$ par le facteur $\min(1, \frac{D_{max}[i]}{C_i^{t+}})$. Une fois que la matrice effective $C^t = (c_{ij}^t)$ est calculée, les joueurs reçoivent les gains déterminés par la formule 4.1. Dans ce jeu, on suppose qu'à chaque étape t, les joueurs ont une observation totale de la matrice des choix de tous les autres joueurs pour toutes les étapes précédentes $\{C^{t'}, t' \in [\![1, t-1]\!]\}$. Nous supposons également qu'ils disposent de la connaissance des potentialités de coopération $\mathcal{C}_{max}$ et D_{max}. En revanche, l'horizon T_{max} leur est inconnu. Notons que ce formalisme de dilemme social permet en particulier de modéliser le cas classique du dilemme du prisonnier à deux joueurs (Figure 4.3a) ainsi que ce que nous appelons les *dilemmes circulaires* c'est-à-dire des dilemmes impliquant $N > 2$ joueurs et pour lesquels le graphe orienté pondéré de la coopération maximale $\mathcal{G}_{max}$ contient un flot cyclique (par exemple les Figures 4.4c ou 4.4d).

4.2.4 Exemples de GIPD

Pour une meilleure compréhension et intuition de notre formalisme, nous allons détailler quelques exemples. Les graphes de la Figure 4.3 représentent quatre exemples de GIPD. Les matrices d'adjacence des graphes de potentiel maximal de coopération sont respectivement les matrices suivantes :

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 1 & 0.5 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0.5 & 0 & 0 & 1 \\ 1 & 0.5 & 0 & 0 \end{bmatrix}$$

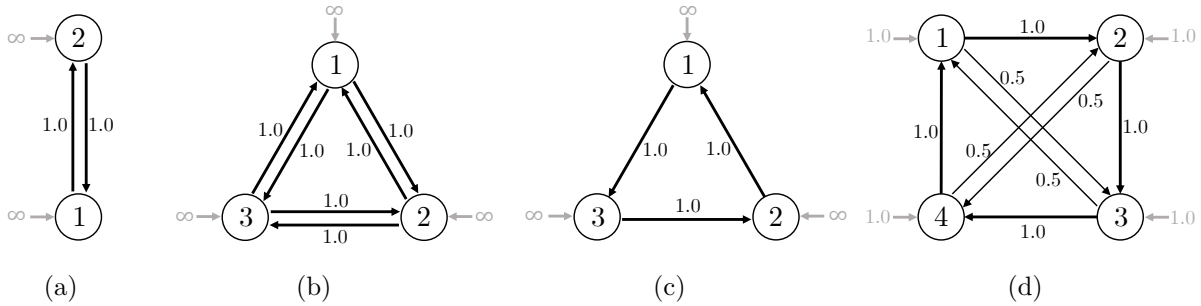
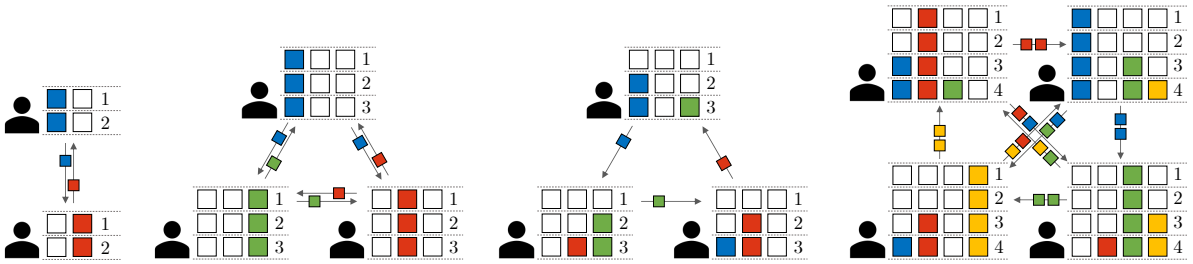


FIGURE 4.3 – Exemples de dilemmes du prisonnier à base de graphe (GIPD) : (4.3a) représente le dilemme du prisonnier simple à deux joueurs. (4.3b) est une version complète et homogène du problème à 3 joueurs. (4.3c) est un cas circulaire à 3 joueurs et (4.3d) est un cas semi-circulaire à 4 joueurs

La matrice A représente un dilemme du prisonnier classique à deux joueurs (Figure 4.3a), B et C représentent des jeux à trois joueurs, l'un réciproque et l'autre circulaire. Enfin D est la matrice d'un graphe semi-circulaire dans lequel il existe un cycle optimal à quatre joueurs et quatre cycles moins optimaux à trois joueurs où le flot de coopération maximal possible est plus faible. Les débits de coopération ($D_{max}[k]$) sont indiqués en gris en entrée des nœuds des graphes. Ce paramètre peut être vu comme la capacité à coopérer. Par exemple, en terme de temps, d'énergie, de ressources, etc. Cela modélise le fait qu'il n'est pas nécessairement possible de rendre service autant que l'on souhaite. Ainsi, pour les trois premiers jeux, on peut fixer ce débit à l'infini (il est de toute manière limité au flot maximal défini par les arcs). Pour le cas semi-circulaire, il peut être intéressant de plafonner ce paramètre à 1.0 pour observer le fait de devoir choisir le cycle le plus pro-social. Ces exemples de GIPD peuvent être reliés aux cas concrets de la Figure 4.4 qui représente des jeux d'échange de ressources. Dans ce problème, N joueurs peuvent se partager K items dont l'utilité marginale est décroissante : $N - k + 1$ pour le $k^{\text{ième}}$ item de la même catégorie. Nous discuterons de ce jeu plus en détails dans le chapitre 5.



(a) (b) (c) (d)

FIGURE 4.4 – Exemple de jeux concrets présentant des patterns de coopération réciproque ou non, homogènes ou non, ou bien circulaire. Il s'agit d'un jeu dans lequel N joueurs ont la possibilité de coopérer avec (ou aider) un autre joueur en donnant des items ou bien les garder. Les items ont une utilité marginale décroissante (*e.g.* $N-k+1$ pour le $k^{ème}$ même item) au motif qu'il est inutile de posséder plusieurs items d'une même catégorie. Cette topologie d'utilité confère alors au jeu une propriété de dilemme. Les exemples de jeux à 2, 3 ou 4, présentés ici, reprennent les patterns de coopération associés aux GIPD présentés dans la Figure 4.3

4.3 Une extension de la politique de Tit-for-Tat classique

Dans cette section, nous présentons notre proposition d'extension de la stratégie Tit-for-Tat qui a pour objectif d'adresser le dilemme du prisonnier multi-joueur à base de graphe que nous avons introduit dans la section 4.2.3. En effet, nous montrerons dans la suite que le Tit-for-Tat classique n'est pas suffisant pour les problèmes asymétriques et en particulier circulaires. Le principe de notre algorithme est de considérer la coopération entre les joueurs comme un flot [FJ56] que l'on injecte dans un graphe. Ensuite, on cherche à déterminer le meilleur chemin de coopération à l'aide d'algorithmes de recherche de flot maximal. La quantité "injectée" ainsi que le graphe sont modifiés par des fonctions de Tit-for-Tat selon le choix des joueurs.

4.3.1 Rappels sur le Tit-for-Tat et notions de réseaux de flots

Avant d'exposer en détails l'architecture de notre approche, rappelons les caractéristiques de notre TFT continu (Chapitre 3) ainsi que quelques notions de théorie des graphes nécessaires pour la compréhension du modèle.

Rappels sur le TFT continu

Nous pouvons résumer les contributions existantes (en particulier celles apportées dans le chapitre 3) sur le Tit-for-Tat continu par cette formulation :

$$\mathrm{TFT}_{\alpha,\beta,\gamma,r_0,c_0}(t,a_{t-1},b_{t-1}),r_t = \begin{cases} c_0, r_0 & \text{if } t=0 \\ \alpha a_{t-1} + (1-\alpha)(r_t + (1-r_t)b_{t-1}), & \\ [r_{t-1} + \beta(b_{t-1} - a_{t-1})]^+ + r_0 \mathcal{B}(1,\gamma)\mathbb{1}_{r_{t-1}=0} & \text{if } t>0. \end{cases} \tag{4.2}$$

Dans cette formulation, $\mathrm{TFT}(t, a_{t-1}, b_{t-1})$ permet de calculer un degré de coopération idéal a_t qu'un joueur A devrait choisir pour une réponse incitative et sûre à un partenaire B en tenant compte des précédents degrés (a_{t-1} et b_{t-1}). Cette fonction comporte un paramètre d'inertie α qui permet de lisser les comportements et les réactions. Elle utilise un coefficient d'incitation dynamique $r_t \in [0,1]$ qui est modifié grâce à un coefficient adaptatif β selon la réponse positive ou négative du partenaire. Enfin, une variable de Bernoulli de paramètre γ permet de réinitialiser avec une probabilité γ le taux d'incitation r_t à r_0 dans le cas il atteint la valeur 0. Ce dernier paramètre permet de pallier une défection mutuelle durable entre les agents.

Notions de réseaux de flots

Nous introduisons ici quelques notions de théorie des graphes, en particulier la théorie des réseaux de flots [FJ56]. Un réseau de flot est un type de graphe orienté dont les arêtes pondérées définissent le flot maximal qui peut circuler d'un sommet à un autre (Figure 4.7). Un réseau de flot contient également une source et un puits. L'objectif courant au sein d'un réseau de flot est de déterminer le flot maximal permettant de faire circuler, de la source au puits, un maximum de quantité de flot sous les contraintes de capacité.

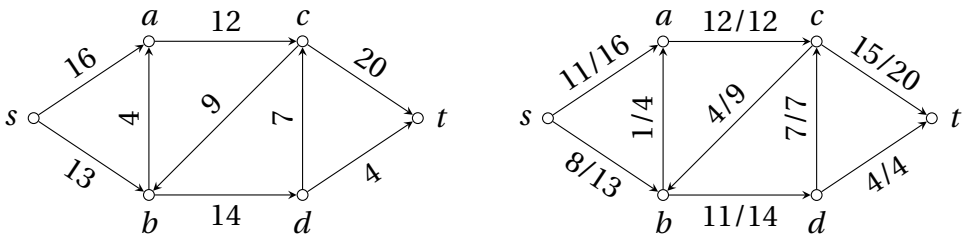
Réseau de flot

Définition 4.3 *: Un réseau de flot est un graphe orienté $G = (S, A, c, s, t)$ avec :*

- *Un ensemble de nœuds S*
- *Un ensemble d'arcs A*
- *Une fonction de capacité $c : A \to \mathbb{N}$ avec la convention $c(u,v) = 0.0$ si $(u,v) \notin A$*
- *Un nœud source $s \in S$ sans arcs entrants*
- *Un nœud puits $t \in S$ sans arcs sortants*

Il est judicieux de supposer qu'il n'existe pas d'arcs anti-parallèles, *i.e.* (v,u) ne peut exister si (u,v) existe. Cependant, dans notre cas appliqué à la coopération potentielle maximale, cette

hypothèse n'est pas nécessaire et survient même de manière courante.



(a) Réseau de flot (b) Flot maximal du réseau

FIGURE 4.5 – Exemple d'un réseau de flot à gauche, et d'un de ses flots maximaux à droite[1]

Pour déterminer le réseau de flot maximal au sein d'un réseau de flot, il existe plusieurs algorithmes tels que :

— L'algorithme de Ford-Fulkerson [FF56] : cet algorithme permet de trouver le flot maximal en temps polynomial (mais de plus court chemin)

— L'approche *Min-cost max flow* [Orl97] : une variante de Ford-Fulkerson avec un critère de minimisation d'un certain coût par arc parcouru. Pour le coût, nous choisissons l'opposé de la coopération empruntée de sorte que le flot choisi privilégie des chemins plus pro-sociaux.

4.3.2 Principe et architecture de notre extension (GTFT)

Le TFT classique repose essentiellement sur un coefficient de coopération qu'un agent garde en mémoire et qu'il modifie dans l'objectif d'inciter la coopération d'un partenaire tout en restant robuste face à la défection. Le principe de notre extension *Graph-based Tit-for-Tat* (GTFT) repose sur deux éléments : d'abord une notion de "quantité" ou de débit de coopération qu'un agent est disposé à délivrer dans le jeu. Deuxièmement, la création d'un réseau de flot (de coopération) dont les capacités sont modifiées selon les comportements des joueurs. Dans ce réseau, on "injecte" le débit de coopération via la source pour ensuite y chercher le flot maximal qui revient indirectement. Notre approche est résumée dans l'algorithme 4.1 et illustrée avec un exemple dans la Figure 4.6. Nous détaillons ci-dessous les différentes étapes de notre approche. Pour commencer, un agent d'indice k dispose de manière interne d'un débit de coopération (un scalaire $\mathcal{D}_k^t$) que l'on va faire varier au cours des étapes ainsi que des degrés de coopération envers chacun des joueurs. Ces degrés sont regroupés dans un vecteur interne de coopération $\overrightarrow{\mathcal{C}_k^t}$ (Notons que la lettre $\mathcal{C}$ est ronde et est à différencier de C). Ces variables sont initialisées selon l'initialisation de f_{TFT} (voir section 4.3.1). Avant le début du jeu (ou si le graphe de coopération maximal change), on calcule le flot maximal F_{max} dans $\mathcal{G}_{max}$ de k vers k (en suivant

1. Figures extraites de `http://igm.univ-mlv.fr/~alabarre/teaching/graphes/chap06-flots.pdf`

le principe expliqué dans l'étape 3 indiqué plus bas). Cela permet de connaître ce que l'on est en mesure d'attendre comme retour de coopération. On établit alors la constante de normalisation $D = \min(D_{max}[k], F_{max})$ qui permettra d'avoir des valeurs dans $[0, 1]$ pour les briques de TFT utilisées dans l'algorithme.

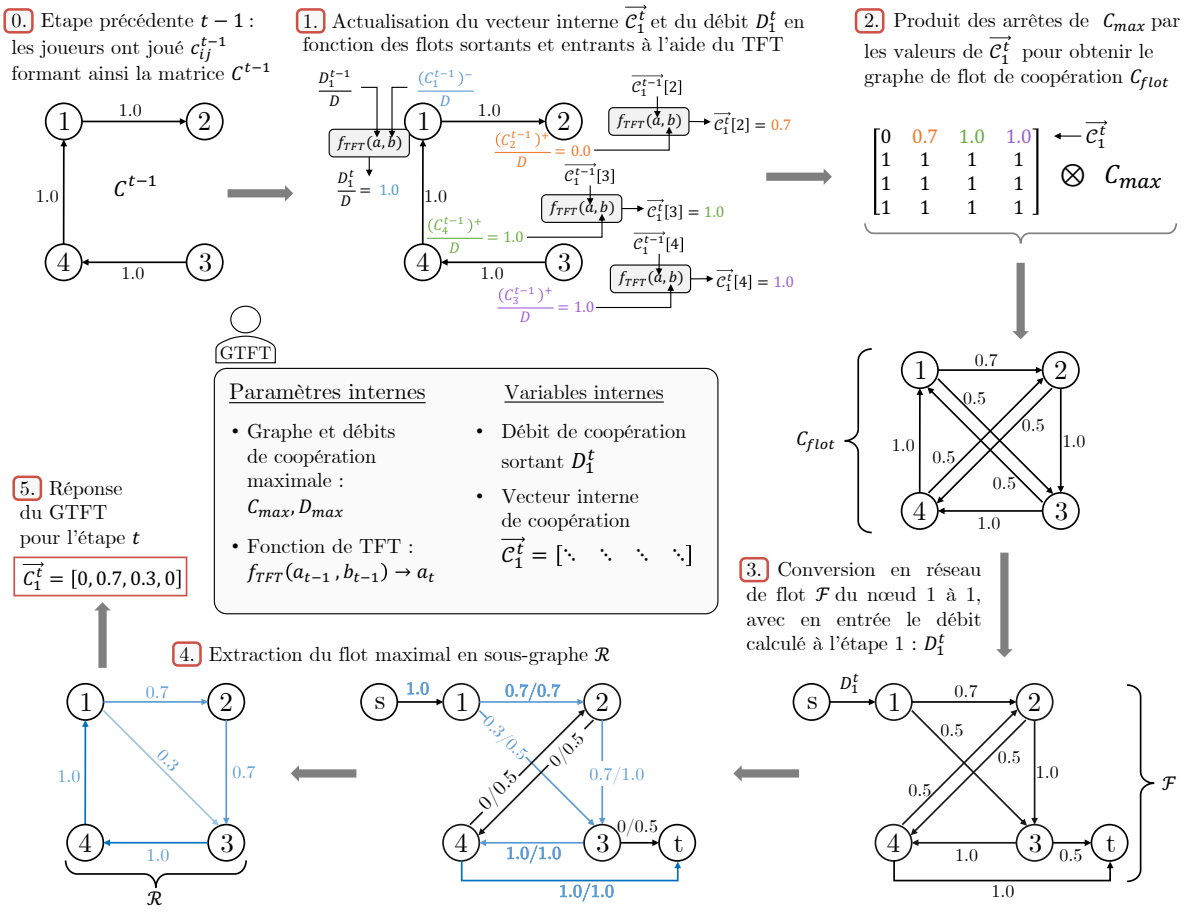


FIGURE 4.6 – Principe de l'architecture de notre approche GTFT sur l'exemple de la Figure 4.3d avec le joueur 2 qui défecte subitement à l'étape $t-1$

Supposons maintenant que le jeu est à l'étape t. A l'étape précédente, les joueurs ont joué les degrés de coopération $c_{i,j}^{t-1} \in [0, 1]$ formant ainsi la matrice (ou le graphe associé) C^{t-1} qui est observable par tous les joueurs. Pour décider des degrés à choisir pour l'étape t, l'agent GRAPHTFT(k) effectue les étapes suivantes (voir l'algorithme 4.1 et la Figure 4.6) :

1. (a) Pour chaque autre joueur j, on actualise la valeur du vecteur interne $\overrightarrow{C_k^t}[j]$ avec la fonction f_{TFT} selon si le flux sortant de j dans C^{t-1} (*i.e.* sa coopération/aide vers les autres) est proche du flot théorique maximal attendu D. En d'autres termes, on

vérifie si le joueur a eu un comportement coopératif ou non.

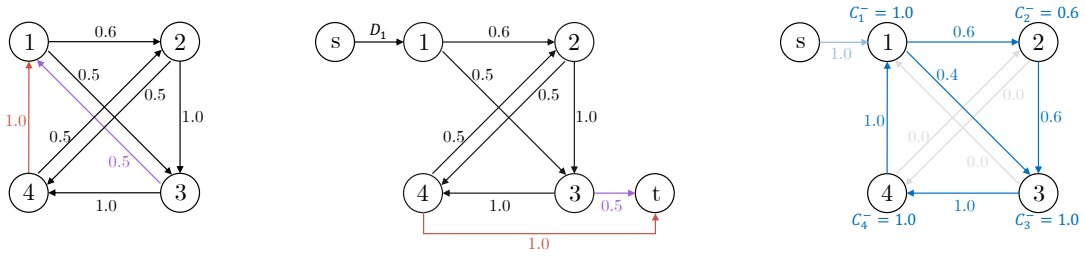
(b) Pour le nœud personnel k, on calcule le flux entrant (*i.e.* à quel point on a reçu de l'aide des autres) et on le compare par TFT avec le débit de coopération que l'on avait injecté à l'étape précédente ($\mathcal{D}_k^{t-1}$). Il en ressort donc le nouveau débit que l'on est prêt à injecter.

2. On crée le graphe C_{flow} dans lequel on va chercher le flot maximal : il s'agit du graphe $\mathcal{G}_{max}$ dans lequel on multiplie chaque arc (k, j) par la valeur $\overrightarrow{C_k^t}[j]$.

3. On convertit le graphe C_{flow} en un réseau de flot $\mathcal{F}$, dont les capacités sont données par C_{flow} où l'on dirige tous les arcs entrants de k vers un nœud puits artificiel t et un nœud source s dirigé vers le nœud k et relié par un arc de capacité $\mathcal{D}_k^t$. Ainsi, nous avons un réseau de flot permettant de trouver le cycle de coopération maximale (*i.e.* de k à k) en supposant que l'on est prêt à "injecter" dans le jeu une coopération de débit égal à $\mathcal{D}_k^t$ (voir Figure 4.7).

4. A l'aide d'un des algorithmes dédiés (par exemple Ford-Fulkerson), le flot maximal $\mathcal{R}$ est extrait du réseau de flot $\mathcal{F}$, il s'agit d'un sous-graphe de $\mathcal{F}$.

5. On déduit du flot maximal $\mathcal{R}$, le choix de coopération à appliquer pour les autres joueurs $\overrightarrow{C_k^t} \leftarrow \mathcal{R}[k, :]$ (par convention $c_{kk} = 0$).



(a) Étape 1 : Tit-for-Tat

(b) Étape 2 : Transformation du graphe 4.7a en un réseau de flot

(c) Étape 3 : Recherche du flot maximal

FIGURE 4.7 – Étapes pour trouver un sous-graphe cyclique de flot maximal partant et revenant au joueur 1 dans le graphe (4.7a). Pour commencer, le graphe est transformé en réseau de flot du point de vue du joueur 1 avec une source s et un puits t (étape (4.7b)). Le flot maximal est alors extrait (étape 4.7c) à l'aide d'un algorithme dédié.

Algorithme 4.1 : GRAPHTFT (pour l'agent d'indice k)

Entrées : Graphe de coopération maximale $\mathcal{C}_{max}$, potentiel maximal de coopération D_{max}

Paramètres : Une fonction de TFT f_{TFT}

Initialisation de $\overrightarrow{\mathcal{C}_k^0}$ et $\mathcal{D}_k^0$ avec $_{TFT}(t=0)$

Calculer le flot maximal F_{max} pour établir la constante de normalisation $D = \min(D_{max}[k], F_{max})$

for $t \in [1, T_{max}]$ **do**

  $C_{flow} \leftarrow \mathcal{C}_{max}$

  for *tout autre agent* j **do**

    Avec C^{t-1}, générer le flot sortant de coopération de j : $(C_j^{t-1})^+$

    Exécuter un TFT sur j : $\overrightarrow{\mathcal{C}_k^t}[j] = f_{TFT}(\overrightarrow{\mathcal{C}_k^{t-1}}[j], \frac{(C_j^{t-1})^+}{D})$

    Modifier le graphe de flot $C_{flow}[k,j] \leftarrow \overrightarrow{\mathcal{C}_k^t}[j] C_{max}[k,j]$

  end

  Avec C^{t-1}, calculer le flot entrant de coopération pour k : $(C_k^{t-1})^-$

  Actualiser par TFT le débit sortant de coopération : $\mathcal{D}_k^t \leftarrow D \times f_{TFT}(\frac{\mathcal{D}_k^{t-1}}{D}, \frac{(C_k^{t-1})^-}{D})$

  Générer un nouveau réseau de flot $\mathcal{F}$ de la source k au puits k avec une source entrante de capacité $\mathcal{D}_k^t$ et des capacités données par C_{flow} (voir Figure 4.7)

  Avec $\mathcal{F}$, extraire le sous-graphe $\mathcal{R}$ de flot maximal (de coopération)

  Choisir les degrés de coopération depuis le flot maximal calculé : $\overrightarrow{C_k^t} \leftarrow \mathcal{R}[k,:]$

end

4.4 Méthode comparative des algorithmes de Tit-for-Tat

Nous allons dans cette section, évaluer la pertinence, les différentes composantes et les paramètres de notre algorithme GRAPHTFT. Dans un premier temps, nous allons montrer l'intérêt de la structure de graphe par rapport au Tit-for-Tat classique proposé dans la littérature. Nous étudierons également l'impact du choix de l'algorithme de graphe utilisé ainsi que les paramètres de la fonction de Tit-for-Tat utilisée.

Pour ce faire, nous introduisons d'abord des métriques permettant d'évaluer le comportement des agents (section 4.4.1). Ensuite nous allons détailler quelques tournois que nous choisissons pour l'évaluation.

4.4.1 Métriques utilisées

Les métriques introduites dans le chapitre 3 sont étendues ici à N joueurs. Ainsi, nous allons définir pour un jeu multi-joueurs les métriques d'efficacité, de vitesse, d'incitation (*Incentive-*

compatibility), de sûreté (*Safety*) et d'indulgence (*Forgiveness*).

Pour commencer, introduisons quelques notations. Nous désignons un tournoi par le quadruplet $\mathcal{T} = (N, \mathcal{C}_{max}, D_{max}, T_{max})$, défini par conséquent par un GIPD à N joueurs de T_{max} étapes. Le jeu est défini par le graphe maximal donné par la matrice d'adjacence $\mathcal{C}_{max}$ et le débit de coopération D_{max}. Par ailleurs, si l'on désigne par $\vec{\pi} = (\pi_i)$ les N agents (ou politiques) choisies par chacun des N joueurs dans le tournoi $\mathcal{T}$, on note $V_i(\mathcal{T}, \vec{\pi})(t)$ les T_{max} gains reçus par le joueur i et $\overline{V_i(\mathcal{T}, \vec{\pi})}$ la somme de ces T_{max} gains.

Bien commun

Aussi appelée *Social welfare* et notée SW, cette métrique désigne la somme des gains des joueurs à une étape donnée t :

$$SW(\mathcal{T}, \vec{\pi}, t) = \sum_{i=0}^{N-1} V_i(\mathcal{T}, \vec{\pi})(t) \tag{4.3}$$

Efficacité

L'efficacité a pour objectif de mesurer à quel point le bien commun final est proche de l'optimal. On définit alors l'efficacité d'une politique π par le bien commun calculé quand tous les joueurs choisissent cette politique. On normalise la métrique sur $[0, 1]$ (centrée et réduite) :

$$E(\mathcal{T}, \pi, t) = \frac{SW(\mathcal{T}, \pi\vec{1}, t) - SW(\mathcal{T}, D\vec{1}, 0)}{SW(\mathcal{T}, C\vec{1}, 0) - SW(\mathcal{T}, D\vec{1}, 0)}, \quad \text{avec } x\vec{1} = (x)_i \tag{4.4}$$

avec respectivement C et D : la coopération et la défection pures.

L'efficacité est ici une fonction de temps. On note E_{max} la valeur finale sur l'épisode $E_{max} = E(\mathcal{T}, \pi, T_{max} - 1)$. Par abus, la valeur finale de cette métrique au cours du tournoi, notée alors simplement E, sera dans la suite appelée efficacité.

Vitesse

La vitesse mesure la rapidité avec laquelle l'efficacité atteint sa valeur finale. On la définit par le rapport :

$$\begin{aligned} &Sp(\mathcal{T}, \pi) = \frac{1}{\tau E_{max}} \int_0^{\tau} E(\mathcal{T}, \pi, t) dt \\ &\text{avec } \tau \in [1, T_{max} - 1] \end{aligned} \tag{4.5}$$

Incitation à coopérer

L'incitation à coopérer (*Incentive-Compatibility*) mesure à quel point une politique jouée par les joueurs incitent un autre joueur à préférer le choix de la coopération plutôt que celui de la défection. Inspiré de [LP17], nous définissons l'incitation d'une politique π dans le tournoi $\mathcal{T}$ notée $IC(\mathcal{T}, \pi)$ par la différence (normalisée) :

$$\begin{aligned} IC(\mathcal{T}, \pi) &= \frac{1}{I_{norm}} \left[\overline{V_0(\mathcal{T}, (C, \pi, ..., \pi))} - \overline{V_0(\mathcal{T}, (D, \pi, ..., \pi))} \right] \\ \text{avec } I_{norm} &= \overline{V_0(\mathcal{T}, C\vec{1})} - \overline{V_0(\mathcal{T}, D\vec{1})} \end{aligned} \tag{4.6}$$

Cette métrique calcule la différence entre ce qu'un agent (d'indice 0) reçoit s'il coopère face à tous les autres agents appliquant la même politique π comparé à ce qu'il aurait gagné en choisissant la défection.

Sûreté

La sûreté (*safety*) mesure le risque qu'un agent prend en décidant de suivre une certaine politique face à des défecteurs. On définit la sûreté d'une politique π notée $Sf(\mathcal{T}, \pi)$ par la différence :

$$\begin{aligned} Sf(\mathcal{T}, \pi) &= \frac{1}{I_{norm}} \left[\overline{V_0(\mathcal{T}, (\pi, D, ..., D))} - \overline{V_0(\mathcal{T}, (D, D, ..., D))} \right] \\ \text{avec } I_{norm} &= \overline{V_0(\mathcal{T}, (D, D, ..., D))} - \overline{V_0(\mathcal{T}, (C, D, ..., D))} \end{aligned} \tag{4.7}$$

Faisant face à des défecteurs, on mesure la différence de gain entre le choix de la politique π plutôt que la défection comme les autres joueurs. Puisque la défection est dominante (par définition du dilemme), cette métrique est toujours négative ou nulle. Plus elle est proche de zéro, plus la politique est sûre.

Indulgence

Commençons par appeler "défecteur repenti" noté L un joueur qui défecte aux étapes $0 \leq t \leq \tau_0$ puis alors se met à coopérer complètement (degré de coopération égal à 1.0). La métrique *Forgiveness* mesure alors la rapidité avec laquelle L se met à gagner de nouveau des gains (compte-tenu de l'indulgence des autres agents dont on étudie la politique π) :

$$\begin{aligned} Fg(\mathcal{T}, \pi) &= \frac{1}{I_{norm}} \int_{\tau_0}^{T_{max}} \left[V_0(\mathcal{T}, (L, \pi, ..., \pi))(t) - V_0(\mathcal{T}, (L, \pi, ..., \pi)(\tau_0) \right] dt \\ \text{avec } I_{norm} &= \tau(E(\mathcal{T}, \pi, T_{max}) - V_0(\mathcal{T}, (L, \pi, ..., \pi)(\tau_0)) \end{aligned} \tag{4.8}$$

Pour une meilleure intuition des métriques définies plus haut, des simulations simples sont disponibles dans la Figure A.4 en Annexe.

4.4.2 Agents étudiés et leurs caractéristiques

Dans nos simulations, nous comparons plusieurs versions du GTFT, en particulier la version classique du TFT *i.e.* le GTFT sans algorithme de traitement de graphe. Nous étudions deux types de traitement de graphe ainsi que diverses fonctions de TFT avec des paramètres différents (voir 4.3.1) :

1. Le choix de l'algorithme pour le traitement des graphes, *i.e.* la recherche de flot optimal. Les trois cas étudiés sont : aucun (TFT classique)/Ford-Fulkerson/Approche Min-cost
2. le choix de la fonction de TFT intervenant dans GTFT. Nous étudions les trois types de variantes de TFT évoqués dans le chapitre 3 : alpha/beta/gamma

fonction de Tit-for-Tat	Traitement de graphe		
	$\varnothing$	Ford-Fulkerson	Min-cost max flow
TFT_alpha	noGraphTFT$_\alpha$	grTFT_Fulkerson$_\alpha$	**grTFT_MinCost$_\alpha$**
TFT_beta	**noGraphTFT$_\beta$**	**grTFT_Fulkerson$_\beta$**	**grTFT_MinCost$_\beta$**
TFT_gamma	noGraphTFT$_\gamma$	grTFT_Fulkerson$_\gamma$	**grTFT_MinCost$_\gamma$**

TABLE 4.1 – Les agents étudiés lors des simulations sont en gras. Comme cela, on étudie d'abord l'impact du choix de traitement de graphe (en bloquant le TFT à un choix de type Beta). Dans un second temps, on évalue le choix du TFT dans une configuration MinCost.

4.4.3 Tournois de simulation

L'objectif est ici de montrer l'intérêt de l'apport d'une structure de graphe sur le TFT dans certaines situations, *i.e.* des cas où une coopération circulaire existe. Nous adoptons donc des tournois basés sur des graphes orientés et pondérés dans lesquels il existe au moins un flot cyclique.

Pour nos simulations, nous créons deux types de tournois de $N > 2$ joueurs. Le premier type est circulaire, il désigne une situation où la coopération potentielle maximale est sous la forme d'une chaîne cyclique. On le dénote CIRC(N) et le graphe de coopération maximale est défini par sa matrice d'adjacence :

$$\forall i,j \in [\![0, N-1]\!]^2, \quad \mathcal{C}_{max}[i,j] = \begin{cases} 1.0 & \text{si } j = i+1 \mod N \\ 0.0 & \text{sinon.} \end{cases} \tag{4.9}$$

Le deuxième type de jeu que nous considérons est similaire au premier mais avec un arc de coopération alternatif pour "court-circuiter" un joueur défecteur. Ce tournoi est nommé

DoubleCirc(N) et la matrice d'adjacence de son graphe de coopération est définie par :

$$\forall i,j \in [\![0, N-1]\!]^2, \quad \mathcal{C}_{max}[i,j] = \begin{cases} 1.0 & \text{si } j = i+1 \pmod N \text{ ou } j = i+2 \pmod N \\ 0.0 & \text{sinon.} \end{cases} \tag{4.10}$$

Dans la figure 4.8, quelques graphes de tournoi qui sont utilisés dans les simulations sont représentés.

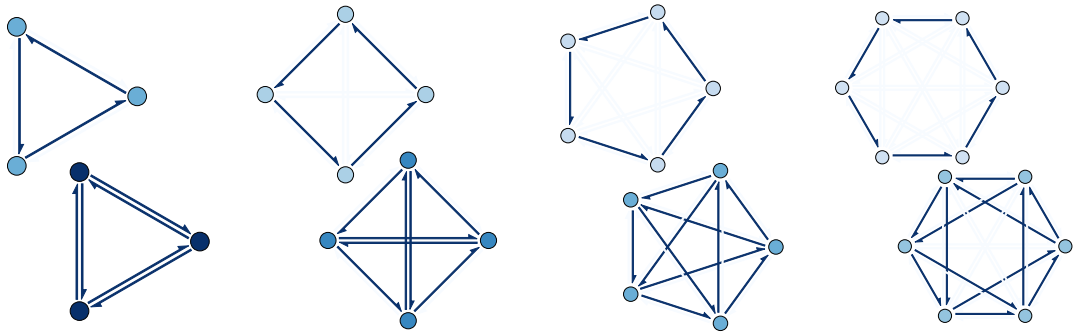


Figure 4.8 – Graphes de coopération maximale des tournois Circ(N) et DoubleCirc(N) pour $N \in \{3,4,5,6\}$. Notons que DoubleCirc(3) est un DP complet.

4.5 Résultats et discussion

Dans cette section, nous présentons les résultats expérimentaux de quelques simulations au cours desquelles nous comparons notre extension du TFT continu adapté pour le GIPD comparé à la version classique. Nous évaluons en particulier la pertinence de la structure de graphe de notre GTFT, puis l'impact du choix de la fonction de TFT utilisée et de ses paramètres.

4.5.1 Impact du choix d'algorithme pour le traitement de graphe

Pour commencer, nous nous concentrons sur la nécessité d'incorporer un algorithme de traitement de graphe au sein de l'agent. Pour étudier cet impact, trois cas sont envisagés :

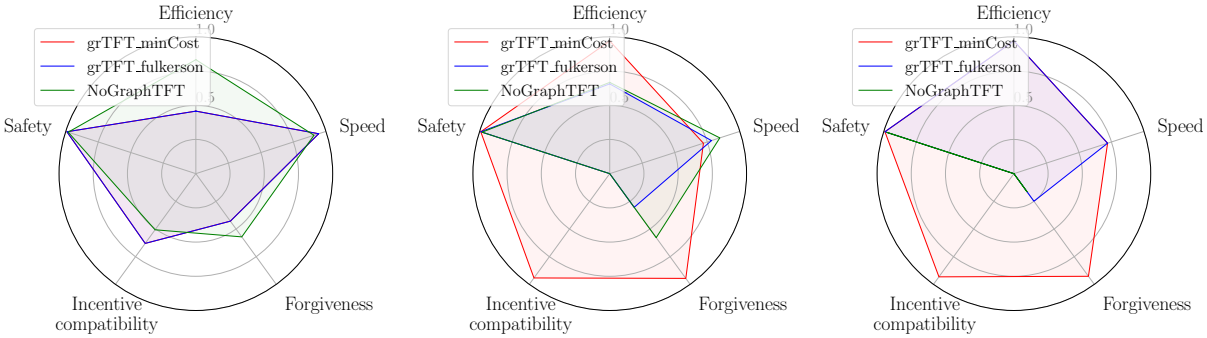
— noGraphTFT : le TFT classique continu sans algorithme de graphe

— grTFT_Fulkerson : notre agent GTFT (algorithme 4.1) avec le choix de l'algorithme de Ford-Fulkerson pour extraire le flot maximal

— grTFT_minCost : notre agent GTFT avec une approche de *min-cost max flow* : cet algorithme recherche le flot maximal mais avec une contrainte de circulation maximale de

coopération/flot (*i.e.* qui, pour plusieurs flots égaux, choisit celui qui propose des chemins plus longs donc profitant au plus de joueurs).

Nous simulons ces trois types d'agent dans les jeux DoubleCirc avec 3, 4 or 6 joueurs et évaluons les comportements avec nos métriques sociales que nous représentons dans la Figure 4.9.



(a) Double Circulaire avec 3 joueurs[1]

(b) Double Circulaire avec 4 joueurs[2]

(c) Double Circulaire avec 6 joueurs[3]

Figure 4.9 – Impact du choix de l'algorithme de traitement de graphe dans les jeux DoubleCirc(N) avec $N \in \{3, 4, 6\}$ joueurs. La fonction de Tit-for-Tat utilisée pour les agents GTFT dans ces simulations est un TFT de type Beta avec les paramètres ($\alpha = 0.8$, $\beta = 0.6$, $r_0 = 0.4$)

Les observations expérimentales confirment l'intuition que l'absence d'algorithme traitant la structure de graphe dans les situations non-réciproques est préjudiciable pour le TFT classique. Par exemple, dans le jeu DoubleCirc(6), le TFT échoue car il n'y a pas de possibilité de coopération réciproque. Sans réponse de la part du récepteur de coopération, l'agent cesse de coopérer et l'ensemble tend vers la défection mutuelle. Un fait intéressant est que dans le jeu DoubleCirc(4), le TFT classique réussit à trouver une solution moins optimale puisqu'il existe deux cycles (triviaux) à deux joueurs. Enfin, le TFT classique peut adresser le DoubleCirc(3) car une solution réciproque alternative est possible bien qu'un peu moins optimale. Concernant le choix de l'algorithme de traitement de graphe, le meilleur au sens de l'efficicaté est celui avec une approche *min cost* parce qu'il permet de sélectionner des cycles plus pro-sociaux sans préjudice personnel (à l'inverse de l'algorithme de Ford-Fulkerson qui choisit le cycle le plus court à flot égal). Il est alors plus approprié compte-tenu des objectifs d'incitation et de robustesse à la défection. Cependant, dès lors qu'il existe plusieurs chemins optimaux et pro-sociaux, les agents GTFT subissent des problèmes de synchronisation et dans certains cas mènent à la défection mutuelle. Nous discutons ces cas dans la section 4.6.

1. Simulation avec le problème de cycle à 3 joueurs : `https://youtu.be/VHhEZ8Wu_XQ` ou avec 5 joueurs : `https://youtu.be/J-weuvOkkBc`
2. Une simulation avec 4 joueurs : `https://youtu.be/s08j24LMr0U`
3. Une simulation avec 6 joueurs : `https://youtu.be/AL29LFbh3n8`

4.5.2 Impact du choix de la fonction de Tit-for-Tat

Notre algorithme GTFT nécessite une fonction dite de négociation qui est un Tit-for-Tat continu. Elle intervient dans l'incitation personnelle ainsi que dans la gestion des arcs dans le graphe de coopération. Nous étudions dans cette section le choix de cette fonction f_{TFT} et en particulier les paramètres qu'elle comporte. Comme rappelé dans la section 4.3.1, nous avons introduit une forme générale de TFT qui peut se décliner en trois versions : TFT_alpha avec une composante d'inertie, TFT_beta avec un paramètre adaptatif d'incitation et TFT_gamma avec un paramètre d'incitation stochastique. Avant d'étudier plus en détails l'impact des paramètres de f_{TFT}, on commence par évaluer brièvement les trois types de fonction avec leurs paramètres $\alpha = 0.8$, $\beta = 0.6$, $r_0 = 0.4$, $\gamma = 0.05$ (quand approprié). Ces trois types sont évalués dans la Table 4.2 (voir aussi la Figure 4.10a).

(métriques $\times 100$, 5 exécutions pour TFT_gamma seulement qui est stochastique)

	Circ(3)[1]				
	E	Sp	IC	Sf	Fg
TFT_alpha	98	51	33	-37	91
TFT_beta	95	51	78	-1.2	33
TFT_gamma	99.0 ± 0.0	50.3 ± 0.9	81.8 ± 1.7	-2.6 ± 0.6	63.1 ± 13.6

	Circ(6)				
	E	Sp	IC	Sf	Fg
TFT_alpha	98	51	33	-37	91
TFT_beta	95	51	78	-1.2	16
TFT_gamma	98.2 ± 1.7	50.2 ± 0.9	79.1 ± 2.6	-2.1 ± 0.7	35.7 ± 4.8

Table 4.2 – Comparaison des trois types de TFT (alpha, beta, gamma). Analyse des paramètres de la fonction de TFT. Si approprié, les paramètres sont $\alpha = 0.8$, $\beta = 0.6$, $r_0 = 0.4$, $\gamma = 0.05$ en utilisant graphTFT avec l'approche *min cost*.

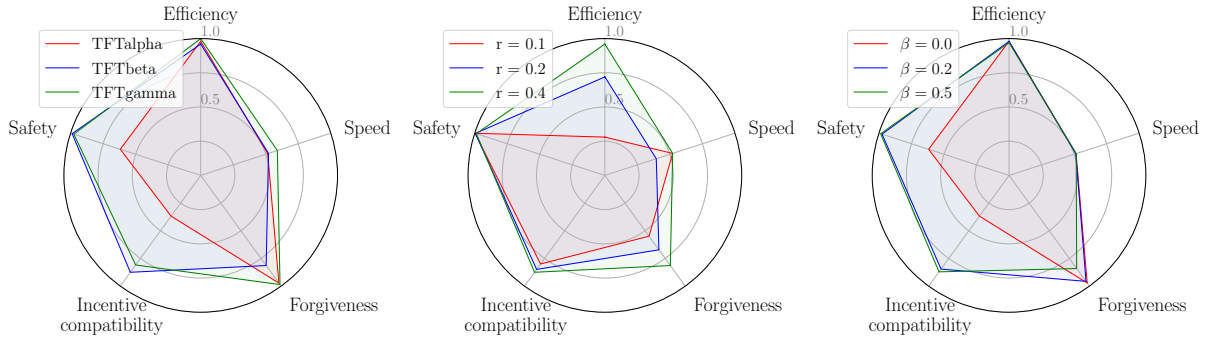
D'après la Table 4.2 (voir aussi la Figure 4.10a), on peut observer que les métriques sont quasiment identiques pour certaines métriques que l'on soit dans le jeu Circ(3) ou Circ(6). On peut observer que TFT_beta et TFT_gamma sont plus sûrs et plus incitatifs que TFT_alpha grâce à l'utilisation du coefficient adaptatif β. TFT_gamma semble être plus performant au sens de l'efficacité globale (*social welfare*). Observons maintenant plus en détails les paramètres r_0 et β du TFT_beta. Nous présentons les résultats sur quelques tournois en Table 4.3, également représentés pour le cas DoubleCirc(6) sur les Figures 4.10b (pour r_0) et 4.10c (pour β).

1. Vidéo de simulation avec 3 joueurs : `https://youtu.be/9nfuQ7dc1xU`
Vidéo de simulation avec 5 joueurs : `https://youtu.be/7BEMfu4bW7U`

(métriques ×100)

	$r_0 = 0.1$				$r_0 = 0.2$				$r_0 = 0.4$			
	E	Sp	IC	Sf	E	Sp	IC	Sf	E	Sp	IC	Sf
Circ(3)	25	52	0	−0.2	66	42	46	−0.5	95	51	78	−1.2
Double(3)	21	61	61	−0.2	44	54	60	−0.5	46	79	60	−1.2
Double(4)	28	52	82	−0.2	72	39.5	87	−0.5	96	52	89	−1.2
Double(6)	28	52	80	−0.2	72	39.5	85	−0.5	96	52	87	−1.2
	$\beta = 0.0$				$\beta = 0.2$				$\beta = 0.5$			
	E	Sp	IC	Sf	E	Sp	IC	Sf	E	Sp	IC	Sf
Circ(3)	98	51	33	−38.0	97	51	79	−2.3	96	51	79	−1.3
Double(3)	63	67	12	−38.2	45	83	58	−2.4	46	80	60	−1.3
Double(4)	98	52	39	−38.2	98	51	87	−2.4	97	52	89	−1.3
Double(6)	98	51	37	−38.0	98	51	85	−2.4	97	52	87	−1.3

Table 4.3 – Impact du choix de la fonction de TFT. Nous évaluons l'impact du taux d'incitation initial r_0 et le coefficient adaptatif β (nul ou non). Si non précisé, les paramètres sont $\alpha = 0.8$, $\beta = 0.6$, $r_0 = 0.4$, $\gamma = 0.05$ avec GRAPHTFT (et une approche *min cost*).



(a) Impact du type de TFT

(b) Impact de l'*incentive* r_0

(c) Impact du coefficient adaptatif β

Figure 4.10 – Impact du choix de la fonction de f_{TFT} sur le jeu DoubleCirc(6) avec GTFT et l'approche *min cost*. Les trois types de TFT sont comparés (Figure 4.10a), puis l'impact du coefficient incitatif initial r_0 (Figure 4.10b) et enfin le coefficient adaptatif β (Figure 4.10c) de l'algo TFT_beta algorithm.

Nous pouvons remarquer qu'un coefficient incitatif initial r_0 plus élevé permet d'atteindre le gain optimal total plus rapidement sans réduire la sûreté et l'incitation. Enfin, concernant l'impact du coefficient adaptatif β, on peut observer dans la Table 4.3 qu'un simple coefficient non-nul est suffisant pour rendre l'agent plus sûr et plus incitatif.

4.5.3 Synthèse des observations

Bien que notre agent GTFT ne soit pas parfaitement optimal pour les situations présentant une ambiguïté de choix du chemin de coopération optimal (avec plusieurs possibilités de cycles), nous pouvons conclure que le point clé des simulations expérimentales est l'importance du traitement à base d'un algorithme de graphe. L'approche *min-cost max flow* est la meilleure des approches compte tenu du gain d'incitation et d'efficacité sans préjudice de la sûreté. Quant au choix de la fonction de TFT, le TFT_beta est clairement plus sûre que TFT_alpha. TFT_gamma est légèrement plus efficace, notamment pour éviter la rancune lors les défections mutuelles temporaires.

4.6 Limitations

Notre formalisme du *Graph-based Iterated Prisoner's Dilemma* (GIPD) est une première proposition d'extension du *N-Iterated Prisoner Dilemma* (NIPD) ayant pour but d'adresser les situations qui ne sont plus nécessairement réciproques. En particulier, lorsqu'un chemin circulaire de coopération existe entre plus de deux joueurs. Ce nouveau formalisme s'accompagne de nombreuses interrogations et de limitations de modélisation que nous discutons dans cette partie. Par ailleurs, nous évoquons également les limites de notre agent *Graph-based Tit-for-Tat* (GTFT) qui étend le TFT classique en incorporant une structure de graphe.

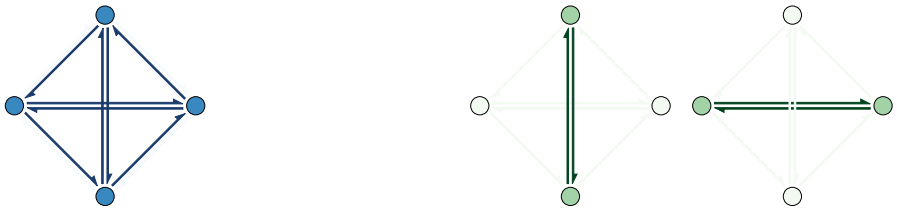
4.6.1 Observation partielle

Dans notre formalisme, nous faisons l'hypothèse forte que chaque joueur dispose d'une connaissance parfaite des choix effectués par chacun des autres joueurs à chaque instant précédent. Une hypothèse plus réaliste pourrait restreindre l'observation à des voisins de coopération (entrant et sortant). La connaissance parfaite des caractéristiques du potentiel de coopération maximale (en particulier du graphe de coopération maximale) peut être également dans certains cas une limitation importante. Des propositions d'estimation de ces informations seront détaillées dans le chapitre 5, notamment en utilisant des fonctions de valeur issues de politiques d'apprentissage par renforcement pré-entraînées.

4.6.2 Ambiguïté du choix commun du cycle

Dans certaines situations présentant plusieurs chemins possibles dans le graphe de coopération maximale et où le potentiel de coopération est limité, les agents peuvent faire face à un soucis de synchronisation quant au choix commun du cycle. Par exemple, la faible efficacité de l'approche Ford-Fulkerson s'explique par le fait que les agents choisissent le plus court chemin qui maximise leur flot retour de coopération mais ne choisissent donc pas le même cycle selon leur position

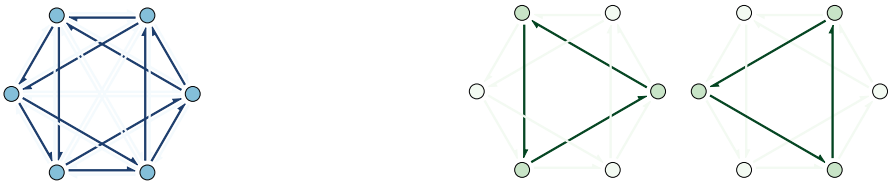
dans le graphe (comme on peut l'observer dans le problème à 4 joueurs (Figure 4.11) ou à 6 joueurs (Figure 4.12).



(a) Graphe de coopération maximale

(b) Deux cycles de flot maximal choisis par l'algorithme de Ford-Fulkerson

FIGURE 4.11 – Ambiguïté du choix de cycles avec quatre joueurs

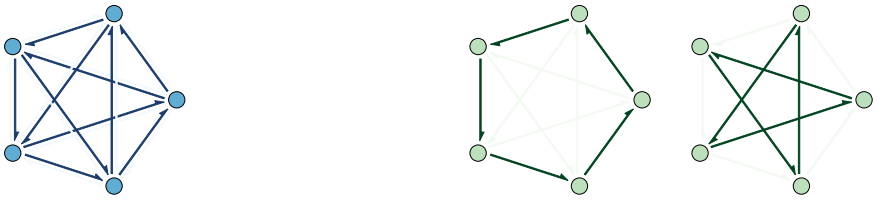


(a) Graphe de coopération maximale

(b) Deux cycles de flot maximal choisis par l'algorithme de Ford-Fulkerson

FIGURE 4.12 – Ambiguïté du choix de cycles avec six joueurs

Cependant, bien que l'approche *min-cost max flow* permet de pallier le problème en choisissant des chemins plus pro-sociaux, elle peut également faire face à des problèmes d'ambiguïté sur le choix du meilleur cycle. En effet, dans certaines situations, il peut exister plusieurs cycles équivalents en terme d'efficacité pro-sociale (exemple de la Figure 4.13) et donc conduire à des problèmes de choix commun, indépendemment de leur volonté de coopérer.



(a) Graphe de coopération maximale

(b) Deux cycles différents mais optimaux choisis par l'approche *min-cost max flow*

FIGURE 4.13 – Ambiguïté du choix des cycles avec cinq joueurs

4.7 Conclusion et perspectives

Nous avons introduit dans ce chapitre un nouveau paradigme de dilemme du prisonnier (DP) continu à N joueurs. Nous avons proposé un modèle dans lequel les coopérations possibles entre les joueurs peuvent être limitées et non nécessairement réciproques. Notre extension propose alors de régir la coopération maximale possible entre les joueurs par un graphe orienté et pondéré. Ce modèle est particulièrement adapté pour adresser les asymétries et les aspects circulaires de coopération : une situation spécifique de dilemme où les joueurs peuvent coopérer le long d'un cycle dans lequel un coopérateur peut aider son successeur mais pas son prédécesseur. L'avantage du modèle à structure de graphe pondéré et orienté est qu'il permet de modéliser ces situations spécifiques tout en pouvant inclure les situations originales (c'est-à-dire le DP à deux joueurs) ou plus classiques (le DP homogène à N joueurs).

Dans un deuxième temps, nous nous sommes penchés sur les fonctions de Tit-for-Tat (TFT) dans divers scénarios avec des structures particulières de coopération. Les résultats montrent que les techniques classiques de TFT échouent dès lors que la réciprocité n'est pas permise. C'est le cas même s'il existe des alternatives de coopération via l'intermédiaire de tiers (avec $N \geq 3$ joueurs). Nous avons alors proposé également une extension de TFT qui permet d'adresser les problèmes de non-réciprocité et de circularité. Notre algorithme *Graph-based Iterated Prisoner's Dilemma* (GIPD) généralise alors le TFT avec une approche de réseau de flot qui modélise un flux de coopération que chaque joueur souhaite recevoir en retour y compris via des tiers.

Nous avons évalué expérimentalement ce nouvel algorithme sur divers scénarios canoniques. Plusieurs métriques sociales ont été implémentées et plusieurs aspects de notre agent ont été évalués en le comparant notamment avec des baselines de TFT classiques. Les principales conclusions expérimentales attestent que l'addition d'un traitement de graphe sur le TFT est pertinente dès lors que la coopération réciproque n'est plus permise. Les résultats montrent que notre approche GTFT est au moins aussi performante que le TFT classique.

Il convient cependant de mentionner des limitations à la fois de notre formalisme de NIPD ainsi que de notre approche GTFT. La principale limite de notre modèle de GIPD est la nécessité d'accéder à l'observation totale du jeu en particulier du graphe de potentiel de coopération. En ce qui concerne notre extension du TFT, il apparaît que dans des situations spécifiques, nos agents GTFT connaissent quelques problèmes d'ambiguïté et de synchronisation dans la recherche de cycles.

Malgré ces limitations, notre extension du modèle du IPD à N joueurs permet de prendre en considération davantage de scénarios. L'extension du TFT offre également de nombreuses pistes

d'amélioration. Notre approche GTFT peut notamment être incorporée dans un algorithme hybride qui combine des stratégies de TFT et des politiques de RL [LP17]. Cette combinaison va permettre d'adresser des dilemmes sociaux plus complexes tels que les *Sequential Social Dilemma* (SSD). Cet ajout permettrait alors d'adresser des situations non-réciproques. Étant convaincus des perspectives et des enjeux concernant ces approches hybrides, nous proposons d'étudier cette extension hybride dans le chapitre 5.

Chapitre 5

Dilemmes sociaux séquentiels asymétriques et méthodes hybrides

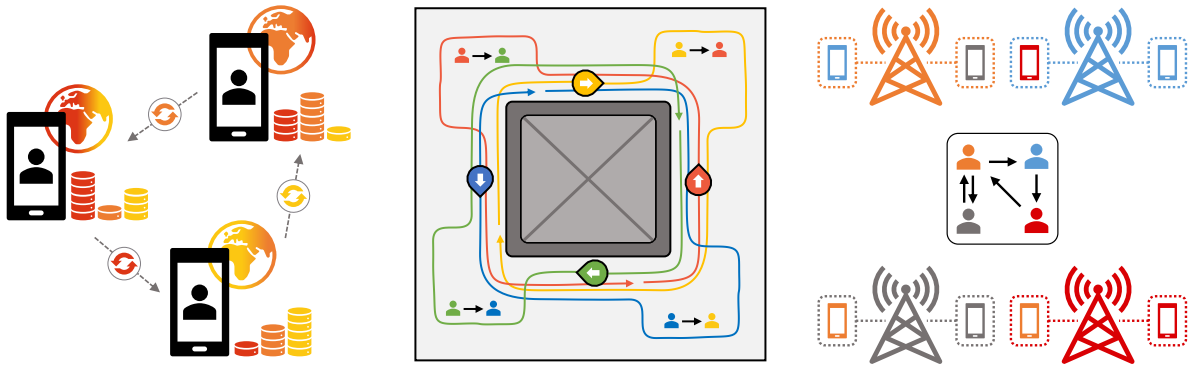
Dans ce chapitre, nous proposons de nous pencher sur un nouveau concept de dilemme social séquentiel ou SSD présenté dans le chapitre 2. Rappelons que les SSD sont une extension du dilemme du prisonnier dans laquelle les actions atomiques (coopération/défection) sont remplacées par des politiques plus complexes de type politiques d'apprentissage par renforcement (RL). L'idée principale est d'étendre le SSD au concept d'asymétrie vu au chapitre 4. En d'autres termes, nous étendons notre concept de *Graph-based Iterated Prisoner's Dilemma* (GIPD) au même titre que le SSD étend lui-même le dilemme du prisonnier. Un des objectifs de notre contribution est de proposer une formalisation générique de jeux non-coopératifs à N joueurs. Nous proposons aussi l'implémentation de jeux dont la structure de coopération est facilement modulable. Enfin, nous proposons également une première version d'algorithme permettant d'adresser de tels jeux.

5.1 Introduction et motivations

Les dilemmes sociaux séquentiels (SSD) sont une extension du dilemme du prisonnier dans laquelle les actions atomiques (soit Coopération/Défection dans le cas discret, soit un degré de coopération dans le cas continu) sont remplacées par des politiques complexes qui modélisent un comportement plus ou moins coopératif. Comme détaillé dans le chapitre 2, le travail de [LZL+17] a introduit les SSD en utilisant des politiques de RL pour modéliser les comportements des joueurs. Dans les études menées dans ces travaux, il apparaît que les politiques complexes et coopératives peinent à être apprises par les agents, ce qui rejoint un problème déjà rencontré dans le cas d'actions atomiques. Des formalisations de SSD avec plus de deux joueurs ont été proposées dans la littérature comme dans [HLP+18] et [PLZ+17]. Cependant, dans chacun de ces modèles, il est supposé que la coopération entre les acteurs ne peut se faire systématiquement que de manière

réciproque et dans une vision "un-vs-les autres" sans distinction de joueurs potentiellement égoïstes, c'est-à-dire des agents qui agissent uniquement en cherchant à maximiser leur intérêt personnel.

Comme motivé dans le chapitre 4 qui étend le dilemme du prisonnier classique à une structure asymétrique, nous nous proposons d'étendre également le modèle des SSD au cas où la coopération n'est pas nécessairement rendue directement mais plutôt de manière circulaire via un ou plusieurs autres acteurs. Rappelons que les SSD sont des situations où des agents agissent en jouant des politiques complexes (*e.g.* de type RL) et où chacun est tenté de suivre une stratégie égoïste par une incitation telle que la crainte de se faire exploiter ou l'avidité et tentation d'exploiter des coopérateurs, ou bien les deux en même temps. Il existe des situations concrètes où la coopération est orientée et il n'est pas possible ou optimal de la rendre directement. Nous présentons dans la Figure 5.1 des exemples liés à l'apprentissage fédéré, la robotique ou bien dans les Télécoms.



(a) Federated Learning (b) Véhicules autonomes/robots (c) Opérateurs de Télécom

FIGURE 5.1 – Exemples d'agents intelligents dans des situations de coopération asymétrique. Fig 5.1a : Dans les cas de Federated Learning avec agents non-coopératifs, il peut arriver que selon la répartition des données de l'environnement dans lequel chaque agent apprend, les agents peuvent être confrontés à des coopérations asymétriques circulaires. Fig 5.1b : Dans cette situation de robotique, les agents dévient à tour de rôle de leur trajectoire pour laisser passer un autre agent. Cette aide (unilatérale) leur cause à court terme une perte de temps et d'énergie. Le service leur est rendu indirectement par un autre agent, de sorte que la coopération mutuelle se fait de manière circulaire. Fig 5.1c : Il peut être intéressant pour des opérateurs de Télécom de prendre en charge des terminaux d'un autre opérateur s'ils sont plus proches des antennes pour gagner en qualité de service. Dans cet exemple, Orange et Gris ont trouvé un arrangement réciproque, tandis qu'il existe un cycle de coopération Orange-Bleu-Rouge dans lequel il n'est pas optimal ou pas possible de rendre service réciproquement.

Nous avons déjà évoqué des cas concrets dans le chapitre 4 pour le partage de ressources, on peut également, comme évoqué en introduction, l'étendre au partage de poids de modèle dans un contexte de *Federated learning* avec agents non-coopératifs (Figure 5.1a). En effet, si

les types de données utilisées pour entraîner les modèles sont répartis de manière hétérogène d'un agent à l'autre, un agent peut être utile à un autre agent sans toutefois que cet agent puisse lui être utile (si le type de données qu'il collecte ne l'intéresse pas etc). Dans le cas du *Federated Learning*, l'utilité que chaque agent cherche à maximiser peut être considérée comme un mélange de consommation d'énergie (liée au calcul) et de précision (*accuracy*) du modèle entraîné. Les situations avec des véhicules autonomes ou des robots domestiques se partageant un même environnement peuvent également illustrer des coopérations intéressantes qui peuvent être rendues via un ou plusieurs tiers. Ici, l'utilité à optimiser est l'opposé du temps, de l'énergie/autonomie ou bien de la distance parcourue (Figure 5.1b). Enfin, dans les contextes d'opérateurs de télécommunications, on peut envisager le cas où un opérateur prend en charge un client lorsqu'il est plus proche de son antenne qu'il ne l'est de l'antenne de son opérateur fournisseur. Si le service réciproque est rendu, il y a gain de qualité de service chez les deux opérateurs. Bien que le service rendu directement soit possible, il peut exister des situations où le service ne peut être rendu que via un autre acteur (Figure 5.1c).

Convaincus de l'intérêt d'étudier l'extension non réciproque des SSD, nous proposons une définition de cette extension que nous appelons dilemme social séquentiel circulaire ou *Circular Sequential Social Dilemma* (CSSD). Il s'agit d'un jeu de Markov où les joueurs n'ont pas d'intérêt à aider seul les autres, et où il existe au moins un cycle de coopération dans lequel chaque joueur contribue au bien commun s'il aide son successeur (mais pas son prédécesseur) dans le cycle. Il est à noter que notre formalisme inclut également le SSD classique, il suffit en effet de considérer plusieurs cycles triviaux de deux joueurs. Pour accompagner ce formalisme, nous avons développé trois jeux de type RL dont la structure de coopération est conférée par un graphe pondéré et orienté qui est ajustable.

En ce qui concerne les approches pour adresser ce type de jeu, comme détaillé dans le chapitre 2, la littérature propose plusieurs approches parmi lesquelles nous pouvons retenir les suivantes : dans [LP17, EHK+19], les auteurs étudient l'association d'un Tit-for-Tat (TFT) avec des politiques de RL pré-entraînées. [HLP+18] modifie la récompense des agents de RL pour les rendre moins attirés par l'inéquité. Enfin [JLH+18] inclut dans la récompense une motivation intrinsèque pour coopérer. L'approche qui nous semble plus robuste à la défection tout en garantissant une incitation à la coopération mutuelle est l'approche hybride à base de TFT et de politiques de RL, elle nous semble prometteuse pour éviter l'écueil des équilibres de Nash non-optimaux. Nous nous en sommes donc inspirés pour adresser notre *Circular Sequential Social Dilemma* (CSSD), et l'avons étendue grâce au *Graph-based Tit-for-Tat* (GTFT) développé dans le chapitre 4. Brièvement, cette approche consiste dans un premier temps à pré-entraîner en *self-play* des politiques de différents niveaux de coopération. Dans un second temps, au cours de l'exécution avec les autres joueurs, chaque itération du jeu se décompose en plusieurs étapes : commencer par détecter le graphe des degrés de coopération pour chaque couple de joueurs ainsi

que le graphe du potentiel de coopération maximale du jeu. Ensuite, à l'aide des graphes détectés, notre brique GTFT recherche le meilleur cycle de coopération (approche de flot maximal) et en déduit une réponse de négociation adaptée en calculant les degrés de coopération idéaux envers chaque autre joueur. Enfin compte tenu de ces degrés, la politique correspondante est choisie pour en déduire l'action optimale, *i.e.* qui est intéressante personnellement, sans oublier d'inciter aussi à coopérer. Nous évaluons notre approche à l'aide de métriques sociales (efficacité, sûreté et incitation) et l'avons comparée à d'autres approches, notamment des politiques égoïstes entraînées indépendemment (*i.e.* qui maximisent le seul gain personnel) ainsi que l'approche dont nous nous sommes inspirés [LP17] mais avec une brique de TFT classique sans structure de graphe. Nous avons évalué les performances sur plusieurs scénarios et structures de coopération au sein des jeux implémentés. Nous avons également évalué indépendamment les différentes composantes de notre approche, notamment les briques de détection de la coopération entre joueurs. Les évaluations montrent que notre approche à base de GTFT fonctionne mieux que les politiques égoïstes et les approches déjà existantes. En effet, les métriques sont meilleures sur les structures non-bilatérales, et pas moins bonnes pour les structures bilatérales. En revanche, la détection de la coopération reste un défi majeur. Si la détection du graphe de potentiel de coopération conféré par le jeu reste correcte, la détection du graphe de coopération entre les joueurs est très instable. Malgré les résultats mitigés sur cette composante, nous sommes convaincus que ce type d'approche offre de nombreuses perspectives et possibilités d'améliorations.

	Actions atomiques : Coopération/Défection, ou degré de coopération	Politiques complexes : Politiques de RL d'un certain degré de coopération
Coopération réciproque, Approches "un-contre-un" ou "un-vs-les autres"	Dilemme du prisonnier : — classique [RCO65, AH81] — continu [Ver98] — à N joueurs [Ham73, YD94]	*Sequential Social Dilemma* (SSD) : — à 2 joueurs [LZL+17, LP17] — à N joueurs [HLP+18]
Coopération asymétrique, Distinction des joueurs	***Graph-based Iterated Prisoner's Dilemma* (GIPD) Chapitre 4**, [LGMLR22]	***Circular Sequential Social Dilemma* (CSSD) Chapitre 5**

Table 5.1 – Positionnement des contributions des chapitres 4 et 5. Le SSD est une extension du DP par sophistication des actions. Notre GIPD du chapitre 4 est une extension du DP par l'apport d'une structure de graphe. Le CSSD que l'on introduit dans ce chapitre est donc à la fois une extension du SSD par l'apport d'une structure de graphe, ainsi que de manière logique une extension de notre GIPD par sophistication des actions (actions atomiques transformées en politiques de RL). Enfin, notons que dans les deux cas, notre extension avec structure de graphe inclut les modèles sans graphe : notre GIPD comprend le DP et notre CSSD comprend le SSD.

Ce chapitre est organisé comme suit : après avoir évoqué dans la section 5.2 les modèles et approches qui existent dans la littérature, la section 5.3 est consacrée à l'introduction de notre formalisme de CSSD, y sont également rappelées quelques formulations et définitions importantes pour la compréhension du formalisme. Comme précisé plus haut, des jeux de Markov ont été implémentés pour illustrer ce formalisme, nous les présentons et détaillons dans la section 5.4. Notre approche hybride à base de TFT et RL est détaillée ensuite dans la section 5.5. Enfin, nous évaluons notre agent dans diverses structures de jeux, nous présentons les résultats et discussions dans la section 5.6. En conclusion, nous nous attarderons sur les limitations de notre modèle ainsi que de notre approche hybride, et nous évoquerons les nombreuses perceptives et enjeux.

5.2 Dilemmes sociaux séquentiels dans la littérature

Les dilemmes sociaux ont été introduits par [Flo58] puis étudiés davantage dans un tournoi de IPD proposé par Axelrod [AH81]. L'algorithme sorti victorieux de ce challenge est le TFT [RCO65]. Dès lors, de nombreuses variantes et approches différentes ont émergé : en particulier les versions continues du IPD et du TFT [Ver98].
Plus récemment, avec l'essor du Deep RL qui permet l'apprentissage de politiques complexes, des dilemmes sociaux plus sophistiqués appelés *Sequential Social Dilemma* (SSD) ont été formalisés par [LZL+17] dans lesquels des simulations de RL multi-agents ou *Multi-Agent Reinforcement Learning* (MARL) ont été menées. [HLP+18] a étendu alors les définitions formelles des SSD aux cas à plus de deux joueurs avec une approche "un-vs-tous les autres". Ce formalisme diffère de notre modèle sur deux points : premièrement, il ne permet pas de différencier certains joueurs et deuxièmement, il suppose une coopération homogène et symétrique, ce qui n'est pas le cas dans certains scénarios réels. En ce qui concerne les approches pour entraîner des politiques à réagir de manière pro-sociale mais sûre, [LP17] a montré empiriquement que les politiques de RL pures apprises par des agents égoïstes convergent vers les équilibres de Nash (non Pareto-optimaux) et qu'une brique de négociation de type TFT était alors utile pour palier le problème d'avidité et de crainte d'être exploité. Une approche mêlant des stratégies de Tit-for-Tat et des politiques de RL est ainsi proposée par [LP17] (voir section 2.3.3). Également convaincus par la nécessité d'inclure, dans les politiques, des stratégies robustes à la défection comme le TFT, notre approche se base également sur une telle approche hybride. Elle diffère fortement cependant de [LP17] puisqu'elle a vocation à adresser les problèmes à $N \geq 2$ joueurs et à les distinguer séparément, là où les anciennes approches proposent un point de vue soit à deux joueurs "1-vs-1", soit "1-vs-tous les autres". Par ailleurs, notre approche a pour objectif d'être adaptée à une coopération qui n'est pas nécessairement réciproque puisqu'elle utilise notre GTFT, une version de TFT à structure de graphe. [EHK+19] utilise également le TFT pour que les agents de RL tendent vers la réciprocité, et ce dans des jeux à plus de deux joueurs. En revanche comme [LP17], l'approche n'adresse

que les situations où la coopération peut se rendre de manière directe. D'autres travaux ont également abordé le sujet de la coopération entre agents de RL dans des jeux de type SSD : une motivation intrinsèque sous la forme d'une récompense modifiée aide à faire émerger des comportements coopératifs [JLH+19] ou encore un *Actor-Critic* qui est modifié pour améliorer la coordination entre plusieurs agents dans des environnements coopératifs ou compétitifs [LWT+17]. Des approches de MARL dans des jeux avec plus de deux joueurs ont été également étudiées dans un point de vue de l'appropriation de ressources communes (question environnementale notamment) [PLZ+17]. Cette approche voit les joueurs comme un tout et n'a pas pour objectif de cibler de manière précise des joueurs. Enfin, [LYNW20] propose une approche complètement différente qui consiste à entraîner un agent tiers à intervenir dans l'environnement pour modifier les récompenses des joueurs selon des critères de coopération. Cependant, cette approche n'est pas adaptée à nos hypothèses puisqu'elle nécessite un contrôleur intermédiaire qui régulerait et inciterait la coopération.

5.3 Modèle de dilemme social séquentiel non réciproque

Dans cette section, nous définissons notre extension du modèle de dilemme social séquentiel (SSD) qui permet en particulier la modélisation d'un jeu à plus de deux joueurs où la coopération peut se faire via un tiers. Nous appelons ce modèle un dilemme social séquentiel circulaire (CSSD). Notons que notre modèle inclut le SSD classique à deux joueurs ainsi que le modèle SSD uniforme à N joueurs. Dans cette section, nous commençons par rappeler brièvement le formalisme des jeux de Markov avant de détailler le formalisme utilisé par [LZL+17] pour définir les SSD. Nous introduirons alors notre extension de SSD : un jeu non nécessairement symétrique à $N > 2$ joueurs.

5.3.1 Quelques rappels sur les dilemmes sociaux séquentiels

Pour la compréhension de notre proposition de modèle de dilemme social séquentiel non nécessairement réciproque, nous allons rappeler quelques notions déjà détaillées dans les chapitres 1 et 2, à savoir les jeux de Markov et les dilemmes sociaux séquentiels à deux joueurs définis par [LZL+17].

Jeux de Markov à N joueurs

Pour modéliser ces jeux, il est commun d'utiliser le formalisme des jeux de Markov, qui sont définis comme des processus de décision markovien partiellement observable (*Partially Observable Markov Decision Processes* (POMDP)) à N joueurs [Sha53, Lit94]. Un jeu de Markov $\mathcal{M}$ à N joueurs est défini dans la littérature par un sextuplet $(\mathcal{I}, \mathcal{S}, \mathcal{A}, O, \mathcal{T}, R)$ où $\mathcal{I} = \{1, ..., N\}$ est un ensemble de joueurs, $\mathcal{S}$ est un ensemble d'états et $O : \mathcal{S} \times \mathcal{I} \to \mathcal{S}$ est une fonction

d'observation. $\mathcal{A} = \mathcal{A}^1 \times ... \times \mathcal{A}^N$ est l'ensemble des actions jointes et une action jointe notée $\vec{a} = (a^1, ..., a^N)$ transforme un état s du jeu selon la fonction de transition stochastiques $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \to \Delta(\mathcal{S})$. Enfin, une fonction individuelle de récompense $R : \mathcal{I} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ donne une récompense r^i à chaque joueur i (la récompense jointe est notée $\vec{r} = (r^1, ..., r^N)$). L'objectif de chaque agent i est de trouver une politique optimale $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ ($\vec{\pi}$ désignant la politique jointe $(\pi^1, ..., \pi^N)$) de manière à maximiser la récompense totale γ-discounted définie par :

$$V^i_{\vec{\pi}}(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^i(s_t, \vec{a}_t) | \vec{a}_t \sim \vec{\pi}, s_{t+1} \sim \mathcal{T}(s_t, \vec{a}_t)]$$

Définitions des dilemmes sociaux séquentiels

Les SSD dont une définition est proposée par [LZL+17] sont une extension des dilemmes sociaux aux actions atomiques (coopération/défection) en des jeux dont les stratégies possibles sont des politiques complexes de coopération ou de défection, telles que des politiques de RL. Pour les définir, on considère des politiques π^C et π^D dites de coopération et de défection. On dit alors que le jeu est un dilemme social séquentiel si les fonctions de gain empirique $S(s), P(s), R(s), T(s)$ définies en 5.1 vérifient les inégalités des dilemmes sociaux (voir Chapitre 1) en particulier celles du DP : $S(s) < P(s) < R(s) < T(s)$ avec :

$$\begin{aligned} R(s) &:= V^1_{\pi^C,\pi^C}(s) = V^2_{\pi^C,\pi^C}(s) \\ P(s) &:= V^1_{\pi^D,\pi^D}(s) = V^2_{\pi^D,\pi^D}(s) \\ S(s) &:= V^1_{\pi^C,\pi^D}(s) = V^2_{\pi^D,\pi^C}(s) \\ T(s) &:= V^1_{\pi^D,\pi^C}(s) = V^2_{\pi^C,\pi^D}(s) \end{aligned} \tag{5.1}$$

5.3.2 Politiques de coopération continue

Avant de rentrer dans les détails de notre nouveau modèle de SSD détaillé en section 5.3.3, il est nécessaire d'introduire une légère extension des définitions de [LZL+17, HLP+18] pour adapter le concept de politiques coopératives à des politiques au degré de coopération continu. Supposons que chaque joueur est libre de choisir un degré de coopération vers chacun des autres joueurs. Formellement, si l'on note l'ensemble des politiques $\Pi^{(i)} = \Delta(\mathcal{A}^i)^S$ pour le joueur i, alors on définit empiriquement la politique $\pi^i_{\overrightarrow{C_i}} \in \Pi^{(i)}$ comme la politique du joueur i dans laquelle il a un comportement graduellement coopératif envers le joueur j décrit par le degré $c_{i,j}$ donné par le vecteur $\overrightarrow{C_i} = (c_{i,j})_j$ (entre 0 pour une défection totale et 1 pour une coopération totale). Notons que nous ne définissons par formellement ce que le degré de coopération continu signifie. Nous y reviendrons plus loin, en expliquant que ce degré sert à pré-entraîner des politiques de sorte à valoriser le gain du joueur j avec un poids $c_{i,j}$ (voir section 5.5.3), et le fait que nous définissons ce degré comme la probabilité de choisir une politique totalement coopérative.

Notons maintenant $G^{(i)}$ la fonction de valeur pour chaque joueur i correspondant à l'ensemble des degrés de coopération : $G^{(i)} : [0,1]^{N\times N} \times \mathcal{S} \to \mathbb{R}$. Cela signifie que pour un état s, le joueur i reçoit une espérance de gain définie par : $G^{(i)}((c_{i,j})_{i,j}, s) = V^i_{\vec{\pi}}(s)$ où $\vec{\pi} = (\pi^j_{\overrightarrow{C_j}})_j$ dépend des N^2 degrés de coopération (par convention, $c_{i,i} = 1$).

5.3.3 Dilemmes sociaux séquentiels circulaires

Nous introduisons une nouvelle notion de dilemme social séquentiel circulaire qui pourrait plus généralement être défini comme un dilemme social séquentiel basé sur un graphe orienté.

Dilemme Social Séquentiel Circulaire (CSSD)

Définition 5.1 *: Avec les notations de 5.3.2, on dit que le tuple $(\mathcal{M}, \Pi, G)$ est un dilemme social séquentiel circulaire s'il existe des états $s \in \mathcal{S}$ pour lesquels il existe au moins une suite finie cyclique $(i_k)_{k\in[\![0,K-1]\!]}$ indexant K joueurs distincts ($2 < K \leq N$) de $\mathcal{M}$ telle que :*

Pour tout $k \in [\![0, K-1]\!]$ (par abus de notation : $k \equiv i_k$ pour une meilleure lisibilité et avec $i_K = i_0$ pour la circularité) :

1. *$G^{(k)}$ est constant par rapport à $c_{k+1,k}$*
   i.e le joueur $k+1$ ne peut pas directement aider k quelque soit sa volonté de coopérer
2. *$G^{(k)}$ augmente par rapport à $c_{k-1,k}$*
   (tentation d'exploiter ou avidité)
3. *$G^{(k)}$ décroît par rapport à $c_{k,k+1}$*
   (crainte d'être exploité)
4. *Il existe des valeurs de $\mathcal{C} \in [0,1]^{N\times N}$ telles que*
   $\frac{\partial G^{(k)}}{\partial c_{k,k+1}}(\mathcal{C}) + \frac{\partial G^{(k+1)}}{\partial c_{k,k+1}}(\mathcal{C}) > 0$
   (propriété de somme positive : le coût du "donneur" est inférieur au gain du "receveur")

Remarquons que la définition proposée fait l'hypothèse qu'il doit exister des cycles de coopération à $K > 2$ joueurs. Néanmoins, on peut considérer le cas où $K = 2$ si on supprime la condition 1 : on est alors ramenés à la définition d'un dilemme social séquentiel (au degré de coopération continu) à deux joueurs et donc réciproque. Enfin, on retrouve les définitions de [LZL+17] si l'on considère des politiques au degré de coopération discret (coopératives ou défectives). Nous soutenons donc que notre modèle est plus générique puisqu'il inclut les versions précédentes des SSD tout en permettant d'adresser des jeux à la structure de coopération plus complexe.

Intuition du modèle

Pour une meilleure compréhension et intuition, nous reprenons l'exemple simple déjà utilisé dans le chapitre 4. Il s'agit d'un problème d'échange de ressources : plusieurs acteurs possèdent des items (*e.g.* des ressources périssables comme de l'électricité, de l'énergie ou de la connectivité) avec l'hypothèse principale que l'utilité marginale de ces items décroît. En d'autres termes, plus la quantité d'un même item augmente, plus l'utilité d'un item supplémentaire est faible en vertu du principe de l'utilité marginale décroissante [Gos54].

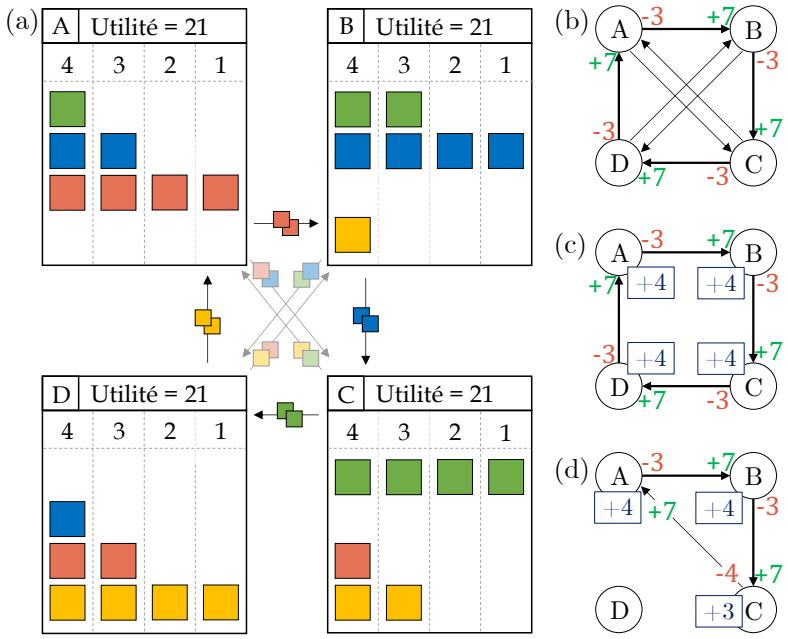


FIGURE 5.2 – Partage de ressources dont l'utilité marginale est décroissante (e.g. $5 - k$ pour le k^{ieme} item). (a) désigne la quantité actuelle des items pour les agents, montrant le potentiel cycle optimal de coopération à 4 joueurs. Le graphe (b) montre le gain potentiel et la coopération, (c) et (d) sont respectivement des situations avec quatre et trois coopérateurs : en rouge le coût du donneur et en vert le gain d'un receveur ; les valeurs encadrées indiquent le gain du bien commun de la transaction.

Dans le jeu présenté dans la Figure 5.2, quatre acteurs A, B, C et D possèdent des items de couleur différente, leur utilité marginale décroît (4 pour le premier, 3 pour le second, etc.). Dans certaines situations, les acteurs ont intérêt à coopérer pour homogénéiser les ressources de sorte que chacun augmente son utilité. Cependant, si l'on suppose que les joueurs sont intéressés uniquement par leur profit, la situation devient alors un dilemme puisqu'il est risqué d'aider seul un autre agent où lui cédant des items. De même, il est tentant de se voir recevoir des items sans en rendre en retour. La situation restreinte à chaque paire de joueurs peut être vue comme un dilemme du prisonnier classique comme exposé en Figure 4.4a (du chapitre 4). Nous pouvons alors utiliser cette situation pour expliquer et donner une intuition du problème en version non

réciproque voire circulaire. Dans l'exemple 5.2, il existe plusieurs cycles de coopération, *e.g.* dans le cycle $< A, B, C, D >$ (Figure 5.2c), chaque joueur peut aider utilement le prochain dans le cycle (utilement au sens où il y a un gain du bien commun, à savoir la somme des utilités). En suivant ce cycle optimal, les joueurs gagnent un surplus d'utilité de +4. Dans le sens opposé, chaque agent pourrait aider de manière moins utile le précédent. Il existe donc une solution moins optimale conduisant à un gain de +1 dans le sens inverse. Notons que dans le cas où les joueurs ne seraient plus que trois à vouloir coopérer (par exemple dans une situation avec un défecteur), il existe des cycles alternatifs moins optimaux à trois (avec un gain d'utilité personnelle de +3), voire des paires de cycles à deux joueurs.

Pour conclure, l'idée principale de notre modèle est de décrire le plus génériquement possible des jeux non-coopératifs dans lequel il existe plusieurs cycles plus ou moins optimaux de coopération entre les joueurs. Notons que notre modèle couvre également les cycles triviaux à deux joueurs et donc les situations parfaitement homogènes à plusieurs cycles triviaux ce qui revient finalement aux dilemmes sociaux séquentiels tels qu'exposés dans [LZL+17] (à deux joueurs) et [PLZ+17, HLP+18] (à N joueurs).

5.4 Jeux et bancs de test

Pour illustrer le modèle de dilemmes sociaux séquentiels asymétriques et circulaires, nous avons implémenté plusieurs jeux dont les structures de coopération possibles sont entièrement modulables. Ces jeux peuvent alors être utilisés pour tester des approches de politiques et agents.

Dans la perspective d'évaluer l'impact de notre extension de la méthode hybride à base de *Graph-based Tit-for-Tat* (GTFT), nous proposons d'abord deux jeux simples dont la particularité est de pouvoir facilement conférer n'importe quelle structure au potentiel de coopération entre les joueurs. En particulier, des structures bilatérales ou bien circulaires. Les deux jeux introduits sont les jeux Collect et Sharing que nous allons détailler dans la suite. Enfin, un troisième jeu est proposé : Traffic, un jeu dont la structure de la coopération non modulable est purement circulaire.

5.4.1 Présentation des jeux

Dans cette section, nous allons présenter les trois jeux implémentés, tout en détaillant leurs caractéristiques modifiables ou non, l'objectif, les fonctions de récompense, les observations et les actions possibles.

Jeu "Collect"

Le jeu COLLECT est un jeu classique de type *grid-world* communément utilisé dans les paradigmes de type agent RL. Un nombre N de joueurs évoluent chacun dans leur chambre (sous-ensemble disjoint de la grille), ils peuvent se déplacer grâce à cinq actions : Haut, Bas, Gauche, Droite, Immobile. Ils ont la possibilité de collecter des pièces de N couleurs différentes qui apparaissent et disparaissent de manière stochastique. Les joueurs obtiennent une récompense de $+2$ quand une pièce de leur couleur est collectée peu importe le joueur qui l'a collectée. En revanche, lorsqu'ils collectent une pièce qui n'est pas de leur couleur, ils reçoivent une récompense de -1. Nous sommes donc en présence d'un dilemme social : le coût de l'aide est plus faible que le gain du service rendu et la coopération seule est pénalisante.

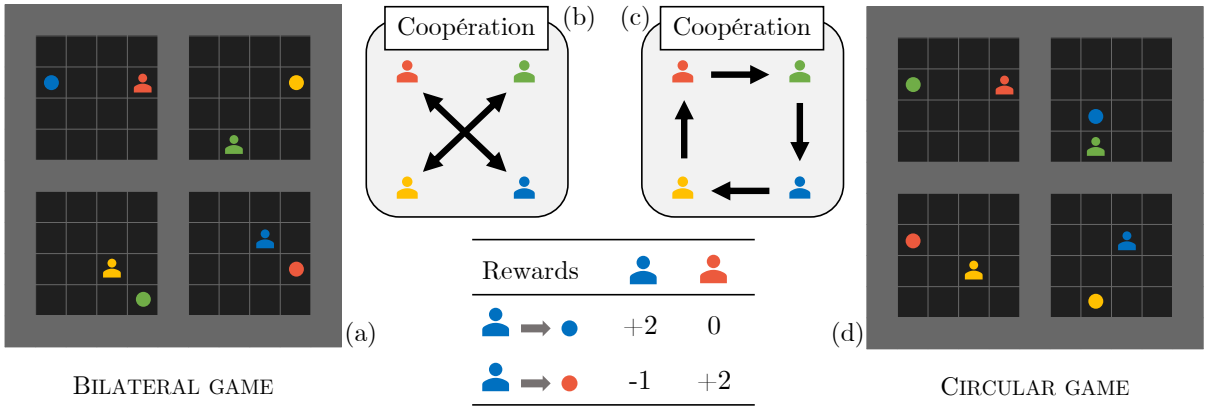


FIGURE 5.3 – Deux exemples du jeu COLLECT : la configuration bilatérale (a) avec son graphe de coopération associé (b) où les pièces apparaissent de manière à ce qu'il existe deux jeux à coopération bilatérale ; et une situation circulaire (d) avec son graphe associé (c).

La taille de la grille et des chambres ainsi que le nombre de pièce par chambre est modifiable. Cependant pour des raisons de simplicité et de visibilité, nous fixons pour la suite le nombre d'agents (et donc de chambres) à quatre car il permet de nombreuses situations intéressantes. Il est supposé que les joueurs peuvent observer l'état total de la grille et toutes les actions des joueurs. La disparition d'une pièce est gouvernée par une distribution exponentielle. Quand une pièce disparaît ou est collectée, une autre pièce réapparaît dans la chambre de sorte qu'il soit présent un nombre fixe de pièces par chambre. Le choix de la couleur de la pièce qui apparaît est régi par une matrice stochastique P dont les coefficients $P(i,k)$ définissent la probabilité que la pièce qui apparaît dans la chambre du joueur i soit de couleur k. Les choix de cette matrice P sont discutés en section 5.4.2.

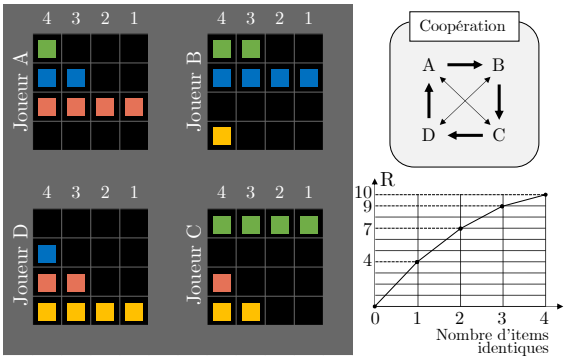
Jeu "Sharing"

Le jeu SHARING symbolise les situations de marché et de négociation où plusieurs agents peuvent partager des ressources dont l'utilité marginale est décroissante [Gos54]. En d'autres termes, une quantité trop importante d'une même ressource n'est pas intéressante (par ex, la 6^{e} part de pizza est moins appréciée que la première, au même titre que 100€ n'ont pas la même valeur que l'on soit riche ou pauvre). Compte tenu d'une telle fonction d'utilité (fonction concave), l'objectif est alors d'échanger des ressources de manière à équilibrer les quantités et ainsi augmenter à la fois l'utilité personnelle et le bien commun (somme des utilités). Le jeu fait intervenir N acteurs qui peuvent partager des items de M catégories différentes (matérialisées par M couleurs différentes). Chaque joueur peut stocker au plus Q items d'une même catégorie (pour illustrer le caractère périssable ainsi que le stockage coûteux voire impossible de certaines ressources telles que l'eau, l'énergie, la connectivité, les données etc). A chaque étape, les joueurs reçoivent une récompense pour chaque item stocké qui est donnée par une fonction d'utilité concave, issue d'une utilité marginale décroissante : $[Q+1-k]^{+}$ pour le k^{ieme} item d'une même catégorie (pour $Q=4$, cela donne 4 pour le 1^{er}, 3 pour le 2^{nd} etc). Chaque item apparaît et disparaît de manière stochastique : la probabilité que le joueur i reçoive un item de catégorie k est donnée par $P(i,k)$ les coefficients d'une matrice P (qui est stochastique quand le nombre d'agents N est égale au nombre d'items M, donnant alors lieu à des situations comme expliqué en section 5.4.2). Chaque joueur est capable d'observer le stock des autres joueurs ainsi que les actions effectuées qui sont les transactions (dons) d'items entre chaque paire de joueurs.

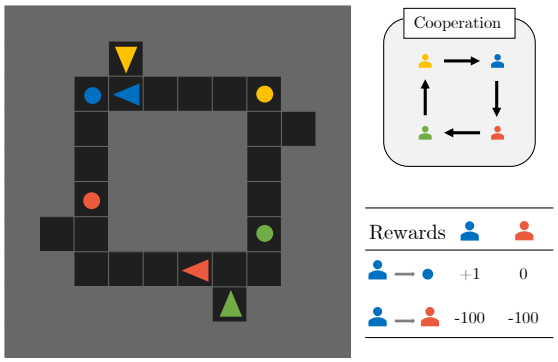
Jeu "Traffic"

Le jeu TRAFFIC reprend l'idée de la Figure 5.1b. Il implique quatre joueurs se déplaçant dans un corridor circulaire étroit. Les joueurs doivent collecter des pièces de leur couleur et ne peuvent se croiser. Ils reçoivent une récompense de $+1$ pour chaque pièce collectée et une pénalité de -100 en cas de collision/rencontre. Les pièces apparaissent de manière stochastique dans un sens horaire pour deux agents et dans un sens anti-horaire pour les deux autres, de sorte à les forcer à choisir un sens optimal (horaire ou anti-horaire). Les pièces apparaissent proches de l'agent dans le sens horaire ou anti-horaire : plus précisément dans un graphe polaire dont la différence d'angle $\Delta\theta$ entre la position du joueur et celle de la nouvelle pièce est choisi stochastivement selon des lois exponentielles : $\Delta\theta \sim Exp(\theta_{mean})$ pour deux des joueurs et $\Delta\theta \sim -Exp(\theta_{mean})$ pour les deux autres. Les actions sont Avancer, Rotation gauche, Rotation droite, Immobile. Les joueurs ne doivent pas se croiser, mais ils sont incités à tourner dans des sens différents. C'est pourquoi pour observer et étudier leur coopération, quatre "voies de garage" sont placées de sorte qu'ils se laissent successivement passer. La coopération optimale est une coopération circulaire [1].

1. Il existe un comportement collectif optimal pour ce jeu : `https://youtu.be/PbOWpItBUgg`



(a) Sharing : chaque joueur reçoit un item de couleur C avec une certaine probabilité. Le k^{ieme} item d'une même couleur vaut $[5-k]^+$. Les joueurs peuvent partager leurs items de manière à tous augmenter leurs gains personnels.



(b) Traffic : Un jeu à quatre joueurs dans une situation de dilemme social séquentiel circulaire : le comportement optimal consiste à laisser passer un et un seul joueur à tour de rôle, de manière à se croiser efficacement.

Figure 5.4 – Jeux Sharing et Traffic

5.4.2 Structure de la coopération configurable

Les jeux Collect et Sharing sont intéressants car on peut modifier les probabilités d'apparition des pièces ou des items grâce à la matrice P. Dans le jeu Collect, la probabilité d'apparition d'une pièce de couleur k disponible pour le joueur i est égale à $P[i,k]$ donnée par la matrice P. Remarquons alors que cette matrice peut être considérée comme la matrice d'adjacence qui définit un graphe pondéré et orienté. Ce graphe équivaut précisément au graphe du potentiel de coopération maximal conféré par le jeu.

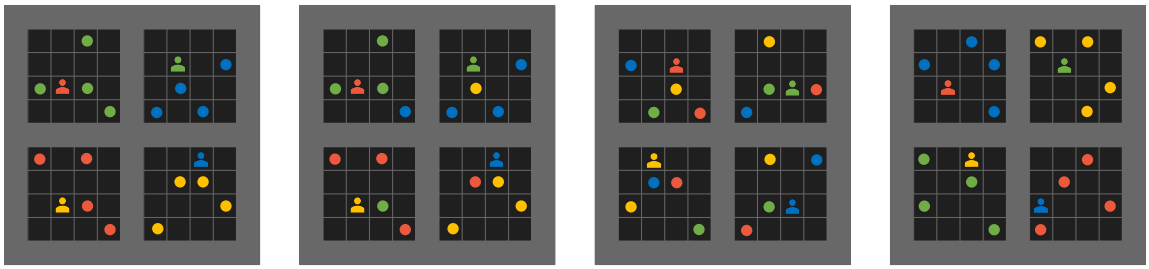
Ce paramètre P peut donc être configuré pour conférer au jeu n'importe quelle structure de potentiel de coopération entre les joueurs. Des exemples sont représentés : les matrices ci-dessous C, S, H et B correspondent aux situations de la Figure 5.5. La matrice C correspond à un environnement avec un potentiel de coopération parfaitement circulaire (Figures 5.5a et 5.5e), c-a-d le joueur Rouge peut n'aider que le joueur Vert qui peut aider le joueur Bleu qui peut aider le joueur Jaune qui enfin peut aider Rouge. La matrice S décrit une situation semi-circulaire, reproduisant le cas circulaire mais comportant des cycles alternatifs moins optimaux en cas de défection de l'un des joueurs (Figures 5.5b et 5.5f). H désigne une situation homogène où chaque joueur peut coopérer de manière équivalente avec chaque autre joueur (Figures 5.5c et 5.5g). Enfin, la matrice B représente un cas bilatéral dans lequel il existe deux cycles triviaux à deux joueurs permettant alors la réciprocité. Le jeu devient en fait deux *Sequential Social Dilemma* (SSD) classiques et indépendants à deux joueurs (Figures 5.5d et 5.5h).

Précisons enfin que ce paramètre P peut être fixé et rester constant au cours du temps.

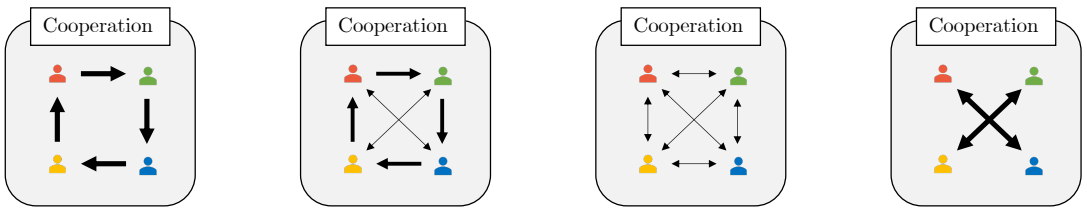
Cependant, il peut être également modifié dynamiquement au cours du jeu, notamment pour observer et étudier la réaction des agents lorsque la structure du potentiel de coopération change subitement (voir section 5.6.4).

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 0 & 0.75 & 0.25 & 0 \\ 0 & 0 & 0.75 & 0.25 \\ 0.25 & 0 & 0 & 0.75 \\ 0.75 & 0.25 & 0 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$



(a) Circulaire (b) Semi-circulaire (c) Homogène (d) Bilatéral



(e) Graphe de coopération, cas Circulaire (f) Graphe de coopération, cas Semi-circulaire (g) Graphe de coopération, cas Homogène (h) Graphe de coopération, cas Bilatéral

FIGURE 5.5 – Jeu COLLECT avec des structures différentes de coopération. Notons que dans le cas homogène, nous n'affichons pas les boucles dans le graphe (de i vers i).

5.5 Une approche hybride

Dans cette section, nous introduisons notre approche, qui est un algorithme hybride mêlant des politiques de RL entraînées en *self-play* et des stratégies de TFT permettant de choisir la

bonne réaction face aux différents partenaires. Notre proposition est inspirée de l'approche de [LP17] et a pour objectif de l'étendre à plus de deux joueurs, avec en particulier la possibilité d'adresser une coopération non nécessairement symétrique et en particulier circulaire. Notre extension utilise la contribution de GTFT introduite au chapitre 4.

5.5.1 Rappels sur les notions de TFT et de RL

L'apprentissage de politiques de RL

L'apprentissage par renforcement (RL) est une méthode pour entraîner des politiques $\pi : s_t \to a$ qui sont des fonctions reliant des observations s_t à des actions a de manière à maximiser un gain total $V_{\vec{\pi}}(s_0)$ [SB+98] (voir section 5.3.1). De nombreuses techniques de RL existent, et certaines ont été détaillées dans le chapitre 2. Nous adoptons une approche agnostique sur le choix de l'algorithme d'entraînement et la structure des politiques. Nous supposons néanmoins que les politiques entraînées en *self-play* sont associées à des fonctions valeur-action qui sont des fonctions qui associent à un couple observation/action (s_t, a_t) une valeur d'espérance de gain obtenu sur le long terme (les fonctions *Q-values* ou d'avantage sont donc adaptées). Par conséquent, l'algorithme d'entraînement peut être un simple Q-learning, ou bien une extension neuronale type Deep Q-network [MKS+13], voire même des méthodes *Actor-Critic* [MBM+16] puisque le modèle du *Critic* est bien une fonction de valeur-action. En revanche, en l'état, notre approche n'est pas adaptée aux méthodes de type *Policy Gradient* car sans fonction de valeur.

Formalisation d'un TFT continu

Notre approche utilise un GTFT développé en détails dans le chapitre 4. Avant de résumer son concept dans la section 5.5.5, rappelons que le principe de base du TFT est de réagir face au choix de coopération d'un partenaire dans un dilemme social avec l'objectif d'inciter à la coopération et d'être robuste à la défection. La version discrète est plutôt simple puisqu'elle consiste à jouer la coopération au début puis de reproduire le choix précédent du partenaire [RCO65]. Dans la version continue du IPD [Ver98], plusieurs approches existent pour la version continue du TFT. On rappelle alors que l'on peut regrouper les variantes existantes dans cette formulation (voir les chapitres 3 et 4 pour plus de détails) :

$$\mathrm{TFT}_{\alpha,\beta,\gamma,r_0,c_0}(t, a_{t-1}, b_{t-1}), r_t = \begin{cases} c_0, r_0 & \text{if } t = 0 \\ \alpha a_{t-1} + (1-\alpha)(r_t + (1-r_t)b_{t-1}), & \\ [r_{t-1} + \beta(b_{t-1} - a_{t-1})]^+ + r_0 \mathcal{B}(1,\gamma)\mathbb{1}_{r_{t-1}=0} & \text{if } t > 0. \end{cases} \tag{5.2}$$

Dans ce formalisme, $\mathrm{TFT}(t, a_{t-1}, b_{t-1})$ permet de calculer le degré de coopération idéal a_t qu'un joueur A devrait choisir pour répondre de manière sure et incitative à un partenaire B

en accord avec la précédente étape (degrés a_{t-1} et b_{t-1}). Il comporte un coefficient d'inertie α permettant de lisser la réponse. Pour encourager l'incitation, on utilise un taux d'incitation dynamique $r_t \in [0, 1]$ qui est modifié au cours des étapes grâce à un coefficient adaptatif β ainsi que par une variable de Bernoulli de paramètre γ dans le cas où l'incitation coopérative s'est totalement arrêtée pour diverses raisons.

5.5.2 Architecture de notre approche

L'idée principale de notre algorithme, à l'instar de [LP17], est la décomposition en deux briques fondamentales : des politiques de RL pré-entraînées séparément et la stratégie de coopération par GTFT. On rappelle que le choix du modèle des politiques de RL et des algorithmes qui les entraînent n'importe peu. Il est néanmoins nécessaire que le modèle de politique comporte une fonction de valeur action/état (*action-value function*) donc tous les algorithmes de type *Value-based* avec une Q-value tels que le Q-learning ou les approches de Deep Q-learning (*Deep Q-Network* (DQN), *Double Deep Q-Network* (DQN)) ainsi que les approches de type *Actor-Critic* avec une fonction avantage (*i.e.* une fonction permettant d'obtenir l'avantage en gain cumulé estimé de choisir une action compte tenu d'un état observé $A_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$) sont compatibles. Par conséquent, les algorithmes de type *Value-based* ou *Actor-Critic* avec une simple fonction de valeur $V_\pi : \mathcal{S} \to \mathbb{R}$ ainsi que les approches simples de type *Policy-Gradient* ne sont pas adaptés à notre approche.

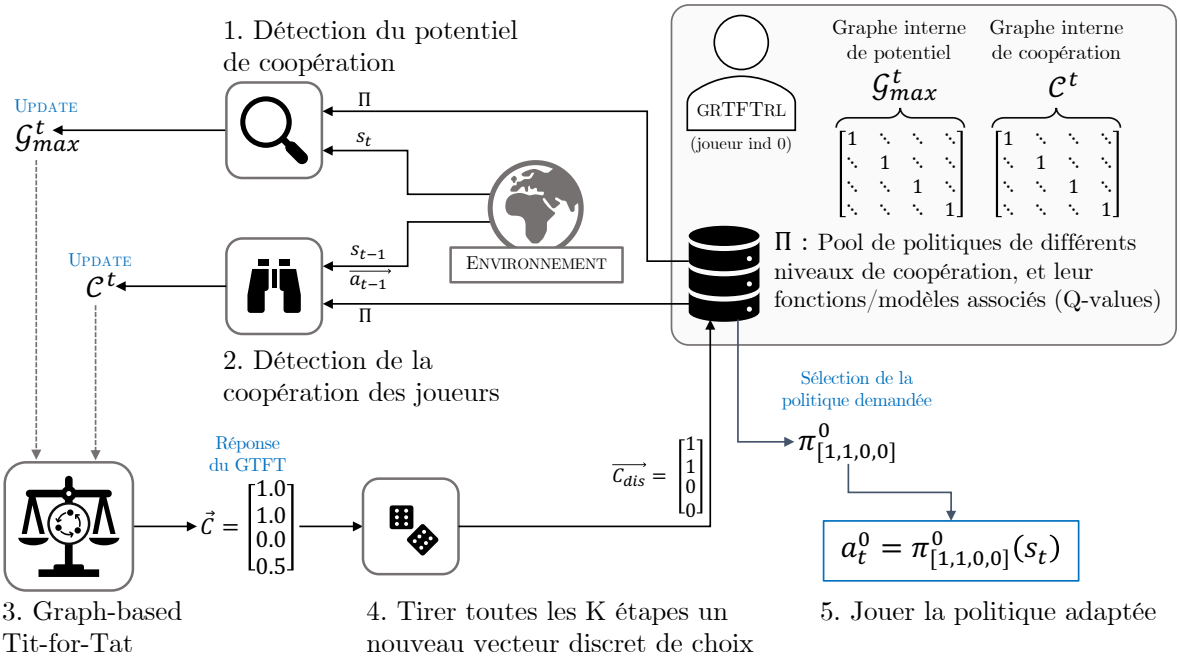


FIGURE 5.6 – Architecture de notre approche GRTFTRL

Pour commencer, notre agent GRTFTRL comporte 3 composantes internes :

- un graphe interne $\mathcal{G}_{max}$ (pondéré et orienté, représenté par sa matrice d'adjacence) qui modélise une estimation du potentiel de coopération entre chaque joueur. C'est le jeu qui définit cette structure et non le comportement des joueurs. Il est actualisé au cours du jeu, dans le cas où la structure du jeu évoluerait.
- un graphe interne $\mathcal{C}$ (pondéré et orienté) qui modélise l'estimation des degrés de coopération pour chaque couple de joueurs. Il est actualisé tout le long du jeu.
- un pool Π de politiques pré-entraînées de différents niveaux et combinaisons de coopération envers les autres joueurs.

Par ailleurs, à chaque itération du jeu, notre agent GRTFTRL exécute cinq étapes (voir Algorithme 5.1). Avant de détailler plus tard les étapes séparément, nous les résumons brièvement :

1. Compte tenu de l'état actuel s_t, et grâce aux politiques pré-entraînées, l'agent estime et actualise son graphe interne de potentiel de coopération $\mathcal{G}_{max}$.
2. Comme pour 1, à l'aide de l'état précédent s_{t-1}, de l'ensemble des actions précédentes $\overrightarrow{a_{t-1}}$ et des politiques pré-entraînées, l'agent estime et actualise son graphe interne $\mathcal{C}$ modélisant l'ensemble des coopérations entre joueurs.
3. La brique de GTFT déduit de $\mathcal{G}_{max}$ et $\mathcal{C}$ une réponse adaptée pour chaque joueur : $\overrightarrow{C}$.
4. Conversion du vecteur continu $\overrightarrow{C}$ en vecteur de choix discret $\overrightarrow{C_{dis}}$ (toutes les K étapes pour stabiliser le comportement). Pour ce faire, une variable de Bernoulli est tirée en utilisant pour probabilité le degré de coopération correspondant. Dans l'exemple de la Figure 5.6, le vecteur de choix désigne donc une stratégie de coopération vers le joueur 1 (et 0 : lui-même).
5. Selon le vecteur de choix de stratégie, la politique adaptée est extraite du pool de politiques, et l'agent peut alors l'exécuter sur l'état actuel s_t pour obtenir l'action a_t^0 à jouer.

5.5.3 Génération de politiques de RL pré-entraînées

Une partie majeure de notre approche repose sur une phase d'entraînement en *self-play* au cours de laquelle des politiques de combinaisons et de niveaux différents de coopération sont entraînées. On suppose alors que que les agents sont capables de générer des politiques $\pi^i_{\overrightarrow{C}}$ définies par un vecteur de coopération $\overrightarrow{C} \in [0,1]^N$ décrivant comment un joueur i devrait interagir avec chacun des autres joueurs. Les politiques $\pi^i_{\overrightarrow{C}}$ sont entraînées par des méthodes de RL à l'aide d'une récompense qui est modifiée en accord avec l'objectif de la coopération cible : $\tilde{r}^i = \langle \overrightarrow{C}, \overrightarrow{r} \rangle = \sum_{j=1}^{N} \overrightarrow{C}[j]\ r^j$ où $\overrightarrow{r}$ est le vecteur de récompenses gagnées par les N agents (défini en section 5.3.1). En théorie, nous pourrions utiliser n'importe quel poids $c_{i,j}$ en suivant le formalisme $\pi^i_{\overrightarrow{C}}$. Cependant, en pratique, les agents n'entraînent qu'un ensemble de politiques issues de vecteurs de coopération discrètes : $\Pi^{(i)} = \{\pi^i_{\overrightarrow{C}}, \overrightarrow{C} \in \{0,1\}^N\}$.

5.5.4 Détection de la coopération

Dans notre approche, les agents doivent détecter d'une part la structure de coopération potentielle maximale que confère le jeu à l'instant présent, d'autre part le comportement le long du jeu des autres joueurs (à savoir leur niveau de coopération envers les uns et les autres). L'agent actualise à chaque étape t un graphe interne $\mathcal{G}^t_{max}$ qui estime la coopération potentielle maximale du jeu (par exemple les Figures 5.5f, 5.5g, 5.5h, 5.5e) ainsi qu'un graphe interne $\mathcal{C}^t$ de coopération des autres joueurs (leur comportement).

Avant de détailler nos propositions, rappelons quelques détails et notations. D'abord, on fait l'hypothèse que chaque agent peut observer l'état entier du jeu s_t ainsi que les actions précédentes choisies par tous les agents $\overrightarrow{a_{t-1}}$. De plus, on rappelle qu'une phase de pré-entraînement en *self-play* a eu lieu dans l'objectif de construire des politiques de différentes structures de coopération qui sont accessibles pendant le jeu. En particulier, notons $Q_{k\to l}$ les *Q-values* de la politique de l'agent k qui coopère avec le joueur l (*i.e.* $Q^k_{\overrightarrow{C}}$ où $\overrightarrow{C}$ a des valeurs 1 aux indices $\{k,l\}$ et 0 sinon), ainsi que $\pi_{k\to l}$ la politique associée. Enfin $V_{k\to l}[s_t] = \max_a Q_{k\to l}[s_t, a]$ caractérise la *V-value* associée.

Pour détecter et actualiser le graphe de coopération maximale $\mathcal{G}_{max}$, nous proposons l'approche suivante. A chaque étape t, l'agent détecte un graphe $\mathcal{G}$ et applique un *soft-update*. Pour la détection actuelle de $\mathcal{G}$, nous proposons d'estimer chaque arc (i,j) (à savoir le potentiel de gain collectif que l'aide du joueur i vers le joueur j permet d'obtenir). On propose alors de se baser sur les fonctions de valeur des politiques apprises en *self-play* et de calculer la différence de gain moyen entre la coopération ou non de la paire de joueurs considérés $V_{i\to j}[s_t] - V_{i\to i}[s_t]$ puis de normaliser cette différence comme suit :

$$\mathcal{G}^t_{max} = (1-\tau)\mathcal{G}^{t-1}_{max} + \tau\mathcal{G}, \quad \text{avec } \mathcal{G}[i,j] = \left[\frac{V_{i\to j}[s_t] - V_{i\to i}[s_t]}{\max_k V_{i\to k}[s_t] - V_{i\to i}[s_t]}\right]^+ \tag{5.3}$$

En ce qui concerne la détection des comportements, à savoir le choix des coopérations des agents, nous proposons également un *soft-update* avec C^{t-1} où chaque arc $C^{t-1}[i,j]$ est calculée par la différence normalisée entre le gain espéré reçu par l'agent i qui suit une politique coopérative dirigée vers j et ce qu'il aurait reçu dans le cas d'une politique égoïste :

$$\mathcal{C}^t = (1-\tau)\mathcal{C}^{t-1} + \tau C^{t-1}, \quad \text{avec } C^{t-1}[i,j] = \left[\frac{Q_{i\to j}[s_{t-1}, a^i_{t-1}] - Q_{i\to i}[s_{t-1}, a^i_{t-1}]}{V_{i\to j}[s_{t-1}] - V_{i\to i}[s_{t-1}]}\right]^+ \tag{5.4}$$

5.5.5 TFT à structure de graphe

La brique de TFT utilisée est celle développée dans le chapitre 4 à base de graphe orienté : GTFT. Rappelons brièvement son principe, ses entrées et sa sortie.

Un agent GTFT a pour paramètres un graphe de coopération potentielle maximale $\mathcal{G}_{max}$ qui est connu ainsi qu'une fonction de TFT classique à deux variables. L'entrée de l'algorithme est une matrice de degrés de coopération et la sortie est la décision de coopération envers tous les autres agents. Pour plus de détails, voir le chapitre 4. On peut donc résumer le formalisme de notre GTFT comme suit :

$$\begin{aligned} &\text{grTFT} : \mathbb{N} \times [\![1, N]\!] \times [0,1]^{N\times N} \times [0,1]^{N\times N} \to [0,1]^N \\ &\text{grTFT}(t, i, \mathcal{G}_{max}, \mathcal{C}^t) = \overrightarrow{C} \end{aligned} \tag{5.5}$$

Pour la recherche de flot maximal, l'agent utilise une variante de l'algorithme de Ford-Fulkerson avec un coût de calcul polynomial ($\mathcal{O}(\Delta N^2)$ où Δ est un coefficient de discrétisation que l'on choisit égal à 10, cela permet d'avoir 11 valeurs : $0.0, 0.1, ..., 1.0$).

Algorithme 5.1 : Algorithme grTFTrl pour l'agent i

Entrées : $\forall j$, un ensemble de politiques au degré de coopération discret $\Pi^{(j)} = \{\pi^j_{\overrightarrow{C}}\}$ et les fonctions de valeurs associées $Q^j_{\overrightarrow{C}}$

Initialiser $\mathcal{G}^0_{max} \leftarrow I_N$, $\mathcal{C}^0 \leftarrow I_N$

for $t \in [1, T_{max}]$ **do**

- Actualiser le graphe de coopération potentiel maximale $\mathcal{G}^t_{max}$; ▷ Voir eq 5.3 (section 5.5.4)
- Détecter et actualiser le graphe de coopération $\mathcal{C}^t$; ▷ Voir eq 5.4
- Appliquer le graph-based TFT : $\overrightarrow{C} = \text{grTFT}(t, i, \mathcal{G}^t_{max}, \mathcal{C}^t)$); ▷ Voir section 5.5.5
- **if** $t \equiv 0 \pmod K$; ▷ Toutes les K étapes (pour stabiliser le comportement)
- **then**
  - Modifier le choix de coopération vers les partenaires : $\overrightarrow{C_{dis}}[k] = \mathcal{B}(1, \overrightarrow{C}[k])$; ▷ Discrétisation stochastique
- **end**
- Obtenir la politique $\pi^i_{\overrightarrow{C_{dis}}}$ en suivant les degrés (discrets) $\overrightarrow{C_{dis}}$
- Choisir l'action $a^i \leftarrow \pi^i_{\overrightarrow{C_{dis}}}(s_t)$

end

5.6 Simulations et résultats

Dans cette section, notre algorithme GRTFTRL proposé est évalué sur le jeu COLLECT [2]. Notre approche est faite de plusieurs composants qui peuvent être évalués séparément. Nous présentons ici entre autres l'impact du choix de l'algorithme de TFT choisi pour la négociation, ainsi que l'impact de ses paramètres. Nous évaluons également la pertinence des modules d'estimation de graphes de coopération, à savoir le graphe du choix de coopération des agents, ainsi que celui correspondant au potentiel de coopération maximale conféré par le jeu. Nous choisissons d'utiliser le jeu COLLECT sous deux types de structure de coopération : BILATERAL et CIRCULAR (voir section 5.4.2 et Figure 5.3).

5.6.1 Métriques sociales

Pour commencer, comme pour les chapitres précédents, on introduit quelques métriques sociales qui permettent d'étudier les aspects coopératifs des politiques. On suppose que $G^i(X_1, ..., X_N)$ se réfère à la somme des récompenses sur le temps de l'agent i dans une situation où chacun des agents j suit une politique X_j dans un jeu à T étapes. De plus, on note C la politique de la coopération naïve (consistant à maximiser le bien commun) et D la politique de défection égoïste (consistant à maximiser son gain personnel). On définit donc trois métriques : la métrique utilitaire U qui mesure le bien commun moyen du jeu, la sûreté (*safety*) Sf mesure à quel point il est risqué de choisir une certaine politique π au lieu de la défection lorsque l'on fait face à des défecteurs. Enfin, l'incitation (*Incentive-compatibility*) IC mesure comment une politique π choisie par $N-1$ agents encourage un dernier agent à choisir la coopération plutôt que la défection.

$$U(\pi) = \mathbb{E}\left[\frac{\sum_{i=1}^{N} G^i(\pi, ..., \pi)}{T}\right], \quad Sf(\pi) = \mathbb{E}\left[\frac{G^1(\pi, D, ..., D) - G^1(D, D, ..., D)}{T}\right],$$

$$IC(\pi) = \mathbb{E}\left[\frac{G^1(C, \pi, ..., \pi) - G^1(D, \pi, ..., \pi)}{T}\right]$$

5.6.2 Performances du Graph-based TFT

Dans cette section, nous évaluons la pertinence de la brique de GTFT dans notre agent GRTFTRL en comparant avec une baseline de type égoïste qui est une simple politique RL (DQN) sans aucune brique de négociation de type TFT. Nous procédons également à une comparaison avec TFTRL qui est une approche similaire [LP17] mais en utilisant la version classique réciproque du TFT sans aucune structure de graphe. Enfin, pour mettre en valeur le compromis entre

2. Le code des jeux, des algorithmes et des simulations est disponible ici : `https://github.com/tlgleo/Circular_Cooperation_MARL`

efficacité et sûreté, nous introduisons dans les évaluations un agent naïf coopératif NICE (qui est une instance de GRTFTRL avec un GTFT naïf coopératif *i.e.* un GTFT qui utilise comme fonction de TFT une fonction triviale toujours égale à 1.0, voir chapitre 4). Cet agent met en valeur l'inefficacité d'une coopération systématique sans conditions [3]. Pour isoler l'étude de l'impact du choix de l'agent de GTFT, nous avons fait le choix de communiquer artificiellement à chaque étape les deux graphes de coopération qui sont normalement estimés par la brique de détection.

(métriques ×100, 5 exécutions (*runs*) de 500 étapes)

	BILATERAL			CIRCULAR		
	U	IC	Sf	U	IC	Sf
EGOIST	2.5 ± 0.5	-15.7 ± 2.4	$\mathbf{0.3 \pm 0.7}$	2.1 ± 0.5	-15.5 ± 1.8	-1.0 ± 0.5
TFTRL	64.1 ± 5.2	12.5 ± 2.5	-1.6 ± 1.3	5.8 ± 0.9	-18.0 ± 2.9	-1.2 ± 1.0
GRTFTRL	67.1 ± 8.0	$\mathbf{16.8 \pm 3.1}$	-1.2 ± 1.4	65.3 ± 1.4	$\mathbf{18.8 \pm 4.1}$	$\mathbf{-0.4 \pm 0.5}$
NICE	$\mathbf{67.8 \pm 2.4}$	-17.8 ± 8.4	-15.2 ± 0.5	$\mathbf{67.8 \pm 2.4}$	-17.8 ± 8.4	-15.2 ± 0.5

TABLE 5.2 – Résultats avec les environnements BILATERAL et CIRCULAR, les paramètres des TFT sont $(\alpha, \beta, \gamma, r_0, c_0) = (0.6, 0.6, 0.1, 0.3, 0.0)$, et l'algorithme de graphe pour GRTFTRL est l'approche *min cost*, voir chapitre 4.

Les premières observations montrent que l'addition d'un TFT confère à l'agent des propriétés de sûreté et d'incitation, TFTRL est en effet plus sûr et incitatif que NICE et EGOIST. Cependant, dans le cas où la coopération réciproque n'est plus possible (cas circulaire), TFTRL avec un simple TFT échoue à converger vers une coopération mutuelle alors que notre approche parvient à trouver un arrangement.

5.6.3 Impact des paramètres de la fonction de TFT

Comme mentionné dans la section 5.5.5, un composant principal de notre graph-based Tit-for-Tat (GTFT) est une fonction de TFT classique à deux variables qui contient elle-même cinq paramètres. L'objectif, ici, n'est pas d'étudier exhaustivement l'impact de la valeur de ces paramètres. Une étude plus complète sur la version seule du GTFT a été faite dans le chapitre 4. Dans cette section, nous proposons d'étudier les trois types de TFT dont nous rappelons plus bas les caractéristiques avec les notations de la section 5.5.1 :

- $TFT\alpha$: comporte simplement un coefficient d'inertie pour lisser les réactions, ainsi qu'un taux d'incitation constant r_0. Les paramètres β et γ sont nuls.
- $TFT\beta$: addition du coefficient β qui permet d'adapter dynamiquement le taux r_t en fonction de la réaction positive ou négative du partenaire. Le coefficient γ est nul.

3. Quelques vidéos de simulations sont disponibles ici :
`https://youtube.com/playlist?list=PLs7f7KUeiGoVeIldhO2wb6NyiNMoCVCQw`

- $TFT\gamma$: addition du coefficient stochastique γ. Il permet avec probabilité γ de réinitialiser le taux r_t à r_0 s'il est devenu nul.

Pour se concentrer sur l'étude du TFT, nous conduisons les simulations dans une configuration où les graphes de coopération sont artificiellement communiqués sans estimation et détection. Nous relâchons cette contrainte dans une expérience avec le $TFT\gamma$ pour évaluer l'impact de la détection (derrière ligne de la Table 5.3).

(métriques $\times 100$, 3 runs de 500 étapes)

	Bilateral			Circular		
	U	IC	Sf	U	IC	Sf
$TFT\alpha$	$\mathbf{66.7 \pm 1.3}$	2.1 ± 4.0	-6.6 ± 3.1	64.5 ± 1.7	2.9 ± 4.4	-6.9 ± 2.9
$TFT\beta$	62.7 ± 1.3	$\mathbf{14.7 \pm 2.1}$	$\mathbf{0.6 \pm 1.7}$	61.3 ± 3.1	13.3 ± 6.5	$\mathbf{1.3 \pm 0.7}$
$TFT\gamma$	62.8 ± 3.5	14.5 ± 2.0	-0.4 ± 0.5	$\mathbf{64.9 \pm 3.2}$	$\mathbf{14.5 \pm 1.9}$	0.3 ± 0.6
$TFT\gamma$-DET	59.9 ± 4.6	-7.7 ± 2.9	-13.2 ± 1.4	57.9 ± 2.0	-16.0 ± 5.7	-14.1 ± 1.9

Table 5.3 – Évaluation de l'impact du choix du GTFT (3 premières lignes) ainsi que l'impact de la détection de la coopération (dernière ligne) sur les jeux Bilateral et Circular. Les paramètres des TFT sont $(\alpha, \beta, \gamma, r_0, c_0) = (0.6, 0.6, 0.1, 0.3, 0.0)$, et l'algorithme de graphe pour grTFTrl est l'approche *min cost*.

On peut observer qu'un paramètre adaptatif β non nul est pertinent pour augmenter la sûreté et l'incitation, mais il conduit à une légère baisse de l'efficacité U. En effet, si le GTFT est trop méfiant, cela peut conduire à une défection qui s'installe durablement et n'évolue plus. Cela est notamment résolu par le paramètre stochastique γ qui fournit un peu de résilience en augmentant un peu l'efficacité, sans pénaliser la sûreté. $TFT\gamma$ offre donc le meilleur compromis.

5.6.4 Impact de la détection de la coopération

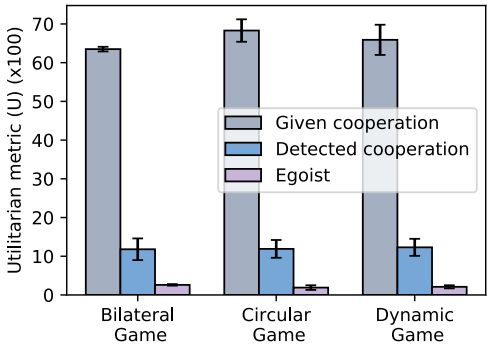


Figure 5.7 – Impact de la détection du potentiel de coopération sur U. 3 runs de 500 étapes sur trois jeux : Bilateral, Circular et Dynamic (jeu dont la structure de coopération change d'une structure bilatérale à une structure circulaire au milieu du jeu à $t = 250$)

Un des challenges, voire le plus difficile, provient de la détection de la coopération. D'une part, l'agent doit être capable de détecter le potentiel de coopération conféré par le jeu. D'autre part et surtout, il doit être capable de détecter le comportement des autres joueurs et d'en déduire une estimation de la volonté de chaque joueur de coopérer avec chacun des autres joueurs. Pour évaluer l'impact de cette détection, nous comparons deux situations : l'une où les graphes de coopération sont artificiellement communiqués

aux agents (le graphe de potentiel de coopération maximale ainsi que celui correspondant aux choix de coopération des agents au coup précédent), et l'une où l'agent doit détecter et estimer lui-même les graphes.

Dans la Table 5.3, les deux dernières lignes ($TFT\gamma$ et $TFT\gamma$-DET) permettent d'évaluer l'impact de la détection du choix de coopération des joueurs. La tache est semble t-il très difficile : on observe que sans la communication de la détection, l'agent peine à maintenir une sûreté et une incitation. La fonction de détection proposée est donc peu optimale. Comme le mentionne [LP17], détecter de tels comportements à l'aide de simples observations et en se basant sur des politiques de RL pré-entraînées est sujet à un fort biais. Cependant, en ce qui concerne l'estimation du potentiel de coopération et donc le calcul du graphe (on rappelle que dans le cas du jeu COLLECT, ce graphe peut être compris comme la matrice stochastique des probabilités d'apparition des pièces), notre approche parvient à calculer un graphe raisonnablement estimé. Empiriquement, il est possible de se faire une idée dans cette vidéo exemple où le graphe change subitement de structure [4]. Dans la Figure 5.7, on montre la métrique utilitaire U (la somme des gains des joueurs) dans quelques situations de structure de coopération (fixée ou dynamique). Bien que les métriques soient assez loin de l'optimal (correspondant au cas où le graphe est artificiellement donné), notre agent parvient néanmoins à atteindre un bien commun meilleur que dans le cas où les agents sont purement égoïstes. Ces écarts importants pourraient être expliqués par le fait que le graphe estimé n'étant pas totalement exact, les arcs de coopération qui sont, on le rappelle, transformés ensuite en probabilité de choix de politiques conduisent alors à un manque de coopération complète qui impacte ensuite la réaction des agents, qui ne veulent pas coopérer pleinement, ce qui conduit à une baisse du bien commun.

5.6.5 Limitations

Un des challenges, voire le plus difficile, est la détection de la coopération, d'une part la détection des comportements des autres joueurs, d'autre part le graphe du potentiel de coopération maximale. Comme le montrent nos évaluations, notre agent peine à détecter correctement la coopération, dû en particulier très probablement au fort biais des agents de RL. Par ailleurs, cette détection nécessite une observation totale des états et surtout des actions des autres joueurs, afin notamment d'estimer le potentiel de coopération et le comportement des autres joueurs. Ce n'est malheureusement pas une hypothèse réaliste dans la plupart des situations. Quelques perspectives sont discutées en conclusion.

Une autre limitation réside dans la génération de politiques de RL en *self-play*. Comme il est

4. Simulation avec la détection du potentiel de coopération dans un environnement dynamique : `https://youtu.be/2Q48lRoJUwo`

nécessaire de pré-entraîner des politiques de différents degrés et combinaisons de coopération vers chacun des autres joueurs, le coût en calcul augmente exponentiellement avec le nombre de joueurs. En effet, même si en pratique, on peut entraîner uniquement des politiques de degrés binaires comme par exemple coopérer ou non avec tel ou tel joueur, cela correspond tout de même à 2^{N-1} politiques différentes à entraîner. Dans la section suivante, nous discutons de possibles améliorations, notamment sur la possibilité d'inclure les degrés de coopération dans le modèle (neuronal par exemple) des politiques pré-entraînées.

5.7 Conclusion et perspectives

L'objectif de ce chapitre a été l'extension de Dilemmes Sociaux Séquentiels (SSD) dans les cas où la coopération réciproque n'est pas systématiquement possible ou du moins pas nécessairement optimale. Nous avons donc proposé un formalisme qui repose sur les définitions des SSD [LZL+17] et qui reprend l'esprit de coopération circulaire du *Graph-based Iterated Prisoner's Dilemma* (GIPD) détaillé dans le chapitre 4. Les principales contributions sont :

— l'introduction d'un formalisme pour modéliser un SSD dont la structure de coopération possible peut prendre n'importe quelle forme. Notre extension CSSD est un jeu de Markov dont le potentiel de coopération entre les joueurs peut être modélisé sous la forme d'un graphe orienté et pondéré, permettant ainsi des coopérations mutuelles asymétriques et en particulier circulaires via un ou plusieurs autres acteurs.

— l'implémentation de bancs de tests pour expérimentation sous la forme de jeux dont la structure de coopération est facilement modifiable. Le graphe du potentiel de coopération maximale est en effet adaptable pour obtenir selon les besoins, des structures de coopération bilatérales ou circulaires.

— une première approche pour adresser ces jeux. Il s'agit d'une méthode hybride à base de TFT et de politiques de RL pré-entraînées par *self-play*. C'est une extension de l'approche proposée par [LP17] mais qui au lieu d'utiliser un TFT classique, utilise notre *Graph-based Tit-for-Tat* (GTFT) développé dans le chapitre 4. Cela permet à la différence de l'ancienne approche, d'adresser les situations non-réciproques et circulaires.

Compte tenu de la nouveauté de ce modèle de jeu, il existe peu d'approches existantes pour comparer notre première proposition d'agent. Néanmoins, lorsque l'on compare notre approche à l'approche originelle [LP17] ou bien à des politiques égoïstes entraînées indépendemment, les métriques sociales forment un compromis bien plus optimal dans toutes les situations (et surtout dans les situations non réciproques puisque le TFT classique utilisé dans [LP17] ne suffit pas à adresser la circularité de la coopération). En revanche, notre agent peine à détecter correctement la coopération, en particulier la détection du graphe de coopération entre les joueurs. Il s'agit probablement du plus grand challenge relatif au problème étudié. En effet, détecter le comportement

via la simple observation des actions des autres joueurs et de l'état de l'environnement est extrêmement instable. Une raison probable est que cette détection repose sur les politiques pré-entraînées de différents niveaux de coopération. Il y a là un fort risque de biais dû à l'entraînement et l'utilisation de celles-ci. Il y a un risque que les politiques ne soient pas assez explicites, au sens où elles ont peut-être ignoré lors de l'entraînement des stratégies équivalentes. Par conséquent, des joueurs ne suivant pas les stratégies connues pour être coopératives mais pour autant équivalentes, seront alors considérés à tort comme non-coopératifs, et donc subiront une défection en vertu du TFT. Pour résumer, le challenge est très important sur le seul aspect de la détection de la coopération. Il existe quelques propositions d'améliorations consistant à dérouler (*rollout*) des politiques sur plusieurs itérations, et plusieurs exécutions pour mieux estimer le choix coopératif de tel joueur qui joue telle action en observant tel état. Cependant, cela peut être très coûteux en calcul lors de la phase d'exécution, ce qui pourrait être une forte limitation pour une utilisation en temps réel.

En ce qui concerne le fait qu'il est nécessaire d'accéder à l'observation totale de l'environnement, il pourrait être intéressant d'étudier des techniques d'apprentissage par consensus [DCH+04, ZZ19] et envisager une possibilité pour les agents de communiquer et partager leur connaissance de l'environnement entre eux lorsqu'ils en ont l'occasion (*e.g.* s'ils se croisent), par exemple sur leur estimation de la coopération des autres joueurs. Ce partage de modèle par consensus revient au problème de l'optimisation par consensus lorsque les acteurs ne sont pas coopératifs [ZPDW18, TNWL20]. Le passage à l'observation partielle représente également un défi important avec beaucoup d'enjeux puisqu'il s'agit en général d'une hypothèse très courante dans les situations rencontrées.

Enfin, concernant la génération de politiques pré-entraînées par *self-play*, il a été évoqué plus haut le coût important en calcul qu'impliquait l'explosion combinatoire des politiques de degrés différents vers les autres joueurs (même le cas "simple" avec degrés binaires : 2^{N-1} politiques différentes à entraîner avec N joueurs). Comme suggéré plus haut, une idée pourrait être d'inclure une sorte d'*embedding* de comportement dans la représentation de l'état pour modéliser les politiques dans un unique modèle [AMCB21]. Cela permettrait également de considérer des politiques aux degrés continus et ainsi s'affranchir de l'étape 4 de notre algorithme (section 5.5.2) qui permet par tirage de distribution de convertir des degrés continus ou degrés discrets.

Pour résumer, notre proposition d'agent comporte encore de nombreux challenges, et suscite de nombreux enjeux et de perspectives. L'approche hybride à base de TFT est une méthode sûre et incitative, elle est donc très prometteuse car elle évite l'écueil des équilibres de Nash non-optimaux. Enfin, pour ce qui concerne le formalisme générique de dilemmes sociaux séquentiels à $N \geq 2$ joueurs, il représente un intérêt important notamment pour l'étude d'agents décisionnels

dans des situations non-coopératives. Ceci a le potentiel de contribuer à répondre à des enjeux croissants tels que ceux liés à l'économie de ressources et d'énergie par la coopération d'acteurs intelligents.

Partie III

Vers une collaboration entre opérateurs de Télécoms

Chapitre 6

Un framework multi-agents pour la modélisation de la collaboration multi-opérateurs

Dans ce chapitre, nous nous concentrons sur l'implémentation d'un framework qui génère des jeux multi-opérateurs afin d'étudier leur collaboration. Ce framework permet de créer plusieurs environnements flexibles et personnalisables en utilisant le langage Python avec le formalisme OpenAI Gym. Ce dernier est utilisé notamment pour entraîner et étudier l'apprentissage des agents tels que des politiques d'apprentissage par renforcement (RL). L'objectif n'est pas de construire des environnements exhaustifs et réalistes mais d'obtenir quelques jeux multi-opérateurs clés sur lesquels nous pourrons étudier et évaluer les algorithmes développés précédemment.

6.1 Introduction et motivations

Aujourd'hui, de plus en plus de connectivité est nécessaire pour répondre aux besoins dus à l'explosion des nombreux usages d'Internet (streaming, Internet des objets, véhicules autonomes, télémedecine, etc). Pour les opérateurs, un challenge majeur de la connectivité est la gestion de cette augmentation de trafic. Étant donné que le déploiement de nouvelles infrastructures a un coût financier et environnemental non négligeable, une solution intéressante serait pour les opérateurs de Télécom (*Mobile Network Operator* (MNO)) de partager les antennes entre eux de manière à homogénéiser la connectivité de leurs utilisateurs. Par exemple en utilisant des procédés de *roaming* ou de *Radio Access Network* (RAN) *sharing* ou bien de *Radio access network slice* [Tur20]. Plus récemment, les places de marché de connectivité ont été considérées impliquant des transactions financières comme des mécanismes d'enchères [ZLNW12, CBA+19]. Dans les réseaux cellulaires mobiles, deux configurations principales peuvent être considérées : l'extension

de couverture réseau et l'extension de capacité. Dans le premier cas, quelques utilisateurs peuvent ne pas avoir de connectivité dû à un manque de couverture de leur opérateur. Dans le second cas, il peut y avoir trop d'utilisateurs par rapport à la capacité de la cellule de leur opérateur. Pour palier ces problèmes, un opérateur peut vouloir coopérer avec d'autres opérateurs pour bénéficier de ses ressources si la cellule correspondante est moins chargée ou si l'infrastructure est géographiquement plus proche des utilisateurs demandeurs [TR20]. Dans ces situations (Figure 6.1), les opérateurs peuvent envisager des scénarios basés sur la réciprocité et les échanges équitables de manière à ce que chaque participant tire un bénéfice de la coopération.

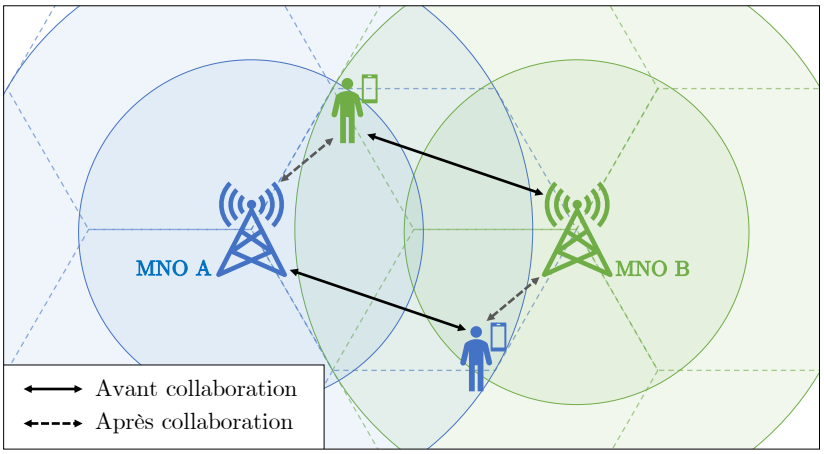


FIGURE 6.1 – Principe et exemple d'une coopération multi-MNOs : par service mutuel, les MNOs prennent en charge des utilisateurs d'autres opérateurs. Cela a pour effet de diminuer la charge de la cellule et ainsi d'optimiser la connectivité générale.

Pour un opérateur, afin d'évaluer et identifier la participation dans de tels scénarios de coopération, l'apprentissage multi-agents offre des perspectives intéressantes pour déterminer les interactions optimales entre les agents. En particulier, l'émergence récente du MARL dans des jeux non-coopératifs est très prometteuse [LP17, PLZ+17, LWT+17] comme l'utilisation des algorithmes à base de Tit-for-Tat (TFT). Cependant, l'étude de ces agents et l'évaluation de ces stratégies requiert un environnement de simulation et de test. Bien qu'il existe de nombreux environnements de simulation de réseau mobile [BTA+99, IH09], à notre connaissance, aucun ne propose un cadre de collaboration multi-opérateurs. Pour pallier le manque d'un tel environnement, nous proposons un nouveau framework capable de générer des environnements RL pour la simulation de coopération multi-opérateurs. Nous avons décidé d'adopter le formalisme Gym de OpenAI [BCP+16] qui est une référence en matière d'étude d'agents.

Pour commencer, nous expliquons quelques notions et éléments de vocabulaire liés aux réseaux de téléphonie mobile, et rappelons quelques notions liées aux jeux. Nous détaillons ensuite les

caractéristiques de notre framework dans la section 6.4 et proposons quelques exemples dans la section 6.5. Nous conclurons par une brève étude de théorie des jeux (dans la section 6.6), notamment pour illustrer les situations de dilemme social dans lesquelles ces jeux multi-opérateurs se trouvent.

6.2 Notions et vocabulaire liés aux réseaux mobiles

Avant d'expliciter en détail l'implémentation et les caractéristiques des nos environnements de simulation, il convient d'apporter quelques définitions simplifiées liées au contexte des opérateurs de réseau mobile.

Les *Mobile Network Operator*s

Un *Mobile Network Operator* (MNO) est ce qu'on appelle communément un opérateur de télécommunication qui propose à des clients un service de téléphonie mobile et/ou un accès mobile à Internet. Les MNOs détiennent des licences d'utilisation de fréquence et déploient leur propre réseau mobile via leurs propres infrastructures (antennes, etc.). Ces MNOs sont aussi appelés opérateurs classiques. La France en compte quatre : Bouygues Télécom, Free Mobile, Orange et SFR. Les opérateurs classiques sont à différencier des opérateurs virtuels ou en anglais *Mobile Virtual Network Operator* (MVNO). Les MVNOs sont des opérateurs ne possédant pas de licence d'utilisation de fréquence, ni d'infrastructures de réseau propres. Cependant, ils délivrent un service de connectivité à des clients par le biais de contrats avec des MNOs qui leur concèdent une partie de leur réseau. NRJ Mobile ou La Poste Mobile sont des exemples de MVNO français utilisant le réseau des MNOs français.

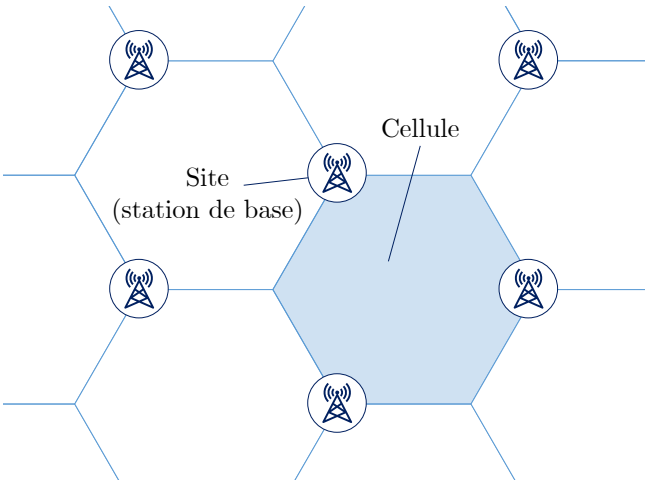


FIGURE 6.2 – Sites et cellules dans un maillage hexagonal pour la couverture mobile

Les stations de base et cellules

Chaque MNO possède un ensemble de stations de base. Une station de base (ou antenne-relai), est une infrastructure qui permet de délivrer un service de connectivité à des terminaux (téléphones mobiles) par l'envoi et la réception de signaux radio-électriques. Notons que chaque station de base comporte généralement trois antennes émettant dans un secteur de 120 degrés. L'ensemble de ces secteurs issus des différentes stations de base forme un maillage hexagonal et la zone délimitée

par trois stations de base, pour une fréquence donnée, est appelée cellule (Figure 6.2). Dans la suite, on utilisera le terme cellule pour désigner chacun des trois secteurs par antenne. Pour conserver l'idée du maillage hexagonal, nous fixerons ce nombre à 3. Notons que ce nombre peut être supérieur, voir nettement supérieur. En effet, la prochaine génération de réseau mobile (5G) pourra accueillir au moins jusqu'à 256 antennes pour cibler précisément les utilisateurs [Ser17].

Les terminaux utilisateurs

Les utilisateurs (ou *User Equipments*) sont les téléphones mobiles des clients respectifs à qui les MNOs délivrent le service de connectivité. Dans notre configuration, on considère que chaque utilisateur est associé à un unique MNO de rattachement, une position fixe ou mobile (voir section 6.4.1) connue ou non de l'opérateur selon la configuration de l'environnement. Enfin, dans une première estimation, notre environnement lie initialement chaque utilisateur à la station de base de son MNO de rattachement qui est la plus proche géographiquement. Les zones distinctes, regroupant les mobiles qui partagent la même station de base, forment une partition du territoire qu'on appelle diagramme de Voronoï [PA08] (voir Figure 6.3).

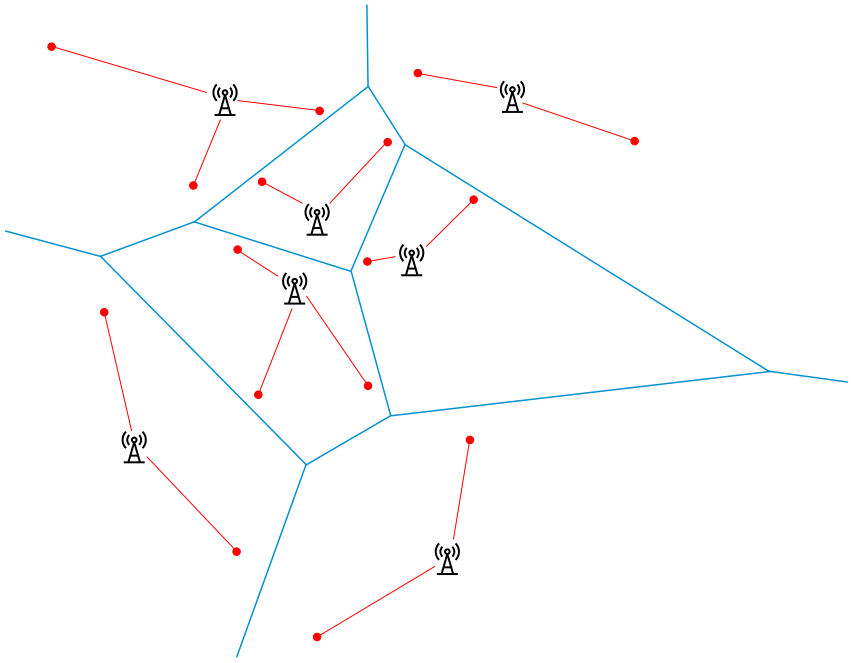


FIGURE 6.3 – Partition (ou diagramme) de Voronoï pour paver la couverture mobile et relier les utilisateurs (points rouges) à une station de base

Les ressources radio

Dans notre point de vue, nous faisons l'hypothèse que chaque MNO possède pour chaque antenne une quantité de ressources disponibles type *Physical Ressource Block* (PRB) [1]. Il s'agit d'une part de fréquence pendant une portion de temps. C'est sur ce type de quantité que les MNOs se basent pour s'échanger des parts de ressources radio pour une cellule donnée.

Le *Roaming* et le *RAN sharing*

Le *roaming* est le fait qu'un utilisateur (*e.g.* en voyage) puisse se connecter via le réseau d'un autre opérateur. Le RAN *sharing* est une catégorie de partage de réseau d'accès qui n'utilise qu'une partie de l'infrastructure d'accès d'un opérateur mais pas son coeur de réseau (en quelque sorte le logiciel de gestion).

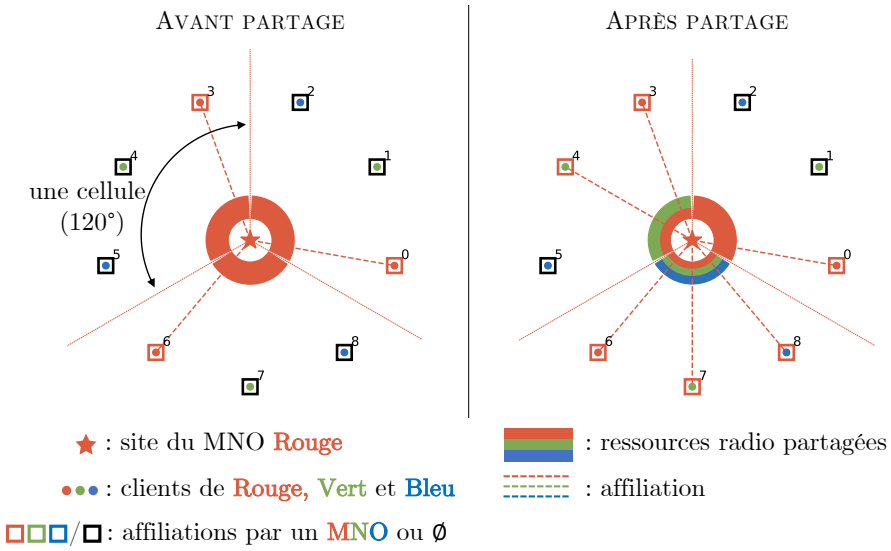


Figure 6.4 – Exemple de partage de ressources radio. Le jeu présenté est simple et comporte trois opérateurs (Rouge, Vert, Bleu). Il y a une seule station de base (gérée par Rouge) et neuf utilisateurs (trois clients par opérateur). La couleur des points identifie le MNO (*home*) dont est client l'utilisateur en question. La couleur des rectangles entourant les utilisateurs indique quel opérateur les prend en charge (*visited* MNO, c-a-d le MNO auquel appartient l'infrastructure qui délivre la ressource radio)

Dans l'exemple de la Figure 6.4, l'opérateur *Rouge* dans la situation de gauche, ne partage pas son connectivité, les clients des autres opérateurs ne peuvent donc se rattacher à son antenne. Dans la situation de droite, *Rouge* alloue une partie de son connectivité à *Vert* pour une de ses antennes (celle de gauche) et à *Bleu* et *Vert* pour son antenne du bas.

1. `https://www.sciencedirect.com/topics/computer-science/physical-resource-block`

6.3 Notions de jeux et environnements OpenAI Gym

Dans cette section, nous rappelons les éléments nécessaires pour définir un jeu (Processus de décision markovien partiellement observable : POMDP) puis nous détaillons le concept des environnements OpenAI Gym : un format utile pour implémenter des jeux.

6.3.1 Rappels des processus de décision markoviens POMDP

Dans la littérature, les environnements multi-agents sont communément formalisés par les *Partially Observable Markov Decision Processes* (POMDP) qui sont une généralisation des *Markov Decision Process* (MDP) [Bel57]. Les POMDP sont généralement définis par un sextuplet (S, A, T, R, Ω, O) où :

— S est un ensemble d'états

— A un ensemble d'actions

— $T : S \times A \to [0, 1]$ est une fonction (stochastique) de transition

— $R : S \times A \to \mathbb{R}$ est une fonction de récompense

— Ω est un ensemble d'observations

— $O : S \times A \times \Omega \to [0, 1]$ est une fonction d'observation

Pour rappel (voir chapitre 2 et 5), l'objectif dans un POMDP est de déterminer une politique optimale $\pi : s \mapsto a$ de sorte que la récompense cumulée soit maximisée.

6.3.2 Environnements au format Gym de OpenAI

OpenAI Gym est un framework [BCP+16] dont le rôle principal est de permettre une interaction dans un format générique entre un ou des agent(s) (en particulier de type RL) et des environnements créés manuellement. Ainsi, Gym est adapté pour entraîner, évaluer et comparer des politiques dans des environnements donnés.

Un environnement Gym noté `env` en version mono-agent est une instance d'une classe héritière de la principale classe fournie par la bibliothèque Gym. Il comporte quatre fonctions principales :

1. `env.init()` : l'environnement est instancié et établi dans son état initial (qui peut être choisi manuellement ou aléatoirement)
2. `env.reset()` : une fois que l'environnement a déjà été instancié à l'aide de `env.init()`, la fonction `env.reset()` permet alors de réinitialiser l'environnement.

3. `env.step(action)` → `[new_state, reward, done, info]` : est la fonction la plus importante puisqu'elle permet l'exécution d'une action afin de modifier l'état de l'environnement et de recevoir une récompense. Ainsi, l'agent injecte l'action `action`, et il peut alors recevoir le nouvel état de l'environnement `new_state`, une récompense `reward`, un booléen `done` qui indique si le jeu est terminé et enfin de manière optionnelle, quelques informations supplémentaires `info`.

4. `env.render()` : une fonction de rendu visuel qui permet d'afficher l'environnement dans son état actuel. Bien que cette fonction puisse être considérée comme facultative, elle a fait néanmoins l'objet d'un intérêt et effort particulier dans notre application.

Le formalisme en version multi-agents est similaire au formalisme à un agent : les variables adaptées sont transformées en liste. Ainsi, `action` devient une liste `actions` constituée des N actions des N joueurs. De même pour `new_states`, `rewards`, etc.

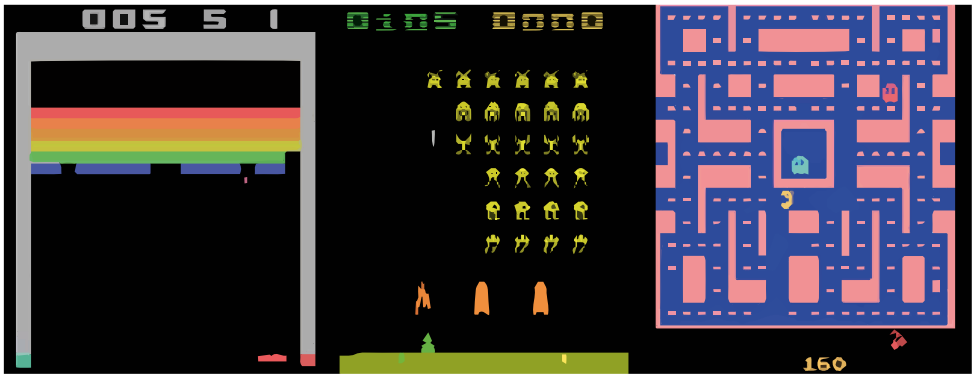


FIGURE 6.5 – De nombreux jeux Atari ont été développés en Gym afin de fournir des bancs de tests aux algorithmes de RL [MKS+13].

6.4 Proposition d'un environnement multi-opérateurs

Après avoir introduit les notions liées au domaine des télécommunications et rappelé les caractéristiques nécessaires à la définition de jeux, nous allons dans cette section détailler l'implémentation de nos environnements [2].

Pour être intégré comme un environnement OpenAI Gym et ainsi profiter de la flexibilité associée, il est nécessaire d'implémenter les quatre fonctions mentionnées plus haut. En particulier, la fonction `env.step(action)` qui fait l'intérêt de l'environnement. Dans un premier temps, il

2. Le code source de notre framework est disponible sur GitHub : `https://github.com/tlgleo/gym-MNOs-cooperation`

convient de détailler les caractéristiques et paramètres principaux de notre framework [LGMLR21].

6.4.1 Caractéristiques de l'environnement

Notre framework implémenté permet de créer des environnements Gym. Nous listons dans la suite les paramètres qui sont modifiables ou non pour la génération de tels environnements.

Nombre d'agents

Un nombre `n_agents` d'agents (ou de joueurs) peut être défini. Dans notre cas, les joueurs correspondent aux MNOs (opérateurs). Nous nous intéressons à la notion de joueur au sens de la théorie des jeux, à savoir des acteurs qui cherchent à maximiser une fonction d'utilité personnelle. Dans nos simulations, il sera généralement égal au plus à 4 (nombre de MNO français).

Positions des stations de base

Chaque joueur (MNO) possède des stations de base (ou sites) qui sont contenues dans le paramètre `positions_sites`. Ce dernier est une liste de `n_agents` listes qui contiennent chacune les positions aux coordonnées (x, y) des stations de base. Dans les situations réelles que l'on aborde dans la suite, les coordonnées des stations de base sont extraites des données de l'Agence Nationale des Fréquences (ANFR), elles correspondent aux coordonnées géographiques (longitude, latitude). Dans le cas de données réelles, il convient de faire la distinction (x, y) et (longitude/latitude) notamment dans le calcul de la distance la plus courte qui intervient dans les transitions. En effet, on ne peut pas faire l'abus de considérer (x, y) par (longitude/latitude), en particulier sur le territoire français où la latitude est non négligeable : $\cos(48°) \approx 0.67...$ On considère par ailleurs que ces positions sont fixes. Cependant, la modification de ses positions au cours du temps pourrait être envisageable comme dans les perspectives d'antennes relais par drone [MSBD16, SH18].

Nombre de cellules ou secteurs

Comme évoqué plus haut, chaque station de base comporte des antennes qui peuvent émettre pour une bande de fréquence donnée dans trois directions différentes. Ceci forme des secteurs de 120 degrés que l'on appelle communément des cellules. Bien qu'il soit très courant que ce nombre soit fixé à trois (maillage hexagonal), les nouvelles générations de réseau mobile (5G) ont notamment pour objectif d'augmenter significativement ce nombre afin de cibler plus précisément la direction du signal. C'est pourquoi dans nos environnements, il est possible de définir le nombre de secteurs avec le paramètre `n_cells`.

Positions des terminaux utilisateurs

Pour simuler des situations et estimer les qualités de services, des terminaux utilisateurs (les mobiles des clients) sont placés dans l'environnement. Ils peuvent être considérés fixes ou mobiles (voir section 6.4.1). Dans les deux cas, leur nombre et leur position initiale doivent être définis. Les positions (x, y) ou (longitude/latitude) des terminaux sont indiquées dans la liste `positions_users`. Pour la simplification, une fonction `random_users` est implémentée pour définir aléatoirement des positions dans la zone du jeu. Cette fonction se contente en l'état de placer aléatoirement des positions sur une zone rectangulaire sans prendre en compte la densité éventuelle, ou la topologie de la zone étudiée (par exemple le littoral, voir Figure 6.9). Enfin, chaque utilisateur se voit attribuer une affiliation fixe à un unique opérateur dont il est le client. Ces affiliations sont définies par un indice d'un unique MNO contenu dans une liste `clients`.

Mobilité des terminaux utilisateurs

Les utilisateurs peuvent être considérés fixes ou mobiles. Pour ce dernier cas, une classe `Kinematics` a été implémentée. Elle permet d'induire un mouvement aux utilisateurs [3].

Discrétisation des observations

Dans le chapitre 2, nous avons évoqué le fait que certains algorithmes nécessitent une représentation vectorielle (ou tensorielle) des données et des observations. Ainsi, une option possible de l'environnement permet d'encoder les observations sous la forme d'un *gridworld* multi-couches. Cette caractéristique est notamment adaptée à des algorithmes de type *Deep Reinforcement Learning* (DRL) via l'utilisation en particulier de *Convolutionnal Neural Network* (CNN) (Voir aussi 6.4.2)

6.4.2 Caractéristiques liées au framework Gym

Dans cette section, nous décrivons des caractéristiques des POMDP (section 6.3.1) de notre framework Gym. On détaille en particulier les actions, c'est-à-dire les transactions des joueurs (opérateurs), les observations, la fonction de transition et enfin la fonction de récompense qui correspond à l'utilité de chaque joueur qui doit être maximisée.

3. Un exemple de mobilité pour l'environnement nommé `env_3A_5S_30U-v0` peut être visionné ici : `https://www.youtube.com/watch?v=ZfCgvPOmUoc`

Actions

A chaque étape du jeu, les joueurs (opérateurs) sont invités à allouer (ou non) une portion de leurs ressources aux autres joueurs sur l'une ou plusieurs de leurs cellules. Ainsi, une action consiste à lister une partition des ressources pour chacune des cellules de chaque station de base. Formellement, une action a pour un joueur donné est un tenseur (table `numpy` en Python) de dimension 3 et de taille `(n_sites, n_cells, n_agents)`. Nous avons la condition suivante de partition de ressources (en proportion) pour chaque cellule :

$$\forall i_S,\ \forall k_C,\ \sum_{i_P=1}^{N} a[i_S, k_C, i_P] = 1 \tag{6.1}$$

avec N le nombre de joueurs (MNO) et i_S, k_C, i_P respectivement les indices des sites, cellules et joueurs.

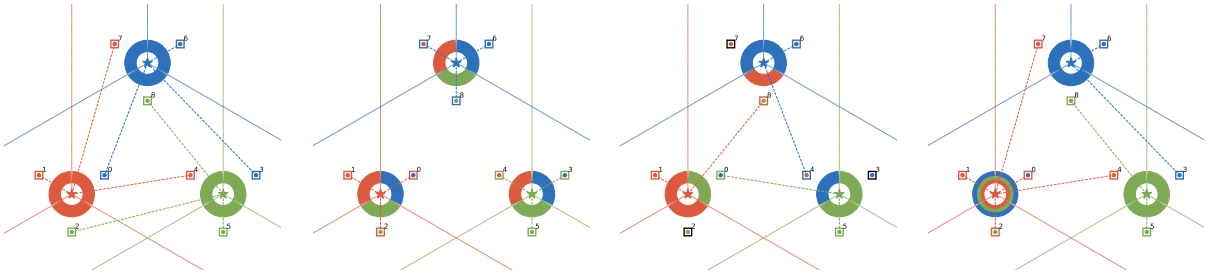
Par exemple, la Figure 6.6 montre un environnement (nommé `env_3A_3S_9U-v2`) à trois joueurs (MNO) qui fait partie des jeux prédéfinis directement disponibles (voir 6.5.1). Dans ce jeu, chaque joueur possède une station de base et couvre chacun trois clients. Sur la figure, sont représentées quatre versions du jeu. Les quatre différentes actions choisies par le premier agent (le Rouge) dans chaque version sont respectivement :

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

avec l'indice de la cellule qui s'incrémente dans le sens trigonométrique (anti-horaire). L'indice 0 correspondant au secteur incluant l'angle 0.

Observations

La fonction d'observation permettant de convertir l'environnement en un état que les agents observent est sujet à discussion. Un environnement multi-agents à observation partielle est considéré comme non-markovien ce qui nécessite alors de créer un état comportant tout l'historique des actions et des états. De telles configurations pour l'état ont été étudiées au sein des algorithmes proposés dans [HS15, LM92]. C'est pourquoi les environnements ne fournissent que l'observation actuelle, le travail d'historique est effectué au sein de ces agents. Nous proposons alors un état disponible en deux versions : une observation totale ou une partielle. L'observation fournit une carte de type *bitmap* montrant la position des sites et des utilisateurs sur des canaux différents. De tels tenseurs sont très adaptés notamment pour l'étude d'algorithmes de type Deep RL qui



(a) Situation initiale (b) Situation optimale (c) Situation non optimale (d) Rouge est seul à coopérer

FIGURE 6.6 – Quatre instances du jeu simple nommé `env_3A_3S_9U` avec 3 joueurs (Rouge/Vert/Bleu) possédant chacun un site et trois utilisateurs. La Figure 6.6b illustre la coopération optimale. La Figure 6.6d montre une situation où seul un joueur (Rouge) accepte de coopérer.

utilisent des réseaux de neurones convolutionnels [MKS+15, MKS+13]. Dans la Figure 6.7, les deux types d'observation sont représentés pour le jeu prédéfini `env_3A_3S_9U-v1`.

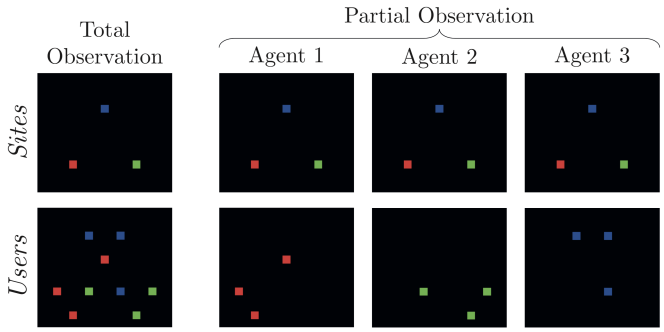


FIGURE 6.7 – Observation totale et partielle du jeu prédéfini `env_3A_3S_9U-v1` (correspondant à la Figure 6.6)

Fonction de transition

Dans nos environnements, il est nécessaire de développer une fonction de transition. Il s'agit de définir la manière dont l'état de l'environnement est transformé, selon les actions effectuées par les agents. Nous proposons qu'à chaque itération du jeu, les liaisons entre antennes et utilisateurs soient redéfinies selon les parts de ressources que les joueurs se partagent entre eux. Pour déterminer le lien, nous proposons une manière simple : chaque utilisateur est relié à l'antenne la plus proche qui l'accepte. En d'autres termes, soit l'antenne appartient à l'opérateur original du client, soit cet opérateur s'est vu allouer suffisamment de part pour la cellule correspondante. Notons que bien que cette allocation n'est pas optimale, elle est suffisamment réaliste pour modéliser un jeu de collaboration. De plus, une solution plus optimale aurait nécessité à chaque

itération une optimisation NP-difficile et dont les améliorations ne sont pas l'objet du chapitre. La Figure 6.6 illustre plusieurs liens dans le même jeu avec différentes allocations.

Fonction de récompense

Pour modéliser la récompense de notre environnement Gym, nous proposons une fonction d'utilité simplifiée pour chaque joueur (opérateur) qui dépend des allocations de ressources et de la position des utilisateurs personnels (clients). Il s'agit d'une expression qui pour chaque joueur P somme une qualité de service q_{u_P} que reçoit chaque client u_P de l'opérateur P. Cette qualité de service doit diminuer fortement avec la distance à laquelle se trouve le client de sa station de base de rattachement. Elle augmente avec la quantité de ressources radio qui a été allouée à la cellule dans laquelle il se trouve. Enfin la qualité diminue proportionnellement avec le nombre de terminaux qui se partagent la cellule. En s'inspirant des travaux de [KHH+12], il est proposé une version simplifiée de q_{u_P} comme suit :

$$q_{u_P} = \frac{\Gamma_{u_P}}{m_{u_P}} \exp(-\frac{D_{u_P}}{D_{max}}) \tag{6.2}$$

où :

— Γ_{u_P} est la portion de ressources radio allouée pour l'opérateur P dans la cellule de u_P, qu'elle soit détenue (antenne propre) ou reçue (allocation d'une portion de ressource radio par un autre opérateur).

— m_{u_P} est le nombre de clients de l'opérateur P qui se trouvent dans la cellule du client u_P et donc qui se partagent la portion correspondante

— D_{u_P} est la distance à laquelle se trouve u_P de son antenne de rattachement, D_{max} est une constante de normalisation/homogénéisation

Enfin, pour la récompense totale du joueur (opérateur), les qualités de service des clients sont sommées. Dans ce cas, nous n'appliquons aucune normalisation car l'objectif des opérateurs est de maximiser le nombre d'utilisateurs ayant une bonne qualité de service et non pas maximiser la qualité de service moyenne par utilisateur. On obtient donc l'expression finale de la récompense R_P de l'opérateur P :

$$R_P = \sum_{u_P \in \mathcal{C}_P} q_{u_P} \tag{6.3}$$

avec :

— $\mathcal{C}_P$ l'ensemble des clients de l'opérateur P

— q_{u_P} la qualité de service simplifiée de l'utilisateur u_P (équation 6.2)

Remarquons qu'une telle expression de l'utilité peut servir de fonction de coût pour optimiser le rattachement des utilisateurs à la meilleure antenne. En effet, la formule prend en compte le

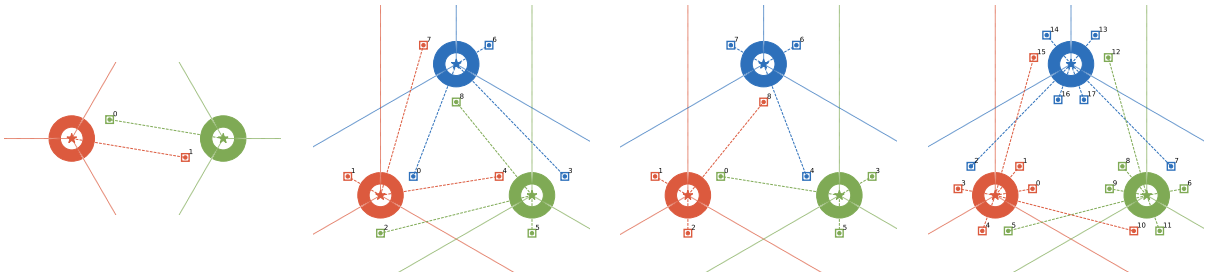
nombre d'utilisateurs se partageant la cellule en plus de la distance à laquelle ils se trouvent. Par conséquent, il peut exister un agencement plus optimal qui n'est pas l'antenne la plus proche. Notons que l'objectif de ces environnements n'est pas d'être le plus proche du réalisme mais plutôt d'implémenter une première version d'environnement de simulation pour étudier et évaluer les aspects coopératifs des opérateurs. Comme mentionné plus haut, pour des raisons de temps de calcul et dans un soucis de permettre des interactions rapides avec l'environnement, nous maintiendrons le rattachement à l'antenne la plus proche au sens des partitions de Voronoï.

6.5 Exemples d'applications

Notre framework propose quelques environnements prédéfinis très simples que nous présentons dans la section 6.5.1. Ces jeux ont pour objectif principal d'évaluer des premières versions d'agents. Il est également possible de modéliser des situations réelles du territoire français à l'aide des données de l'Agence Nationale des Fréquences (ANFR) qui met à disposition le type de fréquence et la position des antennes pour l'ensemble des quatre opérateurs français. Nous présentons cette extension en section 6.5.2.

6.5.1 Jeux prédéfinis simples

Notre framework propose quelques environnements pratiques très simples considérant 2 ou 3 joueurs (opérateurs). Par exemple, la Figure 6.8 illustre quatre situations simples et symétriques avec deux ou trois MNOs disposant chacun d'une seule antenne et quelques utilisateurs (clients) artificiellement répartis de manière à obtenir un jeu simple.



(a) `env_2A_2S_2U-v1` (b) `env_3A_3S_9U-v0` (c) `env_3A_3S_9U-v2` (d) `env_3A_3S_18U-v0`

FIGURE 6.8 – Quelques exemples de jeux prédéfinis à deux ou trois joueurs

Sans rentrer dans les détails, ces jeux ont été agencés de telle sorte qu'ils représentent des dilemmes sociaux. En particulier en considérant certains *Graph-based Iterated Prisoner's Dilemma* (GIPD) vus dans le chapitre 4 (voir la Figure 4.3) comme un dilemme du prisonnier classique à deux joueurs (Fig 6.8a) et trois jeux à trois joueurs avec une coopération réciproque (Fig 6.8b et

6.8d) ou circulaire (Fig 6.8c). Nous reviendrons brièvement sur ces aspects de théorie des jeux dans la section 6.6.

6.5.2 Jeux de situations réelles avec les données de l'ANFR

Il est intéressant de créer des mini-environnements modélisant de véritables situations sur des portions de territoires impliquant les quatre opérateurs français. L'emplacement des antennes des opérateurs étant publique, il est alors possible de renseigner la position des sites dans notre framework pour créer des jeux s'approchant de situations réelles.

Sélection des données

Le site de l'ANFR met à disposition[1] les coordonnées de latitude et longitude de chaque site avec antenne indiquant entre autres le nom de l'opérateur, la génération de l'antenne (2G/3G/4G) et sa bande de fréquence. Dans un contexte d'échange de ressources et de prise en charge de terminaux de concurrents, il est nécessaire de considérer des services équivalents puisqu'ils sont voués à être échangés de manière équitable. Pour ces raisons, on considère que ces services sont issus d'une même technologie, à savoir d'une même génération de téléphonie mobile (2G/3G/4G) et d'une même bande fréquence qui sont listées ci-dessous (en MHz) :

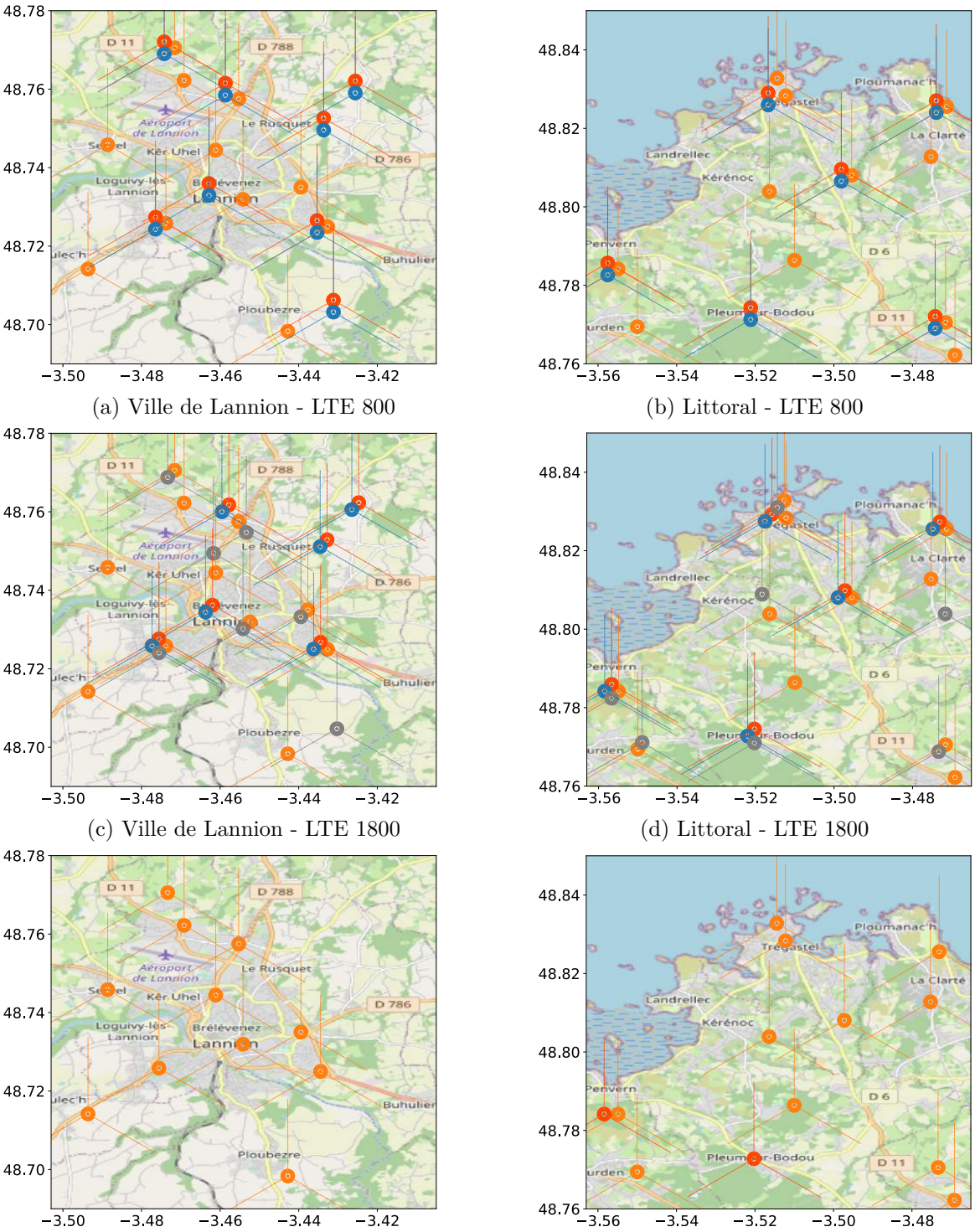
2e génération (2G)	3e génération (3G)	4e génération (4G)
— GSM 900	— UMTS 900	— LTE 700
— GSM 1800	— UMTS 2100	— LTE 800
		— LTE 1800
		— LTE 2100
		— LTE 2600

Notons que le principe de collaboration inter-opérateurs a vocation à être davantage utilisé pour la 5e génération de téléphonie mobile (5G) compte tenu de la multiplication d'antennes à plus faible portée. Cependant, dans les quelques exemples proposés, nous considérons uniquement la 4G qui est suffisamment déployée.

Sélection d'une zone géographique

Notre framework permet la sélection d'une zone rectangulaire par ses coordonnées géographiques (latitude, longitude). Les données sont directement extraites de celles de l'ANFR avec la technologie choisie. Nous proposons sur la Figure 6.9, deux zones géographiques avec trois bandes de fréquence du réseau 4G. Les couleurs bleue, grise, orange et rouge désignent respectivement Bouygues Télécom, Free Mobile, Orange et SFR. Notons que lorsque des antennes sont colocalisées (*i.e.* partagent le même support), les sites sont représentés légèrement décalés pour plus de lisibilité.

1. Les données sont téléchargeables en CSV ici : `https://data.anfr.fr/anfr/portail`



(a) Ville de Lannion - LTE 800

(b) Littoral - LTE 800

(c) Ville de Lannion - LTE 1800

(d) Littoral - LTE 1800

(e) Ville de Lannion - LTE 2100

(f) Littoral - LTE 2100

FIGURE 6.9 – Antennes du réseau 4G dans la région du Trégor : sur la ville de Lannion, et sur la côte de Granit Rose à droite avec les bandes de fréquences : 800 MHz, 1800 MHz et 2100 MHz.

6.6 Quelques aspects liés à la théorie des jeux

Bien que ce chapitre n'ait pour objectif principal que de présenter notre environnement de simulation de coopération entre opérateurs de télécommunications, il est néanmoins intéressant de considérer quelques aspects de théorie des jeux, notamment de relever le fait que le jeu est de type non-coopératif et à somme positive (voir chapitre 1). Par conséquent, il existe des stratégies gagnantes pour plusieurs joueurs. On pourra retrouver que l'on est en présence de jeux de type dilemme social.

6.6.1 Jeux simples

De manière générale, les joueurs dans un environnement multi-opérateurs n'ont pas d'intérêt à aider d'autres joueurs. En effet, rendre service est une stratégie dominée ce qui conduit au comportement rationnel mutuel qui est de rien faire (équilibre de Nash). Pour une meilleure intuition, considérons un jeu avec $N = 2$ opérateurs et qu'il y a deux actions possibles : coopération et défection. On peut représenter de manière rudimentaire le jeu sous la forme d'une table que l'on représente sur la Figure 6.10 et ainsi faire apparaître un dilemme social avec les quatre gains possibles.

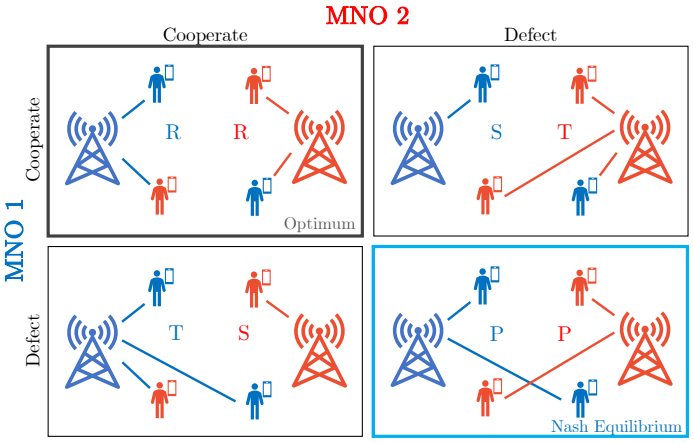
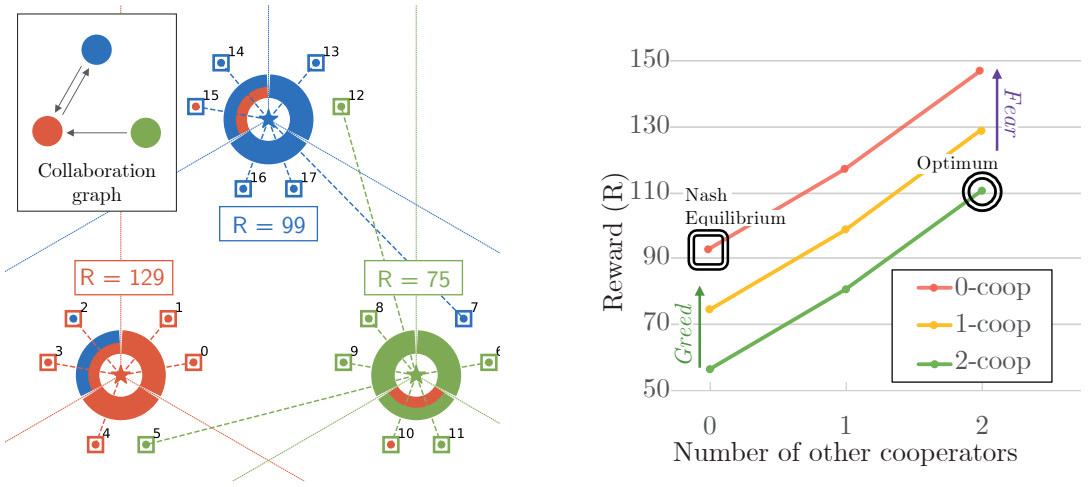


FIGURE 6.10 – Intuition du concept de dilemme à deux opérateurs de Télécom : un dilemme social itéré où les gains symétriques vérifient $S < P < R < T$ (voir chapitre 1). L'équilibre de Nash est la défection mutuelle tandis que l'optimal est la coopération mutuelle.

Si l'on considère les fonctions d'utilité (de récompense) vues en section 6.4.2, on peut se convaincre que l'issue optimale (optimum de Pareto) pour les deux opérateurs est celle où les deux prennent en charge mutuellement des clients du concurrent qui seraient les plus proches de leur antenne. Pour les jeux à plus de deux joueurs, il est possible de faire figurer les gains sous la forme d'un diagramme de Schelling [Sch73] (voir section 2.3.2). Nous représentons dans

la Figure 6.11, la situation du jeu nommé `env_3A_3S_18U-v0` à trois joueurs avec les utilités du point de vue du joueur *Rouge*. Selon le nombre d'autres opérateurs qui coopèrent avec *Rouge* et le nombre à qui il rend lui-même service, on peut faire apparaître les neuf gains. Notons que ce diagramme a du sens puisque la situation est symétrique, sinon il faudrait faire apparaître les intervalles et non des courbes.



(a) `env_3A_3S_18U-v0`

(b) Schelling Diagram

FIGURE 6.11 – Une situation de dilemme social à trois joueurs représenté par un diagramme de Schelling [LZL+17, Sch06] (section 2.3.2). Dans la situation précise : *Rouge* et *Bleu* coopèrent ensemble alors que *Vert* ne coopère qu'avec *Rouge*.

6.6.2 Jeux de situation réelle

Dans cette section, nous évoquons brièvement les capacités de coopération entre opérateurs compte tenu de l'agencement des antennes respectives sur une zone donnée.

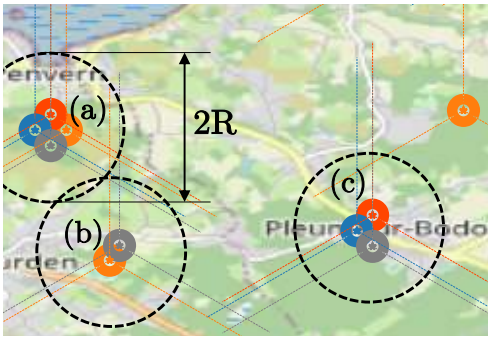


FIGURE 6.12 – Illustration du rayon de voisinage pour la densification avec le cas de la Fig 6.9d

Potentiel de coopération

Commençons par nous intéresser à la définition du potentiel de coopération entre les joueurs de sorte à obtenir un graphe de potentiel de coopération maximale, comme vu dans les chapitres 4 et 5. On peut estimer qu'un opérateur, pour une antenne donnée, a la possibilité de contribuer à l'augmentation de l'utilité d'un autre opérateur si lui-même ne possède pas d'antennes dans le proche voisinage de cette antenne. En effet, à l'intérieur d'un certain rayon de voisinage R, les distances des clients de l'autre opérateur

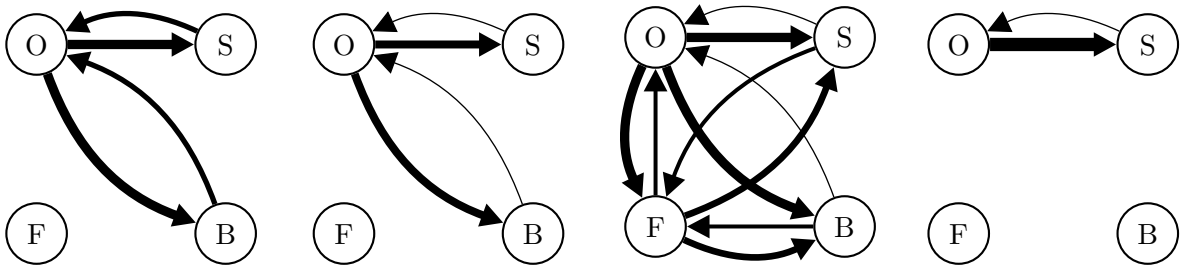
à leur antenne de raccordement sont déjà suffisamment faibles. L'intérêt est trop limité : on considère que la prise en charge d'un opérateur n'apporte pas de gain et donc qu'il n'y a pas de potentiel de coopération. Certaines études ont montré qu'un rayon de voisinage de $R = 500m$ est adapté [HIT13] à la situation de ce partage de ressource. Ainsi, dans l'exemple de la Figure 6.12, on peut voir que sur le site (a), les quatre opérateurs ont chacun un site colocalisé (donc dans le rayon de voisinage). Il n'y a donc pas de potentialité de coopération. Concernant le site (b), Orange et Free ne peuvent pas s'entraider. En revanche, ils pourraient alors prendre en charge les clients des deux autres opérateurs dans ce voisinage. Enfin, sur le site (c), Bouygues, Free et SFR peuvent rendre service à Orange puisque ce dernier n'a pas d'antennes dans le voisinage.

Graphes de potentiel de coopération

Compte tenu des considérations précédentes, il est possible d'établir un graphe de potentiel de coopération comme introduit dans les chapitres 4 et 5. Il s'agit d'un graphe orienté et pondéré comportant un nombre de nœuds égal au nombre de joueurs. Chaque arc (i, j) correspond à une capacité de coopération du joueur i vers le joueur j.

Dans notre situation, une estimation de cette capacité est, comme évoqué plus haut : le nombre de sites dont dispose l'opérateur i pour lequel il n'existe pas de sites de l'opérateur j dans un rayon fixé. On peut donc établir une matrice $C \in \mathcal{M}_{4,4}(\mathbb{N})$ qui sera la matrice d'adjacence des graphes de potentiel de coopération entre les quatre opérateurs.

Pour illustrer ce concept, reprenons les jeux (avec les cartes du Trégor) de la Figure 6.9. Selon le principe introduit plus haut, on peut alors générer les graphes de potentiel de coopération. Nous en représentons certains dans la Figure 6.13.



(a) Lannion LTE 800 (b) Littoral LTE 800 (c) Littoral LTE 1800 (d) Littoral LTE 2100

FIGURE 6.13 – Estimation des graphes de potentiel de coopération des situations présentées sur la Figure 6.9 avec un rayon de voisinage fixé à 500m. B, F, O et S désignent respectivement les opérateurs Bouygues, Free, Orange et SFR. L'épaisseur de chaque arc représente son poids.

Remarques : Lorsqu'un MNO (opérateur) n'a pas d'antennes sur la zone considérée, nous ne faisons pas apparaître les arc entrants pour cet opérateur, même si chaque autre joueur aurait la possibilité de lui rendre service. Par exemple, dans le cas de Lannion/Littoral en LTE 800 (Fig 6.9a et 6.9b), l'opérateur Free n'a pas d'infrastructures. Par conséquent, nous considérons dans les graphes 6.13a et 6.13b que la structure de coopération se restreint aux trois MNOs restants.

Sur ces graphes, on peut observer le caractère asymétrique des structures de coopération entre les joueurs. En ce qui concerne le dernier exemple (graphe 6.13d), il s'agit d'un jeu qui n'implique que SFR et Orange puisqu'ils sont les seuls opérateurs à disposer de sites en bande de fréquence 2100 Mhz sur le littoral trégorrois. Comme on peut le constater sur la Figure 6.9f, SFR (en rouge, couleur assez proche du orange...) ne dispose que de deux antennes dont une seule non colocalisée avec celles d'Orange. L'opérateur Orange, en revanche dispose sur la zone considérée de 10 antennes hors du voisinage de celles de SFR. C'est pourquoi, on peut observer l'asymétrie dans le graphe 6.13d. Notons que dans cette situation, un comportement équitable (de type donnant/donnant : Tit-for-Tat (TFT)) serait de coopérer sur une seule antenne, en vertu notamment de notre algorithme *Graph-based Tit-for-Tat* (GTFT) puisqu'alors le cycle optimal de coopération serait limité par l'arc de SFR vers Orange.

6.7 Conclusion et perspectives

Pour palier le manque d'environnement de simulation adapté à notre problème de coopération entre opérateurs de télécommunications, nous avons proposé un nouveau framework permettant de générer des environnements adaptables. Ces environnements sont générés au format OpenAI Gym, une API qui permet une interaction générique entre agents et environnement, en particulier pour simplifier certains apprentissages et évaluations d'agents (par exemple de type RL).

Notre implémentation permet de définir le nombre de joueurs/opérateurs ainsi que la position de leurs antennes. Elle permet également de simuler la présence des terminaux mobiles de leurs clients. Le rattachement des mobiles à leurs antennes a été implémenté dans une première approximation ainsi que l'utilité (récompense) qui est attribuée à chaque opérateur.

Il est possible dans un premier temps d'utiliser les jeux simples prédéfinis mis à disposition. Grâce aux données mises à disposition par l'ANFR, il est également possible d'instancier des jeux plus réalistes correspondant à n'importe quelle zone géographique donnée et à une technologie donnée (génération et bande de fréquence).

Nous avons également proposé une brève analyse de théorie des jeux, afin d'illustrer les

potentielles structures de coopération qui peuvent exister entre les opérateurs et mettre en avant le caractère de dilemme social qui peut survenir dans ces jeux.

L'implémentation de ces jeux avait pour but de créer une première version d'environnements de simulation de collaboration multi-opérateurs. Ils présentent de nombreuses perspectives d'amélioration. On peut par exemple envisager de prendre en compte les données de l'INSEE telles que la densité de population, voire la mobilité quotidienne de population. Il peut également être intéressant d'introduire les données de qualité de service pour chaque antenne issues des mesures internes et de les recaler selon la position des sites des autres opérateurs et en déduire ainsi un modèle plus sophistiqué prenant en compte les combinaisons d'allocation de ressource pour chaque site.

Dans le prochain chapitre, quelques expériences de partage de ressources seront menées, en particulier en utilisant quelques jeux de notre framework.

Chapitre 7

Collaborations inter-opérateurs : algorithmes et simulations

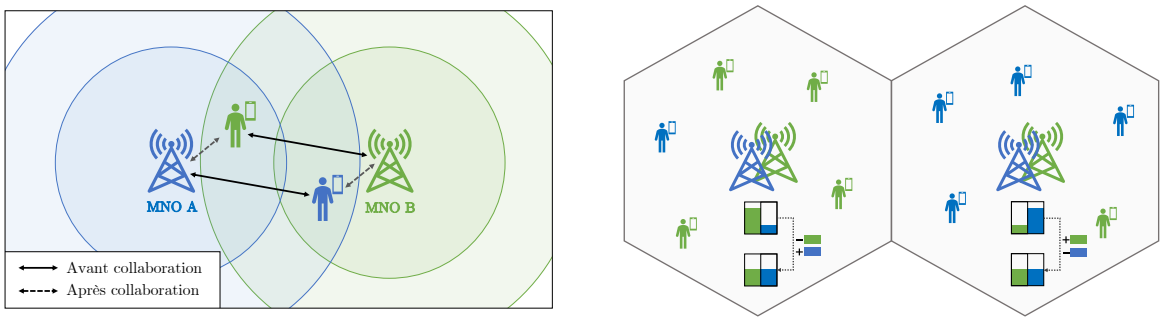
Dans ce chapitre, nous étudions les possibilités de mettre en application certains des algorithmes développés dans les chapitres précédents dans un cadre d'échange de services ou de ressources entre opérateurs de télécommunications.

7.1 Introduction

Partager des ressources ou des services entre plusieurs agents est très commun dans les situations industrielles. En particulier, dans les télécommunications, partager des ressources entre différents opérateurs a été suggéré, notamment avec l'arrivée de la nouvelle génération de réseau mobile (5G [1]), dans le but d'étendre et améliorer la capacité et la couverture de la connectivité de chaque opérateur. Un modèle bien adapté pour l'échange de ressources de connectivité est le framework appelé *Licensed Shared Access* (LSA) [BER14] qui vise à optimiser l'utilisation du spectre de fréquence. Dans la littérature, il y a plusieurs travaux qui abordent la question du partage de ressources radio de type *Licensed Shared Access* (LSA), en utilisant en général des mécanismes d'enchères [CBA+19, ZZ09, CZWZ13, WDF+15]. En particulier, certaines de ces approches étudient des mécanismes d'incitation (tels que les mécanismes de *Vickrey-Clarke-Groves* (VCG)) qui ont la particularité d'avoir de très bonnes propriétés "sociales" comme l'équité et l'*incentive-compatibility*. Cependant, ce type de mécanismes implique des transactions financières. Comme expliqué en introduction de cette thèse, nous nous intéressons à des situations de coopération à la fois sans intermédiaire et sans argent. L'objectif que nous nous fixons est d'échanger des services par le seul intérêt de voir son utilité globale augmenter. Par conséquent, nous visons des mécanismes qui incitent à la coopération sans prendre le risque d'une

1. https ://www.tmforum.org/blockchain-based-telecom-infrastructure-marketplace/

exploitation par un acteur égoïste. Dans la suite, nous allons nous concentrer sur les deux types de scénarios vus en introduction de la thèse, à savoir l'extension de couverture et l'extension de capacité (Figure 7.1).



(a) Extension de capacité (b) Extension de couverture

FIGURE 7.1 – Exemples de collaboration inter-opérateurs. Une intuition de jeux avec deux opérateurs (représentés en vert et en bleu). Ils sont concurrents mais peuvent faire face à des situations où une coopération mutuelle est optimale pour tous.

Nous supposons que les opérateurs peuvent échanger des ressources de manière à homogénéiser une certaine quantité de ressource par cellule et ainsi atteindre une issue optimale. On considère que les opérateurs sont intéressés uniquement par maximiser leur utilité (ce qui peut être considéré comme la qualité de service des utilisateurs qui ont souscrit un abonnement auprès d'eux). Nous travaillons sous trois hypothèses principales. Premièrement, pour des raisons stratégiques, les joueurs (les opérateurs) ne souhaitent pas partager la totalité de l'état de leurs ressources (on est donc dans un jeu avec une observation partielle). Deuxièmement, il n'y a pas de régulateur, un contrôleur tiers qui aurait une vue globale des ressources et serait à même de calculer les transactions optimales et de les imposer aux participants. Enfin, nous rappelons que le partage ne s'effectue pas par le biais de transactions financières pour des raisons d'architecture, de stratégie et de conviction. En outre, notre souhait est de rendre adaptable notre approche à d'autres paradigmes où la rétribution financière n'est pas fondamentale ou pas possible tels que l'échange de data, de ressources en *peer-to-peer*, etc.

Dans ce chapitre, nous décrivons les deux types de collaboration, puis nous présentons un algorithme à base d'une fonction de Tit-for-Tat (TFT) continu qui permet d'effectuer des transactions dans les jeux considérés. Enfin, nous procédons à quelques évaluations.

7.2 Problèmes de collaboration

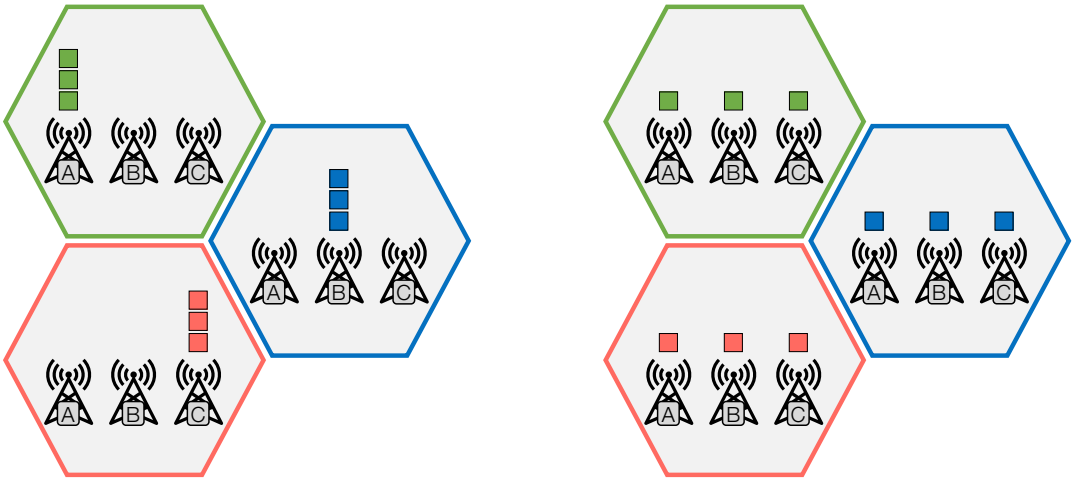
Commençons par définir et formaliser les deux types de problèmes que nous avons évoqué en introduction, à savoir l'extension de couverture ainsi que l'extension de capacité.

7.2.1 Extension de couverture

Nous commençons par formaliser dans cette section le problème de l'extension de couverture au sein des opérateurs de télécommunications.

Introduction et intuition

Le problème de l'extension de couverture concerne les situations où des opérateurs de télécommunications gèrent des cellules de réseau mobile pour lesquelles ils ne sont pas capables de fournir assez de ressources radio pour délivrer une qualité de service suffisante à leurs clients.



(a) Situation sans partage

(b) Situation après partage

FIGURE 7.2 – Extension de la couverture réseau. On peut envisager trois opérateurs A, B et C qui doivent délivrer un service de connectivité à des clients sur trois cellules différentes (rouge, verte et bleue). Dans la Figure 7.2a, un seul des opérateurs a de la ressource dans chaque cellule. Il pourrait donc être intéressant d'homogénéiser ces ressources de manière à fournir une connectivité suffisante à tous les clients des trois opérateurs (Figure 7.2b)

Formulation d'un problème d'échange de ressources

Nous considérons un jeu dans lequel N agents (opérateurs) $A_1, ..., A_N \in \mathcal{I}$ souhaitent se partager M items différents $B_1, ..., B_M \in \mathcal{J}$. $\mathcal{S}$ est l'ensemble des états qui correspondent aux ressources des agents. A chaque étape t, l'état est défini par $s(t) = \{s_{i,k}(t),\ i \in [\![1, N]\!]\ ,\ k \in [\![1, M]\!]\}$, où $s_{i,k}(t)$ correspond à la quantité de l'item B_k détenue par l'agent A_i. On suppose que la fonction d'utilité pour chaque agent A_i est $f^{(i)}(s(t))$ définie par :

$$f^{(i)}(s(t)) = \sum_{k=1}^{M} f_k^{(i)}(s_{i,k}(t))$$

où $f_k^{(i)}$ désigne l'utilité de l'item B_k pour l'agent A_i. On suppose que chacune de ces fonctions est considérée monotone croissante et concave en vertu du principe de l'utilité marginale décroissante des ressources [Gos54, Eas05], ainsi que compte tenu des estimations de qualité de service en fonction des ressources disponibles.

A chaque étape t, chaque agent A_i exécute une action $a^{(i)}(t)$ qui est un ensemble de transactions :

$$a^{(i)}(t) = (u^{(1)}, ..., u^{(m)})$$

où $u^{(k)} \in \mathcal{J} \times \mathcal{I} \times \mathbb{R}$ est une transaction qui représente le fait de céder une certaine quantité d'un item donné à un autre agent. Pour simplifier, on considère que les actions sont regroupées dans une action jointe $X = (X_1, ..., X_M)$ où chaque $X_k \in \mathcal{M}_N(\mathbb{R})$ est la somme des transactions de l'item B_k. La matrice X est alors anti-symétrique car céder une quantité Δ à un autre agent revient à recevoir $-\Delta$ de la part de ce même agent.

La fonction de transition $\mathcal{T}$ relie les états et les actions à l'état suivant par l'exécution des transactions comme résumé par la formule suivante :

$$s(t+1) = \mathcal{T}(s(t), X = (X_1, ..., X_M))$$
$$s_{i,k}(t+1) = s_{i,k}(t) + \sum_{j=1}^{N} (X_k)_{i,j}$$

où $(X_k)_{i,j}$ est la quantité de l'item B_k qui a été transmise de A_i à A_j. Enfin, on suppose que le jeu est à observation partielle : à chaque étape t, l'agent A_i n'observe qu'une part $\mathcal{O}(s_t, i)$ de l'état total s_t avec $\mathcal{O}(s_t, i) = \{s_{i,k}(t),\ k \in [\![1, M]\!]\} \cup \{(X_k)_{i,j}(t'),\ \forall t' < t\}$.

L'objectif prosocial du jeu serait idéalement de maximiser le bien commun (*social welfare*) défini par la somme des utilités :

$$\max \sum_{i=1}^{N} f^{(i)}(s(t))$$

où chaque agent A_i souhaite maximiser indépendemment sa propre fonction d'utilité $f^{(i)}(s(t))$ ce qui conduit à des équilibres de Nash non optimaux que l'on détaille dans la suite.

Situation de dilemme due à la concavité des fonctions d'utilité

Le problème posé comporte naturellement des questions liées à la théorie des jeux dû au simple fait que les agents ne sont pas coopératifs. En effet, les agents ont des intérêts personnels et leur objectif est de maximiser leur utilité personnelle. Dans cette partie, nous proposons une courte analyse de la situation sous l'angle de la théorie des jeux, à savoir montrer que l'équilibre de Nash [Nas51] n'est pas Pareto-optimal et que, par conséquent, notre problème peut être considéré comme un dilemme social. Rappelons qu'une stratégie est une fonction qui relie les états aux actions. Dans le problème posé, cela revient à relier les quantités d'items des agents à des transactions, ce que l'on peut poser par :

$$\pi^i : s^{(i)} \mapsto (X_k)_{i,j}$$

Si $G^{(i)}(\pi_j)$ est le gain de l'agent i, une stratégie jointe $(\pi_i^*)_{i\in\mathcal{I}}$ est un équilibre de Nash si :

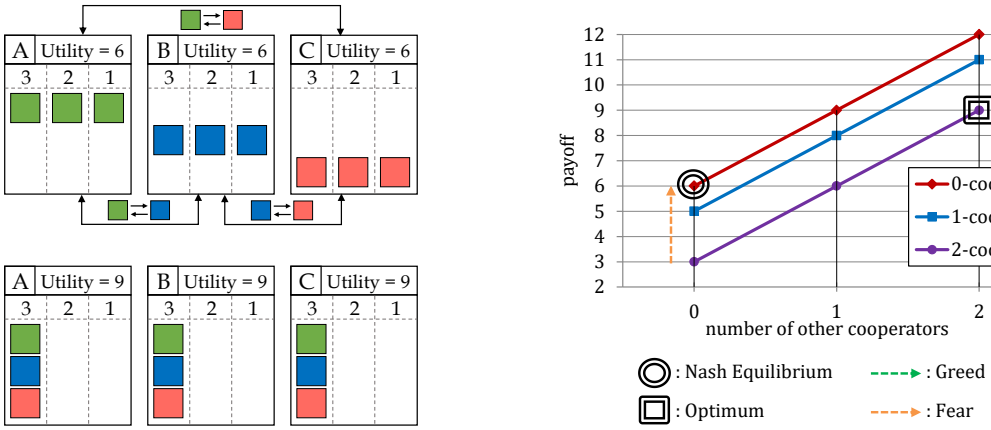
$$\forall i \in \mathcal{I},\ \forall \pi_i,\ G^{(i)}(\pi_i^*, \pi_{-i}^*) > G^{(i)}(\pi_i, \pi_{-i}^*)$$
$$\text{avec } \pi_{-i} = [\pi_1, ..., \pi_{i-1}, \pi_{i+1}, ..., \pi_N]$$

Comme les fonctions d'utilité f_k^i sont toutes strictement croissantes, on a donc que la fonction totale d'utilité de chaque agent A_i est strictement décroissante par rapport à n'importe quelle quantité $(X_k)_{i,j} > 0$ cédée à tout autre agent A_j. Par conséquent, la stratégie consistant à ne rien faire $\forall j,\ \forall k,\ (X_k)_{i,j} = 0$ est un équilibre de Nash.

Supposons maintenant qu'il existe au moins une paire d'agents (A_i, A_j) et une paire d'items (B_k, B_l) telles que $s_{i,k}(0) > s_{j,k}(0)$ et $s_{j,l}(0) > s_{i,l}(0)$ (*i.e.* un agent qui pourrait céder des items à un autre car il en a "trop" et vice versa) et telles que les fonctions (concaves) d'utilité vérifient :

$$\frac{\partial f_k^{(i)}}{\partial x}(s_{i,k}(0)) < \frac{\partial f_k^{(j)}}{\partial x}(s_{j,k}(0)) \quad \text{et} \quad \frac{\partial f_l^{(j)}}{\partial x}(s_{j,l}(0)) < \frac{\partial f_l^{(i)}}{\partial x}(s_{i,l}(0)) \tag{7.1}$$

Par conséquent, on peut affirmer qu'il existe des valeurs Δ_k et Δ_l telles que si A_i cède une quantité Δ_k de l'item B_k à A_j et que A_j cède une quantité Δ_l de l'item B_l à A_i, alors les joueurs auront augmenté tous les deux leur utilité totale. Notons que la condition 7.1 est vérifiée dès que les fonctions d'utilité sont identiques en vertu de la concavité des fonctions (dérivées décroissantes) et qu'alors les valeurs Δ_k et Δ_l seront idéalement choisies égales. C'est ce que nous choisirons dans la suite. Nous considérons les situations qui impliquent les conditions précédentes. Compte tenu alors du fait qu'il existe des transactions différentes de l'équilibre de Nash qui sont plus optimales au sens de Pareto, nous sommes donc en présence des situations dites de dilemme social. C'est pourquoi, nous avons fait le choix d'utiliser un algorithme comportant une brique de TFT que nous présentons dans la section suivante.



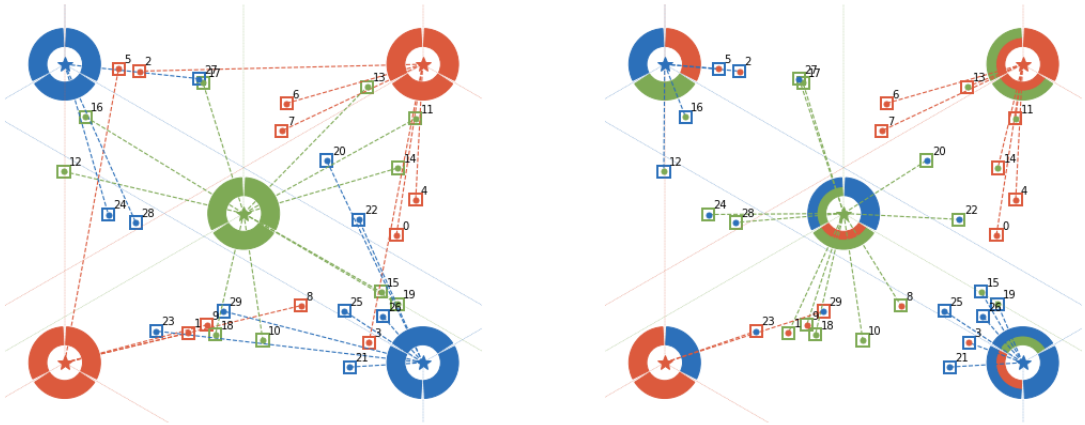
(a) Transactions optimales

(b) Diagramme de Schelling

FIGURE 7.3 – Jeu d'échange de ressources discrètes à trois joueurs : chaque joueur peut céder trois types d'items différents qui ont une utilité concave (utilité marginale décroissante). La Figure 7.3a montre les transactions optimales qui permettent d'élever les utilités initiales de 6 à 9 pour chaque agent. La Figure 7.3b représente la situation de dilemme à trois joueurs [LZL+17, Sch06] (voir section 2.3).

7.2.2 Extension de capacité

Dans cette section, nous considérons les situations d'extension de capacité aussi appelées densification. Les modèles utilisés seront issus du framework développé dans le chapitre 6.



(a) Situation initiale

(b) Situation optimale

FIGURE 7.4 – Exemple de jeux avec quatre opérateurs. La Figure 7.4a montre la situation initiale et la Figure 7.4b illustre la situation optimale après échanges de ressources radio sensiblement équitables d'un opérateur à l'autre.

7.3 Un modèle de partage par TFT

Dans cette section, nous présentons notre agent dont l'objectif est de choisir les transactions pour atteindre une situation optimale de manière incitative tout en tenant compte du potentiel égoïsme de certains des autres joueurs.

7.3.1 Architecture

Comme mentionné en introduction, nous proposons un modèle à base de TFT continu [LGMLR20]. Son principe, résumé dans la Figure 7.5 consiste à effectuer à chaque itération de jeu les quatre étapes suivantes :

1. Politique d'offre : consiste à effectuer le calcul des demandes et offres optimales. Seules les demandes sont ensuite communiquées aux autres joueurs (section 7.3.2)
2. Estimation du degré de coopération : en tenant compte des allocations précédentes des autres joueurs, chaque agent calcule une estimation du degré de coopération issu de chaque autre joueur (section 7.3.3)
3. Politique de réponse de coopération : il s'agit d'une fonction de TFT. Compte-tenu des degrés de coopération estimés, l'agent calcule une réponse de coopération adaptée (section 7.3.4).
4. Allocation : en fonction des degrés de coopération, des demandes en provenance des autres agents et des disponibilités (offres), une allocation de ressource est calculée pour chaque agent et chaque item (section 7.3.5).

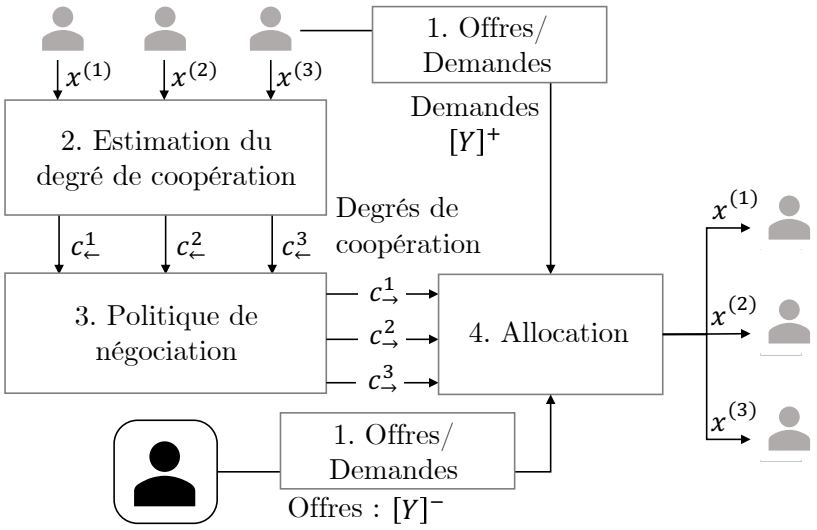


FIGURE 7.5 – Architecture de notre agent à base de TFT

7.3.2 Offres et demandes

A chaque itération du jeu, les agents communiquent leurs besoins (leurs demandes) à d'autres agents. Ils connaissent leurs offres (ce qu'ils sont prêts à céder) mais ne les divulguent pas. Pour notre agent, une possible approche en condition d'observation partielle, est de formuler une demande dont la somme des quantités est égale à ce qu'il est prêt à donner dans le cadre d'une coopération de type "un prêté pour un rendu". De cette façon, on peut proposer que chaque agent optimise un gain d'utilité "virtuel" :

$$d^{(i)} = \max_{Y^{(i)}} \sum_{k=1}^{M} f_k^{(i)}(s_{i,k} + Y_k^{(i)}) \tag{7.2}$$

où $d^{(i)}$ est le vecteur des demandes/offres pour l'agent A_i, avec une composante positive (resp. négative) qui correspond à une demande (resp. une offre).

7.3.3 Détection du degré de coopération

Un problème majeur de ce type de partage donnant-donnant est bien entendu la mesure de la coopération d'un comportement adverse. Savoir si les partenaires sont coopératifs ou non est primordial pour ne pas se faire exploiter. Il est même intéressant de calculer un degré (continu) de coopération pour être en mesure d'appliquer les fonctions de TFT que nous avons précédemment développées et étudiées. Pour commencer, nous considérons qu'il n'y a pas de distinction entre volonté et capacité de coopération. La première raison est que la collaboration entre agents sans transaction financière signifie échanger des quantités équivalentes de ressources en les allouant différemment de façon à ce que l'utilité maximale soit augmentée. Par conséquent, si un agent n'est pas (ou n'est plus) à même de délivrer une certaine quantité de ressources d'items, il n'y a plus d'intérêt de continuer à coopérer avec lui. La seconde raison, pour laquelle nous ne prenons pas en compte la possible incapacité temporaire à coopérer, est que les fonctions de TFT continues développées prennent elles-mêmes "en compte" (par construction) une forme d'indulgence et d'inertie. Ainsi, à chaque étape, chaque agent A_j estime le degré de coopération $c_{i,j}$ que chaque agent A_i a dirigé vers lui. Une possibilité est d'estimer ce degré comme le rapport entre ce que l'agent A_i vient de donner au total (estimé par ce que A_j a reçu de sa part multiplié par le nombre d'agents) comparé à ce que A_j est prêt à céder (l'offre de l'agent A_j pour l'item B_k est notée $d_k^{(i)}$ et est donnée par l'équation 7.2). On peut alors estimer le degré de coopération de A_i vers A_j au cours de la transaction précédente de la façon suivante :

$$c_{i,j}(t) = \frac{(N-1)\sum_{k=1}^{M}(X_k)_{i,j}}{\sum_{k=1}^{M}[d_k^{(j)}]^+} \tag{7.3}$$

7.3.4 Politique de réponse de coopération

Pour arriver à une négociation incitative et sûre entre des joueurs intéressés uniquement par leur gain personnel, l'idée est de se baser sur des algorithmes de TFT. Dans notre cas, il est possible d'utiliser une stratégie de TFT continu que nous avons pu étudier dans les chapitres précédents. On rappelle que le principe d'une politique de TFT est d'associer une observation de coopération du partenaire à une réponse de coopération. L'objectif est d'inciter à un comportement de coopération mutuelle tout en s'assurant de ne pas se faire exploiter. Dans la version continue du TFT, les choix d'actions représentent des degrés de coopération [Ver98]. Si l'on suppose qu'à chaque étape t, un joueur A observe de la part d'un joueur B un degré de coopération b_t, alors il peut choisir sa réponse a_t en fonction de la stratégie suivante :

$$\text{TFT}_{\alpha,\beta,r_0,c_0}(t, a_{t-1}, b_{t-1}), r_t = \begin{cases} c_0, r_0 & \text{si } t = 0 \\ \alpha a_{t-1} + (1-\alpha)(r_t + (1-r_t)b_{t-1}), & \\ [r_{t-1} + \beta(b_{t-1} - a_{t-1})]^+ & \text{si } t > 0. \end{cases} \tag{7.4}$$

avec $[x]^+ = \min(1, \max(0, x))$

et $\alpha, \beta, r_0, c_0 \in [0,1]^4$

Le coefficient α est un coefficient d'inertie permettant de lisser les réactions, r_t est un taux d'incitation dynamique qui est adapté au cours des itérations par un coefficient adaptatif β (voir Chapitre 3).

7.3.5 Allocation

L'allocation est la dernière étape de notre agent. Cette étape consiste à identifier les items qu'on souhaite donner et à quelle quantité. L'idée principale de cette étape est qu'un agent A_i calcule pour chaque item B_k la part maximale de ressource qu'il peut offrir $\Gamma_k^{(i)} = -[d_k^{(i)}]^-$ ($d_k^{(i)}$ est calculé par A_i à l'étape 2, voir section 7.3.2) et alloue à chaque autre agent A_j une part de cette offre qui est proportionnelle à la demande (plafonnée) de l'agent A_j : $\tilde{d}_k^{(j)} = \min(d_k^{(j)}, \Gamma_k^{(i)})$ ($d_k^{(j)}$ calculé par A_j à l'étape 2, voir section 7.3.2) et $c_{i,j}$ le degré de coopération entre A_i et A_j :

$$(X_k)_{(i,j)} = \frac{c_{i,j}\tilde{d}_k^{(j)}}{\sum_{l=1,l\neq i}^{N} c_{i,l}\tilde{d}_k^{(l)}} \Gamma_k^{(i)}$$

Avec cette allocation, on s'assure que $\sum_j (X_k)_{(i,j)} < \Gamma_k^{(i)}$ et que la quantité $(X_k)_{(i,j)}$ augmente par rapport à la fois à $c_{i,j}$ et à $\tilde{d}_k^{(j)}$.

7.4 Évaluation de l'extension de couverture

Nous procédons dans cette section à quelques évaluations du partage de ressources [2].

7.4.1 Modèle

Pour commencer, nous supposons que N agents disposent de M items dont les quantités de ressource sont des valeurs comprises dans $]-2, +\infty[$ avec des valeurs initiales comprises dans $]-1, +\infty[$. Une quantité négative de ressource indique un très fort déficit (très fort besoin) d'un item donné ce qui conduit à une utilité très faible. C'est pourquoi, nous avons associé les quantités d'items à une des fonctions concaves les plus simples : une fonction logarithme, qui chute en cas de valeurs proches de zéro et dont le taux d'accroissement décroît. Pour nos simulations, on choisit une fonction d'utilité commune à tous les items : $\forall k,\ \forall i,\ f_k^{(i)} : x \mapsto \ln(x+2)$. Les valeurs proches de -2 conduisant donc à une utilité proche de $-\infty$. Pour les simulations, nous choisissons un cas simple avec 3 agents et 3 items, ce qui peut correspondre par exemple aux situations de la Figure 7.2 ou celle de la Figure 7.7a. Dans cette dernière, par exemple, l'agent dispose de 3 items (-1 pour l'item A, $+2$ pour le B et $+2$ pour le C).

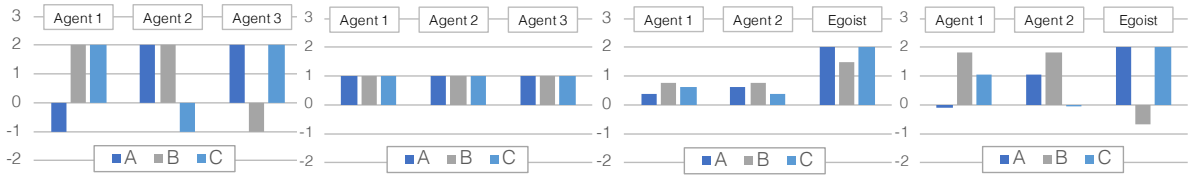
FIGURE 7.6 – Modèle de simulation

7.4.2 Expérimentations

Nous simulons donc notre politique introduite en section 7.3 dans le jeu simple présenté dans la Figure 7.7a. Nous étudions ainsi le cas où les trois agents suivent la politique que l'on a introduite, ainsi qu'un second cas où il y a un agent égoïste parmi les joueurs afin d'observer la robustesse de notre politique face à l'exploitation.

Les Figures 7.8, 7.9 et 7.10 représentent les résultats d'une simulation de notre modèle d'agent dans des situations différentes. Dans chacune de ces figures : à gauche on représente les utilités personnelles des trois agents ainsi que le bien commun (la somme des trois) ; à droite, on représente le degré moyen de coopération entrant (*receiving*) et sortant (*sending*). Pour commencer, la Figure 7.8 représente la situation où les trois agents suivent notre modèle de TFT dont l'issue est illustrée par la Figure 7.7b. Dans la Figure 7.8a on peut observer que le bien commun augmente jusqu'à un maximum. Remarquons qu'après une première phase de convergence, la détection de coopération est très instable (Figure 7.8b), ceci s'explique par le fait qu'elle est reliée aux précédentes transactions qui deviennent infimes. Cependant, cela n'a pas d'importance, seule la convergence du bien commun est importante. La Figure 7.9 implique, cette fois-ci, deux agents de TFT classique (avec un coefficient adaptatif β nul). Les deux agents

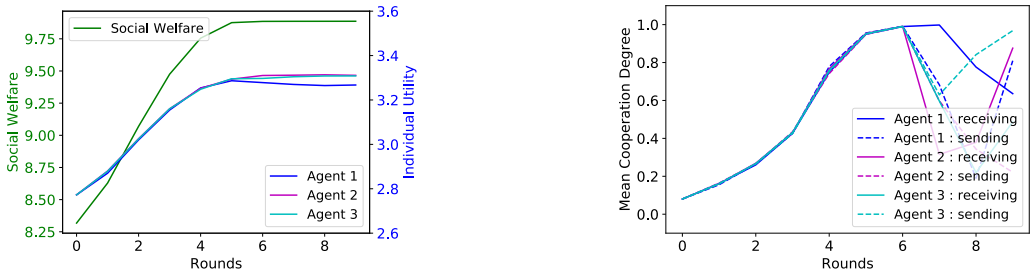
2. Le code source de nos simulations est disponible sur GitHub : `https://github.com/tlgleo/sharing_resources`



(a) État initial (b) Situation optimale (c) TFT classique (d) TFT amélioré

FIGURE 7.7 – Exemple de partage de ressources entre trois agents avec trois items dont l'utilité marginale est décroissante (utilité concave). La Figure 7.7a montre la situation initiale où les items ne sont pas partagés. 7.7b désigne la situation optimale, *i.e.* où les agents auraient effectué les transactions optimisées pour maximiser le bien commun. 7.7c et 7.7d montrent le cas où deux agents (suivant une politique de TFT) se font exploiter par un égoïste dans deux cas : le premier où les deux agents choisissent un TFT classique (7.7c) et le second où la politique choisie par ces deux agents est notre TFT amélioré (7.7d).

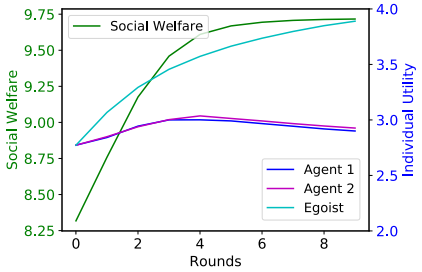
font face à un agent égoïste (un défecteur pur). On observe que ces deux agents sont un peu exploités puisque le défecteur profite alors du taux d'incitation constant r_0 et donc continue de bénéficier de quelques transactions à son avantage. En effet, son utilité est plus élevée que celles des deux autres. On peut constater les quantités finales des items sur la Figure 7.7c. Enfin, sur la Figure 7.10, la pertinence du coefficient adaptatif β est montrée. En effet, le taux d'incitation est rapidement diminué ce qui a pour effet d'arrêter les transactions au profit du défecteur et ainsi permettre aux deux agents TFT d'être moins exploités. Les quantités échangées avec le défecteur à l'issue du jeu auront donc été plus faibles qu'avec le cas du TFT classique (voir Figure 7.7d).



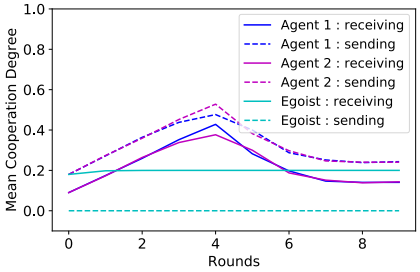
(a) Évolution du bien commun et des utilités individuelles

(b) Évolution des degrés de coopération entrant et sortant

FIGURE 7.8 – Simulation dans le cadre de l'exemple de la Figure 7.7 avec trois agents TFT avec ($\alpha = 0.5$, $r_0 = 0.2$, $\beta = 0$).

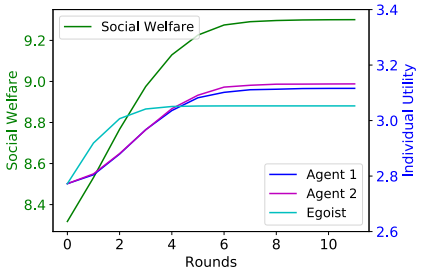


(a) Évolution du bien commun et des utilités individuelles

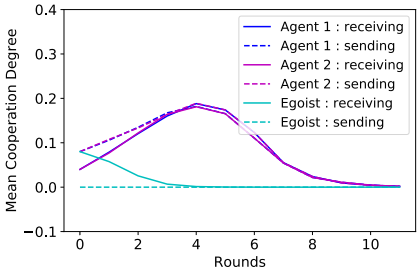


(b) Évolution des degrés de coopération entrant et sortant

FIGURE 7.9 – Simulation dans le cadre de l'exemple de la Figure 7.7 avec deux agents TFT avec ($\alpha = 0.5$, $r_0 = 0.2$, $\beta = 0$) faisant face à un défecteur pur.



(a) Évolution du bien commun et des utilités individuelles



(b) Évolution des degrés de coopération entrant et sortant

FIGURE 7.10 – Simulation identique à la Figure 7.9 mais les deux agents TFT ont un coefficient adaptatif β non nul : TFT($\alpha = 0.5$, $r_0 = 0.2$, $\beta = 0.3$). Cela permet pour ces deux agents d'être moins exploités par le défecteur pur.

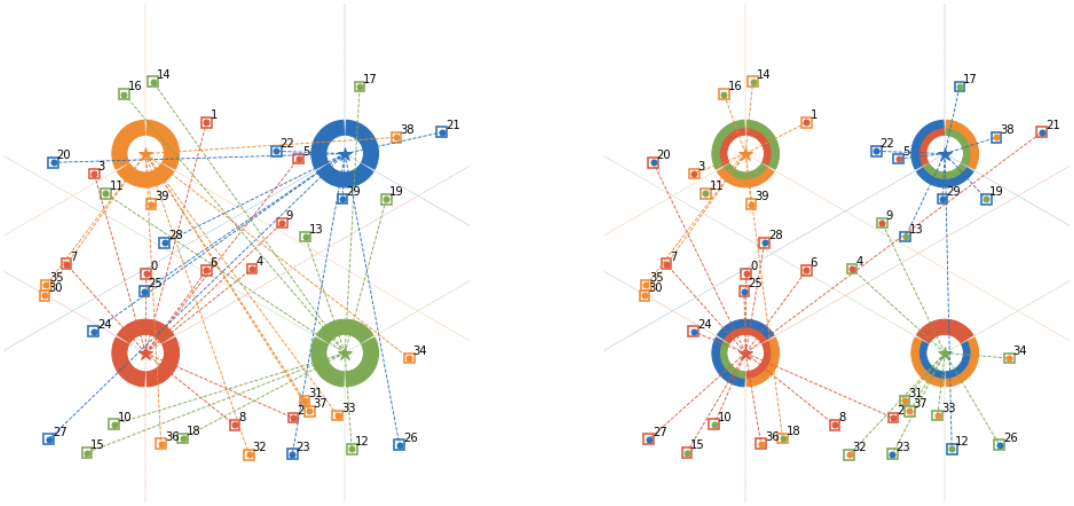
7.5 Évaluation de l'extension de capacité

Comme évoqué en introduction, nous nous intéressons également au concept de densification, c'est-à-dire, le principe qui consiste à échanger des ressources réseau de manière à augmenter la capacité de réseau mobile. Pour ce faire, l'agent consiste à demander et offrir des ressources pour une cellule donnée et ainsi permettre de mieux répartir les utilisateurs, en les prenant (et faisant prendre) en charge par des antennes géographiquement plus proches.

7.5.1 Exemple de jeu pour les simulations

Considérons un jeu à quatre opérateurs (disposant chacun d'une station de base) autour desquelles se tiennent de manière homogène et aléatoire 40 terminaux mobiles de clients. Chacun des opérateurs gèrent 10 clients différents. La Figure 7.11a représente une situation où aucun partage n'a été effectué. La Figure 7.11b propose une situation plus optimale où chaque opérateur

aurait échangé des ressources réseau auprès d'opérateurs précis sur des cellules bien identifiées.



(a) Situation avant partage

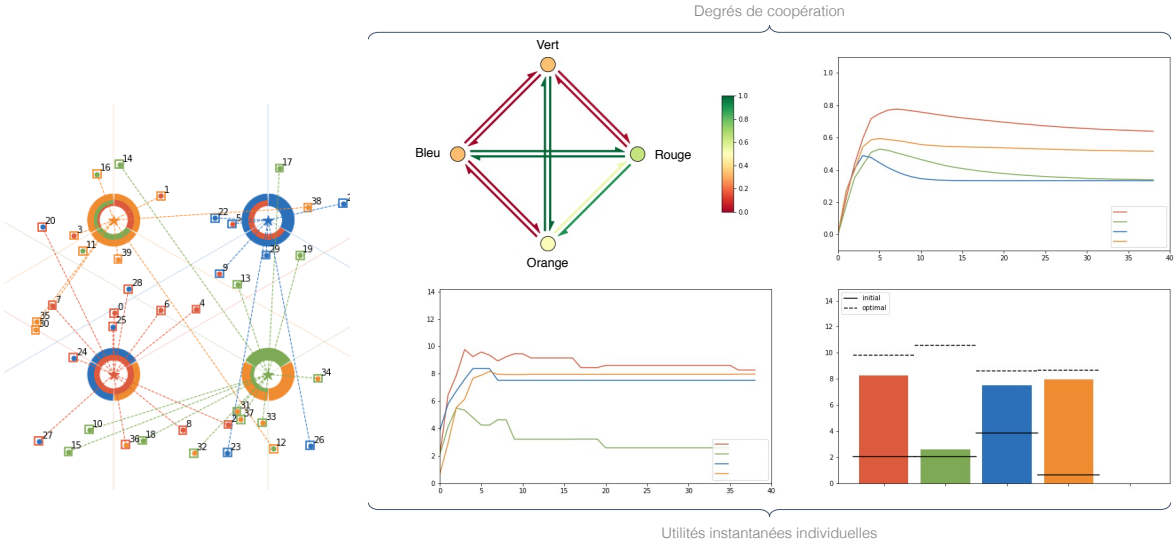
(b) Situation après partage

FIGURE 7.11 – Exemple de jeux avec quatre opérateurs. La Figure 7.11a montre la situation initiale avant le partage des ressources. La Figure 7.11b illustre une situation plus optimale où les opérateurs ont procédé à des échanges équitables pour augmenter leur utilité.

7.5.2 Évaluations

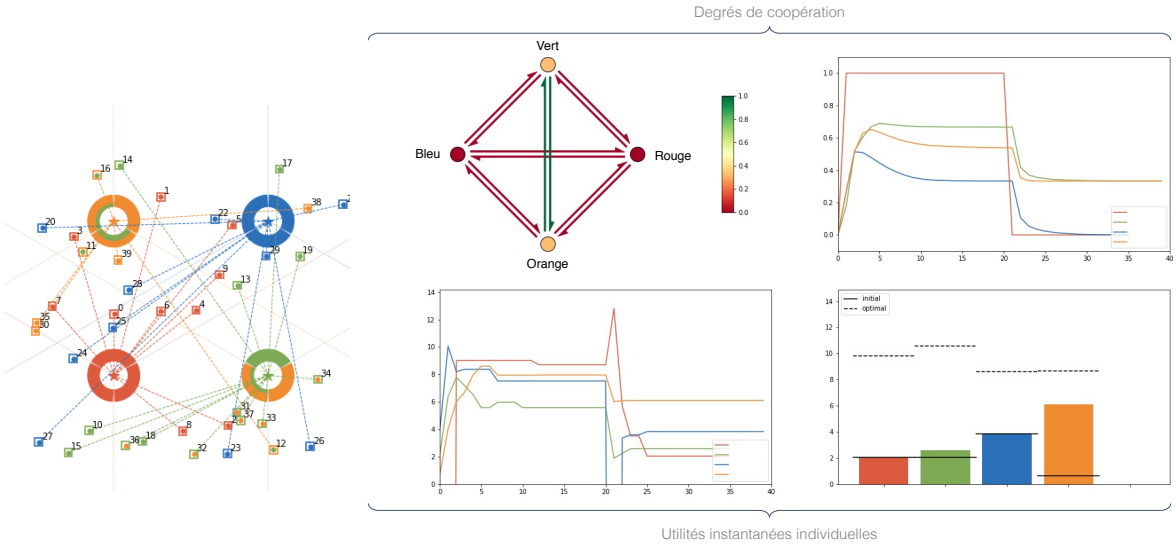
Comme dans la section 7.4, nous proposons de simuler, des cas où les agents suivraient la même politique (introduite en section 7.3) avec ou sans agents égoïstes. Nous reprenons le jeu de la section 7.5.1 pour procéder aux évaluations suivantes.

Nous pouvons observer dans la première simulation (Figure 7.12) que les quatre opérateurs ont pu trouver un arrangement, au moins de manière bilatérale (Orange/Vert et Rouge/Bleu) de manière à tous augmenter leur utilité. Leurs clients sont maintenant rattachés à des antennes plus proches. En revanche, on peut constater que les gains d'utilité sont très hétérogènes d'un opérateur à l'autre. Par exemple, l'opérateur Vert est pénalisé : bien que sa fonction d'utilité augmente légèrement, il en gagne bien moins que ses concurrents. Cela s'expliquerait principalement par le fait que les différentes optimisations qui ont lieu au cours de l'algorithme (vu en section 7.3) sont effectuées dans un cadre d'observation partielle. Par conséquent, selon les configurations, certaines offres et demandes idéales peuvent être mal calculées ce qui conduit à un accord final non optimal. Nous verrons en conclusion que certaines solutions pourraient aborder des approches de consensus. Concentrons-nous maintenant sur la seconde simulation (Figure 7.13), où l'un des agents (le Rouge) se met à ne plus coopérer (volontairement) à partir d'un certain moment.



(a) Environnement final (b) Degrés de coopération et utilités instantanées

FIGURE 7.12 – Exemple de jeux avec quatre opérateurs qui jouent une politique de TFT. La Figure 7.12a montre l'environnement à la fin du jeu après le partage des ressources. La Figure 7.12b montre l'évolution des degrés de coopération et des utilités instantanées[1].



(a) Environnement final (b) Degrés de coopération et utilités instantanées

FIGURE 7.13 – Exemple de jeux avec quatre opérateurs. Trois opérateurs suivent une politique de TFT, un dernier (le Rouge) va coopérer jusqu'à $t = 20$ puis se comporter de manière purement égoïste. La Figure 7.13a représente l'environnement final, la Figure 7.13b montre l'évolution des degrés de coopération et des utilités instantanées[2].

Dans cette situation, on constate que les autres opérateurs réagissent grâce au TFT inclus dans l'algorithme. On observe que deux des opérateurs (le Vert et le Orange) trouvent alors une entente en délaissant le troisième (le Bleu) qui est pourtant volontaire. Cela s'expliquerait également par les problèmes d'observation partielle que nous discuterons brièvement en conclusion.

7.6 Conclusion et perspectives

Dans ce chapitre, nous avons présenté un formalisme de partage de ressources dont l'utilité marginale est décroissante. Les différents acteurs agissent dans un contexte d'échange sans transaction financière et sans tiers, ni contrôleur. Nous avons adopté ce point de vue pour modéliser les deux types de scénarios de collaboration entre opérateurs, à savoir : l'extension de couverture et l'extension de capacité. Partant du constat que ces situations rentrent dans la catégorie des dilemmes sociaux, nous avons proposé un agent utilisant une fonction de TFT continu comme module interne pour négocier des transactions avec les autres participants. Nous avons alors simulé des situations d'extension de couverture et de capacité avec les agents que nous avons introduits en leur faisant faire face notamment à des agents égoïstes. Nous avons observé que les agents arrivent assez bien à converger vers une coopération mutuelle et sont robustes contre la défection, particulièrement avec notre TFT amélioré avec le coefficient adaptatif β. Cependant, les limites de notre modèle sont assez nombreuses. En effet, dans notre configuration à observation partielle, il est nécessaire que les ressources soient bien réparties, soit en quantités dans le premier cas étudié, soit en répartition géographique de clients dans le second cas. Une des perspectives pour adresser ces problèmes pourrait être l'utilisation d'algorithmes d'optimisation par consensus où des agents parviennent à optimiser une fonction de coût sur des variables locales sans les partager [RBA05, Ren07, ZK14, OSM04, MB14]. Dans ce cas, on conserverait alors l'aspect d'observation partielle tout en améliorant l'optimisation effectuée par les agents.

1. Vidéo de la simulation disponible ici : `https://youtu.be/YaF_eIncJiE`
2. Vidéo de la simulation disponible ici : `https://youtu.be/_EAZlxmVDcQ`

Conclusions et perspectives

Cette thèse s'est concentrée sur l'étude d'agents intelligents au sein de dilemmes sociaux, à savoir des situations dans lesquelles des acteurs sont tentés de préférer leur intérêt personnel au détriment de l'intérêt commun. Des études ont été menées sur l'apprentissage de politiques dans ces situations, notamment dans le dilemme du prisonnier itéré. La thèse a notamment fait l'objet de contributions sur divers modèles et variantes du dilemme du prisonnier et des algorithmes pour les adresser. Ce paradigme a été appliqué ensuite dans des scénarios de collaboration entre les opérateurs de télécommunications, en particulier, avec de nombreuses simulations et évaluations effectuées dans des environnements développés pour l'occasion. Dans cette conclusion, nous revenons sur les diverses contributions de cette thèse, en relevant notamment leurs limites pour enfin ouvrir sur les nombreuses perspectives de futurs travaux que ces sujets offrent.

Rappels du positionnement de la thèse et des contributions

La thèse s'inscrit dans un contexte de jeux multi-agents où les différents joueurs n'ont a priori pas d'intérêt à s'entraider bien qu'ils sortiraient tous gagnants à adopter une approche collective. Ce type de situation est connu sous le nom de dilemme social, avec en particulier le dilemme du prisonnier. Avec le nouvel essor des techniques d'apprentissage par renforcement, ces types de jeux ont refait leur apparition sous des formes plus complexes. Pour appréhender ces situations, nous avons proposé des contributions de modèles de jeux, des améliorations d'algorithmes ; sous ce paradigme, nous avons aussi proposé de modéliser certaines collaborations entre opérateurs de télécommunication afin d'étudier les comportements d'agents dans ces environnements de coopération.

Pour commencer, nous nous sommes penchés sur des dilemmes avec des actions simples de type dilemme du prisonnier itéré. Après avoir brièvement fait le même constat que le consensus de la littérature, qui suggère que des agents d'apprentissage par renforcement peinent à converger

vers une coopération mutuelle, nous avons choisi de nous pencher vers des stratégies connues pour adresser les dilemmes sociaux. Un exemple de telles stratégies est le Tit-for-Tat (TFT) : une stratégie de type donnant-donnant qui permet à la fois d'inciter à la coopération mutuelle et qui est robuste à l'éventuel égoïsme d'un adversaire. Nous avons donc proposé des améliorations de fonction de TFT continu que nous avons évaluées grâce à des métriques sociales que nous avons pour la plupart introduites [LGMLR20].

Dans un second temps, nous nous sommes penchés sur les modèles du dilemme du prisonnier à plus de deux joueurs. Nous nous sommes particulièrement intéressés aux situations où des agents ne peuvent s'entraider que via un ou plusieurs joueur(s) tiers. Étant donné que les modèles existants ne prennent pas en compte la non-réciprocité de la coopération ainsi que l'existence potentielle de chemins circulaires de coopération au sein des joueurs, nous avons introduit un nouveau modèle de dilemme du prisonnier à base de graphe (*Graph-based Iterated Prisoner's Dilemma* (GIPD)) qui permet de prendre en compte des coopérations maximales entre les joueurs qui ne seraient pas réciproques [LGMLR22]. A cette occasion, nous avons étendu notre précédente fonction de TFT continu par l'ajout d'un algorithme de traitement de graphe. Ce nouvel algorithme appelé *Graph-based Tit-for-Tat* (GTFT) consiste à chercher le meilleur cycle de coopération dans un graphe orienté et pondéré de manière à ce que chaque joueur puisse recevoir autant de coopération qu'il en a proposé, même si le joueur doit passer par un autre joueur tiers quand la coopération réciproque n'est pas permise. A notre connaissance, notre modèle est nouveau et nous sommes convaincus qu'il permet de modéliser la plupart des structures de coopération existantes au sein de situations à deux joueurs ou plus. Notre modèle comporte néanmoins quelques limitations. Il est en effet difficile de quantifier et définir une valeur de coopération maximale. Par ailleurs, nos agents GTFT peuvent faire face, dans certaines situations, à des problèmes de synchronisation pour le choix des meilleurs cycles optimaux.

Nous nous sommes ensuite penchés sur une extension existante du dilemme social qui est le dilemme social séquentiel (*Sequential Social Dilemma* (SSD)), il s'agit d'un jeu qui étend les dilemmes sociaux en des situations plus complexes où les agents jouent des politiques d'apprentissage par renforcement (RL) au lieu des actions atomiques (coopération/défection). Dans le même esprit que le GIPD, nous avons étendu les SSD dans une version non nécessairement réciproque. Pour adresser ces nouveaux jeux, nous avons étendu une version d'algorithme existante qui propose une approche hybride à base de stratégies de TFT et de politiques de RL pré-entraînées. Notre extension se base sur l'utilisation du GTFT en tant que module interne. Bien que notre premier agent candidat proposé pour ce paradigme soit globalement satisfaisant, il rencontre quelques difficultés comme la détection de coopération qui reste parfois instable. Par ailleurs, la question du pré-entraînement de politiques de RL aux degrés de coopération différents soulève

bien entendu la question de l'explosion combinatoire des agencements possibles. Enfin, l'hypothèse d'observation totale nécessaire à l'adaptabilité de notre algorithme est évidemment très forte et est parfois inadaptée à certaines situations réelles. Nous discuterons des pistes possibles d'améliorations dans la prochaine section.

Dans cette thèse, il a également été l'occasion de répondre à un paradigme de collaboration entre les opérateurs de télécommunications. Plutôt que de nous diriger vers des approches à base de transactions financières, nous avons fait le choix d'adopter un point de vue dans lequel les opérateurs sont les joueurs d'un dilemme social dans lequel ils doivent chercher à inciter à la coopération sans se faire exploiter. L'objectif de la collaboration dans ce cas est d'échanger équitablement des ressources radio de manière à obtenir des gains de performance et de qualité de connectivité pour les clients. Un autre objectif est celui d'aboutir à des baisses de consommation énergétiques. Pour pallier le manque d'environnement de simulation multi-opérateurs, une des premieres contributions a été l'élaboration d'un framework destiné à générer des jeux au format OpenAI Gym de Python [LGMLR21]. Les jeux générés sont alors flexibles et entièrement modifiables. En effet, il est possible de définir le nombre d'opérateurs, le nombre et la position des antennes relais ainsi que les fonctions de récompense. Les positions des antennes peuvent être artificiellement définies pour créer des jeux simples pour des évaluations brèves. Notre framework propose également de considérer des scénarios plus complexes et réalistes en exploitant les données publiques de l'Agence Nationale des Fréquences (ANFR) qui met à disposition notamment les positions des antennes des quatre opérateurs français. Enfin, nous avons implémenté des algorithmes à base de TFT [LGMLR20] pour traiter le problème d'échange de ressources équitables entre les opérateurs de télécommunication. Nous avons mené des simulations et évaluations en particulier sur les environnements issus de notre framework. Les résultats des évaluations sont prometteurs, cependant, compte tenu du caractère novateur de l'approche, nous faisons face à des difficultés liées au réalisme de la modélisation des échanges entre opérateurs. Par ailleurs, le passage à l'échelle pose également quelques difficultés compte tenu des coûts en calcul dès lors que le nombre d'antennes augmente.

Perspectives

Les axes de recherche suivis tout le long de la thèse offrent de nombreuses perspectives tant sur les débuts prometteurs de l'apprentissage par renforcement multi-agents au sein de dilemmes sociaux séquentiels que sur les agents de collaboration inter-opérateurs. Commençons par rappeler que certaines des contributions présentées dans ce manuscrit portaient sur des propositions de modèles de dilemmes sociaux séquentiels asymétriques dans lesquels nous étudions des agents de RL augmentés d'une politique de TFT à structure de graphe. Les limites de ce

modèle étaient entre autres le besoin d'avoir accès à l'état total de l'environnement. Il serait alors envisageable comme futurs travaux d'étudier la communication entre agents de sorte à identifier les agents égoïstes même en cas d'observation partielle. Les algorithmes de consensus pour un apprentissage commun et partagé en conservant des données privées pourraient être une première approche [RBA05, ZK14, MB14]. Par ailleurs, au sein d'un environnement multi-agents, le pré-entraînement de politiques de RL de divers degrés de coopération est sujet à une explosion combinatoire. Il serait alors intéressant d'étudier les possibilités de limiter le nombre de modèles pré-entraînés. Une idée possible serait d'entraîner un unique modèle qui incorporerait, dans l'observation de cette unique politique, un vecteur comportant les degrés de coopération attendus. La littérature sur le sujet des dilemmes sociaux séquentiels est encore assez limitée, c'est pourquoi nous sommes convaincus que de nombreux travaux seront réalisés prochainement.

En ce qui concerne la collaboration entre les opérateurs de télécommunications, notre approche pourrait être enrichie par de futures simulations plus réalistes. Un des obstacles les plus importants à nos travaux a été la disposition d'environnements réalistes capables de simuler les transactions entre opérateurs. Notre première version de framework générateur de jeux multi-opérateurs pourrait être largement améliorée. Il pourrait être pertinent de calculer de meilleures estimations de fonctions d'utilité des opérateurs en utilisant des mesures internes d'un opérateur et de les recaler auprès des autres opérateurs (dont on ne dispose pas forcément des données) à l'aide de données extérieures telles que celles de l'INSEE. Avec de tels environnements de simulation, il serait possible d'accéder à toutes les fonctions d'utilité pour chaque transaction de ressources effectuées et ainsi émettre des conclusions sur la faisabilité des collaborations inter-opérateurs comme nous l'envisageons. Par ailleurs, après l'extension de couverture et l'extension de capacité, un troisième paradigme est en cours d'étude dans la suite de nos travaux. Il s'agit de la réduction des coûts énergétiques au cours de la nuit. En effet, la demande nocturne de connectivité étant assez faible, il serait envisageable d'éteindre certaines antennes pour alléger la consommation. Il serait alors judicieux d'envisager également de le faire en collaboration avec d'autres opérateurs.

Pour conclure, nous sommes convaincus que davantage de problèmes autour de l'intelligence artificielle coopérative vont émerger compte tenu notamment des enjeux liés à l'énergie et aux ressources. En effet, avec l'explosion des agents intelligents qui nous entourent, il est primordial de tenir compte désormais de l'impact environnemental du numérique et de favoriser une certaine sobriété par une coopération quand cela est profitable. Le dernier essor des aspects coopératifs autour de l'intelligence artificielle est relativement récent et intéresse maintenant de grands acteurs liés au numérique et aux données. Nous sommes donc confiants quant au développement de ce domaine et des nombreuses perspectives prometteuses.

Liste des publications

[1] T. Le Gléau, X. Marjou, T. Lemlouma, and B. Radier, "Multi-agents ultimatum game with reinforcement learning," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2020, pp. 267–278.

[2] T. Le Gléau, X. Marjou, T. Lemlouma, and B. Radier, "Game theory approach in multi-agent resources sharing," in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020, pp. 1–6.

[3] T. Le Gléau, X. Marjou, T. Lemlouma, and B. Radier, "A multi-agent openai gym environment for telecom providers cooperation," in *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2021, pp. 28–32.

[4] T. Le Gléau, X. Marjou, T. Lemlouma, and B. Radier, "Towards circular and asymmetric cooperation in a multi-player graph-based iterated prisoner's dilemma," in *14th International Conference on Agents and Artificial Intelligence (ICAART)*, 2022.

Bibliographie

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2) :235–256, 2002.

[ACBFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1) :48–77, 2002.

[AF11] Juan Enrique Agudo and Colin Fyfe. Reinforcement learning for the n-persons iterated prisoners' dilemma. In *2011 Seventh International Conference on Computational Intelligence and Security*, pages 472–476. IEEE, 2011.

[AH81] Robert Axelrod and William Donald Hamilton. The evolution of cooperation. *science*, 211(4489) :1390–1396, 1981.

[ALRW07] Toh-Kyeong Ahn, Myungsuk Lee, Lore Ruttan, and James Walker. Asymmetric payoffs in simultaneous and sequential prisoner's dilemma games. *Public Choice*, 132(3-4) :353–366, 2007.

[AMCB21] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv :2101.05265*, 2021.

[Ash07] Daniel A Ashlock. Cooperation in prisoner's dilemma on graphs. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 48–55. IEEE, 2007.

[Aze19] Chloé-Agathe Azencott. *Introduction au machine learning.* Dunod, 2019.

[BBS91] Andrew Gehret Barto, Steven J Bradtke, and Satinder P Singh. *Real-time learning and control using asynchronous dynamic programming.* University of Massachusetts at Amherst, Department of Computer and . . . , 1991.

[BCP+16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, et al. Openai gym. *arXiv preprint arXiv :1606.01540*, 2016.

[BDM+01] Bruno Beaufils, Jean-Paul Delahaye, Philippe Mathieu, et al. Adaptive behaviour in the classical iterated prisoner's dilemma. In *Proc. Artificial Intelligence & Simul. Behaviour Symp. on Adaptive Agents & Multi-Agent Systems*. Citeseer, 2001.

[BEBS+19] Mounia Bouabdellah, Faissal El Bouanani, Paschalis C Sofotasios, Sami Muhaidat, Daniel Benevides Da Costa, Kahtan Mezher, Hussain Ben-Azza, and George K Karagiannidis. Cooperative energy harvesting cognitive radio networks with spectrum sharing and security constraints. *IEEE Access*, 7 :173329–173343, 2019.

[BEKN09] Mariana Blanco, Dirk Engelmann, Alexander K Koch, and Hans-Theo Normann. Preferences and beliefs in a sequential social dilemma : A within-subjects analysis. 2009.

[Bel57] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5) :679–684, 1957.

[BER14] Karsten Buckwitz, Jan Engelberg, and Gernot Rausch. Licensed shared access (LSA)—Regulatory background and view of administrations. In *9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 413–416. IEEE, 2014.

[BHSMR07] Martin Beckenkamp, Heike Hennig-Schmidt, and Frank P Maier-Rigaud. Cooperation in symmetric and asymmetric prisoner's dilemma games. *MPI Collective Goods Preprint*, (2006/25), 2007.

[Bot12] Matthew Michael Botvinick. Hierarchical reinforcement learning and decision making. *Current opinion in neurobiology*, 22(6) :956–962, 2012.

[BTA+99] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Ken Tang, Rajive Bagrodia, and Mario Gerla. Glomosim : A scalable network simulation environment. *UCLA computer science department technical report*, 990027(1999) :213, 1999.

[CAK96] Xiao-Ping Chen, Wing Tung Au, and SS Komorita. Sequential choice in a step-level public goods dilemma : The effects of criticality and uncertainty. *Organizational Behavior and Human Decision Processes*, 65(1) :37–47, 1996.

[CBA+19] Ayman Chouayakh, Aurelien Bechler, Isabel Amigo, et al. Auction mechanisms for Licensed Shared Access : reserve prices and revenue-fairness trade offs. *ACM SIGMETRICS Performance Evaluation Review*, 46(3) :43–48, 2019.

[CK05] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 67–73, 2005.

[Cou38] Antoine Augustin Cournot. *Recherches sur les principes mathématiques de la théorie des richesses*. L. Hachette, 1838.

[CZWZ13] Yanjiao Chen, Jin Zhang, Kaishun Wu, and Qian Zhang. Tames : A truthful auction mechanism for heterogeneous spectrum allocation. In *Proceedings IEEE INFOCOM*, pages 180–184, 2013.

[Daw10] Marian Stamp Dawkins. Do asymmetries destabilize the prisoner's dilemma and make reciprocal altruism unlikely ? *Animal Behaviour*, 80(2) :339, 2010.

[DCH+04] Jim Dowling, Raymond Cunningham, Anthony Harrington, Eoin Curran, and Vinny Cahill. Emergent consensus in decentralised systems using collaborative reinforcement learning. In *Self-star Workshop*, pages 63–80. Springer, 2004.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*, 2018.

[DM95] Jean-Paul Delahaye and Philippe Mathieu. Complex strategies in the iterated prisoner's dilemma. *Chaos and society*, 29 :283–292, 1995.

[DM96] Jean-Paul Delahaye and Philippe Mathieu. Random strategies in a two levels iterated prisoner's dilemma : How to avoid conflicts. In *Proceedings of the ECAI*, volume 96, pages 68–72, 1996.

[DMB00] Jean-Paul Delahaye, Philippe Mathieu, and Bruno Beaufils. The iterated lift dilemma. In *Computational conflicts*, pages 202–223. Springer, 2000.

[Eas05] Richard A Easterlin. Diminishing marginal utility of income ? caveat emptor. *Social Indicators Research*, 70(3) :243–255, 2005.

[EHK+19] Tom Eccles, Edward Hughes, János Kramár, Steven Wheelwright, and Joel Z Leibo. Learning reciprocity in complex sequential social dilemmas. *arXiv preprint arXiv :1903.08082*, 2019.

[FADFW16] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29 :2137–2145, 2016.

[Fan53] Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences of the United States of America*, 39(1) :42, 1953.

[FCAS+17] Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv :1709.04326*, 2017.

[FF56] Lester Randolph Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8 :399–404, 1956.

[FJ56] Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.

[Flo58] Merrill M Flood. Some experimental games. *Management Science*, 5(1) :5–26, 1958.

[Fog93] David B Fogel. Evolving behaviors in the iterated prisoner's dilemma. *Evolutionary Computation*, 1(1) :77–97, 1993.

[Fra81] Marguerite Frank. The braess paradox. *Mathematical Programming*, 20(1) :283–302, 1981.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[Gos54] Hermann Heinrich Gossen. *Entwickelung der Gesetze des menschlichen Verkehrs, und der daraus fließenden Regeln für menschliches Handeln*. Friedrich Vieweg und Sohn, 1854.

[Ham73] Henry Hamburger. N-person prisoner's dilemma. *Journal of Mathematical Sociology*, 3(1) :27–48, 1973.

[Har67] John C Harsanyi. Games with incomplete information played by "bayesian" players, i–iii part i. the basic model. *Management science*, 14(3) :159–182, 1967.

[Har68] Garrett Hardin. The tragedy of the commons. *Science*, 162(3859) :1243–1248, 1968.

[HCS+16] Robert Hult, Gabriel R Campos, Erik Steinmetz, Lars Hammarstrand, Paolo Falcone, and Henk Wymeersch. Coordination of cooperative autonomous vehicles : Toward safer and more efficient road transportation. *IEEE Signal Processing Magazine*, 33(6) :74–84, 2016.

[HIT13] Fabien Héliot, Muhammad Ali Imran, and Rahim Tafazolli. Low-complexity energy-efficient resource allocation for the downlink of cellular systems. *IEEE transactions on communications*, 61(6) :2271–2281, 2013.

[HK04] Philip Hingston and Graham Kendall. Learning versus evolution in iterated prisoner's dilemma. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, volume 1, pages 364–372. IEEE, 2004.

[HKJ+17] Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E. Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PLOS ONE*, 12(12) :1–33, 12 2017.

[HLP+18] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in neural information processing systems*, pages 3326–3336, 2018.

[HS15] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *arXiv preprint arXiv :1507.06527*, 2015.

[IH09] Teerawat Issariyakul and Ekram Hossain. Introduction to network simulator 2 (ns2). In *Introduction to network simulator NS2*, pages 1–18. Springer, 2009.

[IZL+18] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, pages 2117–2126. PMLR, 2018.

[JLH+18] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, et al. Intrinsic social motivation via causal influence in multi-agent RL. 2018.

[JLH+19] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049. PMLR, 2019.

[JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3) :264–323, 1999.

[Jol05] Ian Jolliffe. Principal component analysis. *Encyclopedia of statistics in behavioral science*, 2005.

[KBP13] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics : A survey. *The International Journal of Robotics Research*, 32(11) :1238–1274, 2013.

[KD02] Timothy Killingback and Michael Doebeli. The continuous prisoner's dilemma and the evolution of cooperation through reciprocal altruism with variable investment. *The American Naturalist*, 160(4) :421–438, 2002.

[KHH+12] Mohammad T Kawser, Nafiz Imtiaz Bin Hamid, Md Nayeemul Hasan, et al. Downlink snr to cqi mapping for different multipleantenna techniques in lte. *International journal of information and electronics engineering*, 2(5) :757, 2012.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1) :79–86, 1951.

[KM97] Atsushi Kajii and Stephen Morris. The robustness of equilibria to incomplete information. *Econometrica : Journal of the Econometric Society*, pages 1283–1309, 1997.

[KT00] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[KXN+19] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning : A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6) :10700–10714, 2019.

[LB07] Stephen Le and Robert Boyd. Evolutionary dynamics of the continuous iterated prisoner's dilemma. *Journal of theoretical biology*, 245(2) :258–267, 2007.

[LCS10] Lingzhi Luo, Nilanjan Chakraborty, and Katia Sycara. Prisoner's dilemma in graphs with heterogeneous agents. In *2010 IEEE Second International Conference on Social Computing*, pages 145–152. IEEE, 2010.

[Lev96] Robert Levinson. General game-playing and reinforcement learning. *Computational Intelligence*, 12(1) :155–176, 1996.

[LGMLR20] Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, and Benoît Radier. Game theory approach in multi-agent resources sharing. In *25th IEEE Symposium on Computers and Communications (ISCC)*, 2020.

[LGMLR21] Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, and Benoit Radier. A multi-agent openai gym environment for telecom providers cooperation. In *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 28–32. IEEE, 2021.

[LGMLR22] Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, and Benoit Radier. Towards circular and asymmetric cooperation in a multi-player graph-based iterated prisoner's dilemma. In *14th International Conference on Agents and Artificial Intelligence*, 2022.

[LH19] Renzhi Lu and Seung Ho Hong. Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Applied energy*, 236 :937–949, 2019.

[LHP+15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv :1509.02971*, 2015.

[Lit94] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[LM92] Long-Ji Lin and Tom M Mitchell. *Memory approaches to reinforcement learning in non-Markovian domains*. Citeseer, 1992.

[LMFP+11] Guillaume J Laurent, Laëtitia Matignon, Le Fort-Piat, et al. The world of independent learners is not markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15(1) :55–64, 2011.

[LP17] Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv :1707.01068*, 2017.

[LVC18] Tuyen P Le, Ngo Anh Vien, and TaeChoong Chung. A deep hierarchical reinforcement learning algorithm in partially observable markov decision processes. *Ieee Access*, 6 :49089–49102, 2018.

[LWT+17] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv :1706.02275*, 2017.

[LYD+17] Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal ? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv :1706.05125*, 2017.

[LYNW20] Jiayang Li, Jing Yu, Yu Nie, and Zhaoran Wang. End-to-end learning and intervention in games. *Advances in Neural Information Processing Systems*, 33, 2020.

[LZ07] John Langford and Tong Zhang. Epoch-greedy algorithm for multi-armed bandits with side information. *Advances in Neural Information Processing Systems (NIPS 2007)*, 20 :1, 2007.

[LZL+17] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, et al. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 464–473, 2017.

[MB14] Anup Menon and John S Baras. Collaborative extremum seeking for welfare optimization. In *53rd IEEE Conference on Decision and Control*, pages 346–351, 2014.

[MBM+16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[MF02] Michael W Macy and Andreas Flache. Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences*, 99(suppl 3) :7229–7236, 2002.

[MKS+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv :1312.5602*, 2013.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540) :529–533, 2015.

[MLLFP07] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic q-learning : an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 64–69. IEEE, 2007.

[MLLFP12] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games : a survey regarding coordination problems. *Knowledge Engineering Review*, 27(1) :1–31, 2012.

[MMR+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[MSBD16] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8) :1647–1650, 2016.

[MSGR18] Babak Mafakheri, Tejas Subramanya, Leonardo Goratti, and Roberto Riggio. Blockchain-based infrastructure sharing in 5g small cell networks. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 313–317. IEEE, 2018.

[MVN53] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[Mye79] Roger B Myerson. Incentive compatibility and the bargaining problem. *Econometrica : journal of the Econometric Society*, pages 61–73, 1979.

[Nas51] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[NS93] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432) :56–58, 1993.

[NW89] N. Nguyen and B. Widrow. The truck backer-upper : An example of self learning in neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, pages 357–363. IEEE Press, 1989.

[OGW+94] Elinor Ostrom, Roy Gardner, James Walker, James M Walker, and Jimmy Walker. *Rules, games, and common-pool resources*. University of Michigan Press, 1994.

[Orl97] James B Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2) :109–129, 1997.

[OSM04] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9) :1520–1533, 2004.

[PA08] José N Portela and Marcelo S Alencar. Cellular coverage map as a voronoi diagram. *Journal of Communication and Information Systems*, 23(1), 2008.

[PL17] Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv preprint arXiv :1709.02865*, 2017.

[PLZ+17] Julien Perolat, Joel Z Leibo, Vinicius Zambaldi, Charles Beattie, Karl Tuyls, and Thore Graepel. A multi-agent reinforcement learning model of common-pool resource appropriation. In *Advances in Neural Information Processing Systems*, pages 3643–3652, 2017.

[PS08] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4) :682–697, 2008.

[RBA05] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the American Control Conference*, pages 1859–1864. IEEE, 2005.

[RCO65] Anatol Rapoport, Albert M Chammah, and Carol J Orwant. *Prisoner's dilemma : A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.

[Ren07] Ren, Wei and Beard, Randal W and Atkins, Ella M. Information consensus in multivehicle cooperative control. *IEEE Control systems magazine*, 27(2) :71–82, 2007.

[RG04] DR Robinson and DJ Goforth. Alibi games : The asymmetric prisoner's dilemmas. *Toronto, June*, 4, 2004.

[Rob52] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5) :527–535, 1952.

[Ros57] Frank Rosenbaltt. The perceptron–a perciving and recognizing automation. *Cornell Aeronautical Laboratory*, 1957.

[RPP11] Vandana Milind Rohokale, Neeli Rashmi Prasad, and Ramjee Prasad. A cooperative internet of things (iot) for rural healthcare monitoring and control. In *2011 2nd international conference on wireless communication, vehicular technology, information theory and aerospace & electronic systems technology (Wireless VITAE)*, pages 1–6. IEEE, 2011.

[SB+98] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[SBW92] Richard S Sutton, Andrew G Barto, and Ronald J Williams. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2) :19–22, 1992.

[Sch73] Thomas C Schelling. Hockey helmets, concealed weapons, and daylight saving : A study of binary choices with externalities. *Journal of Conflict resolution*, 17(3) :381–428, 1973.

[Sch06] Thomas C Schelling. *Micromotives and macrobehavior*. WW Norton & Company, 2006.

[Ser17] M Series. Guidelines for evaluation of radio interface technologies for imt-2020. *Report ITU-R M. 2412-0, Tech. Rep.*, 2017.

[SF89] Arvind Sathi and Mark S Fox. Constraint-directed negotiation of resource reallocations. In *Distributed artificial intelligence*, pages 163–193. Elsevier, 1989.

[SGJ73] John P Sheposh and Philip S Gallo Jr. Asymmetry of payoff structure and cooperative behavior in the prisoner's dilemma game. *Journal of Conflict Resolution*, 17(2) :321–333, 1973.

[SH18] Andrey V Savkin and Hailong Huang. Deployment of unmanned aerial vehicle base stations for optimal quality of coverage. *IEEE Wireless Communications Letters*, 8(1) :321–324, 2018.

[Sha53] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10) :1095–1100, 1953.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587) :484–489, 2016.

[SK02] William D Smart and L Pack Kaelbling. Effective reinforcement learning for mobile robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 3404–3410. IEEE, 2002.

[SLA+15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[Sli19] Aleksandrs Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv :1904.07272*, 2019.

[SMSM00] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[SR97] Simon Singh and David Rintoul. *Fermat's last theorem*. Fourth Estate London, 1997.

[SS15] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[SSAAY13] Yevgeny Seldin, Csaba Szepesvári, Peter Auer, and Yasin Abbasi-Yadkori. Evaluation and analysis of the performance of the exp3 algorithm in stochastic environments. In *European Workshop on Reinforcement Learning*, pages 103–116. PMLR, 2013.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv :1707.06347*, 2017.

[Tho33] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4) :285–294, 1933.

[TMK+17] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4) :e0172395, 2017.

[TNWL20] Zhang-Peng Tian, Ru-xin Nie, Jian-qiang Wang, and Ru-yin Long. Adaptive consensus-based model for heterogeneous large-scale group decision making : Detecting and managing non-cooperative behaviors. *IEEE Transactions on Fuzzy Systems*, 2020.

[TR20] Ricardo Tavares and Martyn Roetter. Beyond mobile-network sharing : Regulatory challenges in dense urban areas. 2020.

[Tur20] Annie Turner. Inception : Digital twins for 5g network infrastructure-sharing, Oct 2020.

[TW00] Kagan Tumer and David H Wolpert. Collective intelligence and braess' paradox. In *Aaai/iaai*, pages 104–109, 2000.

[Ver93] Tom Verhoeff. A continuous version of the prisoner's dilemma. 1993.

[Ver98] Tom Verhoeff. The trader's dilemma : A continuous version of the prisoner's dilemma. *Computing Science Notes*, 93(02), 1998.

[WA95] Jianzhong Wu and Robert Axelrod. How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict resolution*, 39(1) :183–189, 1995.

[WD92] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4) :279–292, 1992.

[WDF+15] Huiyang Wang, Eryk Dutkiewicz, Gengfa Fang, et al. Spectrum sharing based on truthful auction in licensed shared access systems. In *IEEE 82nd Vehicular Technology Conference*, pages 1–5, 2015.

[WDM08] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1) :9–9, 2008.

[WN99] Lindi M Wahl and Martin A Nowak. The continuous prisoner's dilemma : Ii. linear reactive strategies with noise. *Journal of Theoretical Biology*, 200(3) :323–338, 1999.

[WT13] Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. *arXiv preprint arXiv :1301.2315*, 2013.

[WYZL20] Kangzhou Wang, Biao Yuan, Mengting Zhao, and Yuwei Lu. Cooperative route planning for the drone and truck in delivery services : A bi-objective optimisation approach. *Journal of the Operational Research Society*, 71(10) :1657–1674, 2020.

[Yao96] Xin Yao. Evolutionary stability in the n-person iterated prisoner's dilemma. *Bio-Systems*, 37(3) :189–197, 1996.

[YD94] Xin Yao and Paul J Darwen. An experimental study of n-person iterated prisoner's dilemma games. *Informatica*, 18(4) :435–450, 1994.

[YKAD12] Elias Yaacoub, Abdullah Kadri, and Adnan Abu-Dayya. Cooperative wireless sensor networks for green internet of things. In *Proceedings of the 8h ACM symposium on QoS and security for wireless and mobile networks*, pages 79–80, 2012.

[ZK14] Ruiliang Zhang and James Kwok. Asynchronous distributed ADMM for consensus optimization. In *International conference on machine learning*, pages 1701–1709, 2014.

[ZLNW12] Yang Zhang, Chonho Lee, Dusit Niyato, and Ping Wang. Auction approaches for resource allocation in wireless systems : A survey. *IEEE Communications surveys & tutorials*, 15(3) :1020–1041, 2012.

[ZLQ+20] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7) :6360–6368, 2020.

[ZPDW18] Hengjie Zhang, Iván Palomares, Yucheng Dong, and Weiwei Wang. Managing non-cooperative behaviors in consensus-based multiple attribute group decision making : An approach based on social network analysis. *Knowledge-Based Systems*, 162 :29–45, 2018.

[ZZ09] Xia Zhou and Haitao Zheng. TRUST : A general framework for truthful double spectrum auctions. In *IEEE INFOCOM*, pages 999–1007, 2009.

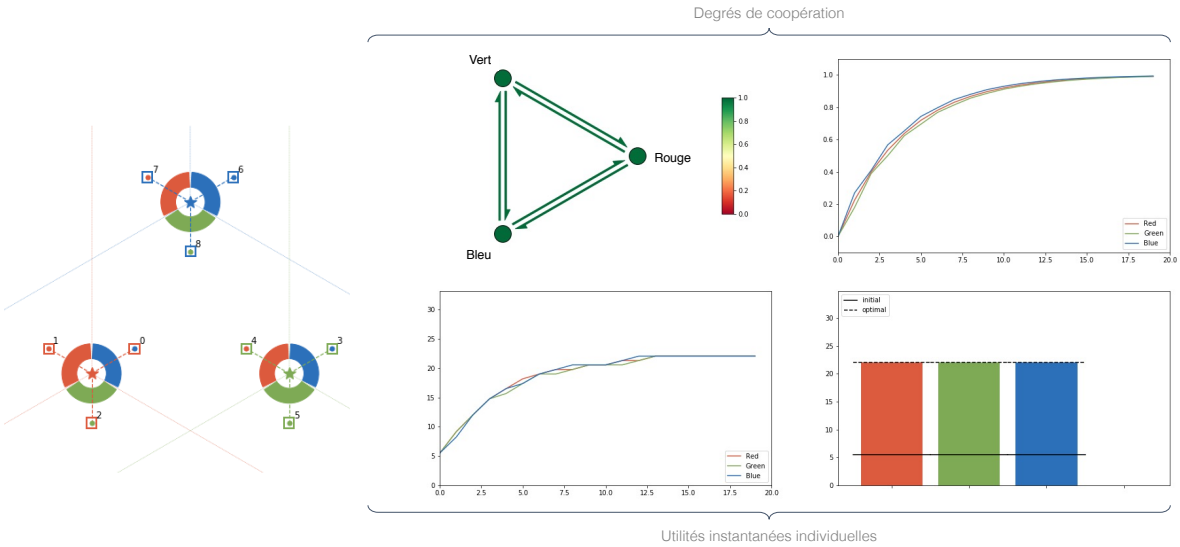
[ZZ19] Yan Zhang and Michael M Zavlanos. Distributed off-policy actor-critic reinforcement learning with policy consensus. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4674–4679. IEEE, 2019.

Annexe A

Simulations additionnelles et détails des métriques sociales

A.1 Simulations additionnelles

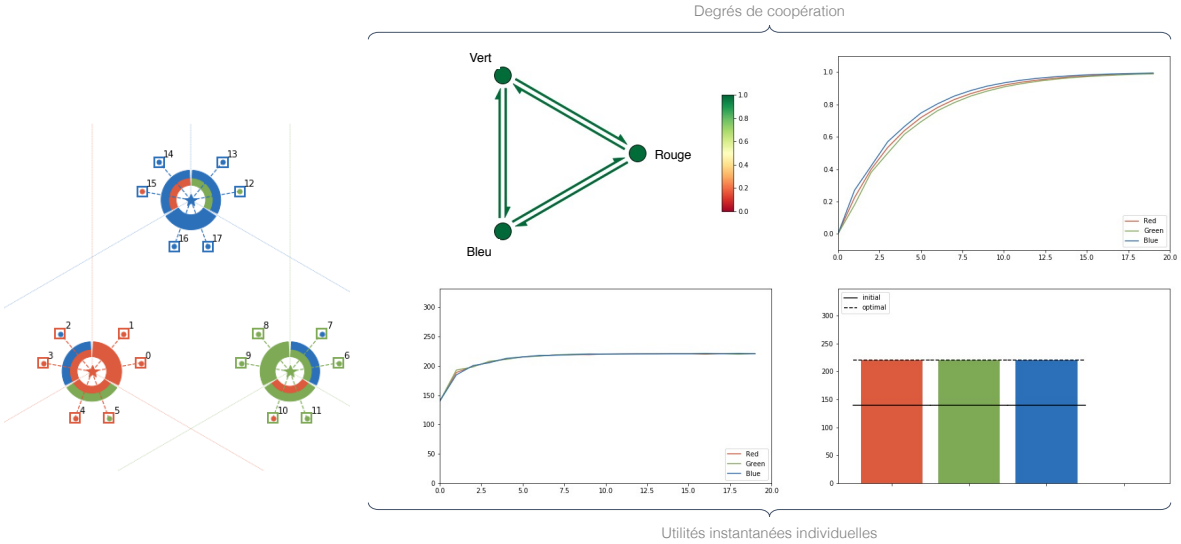
Nous présentons quelques simulations qui viennent compléter à celles de la section 7.5 du chapitre 7.



(a) Environnement final (b) Degrés de coopération et utilités instantanées

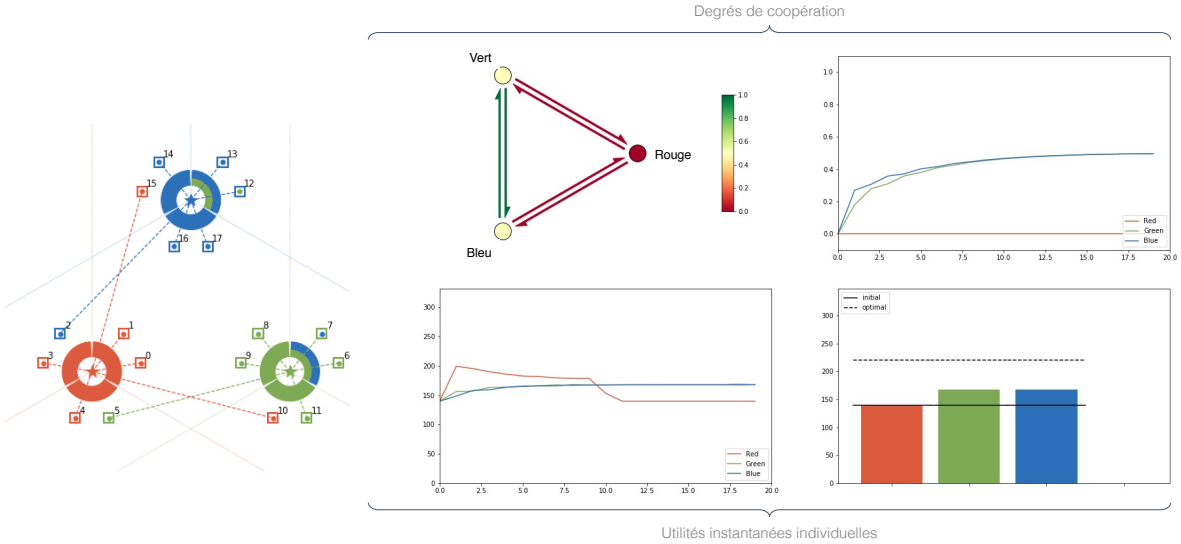
FIGURE A.1 – Exemple de situation avec trois opérateurs impliquant trois agents de TFT dans le jeu `env_3A_3S_9U-v0`[1].

1. Vidéo de la simulation disponible ici : `https://youtu.be/OI4Hf-TEhwc`



(a) Environnement final (b) Degrés de coopération et utilités instantanées

FIGURE A.2 – Exemple de situation avec trois opérateurs impliquant trois agents de TFT dans le jeu `env_3A_3S_18U-v0`[2].



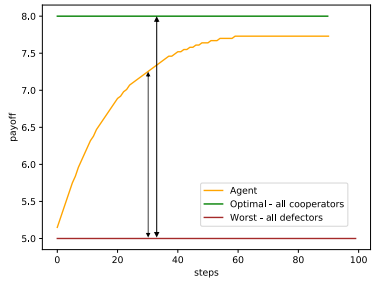
(a) Environnement final (b) Degrés de coopération et utilités instantanées

FIGURE A.3 – Exemple de situation avec trois opérateurs impliquant deux agents de TFT et un agent égoïste dans le jeu `env_3A_3S_18U-v0`[3].

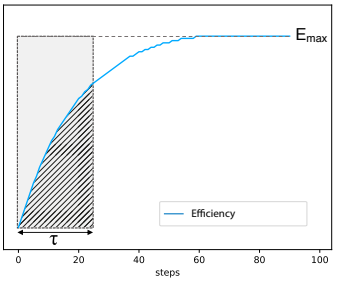
2. Vidéo de la simulation disponible ici : `https://youtu.be/TTKzdQ2mljc`
3. Vidéo de la simulation disponible ici : `https://youtu.be/P30ekeNIRSo`

A.2 Intuition des métriques sociales

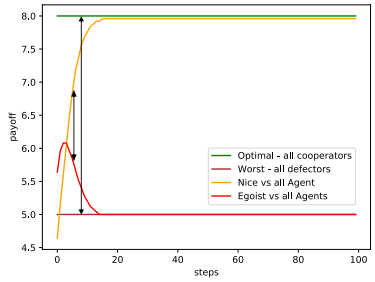
Pour mieux comprendre les métriques sociales que nous avons introduites lors des études des Tit-for-tat, nous proposons de les représenter sous forme graphique dans la Figure A.4.



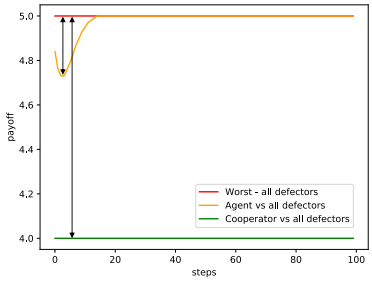
(a) L'efficacité d'une politique est une fonction du temps et est définie sur la figure par le ratio entre les différences indiquées par les double flèches.



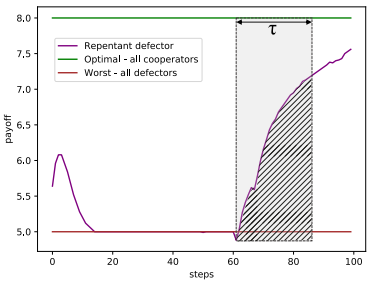
(b) Nous définissons la vitesse par le rapport entre l'intégrale de l'efficacité sur une période donnée τ (partie hachurée) et la variation maximale (rectangle gris)



(c) L'incitation est calculée par le ratio des différences indiquées par les flèches.



(d) la sûreté est calculée par le ratio des différences indiquées par les flèches.



(e) La métrique *Forgiveness* d'un agent désigne le rapport entre l'intégrale (partie hachurée) et la variation maximale (rectangle gris), similaire au formalisme pour la métrique vitesse

FIGURE A.4 – Intuition des métriques sociales sous forme graphique

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Apprentissage de stratégies coopératives dans un contexte de jeu multi-opérateurs de Télécom : L'intelligence artificielle coopérative au service des dilemmes sociaux

Mot clés : Théorie des jeux, Dilemme du prisonnier itéré, Apprentissage par renforcement

Résumé : L'objectif de la thèse est l'étude de l'apprentissage de politiques au sein de jeux non-coopératifs à somme non-nulle (de type dilemmes sociaux) dans le but de modéliser des interactions de coopération entre opérateurs de Télécom. Pour commencer, il a été intéressant d'étudier le comportement d'agents apprenants (tels que des politiques d'apprentissage par renforcement (RL) ou des bandits manchots) dans un dilemme du prisonnier itéré (IPD). Les premières conclusions montrent que le RL peine à converger vers des politiques de coopération mutuelles. Étant donné ce constat, il devient important de nous intéresser à des stratégies simples comme le Tit-for-tat (TFT) qui viendront à terme s'ajouter à des politiques plus complexes de type RL. Les principales contributions de la thèse ont été dans un premier temps des propositions d'améliorations de stratégies simples à deux joueurs telles que le TFT continu. Nous nous sommes ensuite intéressés aux modèles de dilemmes du prisonnier à N joueurs. Nous avons introduit une extension qui permet de modéliser une coopération non nécessairement bilatérale et potentiellement circulaire, ce qui a conduit alors à une proposition de stratégie adaptée, basée sur du TFT continu et des algorithmes de traitement de graphe. Dans un second temps, nous avons étendu le paradigme précédent au formalisme des dilemmes sociaux séquentiels (une extension existante de l'IPD qui permet d'étendre les actions atomiques des joueurs en des politiques plus complexes). Pour adresser ce nouveau modèle de jeu, nous avons alors proposé une stratégie qui utilise des politiques de RL et des stratégies de TFT. Enfin, nous avons procédé à quelques simulations dans un contexte Télécom. La première contribution a été l'implémentation d'un environnement de simulation de collaboration multi-opérateurs. Quelques simulations ont été ensuite conduites : les stratégies précédemment développées ont été mises en jeu dans divers scénarios de coopération multi-opérateurs.

Title: Learning cooperative strategies in a game of multiple Telecom providers: Cooperative artificial intelligence at the service of social dilemmas

Keywords: Game Theory, Iterated Prisoner's Dilemma, Reinforcement Learning

Abstract: The objective of this PhD thesis is the study of policy learning within general-sum non-cooperative games (in particular the social dilemmas) in order to model cooperative interactions between telecom providers. First, it has been interesting to study the behavior of learning agents (such as reinforcement learning (RL) policies or multi-armed bandits) in an iterated prisoner's dilemma (IPD). The first conclusions show that RL polices struggle to converge towards mutual cooperation. Given this observation, it becomes important to focus on simple strategies like Tit-for-tat (TFT) which will eventually be added to more complex policies (such Deep RL). The main contributions of the thesis were initially improvements proposal for simple two-player strategies such as continuous TFT. We then turned to N-player prisoner dilemma models. We have introduced an extension allowing to model a cooperation that is not necessarily bilateral and can be potentially circular, which then led to a proposal for a suitable strategy, based on continuous TFT and graph-processing algorithms. Secondly, we extended the previous paradigm to the formalism of sequential social dilemmas (SSD) (an existing extension of the IPD that extends the atomic actions of players into complex RL policies). To address this new game model, we then proposed a strategy that uses RL policies and TFT strategies. Finally, we carried out some simulations in a Telecom context. The first contribution was the implementation of a multi-provider environment for the cooperation simulation. A few simulations were then carried out: some of the previously developed strategies were used to study the agent's behavior in various multi-provider scenarios of cooperation.