



UNIVERSITÉ  
DE NAMUR

University of Namur

# Institutional Repository - Research Portal Dépôt Institutionnel - Portail de la Recherche

researchportal.unamur.be

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Etude de mécanismes de maintien de la réputation de services web collaboratifs

Mathy, M; Oversteyns, M

*Award date:*  
2012

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 23. Jun. 2020



Facultés Universitaires Notre-Dame de la Paix de Namur  
Faculté d'informatique  
Année académique 2011-2012

# Étude de mécanismes de maintien de la réputation de services web collaboratifs

Maxime MATHY et Michaël OVERSTEYNS

---

Mémoire présenté en vue de l'obtention du diplôme de Master en Sciences Informatiques



---

## Résumé

Dans le domaine de la collaboration de services web, les communautés de services web constituent un rassemblement de plusieurs de ces entités afin d'améliorer la capacité globale et la couverture géographique de leur service. Plusieurs services web fournissant le même service vont donc collaborer. Pour gérer une telle communauté, un service web spécial, appelé le maître de communauté, se charge entre autre de recruter de nouveaux services web volontaires. Cependant, afin de s'assurer que ceux-ci ne dégraderont pas la qualité de service de la communauté, le maître de communauté a tout intérêt à connaître leur qualité de service. Pour ce faire, il va consulter et payer des services d'information afin que ceux-ci lui fournissent les informations voulues, à savoir une notation de la qualité de service du service web désirant entrer. Le problème est que de tels services d'information pourraient être corrompues, soit par le service web voulant absolument rentrer dans la communauté, soit par un service web concurrent déjà membre de celle-ci mais ne voulant pas que le nouveau service web rentre. Il est donc nécessaire d'établir un mécanisme d'incitation afin que les services d'information soient honnêtes. Ce mécanisme se composera d'un système de réputation, reflétant l'honnêteté de chacun des services d'information, et d'une fonction de paiement, rémunérant les services d'information à la hauteur de leurs informations. On étudie les stratégies à long terme des services d'information, afin de mettre en évidence leurs choix selon leur niveau de patience. Ces thèmes sont étudiés et développés dans ce mémoire, ainsi que les résultats provenant de simulations sur le modèle exposé, afin de pouvoir analyser l'efficacité du système mis en place.

## Mots-clés

Services web, communautés de services web, mécanismes d'incitation, réputation, théorie des jeux.

## Abstract

In the field of web services collaboration, communities of web services are collections of many of these entities to improve the overall capacity and the coverage of their service. Several web services providing the same service will therefore cooperate. To manage such a community, a special web service, called the master of community, has to recruit new web services. The community master's highest interest is to know what is the quality of service of the web service. To do this, he will consult and pay some information services to give him the needed information. The problem is that such information services could be corrupted either by the web service willing to enter the community, either through a web service already member of the community which doesn't want that the new web service comes in. It is therefore necessary to establish an incentive mechanism so that information services keep being honest. This mechanism will consist of a reputation system, reflecting the honesty of each of the information services and a payment function, paying information services according to their information. We study the long-term strategies of information services, in order to highlight their choices according to their level of patience. These themes are developed in this paper, as well as results from simulations on the model presented, in order to analyze the effectiveness of the system in place.

## Keywords

Web services, Community of web services, incentive mechanism, reputation, game theory.

---

## Remerciements

Nous tenons à remercier notre promoteur, le professeur des FUNDP Philippe Thiran, ainsi que le promoteur de notre stage à la Concordia University de Montréal, le professeur Jamal Bentahar.

Nous remercions également la famille pour les conseils, remarques et corrections orthographiques.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Le contexte . . . . .	2
1.1.1	Services web, communautés, et réputation . . . . .	2
1.2	Le sujet . . . . .	5
1.2.1	Le problème . . . . .	5
1.2.2	Notre recherche . . . . .	6
1.3	Méthodologie . . . . .	7
1.3.1	Les systèmes multi-agents . . . . .	7
1.3.2	Les systèmes de réputation . . . . .	7
1.3.3	La théorie des jeux . . . . .	8
1.3.4	Le problème . . . . .	8
1.3.5	Élaboration de la fonction de paiement . . . . .	8
1.3.6	La corruption par un tiers . . . . .	8
1.3.7	Étude des jeux sur le long terme . . . . .	8
1.3.8	Expérimentation . . . . .	8
<b>2</b>	<b>Les mécanismes de réputation</b>	<b>9</b>
2.1	But et principe d'un système de réputation . . . . .	9
2.2	Les modèles de réputation utilisés par les sites de commerce électronique . . . . .	10
2.3	Points faibles du type de modèles précédent et améliorations possibles . . . . .	11
2.3.1	Sporas : un modèle de réputation pour les communautés faiblement connectées . . . . .	11
2.3.2	Histos : un modèle de réputation pour les communautés fortement connectées . . . . .	12
2.4	Le système BRS . . . . .	12
2.4.1	La fonction Beta . . . . .	12
2.4.2	Calcul de la réputation d'un agent . . . . .	13
2.4.3	Combiner les feedbacks . . . . .	13
2.4.4	Le système BRS dans notre travail . . . . .	14
2.5	Capacités d'une approche efficace selon Zhang Jie . . . . .	14
2.6	Mécanismes d'incitation et système de réputation . . . . .	14
2.7	Deux grands types de mécanismes d'incitation pour les plateformes de commerce électronique [Jie09] . . . . .	15
2.7.1	Les mécanismes de paiement secondaire . . . . .	15

2.7.2	Les mécanismes de crédibilité . . . . .	15
<b>3</b>	<b>Éléments de la théorie des jeux</b>	<b>17</b>
3.1	Théorie des jeux et choix rationnel . . . . .	17
3.1.1	Les actions . . . . .	18
3.1.2	Préférences et fonction d'utilité . . . . .	18
3.1.3	Interactions entre agents . . . . .	19
3.2	Jeux stratégiques avec information parfaite . . . . .	19
3.2.1	Un exemple bien connu : le dilemme du prisonnier . . . . .	20
3.2.2	L'équilibre de Nash . . . . .	21
3.2.3	Football ou opéra . . . . .	21
3.2.4	Pile ou face . . . . .	22
3.2.5	Fonction de meilleures réponses . . . . .	22
3.2.6	Actions dominantes . . . . .	23
3.3	Jeux étendus avec information parfaite . . . . .	23
3.3.1	Définition d'un jeu étendu avec information parfaite . . . . .	23
3.3.2	Stratégies . . . . .	25
3.3.3	L'équilibre de Nash dans un jeu étendu . . . . .	26
3.3.4	Équilibre parfait de sous-jeu . . . . .	26
3.3.5	Mouvements simultanés . . . . .	29
3.3.6	Mouvements aléatoires . . . . .	30
3.4	Les jeux répétés . . . . .	31
3.4.1	Le dilemme du prisonnier répété . . . . .	31
3.4.2	Préférences . . . . .	32
3.4.3	Définition d'un jeu répété . . . . .	33
3.4.4	Jeu répété fini et infini . . . . .	33
<b>4</b>	<b>Le problème</b>	<b>35</b>
4.1	Problème de base et solution existante . . . . .	35
4.2	Le modèle existant . . . . .	36
4.2.1	La fonction de paiement . . . . .	36
4.2.2	La corruption . . . . .	38
4.2.3	Les différents paiements du modèle de départ . . . . .	38
4.2.4	Les différents cas possibles . . . . .	40
4.3	Notre sujet . . . . .	42
<b>5</b>	<b>Élaboration d'une nouvelle fonction de paiement</b>	<b>45</b>
5.1	Introduction . . . . .	45
5.2	5 propriétés pour rendre une fonction de paiement juste . . . . .	46
5.2.1	Le contexte de base de l'article [GK09] . . . . .	46
5.2.2	Le modèle de l'article [GK09] . . . . .	47
5.2.3	Les 5 propriétés . . . . .	48
5.3	Élaboration de la nouvelle fonction de paiement . . . . .	50
5.3.1	La nouvelle fonction $h_k(x)$ . . . . .	50

5.4	Vérification des propriétés . . . . .	52
5.5	Graphiques . . . . .	55
5.5.1	Forme de la fonction $h_k(x)$ . . . . .	55
5.5.2	Permettre une valeur $h_k(x)$ négative : $\gamma^-$ . . . . .	62
5.6	La fonction $g_k(x)$ . . . . .	63
5.7	Récapitulation du chapitre et remarques sur les notations . . . . .	64
5.7.1	Récapitulation . . . . .	64
5.7.2	Remarques sur les notations . . . . .	64
<b>6</b>	<b>Un nouveau cas : la corruption par un tiers</b>	<b>67</b>
6.1	Modélisation du nouveau cas . . . . .	67
6.1.1	Jeu stratégique . . . . .	67
6.1.2	Problème spécifique à ce cas . . . . .	68
6.1.3	Une nouvelle modélisation avec les jeux étendus . . . . .	68
6.1.4	Révision du cas où le service web a une mauvaise qualité de service .	69
6.1.5	Intérêt de la nouvelle modélisation . . . . .	70
6.2	Étude des nouveaux jeux . . . . .	71
6.2.1	Cas où le service web a une bonne qualité de service . . . . .	71
6.2.2	Cas où le service web a une mauvaise qualité de service . . . . .	74
<b>7</b>	<b>Étude des jeux sur le long terme</b>	<b>77</b>
7.1	La patience des services d'information . . . . .	77
7.2	La réputation des services d'information . . . . .	78
7.2.1	Le système de réputation que nous avons choisi d'utiliser : BRS . . .	78
7.2.2	Mise à jour de la réputation d'un service d'information . . . . .	80
7.2.3	Pondération par la réputation . . . . .	81
7.3	Analyse des jeux répétés . . . . .	83
7.3.1	Équilibre de Nash . . . . .	83
7.3.2	Remarques sur les jeux répétés . . . . .	83
7.3.3	Analyse intuitive du cas d'un service web malhonnête avec une mau- vaise qualité de service . . . . .	84
7.3.4	Conclusion . . . . .	85
<b>8</b>	<b>Expérimentations par simulations</b>	<b>87</b>
8.1	Présentation rapide du simulateur . . . . .	87
8.1.1	Les services web . . . . .	88
8.1.2	Les communautés . . . . .	89
8.1.3	Les services d'information . . . . .	89
8.2	Paramétrage d'une simulation et hypothèses posées . . . . .	90
8.2.1	Remarques sur le fonctionnement des simulations . . . . .	93
8.3	Résultats et analyse de cas spécifiques et pertinents . . . . .	95
8.3.1	Les paramètres par défaut . . . . .	95
8.3.2	Plan des simulations . . . . .	96
8.3.3	Simulation 1 . . . . .	98

8.3.4	Simulation 2 . . . . .	101
8.3.5	Simulation 3 . . . . .	107
8.3.6	Simulation 4 . . . . .	111
8.3.7	Simulation 5 . . . . .	112
8.3.8	Simulation 6 . . . . .	116
8.3.9	Simulation 7 . . . . .	119
8.3.10	Simulation 8 . . . . .	120
8.3.11	Simulation 9 . . . . .	128
8.3.12	Simulation 10 . . . . .	135
8.3.13	Simulation 11 . . . . .	137
8.3.14	Simulation 12 . . . . .	137
8.3.15	Simulation 13 . . . . .	139
8.4	Conclusion . . . . .	140
8.4.1	Comparaisons . . . . .	140
8.4.2	Implications des résultats . . . . .	142
<b>9</b>	<b>Conclusion</b>	<b>145</b>
<b>A</b>	<b>Le simulateur</b>	<b>151</b>
A.1	Introduction . . . . .	151
A.2	Paramétrage des agents et hypothèses faites . . . . .	151
A.3	Structure du programme . . . . .	151
A.3.1	Le package <i>extRepEntities</i> . . . . .	152
A.3.2	Le package <i>extRepGameTheory</i> . . . . .	153
A.3.3	Le package <i>utils</i> . . . . .	153
A.4	Fonctionnement global d'une simulation . . . . .	154
A.5	Interface utilisateur . . . . .	155
A.5.1	Les paramètres . . . . .	155
A.5.2	Les graphiques résultants d'une simulation . . . . .	158

# Liste des tableaux

3.1	Modélisation du dilemme du prisonnier . . . . .	19
3.2	Modélisation du dilemme du prisonnier . . . . .	20
3.3	Modélisation de la situation de choix entre deux activités pour un couple . . . . .	22
3.4	Modélisation du jeu pile ou face . . . . .	22
3.5	Stratégies possibles pour le joueur 2 . . . . .	25
3.6	Jeu stratégique joué si la femme décide de sortir. . . . .	30
3.7	Modélisation du dilemme du prisonnier . . . . .	31
4.1	Jeu lorsque $S_j$ a une bonne qualité de service. . . . .	40
4.2	Jeu lorsque $S_j$ a une mauvaise qualité de service et est honnête. . . . .	41
4.3	Jeu lorsque $S_j$ a une mauvaise qualité de service et est malhonnête. . . . .	42
5.1	Jeu lorsque $S_j$ a une bonne qualité de service. . . . .	64
5.2	Jeu lorsque $S_j$ a une mauvaise qualité de service et est honnête. . . . .	65
6.1	Jeu représentant la situation où le service web est bon et un tiers tente de corrompre les services d'information. . . . .	67
6.2	Le service web est bon et aucune tentative de corruption n'a lieu. . . . .	71
6.3	Le service web est bon mais un tiers tente de corrompre les services d'information. . . . .	72
6.4	Le service web a une mauvaise qualité de service et est honnête. . . . .	74
6.5	Le service web a une mauvaise qualité de service et est malhonnête. . . . .	75
7.1	Jeu lorsque le service web est malhonnête avec une mauvaise qualité de service. . . . .	84
8.1	Paramètres par défaut d'une simulation. . . . .	95



# Table des figures

3.1	Un exemple de jeu étendu à 2 joueurs . . . . .	25
3.2	Jeu étendu à 2 joueurs avec 4 stratégies possibles pour le joueur 2 . . . . .	25
3.3	Un des 2 sous-jeux stricts de l'exemple 2.2 . . . . .	27
3.4	L'autre sous-jeu strict de l'exemple 2.2 . . . . .	27
3.5	Jeu étendu à 2 joueurs avec 4 stratégies possibles pour le joueur 2 . . . . .	28
3.6	Backward induction (1) . . . . .	28
3.7	Backward induction (2) . . . . .	28
3.8	Backward induction (3) . . . . .	29
3.9	Maison, Football, ou Opéra . . . . .	29
3.10	Jeu étendu à 2 joueurs avec mouvement aléatoire . . . . .	30
5.1	Variation du paiement avec $a = 400$ , $b = 0$ , et $c=0,5$ . . . . .	55
5.2	Variation du paiement avec $a = 400$ , $b = 0$ , et $c=0,1$ . . . . .	55
5.3	Variation du paiement avec $a = 400$ , $b = 0$ , et $c=0,9$ . . . . .	56
5.4	Variation du paiement avec $a = 400$ . . . . .	57
5.5	Variation de $\Delta(r, \epsilon, \theta)$ en fonction de $\epsilon$ . . . . .	58
5.6	Variation de $\Delta(r, \epsilon, \theta)$ en fonction de $r$ lorsque le service web est bon et que $c = 1 - r$ . . . . .	59
5.7	Variation de $\Delta(r, \epsilon, \theta)$ en fonction de $r$ lorsque le service web est mauvais et que $c = r$ . . . . .	60
5.8	Variation de $\Delta(r, \epsilon, \theta)$ en fonction de $r$ lorsque le service web est bon et que $c = r$ . . . . .	60
5.9	Variation de $\Delta(r, \epsilon, \theta)$ en fonction de $r$ lorsque le service web est mauvais et que $c = 1 - r$ . . . . .	61
5.10	Respect de la propriété 5 . . . . .	61
5.11	Non respect de la propriété 5 . . . . .	62
5.12	Permettre une valeur $h_k(x)$ négative : $\gamma^-$ . . . . .	62
5.13	$g_k(x)$ lorsque $d = 200$ , $e = 100$ , et $k = 3$ . . . . .	63
6.1	Modélisation du cas avec tiers . . . . .	69
6.2	Modélisation du second cas avec tiers . . . . .	70
8.1	Évolution du nombre de mensonges/vérités pour la simulation 1 . . . . .	98
8.2	Répartition dans les communautés pour la simulation 1. Tous les services web ont été acceptés à tort (ils étaient mauvais). . . . .	99

8.3	Nombre de vérités/mensonges du service d'information le plus patient pour la simulation 1. On peut voir qu'il ment toujours. . . . .	99
8.4	Évolution des paiements du service d'information le plus patient pour la simulation 1. . . . .	100
8.5	Évolution de la réputation du service d'information le plus patient pour la simulation 1. . . . .	101
8.6	Évolution du nombre de mensonges/vérités pour la simulation 2. . . . .	102
8.7	Répartition dans les communautés pour la simulation 2. La plupart des services web ont été refusés à raison (ils étaient tous mauvais). . . . .	103
8.8	Nombre de vérités/mensonges du service d'information le plus patient pour la simulation 2. On peut voir qu'il dit quasiment toujours la vérité. . . . .	103
8.9	Évolution des paiements du service d'information le plus patient pour la simulation 2. . . . .	104
8.10	Évolution de la réputation du service d'information le plus patient dans chaque communauté pour la simulation 2. . . . .	104
8.11	Nombre de vérités/mensonges du service d'information 5 pour la simulation 2. On peut voir qu'il ment beaucoup plus que le premier, plus patient. . . . .	105
8.12	Évolution des paiements du service d'information 5 pour la simulation 2. Au total, il gagne moins que le premier service d'information, plus patient. . . . .	106
8.13	Évolution de la réputation du service d'information 5 dans chaque communauté pour la simulation 2. . . . .	106
8.14	Évolution du nombre de mensonges/vérités pour la simulation 3. Il est similaire à la simulation 2. . . . .	107
8.15	Répartition dans les communautés pour la simulation 3. Beaucoup de services web ont été acceptés à tort (ils étaient tous mauvais). . . . .	108
8.16	Nombre de vérités/mensonges du service d'information 5 pour la simulation 3. . . . .	109
8.17	Évolution des paiements du service d'information 5 pour la simulation 3. . . . .	109
8.18	Évolution des paiements du service d'information 1 pour la simulation 3. . . . .	110
8.19	Évolution du nombre de mensonges/vérités pour la simulation 4. Il y a très peu de mensonges . . . . .	111
8.20	Nombre de mensonges/vérités pour la simulation 5. . . . .	112
8.21	Nombre de mensonges/vérités pour la simulation 5 bis. Les services d'information pensent un peu plus à long terme. . . . .	113
8.22	Nombre de vérités/mensonges du service d'information 1 pour la simulation 5. . . . .	113
8.23	Évolution de la réputation du service d'information le plus patient pour la simulation 5 . . . . .	114
8.24	Évolution des paiements reçus par le service d'information le plus patient pour la simulation 5. . . . .	115
8.25	Graphique des paiements reçus par le service d'information le plus patient à chaque tout pour la simulation 5. . . . .	115
8.26	Nombre de mensonges/vérités pour la simulation 6. . . . .	116
8.27	Nombre de mensonges/vérités du service d'information le plus patient pour la simulation 6. . . . .	117

8.28	Évolution de la réputation du service d'information le plus patient pour la simulation 6. . . . .	117
8.29	Évolution des paiements du service d'information le plus patient pour la simulation 6. . . . .	118
8.30	Graphique des paiements du service d'information le plus patient pour la simulation 6. . . . .	118
8.31	Nombre de mensonges/vérités pour la simulation 7. . . . .	119
8.32	Nombre de mensonges/vérités pour la simulation 8. . . . .	120
8.33	Répartition des services web dans les communautés pour la simulation 8. . .	121
8.34	Répartition des services web dans les communautés lorsque les mensonges sont plus conséquents. . . . .	121
8.35	Nombre de mensonges/vérités du premier service d'information (patience = 1) pour la simulation 8. . . . .	122
8.36	Évolution de la réputation du premier service d'information pour la simulation 8. . . . .	122
8.37	Cumulation des différents paiements accordés au premier service d'information pour la simulation 8. . . . .	123
8.38	Graphique des paiements accordés au premier service d'information pour la simulation 8. . . . .	123
8.39	Nombre de mensonges/vérités du deuxième service d'information (patience = 1) pour la simulation 8. . . . .	124
8.40	Évolution de la réputation du deuxième service d'information pour la simulation 8. . . . .	124
8.41	Cumulation des différents paiements accordés au deuxième service d'information pour la simulation 8. . . . .	125
8.42	Graphique des paiements accordés au deuxième service d'information pour la simulation 8. . . . .	125
8.43	Nombre de mensonges/vérités du cinquième service d'information (patience = 0,8) pour la simulation 8. . . . .	126
8.44	Évolution de la réputation du cinquième service d'information pour la simulation 8. . . . .	127
8.45	Cumulation des différents paiements accordés au cinquième service d'information pour la simulation 8. . . . .	127
8.46	Nombre de mensonges/vérités pour la simulation 9. . . . .	128
8.47	Nombre de mensonges/vérités du premier service d'information (patience = 1) pour la simulation 9. . . . .	129
8.48	Évolution de la réputation du premier service d'information pour la simulation 9. . . . .	129
8.49	Cumulation des différents paiements accordés au premier service d'information pour la simulation 9. . . . .	130
8.50	Nombre de mensonges/vérités du cinquième service d'information (patience = 0,8) pour la simulation 9. . . . .	131

8.51 Évolution de la réputation du cinquième service d'information pour la simulation 9. . . . .	131
8.52 Cumulation des différents paiements accordés au cinquième service d'information pour la simulation 9. . . . .	132
8.53 Nombre de mensonges/vérités du dixième service d'information (patience = 0,6) pour la simulation 8. . . . .	133
8.54 Évolution de la réputation du dixième service d'information pour la simulation 8. . . . .	134
8.55 Cumulation des différents paiements accordés au dixième service d'information pour la simulation 9. . . . .	134
8.56 Nombre de mensonges/vérités pour la simulation 10. . . . .	135
8.57 Répartition des services web dans les communautés pour la simulation 10 lorsque les mensonges sont petits. . . . .	136
8.58 Répartition des services web dans les communautés pour la simulation 10 lorsque les mensonges sont gros. . . . .	136
8.59 Nombre de mensonges/vérités pour la simulation 11. . . . .	137
8.60 Répartition des services web dans les communautés pour la simulation 11. . . . .	138
8.61 Nombre de mensonges/vérités pour la simulation 12. . . . .	138
8.62 Nombre de mensonges/vérités pour la simulation 13. . . . .	139
8.63 Pourcentage de vérité/ mensonge pour les 5 premières simulations. . . . .	141
8.64 Pourcentage de vérité/ mensonge pour les simulations 7 à 9. . . . .	141
8.65 Pourcentage de vérité/ mensonge pour les simulations 12 et 13. . . . .	142
8.66 Comparaison des décisions des communautés. . . . .	143
A.1 L'écran des paramètres sur les populations d'agents . . . . .	156
A.2 L'écran des paramètres sur les paiements . . . . .	157
A.3 L'écran des paramètres supplémentaires . . . . .	157
A.4 L'écran des résultats généraux sur les QoS des services web . . . . .	158
A.5 L'écran des résultats généraux sur le nombre de mensonges et de vérités . . . . .	159
A.6 L'écran des résultats généraux sur les décisions des communautés . . . . .	159
A.7 L'écran d'un service d'information sur les choix à chaque participation au conseil . . . . .	160
A.8 L'écran d'un service d'information sur l'évolution de sa réputation au fil des itérations et par communauté . . . . .	160
A.9 L'écran d'un service d'information sur l'évolution de ses paiements au fil des itérations . . . . .	161
A.10 L'écran d'un service d'information sur ses paiements à chaque itérations . . . . .	161

# Chapitre 1

## Introduction

A l'heure du « tout connecté », l'importance des technologies web est de plus en plus flagrante, de par la recherche active sur le sujet, et leur utilisation grandissante. Depuis plusieurs années, le développement du web s'accroît de manière exponentielle, et de simples sites statiques servant de miroir à une entreprise, ou présentant certains contenus sur un thème instructif ou divertissant, nous sommes passés à un web collaboratif (dit 2.0), où créateurs et lecteurs de contenu se mélangent. Dans ce contexte, les services web ont été largement mis en avant, notamment par le W3C ([Laf]). Aujourd'hui, le web constitue certes une véritable collection de données, mais aussi de fonctions et de services mis à disposition d'un large public via un serveur web.

Les services web sont en quelque sorte les représentants de la coopération distribuée. Ils sont capables d'interactions et de compositions, et sont amenés à coopérer dans le but d'améliorer l'efficacité des réseaux d'entreprises ([BK11a]). Ils permettent la création de systèmes distribués en usant des technologies web. Les services web nous apportent une manière standard d'invoquer une application distante et d'en récupérer les résultats à travers le web. Ils ont de nombreux avantages dont par exemple :

- le rassemblement de toutes les données nécessaires au fonctionnement de plusieurs applications en un seul et même endroit ainsi que l'accès à ces données via un service web ;
- la capacité de rassembler du hardware très puissant mais aussi très coûteux que les utilisateurs ne peuvent pas acquérir, mais qu'ils pourront utiliser via le service web en contrepartie d'un paiement ;
- l'utilisation massive et simultanée d'un même service partout dans le monde mais hébergé en un seul endroit ;
- la mise à jour unique du service ;
- l'interaction automatique entre différents services web, notamment pour la mise à jour de données diverses, sans l'intervention humaine ;
- la possibilité de rendre des services interopérables malgré la différence d'environnement (système d'exploitation, etc.).

Ce mémoire a pour thème les services web et la coopération de ceux-ci. Dans la pre-

mière section de ce chapitre, nous allons poser plus précisément le contexte dans lequel notre travail s'inscrit. Dans la deuxième section, nous exposerons notre sujet de manière générale ainsi que nos motivations. Dans la dernière section, nous décrirons la méthodologie employée pour la construction de ce mémoire.

## 1.1 Le contexte

Lorsque plusieurs services web proposent le même type de service, il peut être intéressant pour eux de s'unir dans une communauté de services web. Cette communauté répartira les clients et les charges, et permettra d'offrir une plus grande capacité de service aux clients dans plusieurs endroits du monde (si les services web sont présents dans différents pays). Le problème pouvant apparaître est celui de la qualité d'un service web. Si un service web médiocre se joint à la communauté, celle-ci s'en retrouvera affaiblie, et perdra éventuellement des clients. La communauté pourrait donc décider de demander l'avis d'autres types de service, les services d'information, qui donneraient des évaluations à propos des différents services web désirant entrer dans la communauté. Ceux-ci peuvent être malhonnêtes s'ils sont corrompus par lesdits services web. Le but principal de ce mémoire est d'élaborer un système d'incitation qui limiterait les actes malhonnêtes.

### 1.1.1 Services web, communautés, et réputation

Nous allons commencer par parler de services web, de communauté de services web et de réputation.

#### Service web

Un service web peut être défini comme suit : « *Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et de manière synchrone. Dans sa présentation la plus générale, un service web se concrétise par un agent, réalisé selon une technologie informatique précise, par un fournisseur du service. Un demandeur, à l'aide d'un agent de requête, utilise ce service. Fournisseur et demandeur partagent une même sémantique du service web, tandis qu'agent et agent de requête partagent une même description du service pour coordonner les messages qu'ils échangent.* » [ca]

La motivation principale dans le développement des services web est le développement de processus inter-entreprises faiblement couplés (applications B2B). Les services web peuvent être utilisés sans se soucier de la manière dont ils sont implémentés ou de l'endroit où ils sont hébergés. Comme dans les recherches précédemment faites (entre autres : [BK11b], [CK11], [BK11a]), nous considérons les services web comme des agents rationnels désirant maximiser leurs profits. En regardant le contexte d'un point de vue « système multi-agents », nous pouvons considérer les services web comme des agents ayant chacun

leurs propres intérêts. Il est donc nécessaire d'étudier et de mettre en place des systèmes permettant la coordination, la coopération et la négociation entre ces agents.

### **Communauté de services web**

Comme dit précédemment, les services web peuvent être perçus comme des agents évoluant dans un même système et ayant chacun leurs propres intérêts. Dans ce but, ils pourraient se réunir afin de former des communautés qui leur permettent de collaborer entre eux et ainsi accroître leurs intérêts. Des services web peuvent ainsi se réunir pour former des communautés, afin de collaborer entre eux. Dans l'article [BK11a], deux aspects sont présentés vis-à-vis de la coopération : la composition et la collaboration de services web. Bien que ces termes soient liés, une légère distinction est faite. La composition de services web a pour but de réaliser une tâche en la décomposant en sous-tâches et en distribuant ces dernières à plusieurs services web spécialisés. La simple collaboration de services web signifie quant à elle la création d'une communauté de services web capable de gérer efficacement des tâches, en les distribuant aux services web membres, selon leur capacité (leur charge). L'objectif de la collaboration est d'augmenter l'efficacité du système distribué. Dans nos communautés de services web, il s'agit d'une collaboration sans composition. Les services web présents dans une communauté ont tous la même fonction.

L'efficacité d'un service web peut être définie comme le nombre de requêtes qu'il peut recevoir et gérer. Celle-ci dépend de plusieurs paramètres. Lorsque plusieurs services web forment une communauté, les requêtes ne sont plus adressées à un service web en particulier mais à la communauté entière. Un service web dit maître va alors distribuer les requêtes au membres de la communauté. Les services web peuvent avoir intérêt à former des communautés. D'une part, un service web très utilisé et risquant d'arriver à saturation peut se décharger de certaines requêtes en rejoignant une communauté. D'autre part, un service web capable de gérer davantage de requêtes qu'il n'en reçoit peut rejoindre une communauté bien connue pour accroître le nombre de requêtes reçues.

Dans une communauté, on distingue donc le service maître et les autres. Le maître a la responsabilité de gérer la communauté. Il doit attirer de nouveaux services web au sein de la communauté, convaincre ceux qui y sont de rester, et répondre aux requêtes des utilisateurs. Une analyse de l'efficacité des services web collaborant entre eux peut être trouvée dans [BK11a].

### **Réputation**

La réputation d'un service web ou d'une communauté de services web est une valeur qui représente sa fiabilité, sa capacité de répondre correctement à une requête dans un temps relativement court. La réputation est intrinsèquement liée au concept de qualité de service (QoS). La réputation est calculée à partir d'évaluations de la qualité de service (données par les services d'information). Ce n'est donc pas la qualité de service mais une approximation de celle-ci. Avoir une bonne réputation est essentiel puisque cela déterminera le rang du service web ou de la communauté dans les classements réalisés par les algorithmes de

référencement.

Ainsi, des systèmes de réputation sont utilisés sur les sites de commerces électroniques. Le système d'étoiles d'Ebey en est un exemple. Bien qu'il ne permette pas forcément de détecter les fraudes (un vendeur pourrait payer d'autres personnes pour faire de faux achats et le gratifier avec beaucoup d'étoiles), le système semble suffisant pour mettre les utilisateurs en confiance et permettre les transactions. Ce système, bien que simple, semble déjà suffisant pour garantir une certaine fiabilité : un vendeur honnête ne voudra pas voir sa réputation diminuer en vendant un objet abîmé ou inexistant, tandis qu'un client achètera plus facilement chez les vendeurs paraissant dignes de confiance. Chacun y gagne. L'augmentation du nombre de services web et de communautés ne peut que provoquer l'augmentation de l'utilisation de mécanismes de réputation afin de garantir une certaine fiabilité.

### **Un exemple concret de communauté**

Nous avons choisi d'illustrer l'intérêt de la collaboration de services web par un exemple concret. Celui-ci consiste en une communauté de téléchargements payants. Dans cette communauté, plusieurs services web mettent à disposition leurs serveurs, leurs bases de données et donc de la place de stockage, leur bande passante, et leur proximité par rapport à un client. Les clients sont des personnes désirant héberger du contenu d'enseignement (des vidéos de cours, des articles, des slides, des exercices et solutions, etc.). Nous posons l'hypothèse que le contenu est massivement constitué de vidéos longues concernant des cours (par exemple 24h de vidéos où l'on voit le professeur donner cours, ou bien de l'audio). L'intérêt pour les utilisateurs est donc d'avoir une grande capacité de stockage, une bonne bande passante, une certaine flexibilité à propos de l'agencement des données sur le serveur. De plus, ce projet a pour but d'être mondialement utilisé, et la présence des services web est nécessaire afin d'avoir une certaine réactivité et rapidité d'opération.

Dès lors, chaque service web a ses caractéristiques en termes de localisation, bande passante, capacité de stockage, mais aussi, et cela est important, une fiabilité, et une réputation (sa bande passante annoncée est-elle bien réelle, ses données stockées sont-elles bien sécurisées et non détériorées, etc.).

Bien sûr, un service web isolé ne servira pas à grand chose ; il voudra sans aucun doute entrer dans une communauté de services web offrant des clients, des interactions entre services web pour faciliter la satisfaction de ceux-ci, une certaine stabilité, une bonne réputation, une aide ou un remplacement en cas de panne, etc. Notre communauté, elle, aimerait bien avoir beaucoup de services web, pour attirer une large quantité de clients partout dans le monde, et leur promettre une bonne qualité de service. De plus, cela offrirait plus de capacité de stockage, plus de territoire couvert, une meilleure réactivité due à la proximité d'un service web, etc. Tous ces facteurs contribuent à la réputation de la communauté. Plus les services sont rapides et corrects, plus les clients seront satisfaits.

Elle a une réputation à maintenir et en fonction de celle-ci, plus ou moins de clients la solliciteront. Elle a donc tout intérêt à conserver un haut degré de qualité de service car sa réputation dépendra de celle-ci.

## 1.2 Le sujet

Dans cette section, nous allons présenter rapidement notre sujet. Nous le développerons davantage dans le chapitre 4.

### 1.2.1 Le problème

Le problème sur lequel nous allons travailler est celui de l'entrée d'un nouveau service web dans une communauté. Pour ne pas entacher la réputation de la communauté, le maître aimerait connaître la réputation du service web pour savoir s'il peut l'accepter au sein de sa communauté ou non. Pour cela, le maître va demander des informations à un groupe de services web spécifiques appelés services d'information dont le rôle est de noter les différents services web. Le maître va payer les services d'information de manière à ce que ceux-ci soient enclins à transmettre une information correcte. En effet, rien ne garantit que les services d'information transmettent la vérité, ils pourraient mentir, s'ils reçoivent des pots-de-vin de la part des services web. Le but est d'essayer de définir des bonnes pratiques, de chercher des incitants, des récompenses ou des punitions, afin que chaque service d'information ait un comportement honnête.

Le processus suivi lors de la demande d'entrée d'un service web dans une communauté est le suivant. Lorsqu'un service web désire entrer dans la communauté, il annonce au maître de celle-ci sa propre réputation. Le problème est évidemment qu'il pourrait décider de mentir et de la surévaluer. Le maître de communauté s'en apercevra sans doute par la suite, mais il sera trop tard, un accord aura peut-être été conclu, des clients seront déjà partis et la réputation de la communauté toute entière aura baissé. Il est donc important que cela n'arrive pas. Le maître de communauté, soucieux de préserver une bonne réputation, fait alors appel aux services d'information. Ceux-ci ont un rôle de conseiller, ils vendent les évaluations qu'ils font des services web. Ils connaissent chacun des services web avec une bonne précision de leur réputation. Néanmoins, les services d'informations ont des évaluations légèrement différentes de chaque service web. En effet, chaque service d'information possède son propre mécanisme de notation. Les services d'information aussi sont dotés d'une réputation, car les communautés interagissant avec eux seront plus ou moins satisfaites selon la fiabilité des informations qu'ils transmettent.

Bien que l'on puisse penser que cette vérification suffit, il subsiste plusieurs problèmes. Un service d'information pourrait avoir intérêt à mentir si le service web désirent entrer dans la communauté le paie suffisamment pour intervenir en sa faveur lors de la demande de conseil de la part du maître de communauté. Cela arrive évidemment uniquement dans le cas où le service web est malhonnête et qu'il a une mauvaise réputation, vu que nous

posons l'hypothèse et qu'il semble évident qu'un service web avec une bonne réputation n'aura pas d'intérêt à mentir. En effet, il suffit, nous en reparlerons plus tard, d'avoir une réputation au moins égale à celle requise pour entrer dans la communauté.

En plus de ce cas déjà ennuyeux, plusieurs services web déjà membres de la communauté pourraient être en conflit avec le demandeur. Cela pourrait se produire si, par exemple, ils partagent le même territoire et qu'il en résulterait une perte due au partage de clients. Ceux-ci interviendraient en payant à nouveau les services d'information pour démonter la réputation du demandeur bien que celle-ci soit réellement bonne.

Dans les deux cas, il y a possibilité de mensonge et de surcroît perte de performance. Si le service web est mauvais, mais parvient malgré tout à rentrer dans la communauté, il fera baisser la réputation de celle-ci de par sa mauvaise qualité de service ; les clients servis par celui-ci partiront et la communauté s'en trouvera affaiblie. Elle ne veut pas de mauvais services web. Si le service web avait en revanche amélioré la réputation de la communauté sans être accepté, cette dernière et le service web auraient tous les deux été lésés.

Ces différentes tentatives de corruption peuvent aisément se comprendre dans l'exemple de la communauté de services de téléchargement. Admettons qu'un service de téléchargement situé sur un serveur à Paris désire entrer dans la communauté. Celui-ci pourrait concurrencer le service situé à Bruxelles, déjà dans la communauté, qui est en général chargé de répondre aux requêtes des clients européens. Ce service web pourrait être tenté d'offrir aux services d'information une certaine somme d'argent afin que ceux-ci mentent sur les notations qu'ils accordent chacun au service web candidat. Bien sûr cela se passe dans des communautés où chaque agent aimerait avoir les meilleures relations et services de la part des autres, en échange des siens. De même, un service web peu réputé désirent entrer dans une communauté pour avoir plus de clients, pourrait tenter de corrompre les services d'information pour que ceux-ci réévaluent les notations qu'ils lui accordent à la hausse. Le comportement des services d'information dépend de la balance entre l'incitation à frauder et celle à être honnête.

Le travail effectué par les stagiaires des FUNDP de l'année dernière a abordé ce problème. L'article [BK11b] en expose les résultats. C'est sur cet article que nous nous baserons pour démarrer notre travail.

### **1.2.2 Notre recherche**

Notre recherche, précisément, s'est portée sur l'élaboration et l'optimisation d'une fonction de paiement déterminant le montant qu'un service d'information doit recevoir du maître de communauté pour lui avoir transmis son information. Cette fonction de paiement a pour but d'inciter les services d'informations à être honnêtes, malgré leur avidité de paiements (ce sont des agents rationnels qui tentent de maximiser leurs profits).

### 1.3 Méthodologie

Nous avons effectué notre recherche dans un certain cadre théorique. Celui-ci concerne les systèmes multi-agents, la théorie des jeux, ainsi que les systèmes de réputation et d'incitation. Nous allons commencer par une courte présentation des systèmes multi-agents, paradigme qui sera souvent utilisé dans la suite de ce travail. Les mécanismes de réputation et d'incitation dont nous parlerons dans le chapitre 2 s'inscrivent dans le paradigme des systèmes multi-agents. Dans le chapitre 3, nous présenterons la théorie des jeux, un outil important que nous utiliserons dans la suite, en nous cantonnant à exposer uniquement les parties nécessaires à la compréhension du mémoire, la théorie des jeux pouvant faire office d'un livre à elle seule.

#### 1.3.1 Les systèmes multi-agents

DÉFINITION 1.1. Un système multi-agents est un système composé d'entités appelées agents, situés dans un certain environnement, et interagissant selon certaines relations. Un agent est caractérisé par le fait qu'il est autonome. Il s'agit d'une entité abstraite pouvant représenter un programme, un processus, une organisation, ou encore un être humain.

Les systèmes multi-agents ont un champ d'application relativement large, notamment en intelligence artificielle distribuée. Le paradigme des systèmes multi-agents peut être utilisé pour modéliser des sociétés, ce qui lui donne une utilité dans le domaine des sciences humaines. Nous utilisons les systèmes multi-agents comme cadre théorique pour notre travail. La théorie des jeux et les différentes techniques utilisées comme les mécanismes de réputation ou d'incitation s'inscrivent toutes dans ce cadre.

#### 1.3.2 Les systèmes de réputation

Comme précisé dans l'introduction, nous allons utiliser des systèmes de réputation. Un système de réputation est un système permettant d'attribuer une réputation à un agent et de la partager. Dans notre cas, le fait que des services d'information notent les services web constitue un système de réputation. Une communauté pourra faire appel à différents services d'information pour connaître la réputation d'un service web désirant y entrer. Pour rappel, la réputation d'un service web reflète sa qualité de service. Un second système de réputation vient se greffer au premier. Il s'agit d'un système attribuant une réputation aux services d'information eux-mêmes. Nous donnerons plus de détails sur ce point dans le chapitre 7.

Il existe plusieurs manières de construire un système de réputation. La réputation d'un certain agent est-elle globale ou est-elle calculée différemment par chaque agent du système ? Si elle n'est pas globale, un agent va-t-il calculer la réputation d'un autre agent selon ses propres expériences passées, ou va-t-il faire appel à des « témoins » ? S'il fait appel à des témoins, comment va-t-il les sélectionner ? De manière plus générale, comment va-t-on propager la réputation ? Il existe une multitude de systèmes de réputation. Le chapitre 2 présentera des systèmes de réputation en commençant par ceux utilisés par les plateformes d'e-commerce (Ebay dans notre cas). Nous exposerons ensuite le système BRS

dans la section 2.4 que nous utiliserons nous-mêmes pour calculer la réputation des services d'information.

### **1.3.3 La théorie des jeux**

La théorie des jeux constitue un très bon cadre théorique afin d'étudier le comportement d'agents dans un système. La théorie des jeux est utile pour modéliser certains systèmes, et nous l'utiliserons dans la suite de ce mémoire. Il existe plusieurs types de « jeu » correspondant à plusieurs types de modélisation, plus ou moins utiles selon la situation à analyser : jeu stratégique avec information parfaite, jeu bayésien et pré-bayésien, jeu étendu avec information parfaite ou imparfaite, jeux répétés, etc. Le chapitre 3 présentera les éléments de la théorie des jeux nécessaires à la compréhension de la suite du mémoire.

### **1.3.4 Le problème**

Après avoir exposé les différentes techniques de résolution, nous expliciterons en détail le problème constituant le sujet de notre mémoire dans le chapitre 4. Nous présenterons ensuite un mécanisme d'incitation existant, qui servira de base à notre travail.

### **1.3.5 Élaboration de la fonction de paiement**

Nous verrons dans le chapitre 5 comment nous avons élaboré la fonction de paiement et pourquoi. Le principal problème consistera à essayer de la rendre la plus juste possible.

### **1.3.6 La corruption par un tiers**

Le chapitre 6 se concentre sur le cas où un service web est bon, et qu'un tiers est contre l'acceptation de ce service web dans sa communauté. Celui-ci essayera de corrompre les services d'information afin de dissuader le maître de communauté d'accepter ce nouveau service web.

### **1.3.7 Étude des jeux sur le long terme**

Afin d'enrichir notre modèle construit jusqu'à présent, nous décidons de modifier celui-ci afin de permettre l'étude à long terme des incitations à être honnête. En effet, le but principal de notre mémoire est de découvrir un mécanisme d'incitation qui, à long terme, motive les services d'information à être honnêtes. Cela fera l'objet du chapitre 7.

### **1.3.8 Expérimentation**

Nous terminerons ce mémoire en montrant les résultats de nos expérimentations. Nous exposerons les résultats de différentes simulations dans le chapitre 8. Nous discuterons ensuite de l'efficacité de notre mécanisme d'incitation.

## Chapitre 2

# Les mécanismes de réputation

Dans le domaine des systèmes multi-agents, de nombreux travaux menant à des publications ont été réalisés sur les modèles de réputation entre agents et sur la manière de calculer la confiance que l'on peut accorder à un agent du système. Nous allons voir, dans cette section, plusieurs modèles de réputation bien connus. Parmi ces systèmes, une grande majorité a été conçue dans le contexte du commerce électronique, où des agents dits « acheteurs » attribuent une certaine confiance en des agents dits « vendeurs ». Le but ici n'est pas de présenter l'ensemble des modèles de réputation existants, mais uniquement ceux que nous considérons comme importants par rapport à notre travail. Nous présenterons tout d'abord les modèles de réputation utilisés sur les plateformes de commerce électronique. Nous en aborderons les principaux défauts et passerons rapidement en revue deux améliorations, le système Sporas et le système Histos [Gio99]. Nous verrons ensuite le système BRS [eRI02], car nous l'utiliserons en partie nous-mêmes dans la suite de ce mémoire. Parmi les auteurs les plus prolifiques dans ce domaine, et dont nous ne parlerons pas ici, on peut citer Jurca, auteur de nombreux articles sur les systèmes de réputation et d'incitation. Quelques éléments de sa thèse [Jur07] seront repris ici. Pour plus d'informations sur son travail et sur les systèmes de réputation, les articles [jeBF03], [jeBF04], et [jeBF07] peuvent être lus. Pour une rétrospective (qui date de 2005) sur les systèmes de réputation, l'article [eCS05] est très intéressant.

### 2.1 But et principe d'un système de réputation

Dans un système multi-agents, les agents le composant seront amenés à interagir entre eux, à négocier. Il peut être utile pour tous les agents de savoir si un partenaire dans une négociation est fiable ou non, c'est-à-dire si on peut avoir confiance en lui ou non.

Pour créer un système de réputation, il faut en premier lieu définir la manière dont on va calculer la fiabilité, la valeur de confiance d'un agent. Ensuite, il faut définir comment un agent va prendre connaissance de la valeur de confiance ou de réputation d'un autre agent. Un système de réputation possède donc deux grands aspects :

- un *moteur* qui calcule la valeur de réputation d'un agent à partir de certaines données, dont par exemple les feedbacks des autres agents ;

- un *mécanisme de propagation* qui permet aux agents d’obtenir des valeurs de réputation. Il en existe deux grandes approches : l’approche centralisée, où chaque agent a une valeur de réputation globale stockée sur un serveur centralisé, et l’approche décentralisée, où chaque agent garde et gère la valeur de réputation des autres agents.

EXEMPLE 2.1. Imaginons un système composé de trois agents  $A$ ,  $B$  et  $C$ . Dans une approche centralisée, la réputation de ces trois agents est globale ; il est possible de dire que dans le système,  $R(A) = 0,8$ ,  $R(B) = 0,7$ , et  $R(C) = 0,9$  par exemple. Dans une approche décentralisée, chaque agent calcule la réputation des autres lui même. On pourrait avoir  $R_B(A) = 0,80$  et  $R_C(A) = 0,85$ , où  $R_B(A)$  est la réputation de  $A$  selon  $B$ .

## 2.2 Les modèles de réputation utilisés par les sites de commerce électronique

Les exemples les plus concrets de systèmes de réputation sont sans aucun doute ceux utilisés par les plateformes de commerce électronique. Ceux-ci font partie des systèmes de type centralisés. Chaque utilisateur de la plateforme possède une réputation stockée sur un serveur et accessible à tous.

Lors de la conclusion d’une vente sur une plateforme de commerce électronique, l’information que possède l’acheteur sur le vendeur peut être un facteur de décision crucial. Dans une telle situation, l’utilité d’un système de réputation nous paraît directement évidente. Les plateformes de commerces électroniques comme Ebay ([www.ebay.be](http://www.ebay.be)) en ont d’ailleurs développés. Sur ces plateformes, on retrouve des acheteurs et des vendeurs. Lorsqu’une transaction a lieu entre un acheteur et un vendeur, ils vont se noter l’un l’autre. Un mécanisme est mis en place afin qu’à chaque vendeur, et éventuellement à chaque acheteur, soit attribuée une note globale reflétant sa réputation, sa valeur de confiance. Ces modèles sont en général relativement simples. Afin de présenter ce type de modèle de réputation, nous allons voir celui mis en place par la plateforme d’enchères Ebay. Cette plateforme a été étudiée par Dellacoras en 2003 [Del03]. Le système que nous présentons ici peut donc différer du système réellement utilisé par Ebay à l’heure actuelle. Cependant, l’important n’est pas de présenter le modèle d’Ebay, mais un exemple de modèle de réputation concret dans le cadre des plateformes de commerce électronique.

Dans le système d’Ebay, la bonne coopération entre les utilisateurs (vendeurs et acheteurs) est basée sur un mécanisme de « feedback » très simple. Après chaque transaction, les deux participants sont invités à se noter l’un l’autre sur ladite transaction. Trois notations sont possibles : positive, neutre, ou négative. Ces notations représentent une appréciation du niveau de qualité de service que se font les utilisateurs à propos du vendeur. L’historique des notations de chaque utilisateur est rendue publique et est consultable par tous les autres utilisateurs de la plateforme. De plus, deux notations globales sont attribuées à chaque utilisateur. La première est un nombre de points. Celui-ci correspond au nombre d’évaluations positives reçues moins le nombre d’évaluations négatives. La deuxième est le taux d’évaluations positives sur les douze derniers mois. Nous sommes donc dans un type

de modèle centralisé, où la réputation est une propriété globale. Chaque agent possède une valeur de réputation unique qui provient de l'agrégation des notations données par tous les autres agents au cours d'une certaine période, douze mois en l'occurrence.

### 2.3 Points faibles du type de modèles précédent et améliorations possibles

Bien que le modèle de réputation d'Ebey semble être suffisant pour générer une certaine confiance entre les utilisateurs [Del03], on peut pointer quelques problèmes d'efficacité. Par exemple, un utilisateur ayant une mauvaise réputation peut quitter le système et se réinscrire sous une autre identité. Le modèle vu dans la section 2.2 ne prend pas non plus en compte le fait qu'un utilisateur peut changer de comportement. En effet, un utilisateur peu fiable pourrait très bien changer et devenir fiable et vice-versa. Nous allons voir deux modèles améliorés. Sporas [Gio99] dans le cadre de communautés d'agents faiblement connectées, et Histos [Gio99] dans le cadre des communautés fortement connectées.

#### 2.3.1 Sporas : un modèle de réputation pour les communautés faiblement connectées

Sporas se veut une amélioration des modèles de réputation des plateformes de commerce électronique. Il est basé sur les principes suivants :

1. les nouveaux utilisateurs démarrent avec une valeur de réputation minimale, et construisent leur réputation pendant leur activité sur le système ;
2. la valeur de réputation d'un utilisateur ne peut jamais tomber en dessous de celle d'un nouvel utilisateur ;
3. après chaque transaction, la valeur de réputation de chaque utilisateur est mise à jour en accord avec le feedback fourni par les autres parties ;
4. deux utilisateurs ne peuvent se noter qu'une seule fois. Si plusieurs transactions ont lieu entre deux mêmes utilisateurs, seul le dernier feedback est pris en compte ;
5. les utilisateurs ayant une haute valeur de réputation subissent une modification moindre de celle-ci après chaque feedback ;
6. les notations récentes ont un plus grand poids dans le calcul de la valeur de réputation d'un utilisateur.

Ce système a plusieurs avantages. Le premier est que, la réputation d'un utilisateur ne pouvant jamais tomber plus bas que celle d'un nouvel utilisateur, il n'y a donc aucun intérêt à changer d'identité. Le fait que seul le plus récent feedback d'un utilisateur soit pris en compte empêche le « flood » de mauvaises notes de la part d'un utilisateur malhonnête. Enfin, le fait que les feedbacks récents aient priorité prend en compte le caractère possiblement changeant du comportement de l'utilisateur noté.

### 2.3.2 Histos : un modèle de réputation pour les communautés fortement connectées

Dans Sporas, la valeur de réputation reste une valeur globale. C'est la seule information qu'un agent possède pour juger de la confiance qu'il peut accorder à un autre agent. Histos est un système de notations deux à deux. Un graphe est construit, où les sommets représentent les agents, et les arcs représentent la notation la plus récente qu'un agent donne à un autre. S'il existe un chemin entre deux agents, il est alors possible de calculer une valeur de réputation plus personnalisée. Un certain agent n'aura pas une valeur de réputation globale, il possédera une valeur de réputation différente pour chaque agent du système. Histos fournit une méthode spécifique de calcul de la valeur de réputation d'un agent (au cas où plusieurs chemins existeraient par exemple), ce système est intéressant dans le sens où il montre un exemple possible de système de réputation décentralisé.

## 2.4 Le système BRS

Le système BRS [eRI02], ou Beta Reputation System est un système de calcul de réputation probabiliste. Nous l'utiliserons en partie dans la suite de ce mémoire, d'où l'importance de le présenter ici.

### 2.4.1 La fonction Beta

En mathématique, la fonction de densité de probabilité beta peut être utilisée pour représenter la distribution d'événements binaires. Le système BRS utilise cette fonction afin de calculer une valeur de réputation à partir de feedbacks. La fonction beta est une famille de fonctions dont chaque membre est déterminé par deux paramètres :  $\alpha$  et  $\beta$ . La distribution beta  $f(p|\alpha, \beta)$  est exprimée en utilisant la fonction gamma ( $\Gamma$ ). On a :

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1}$$

avec :

$$0 \leq p \leq 1, \alpha > 0, \beta > 0, p \neq 0 \text{ si } \alpha < 1, \text{ et } p \neq 1 \text{ si } \beta < 1$$

L'espérance mathématique de cette fonction est :

$$E(p) = \frac{\alpha}{\alpha + \beta}$$

Considérons un processus avec deux événements possibles  $\{x, \bar{x}\}$ . Appelons  $i$  le nombre d'événements  $x$  observés, et  $j$  le nombre d'événements  $\bar{x}$  observés. La fonction de densité de la probabilité d'observer  $x$  dans le futur peut être exprimée comme une fonction sur les observations passées en posant :

$$\alpha = i + 1, \text{ et } \beta = j + 1, \text{ avec } r, s \geq 0$$

Si par exemple, l'événement  $x$  a été observé 8 fois, et l'événement  $\bar{x}$ , 5 fois, on peut tracer une courbe représentant la fonction  $f(p|9, 6)$ , qui exprime la probabilité incertaine que

l'événement  $x$  survienne pendant les observations futures. L'espérance mathématique de cette fonction est :

$$E(p) = \frac{9}{9+6} = 0,6$$

Cette valeur représente la fréquence la plus probable d'apparition de l'événement  $x$  dans le futur.

Il faut être vigilant par rapport au fait que  $f(p|\alpha, \beta)$  ne représente pas la probabilité que l'événement  $x$  survienne. Elle représente la probabilité que la probabilité que l'événement  $x$  survienne soit égale à  $p$ . On peut dire qu'il s'agit d'une probabilité de second ordre, alors que  $p$  est une probabilité de premier ordre. Puisque  $p$  est continue sur  $[0, 1]$ , la probabilité de second ordre  $f(p|\alpha, \beta)$  pour une valeur de  $p$  est infiniment petite. La probabilité de second ordre n'a donc un sens que si on la calcule pour un intervalle de  $p$ , ou si on prend son espérance mathématique. Le calcul de la valeur de réputation d'un agent sera basé sur cette espérance mathématique.

### 2.4.2 Calcul de la réputation d'un agent

A chaque fois qu'une transaction a lieu entre deux agents, chaque agent donne une note binaire à l'autre, positive (+), ou négative (-). On fixe alors  $i$  comme le nombre de notes positives qu'a reçues un agent, et  $j$  le nombre de notes négatives.

Si  $i_T^X$  et  $j_T^X$  représentent respectivement le montant de feedbacks positifs et négatifs sur un agent  $T$ , fourni par un agent ou un ensemble d'agents  $X$ , alors la fonction de réputation de  $T$  par  $X$  est :

$$\phi(p|i_T^X, j_T^X) = \frac{\Gamma(i_T^X + j_T^X)}{\Gamma(i_T^X)\Gamma(j_T^X)} p^{i_T^X-1} (1-p)^{j_T^X-1}$$

avec :

$$0 \leq p \leq 1, i_T^X \geq 0, \text{ et } j_T^X \geq 0$$

Par raccourci, nous noterons  $\phi_T^X$  au lieu de  $\phi(p|i_T^X, j_T^X)$

La réputation de l'agent  $T$  selon  $X$  est donc donnée par

$$E(\phi_T^X) = \frac{i_T^X + 1}{i_T^X + j_T^X + 2}$$

### 2.4.3 Combiner les feedbacks

Si on a  $\phi_T^X$  et  $\phi_T^Y$ , deux fonctions de réputation de  $T$  résultant respectivement du feedback de  $X$  et du feedback de  $Y$ , on définit la fonction de réputation combinée  $\phi_T^{X,Y}$  comme suit :

$$\begin{aligned} - i_T^{X,Y} &= i_T^X + i_T^Y; \\ - j_T^{X,Y} &= j_T^X + j_T^Y. \end{aligned}$$

#### 2.4.4 Le système BRS dans notre travail

Pour notre recherche, nous avons utilisé un système de réputation de « type » BRS. En effet, nous allons reprendre le principe de la fonction beta en utilisant l'espérance mathématique de celle-ci comme valeur de réputation pour un agent. Pour plus d'informations sur le fonctionnement du système BRS et sur ses performances, on peut lire l'article [eRI02] le présentant.

### 2.5 Capacités d'une approche efficace selon Zhang Jie

Zhang Jie détermine dans sa thèse [Jie09] plusieurs conditions qu'un système de réputation doit respecter pour être efficace :

1. Majorité : une approche efficace doit gérer le fait qu'une majorité de notations puisse être injuste.
2. Flood : une approche efficace doit gérer le fait qu'un agent fournisse une énorme quantité de notations en peu de temps.
3. Manque d'expérience : une approche efficace doit gérer le fait qu'un agent ait peu voire aucune expérience.
4. Variation de comportement : une approche efficace doit gérer le fait qu'un agent puisse changer de comportement.

Une approche efficace doit donc être capable de repérer ou d'empêcher les notations injustes et de gérer le flood. En effet, une approche ne respectant pas ces deux points n'est pas optimale, car plusieurs agents pourraient s'allier pour fournir une série de notations négatives et injustes sur un agent concurrent. Comme expliqué dans l'introduction, c'est en partie là que se trouve le cœur de notre problème.

### 2.6 Mécanismes d'incitation et système de réputation

Le but d'un mécanisme d'incitation est de faire en sorte que des agents détenteurs d'une information la transmettent sans mentir. Le paiement est un bon moyen d'inciter ces agents à transmettre la vérité. Nous pouvons donc considérer les mécanismes où intervient un paiement ayant pour but de promouvoir l'honnêteté comme un système d'incitation même s'il ne correspond pas aux définitions formelles que nous en donnons.

Ainsi, réputation et incitation se mêlent souvent dans le sens où un système de réputation efficace est déjà en quelque sorte un mécanisme d'incitation très simple. En effet, l'augmentation de sa valeur de réputation peut être une incitation pour un agent à dire la vérité. L'augmentation de la valeur de réputation peut être comparée à un paiement. En effet, un agent a toujours intérêt à avoir la meilleure réputation possible, afin que les autres puissent lui accorder leur confiance. On appelle encore cela « mécanisme de crédibilité ». Nous en parlons dans la section suivante.

## 2.7 Deux grands types de mécanismes d'incitation pour les plateformes de commerce électronique [Jie09]

Dans le domaine du commerce électronique, deux grands types de mécanismes d'incitation existent. Le premier englobe les mécanismes de paiements secondaires. Nous en présenterons deux comme exemple. Le deuxième type englobe les mécanismes dit de crédibilité.

### 2.7.1 Les mécanismes de paiement secondaire

#### Goodwill Hunting (GWH)

Le mécanisme incite des agents acheteurs à fournir un feedback honnête, en offrant des rabais sur les frais périodiques d'adhésion au système si le feedback est consistant. Si un acheteur ne fournit aucun feedback, il ne reçoit pas de rabais. Le rabais se présente comme le remboursement d'une partie des frais d'une période à la fin de celle-ci. Si l'acheteur quitte le système avant la fin de la période, il ne reçoit donc pas le rabais. Cela empêche un acheteur à agir malhonnêtement avant de quitter le système.

#### R-agents

Les R-agents sont des agents qui achètent et vendent des notations aux agents ordinaires. Un agent ordinaire achète d'abord une notation à un R-agent. Après la transaction, celui-ci vend alors sa propre notation au R-agent. Un agent ordinaire ne peut vendre une notation sur un agent vendeur que s'il a d'abord acheté une notation sur ce vendeur. L'agent est payé pour sa notation uniquement si la prochaine notation reçue pour le même vendeur est identique. Il s'agit d'un dilemme du prisonnier répété (voir la sous-section 3.2.1), où la somme des utilités des agents est maximisée lorsque les 2 coopèrent. En rapportant une notation honnête, un agent a une probabilité d'au moins 0,5 d'être payé. Le montant que celui-ci reçoit pour un rapport jugé honnête est déterminé en fonction de la probabilité qu'un agent fasse confiance à un autre, et que cet autre agent ne change pas son comportement. Ce mécanisme est sensible aux « collusions ».

### 2.7.2 Les mécanismes de crédibilité

Au lieu de payer les agents qui fournissent des notations fiables, les mécanismes de crédibilité mesurent la crédibilité des agents par rapport à leurs notations passées. Un agent préférera avoir des transactions avec un agent crédible. Un exemple de mécanisme de crédibilité est CONFESS proposé par Jurca et Faltings [jeBF04]. Un simple mécanisme de réputation constitue déjà un mécanisme de crédibilité et donc un mécanisme d'incitation.



## Chapitre 3

# Éléments de la théorie des jeux

Établie en 1944 avec la publication du livre *Theory of games and economic behavior* de John von Neumann et Oskar Morgenstern, la théorie des jeux constitue un ensemble d'outils ayant pour but d'analyser des situations où des agents, preneurs de décisions, interagissent. Le jeu tel qu'on l'entend dans le sens commun, où plusieurs joueurs sont en compétition pour remporter la victoire, est un exemple d'une telle situation. Cependant le terme « jeu » dans l'expression « théorie des jeux » possède un sens bien plus large. En effet on peut appeler « jeu » toute situation dans laquelle ce qu'il est optimal de faire pour un agent (personne physique, entreprise, animal, etc.) dépend des anticipations qu'il forme sur les actions du ou des autres agents.

La théorie des jeux peut être utilisée dès qu'on se trouve dans une situation qui peut être modélisée par des agents qui veulent maximiser leur profit. Elle est particulièrement connue en économie, où on l'utilise pour analyser la concurrence entre des entreprises en situation d'oligopole. On l'utilise aussi dans les sciences politiques pour analyser la compétition électorale, ou encore en biologie, pour mieux comprendre l'évolution du comportement des espèces face à la modification de leur environnement. Depuis la fin des années 80, l'informatique s'intéresse aussi à la théorie des jeux.

Dans ce chapitre consacré à un tour de vue de la théorie des jeux, nous commencerons par définir exactement celle-ci, les éléments qui la constituent, et les hypothèses qu'elle suscite. Nous commencerons par voir le jeu de base : le jeu stratégique avec information parfaite. Nous verrons ensuite les jeux étendus avec information parfaite, et enfin, nous finirons avec les jeux répétés.

Ce chapitre est principalement basé sur le livre de Martin J. Osborne, *An Introduction to Game Theory* [Os09]. Nous nous basons sur ce livre car la manière dont les auteurs présentaient les choses nous semblait être la plus claire et la plus efficace.

### 3.1 Théorie des jeux et choix rationnel

La théorie des jeux peut être vue comme une collection de modèles. Dans les sciences, un modèle est une abstraction utilisée pour comprendre les observations et les expériences,

celui-ci, par sa simplicité, nous aide à comprendre un phénomène selon un certain point de vue. La théorie des jeux nous donne une méthode de modélisation, conçue pour représenter des situations où des agents interagissent. La modélisation de telles situations par la théorie des jeux nous en livre une certaine compréhension, qui s'avère pertinente dans les domaines politiques, économiques, et sociaux. La modélisation commence à partir d'une idée liée à des aspects de l'interaction de plusieurs agents preneurs de décisions. Le but est d'inclure cette idée dans la modélisation. L'analyse par la théorie des jeux pourra dès lors confirmer cette idée, ou la réfuter.

La théorie du choix rationnel est une hypothèse posée dans la plupart des modèles en théorie des jeux. Cette théorie stipule qu'un agent choisira toujours la meilleure *action* en accord avec ses *préférences*, parmi toutes les *actions* disponibles. La théorie se base donc sur deux éléments : un ensemble d'actions disponibles pour l'agent, et la spécification des préférences de l'agent rationnel.

### 3.1.1 Les actions

On considère un ensemble  $A$  contenant toutes les actions qui, sous certaines circonstances, sont disponibles pour l'agent. Dans une situation donnée, l'agent est face à un sous-ensemble de  $A$ , dans lequel il doit choisir un seul élément (une seule action). L'agent connaît le sous-ensemble d'actions disponibles. Celui-ci est donné et ne dépend pas des préférences de l'agent. Le choix de l'action parmi le sous-ensemble sera, lui, guidé par les préférences de l'agent.

### 3.1.2 Préférences et fonction d'utilité

Pour chaque paire d'actions possible, un agent doit savoir laquelle il préfère, ou s'il est indifférent. Les préférences d'un agent doivent être consistantes, c'est-à-dire que la propriété de transitivité doit être respectée : si un agent préfère l'action  $a$  à l'action  $b$ , et préfère l'action  $b$  à l'action  $c$ , alors l'agent doit préférer l'action  $a$  à l'action  $c$ . Aucune autre condition n'est imposée sur les préférences. Par exemple, bien que chaque agent soit rationnel, rien ne nous empêche de modéliser une situation comportant un agent altruiste, c'est-à-dire préférant choisir l'action maximisant le bien-être d'autres agents.

Afin de décrire les préférences d'un agent, on utilise une fonction d'utilité (en anglais, *payoff function*), qui associe un nombre à chaque action, de manière telle qu'une action sera préférée à une autre si le nombre qui lui est associé par la fonction d'utilité est plus élevé. Les préférences d'un agent consistent en une information ordinale, c'est-à-dire que celles-ci expriment si une action est préférée à une autre, et non « de combien » elle est préférée. Pour illustrer cela, considérons trois actions  $a$ ,  $b$ , et  $c$ , disponibles pour un agent, tel que  $a > b > c$ , où la relation  $>$  est la relation de préférence. De multiples fonctions d'utilité peuvent être utilisées pour exprimer ces préférences. Par exemple :

- $u(a) = 0, u(b) = 1, u(c) = 2$  ;
- $u(a) = 0, u(b) = 1, u(c) = 100$  ;

## 3.2 Jeux stratégiques avec information parfaite

---

- $u(a) = 0, u(b) = 100, u(c) = 200$ .

Ces 3 fonctions d'utilité expriment exactement la même chose. Il ne faut en aucun cas voir dans la deuxième fonction d'utilité une « intensité » de préférence supérieure pour l'action  $c$  que dans la première fonction d'utilité.

### 3.1.3 Interactions entre agents

Jusqu'ici, nous avons parlé d'un agent ayant un ensemble d'actions et des préférences sur ces actions, l'agent ne s'inquiétant que de l'action qu'il choisit. Cependant, un agent n'est pas affecté que par ses propres actions, il l'est aussi par les actions des autres agents. Les jeux modélisent de tels types de situation. Par exemple, les agents pourraient être des entreprises concurrentes. Pour un produit, l'entreprise le fabricant pourra contrôler son prix, cependant, celle-ci doit prendre en considération le prix de son produit, mais aussi le prix des produits concurrents. Comment l'entreprise devrait dès lors fixer un prix dans cette situation ? Les jeux stratégiques nous aident à analyser la situation et à éventuellement trouver la réponse.

## 3.2 Jeux stratégiques avec information parfaite

Un jeu stratégique est un type de modélisation d'interaction d'agents. Le jeu stratégique est en quelque sorte le jeu de base. Formellement, un jeu stratégique est composé :

- d'un ensemble de joueurs (les agents sont appelés joueurs dans le contexte de la théorie des jeux) ;
- pour chaque joueur, d'un ensemble d'actions ;
- pour chaque joueur, des préférences sur l'ensemble des *profils d'actions*.

Un profil d'actions est un ensemble d'actions contenant une action pour chaque joueur. Le résultat d'un jeu est un profil d'actions. Ici, nous parlons de jeu stratégique avec information parfaite, c'est-à-dire, où chaque joueur est au courant de la spécification complète du jeu.

Nous modélisons un jeu stratégique par un tableau. Nous numérotions les cases comme dans le tableau 3.1. Bien entendu quand nous utiliserons ce tableau, les cases contiendront des utilités.

		Joueur 2	
		Action 1	Action 2
Joueur 1	Action 1	1	2
	Action 2	3	4

TABLE 3.1 – Modélisation du dilemme du prisonnier

### 3.2.1 Un exemple bien connu : le dilemme du prisonnier

Les deux suspects d'un crime sont interrogés séparément. Ils peuvent soit se taire, soit dénoncer l'autre. Si les deux se taisent, on ne peut les condamner qu'à un an de prison. Si l'un des deux dénonce l'autre, il sera libéré et utilisé comme témoin contre l'autre qui écoperà d'une peine de 5 ans de prison. Si les deux se dénoncent mutuellement, ils auront chacun une peine de 3 ans de prison. On modélise la situation comme dans le tableau 3.2.

		Suspect 2	
		Se taire	Dénoncer
Suspect 1	Se taire	2 , 2	0 , 3
	Dénoncer	3 , 0	1 , 1

TABLE 3.2 – Modélisation du dilemme du prisonnier

Chaque case du tableau 2.1 correspond à un profil d'actions. Dans chaque case se trouvent les utilités des joueurs dans l'ordre. Le premier nombre de la case correspond à l'utilité du suspect 1, et le deuxième nombre à l'utilité du suspect 2.

Pour le suspect 1, nous considérons les préférences suivantes : (Dénoncer, Se taire) > (Se taire, Se taire) > (Dénoncer, Dénoncer) > (Se taire, Dénoncer). (X, Y) est un profil d'actions où X est l'action choisie par le joueur 1 et Y par le joueur 2. Dans cette situation, le suspect 1 préfère le profil d'actions (Dénoncer, Se taire), car s'il dénonce et que l'autre se tait, il sera libéré. En deuxième lieu, il préfère le profil (Se taire, Se taire), car il n'aura alors qu'un an de prison. Le troisième profil préféré est (Dénoncer, Dénoncer), il aura alors 3 ans de prison. Enfin, le profil le moins préféré est (Se taire, Dénoncer), car s'il se tait et que l'autre dénonce, il aura une peine de 5 ans de prison. Nous créons donc une fonction d'utilité correspondant à ces préférences. Soit une fonction  $u_1$  tel que :

- $u_1(\text{Dénoncer}, \text{Se taire}) = 3$  ;
- $u_1(\text{Se taire}, \text{Se taire}) = 2$  ;
- $u_1(\text{Dénoncer}, \text{Dénoncer}) = 1$  ;
- $u_1(\text{Se taire}, \text{Dénoncer}) = 0$ .

Plus le nombre associé à un profil d'actions par la fonction d'utilité est grand, plus le profil est préféré.

Pour le suspect 2, les préférences sont : (Se taire, Dénoncer) > (Se taire, Se taire) > (Dénoncer, Dénoncer) > (Dénoncer, Se taire). La fonction  $u_2$  modélise ces préférences :

- $u_2(\text{Se taire}, \text{Dénoncer}) = 3$  ;
- $u_2(\text{Se taire}, \text{Se taire}) = 2$  ;
- $u_2(\text{Dénoncer}, \text{Dénoncer}) = 1$  ;
- $u_2(\text{Dénoncer}, \text{Se taire}) = 0$ .

## 3.2 Jeux stratégiques avec information parfaite

---

Pour nous, observateur externe, le meilleur profil d'actions est (Se taire, Se taire) car chaque suspect écoperait alors d'une seule année de prison. Cependant, les deux suspects se dénonceront. Regardons le jeu du point de vue du premier suspect. Sa peine va dépendre de l'action qu'il prend mais aussi de l'action de l'autre. Si le suspect 2 se tait, en se taisant aussi, le suspect 1 prendra 1 an de prison, mais en le dénonçant, il sera libéré ! Donc si le suspect 2 se tait, la meilleure chose à faire pour le suspect 1 est de dénoncer. Si le suspect 2 dénonce, la meilleure chose à faire pour le suspect 1 est de dénoncer aussi, il prendra alors 3 ans au lieu de 5. De cette analyse, il découle que le suspect 1 dénoncera toujours. La même réflexion amène à la même conclusion pour le joueur 2. Les deux suspects se dénonceront mutuellement et prendront chacun 3 ans de prison alors que s'ils s'étaient tous les deux tus, ils auraient écopé d'une seule année de prison chacun. Dans cette situation, les prisonniers ne pouvant pas communiquer entre eux (ils sont interrogés séparément), ils ne peuvent « s'allier » et décider de se taire tous les deux.

### 3.2.2 L'équilibre de Nash

Le profil d'action  $a^*$  dans un jeu stratégique est un équilibre de Nash si, pour chaque joueur  $i$  et chaque action  $a_i$  du joueur  $i$ ,  $a^*$  est au moins aussi bon, selon les préférences du joueur  $i$ , à tout autre profil  $(a_i, a_{-i}^*)$  dans lequel le joueur  $i$  choisit  $a_i$ , et tout autre joueur  $j$  choisit  $a_j^*$ . Autrement dit, un profil d'actions est un équilibre de Nash si aucun joueur ne peut augmenter son utilité en changeant d'action *étant donné* les actions des autres joueurs.

DÉFINITION 3.1. Un équilibre de Nash est un profil d'actions  $a^*$  tel que :

$$u_i(a^*) \geq u_i(a_i, a_{-i}^*) \forall a_i \in A_i$$

où  $u_i$  est la fonction d'utilité du joueur  $i$ ,  $A_i$  son ensemble d'actions, et  $a_{-i}^*$  représente l'ensemble des actions choisies par tous les joueurs sauf  $i$ . Un équilibre de Nash est dit strict si l'inégalité dans la définition est stricte.

Dans un jeu, les joueurs tendront toujours à choisir une action appartenant à un équilibre de Nash. Dans le dilemme du prisonnier, (Dénonce, Dénonce) est un équilibre de Nash. En effet, étant donné l'action de l'autre suspect, aucun suspect ne peut choisir une meilleure action. (Se taire, Se taire) n'est pas un équilibre de Nash : étant donné l'action choisie par le suspect 2 (Se taire), le suspect 1 peut améliorer son utilité en avouant. Idem pour le second suspect. Dans le dilemme du prisonnier, il y a un équilibre de Nash unique, il sera donc toujours atteint.

### 3.2.3 Football ou opéra

Une autre situation modélisable par la théorie des jeux est la suivante : un couple désire sortir. L'homme veut aller voir un match de foot et la femme désire aller à l'opéra. On modélise le jeu comme suit :

		Femme	
		Football	Opéra
Homme	Football	2 , 1	0 , 0
	Opéra	0 , 0	1 , 2

TABLE 3.3 – Modélisation de la situation de choix entre deux activités pour un couple

Premièrement, chacun désire sortir avec l'autre. Ils ont donc chacun une utilité de 0 s'ils choisissent une action différente. Si les deux choisissent le football, la femme a une utilité de 1 et l'homme de 2. Si les deux choisissent l'opéra, c'est le contraire. Il y a deux équilibres de Nash dans ce jeu : (Opéra, Opéra) et (Football, Football). En effet, dans chacune de ces situations, aucun n'a intérêt à changer de choix (étant donné le choix de l'autre) car il se retrouverait seul. (Opéra, Football) et (Football, Opéra) ne sont pas des équilibres de Nash, chaque joueur pouvant augmenter son utilité en changeant son choix (étant donné celui de l'autre).

### 3.2.4 Pile ou face

Deux joueurs choisissent simultanément de montrer à l'autre la face d'une pièce. Si les deux joueurs choisissent la même face, le premier donne 1 euro au second. S'ils ont choisi des faces différentes, le second donne 1 euro au premier. Ce jeu est modélisé comme suit :

		joueur 2	
		Pile	Face
Joueur 1	Pile	-1 , 1	1 , -1
	Face	1 , -1	-1 , 1

TABLE 3.4 – Modélisation du jeu pile ou face

Ce jeu est dit strictement compétitif, car à chaque fois qu'un joueur gagne 1 euro, l'autre perd 1 euro. Il n'y a pas d'équilibre de Nash dans ce jeu. A chaque profil d'actions possibles, il y a toujours un joueur qui perd 1 euro et qui l'aurait gagné en choisissant l'autre face (étant donné la face choisie par l'autre joueur).

### 3.2.5 Fonction de meilleures réponses

Pour un joueur  $i$ , étant donné les actions des autres joueurs, il existe un ensemble d'actions pour lui-même, qui sont les « meilleures » par rapport à l'ensemble des actions choisies par les autres joueurs. En reprenant l'exemple ci-dessus (table 3.4), la meilleure action pour l'homme est de choisir Opéra si la femme a choisi Opéra. La meilleure action est Football si la femme a choisi Football. Dans cet exemple, l'ensemble des meilleures réponses de l'homme ne contient qu'une seule action, pour chaque action choisie par la femme. Ce raisonnement que nous avons fait pour l'homme fonctionne aussi pour la femme.

DÉFINITION 3.2. On définit la fonction de meilleures réponses (en anglais : *best response function*) comme suit :

$$B_i(a_{-i}) = \{a_i \in A_i : u_i(a_i, a_{-i}) \geq u_i(a'_i, a_{-i}) \forall a'_i \in A_i\}$$

### 3.3 Jeux étendus avec information parfaite

---

Chaque action dans  $B_i(a_{-i})$  est au moins aussi bonne pour le joueur  $i$  que tout autre action de celui-ci, lorsque les actions choisies par les autres joueurs sont données par  $a_{-i}$ . La fonction de meilleures réponses associe un ensemble d'actions pour un joueur à chaque liste d'actions choisies par les autres joueurs.

En utilisant la fonction de meilleures réponses, on peut trouver le(s) équilibre(s) de Nash s'il(s) existe(nt) : un profil d'actions  $a^*$  est un équilibre de Nash d'un jeu stratégique si et seulement si l'action prise par chaque joueur est une meilleure réponse par rapport à l'ensemble des actions choisies par les autres joueurs :

$$a_i^* \in B_i(a_{-i}^*) \forall i$$

A partir de là, pour trouver les équilibres de Nash éventuels d'un jeu stratégique, il suffit de trouver les fonctions de meilleures réponses de chaque joueur, puis de trouver les différents profils d'actions satisfaisant l'équation ci-dessus.

#### 3.2.6 Actions dominantes

Dans un jeu stratégique, une action  $a_i''$  du joueur  $i$  domine strictement une action  $a_i'$  si :

$$u_i(a_i'', a_{-i}) > u_i(a_i', a_{-i}) \forall a_{-i}$$

La domination est dite non stricte ou faible si l'inégalité est non stricte pour certaines valeurs de  $a_{-i}$ . L'inégalité doit cependant rester stricte pour au moins une valeur de  $a_{-i}$  pour parler de domination.

### 3.3 Jeux étendus avec information parfaite

Dans un jeu stratégique, tous les joueurs prennent leur décision (c'est-à-dire choisissent une action parmi celles disponibles) en même temps. Les jeux étendus peuvent rendre compte d'une structure séquentielle dans les prises de décisions. Dans un jeu étendu, les joueurs agissent l'un après l'autre, et sont donc au courant des choix précédents des autres joueurs. Chaque joueur connaît les actions qui ont été précédemment jouées, le jeu est donc dit « à information parfaite ». Pour décrire un tel jeu, il faut spécifier les joueurs et leurs préférences, tout comme dans un jeu stratégique. Il faut aussi spécifier l'ordre dans lequel les joueurs jouent, et les actions que chaque joueur peut prendre à chaque point.

#### 3.3.1 Définition d'un jeu étendu avec information parfaite

Un jeu étendu avec information parfaite consiste en :

- un ensemble de joueurs ;
- un ensemble de séquences d'actions appelées *histoires terminales* tel qu'aucune séquence n'est une sous-séquence stricte d'une autre ;
- une fonction appelée la *fonction joueur* qui assigne un joueur à chaque séquence qui est une sous-séquence stricte d'une histoire terminale ;
- pour chaque joueur, des préférences sur l'ensemble des histoires terminales.

### Histoire terminale

DÉFINITION 3.3. On appelle *histoire* d'un jeu la séquence d'actions jouée depuis le début du jeu jusqu'au moment présent. Si le jeu est terminé, on parle d'*histoire terminale*.

Une histoire terminale est donc une séquence d'actions qui peuvent être jouées du début à la fin. Un jeu de plateau, comme les échecs par exemple, pourrait être modélisé par un jeu étendu à information parfaite. Aux échecs, les blancs (joueur 1) commencent. Le joueur 1 choisi un coup parmi l'ensemble des coups disponibles, puis c'est au tour du joueur 2 (les noirs) de jouer, et ainsi de suite. A la fin d'une partie, la séquence des coups joués constitue une histoire terminale. Toute sous-séquence de coups est une sous-histoire. L'ensemble des histoires terminales est l'ensemble de toutes les séquences de coups possibles menant à une situation d'échec et mat. Les joueurs ont des préférences sur les histoires terminales. On pourrait par exemple définir les préférences d'un joueur comme étant de 1 pour toutes les histoires terminales où il termine vainqueur (l'adversaire est mis échec et mat), et de 0 pour toutes les histoires terminales où il est perdant (il est lui-même mis échec et mat). Aux échecs, une sous-histoire serait donc une séquence de coup ne mettant aucun des joueurs échec et mat. La partie continue donc après une telle séquence de coups, et la fonction joueur détermine lequel doit jouer après une certaine séquence de coups (qui est une sous-histoire). Aux échecs, les joueurs jouent l'un après l'autre, la fonction joueur déterminera donc que c'est aux blancs de jouer si le dernier coup d'une sous-histoire est un coup du joueur noir, et vice-versa. Nous n'avons pas pris en compte la possibilité de match nul ou d'abandon dans une partie d'échecs pour simplifier l'exemple.

### Propriétés d'un jeu étendu

DÉFINITION 3.4. Un jeu étendu est dit à horizon fini si toutes ses histoires terminales sont finies.

DÉFINITION 3.5. Un jeu étendu est dit fini s'il est à horizon fini et s'il possède un nombre fini d'histoires terminales.

### Représentation d'un jeu étendu

Si un jeu stratégique peut être représenté sous forme de tableau, un jeu étendu sera représenté sous forme d'arbre.

EXEMPLE 3.1. L'arbre de la figure 3.1 ci-dessous représente un jeu étendu. Le joueur 1 peut choisir l'action A ou l'action B. S'il choisit l'action B, le jeu est terminé ; le joueur 1 gagne alors une utilité de 1 et le joueur 2 gagne une utilité de 2. Si le joueur 1 choisit l'action A, le jeu continue et c'est au tour du joueur 2, qui doit choisir entre l'action C et l'action D. S'il choisit l'action C, le joueur 1 gagne une utilité de 2 et le joueur 2 une utilité de 1. S'il choisit l'action D, le joueur 1 gagne une utilité de 0, et le joueur 2 gagne une utilité de 1.

### 3.3 Jeux étendus avec information parfaite

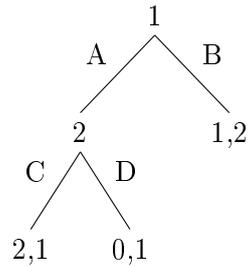


FIGURE 3.1 – Un exemple de jeu étendu à 2 joueurs

#### 3.3.2 Stratégies

Dans un jeu étendu, chaque joueur suit une stratégie.

DÉFINITION 3.6. Une stratégie pour un joueur  $i$  dans un jeu étendu avec information parfaite est une fonction qui assigne à chaque histoire  $h$  après laquelle  $i$  doit jouer, c'est-à-dire  $P(h) = i$ , où  $P$  est la fonction joueur, une action  $a$  tel que  $a \in A(h)$ , où  $A(h)$  représente l'ensemble des actions possibles pour le joueur  $i$  après l'histoire  $h$ .

EXEMPLE 3.2. L'exemple ci-dessous montre un jeu étendu. La table 2.4 montre les 4 stratégies possibles pour le joueur 2 :

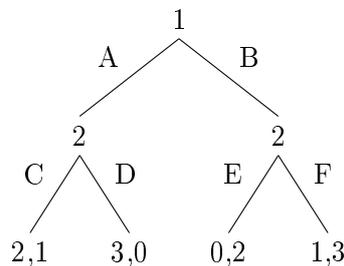


FIGURE 3.2 – Un jeu étendu à 2 joueurs. Le joueur 2 a le choix entre 4 stratégies possibles. Le joueur 1 a le choix entre 2 stratégies possibles.

	Action à prendre après l'histoire [A]	Action à prendre après l'histoire [B]
Stratégie 1	C	E
Stratégie 2	C	F
Stratégie 3	D	E
Stratégie 4	D	F

TABLE 3.5 – Stratégies possibles pour le joueur 2

Le joueur 2 joue soit après l'histoire consistant en la seule action A (notée [A]), ou l'histoire consistant en la seule action B (notée [B]). Après l'histoire [A], le joueur 2 peut choisir entre l'action C et l'action D. Après l'histoire [B], il peut choisir entre l'action E et l'action F. Une stratégie doit déterminer l'action que choisira le joueur 2 pour chaque histoire. Il y a donc 4 stratégies possibles. Ce sont celles montrées dans la table 2.4.

Le joueur 1, lui, joue après l'histoire vide (notée []). Il a le choix entre l'action A et l'action B. Il y a donc 2 stratégies possibles pour le joueur 1 : celle disant de prendre l'action A et celle disant de prendre l'action B.

Une stratégie peut être vue comme un plan d'action. Elle détermine quelle action un joueur doit entreprendre en fonction de la séquence d'actions déjà jouée (l'histoire du jeu). On considère que chaque joueur choisit une stratégie au début du jeu. Dans un jeu stratégique, le résultat (le montant d'utilité gagné par chaque joueur) était déterminé par le profil d'actions, c'est-à-dire l'ensemble des actions choisies par les joueurs. Dans un jeu étendu, le résultat est déterminé par le profil de stratégies, c'est-à-dire l'ensemble des stratégies choisies par les joueurs. En effet, un profil de stratégie détermine l'histoire terminale qui aura lieu, et donc l'utilité gagnée par chaque joueur.

DÉFINITION 3.7. Soit  $s^*$  un profil de stratégies, on note  $O(s^*)$ , l'histoire terminale découlant de  $s^*$ .

### 3.3.3 L'équilibre de Nash dans un jeu étendu

Nous allons maintenant définir l'équilibre de Nash dans le cas d'un jeu étendu.

DÉFINITION 3.8. Le profil de stratégie  $s^*$  dans un jeu étendu (avec information parfaite) est un équilibre de Nash si, pour chaque joueur  $i$ , et chaque stratégie  $s_i$  de  $i$ , l'histoire terminale  $O(s^*)$  est au moins aussi bonne (en accord avec les préférences de  $i$ ) à toute histoire terminale  $O(s_i, s_{-i}^*)$ . Écrit de manière plus formelle :

$$u_i(O(s^*)) \geq u_i(O(s_i, s_{-i}^*)) \forall s_i \in S(i)$$

où  $u_i$  est la fonction d'utilité du joueur  $i$  et  $S(i)$  est l'ensemble des stratégies du joueur  $i$ .

Dans l'exemple 2.2 ci dessus, l'équilibre de Nash consiste à choisir l'action A pour le joueur 1 et la stratégie 2 pour le joueur 2. En effet, peu importe l'action choisie par le joueur 2 après, le joueur 1 aura toujours une meilleure utilité en choisissant l'action A. En choisissant A, il gagnera 2 ou 3 (selon l'action choisie par le joueur 2), alors qu'en choisissant B, il gagnera 0 ou 1. De même, le joueur 2 ne changera pas de stratégie, parce que la stratégie 2 lui assure une utilité maximale, selon l'action choisie par le joueur 1.

### 3.3.4 Équilibre parfait de sous-jeu

La notion d'équilibre de Nash ignore la structure séquentielle du jeu étendu. Lorsqu'on utilise les jeux étendus, on parlera d'équilibre parfait de sous-jeu, qui est un concept plus adapté pour identifier les états stables de tels jeux. Pour comprendre la notion d'équilibre parfait de sous-jeu, il faut d'abord définir ce qu'est un sous-jeu.

DÉFINITION 3.9. Soit  $\Gamma$  un jeu étendu avec information parfaite. Soit  $P$  la fonction joueur. Pour toute histoire non terminale  $h$  de  $\Gamma$ , le sous-jeu  $\Gamma(h)$  est défini comme suit :

- il démarre avec l'histoire  $h$  (un jeu complet démarre avec une histoire vide) ;
- les joueurs de  $\Gamma(h)$  sont les joueurs de  $\Gamma$  ;

### 3.3 Jeux étendus avec information parfaite

---

- l'ensemble des histoires terminales de  $\Gamma(h)$  est l'ensemble des séquences d'actions  $h'$  tel que  $(h, h')$  est une histoire terminale de  $\Gamma$  ;
- le joueur  $P(h, h')$  est assigné à chaque sous-histoire stricte  $h'$  d'une histoire terminale ;
- un joueur préfère  $h'$  à  $h''$  dans le sous-jeu  $\Gamma(h)$  si et seulement si il préfère  $(h, h')$  à  $(h, h'')$  dans le jeu  $\Gamma$ .

EXEMPLE 3.3. Le jeu de l'exemple 2.2 possède 2 sous-jeux stricts (3.3 et 3.4) :

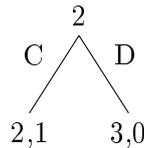


FIGURE 3.3 – Un des 2 sous-jeux stricts de l'exemple 2.2

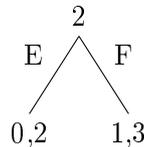


FIGURE 3.4 – L'autre sous-jeu strict de l'exemple 2.2

De manière intuitive, un équilibre parfait de sous-jeu est un profil de stratégie qui est un équilibre de Nash dans le jeu complet et dans chacun de ses sous-jeux stricts.

DÉFINITION 3.10. Le profil de stratégies  $s^*$  dans un jeu étendu avec information parfaite est un équilibre parfait de sous-jeu si, pour chaque joueur  $i$ , chaque histoire  $h$  après laquelle c'est à  $i$  de jouer, et chaque stratégie  $s_i$  du joueur  $i$ , l'histoire terminale  $O_h(s^*)$  générée par  $s^*$  après l'histoire  $h$  est au moins aussi bonne (selon les préférences de  $i$ ) que l'histoire terminale  $O_h(s_i, s_{-i}^*)$  générée par le profil de stratégies  $(s_i, s_{-i}^*)$ . Écrit de manière formelle :

$$(\forall i, \forall h : P(h) = i, \forall s_i) : u_i(O_h(s^*)) \geq u_i(O_h(s_i, s_{-i}^*))$$

#### Interprétation de l'équilibre parfait de sous-jeu

L'équilibre parfait de sous-jeu est un état stable. La différence d'interprétation entre un équilibre de Nash et un équilibre parfait de sous-jeu est que ce dernier entraîne l'interprétation d'une stratégie comme un plan spécifiant les actions d'un joueur, non seulement après les histoires consistantes avec la stratégie, mais aussi après les histoires résultant d'un choix arbitraire d'actions, dû à des erreurs de la part du joueur, ou à des expérimentations. L'équilibre parfait de sous-jeu est donc un concept plus robuste que le simple équilibre de Nash.

#### Trouver un équilibre parfait de sous-jeu

PROPRIÉTÉ 3.1. Tout jeu étendu avec information parfaite et fini a un équilibre parfait de sous-jeu

DÉFINITION 3.11. La longueur d'un sous-jeu est égale à la longueur de la plus longue histoire du sous-jeu.

Il est possible de trouver le(s) équilibre(s) parfait(s) de sous-jeu avec la procédure dite « backward induction ». Elle fonctionne comme suit :

1. on commence par déterminer les actions optimales des joueurs jouant dans les sous-jeux de longueur 1 ;
2. prenant les actions déterminées par l'étape 1 de la procédure comme données, on détermine les actions optimales pour le joueur jouant en premier dans tous les sous-jeux de longueur 2 ;
3. on répète la procédure jusqu'à parvenir au jeu complet. On obtient alors le(s) équilibre(s) parfait(s) de sous-jeu.

EXEMPLE 3.4. Reprenons notre exemple type et appliquons lui la procédure de backward induction :

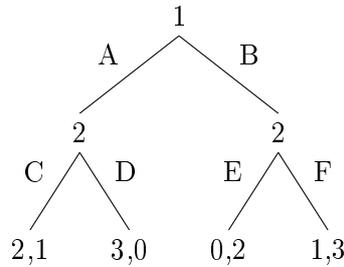


FIGURE 3.5 – Un jeu étendu à 2 joueurs. Le joueur 2 a le choix entre 4 stratégies possibles. Le joueur 1 a le choix entre 2 stratégies possibles.

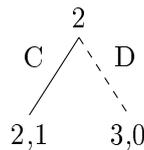


FIGURE 3.6 – Premier sous-jeu de longueur 1. L'action optimale pour le joueur 2 (qui joue dans ce sous-jeu) est l'action C pour laquelle il gagne 1 (au lieu de 0 pour l'action D).

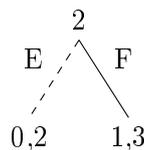


FIGURE 3.7 – Second sous-jeu de longueur 1. L'action optimale pour le joueur 2 (qui joue dans ce sous-jeu) est l'action F pour laquelle il gagne 3 (au lieu de 2 pour l'action E).

La stratégie optimale pour le joueur 1 correspond à choisir l'action A. Nous la notons [A]. La stratégie optimale pour le joueur 2 est de choisir C si le joueur 1 choisit A et F si le joueur 1 choisit B, nous la notons [CF]. L'équilibre parfait de sous-jeu (qui ici, est unique) sera ([A], [CF]). Il correspond à l'état stable (2,1) encadré dans le schéma.

### 3.3 Jeux étendus avec information parfaite

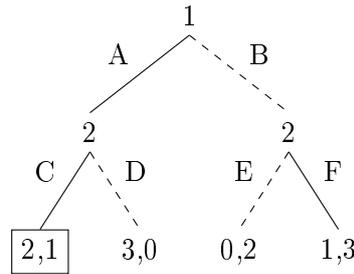


FIGURE 3.8 – Sous-jeu de longueur 2 correspondant au jeu complet. Étant donné les actions optimales fixées pour le joueur 2, l'action optimale pour le joueur 1 est A pour laquelle il gagnera 2 (au lieu de 1 s'il choisissait B).

#### 3.3.5 Mouvements simultanés

Il est possible d'autoriser les mouvements simultanés dans un jeu étendu. Dans ce cas, un tour du jeu étendu est un jeu stratégique. La fonction joueur n'associe plus un joueur après chaque histoire, mais un ensemble de joueurs.

DÉFINITION 3.12. Un jeu étendu avec information parfaite et mouvements simultanés consiste en :

- un ensemble de joueurs ;
- un ensemble de séquences d'actions appelées *histoires terminales* tel qu'aucune séquence n'est une sous-séquence stricte d'une autre ;
- une fonction appelée la *fonction joueur* qui assigne un *ensemble de joueurs* à chaque séquence qui est une sous-séquence stricte d'une histoire terminale ;
- pour chaque sous-histoire stricte  $h$  de chaque histoire terminale, et chaque joueur  $i$  qui est un membre de l'ensemble des joueurs assignés à  $h$  par la fonction joueur, un ensemble  $A_i(h)$  qui est l'ensemble des actions disponibles aux joueurs  $i$  après l'histoire  $h$  ;
- pour chaque joueur, des préférences sur l'ensemble des histoires terminales.

DÉFINITION 3.13. Une stratégie d'un joueur  $i$  dans un jeu étendu avec information parfaite et mouvements simultanés est une fonction qui assigne à chaque histoire  $h$  après laquelle  $i$  est un des joueurs devant jouer ( $i \in P(h)$ ), une action  $a \in A_i(h)$

EXEMPLE 3.5. Comme exemple, on peut imaginer le jeu « Football ou Opéra » modifié comme suit : dans un couple, une des deux personnes (disons la femme) doit choisir entre rester à la maison ou sortir. Si elle choisit de sortir, la question d'aller voir le football ou l'opéra se pose. Le jeu est modélisé comme suit :

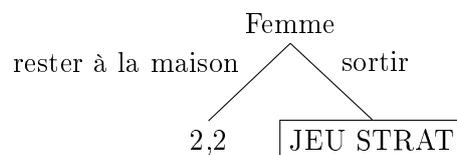


FIGURE 3.9 – Si la femme choisit de rester à la maison, l'état (2,2) est atteint, sinon, le jeu stratégique ci-dessous est joué.

		Femme	
		Football	Opéra
Homme	Football	3 , 1	0 , 0
	Opéra	0 , 0	1 , 3

TABLE 3.6 – Jeu stratégique joué si la femme décide de sortir.

**Quel est ou quels sont le(s) équilibre(s) parfait(s) de sous-jeu ?**

- Dans le sous-jeu suivant l’histoire (sortir), il y a deux équilibres de Nash : (Football, Football), et (Opéra, Opéra) ;
- Si le résultat du jeu stratégique est (Football, Football), alors le choix optimal pour la femme au début du jeu est de choisir de rester à la maison ;
- Si le résultat du jeu stratégique est (Opéra, Opéra), alors le choix optimal pour la femme au début du jeu est de choisir de sortir.

Le jeu a donc deux équilibres parfaits de sous jeu :

- ((Rester à la maison, Football), Football) ;
- ((Sortir, Opéra), Opéra).

**3.3.6 Mouvements aléatoires**

Il est aussi possible d’ajouter certains *mouvements aléatoires* à un jeu. Dans ce cas, après une certaine histoire, la fonction joueur peut assigner une probabilité entre plusieurs mouvements possibles plutôt que d’assigner un joueur.

EXEMPLE 3.6. Le jeu suivant est un exemple de jeu étendu avec mouvement aléatoire. Au début, le joueur 1 peut choisir entre l’action A et l’action B. S’il choisit A, le jeu est terminé et les joueurs gagnent l’utilité (1,1). Si B est choisi, alors le jeu se termine avec une probabilité de 0,5 et les joueurs gagnent l’utilité (3,0), où c’est au tour du joueur 2 de choisir une action (avec une probabilité de 0,5). Le jeu se termine alors après que le joueur 2 ait terminé son tour.

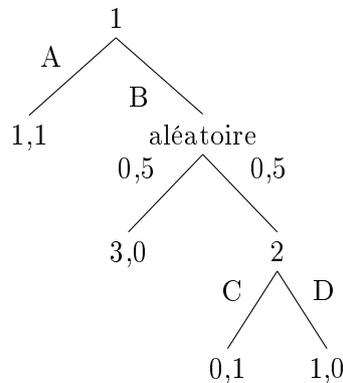


FIGURE 3.10 – Jeu étendu à 2 joueurs et avec un mouvement aléatoire si le premier joueur choisit l’action B.

## 3.4 Les jeux répétés

Lorsqu'un groupe de joueurs interagit de manière répétée, chaque membre peut conditionner son action par rapport aux actions des autres joueurs. Les jeux étendus nous permettent de modéliser cette possibilité. Un jeu répété est un jeu stratégique que l'on répète plusieurs fois. Formellement, on va considérer un jeu répété comme un jeu étendu avec mouvements simultanés où le même jeu stratégique est joué à chaque tour. L'idée principale des jeux répétés est qu'un joueur peut être dissuadé d'utiliser un avantage à court terme (c'est-à-dire de choisir l'action du jeu stratégique lui rapportant le plus) par des menaces de punition des autres joueurs qui réduiraient son utilité à long terme. Le fait qu'un jeu soit répété peut changer la donne, puisqu'un joueur va jouer en ayant connaissance des jeux précédemment joués, et en possédant donc une certaine information sur le comportement des autres joueurs. Des stratégies sont élaborées de manière à « répondre » à un joueur qui aurait choisi une certaine action au tour suivant. Voyons un exemple concret avec le dilemme du prisonnier répété.

### 3.4.1 Le dilemme du prisonnier répété

Reprenons le jeu stratégique du dilemme du prisonnier :

		Suspect 2	
		Se taire	Dénoncer
Suspect 1	Se taire	2 , 2	0 , 3
	Dénoncer	3 , 0	1 , 1

TABLE 3.7 – Modélisation du dilemme du prisonnier

Dans le dilemme du prisonnier simple, l'équilibre de Nash est (Dénoncer, Dénoncer), c'est-à-dire que lorsque le jeu est joué une seule fois (on parlera de *one-shot*), les joueurs ne connaissant pas le comportement que prendra l'autre, ils préfèrent se dénoncer, alors qu'ils pourraient obtenir tous deux une meilleure utilité en se taisant. Cependant, si on répète le dilemme, cela devient différent. En effet, on considère que les joueurs connaissent l'histoire du jeu répété, c'est-à-dire toutes les actions précédemment choisies par les deux joueurs. Les joueurs doivent alors adopter une certaine stratégie (au sens de la définition d'une stratégie pour les jeux étendus). Une des stratégies possibles est de toujours dénoncer l'autre. Cependant, si l'action de dénoncer l'autre est « optimale » dans le jeu stratégique, rien ne dit que dans le jeu répété, cette stratégie l'est. Une autre stratégie possible serait la suivante :

- tant que l'autre joueur se tait : se taire ;
- si l'autre joueur dénonce une fois : toujours dénoncer.

Dans cette stratégie, que nous appellerons *stratégie de coopération*, le joueur tente d'être coopératif et continue tant que l'autre l'est aussi. Mais si l'autre le « trahit » une seule fois, le joueur le dénoncera alors à chaque tour. Ceci peut être vu comme une « punition » infligée à l'autre joueur pour avoir tenté d'accroître son utilité au détriment de l'autre. Comment l'autre joueur devrait répondre à cette stratégie ? Dans le jeu stratégique (one-

shot), lorsqu'un joueur se tait, l'autre a tout intérêt à dénoncer (il aura une utilité de 3 au lieu de 2 s'il se tait). C'est cette constatation qui fait qu'au final, les deux joueurs se dénoncent. Mais dans le jeu répété, c'est différent. En effet, admettons que le joueur 1 choisisse d'opter pour la stratégie explicitée. Si le joueur 2 décide de toujours dénoncer le joueur 1, il aura une utilité de 3 au premier tour, mais une utilité de 1 seulement à tous les tours suivants. Alors qu'en adoptant la même stratégie que le joueur 1, il gagnera une utilité de 2 à chaque tour ! Certes, le joueur 1 peut être tenté d'accroître momentanément son utilité, mais il est sous la « menace » d'une punition de l'autre joueur.

Ce que le joueur 2 choisira de faire dépendra de sa *patience*. En effet, un joueur relativement patient préférera avoir une utilité de 2 à chaque tour et maximiser son utilité à long terme. Si le joueur 2 choisit de coopérer, et qu'il est extrêmement patient (il accorde autant de valeur à une utilité future qu'immédiate), au bout de 3 tours, il est gagnant, puisqu'il obtiendra une utilité de 6 au lieu de 5. Au bout de 10 tours, il aura gagné une utilité de 20 au lieu de 12 s'il avait dénoncé l'autre. Par contre, si le joueur est extrêmement impatient (il n'accorde aucune valeur à une utilité autre que l'immédiate, c'est-à-dire celle du premier tour), il préférera dénoncer l'autre. En fait, on peut voir un jeu répété où les deux joueurs sont extrêmement impatients comme un jeu stratégique (one-shot). Dans le jeu répété du dilemme du prisonnier, cette stratégie de coopération est un équilibre de Nash si les deux joueurs sont suffisamment patients.

Ce n'est cependant pas le seul équilibre de Nash. Le profil de stratégies où les deux joueurs décident de toujours se dénoncer est aussi un équilibre de Nash. En effet, aucun des deux joueurs ne prendra le risque de se taire sachant que l'autre le dénoncera toujours.

### 3.4.2 Préférences

Le résultat d'un jeu répété est la somme des résultats du jeu stratégique. Chaque terme de la somme est pondéré, de manière à ce qu'un terme au début de la somme ait plus de valeur qu'un terme à la fin de la somme. Cette pondération reflète la patience du joueur.

DÉFINITION 3.14. Chaque joueur  $i$  a une fonction d'utilité  $u_i$  pour le jeu stratégique, et un facteur « d'actualisation »  $\delta_i$  entre 0 et 1 tel qu'il évalue la séquence  $(a_1, \dots, a_T)$  de résultats du jeu stratégique comme la somme suivante :

$$u_i(a_1) + \delta_i u_i(a_2) + \delta_i^2 u_i(a_3) + \dots + \delta_i^{T-1} u_i(a_T) = \sum_{t=1}^T \delta_i^{t-1} u_i(a_t)$$

On a  $\delta \in [0..1]$ . Si  $\delta = 1$ , le joueur est extrêmement patient et n'accorde pas plus de valeur à une utilité immédiate qu'à une utilité future. Si  $\delta = 0$ , le joueur est extrêmement impatient et n'accorde de la valeur qu'au premier terme de la somme. Les valeurs de  $\delta$  comprises entre 0 et 1 permettent de fixer différents « niveaux » de patience. Souvent, dans un jeu répété, on considère que tous les joueurs ont la même patience, c'est-à-dire que  $\delta_i = \delta \forall i$ .

#### 3.4.3 Définition d'un jeu répété

DÉFINITION 3.15. Soit  $G$  un jeu stratégique. Le jeu répété à  $T$  périodes de  $G$  avec un facteur d'actualisation  $\delta$  est le jeu étendu avec information parfaite et mouvements simultanés dans lequel :

- l'ensemble des joueurs est  $N$  ;
- l'ensemble des histoires terminales est l'ensemble des séquences  $(a_1, \dots, a_T)$  de profils d'actions dans  $G$  ;
- la fonction joueur assigne l'ensemble de tous les joueurs à chaque histoire  $(a_1, \dots, a_T) \forall t \in [1..T]$  ;
- l'ensemble d'actions disponibles à chaque joueur  $i$  après chaque histoire est  $A_i$  ;
- chaque joueur  $i$  évalue chaque histoire terminale  $(a_1, \dots, a_T)$  en accord avec son utilité moyenne actualisée :

$$(1 - \delta) \sum_{t=1}^T \delta^{t-1} u_i(a^t)$$

#### 3.4.4 Jeu répété fini et infini

Nous avons dit ci-dessus (section 3.4.1) que dans le dilemme du prisonnier répété, la stratégie consistant à se taire tant que l'autre se tait était un équilibre de Nash. En fait, cela n'est vrai que si le jeu est infini (le dilemme est répété indéfiniment). Il est possible de montrer que si le jeu répété est fini (le jeu est répété un nombre  $x$  de fois), alors le seul équilibre de Nash est le profil de stratégie consistant à toujours dénoncer l'autre. Nous n'explicitons pas le raisonnement, car cela dépasserait à notre sens le cadre de cette présentation de la théorie des jeux. Il est aussi possible de démontrer que si le jeu répété est infini, la stratégie de coopération est un équilibre de Nash si  $\delta \geq \frac{1}{2}$ . De même, on peut montrer que cette stratégie n'est pas un équilibre parfait de sous jeu.

#### L'équilibre de Nash dans un jeu répété infini

Nous devons en premier lieu définir ce que sont les « utilités faisables », et « l'utilité minmax. »

DÉFINITION 3.16. On appelle « ensemble de profils d'utilités faisables d'un jeu stratégique » l'ensemble de toutes les moyennes pondérées possibles de profils d'utilités dans le jeu.

Si le facteur d'actualisation est proche de 1, l'ensemble des profils d'utilités moyennes actualisées générés par un jeu répété infini est approximativement égal à l'ensemble des profils d'utilités faisables du jeu stratégique qui est répété.

DÉFINITION 3.17. On appelle l'utilité minmax d'un jeu stratégique :

$$\min_{a_{-i} \in A_{-i}} (\max_{a_i \in A_i} u_i(a_i, a_{-i}))$$

où pour chaque joueur  $i$ ,  $A_i$  est son ensemble d'actions, et  $u_i$  est sa fonction d'utilité.

A partir de là, on peut exposer le théorème de Nash (en anglais, *Nash folk theorem*) :

DÉFINITION 3.18. Soit  $G$ , un jeu stratégique dans lequel chaque joueur a un ensemble fini d'actions.

- $\forall \delta$  avec  $0 < \delta < 1$ , l'utilité moyenne affaiblie de chaque joueur dans tout équilibre de Nash d'un jeu infiniment répété de  $G$  est au moins son utilité minmax.
- Soit  $w$ , un profil d'utilités faisables de  $G$  pour lequel l'utilité de chaque joueur excède leur utilité minmax. Il existe un  $\delta_{lim}$  tel que si  $\delta > \delta_{lim}$ , alors le jeu infiniment répété de  $G$  a un équilibre de Nash dans lequel la moyenne affaiblie de l'utilité de chaque joueur  $i$  est  $w_i$ .
- Si  $G$  a un équilibre de Nash dans lequel l'utilité de chaque joueur est leur utilité minmax, alors pour chaque valeur de  $\delta$ , le jeu infiniment répété  $G$  a un équilibre de Nash dans lequel la moyenne actualisée de l'utilité de chaque joueur  $i$  est son utilité minmax.

Ce théorème est aussi valable pour l'équilibre parfait de sous-jeu.

### L'équilibre de Nash dans un jeu répété fini

Le théorème de Nash est défini comme suit pour les jeux répétés finis :

DÉFINITION 3.19. Soit  $G$ , un jeu stratégique qui, pour chaque joueur, a un équilibre de Nash dans lequel l'utilité du joueur excède son utilité minmax. Soit  $w$ , un profil d'utilités faisables de  $G$  pour lequel l'utilité de chaque joueur excède leur utilité minmax.  $\forall \epsilon > 0$ , il existe un nombre entier  $T^*$  tel que : si  $T > T^*$ , alors le jeu répété à  $T$  périodes de  $G$  avec un facteur d'actualisation 1 a un équilibre de Nash dans lequel l'utilité moyenne de chaque joueur  $i$  est dans  $w_i + -\epsilon$ .

### L'équilibre parfait de sous-jeu dans un jeu répété fini à deux joueurs

DÉFINITION 3.20. Soit  $G$ , un jeu stratégique à deux joueurs qui, pour chaque joueur, a deux équilibres de Nash dans lequel les utilités du joueur diffèrent. Soit  $w$ , un profil d'utilités faisables de  $G$  pour lequel l'utilité de chaque joueur excède leur utilité minmax.  $\forall \epsilon > 0$ , il existe un nombre entier  $T^*$  tel que : si  $T > T^*$ , alors le jeu répété à  $T$  périodes de  $G$  avec un facteur d'actualisation 1 a un équilibre de Nash dans lequel l'utilité moyenne de chaque joueur  $i$  est dans  $w_i + -\epsilon$ .

# Chapitre 4

## Le problème

### 4.1 Problème de base et solution existante

Comme expliqué dans l'introduction, le problème de base concerne l'entrée d'un service web dans une communauté. Il a été montré dans ([BK11a]) que pour certaines raisons, il pouvait être avantageux pour plusieurs services web offrant un même type de service de se rassembler en une communauté. L'exemple donné dans la sous-section 1.1.1 explique cela de manière concrète. Un service web a une certaine qualité de service. Une communauté a intérêt à intégrer en son sein un service web ayant une bonne qualité de service plutôt qu'un service web en ayant une mauvaise. En effet, la réputation de la communauté toute entière pourrait pâtir de la présence d'un service web avec une mauvaise qualité de service. Avant d'accepter un nouveau service web, la communauté doit donc en premier lieu s'assurer que celui-ci est fiable, en calculant sa réputation.

Nous considérons donc un réseau distribué de services web et d'utilisateurs, où certains services web sont assemblés en communauté. Une communauté est représentée par un certain service web que nous appelons le « maître de communauté ». C'est lui qui va décider d'accepter ou non un nouveau service web dans la communauté. Pour cela, le maître va faire appel à certains services web appelés « services d'information » dont le rôle est de fournir l'information dont a besoin le maître pour prendre sa décision. On considère donc qu'il existe dans le réseau un certain nombre de services d'information qui possèdent de l'information sur tous les services web, mais à différents degrés d'exactitude. On peut voir les services d'information comme des espèces de « testeurs » de services web, qui possèdent une valeur de réputation pour chacun d'eux, consistant en une notation de la qualité de service.

**DÉFINITION 4.1.** Une notation d'un service web par un service d'information est un nombre compris dans l'intervalle  $[0, 1]$  notant sa qualité de service, telle qu'elle a été mesurée par le service d'information. Une notation de 1 signifie que le service d'information considère que le service web ne peut pas faire mieux, et une notation de 0 signifie que le service d'information considère que le service web ne peut pas faire pire.

On considère qu'un service d'information peut soit dire la vérité et transmettre l'information qu'il possède sur un certain service web, soit mentir et transmettre une fausse

information.

Le maître de communauté rémunère chaque service d'information qu'il consulte. Nous devons faire en sorte que le paiement donné incite les services d'information à dire la vérité. Pour cela, nous nous basons sur un modèle existant. Nous en dériverons un nouveau modèle adapté aux exigences qui nous ont été fixées.

## 4.2 Le modèle existant

Il s'agit d'un modèle mis en place par deux étudiants des FUNDP lors de leur stage (2010-2011). Celui-ci avait fait l'objet d'un mémoire [CK11]. Ce modèle consiste en une fonction de paiement à 3 composantes incitatives. Nous allons donc commencer par présenter le travail déjà effectué puisqu'il s'agira de notre point de départ. Pour cela, nous nous basons essentiellement sur l'article [BK11b].

### 4.2.1 La fonction de paiement

On considère l'utilité  $u_k(x)$  d'un service d'information  $k$  choisissant une stratégie  $x$  (les deux stratégies possibles étant de mentir ou de dire la vérité) comme le paiement qu'il reçoit du maître de communauté. Ce paiement est déterminé par une fonction ayant trois composantes incitatives :

$$u_k(x) = f_k(x) + g_k(x) + c \cdot h_k(x)$$

PROPRIÉTÉ 4.1.

$$g_{k_{max}} > f_{k_{max}}$$

$$h_{k_{max}} > g_{k_{max}} + f_{k_{max}}$$

Le principe de cette propriété est que chaque composante de paiement soit possiblement plus grosse que la précédente. Cette propriété fait en sorte qu'un service d'information préfère obtenir un bon paiement  $h_k(x)$  et un mauvais paiement  $g_k(x)$  que le contraire.

Le premier composant  $f_k(x)$  est un paiement donné par un maître de communauté au service d'information  $k$  pour que ce dernier lui transmette l'information demandée.

Le second composant  $g_k(x)$  est un paiement qui dépendra de la similitude entre l'information donnée par le service d'information  $k$  et l'information obtenue par les autres services d'information. Ce second paiement peut éventuellement être négatif si la différence d'information est énorme et il agit alors comme une « punition ».

Le troisième et dernier paiement dépendra de la similitude entre l'information donnée par le service d'information et ce qui est véritablement « expérimenté » par la communauté après avoir accepté le service web. Ce troisième paiement ne peut avoir lieu que si le service web est accepté par le maître de communauté. Le rôle de la variable  $c$  s'explique alors :

- $c = 0$  si le service web n'est pas accepté dans la communauté ;

## 4.2 Le modèle existant

---

- $c = 1$  sinon.

Ce troisième paiement peut lui aussi être éventuellement négatif et agir comme une punition.

La fonction de paiement  $u_k(x)$  satisfait les trois propriétés suivantes :

- dire la vérité est une stratégie menant à un équilibre de Nash ;
- si le service web est non fiable (a une mauvaise qualité de service), mentir en transmettant une bonne notation au maître de communauté est une stratégie ne menant pas à un équilibre de Nash ;
- si le service web est fiable (a une bonne qualité de service), mentir en transmettant une mauvaise notation au maître est une stratégie menant à un équilibre de Nash.

Pour rappel, plusieurs équilibres de Nash sont possibles dans un jeu stratégique.

La première propriété dit que le cas où tous les services d'information disent la vérité est un équilibre de Nash. En effet, étant donné que les autres disent la vérité, un service d'information n'a aucun intérêt à changer de stratégie en mentant car il obtiendrait un mauvais paiement  $g_k(x)$ .

La seconde propriété dit que la stratégie consistant à mentir lorsqu'un service web est non fiable ne constitue pas un équilibre de Nash. En effet, étant donné que les autres mentent, un service d'information a tout intérêt à dire la vérité. Il obtiendrait alors un mauvais paiement  $g_k(x)$  mais un bien meilleur paiement  $h_k(x)$ .

Par contre, la troisième propriété affirme que la stratégie consistant à mentir lorsqu'un service web est fiable constitue un équilibre de Nash. En effet, étant donné que les autres mentent, un service d'information n'a pas intérêt à dire la vérité. Il obtiendrait alors un mauvais paiement  $g_k(x)$  qui ne sera pas contrebalancé par un éventuel paiement  $h_k(x)$  puisque le service web ne sera pas accepté et que  $c = 0$ .

Cette troisième propriété pose donc problème. Elle affirme en effet que mentir en donnant une mauvaise notation à un service web fiable constitue un équilibre de Nash. L'équilibre de Nash est un concept théorique permettant de révéler des situations qui sont « à l'équilibre ». C'est ce type de situation qui risque d'arriver dans un cas réel. On voit donc ici que la fonction de paiement  $u_k(x)$  bien qu'incitant les services d'information à dire la vérité (cette stratégie est un équilibre de Nash), ne les « empêche pas totalement de mentir » (la stratégie consistant à mentir est un équilibre de Nash si le service web est fiable).

On obtient donc deux cas possibles :

- soit le service web a une bonne qualité de service, et les deux stratégies (mentir et dire la vérité) sont un équilibre de Nash ;
- soit le service web a une mauvaise qualité de service, et seule la stratégie consistant à dire la vérité est un équilibre de Nash.

Toutes choses égales par ailleurs, bien que dans le premier cas, les deux stratégies constituent un équilibre de Nash, il semble logique qu'un agent rationnel choisisse de dire la vérité puisque chaque service d'information gagnera plus d'argent si tout le monde dit la vérité que si tout le monde ment.

Le fait que les paiements  $g_k(x)$  et  $h_k(x)$  puissent être négatifs signifie que le paiement total  $u_k(x)$  peut lui aussi être négatif. Permettre des paiements négatifs consiste à poser une hypothèse forte, puisqu'on considère alors que les services d'information ont un « portefeuille ouvert », c'est-à-dire qu'un maître peut aussi bien y déposer de l'argent qu'en prendre. C'est l'hypothèse qui est prise par l'article sur lequel nous nous basons : *Game-Theoretic Analysis of a Web Services Collaborative Mechanism* [BK11b]. Par contre, dans le mémoire présenté par les stagiaires l'année dernière [CK11], cette hypothèse n'est pas présente. Les paiements ne peuvent pas être négatifs, au pire ils sont nuls. La présence ou l'absence de cette hypothèse n'agit en rien sur la vérité des trois propriétés énoncées ci-dessus. Elle agit par contre sur le caractère de « punition » que peut prendre un paiement. Dans notre recherche, nous avons travaillé en considérant qu'on puisse choisir un système avec paiements négatifs ou sans. Nous en reparlerons dans le chapitre 5.

#### 4.2.2 La corruption

Nous venons de le voir, le système de paiements à 3 composantes incitent les services d'information à être honnêtes. Mais nous n'avons pas encore pris en compte le fait que les services d'information puissent être corrompus avec un paiement accordé par un service web. Le problème se pose lorsque les services d'information sont corrompus par le service web qui désire entrer dans une communauté. En effet, si celui-ci n'a pas une bonne qualité de service, il pourrait tenter de corrompre un service d'information en lui versant un paiement afin que celui-ci valorise la notation qu'il donne au service web en question. Nous notons ce paiement  $\pi$ .

#### 4.2.3 Les différents paiements du modèle de départ

Dans la suite, nous considérerons un maître de communauté  $M_i$  désirant évaluer un service web  $S_j$  qui désire entrer dans la communauté de  $M_i$ , et un service d'information  $I_k$ . Les paiements que  $I_k$  reçoit peuvent soit venir de  $M_i$ , soit de  $S_j$  (corruption). Par simplification d'écriture, nous utilisons  $\alpha$ ,  $\beta$ , et  $\gamma$  pour désigner les trois composantes  $f_k(x)$ ,  $g_k(x)$ , et  $h_k(x)$ .

Les services d'information pouvant mentir, le maître  $M_i$  accorde une certaine confiance au service d'information  $I_k$ . La manière dont cette confiance est calculée n'apparaît pas dans la fonction de paiement. Comme vu dans la section 2.4, nous utilisons le système BRS, mais cela n'a pas d'importance dans ce chapitre. La valeur de confiance accordée par le maître  $M_i$  à un service d'information  $I_x$ , notée  $Tr_{M_i}^{I_x}$ , est telle que :  $Tr_{M_i}^{I_x} \in [0, 1]$ .

On note  $Rr_{I_x}^{S_j}$  la notation donnée par un service d'information  $I_x$  au service web  $S_j$ . Comme dit plus haut :  $Rr_{I_x}^{S_j} \in [0, 1]$ . Enfin, on note  $Ro_{M_i}^{S_j}$  la notation donnée par le maître

## 4.2 Le modèle existant

---

$M_i$  au service web  $S_j$ .  $Ro_{M_i}^{S_j} \in [0, 1]$ .  $Ro_{M_i}^{S_j}$  n'existe que si le service web a été accepté au sein de la communauté.

Afin de savoir s'il va accepter ou refuser un service web au sein de sa communauté, le maître calcule la réputation du service web en faisant une moyenne pondérée des notations qu'il a reçues des services d'information :

$$\frac{\sum_{x=0}^n Tr_{M_i}^{I_x} Rr_{I_x}^{S_j}}{\sum_{x=0}^n Tr_{M_i}^{I_x}}$$

où  $n$  est le nombre de service d'information ayant transmis une notation au maître.

La moyenne étant pondérée par la réputation des services d'information, une notation transmise au maître par un service d'information hautement réputé aura un plus grand poids dans la moyenne qu'une notation transmise par un service d'information peu réputé. En effet, comme dit dans le chapitre 2, il est logique d'accorder une plus grande importance à une information obtenue d'un service d'information envers lequel on a une grande confiance (réputation proche de 1), qu'à une information obtenue d'un service d'information en lequel on a peu ou moins confiance.

### Le paiement $\alpha$

Ce premier paiement a pour but d'inciter les services d'information à transmettre leur information en les payant pour cela. Il s'agit d'une valeur constante.

### Le paiement $\beta$

Ce second paiement consiste en une fonction logarithmique décroissante sur la différence entre la notation moyenne du service web évalué, calculée par le maître de communauté en fonction de toutes les notations fournies, et la notation fournie par  $I_k$ . Sa forme est la suivante :

$$\beta = \log_b \left( \left| \frac{\sum_{x=0}^n Tr_{M_i}^{I_x} Rr_{I_x}^{S_j}}{\sum_{x=0}^n Tr_{M_i}^{I_x}} - Rr_{I_k}^{S_j} \right| \right)$$

avec  $0 < b < 1$  pour rendre la fonction décroissante.

La fraction réalise la moyenne de toutes les notations reçues. Cette moyenne est pondérée par les valeurs de confiance (réputation) accordées aux services d'information.

### Le paiement $\gamma$

Ce troisième paiement consiste en une fonction logarithmique décroissante sur la différence entre la notation du service web attribuée par le maître après un certain temps d'activité dudit service web au sein de la communauté, et la notation fournie par  $I_k$ . Sa forme est la suivante :

$$\gamma = \log_b (|Rr_{I_x}^{S_j} - Ro_{I_k}^{S_j}|)$$

avec  $0 < b < 1$  pour rendre la fonction décroissante.

### Le paiement $\pi$

Le paiement  $\pi$  est le paiement donné par  $S_j$  pour tenter de corrompre un service d'information. On considère qu'un paiement  $\pi$  proposé par  $S_j$  à un service d'information  $I_x$  n'a lieu que si  $M_i$  décide d'intégrer  $S_j$  à sa communauté (le paiement n'a lieu que si le mensonge a porté ses fruits). Comme nous allons le voir dans la sous-section suivante, un paiement  $\pi$  pousse les services d'information à mentir lorsqu'il dépasse un certain seuil.

### Les autres notations de paiements

$\beta^-$  et  $\gamma^-$  désignent des punitions, c'est-à-dire des paiement négatifs si on permet ceux-ci. Si les paiements négatifs ne sont pas permis,  $\beta^-$  et  $\gamma^-$  représentent des paiements nuls.

#### 4.2.4 Les différents cas possibles

Nous allons maintenant voir les différents jeux qui sont modélisés selon les cas. On considère qu'un jeu est composé de deux joueurs : un service d'information (noté SI) et l'ensemble des autres services d'information (noté ASI).

#### $S_j$ a une bonne qualité de service

Le premier cas est celui où le service web désirant entrer dans une communauté a une bonne qualité de service. Il sait qu'il a toutes ses chances d'entrer dans la communauté et ne tentera donc pas de corrompre les services d'information.

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta + \gamma), (\alpha + \beta + \gamma)$	$(\alpha + \beta^-), (\alpha + \beta_-)$
	Mentir	$(\alpha + \beta^- + \gamma^-), (\alpha + \beta_- + \gamma)$	$(\alpha + \beta), (\alpha + \beta)$

TABLE 4.1 – Jeu lorsque  $S_j$  a une bonne qualité de service.

Analysons les différents résultats possibles. Si les deux joueurs disent la vérité, le service web est accepté dans la communauté et les deux joueurs gagnent chacun  $\alpha + \beta + \gamma$ . Si SI ment alors que ASI dit la vérité, le service web est accepté dans la communauté et SI gagne  $\alpha + \beta^- + \gamma^-$ . ASI ayant dit la vérité, il gagne  $\alpha + \beta_- + \gamma$ .  $\beta_-$  ne représente pas une punition mais une récompense légèrement plus petite. En effet, le paiement  $\beta$  est déterminé en fonction de la différence de la notation transmise par le service d'information et la moyenne de toutes les notations. Le fait que SI mente éloigne un peu la notation moyenne de celle transmise par ASI, ce qui explique son paiement  $\beta$  légèrement inférieur.

Pour éclaircissement,  $\alpha$  est donc le premier paiement reçu, le paiement  $f_k(x)$ , le même pour tous.  $\beta$  représente le paiement reçu lorsque tous les services d'informations disent la même chose. Il s'agit de la valeur maximale de  $g_k(x)$ .  $\beta_-$  représente le paiement reçu

## 4.2 Le modèle existant

---

lorsque tous les services d'information disent la même chose sauf un. Elle est un peu plus petite que  $\beta$ .  $\beta^-$  représente le paiement reçu lorsque les autres services d'information ne disent pas la même chose. Selon l'hypothèse choisie, il peut être :

- un paiement nul ;
- un paiement négatif.

$\gamma$  représente le paiement reçu lorsque le maître de communauté obtient la même évaluation du service web. Il s'agit de la valeur maximale de  $h_k(x)$ .  $\gamma^-$  représente le paiement reçu lorsque le maître a une évaluation différente du service web. Selon l'hypothèse choisie, il peut être :

- un paiement nul ;
- un paiement négatif.

Si ASI ment, le service web n'est pas accepté au sein de la communauté et aucun paiement  $\gamma$  n'est versé. Si SI ment aussi, les deux joueurs gagnent  $\alpha + \beta$ . Par contre, si SI dit la vérité, il gagne  $\alpha + \beta^-$  (il est puni car ce qu'il dit s'éloigne fortement de ce que les autres disent). ASI gagne alors  $\alpha + \beta_-$ .

Dans ce jeu, dire la vérité est le seul équilibre de Nash. De plus, c'est aussi un optimum de Pareto.

### $S_j$ a une mauvaise qualité de service

Si le service web désirant entrer dans une communauté a une mauvaise qualité de service, deux cas sont possibles. Il peut soit être honnête et attendre la décision du maître de communauté, soit être malhonnête et tenter de corrompre les services d'information en leur offrant un paiement  $\pi$ .

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta), (\alpha + \beta)$	$(\alpha + \beta^- + \gamma), (\alpha + \beta_- + \gamma^-)$
	Mentir	$(\alpha + \beta^-), (\alpha + \beta_-)$	$(\alpha + \beta + \gamma^-), (\alpha + \beta + \gamma^-)$

TABLE 4.2 – Jeu lorsque  $S_j$  a une mauvaise qualité de service et est honnête.

**$S_j$  a une mauvaise qualité de service et est honnête.** Ce cas est représenté par le jeu de la table 4.2. Dire la vérité est encore une fois le seul équilibre de Nash. C'est aussi un optimum de Pareto. En effet, SI n'a pas intérêt à changer de stratégie et mentir car il gagnerait  $\alpha + \beta^-$  au lieu de  $\alpha + \beta$ . De même pour ASI qui gagnerait  $\alpha + \beta_- + \gamma^-$  au lieu de  $\alpha + \beta$ . On peut facilement vérifier qu'aucun des trois autres résultats possibles ne sont des équilibres de Nash.

**$S_j$  a une mauvaise qualité de service et est malhonnête** Ce cas est représenté par le jeu de la table 4.2.

ASI		
	Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta), (\alpha + \beta)$
	Mentir	$(\alpha + \beta^- + \gamma), (\alpha + \beta_- + \gamma^- + \pi)$
		$(\alpha + \beta^-), (\alpha + \beta_-)$
		$(\alpha + \beta + \gamma^- + \pi), (\alpha + \beta + \gamma^- + \pi)$

TABLE 4.3 – Jeu lorsque  $S_j$  a une mauvaise qualité de service et est malhonnête.

LEMME 4.1. Si  $\gamma^- + \pi < \beta_- - \beta$  ou  $\gamma - \gamma^- > \beta - \beta^- + \pi$ , alors dire la vérité est le seul équilibre de Nash (et optimum de Pareto). Sinon, mentir est le seul équilibre de Nash (et optimum de Pareto).

LEMME 4.2.  $\gamma^- + \pi < \beta - \beta^-$  est une condition suffisante pour que dire la vérité soit un équilibre de Nash.

### 4.3 Notre sujet

Notre travail consiste à répondre à de nouveaux problèmes qui ne sont pas pris en charge par le modèle ci-dessus. En reprenant le concept de la formule de paiement à 3 composantes, nous allons élaborer un nouveau mécanisme d'incitation répondant le plus efficacement possible à ces problèmes supplémentaires.

Le premier problème consiste à s'assurer que la fonction de paiement est juste, c'est-à-dire qu'elle rétribue chaque service d'information correctement en fonction de la valeur de l'information transmise. Georgia Kastidou, Kate Larson, Robin Cohen exposent dans leur article *Exchanging Reputation Information Between Communities : A Payment-Function Approach ([GK09])* une série de conditions que doit respecter une formule de paiement afin de rétribuer les services d'information correctement, de manière « juste ». Comme nous le verrons dans le chapitre 5, la fonction de paiement présentée ci-dessus ne respecte pas ces conditions. Nous allons donc la redéfinir en ne gardant que le concept de la séparation en trois composants. Nous redéfinirons les paiements  $g_k(x)$  et  $h_k(x)$  en justifiant nos choix.

Le second problème qui se pose est celui de la concurrence des services web. Le modèle présenté ci-dessus considère qu'une tentative de corruption ne peut venir que d'un service web ayant une mauvaise qualité de service et désirent que les services d'information la valorise. Cependant, les systèmes multi-agents sont des systèmes complexes où la concurrence a lieu. Par conséquent, lorsqu'un service web ayant une bonne qualité de service désire entrer dans une certaine communauté, un agent tiers (un service web faisant déjà partie de la communauté et voyant d'un mauvais œil l'entrée d'un nouveau service web, ou encore un service web concurrent désirent aussi entrer dans la communauté) pourrait tenter de corrompre les services d'information afin que ceux-ci dévalorisent la notation qu'ils donnent au service web désirent entrer dans la communauté. Ce cas spécifique que nous avons étudié fera l'objet du chapitre 6.

Dans les chapitres 5 et 6, nous modifierons donc le modèle de départ, afin d'obtenir un nouveau modèle conforme aux exigences énoncées, c'est à dire une fonction de paiement

juste, et la prise en compte du nouveau cas de corruption possible.

Enfin, les jeux présentés ci-dessus dans la section 4.2.4 sont des jeux stratégiques de base, des « one-shot ». Ils ne sont utiles que pour étudier le comportement de services d'information agissant en pensant à court terme. Nous allons tenter de faire une étude sur le long terme en utilisant les jeux répétés. Nous aimerions faire en sorte que les services d'information soient plus incités à dire la vérité lorsqu'ils agissent sur le long terme que sur le court terme. Cette étude sur le long terme fera l'objet du chapitre 7.

Dans le chapitre 8 nous exposerons une série de simulations effectuées avec un simulateur (présenté en annexe). Ce simulateur utilise les jeux répétés afin que les services d'information prennent leurs décisions en réfléchissant sur le long terme. Nous espérons ainsi obtenir des simulations dont les résultats soient les plus proches possible de la réalité, avec des services d'information agissant plus ou moins à long terme, selon leur degré de patience. Une présentation du simulateur est faite dans l'annexe A.



## Chapitre 5

# Élaboration d'une nouvelle fonction de paiement

Ce chapitre présente la nouvelle fonction de paiement que nous avons développée. Nous allons commencer par un bref rappel de la forme de la fonction de paiement présentée dans le chapitre précédent. Ensuite, dans la section 5.2, nous présenterons les conditions que doit respecter une fonction de paiement pour être juste (nous en avons brièvement parlé dans la section 4.3). Ces conditions se présentent sous la forme de 5 propriétés. Dans la section 5.3, nous développerons la nouvelle fonction  $h_k(x)$  de manière à ce qu'elle respecte les 5 propriétés en question. Dans la section 5.4, nous vérifierons que la nouvelle fonction  $h_k(x)$  respecte bien les 5 propriétés, et nous l'illustrerons dans la section 5.5 avec des graphes de la fonction. Enfin, nous présenterons la nouvelle fonction  $g_k(x)$  dans la section 5.6, et nous ferons un récapitulatif du nouveau modèle ainsi créé dans la section 5.7.

### 5.1 Introduction

Pour rappel, la structure de la fonction de paiement est :

$$u_k(x) = f_k(x) + g_k(x) + c.h_k(x)$$

$f_k(x)$  donne un paiement de base (un montant fixé) à chaque service d'information sélectionné par le maître de communauté, afin que ceux-ci transmettent une notation.  $g_k(x)$  donne le deuxième paiement, qui dépend de la réputation calculée par le maître en prenant la moyenne de toutes les notations reçues (comme vu dans la section 4.2). Selon la différence entre la moyenne des notations reçues par le maître et celle transmise par le service d'information à payer, le paiement est calculé et donné à celui-ci. Chaque service d'information a une incitation à donner l'évaluation la plus proche de la moyenne, et donc à dire la vérité. Cependant, nous ne sommes pas à l'abri d'un problème de collusion (si tous les services d'information mentent, ils maximisent aussi  $g_k(x)$ ).  $h_k(x)$  donne le paiement final, qui dépend de l'évaluation du maître de communauté lui-même lorsqu'il accepte le service web (après que celui-ci ait fonctionné un certain temps comme membre de la communauté).  $c$  est une constante égale à 1 si le maître de communauté accepte le service web, et 0 sinon.  $h_k(x)$  n'est donc distribué que si le service web est accepté au sein de la communauté.

Nous développons dans ce chapitre les changements effectués sur la fonction de paiement telle que présentée dans le chapitre 4, et le raisonnement qui nous a conduit à l'élaboration d'une nouvelle. Le but est de rendre les paiements les plus justes possible envers les services d'information. Pour cela, la fonction de paiement doit rémunérer chaque service d'information par rapport à la qualité de l'information transmise. Pour juger de la qualité d'une information, le maître doit comparer la notation du service d'information avec sa propre notation. Ces considérations ne concerneront donc que la composante  $h_k(x)$  de la fonction de paiement.

Déterminer la qualité d'une information n'est pas facile. Chaque service d'information possède son propre mécanisme de notation, et les notations d'un même service web par plusieurs services d'information peuvent être très variables. Le principe de la fonction  $h_k(x)$  est d'être décroissante sur la différence entre les notations du maître et du service d'information. Autrement dit, plus la différence entre la notation du maître et celle transmise par le service d'information est grande, plus le paiement accordé est petit. Cependant, comme nous allons le voir dans les sections suivantes, ce principe ne suffit pas à avoir une fonction juste. Par exemple, si un maître note un service web à 0,80, un service d'information l'ayant noté 0,85 devrait recevoir un meilleur paiement qu'un service d'information l'ayant noté à 0,75, alors que la différence est pourtant la même (0,05). Nous verrons pourquoi dans la section suivante.

Faire en sorte d'avoir une fonction de paiement la plus juste possible semble n'avoir que des avantages. On peut penser que cela incitera les services d'information à être plus honnêtes puisqu'ils seront rémunérés à leur juste valeur. Cependant, élaborer une fonction de paiement « juste » n'est pas un moyen de garantir que nous sommes à l'abri d'actes malhonnêtes. Un service d'information, agent rationnel souhaitant maximiser ses revenus, peut toujours être enclin à mentir si on lui propose un « pot-de-vin » conséquent.

Afin de rendre la fonction  $h_k(x)$  plus juste, nous allons nous inspirer de l'article [GK09] présentant 5 propriétés qu'une fonction de paiement doit respecter pour être considérée comme telle.

## 5.2 5 propriétés pour rendre une fonction de paiement juste

Nous allons commencer par présenter rapidement l'article [GK09].

### 5.2.1 Le contexte de base de l'article [GK09]

Le contexte de base de l'article [GK09] est légèrement différent du nôtre puisqu'il s'agit de communautés s'informant auprès d'autres communautés et non pas de communautés s'informant auprès de services d'information. Le contexte est en effet le suivant. Un service web peut faire partie de plusieurs communautés. Lorsqu'un certain service web veut entrer dans une nouvelle communauté, le maître de celle-ci consulte les maîtres des communautés auxquelles le service web appartient déjà, afin d'avoir une appréciation de sa qualité de

## 5.2 5 propriétés pour rendre une fonction de paiement juste

---

service. Cependant, les autres communautés n'ayant pas forcément intérêt à ce qu'un de leurs services web entre dans une nouvelle communauté, les maîtres de celles-ci peuvent mentir en fournissant une information inexacte. Une fonction de paiement est alors mise en place. Celle-ci maximise le paiement lorsqu'une bonne information est fournie. De plus, les auteurs introduisent un ensemble de propriétés qu'une fonction de paiement devrait respecter pour être juste et promouvoir l'honnêteté.

Ce contexte est très proche du nôtre, la différence étant que dans notre cas, un service web ne fait partie que d'une communauté, et que ce sont les services d'information qui détiennent et transmettent des notations sur les services web. Nous allons donc dans la suite substituer notre contexte au leur.

### 5.2.2 Le modèle de l'article [GK09]

Notons  $C_i$  la communauté  $i$  et  $S_j$  le service web  $j$ . Notons  $I_k$  le service d'information  $k$ . Celui-ci possède une évaluation de  $S_j$  que l'on note  $r_j^k$ . A cause du fait que les systèmes de notation utilisés par les services d'information sont éventuellement différents, et que l'interprétation d'une notation peut différer selon l'agent, les services d'information doivent transmettre d'une part, la notation qu'ils donnent chacun à  $S_j$ , mais aussi une seconde information, notée  $\theta_j^k$ , qui consiste à dire si le service d'information trouve au final le service web bon ou mauvais.  $\theta_j^k$  est donc une information binaire :  $\theta_j^k \in \{\text{bon}, \text{mauvais}\}$ . Cette seconde information est en quelque sorte l'interprétation que le service d'information donne de sa notation.

Le paiement que va recevoir un service d'information dépendra de la qualité, de l'utilité, ou encore de l'importance de l'information qu'a donnée le service d'information à  $C_i$ . Le maître de  $C_i$  va prendre sa décision en tenant compte des différentes notations reçues. S'il accepte le service web  $S_j$ ,  $C_i$  pourra donc l'observer et avoir son propre jugement consistant en une évaluation notée  $r$ , et en une interprétation notée  $\theta_j^i$ . Si  $C_i$  n'accepte pas  $S_j$ , elle décide alors que  $\theta_j^i = \text{mauvais}$  et calcule  $r$  comme étant la moyenne des notations accompagnées d'une interprétation de type *mauvais*.

$$r = \frac{\sum_{I_x \in B} r_j^x}{\#B}$$

avec  $B$  l'ensemble des services d'information ayant transmis une interprétation *mauvais*.

La fonction de paiement pour le service d'information  $I_k$  est la suivante :

$$P(r_j^k, r, \theta_j^i) = \phi.I(r_j^k, r, \theta_j^i) + \omega$$

$I$  est la fonction d'importance utilisée pour déterminer l'importance de l'information de  $I_k$  dans la décision, c'est-à-dire la qualité de l'information  $I_k$ . C'est sur cette fonction que s'appliquent les 5 propriétés que nous présenterons dans la sous-section 5.2.3.

### Comparaison avec notre fonction de paiement

Comparons la fonction de paiement de l'article [GK09] avec la nôtre. Leur  $\omega$  s'apparente à notre  $f_k(x)$ . Il s'agit d'un paiement fixe qui a toujours lieu. Quant à la première composante  $\phi.I(r_j^k, r, \theta_j^i)$ , celle-ci s'apparente à notre  $h_k(x)$ . Cependant, alors que notre paiement  $h_k(x)$  n'est pas donné si le service web n'est pas accepté, il est, dans l'article [GK09], calculé en remplaçant la notation du maître par la moyenne des notations transmises par les services d'information qui interprétaient le service web comme *mauvais*.

Nous allons donc, dans la section 5.3, adapter les 5 propriétés à la composante  $h_k(x)$  de notre fonction de paiement. Nous ne prendrons cependant pas le parti de verser  $h_k(x)$  même lorsque le service web n'est pas accepté, en remplaçant la notation du maître par une moyenne, car cela va à l'encontre du principe du paiement  $h_k(x)$ . De plus, nous avons déjà le paiement  $g_k(x)$  qui utilise la moyenne des évaluations transmises, et qui n'existe pas dans le modèle de l'article [GK09].

#### 5.2.3 Les 5 propriétés

Nous considérons désormais que  $x$  représente la notation transmise par le service d'information  $k$  à payer.  $r$  est la notation du maître.  $\theta$  est l'interprétation du maître.

##### Première propriété

Sur le domaine  $[0, 1] \times [0, 1] \times \{bon, mauvais\}$ ,  $I$  est continue.

##### Deuxième propriété

Pour tout  $r \in [0, 1]$  et  $\theta \in \{bon, mauvais\}$ , si  $x \in [0, r]$ , alors  $I(x, r, \theta)$  est strictement croissante, et si  $x \in [r, 1]$ , alors  $I(x, r, \theta)$  est strictement décroissante.

La signification est la suivante : plus la notation transmise par le service d'information est proche de celle donnée par le maître de communauté, plus le service d'information est récompensé.

Ces deux premières propriétés étaient déjà respectées par le paiement  $h_k(x)$  du modèle présenté au chapitre 4.

##### Troisième propriété

Pour tout  $r \in [0, 1]$  et  $\epsilon \in (0, \min[r, 1 - r])$ , on définit :

$$\Delta(r, \epsilon, \theta) = I(r + \epsilon, r, \theta) - I(r - \epsilon, r, \theta)$$

$\Delta$  est la différence de paiement reçu entre un service d'information ayant surévalué le service web, et un service d'information l'ayant sous-évalué. La troisième propriété exige que  $\Delta(r, \epsilon, bon) > 0$ , et  $\Delta(r, \epsilon, mauvais) < 0$ .

## 5.2 5 propriétés pour rendre une fonction de paiement juste

---

La signification est la suivante : le maître de communauté récompensera plus les services d'information lui ayant transmis une notation le menant dans le bon sens, que ceux lui ayant transmis une notation le menant dans le mauvais sens. Si le maître considère le service web comme *bon*, il récompensera plus les services d'information ayant surévalué le service web que ceux l'ayant sous-évalué. Par contre, si le maître considère le service web comme *mauvais*, il récompensera plus les services d'information l'ayant sous-évalué que ceux l'ayant surévalué. Par exemple, si un service web est considéré comme *bon* par le maître, un service d'information l'ayant surévalué de 0,05 aura un meilleur paiement qu'un service d'information l'ayant sous-évalué de 0,05.

Dans le modèle présenté au chapitre 4, cette propriété n'est pas respectée. Si un service d'information transmet la notation 0,7, qu'un autre transmet la notation 0,9, et que le maître évalue le service web à 0,8, les deux services d'information auront le même paiement. Une fonction respectant la troisième propriété devrait mieux récompenser le service d'information ayant transmis 0,9.

### Quatrième propriété

Pour tout  $r$  :  $\Delta(r, \epsilon, \textit{bon})$  est strictement croissante sur  $\epsilon$ , et  $\Delta(r, \epsilon, \textit{mauvais})$  est strictement décroissante sur  $\epsilon$ .

La signification est la suivante :

- si  $\theta = \textit{bon}$  : plus  $x$  est loin de  $r$ , plus la sous-évaluation est grave ;
- si  $\theta = \textit{mauvais}$  : plus  $x$  est loin de  $r$ , plus la surévaluation est grave

### Cinquième propriété

Pour tout  $\epsilon \in (0, \min[r, 1 - r])$  et  $\theta \in \{\textit{bon}, \textit{mauvais}\}$ ,  $\Delta(r, \epsilon, \theta)$  est décroissante sur  $r$ .

La cinquième propriété dit qu'une déviation donnée  $\epsilon$  a une importance différente pour différentes valeurs de  $r$ . Plus  $r$  s'approche de 1, moins la sous-évaluation est grave. La différence de paiement entre une sous-évaluation et une surévaluation de même grandeur quand  $r = 0,9$  est plus petite que quand  $r = 0,6$ .

EXEMPLE 5.1. Lorsque  $\theta = \textit{bon}$  il est moins grave de surévaluer que de sous-évaluer. Par conséquent,  $I(r + \epsilon, r, \textit{bon}) > I(r - \epsilon, r, \textit{bon})$  et  $\Delta(r, \epsilon, \textit{bon}) > 0$ . La propriété 5 nous dit que  $\Delta(0,9, \epsilon, \textit{bon}) < \Delta(0,6, \epsilon, \textit{bon})$ . La différence absolue de paiement entre la surévaluation et la sous-évaluation est plus petite lorsque  $r$  s'approche de 1.

De même, plus  $r$  s'approche de 0, moins la surévaluation est grave.

EXEMPLE 5.2. Lorsque  $\theta = \textit{mauvais}$  il est plus grave de surévaluer que de sous-évaluer. Par conséquent,  $I(r + \epsilon, r, \textit{bon}) < I(r - \epsilon, r, \textit{bon})$  et  $\Delta(r, \epsilon, \textit{bon}) < 0$ . La propriété 5 nous dit que  $\Delta(0,4, \epsilon, \textit{mauvais}) < \Delta(0,1, \epsilon, \textit{mauvais})$ . La différence absolue de paiement entre la surévaluation et la sous-évaluation est plus petite lorsque  $r$  s'approche de 0.

L'interprétation  $\theta$  dépend de  $r$  :  $\theta = \textit{bon}$  lorsque  $r \geq z$  et  $\theta = \textit{mauvais}$  lorsque  $r < z$ . Typiquement,  $z = 0,5$ , sous-évaluer un service web noté  $0,6$  est donc plus grave que sous-évaluer un service web noté  $0,9$ , car  $0,6$  est plus proche de la note limite déterminant l'interprétation. De même, il est moins grave de surévaluer un service web si  $r = 0,1$  que si  $r = 0,4$ .

Ces trois dernières propriétés ajoutent une certaine équité dans le système. Nous allons, dans la section qui suit, adapter le paiement  $h_k(x)$  afin qu'il les respecte.

### 5.3 Élaboration de la nouvelle fonction de paiement

Nous allons maintenant développer la nouvelle fonction  $h_k(x)$ , en essayant qu'elle respecte les 5 propriétés.

#### 5.3.1 La nouvelle fonction $h_k(x)$

$h_k(x)$  était définie comme :

$$h_k(x) = \log_b(|Rr_I x_j^S - Ro_M i_j^S|)$$

avec  $0 < b < 1$ . Nous définissons  $x = Rr_I x_j^S$  et  $r = Ro_M i_j^S$ .  $x$  est l'évaluation du service d'information et  $r$  est celle du maître de communauté. Nous avons :  $h_k(x) = (\log_b|x - r|)$ .

Tout d'abord, nous allons remplacer le logarithme par une fraction dans laquelle le numérateur est la valeur maximale que le maître de communauté est prêt à donner au service d'information comme paiement. Cela permet de pouvoir paramétrer facilement la fonction en fixant la valeur maximale de  $h_k(x)$ , celle atteinte uniquement lorsque  $x = r$ . La fonction est ensuite modifiée de la façon suivante :

$$h_k(x) = \begin{cases} \frac{a}{|x-r-(\frac{x-r}{c})|+1} & \text{lorsque } x \geq 0,5(\theta = \textit{bon}) \\ \frac{a}{|x-r+(\frac{x-r}{c})|+1} & \text{lorsque } x < 0,5(\theta = \textit{mauvais}) \end{cases}$$

avec  $a > 0$ ,  $c \in ]0, 1[$ ,  $r \in [0, 1]$ . La fonction est définie sur le domaine  $[0, 1]$ .  $x$  représente ici la notation du service web par le service d'information  $k$ .

L'idée de cette amélioration est qu'un service d'information devrait être récompensé de façon plus importante s'il influence le maître de la communauté à prendre la bonne décision concernant l'entrée du service web en donnant :

- une évaluation plus grande ou égale à  $r$  dans le cas d'un bon service web ;
- une évaluation plus petite ou égale à  $r$  dans le cas d'un mauvais service web.

Il devrait en tout cas être mieux rémunéré qu'un service d'information ayant fait le contraire.

### 5.3 Élaboration de la nouvelle fonction de paiement

---

EXEMPLE 5.3. Pour un certain service web, le service d'information  $SI_1$  donne comme évaluation 0,85, et  $SI_2$  donne comme évaluation 0,75. L'évaluation du maître de communauté est :  $r = 0,80$  et  $\theta = bon$ . Dans ce cas,  $SI_1$  sera mieux récompensé que son collègue  $SI_2$  car il aura influencé positivement le maître de communauté dans sa décision.

#### Fonctionnement de la fonction lorsque le service web est considéré comme bon par le maître de communauté

Si  $x > r$  (bonne direction), alors  $x - r > 0$ . Si, par contre,  $x < r$  (mauvaise direction), alors  $x - r < 0$ .

$|\frac{x-r}{c}| > |x-r|$  puisque  $0 < c < 1$ , donc  $x-r - (|\frac{x-r}{c}|) < 0$ , et ce, que  $x-r$  soit positif (bonne direction), ou négatif (mauvaise direction).

Cependant,  $x-r - (|\frac{x-r}{c}|)$  est plus petit lorsque  $x-r < 0$  (mauvaise direction) que lorsque  $x-r > 0$  (bonne direction).

Nous pouvons conclure que  $|x-r - (|\frac{x-r}{c}|) + 1$  est plus petit si  $x$  est dans la bonne direction et donc que  $h_k(x)$  est plus grand si  $x$  est dans la bonne direction.

#### Fonctionnement de la fonction lorsque le service web est considéré comme mauvais par le maître de communauté

Le fonctionnement est similaire. Si  $x > r$  (mauvaise direction), alors  $x - r > 0$ . Si, par contre,  $x < r$  (bonne direction), alors  $x - r < 0$ .

$|\frac{x-r}{c}| > |x-r|$  puisque  $0 < c < 1$ , donc  $x-r + (|\frac{x-r}{c}|) > 0$ , et ce, que  $x-r$  soit positif (mauvaise direction), ou négatif (bonne direction).

Cependant,  $x-r + (|\frac{x-r}{c}|)$  est plus petit lorsque  $x-r < 0$  (bonne direction) que lorsque  $x-r > 0$  (mauvaise direction).

Nous pouvons conclure que  $|x-r + (|\frac{x-r}{c}|) + 1$  est plus petit si  $x$  est dans la bonne direction et donc que  $h_k(x)$  est plus grand si  $x$  est dans la bonne direction.

#### La notation du maître et celle du service d'information sont identiques

Lorsque  $x = r$  :

$$\frac{a}{|x-r - (|\frac{x-r}{c}|) + 1} = \frac{a}{|0 - (|\frac{0}{c}|) + 1} = a$$

lorsque le service web est considéré comme bon par le maître, et

$$\frac{a}{|x-r + (|\frac{x-r}{c}|) + 1} = \frac{a}{|0 + (|\frac{0}{c}|) + 1} = a$$

lorsque le service web est considéré comme mauvais par le maître.

Le paiement maximum est donc obtenu par le service d'information lorsque  $x = r$ .

Le logarithme décroissant est remplacé par la fraction avec  $a$  comme numérateur.  $a$  est le paiement maximal que le maître de communauté est prêt à donner à un service d'information qui a évalué le service web exactement comme le maître de communauté, c'est à dire lorsque  $x = r$ . Nous pensons qu'il est plus simple pour le maître de communauté de décider quel paiement maximum il est prêt à donner plutôt que de trouver un paramètre pour le logarithme.

Les propriétés sont vérifiées avec cette nouvelle forme de la fonction de paiement, alors que la forme logarithmique donnait des résultats moins satisfaisants et moins facilement analysables lors de la vérification des propriétés. Nous allons maintenant vérifier cette adéquation entre la nouvelle fonction de paiement et les 5 propriétés à respecter.

### Permettre $h_k(x)$ de prendre des valeurs négatives

La nouvelle fonction  $h_k(x)$  que nous avons élaborée ne peut pas être négative. Afin de permettre au maître de communauté de punir un service d'information ayant menti, nous pouvons juste soustraire une valeur constante à la fin de la fonction. Cette fonction devient alors :

$$h_k(x) = \begin{cases} \frac{a}{|x-r-(\frac{x-r}{c})|+1} - b & \text{lorsque } x \geq 0, 5(\theta = \textit{bon}) \\ \frac{a}{|x-r+(\frac{x-r}{c})|+1} - b & \text{lorsque } x < 0, 5(\theta = \textit{mauvais}) \end{cases}$$

Par exemple, nous pourrions fixer la valeur :  $b = \frac{a}{2}$ .

## 5.4 Vérification des propriétés

Les calculs de dérivées ont été réalisés par le service en ligne Wims ([dSAN]).

**Propriété 1 :**  $h_k(x)$  doit être continu sur l'intervalle  $[0, 1] \times [0, 1] \times \{\textit{bon}, \textit{mauvais}\}$

La nouvelle fonction  $h_k(x)$  respecte cette propriété.

**Propriété 2 :**  $\forall r \in [0, 1]$  et  $\forall \theta \in \{\textit{bon}, \textit{mauvais}\}$ , si nous avons  $x \in [0, r]$ , alors  $I$  est strictement croissant, et si  $x \in [r, 1]$ , alors  $I$  est strictement décroissant

Dans le cas où  $\theta = \textit{bon}$ , pour que  $h_k(x)$  soit strictement croissant, la fonction

$$h_{denom}(x) = |x - r - (\frac{x-r}{c})| + 1$$

doit être strictement décroissante, et vice versa. En prenant la dérivée de la fonction, nous obtenons :

$$h'_{denom}(x) = \frac{(\frac{x-r}{|c|x-r} - 1)(\frac{|x-r|}{|c|} - x + r)}{|\frac{|x-r|}{|c|} - x + r|}$$

$\frac{|x-r|}{|c|} - x + r$  est toujours positif car :

$$\frac{|x-r|}{|c|} - x + r > 0 \Leftrightarrow \frac{|x-r|}{|c|} > x - r$$

## 5.4 Vérification des propriétés

---

On voit facilement que la dernière inégalité est toujours vérifiée puisque  $c \in ]0, 1[$ . On peut donc simplifier la dérivée :

$$h'_{denom}(x) = \frac{x-r}{|c||x-r|} - 1$$

Lorsque  $x \in [0, r[$  :

$$\frac{x-r}{|c||x-r|} - 1 = \frac{-1}{|c|} - 1$$

$h'_{denom} < 0$  et  $h_{denom}$  est strictement décroissant sur  $[0, r]$ .

Lorsque  $x \in ]r, 1]$  :

$$\frac{x-r}{|c||x-r|} - 1 = \frac{1}{|c|} - 1$$

$h'_{denom} > 0$  puisque  $c \in ]0, 1[$ , et  $h_{denom}$  est strictement croissant sur  $[r, 1]$  En conclusion,  $h_k(x)$  est strictement croissant sur  $[0, r]$ , et strictement décroissant sur  $[r, 1]$  lorsque  $\theta = bon$ .

Le cas où  $\theta = mauvais$  est vérifié de façon similaire. En prenant la dérivée de la fonction, nous obtenons :

$$h'_{denom}(x) = \frac{\left(\frac{x-r}{|c||x-r|} + 1\right)\left(\frac{|x-r|}{|c|} + x-r\right)}{\left|\frac{|x-r|}{|c|} + x-r\right|}$$

$\frac{|x-r|}{|c|} - x + r$  est toujours positif car :

$$\frac{|x-r|}{|c|} + x-r > 0 \Leftrightarrow \frac{|x-r|}{|c|} > r-x$$

On voit facilement que la dernière inégalité est toujours vérifiée puisque  $c \in ]0, 1[$ . On peut donc simplifier la dérivée :

$$h'_{denom}(x) = \frac{x-r}{|c||x-r|} + 1$$

Lorsque  $x \in [0, r[$  :

$$\frac{x-r}{|c||x-r|} + 1 = \frac{-1}{|c|} + 1$$

$h'_{denom} < 0$  puisque  $c \in ]0, 1[$ , et  $h_{denom}$  est strictement décroissant sur  $[0, r]$ .

Lorsque  $x \in ]r, 1]$  :

$$\frac{x-r}{|c||x-r|} + 1 = \frac{1}{|c|} + 1$$

$h'_{denom} > 0$  puisque  $c \in ]0, 1[$ , et  $h_{denom}$  est strictement croissant sur  $[r, 1]$  En conclusion,  $h_k(x)$  est strictement croissant sur  $[0, r]$ , et strictement décroissant sur  $[r, 1]$  lorsque  $\theta = mauvais$ .

**Propriété 3 :**  $\forall r$  et  $\forall \epsilon \in ]0, \min(r, 1 - r)]$ ,  $\Delta(r, \epsilon, \text{bon}) > 0$  et  $\Delta(r, \epsilon, \text{mauvais}) < 0$

Pour une valeur fixée de  $r$ , nous obtenons :

$$\Delta(\epsilon, \text{bon}) = \frac{a}{|\epsilon - (|\frac{\epsilon}{c}|)| + 1} - \frac{a}{|-\epsilon - (|\frac{-\epsilon}{c}|)| + 1}$$

$\epsilon > 0$  implique que :  $|\epsilon - (|\frac{\epsilon}{c}|)| < |-\epsilon - (|\frac{-\epsilon}{c}|)|$ , et donc  $\Delta(\epsilon, \text{bon}) > 0$

Pour une valeur fixée de  $r$ , nous obtenons :

$$\Delta(\epsilon, \text{mauvais}) = \frac{a}{|\epsilon + (|\frac{\epsilon}{c}|)| + 1} - \frac{a}{|-\epsilon + (|\frac{-\epsilon}{c}|)| + 1}$$

$\epsilon > 0$  implique que :  $|\epsilon + (|\frac{\epsilon}{c}|)| > |-\epsilon + (|\frac{-\epsilon}{c}|)|$ , et donc  $\Delta(\epsilon, \text{mauvais}) < 0$ .

**Propriété 4 :**  $\Delta(r, \epsilon, \text{bon})$  est strictement croissant sur l'intervalle  $\epsilon$  et  $\Delta(r, \epsilon, \text{mauvais})$  est strictement décroissant sur  $\epsilon$

Nous allons vérifier cette propriété dans la section suivante en vérifiant que les graphes de la fonction  $\Delta$  ont bien la forme attendue.

**Propriété 5 :**  $\Delta(r, \epsilon, \theta)$  est décroissant sur l'intervalle  $r$

Dans le cas où  $c$  est une constante, nous avons que pour  $\Delta(r, \epsilon, \theta)$ , la dérivée par rapport à  $r$  est toujours égale à 0.  $\Delta(r, \epsilon, \theta)$  est donc une constante sur  $r$ , ce qui respecte bien la propriété (car une fonction constante est - non strictement - décroissante). Dans le cas où  $c$  est défini en fonction de  $r$ , nous vérifierons la propriété avec des exemples dans la section 5.5.

## 5.5 Graphiques

Dans cette section, nous allons illustrer la nouvelle fonction de paiement et le respect des propriétés avec des graphiques.

### 5.5.1 Forme de la fonction $h_k(x)$

Commençons par regarder la forme de la nouvelle fonction  $h_k(x)$ . Pour les exemples suivants, nous fixons  $a = 400$ ,  $b = 0$ . Nous considérons que  $\theta = \text{bon}$  lorsque  $r \geq 0,5$  et que  $\theta = \text{mauvais}$  sinon. Observons les graphes lorsque  $c = 0,5$ ,  $c = 0,1$ , et  $c = 0,9$ .

**c=0,5** Le graphe de la fonction est le suivant :

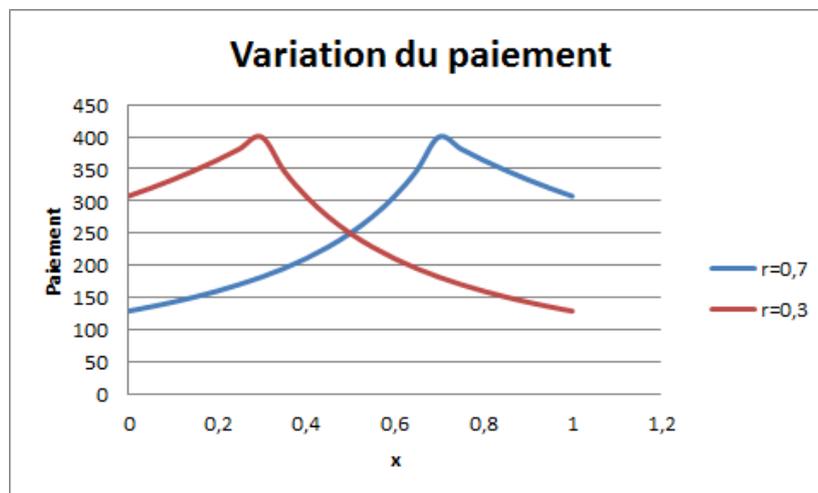


FIGURE 5.1 – Variation du paiement avec  $a = 400$ ,  $b = 0$ , et  $c=0,5$

**c=0,1** Le graphe de la fonction est le suivant :

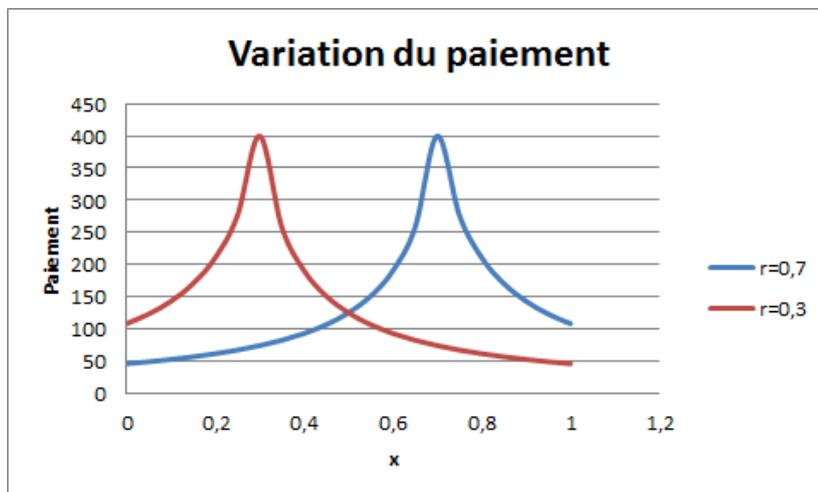


FIGURE 5.2 – Variation du paiement avec  $a = 400$ ,  $b = 0$ , et  $c=0,1$

$c=0,9$  Le graphe de la fonction est le suivant :

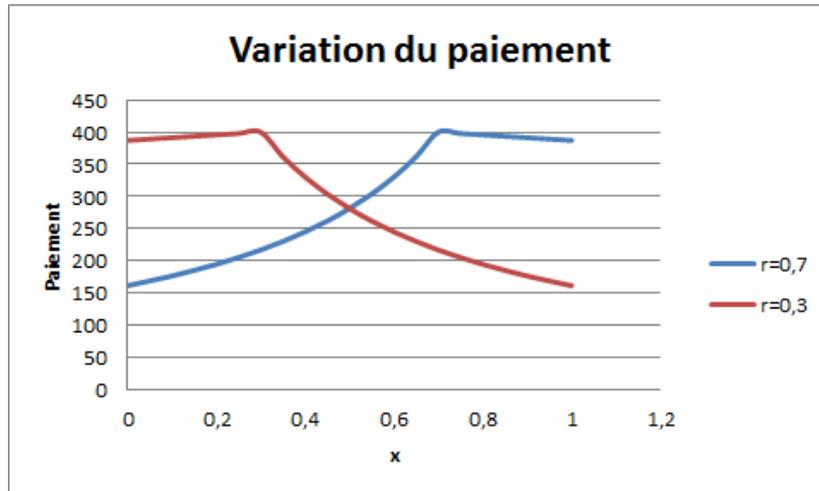


FIGURE 5.3 – Variation du paiement avec  $a = 400$ ,  $b = 0$ , et  $c=0,9$

### Discussion

Nous observons que plus  $c$  est proche de 1, plus la fonction décroît vite. Cependant, la différence de paiement selon la direction lorsque  $c$  est proche de 1 est minime.

Sur le graphe  $r = 0,7$  de la figure 5.1, le paiement est de 250 lorsque  $x = 0,5$ , et le paiement est de 333,33 lorsque  $x = 0,9$ , soit une différence de 83,33 entre le paiement pour le service d'information ayant surévalué (sa notation du service web) de 0,2 et celui ayant sous-évalué de 0,2.

Sur le graphe  $r = 0,7$  de la figure 5.2, le paiement est de 125 lorsque  $x = 0,5$ , et le paiement est de 142,86 lorsque  $x = 0,9$ , soit une différence de 17,86 entre le paiement pour le service d'information ayant surévalué de 0,2 et celui ayant sous-évalué de 0,2.

Enfin, sur le graphe  $r = 0,7$  de la figure 5.3, le paiement est de 281,25 lorsque  $x = 0,5$ , et le paiement est de 391,3 lorsque  $x = 0,9$ , soit une différence de 110,05 entre le paiement pour le service d'information ayant surévalué de 0,2 et celui ayant sous-évalué de 0,2.

### Faire varier $c$ en fonction de $r$

Une autre idée serait de ne pas fixer  $c$ , mais de le faire varier en fonction de  $r$ . Par exemple, on pourrait poser  $c = 1 - r$  quand  $\theta = \text{bon}$ , et  $c = r$  quand  $\theta = \text{mauvais}$ . Dans le cas où le service web est considéré comme bon par le maître de la communauté, avec  $r = 0.7$ , la formule utilisée est :

$$\frac{400}{|x - 0,7 - |\frac{(x-0,7)}{1-0,7}|| + 1}$$

## 5.5 Graphiques

Dans le cas où le service web est considéré comme mauvais par le maître de communauté, avec  $r = 0.3$ , la formule utilisée est :

$$\frac{400}{|x - 0,3 + |\frac{x-0,3}{0,3}|| + 1}$$

Les graphes sont montrés sur la figure 5.4.

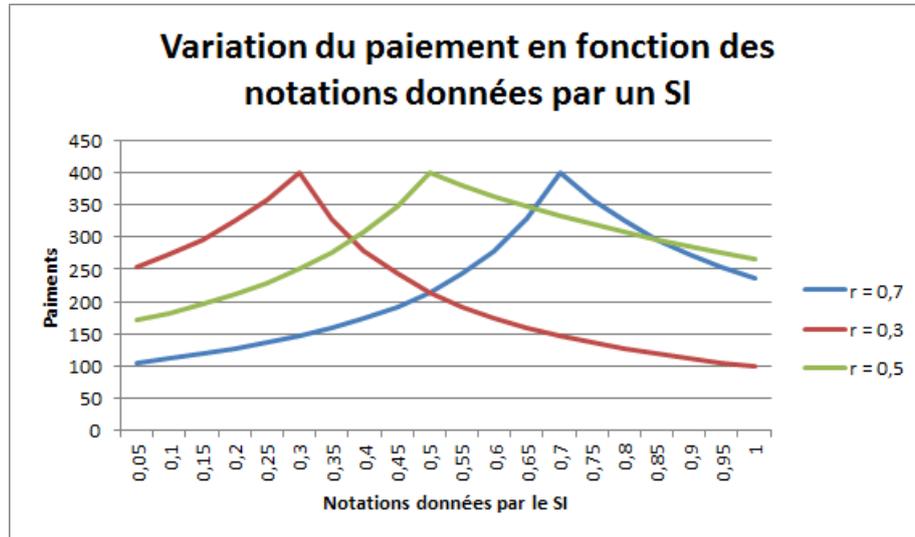


FIGURE 5.4 – Variation du paiement avec  $a = 400$

Poser  $c = 1 - r$  quand  $\theta = bon$ , et  $c = r$  quand  $\theta = mauvais$ , signifie que plus  $r$  est proche de 1 ou de 0, plus  $c$  est petit, et donc, plus la fonction décroît. Cela signifie qu'un service d'information ayant transmis  $x = 0,7$  alors que  $r = 0,9$ , ou  $x = 0,3$  alors que  $r = 0,1$ , serait moins bien payé qu'un service d'information ayant transmis  $x = 0,5$  alors que  $r = 0,7$ , ou  $x = 0,5$  alors que  $r = 0,3$ . Plus un service web est exceptionnellement bon ou exceptionnellement mauvais, moins le maître tolère un écart de notation.

Par contre, poser  $c = r$  quand  $\theta = bon$ , et  $c = 1 - r$  quand  $\theta = mauvais$ , signifie que plus  $r$  est proche de 1 ou de 0, plus  $c$  est grand, et donc, moins la fonction décroît. Cela signifie qu'un service d'information ayant transmis  $x = 0,7$  alors que  $r = 0,9$ , ou  $x = 0,3$  alors que  $r = 0,1$ , serait mieux payé qu'un service d'information ayant transmis  $x = 0,5$  alors que  $r = 0,7$ , ou  $x = 0,5$  alors que  $r = 0,3$ . Plus un service web est exceptionnellement bon ou exceptionnellement mauvais, plus le maître tolère un écart de notation.

Attention cependant, lorsqu'on pose  $c = 1 - r$  quand  $\theta = bon$ , et  $c = r$  quand  $\theta = mauvais$ , ou vice-versa, la fonction  $h_k(x)$  n'est plus définie que sur l'intervalle  $]0, 1[$ , et non plus  $[0, 1]$

### Vérification de la propriété 2

Sur les figures 5.1, 5.2, 5.3, et 5.4, on peut voir que dans le cas où le service web a une bonne qualité de service, les services d'information qui surévaluent le service web

entrant sont mieux récompensés que ceux qui le sous-évaluent. Et dans l'autre cas, les services d'information qui surévaluent le service web sont moins récompensés que ceux qui le sous-évaluent. Cela correspond bien à la deuxième propriété.

**Vérification des propriétés 3 et 4**

Vérifions les propriétés 3 et 4 dans le cas où le service web est considéré comme bon par le maître de communauté, avec  $r = 0.7$ , et  $c = 0,3$ . Dans ce cas,  $\epsilon$  varie entre 0 et 0,3. La formule utilisée est :

$$\frac{400}{|(0,7 + \epsilon) - 0,7 - (|\frac{(0,7+\epsilon)-0,7}{0,3}|)| + 1} - \frac{400}{|(0,7 - \epsilon) - 0,7 - (|\frac{(0,7-\epsilon)-0,7}{0,3}|)| + 1}$$

La vérification des propriétés 3 et 4 dans le cas où le service web est considéré comme mauvais par le maître de communauté, avec  $r = 0.3$ , et  $c = 0,3$  donne le cas suivant. Dans ce cas,  $\epsilon$  varie entre 0 et 0.3. La formule utilisée est :

$$\frac{400}{|(0,3 + \epsilon) - 0,3 + (|\frac{(0,3+\epsilon)-0,3}{0,3}|)| + 1} - \frac{400}{|(0,3 - \epsilon) - 0,3 + (|\frac{(0,3-\epsilon)-0,3}{0,3}|)| + 1}$$

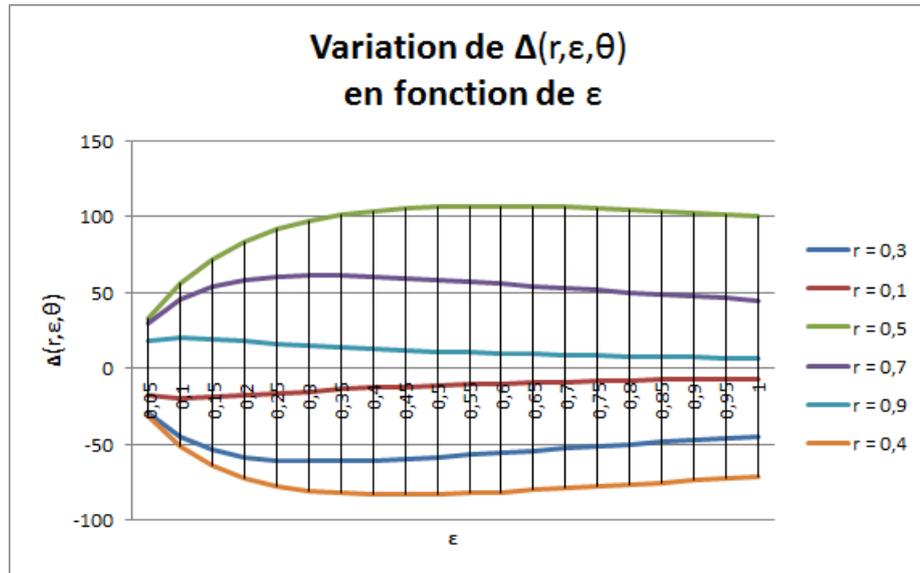


FIGURE 5.5 – Variation de  $\Delta(r, \epsilon, \theta)$  en fonction de  $\epsilon$ .

Nous voyons sur la figure 5.5 que la fonction reste positive comme cela est requis par la propriété 3 et qu'elle est croissante comme le demande la propriété 4 quand  $r \geq 0,5$  ( $\theta = \text{bon}$ ). Nous voyons aussi que la fonction est négative et décroissante comme il est exigé par les propriétés 3 et 4 lorsque  $r < 0,5$  ( $\theta = \text{mauvais}$ ). Attention :  $\epsilon$  peut varier entre  $[0, \min(r, 1 - r)]$ , donc les points des graphiques après cette limite ne sont plus pertinents.

**Vérification de la propriété 5**

En ce qui concerne la cinquième propriété, il est demandé que, selon la valeur de  $r$ , la fonction  $\Delta(r, \epsilon, \theta)$  soit décroissante. Lorsque  $c$  est une constante, la fonction  $\Delta(r, \epsilon, \theta)$  est

## 5.5 Graphiques

constante sur  $r$ . La propriété 5 est donc « non strictement » respectée.

En fixant  $c = 1 - r$  lorsque  $r \geq 0,5$  ( $\theta = \textit{bon}$ ) et  $c = r$  lorsque  $r < 0,5$  ( $\theta = \textit{mauvais}$ ), la fonction  $\Delta(r, \epsilon, \theta)$  est strictement décroissante sur  $r$ . En fixant beaucoup de valeurs  $\epsilon$  différentes, nous voyons que cela est vérifié. La figure 5.6 nous montre des exemples pour plusieurs valeurs  $\epsilon$  lorsque  $\theta = \textit{bon}$ , et la figure 5.7 montre plusieurs exemples lorsque  $\theta = \textit{mauvais}$ . Nous voyons que dans tous ces cas, la fonction est bien décroissante sur  $r$ . La propriété 5 est donc vérifiée.

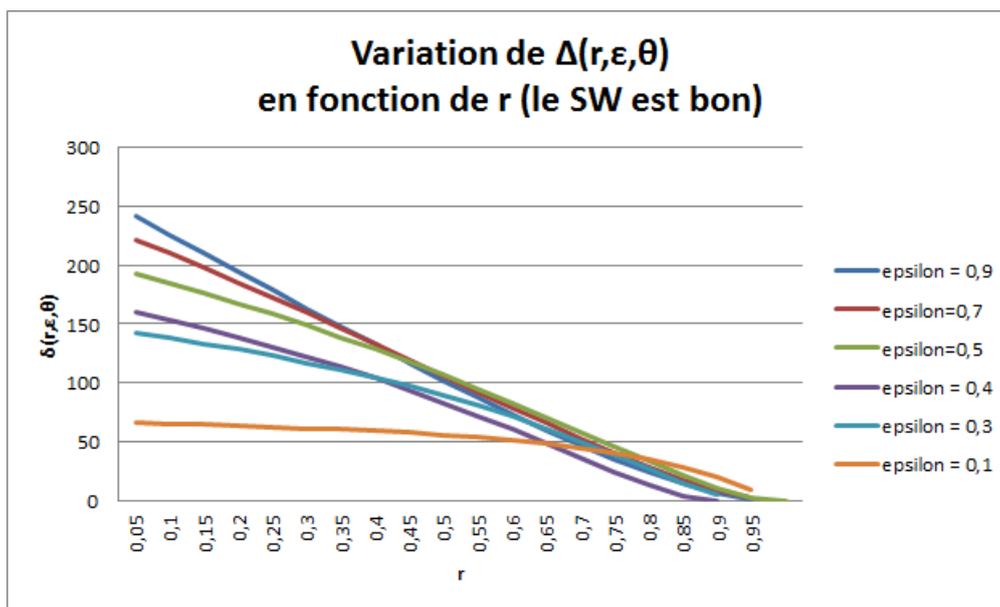


FIGURE 5.6 – Variation de  $\Delta(r, \epsilon, \theta)$  en fonction de  $r$  lorsque le service web est bon et que  $c = 1 - r$ .

En fixant  $c = r$  lorsque  $r \geq 0,5$  ( $\theta = \textit{bon}$ ) et  $c = 1 - r$  lorsque  $r < 0,5$  ( $\theta = \textit{mauvais}$ ), la fonction  $\Delta(r, \epsilon, \theta)$  est strictement croissante sur  $r$ . En fixant beaucoup de valeurs  $\epsilon$  différentes, nous voyons que cela est vérifié. La figure 5.13 nous montre des exemples pour plusieurs valeurs  $\epsilon$  lorsque  $\theta = \textit{bon}$ , et a figure 5.9 montre plusieurs exemples lorsque  $\theta = \textit{mauvais}$ . Nous voyons que dans tous ces cas, la fonction est bien décroissante sur  $r$ . La propriété 5 n'est donc pas vérifiée.

Comme vu dans la sous-sous-section 5.5.1, plus  $c$  est petit, plus la différence de paiement entre la sous-évaluation et la surévaluation est faible. La propriété 5 veut que cette différence s'affaiblisse lorsque  $r$  est proche de 1 ou de 0. Il faut donc que  $c$  soit petit lorsque  $r$  est proche de 1 ou de 0, et plus grand sinon. Poser  $c = 1 - r$  lorsque  $\theta = \textit{bon}$  et  $c = r$  lorsque  $\theta = \textit{mauvais}$  permet donc de respecter strictement cette propriété. Vouloir respecter strictement cette propriété génère un énorme inconvénient. Certes, la différence de paiement entre la surévaluation et la sous-évaluation est plus faible lorsque  $r$  est proche de 1 ou 0, mais la courbe  $h_k(x)$  décroît bien plus rapidement lorsque  $c$  est petit, ce qui fait qu'un service d'information ayant sous-évalué un très bon service web ( $r = 0,9$  par

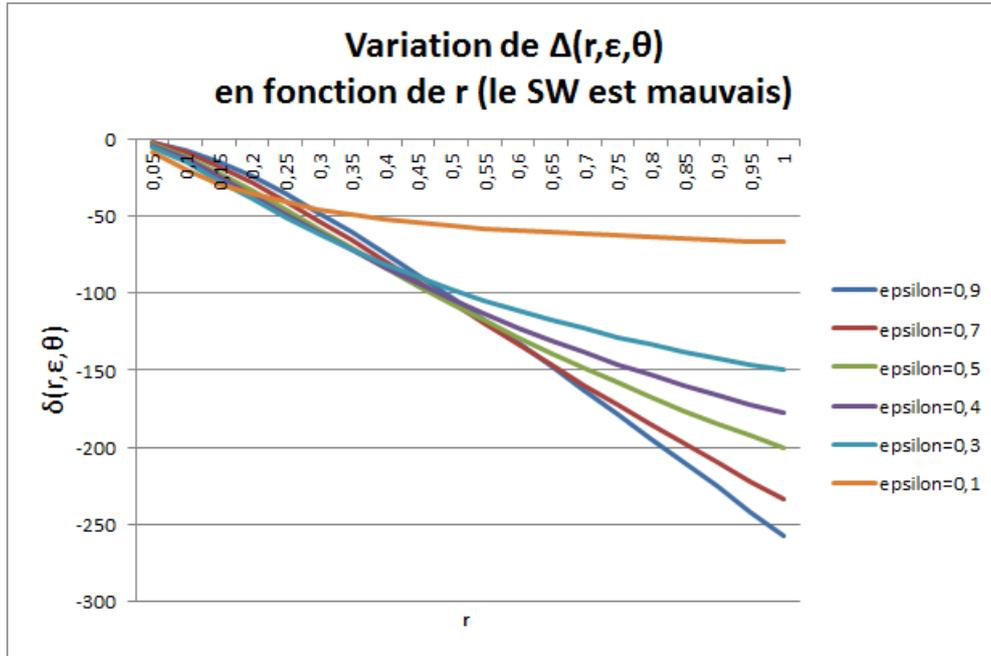


FIGURE 5.7 – Variation de  $\Delta(r, \epsilon, \theta)$  en fonction de  $r$  lorsque le service web est mauvais et que  $c = r$ .

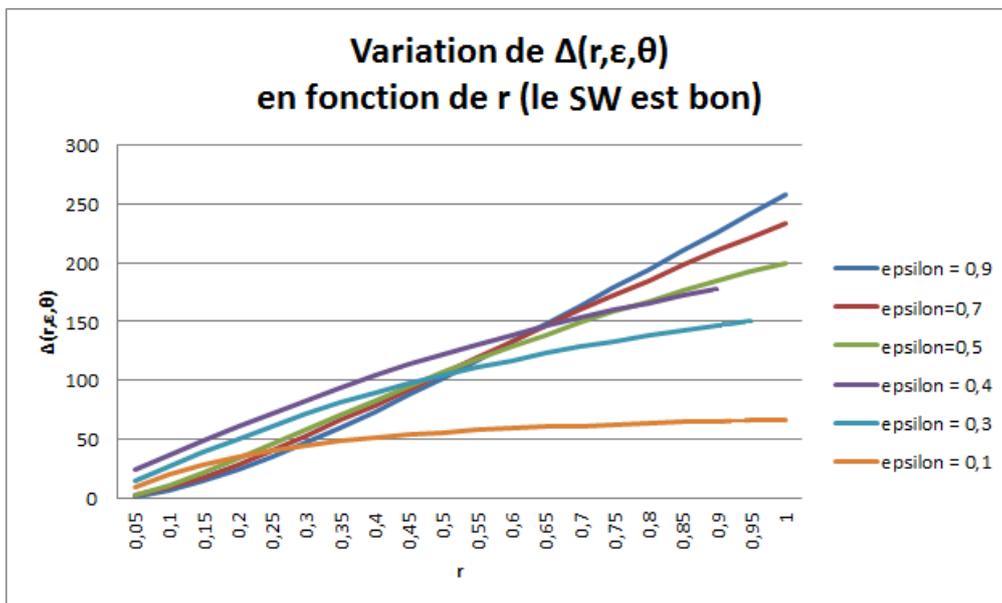


FIGURE 5.8 – Variation de  $\Delta(r, \epsilon, \theta)$  en fonction de  $r$  lorsque le service web est bon et que  $c = r$ .

exemple) gagne moins qu'un service d'information ayant sous-évalué un service web bon mais proche de 0,5 ( $r = 0,6$  par exemple) ! Or la propriété 5 a été créée de manière à faire en sorte que la sous-évaluation soit moins grave lorsque le service web est très bon (car l'interprétation  $\theta$  risque moins de changer).

Nous avons donc décidé de ne pas respecter la propriété 5 et de poser  $c = r$  lorsque

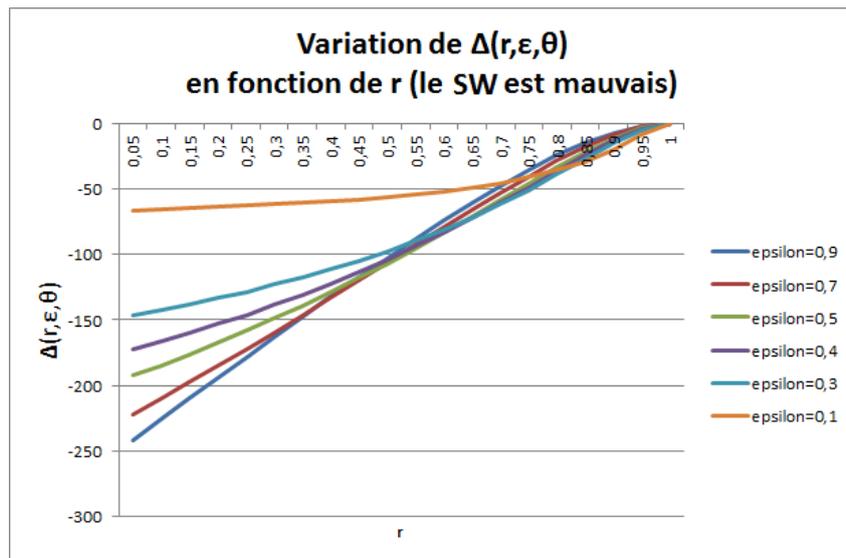


FIGURE 5.9 – Variation de  $\Delta(r, \epsilon, \theta)$  en fonction de  $r$  lorsque le service web est mauvais et que  $c = 1 - r$ .

$r \geq 0,5$  ( $\theta = \text{bon}$ ) et  $c = 1 - r$  lorsque  $r < 0,5$  ( $\theta = \text{mauvais}$ ) pour les simulations dans le chapitre 8. Il est bien sûr possible de respecter cette propriété en posant  $c = 1 - r$  lorsque  $r \geq 0,5$  ( $\theta = \text{bon}$ ) et  $c = r$  lorsque  $r < 0,5$  ( $\theta = \text{mauvais}$ ) ou tout simplement  $c = \text{constante}$ , mais nous choisissons de ne pas le faire. En effet, l'intérêt de respecter les 5 propriétés est de rendre la fonction plus juste mais dans notre cas, respecter la propriété 5 semble justement rendre la fonction moins juste en donnant de moins gros paiements aux services d'information sous-évaluant des services web extrêmement bons. Ce n'est pas la propriété 5 en elle-même qui pose problème, mais notre fonction qui accentue ce côté désavantageux. Nous trouvons plus juste que les surévaluations ou les sous-évaluations soient plus pénalisable lorsque  $r$  est proche de 0,5 que lorsque  $r$  est respectivement très bas ou très élevé.

Voici deux graphiques pour représenter la situation, le premier est celui concernant la fonction qui respecte la propriété 5, et le second non.

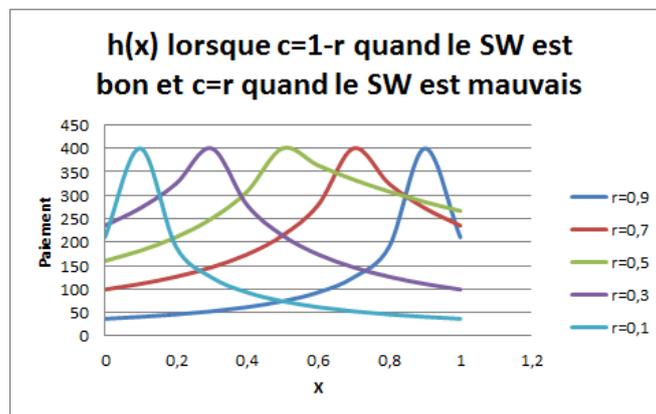


FIGURE 5.10 – Respect de la propriété 5

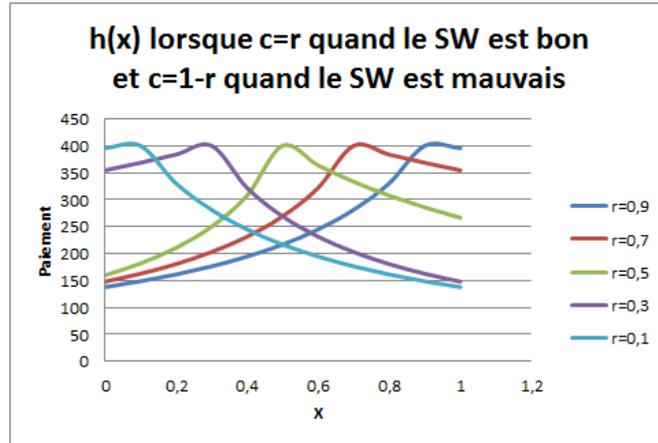


FIGURE 5.11 – Non respect de la propriété 5

Sur le premier, plus  $r$  est proche de 0 ou de 1, plus la différence d'évaluation est grave (la forme de « montagne » du graphe est accentuée), et dans le second c'est l'inverse, c'est au milieu (la charnière) que cette différence d'évaluation est grave.

### 5.5.2 Permettre une valeur $h_k(x)$ négative : $\gamma^-$

Avant de terminer cette section, regardons ce que nous obtenons lorsque nous permettons une valeur  $h_k(x)$  négative. La figure 5.12 représente les mêmes fonctions que la figure 5.4, mais lorsque  $b = 200$ .

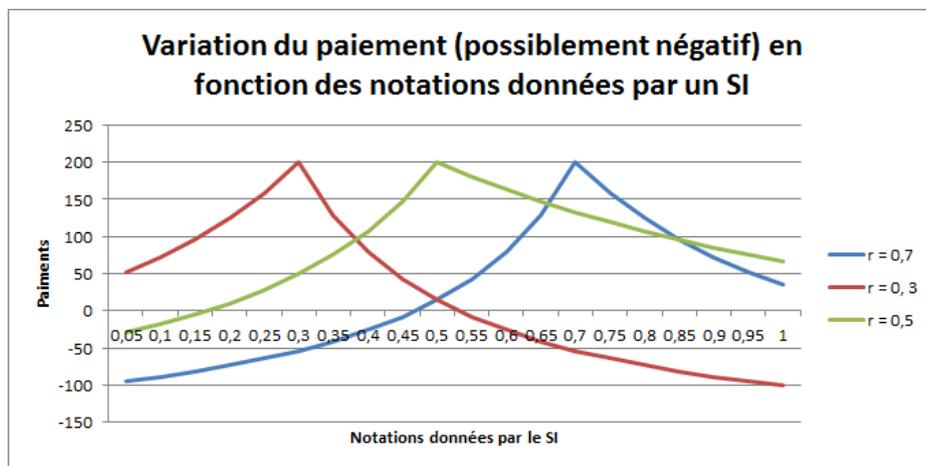


FIGURE 5.12 – Permettre une valeur  $h_k(x)$  négative :  $\gamma^-$

## 5.6 La fonction $g_k(x)$

Une question reste toutefois concernant la fonction  $g_k(x)$ . En effet, le paiement  $h_k(x)$  n'est parfois pas donné, et dans ce cas, seules les fonctions  $f_k(x)$  et  $g_k(x)$  sont données en paiement. Cette fonction de paiement résultante ( $f_k(x) + g_k(x)$ ) ne respecte pas les propriétés de l'article [GK09].

Nous pourrions retirer la fonction  $g_k(x)$  de la fonction de paiement, et donner le  $h_k(x)$  dans tous les cas. Sa valeur serait alors calculée en accord avec le principe donné dans [GK09] : nous fixerions l'évaluation du maître de communauté  $r$  comme la moyenne de toutes les évaluations des services d'information qui ont donné comme type d'information  $\theta = mauvais$ . Nous ne poursuivrons donc pas cette voie, car cela reviendrait à simplement copier la fonction de paiement de l'article [GK09]. Cependant, nous proposons de remplacer le logarithme de  $g_k(x)$  par une fraction comme pour  $h_k(x)$ .  $g_k(x)$  devient donc :

$$g_k(x) = \frac{d}{l \left| \frac{\sum_{j=0}^n Rep_M^{I_j} \cdot Not_{I_j}}{\sum_{j=0}^n Rep_M^{I_j}} - x \right| + 1} - e$$

où  $d - e$  est le paiement maximum (comme  $a - b$  pour  $h_k(x)$ ).  $e$  est une constante de retrait permettant à  $g_k(x)$  de prendre des valeurs négatives (comme  $b$  pour  $h_k(x)$ ).  $Rep_M^{I_j}$  est la réputation du service d'information  $j$ ,  $x$  est l'évaluation du service web donnée par le service d'information  $k$  (pour lequel le paiement est calculé), et  $Not_{I_j}$  est l'évaluation du service web donnée par le service d'information  $j$ . Quant à  $l$ , il s'agit d'une constante supérieure ou égale à 1, tel que plus  $l$  est grand, plus la fonction décroît rapidement. Pour la fonction  $g_k(x)$ , aucune propriété intéressante ne semble pouvoir s'y adapter. Nous utilisons donc cette nouvelle forme, mais n'ajoutons pas de propriétés supplémentaires. A nouveau, cette forme nous semble plus facilement paramétrable que l'ancienne fonction logarithmique, car nous pouvons notamment fixer le paramètre  $d$  comme la valeur maximale que le maître de communauté est prêt à donner pour le paiement  $g_k(x)$ . Pour  $d = 200$ ,  $e = 100$ , et  $l = 3$ , la fonction  $g_k(x)$  prend la forme suivante :

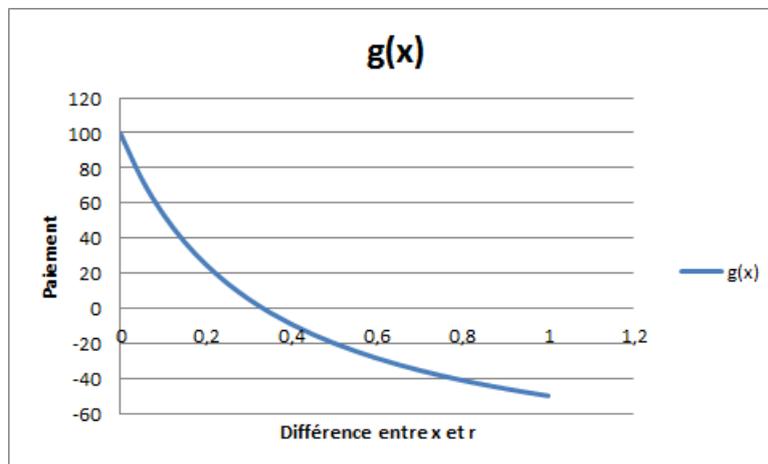


FIGURE 5.13 –  $g_k(x)$  lorsque  $d = 200$ ,  $e = 100$ , et  $k = 3$ .

## 5.7 Récapitulation du chapitre et remarques sur les notations

### 5.7.1 Récapitulation

Nous venons de voir les changements apportés à notre fonction de paiement, selon les 5 propriétés (section 5.2). Afin de vérifier ces 5 propriétés, les différentes fonctions de paiement (composant la fonction de paiement générale) ont été modifiées (à l'exception de  $f_k(x)$  qui reste une valeur constante).

Le paiement  $g_k(x)$  vaut maintenant :

$$g_k(x) = \frac{d}{l \left| \frac{\sum_{j=0}^n \text{Rep}_M^{I_j} \cdot \text{Not}_{I_j}}{\sum_{j=0}^n \text{Rep}_M^{I_j}} - x \right| + 1} - e$$

Le paiement  $h_k(x)$  vaut maintenant :

$$h_k(x) = \begin{cases} \frac{a}{|x-r-(\frac{x-r}{c})|+1} - b & \text{si le service web est considéré comme bon par le maître} \\ \frac{a}{|x-r+(\frac{x-r}{c})|+1} - b & \text{si le service web est considéré comme mauvais par le maître} \end{cases}$$

$x$  est la notation transmise par le service d'information  $k$ .  $x \in [0, 1]$

Pour les simulations du chapitre 8,  $c$  vaudra  $r$  lorsque  $r \geq 0,5$  et  $c$  vaudra  $1-r$  lorsque  $r < 0,5$ , pour la fonction  $h_k(x)$ . Dans notre modèle,  $h_k(x)$  ne respectera donc que les 4 premières propriétés, et pas la cinquième. Nous fixons  $l$  à 3 pour la fonction  $g_k(x)$ . Nous limiterons  $x$  en l'interdisant de prendre les valeurs 1 ou 0 (car  $c$  ne peut prendre ces 2 valeurs). Dans le modèle exact utilisé pour les simulations,  $x \in ]0, 1[$ .

### 5.7.2 Remarques sur les notations

Des précisions sont nécessaires quant aux significations des symboles et notations utilisés dans les tableaux du chapitre 4, et qui seront utilisés dans les chapitres suivants également. Reprenons par exemple le premier tableau exposé :

ASI		
	Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta + \gamma), (\alpha + \beta + \gamma)$
	Mentir	$(\alpha + \beta^-), (\alpha + \beta_-)$
		$(\alpha + \beta^- + \gamma^-), (\alpha + \beta_- + \gamma)$
		$(\alpha + \beta), (\alpha + \beta)$

TABLE 5.1 – Jeu lorsque  $S_j$  a une bonne qualité de service.

Nous avons vu dans la sous-sous-section 4.2.3 que  $\beta^-$  et  $\gamma^-$  désignent des punitions, c'est-à-dire des paiements négatifs si on permet ceux-ci, ou des paiements nuls. Ces significations sont celles de l'ancien travail, et ne sont plus exactes dans le nôtre. Il est donc nécessaire de préciser les changements.

## 5.7 Récapitulation du chapitre et remarques sur les notations

La première question à se poser est comment le maître décide de déclarer un service d'information menteur ou non. Pour cela, il va non pas comparer les notations du maître et du service d'information, mais il va comparer les interprétations, c'est à dire les deux valeurs de  $\theta$ . Rappelons qu'une interprétation est binaire, c'est-à-dire que  $\theta \in \{bon, mauvais\}$ . Pour notre modèle, nous décidons de poser  $\theta = bon$  lorsque  $r \geq 0,5$ , et  $\theta = mauvais$  lorsque  $r < 0,5$ . Le service d'information doit donc juste transmettre sa notation. Cependant, les paiements sont calculés par une fonction qui ne dépend pas du fait que le maître soupçonne un service d'information d'avoir menti. Cela signifie que par exemple, si un service d'information donne une évaluation de 0,51 ( $\theta = bon$ ) et que le maître de communauté évalue le service web à 0,49 ( $\theta = mauvais$ ), le maître considérera le service d'information comme menteur et baissera sa réputation, mais ce dernier recevra quand même un bon paiement  $h_k(x)$  car 0,49 est très proche de 0,51. L'analyse théorique des jeux n'en devient que plus ardue, car un service d'information peut mentir et obtenir un  $\gamma^-$  positif, voir grand.

Pour faciliter l'étude théorique des jeux réalisée dans le chapitre suivant, nous allons cependant considérer qu'en général, un service d'information ayant menti recevra un plus petit paiement  $h_k(x)$ . Autrement dit, un paiement  $\gamma^-$  peut être positif comme négatif mais dans tous les cas, l'inégalité suivante est toujours respectée :

$$\gamma^- < \gamma$$

De même, pour le  $\beta^-$ , on considère que dans tous les cas, l'inégalité suivante est toujours respectée :

$$\beta^- < \beta_- < \beta$$

Les notations de type  $\beta^-$  et  $\gamma^-$  ne sont donc plus utilisées pour représenter des paiements négatifs, mais pour différencier les différentes proportions de paiement selon l'action, et indiquent ce que ces paiements sont censés faire à l'optimum, dans une situation non « limite ».

Ces nouvelles considérations nous obligent à modifier l'analyse du cas où un service web a une mauvaise qualité de service mais n'est pas malhonnête. Ce cas est représenté par le tableau suivant :

ASI			
	Dire la vérité	Mentir	
SI	Dire la vérité	$(\alpha + \beta), (\alpha + \beta)$	$(\alpha + \beta^- + \gamma), (\alpha + \beta_- + \gamma^-)$
	Mentir	$(\alpha + \beta^-), (\alpha + \beta_-)$	$(\alpha + \beta + \gamma^-), (\alpha + \beta + \gamma^-)$

TABLE 5.2 – Jeu lorsque  $S_j$  a une mauvaise qualité de service et est honnête.

Nous avons conclu dans la section 4.2.4 que le profil d'actions (Dire la vérité, Dire la vérité) est le seul équilibre de Nash. Cependant, en considérant que  $\gamma^-$  puisse être positif, le profil d'actions (Mentir, Mentir) peut lui aussi devenir un équilibre de Nash. Nous testerons ce cas avec des simulations dans le chapitre 8.



## Chapitre 6

# Un nouveau cas : la corruption par un tiers

Nous avons vu dans le chapitre 4 qu'un service web avec une mauvaise qualité de service pouvait être malhonnête et tenter de corrompre les services d'information avec un paiement  $\pi$ , afin que chacun augmente la notation qu'il accorde au service web. Lorsqu'un service web a une bonne qualité de service, ce problème ne se pose pas. Nous allons élargir le problème en considérant désormais que lorsqu'un service web avec une bonne qualité de service désire entrer dans une communauté, un concurrent, déjà membre de la communauté ou candidat, pourrait tenter de corrompre les services d'information, afin que ceux-ci baissent la notation qu'ils attribuent chacun au service web.

### 6.1 Modélisation du nouveau cas

Nous modélisons ce nouveau cas avec un jeu stratégique, comme cela a déjà été fait pour les trois autres cas vus dans le chapitre 4.

#### 6.1.1 Jeu stratégique

Le jeu stratégique modélisant cette nouvelle situation est représenté par le tableau suivant :

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta + \gamma), (\alpha + \beta + \gamma)$	$(\alpha + \beta^-), (\alpha + \beta_- + \pi)$
	Mentir	$(\alpha + \beta^- + \gamma^-), (\alpha + \beta_- + \gamma)$	$(\alpha + \beta + \pi), (\alpha + \beta + \pi)$

TABLE 6.1 – Jeu représentant la situation où le service web est bon et un tiers tente de corrompre les services d'information.

Si  $\pi > \gamma - \beta + \beta_-$ , le profil d'actions (Mentir, Mentir) est un équilibre de Nash. Si  $\pi > \gamma$ , le même profil est aussi un optimum de Pareto. Par contre, si  $\gamma > \pi - \beta + \beta_-$ , le profil d'actions (Dire la vérité, Dire la vérité) est un équilibre de Nash, et si  $\gamma > \pi$ , il est optimum de Pareto.

### 6.1.2 Problème spécifique à ce cas

Tout comme dans l'autre cas de corruption présenté dans la sous-section 4.2.4, le comportement des services d'information dépendra des inéquations présentées. Cependant, un problème supplémentaire se pose ici : celui de l'impunité.

Dans l'autre cas de corruption possible, un service web, mauvais et malhonnête, paie les services d'information pour voir les notations qui lui sont données surévaluées. Autrement dit, lorsque les services d'information décident de mentir, le service web est accepté au sein de la communauté (alors qu'il est mauvais), et le maître peut alors l'évaluer lui-même. Les services d'information reçoivent alors un paiement faible voire négatif, que l'on note  $\gamma^-$ . Cependant, un autre incitant à l'honnêteté et qui n'est pas pris en compte dans les jeux stratégiques, est le fait que le maître de communauté soit au courant ou non de la malhonnêteté des services d'information. En effet, un maître de communauté ne perdra pas son temps et son argent à acheter les notations d'un service d'information qui ne lui a que trop souvent menti. Ainsi, un service d'information peut hésiter à mentir, même s'il obtient un meilleur paiement sur le moment. Un service d'information désire agir sur le long terme, il préfère ne pas prendre le risque de cesser ses activités pour obtenir un paiement quelque peu supérieur à un moment donné. Concrètement, comme nous l'avons déjà mentionné dans le chapitre 4, les services d'information auront eux-mêmes une réputation. Si obtenir un paiement supérieur incite à mentir, la menace d'une baisse de réputation incite à dire la vérité.

Le problème dans le cas présent, par rapport à l'autre situation possible de corruption, est que si une majorité de services d'information corrompus mentent en baissant les notations d'un bon service web, celui-ci ne sera pas accepté au sein de la communauté, et le maître ne saura pas que les services d'information ont menti. Il y a donc un sentiment d'impunité qui incite les services d'information à mentir dès que  $\pi > \gamma - \beta + \beta_-$ . Les services d'information corrompus pourraient donc s'allier en s'assurant qu'ils sont une majorité afin d'être sûrs d'être gagnants sur tous les plans (réception d'un bon paiement et pas de baisse de réputation).

Pour corriger ce problème, nous proposons d'enrichir notre modélisation, en permettant à un service web considéré comme mauvais de quand même rentrer dans la communauté avec une certaine probabilité, afin que le maître puisse l'évaluer et vérifier l'honnêteté des services d'information. L'impunité garantie disparaît alors, puisqu'il y a une chance que la malhonnêteté des services d'information soit découverte.

### 6.1.3 Une nouvelle modélisation avec les jeux étendus

Nous allons faire notre nouvelle modélisation en utilisant un concept des jeux étendus : le mouvement aléatoire. Nous décidons de modéliser cette situation non pas par un jeu stratégique mais par un jeu étendu avec mouvements simultanés et mouvements aléatoires. Ce type de jeux est présenté dans la sous-section 3.3.6.

## 6.1 Modélisation du nouveau cas

Trois joueurs participent à ce jeu. Un service d'information typique noté SI, l'ensemble des autres services d'information noté ASI, et le maître de communauté noté MA. Le premier tour est un mouvement simultané où SI et ASI jouent. Il correspond au jeu stratégique présenté ci-dessus (table 6.1). Le second tour correspond au tour du maître. Si le service web est considéré comme bon (après avoir fait la moyenne des notations transmises par les services d'information), le maître accepte le service web au sein de la communauté. Si le service web est considéré comme mauvais, le tour du maître est un mouvement aléatoire avec une probabilité  $P$  que le service web soit accepté, et une probabilité  $1 - P$  qu'il soit refusé. On peut voir cela comme si le maître de communauté commençait son tour par jeter les dés pour voir s'il prendra la décision logique par rapport aux informations qui lui ont été rapportées en refusant le service web, ou la décision inverse.

Le jeu modélisé est représenté par l'arbre ci-dessous. L'action consistant à dire la vérité est notée « DV » et l'action consistant à mentir est notée « ME ».

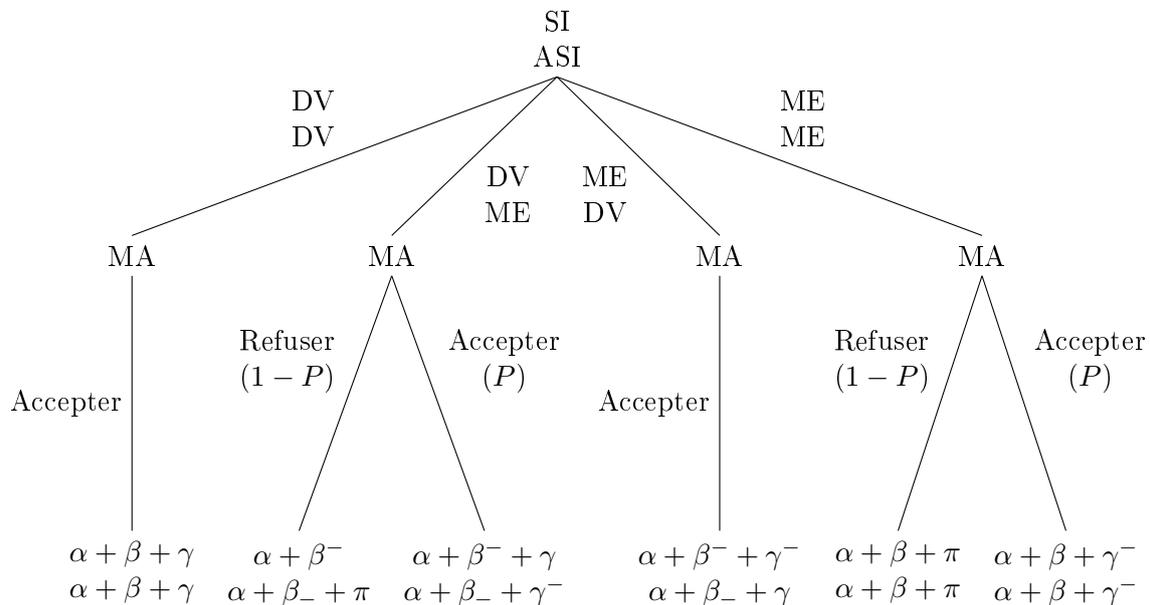


FIGURE 6.1 – Jeu étendu modélisant le cas où un service web est bon et où un tiers tente de corrompre les services d'information. On introduit une probabilité  $P$  que le maître accepte quand même ledit service web s'il a reçu l'information que le service web est mauvais.

Le jeu ci-dessus (figure 6.1) représente le cas où le service web a une bonne qualité de service, et un tiers tente de corrompre les services d'information. Le cas où le service web est bon, et qu'il n'y a pas de tentative de corruption est semblable, il suffit d'enlever la partie  $\pi$  des paiements quand celle-ci est présente.

### 6.1.4 Révision du cas où le service web a une mauvaise qualité de service

Plaçons-nous un moment du point de vue du maître de communauté. Celui-ci doit, lorsqu'il obtient l'information que le service web a une mauvaise qualité de service, « jeter les dés », pour savoir s'il refuse le service web ou s'il l'accepte. Ceci afin de dissuader les services d'information de mentir. Mais nous pouvons très bien nous trouver dans le cas où le

service web a véritablement une mauvaise qualité de service et où les services d'information transmettent la vérité. Le maître ne pouvant évidemment pas faire la différence entre les deux situations, ce dernier cas est lui aussi remodelisé. L'arbre ci-dessous représente le nouveau jeu étendu lorsque le service web a une mauvaise qualité de service. La partie  $\pi$  des paiements où elle est présente dépend de l'honnêteté ou non du service web, car il n'y a que lui qui serait motivé à payer un paiement  $\pi$  puisque dans ce cas-ci il est réellement mauvais.

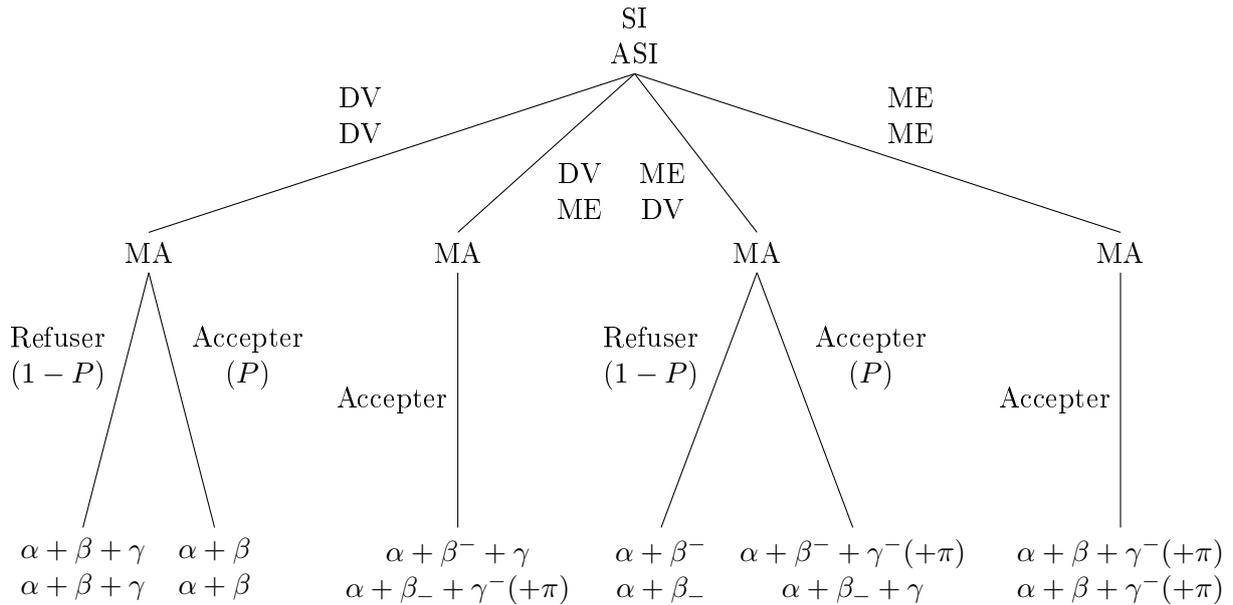


FIGURE 6.2 – Jeu étendu modélisant le cas où un service web est mauvais. On introduit une probabilité  $P$  que le maître accepte quand même le service web s'il a reçu l'information que le service web est mauvais.

### 6.1.5 Intérêt de la nouvelle modélisation

L'intérêt d'introduire cette probabilité d'accepter un service web dont on a reçu l'information qu'il est mauvais, vise à réduire l'incitation qu'ont les services d'information à mentir pour obtenir le paiement  $\pi$ . En effet, les services d'information n'ont plus la garantie d'impunité puisqu'il existe une probabilité que le maître accepte le service web, découvre que les services d'information ont menti, et décide de punir ceux-ci. Certes, peut-être que le paiement  $\pi$  compense largement la punition, mais il ne faut pas oublier qu'à cela s'ajoute une baisse de la réputation des services d'information concernés auprès du maître ; or, plus sa réputation est basse, moins un service d'information a de chance d'être sollicité par un maître de communauté.

La valeur que l'on donne à la probabilité  $P$  est cruciale. Trop petite, elle ne sert à rien car l'effet de dissuasion devient quasi nul, et trop grande, le maître se mettra à accepter un nombre trop important de mauvais services web. En effet, « le dommage collatéral » de ce « jet de dés », est que le maître se mettra à accepter des services web véritablement mauvais. Il faut donc choisir une valeur de  $P$  qui soit un bon compromis entre, d'une part,

## 6.2 Étude des nouveaux jeux

l'effet de dissuasion, et d'autre part, le dommage collatéral. La probabilité  $P$  sera assez petite, et de toute évidence  $< 0.49$ .

## 6.2 Étude des nouveaux jeux

Les nouveaux jeux étendus que nous avons modélisés peuvent facilement se réduire à de simple jeux stratégiques. Reprenons la figure 6.2. Si tout le monde dit la vérité, SI a  $P$  chance de gagner  $\alpha + \beta$  et  $1 - P$  chance de gagner  $\alpha + \beta + \gamma$ . On peut donc dire que son utilité moyenne dans cette situation est  $P(\alpha + \beta) + (1 - P)(\alpha + \beta + \gamma)$ . Nous pouvons donc représenter les quatre cas par de simples jeux stratégiques.

### 6.2.1 Cas où le service web a une bonne qualité de service

Nous commençons par les deux cas où le service web désirant entrer dans une communauté a une bonne qualité de service. Rappelons que deux situations sont alors possibles : soit un tiers tente de corrompre les services d'information, soit aucune tentative de corruption n'a lieu.

#### Pas de corruption

Le jeu est représenté par ce tableau :

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta + \gamma), (\alpha + \beta + \gamma)$	$(1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma),$ $(1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma^-)$
	Mentir	$(\alpha + \beta^- + \gamma^-), (\alpha + \beta_- + \gamma)$	$(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma^-),$ $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma^-)$

TABLE 6.2 – Le service web est bon et aucune tentative de corruption n'a lieu.

Dans ce cas, le profil d'actions (Dire la vérité, Dire la vérité) reste le seul équilibre de Nash. En effet :

1. la deuxième case n'est pas un équilibre de Nash. On voit que ASI a intérêt à changer de stratégie car  $\alpha + \beta + \gamma > (1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma^-)$ . On peut le montrer facilement :

$$\alpha + \beta + \gamma > (1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma^-)$$

$$\Leftrightarrow \alpha + \beta + \gamma > \alpha + \beta_- + P\gamma^-$$

Cette dernière inéquation est vérifiée, car  $\beta > \beta_-$  et  $\gamma > P\gamma^-$  ;

2. la troisième case n'est pas un équilibre de Nash. On voit que SI a intérêt à changer de stratégie car  $\alpha + \beta + \gamma > \alpha + \beta^- + \gamma^-$  ;
3. la quatrième case n'est normalement pas un équilibre de Nash. On voit que ASI a normalement intérêt à changer de stratégie car  $\alpha + \beta_- + \gamma > (1 - P)(\alpha + \beta) + P(\alpha +$

$\beta + \gamma^-$ ). On peut le montrer comme ceci :

$$\begin{aligned} \alpha + \beta_- + \gamma &> (1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma^-) \\ \Leftrightarrow \alpha + \beta_- + \gamma &> \alpha + \beta + P\gamma^- \\ \Leftrightarrow \gamma &> \beta - \beta_- + P\gamma^- \end{aligned}$$

La dernière inéquation est normalement vérifiée car  $\gamma^-$  est au mieux un paiement positif faible (encore affaiblit par  $P$ ), au pire un paiement négatif, et que  $\beta - \beta_-$  est petit ;

4. la première case est un équilibre de Nash. SI n'a pas intérêt à changer de stratégie car  $\alpha + \beta + \gamma > \alpha + \beta^- + \gamma^-$  et ASI n'a pas intérêt à changer de stratégie non plus car  $\alpha + \beta + \gamma > (1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma^-)$ . De plus, le profil est un optimum de Pareto.

L'instauration de la probabilité  $P$  ne change donc pas la situation. Le système incite toujours les services d'information à dire la vérité dans ce cas.

#### Tentative de corruption par un tiers

Le tableau ci-dessous représente ce jeu :

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(\alpha + \beta + \gamma), (\alpha + \beta + \gamma)$	$(1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma),$ $(1 - P)(\alpha + \beta_- + \pi) + P(\alpha + \beta_- + \gamma^-)$
	Mentir	$(\alpha + \beta^- + \gamma^-), (\alpha + \beta_- + \gamma)$	$(1 - P)(\alpha + \beta + \pi) + P(\alpha + \beta + \gamma^-),$ $(1 - P)(\alpha + \beta + \pi) + P(\alpha + \beta + \gamma^-)$

TABLE 6.3 – Le service web est bon mais un tiers tente de corrompre les services d'information.

Pour que le profil d'actions (Dire la vérité, Dire la vérité) soit un équilibre de Nash, les conditions suivantes doivent être remplies :

- 1.

$$\alpha + \beta + \gamma \geq \alpha + \beta^- + \gamma^-$$

- 2.

$$\begin{aligned} \alpha + \beta + \gamma &\geq (1 - P)(\alpha + \beta_- + \pi) + P(\alpha + \beta_- + \gamma^-) \\ \Leftrightarrow \alpha + \beta + \gamma &\geq \alpha + \beta_- + \pi - P\pi + P\gamma^- \\ \Leftrightarrow \beta + \gamma &\geq \beta_- + (1 - P)\pi + P\gamma^- \\ \Leftrightarrow \beta - \beta_- + \gamma - P\gamma^- &\geq (1 - P)\pi \end{aligned}$$

## 6.2 Étude des nouveaux jeux

---

La première inégalité est toujours vérifiée. La seconde inégalité dépend de la valeur des paiements donnés par le maître et de  $\pi$ . Si les paiements donnés par le maître de communauté sont assez grands, cela devrait encourager les services d'information à dire la vérité. On peut voir par exemple, que si le maître décide de punir un service d'information avec un  $\gamma^-$  négatif, la valeur de la partie gauche de l'inéquation grandit, et l'incitant à dire la vérité est plus élevé. Ce modèle ne nous permet cependant en rien d'affirmer que les services d'information diront toujours la vérité. En effet, un service web pourrait très bien offrir un paiement  $\pi$  énorme, de manière à ce qu'il surpasse toujours les paiements donnés par le maître de communauté. Dans le chapitre suivant, nous effectuerons une étude des jeux sur le long terme, via les jeux répétés, ce qui nous permettra de donner de l'importance à la réputation des services d'information et d'avoir un modèle incitant plus à l'honnêteté.

Voyons maintenant le profil (Mentir, Mentir). Pour qu'il soit un équilibre de Nash, les conditions suivantes doivent être remplies :

1.

$$\begin{aligned}
 (1 - P)(\alpha + \beta + \pi) + P(\alpha + \beta + \gamma^-) &\geq (1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma) \\
 \Leftrightarrow \alpha + \beta + (1 - P)\pi + P\gamma^- &\geq \alpha + \beta^- + P\gamma \\
 \Leftrightarrow \beta - \beta^- + (1 - P)\pi + P(\gamma^- - \gamma) &\geq 0 \\
 \Leftrightarrow (1 - P)\pi &\geq \beta^- - \beta + P\gamma - P\gamma^-
 \end{aligned}$$

2.

$$\begin{aligned}
 (1 - P)(\alpha + \beta + \pi) + P(\alpha + \beta + \gamma^-) &\geq \alpha + \beta_- + \gamma \\
 \Leftrightarrow \alpha + \beta + (1 - P)\pi + P\gamma^- &\geq \alpha + \beta_- + \gamma \\
 \Leftrightarrow \beta - \beta_- + (1 - P)\pi + P\gamma^- - \gamma &\geq 0 \\
 \Leftrightarrow (1 - P)\pi &\geq \beta_- - \beta + \gamma - P\gamma^-
 \end{aligned}$$

Jetons d'abord un œil à la première inégalité.  $\beta^- - \beta < 0$ , et  $P\gamma - P\gamma^- > 0$ . Nous pouvons raisonnablement dire que  $P$  est petit et qu'il réduit donc assez la valeur  $P\gamma - P\gamma^- > 0$ . Il y a donc de fortes chances que le membre de droite de l'inégalité soit négatif.  $(1 - P)\pi$  étant positif, l'inégalité est souvent respectée.

Regardons maintenant la seconde inégalité.  $\beta_- - \beta < 0$ , et  $\gamma - P\gamma^- > 0$ . La même observation que ci-dessus a lieu. Cependant, ici la valeur  $\beta - \beta_-$  est bien plus proche de 0 que ne l'était  $\beta - \beta^-$ . Il est donc difficile de dire si le membre de droite de l'inégalité sera positif ou négatif, et prédire la vérification de l'inégalité s'avère bien plus difficile.

Nous pouvons conclure cette analyse par les deux lemmes suivants :

LEMME 6.1. Si  $\pi \leq \frac{\beta - \beta_- + \gamma - P\gamma^-}{1 - P}$ , alors le profil d'actions (Dire la vérité, Dire la vérité) est un équilibre de Nash.

LEMME 6.2. Si  $\pi \geq \frac{\beta_- - \beta + P\gamma - P\gamma^-}{1 - P}$   $\pi \geq \frac{\beta^- - \beta + \gamma - P\gamma^-}{1 - P}$ , alors le profil d'actions (Mentir, Mentir) est un équilibre de Nash.

### 6.2.2 Cas où le service web a une mauvaise qualité de service

Voyons maintenant les cas où le service web désirent entrer dans une communauté a une mauvaise qualité de service. A nouveau, deux possibilités existent : soit le service web tente de corrompre les services d'information, soit il est honnête.

#### Pas de corruption

Le jeu est représenté par le tableau suivant :

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma),$ $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma)$	$(\alpha + \beta^- + \gamma),$ $(\alpha + \beta_- + \gamma^-)$
	Mentir	$(1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma^-),$ $(1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma)$	$(\alpha + \beta + \gamma^-),$ $(\alpha + \beta + \gamma^-)$

TABLE 6.4 – Le service web a une mauvaise qualité de service et est honnête.

L'étude de ce cas dépend des hypothèses faites sur les valeurs  $\beta^-$  et  $\gamma^-$ . Si on considère qu'elles sont négatives ou nulles, le profil d'actions (Dire la vérité, Dire la vérité) reste le seul équilibre de Nash. En effet :

1. la deuxième case n'est pas un équilibre de Nash. On voit que ASI a intérêt à changer de stratégie car  $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) > (\alpha + \beta_- + \gamma^-)$ . On peut le montrer facilement :

$$(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) > (\alpha + \beta_- + \gamma^-)$$

$$\Leftrightarrow \alpha + \beta + P\gamma > \alpha + \beta_- + \gamma^-$$

Cette inégalité est vérifiée car  $\beta > \beta_-$  et  $P\gamma > \gamma^-$  ;

2. la troisième case n'est pas un équilibre de Nash. On voit que SI a intérêt à changer de stratégie car  $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) > (1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma^-)$ . On peut le monter facilement :

$$(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) > (1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma^-)$$

$$\alpha + \beta + P\gamma > \alpha + \beta^- + P\gamma^-$$

Cette inégalité est vérifiée car  $\beta > \beta^-$  et  $P\gamma > P\gamma^-$  ;

3. la quatrième case n'est normalement pas un équilibre de Nash. On voit que SI a normalement intérêt à changer de stratégie car  $\alpha + \beta^- + \gamma > \alpha + \beta + \gamma^-$ . L'inéquation est normalement vérifiée car dans les hypothèses l'écart entre  $\gamma$  et  $\gamma^-$  est plus grand que celui entre  $\beta$  et  $\beta^-$  ;
4. la première case est un équilibre de Nash. SI n'a pas intérêt à changer de stratégie car  $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) > (1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma^-)$  et ASI n'a pas intérêt

## 6.2 Étude des nouveaux jeux

à changer de stratégie non plus car  $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) > (\alpha + \beta_- + \gamma^-)$ .

L'ajout de la probabilité  $P$  ne change pas grandement l'analyse de ce jeu par rapport à celle faite au chapitre 4. Si on considère  $\beta^-$  et  $\gamma^-$  comme des valeurs négatives ou nulles, le système d'incitation mis en place garantit que les services d'information diront la vérité dans ce cas. Par contre, si on considère  $\beta^-$  et  $\gamma^-$  comme des paiements positifs mais petits (autrement dit, les paiements négatifs ne sont pas autorisés, ou pas atteints), on ne peut plus garantir que le profil d'actions (Dire la vérité, Dire la vérité) est un équilibre de Nash. Un service d'information pourrait donc être incité à mentir pour recevoir le  $\gamma^-$ , faible mais positif, comme expliqué dans la section 5.7.

### Tentative de corruption par le service web

Le jeu est représenté par ce tableau :

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma),$ $(1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma)$	$(\alpha + \beta^- + \gamma),$ $(\alpha + \beta_- + \gamma^- + \pi)$
	Mentir	$(1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma^- + \pi),$ $(1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma)$	$(\alpha + \beta + \gamma^- + \pi),$ $(\alpha + \beta + \gamma^- + \pi)$

TABLE 6.5 – Le service web a une mauvaise qualité de service et est malhonnête.

Pour que le profil d'actions (Dire la vérité, Dire la vérité) soit un équilibre de Nash, les conditions suivantes doivent être respectées :

1.

$$\begin{aligned} (1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) &\geq (1 - P)(\alpha + \beta^-) + P(\alpha + \beta^- + \gamma^- + \pi) \\ \Leftrightarrow \alpha + \beta + P\gamma &\geq \alpha + \beta^- + P\gamma^- + P\pi \\ \Leftrightarrow \beta - \beta^- + P\gamma - P\gamma^- &\geq P\pi \end{aligned}$$

2.

$$\begin{aligned} (1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma) &\geq (\alpha + \beta_- + \gamma^- + \pi) \\ \Leftrightarrow \alpha + \beta + P\gamma &\geq \alpha + \beta_- + \gamma^- + \pi \\ \Leftrightarrow \beta - \beta_- + P\gamma - \gamma^- &\geq \pi \end{aligned}$$

Les deux parties de la première inégalité sont positives. La vérification de l'inégalité dépendra des valeurs des différents paiements, notamment de  $\pi$  et de  $\gamma^-$  qui peuvent être positives comme négatives. Si le paiement  $\pi$  n'est pas très gros, l'inégalité devrait être respectée. On fait la même observation pour la seconde inégalité.

Regardons les conditions pour que le profil d'actions (Mentir, Mentir) soit un équilibre de Nash :

1.

$$(\alpha + \beta + \gamma^- + \pi) \geq (\alpha + \beta^- + \gamma)$$

$$\Leftrightarrow \beta + \gamma^- + \pi \geq \beta^- + \gamma$$

$$\Leftrightarrow \beta^- + \gamma - \beta - \gamma^- \leq \pi$$

2.

$$(\alpha + \beta + \gamma^- + \pi) \geq (1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma)$$

$$\Leftrightarrow \alpha + \beta + \gamma^- + \pi \geq \alpha + \beta_- + P\gamma$$

$$\Leftrightarrow \beta - \beta_- + \gamma^- - P\gamma + \pi \geq 0$$

$$\Leftrightarrow \beta - \beta_- + \gamma^- - P\gamma \geq -\pi$$

$$\Leftrightarrow \beta_- - \beta + P\gamma - \gamma^- \leq \pi$$

La même réflexion que ci-dessus est faite. La vérification ou non des inégalités dépend des valeurs des différentes variables.

Nous pouvons conclure l'analyse de ce cas par les deux lemmes suivants :

LEMME 6.3. Si  $\pi \leq \frac{\beta - \beta^- + P\gamma - P\gamma^-}{P}$  et  $\pi \leq \beta_- - \beta + P\gamma - \gamma^-$ , alors le profil d'actions (Dire la vérité, Dire la vérité) est un équilibre de Nash.

LEMME 6.4. Si  $\pi \geq \beta^- + \gamma - \beta - \gamma^-$  et  $\pi > \beta_- - \beta + P\gamma - \gamma^-$ , alors le profil d'actions (Mentir, Mentir) est un équilibre de Nash.

## Chapitre 7

# Étude des jeux sur le long terme

Dans le chapitre 5, nous avons développé la composante  $h_k(x)$  de la fonction de paiement de manière à ce que les maîtres de communauté soient les plus justes possible envers les services d'information lorsqu'ils les rémunèrent. Cette manière de rémunérer les services d'information plus justement en réponse à leur comportement devrait les inciter à être honnêtes, car ils seront récompensés à leur juste valeur. Cependant, comme nous le montrent les jeux analysés dans le chapitre 6, les actions consistant à mentir peuvent, dans certains cas, mener à un équilibre de Nash. Tout dépend des tailles relatives des différents paiements, notamment du paiement  $\pi$  en cas de corruption. Les jeux présentés dans le chapitre 6 sont des jeux stratégiques, c'est-à-dire des « one-shot ». Leur analyse ne nous permet donc de prédire le comportement des services d'information qu'à court terme. Cependant, un service d'information est un agent qui agit sur le long terme. Si mentir rapporte parfois gros dans l'immédiat, les services d'information pourraient être plus enclins à dire la vérité si mentir les pénalise sur le long terme. Dans ce chapitre, nous allons adapter notre modèle en faisant en sorte d'obliger les services d'information à agir en pensant sur le long terme. Nous espérons que sur le long terme, les services d'information maximisent leurs gains en disant la vérité. Pour étudier une situation modélisée par un jeu sur le long terme, nous pouvons le répéter. Une présentation des jeux répétés a été faite dans la section 3.4.

### 7.1 La patience des services d'information

Comme nous l'avons vu dans la section 3.4, les jeux répétés introduisent la notion de *patience*, matérialisée par le *facteur d'actualisation*  $\delta$ . Celui-ci est compris entre 0 et 1. Plus  $\delta$  est proche de 1, plus les joueurs sont patients. Pour notre problème, nous pouvons considérer que les services d'information sont plutôt patients et ont donc une valeur de  $\delta$  proche de 1. En effet, comme dit dans l'introduction de ce chapitre, ceux-ci ont une activité sur le long terme, qui consiste à tester et noter des services web, puis à « vendre » ces notations. Les services d'information veulent gagner le plus d'argent possible sur le long terme ; ils n'ont pas vraiment d'intérêt à vouloir gagner plus d'argent maintenant au détriment de plus tard. Dans le chapitre 8, lorsque nous effectuerons des simulations, nous paramètrons les services d'information avec des niveaux de patience différents, afin d'analyser les différences de comportement.

## 7.2 La réputation des services d'information

Un moyen efficace pour obliger les services d'information à réfléchir sur le long terme avant de prendre une décision (mentir ou dire la vérité) est de leur donner une réputation. Comme nous l'avons vu dans la section 2.7.2, un simple système de réputation fait office de mécanisme de crédibilité et est donc une incitation à être honnête. En effet, les maîtres de communauté ne consulteront pas les services d'information qui ont une mauvaise réputation. Donner une réputation aux services d'information est donc un premier incitant à l'honnêteté, en considérant que ceux-ci agissent en pensant sur le long terme. Parmi la population de services d'information, les maîtres de communauté ne choisiront que ceux en qui ils ont le plus confiance, c'est-à-dire ceux qui ont les réputations les plus élevées. Si un service d'information pense qu'il doit mentir pour obtenir le plus gros paiement possible (dans l'immédiat), il pourrait hésiter si ce mensonge entraînait une baisse de réputation qui nuirait à ses paiements futurs. Car plus sa réputation est basse, moins un service d'information sera sollicité, et moins il pourra recevoir de paiements.

### 7.2.1 Le système de réputation que nous avons choisi d'utiliser : BRS

Afin d'attribuer une réputation à chaque service d'information, nous avons décidé d'utiliser le système BRS présenté dans la section 2.4. Chaque communauté note les services d'information qu'elle connaît selon les interactions passées qu'elle a eues avec chacun d'eux. Lorsqu'un service web est accepté dans la communauté, le maître peut juger sa qualité de service et déterminer le paiement  $h_k(x)$  qu'il attribue à chaque service d'information à qui il a demandé conseil. En plus de ce paiement, le maître va donner à chaque service d'information une notation binaire selon qu'il juge le service d'information honnête ou malhonnête. A partir de l'ensemble des notations binaires qu'une communauté  $C_i$  a accordées à un service d'information  $I_k$ , on calcule sa réputation comme :

$$R_i(I_k) = \frac{P_i^k + 1}{P_i^k + N_i^k + 2}$$

avec  $P_i^k$  le nombre de notes positives (1) données à  $I_k$  par  $C_i$  pour chaque vérité, et  $N_i^k$  le nombre de notes négatives données à  $I_k$  par  $C_i$  pour chaque mensonge.

#### Pourquoi le système BRS ?

Nous avons choisi le système BRS car la valeur de réputation calculée par celui-ci représente la probabilité moyenne que le service d'information soit honnête dans le futur. Les services d'information ayant le choix entre deux actions (ils disent la vérité ou ils mentent), un type de notation binaire convient parfaitement. La difficulté consiste à savoir si un service d'information a menti ou non. En effet, un service d'information peut très bien être honnête et transmettre une notation assez différente de celle calculée par le maître, car celui-ci n'a pas la même manière d'évaluer les services web. C'est d'ailleurs pour cette raison que nous avons adapté le paiement  $h_k(x)$  aux propriétés, comme dit dans la section 5.2. Cependant, la mise à jour de la réputation ne se fait pas par rapport à la différence entre les notations (celle du maître et celle du service d'information), mais par rapport à

## 7.2 La réputation des services d'information

---

la différence d'interprétation. Comme expliqué dans le chapitre 5, une interprétation est binaire, elle vaut soit *bon*, soit *mauvais*. Dans notre modèle, un système d'information n'a pas à transmettre d'interprétation, celle-ci dépendant de la notation qu'il donne : si un service d'information accorde à un service web une notation égale ou supérieure à 0,5, l'interprétation associée est *bon*, sinon, l'interprétation est *mauvais*. Un maître de communauté considère un service d'information comme menteur si l'interprétation associée à la notation transmise est différente de l'interprétation associée à la notation du service web par le maître.

EXEMPLE 7.1. Un service d'information transmet la notation 0,7 pour un certain service web à un maître de communauté. L'interprétation associée est *bon* car  $0,7 > 0,5$ . Le maître accepte le service web et peut donc le noter lui même. Le service web est noté 0,5 par le maître et est aussi considéré comme *bon*. Pour le maître, le service d'information a dit la vérité. Celui-ci reçoit une note positive et sa réputation augmente.

EXEMPLE 7.2. Un service d'information transmet la notation 0,6 pour un certain service web à un maître de communauté. L'interprétation associée est *bon* car  $0,6 > 0,5$ . Le maître accepte le service web et peut donc le noter lui même. Le service web est noté 0,4 par le maître et est donc considéré comme *mauvais* par celui-ci. Pour le maître, le service d'information a menti. Celui-ci reçoit une note négative et sa réputation diminue.

EXEMPLE 7.3. Un service d'information transmet la notation 0,4 pour un certain service web à un maître de communauté. L'interprétation associée est *mauvais* car  $0,4 < 0,5$ . Le maître n'accepte pas le service web et ne peut donc pas le noter lui même. La réputation du service d'information est inchangée.

On pourrait trouver notre choix étrange, puisque la manière de mettre à jour la réputation d'un service d'information dépend du jugement subjectif qu'aura le maître sur l'honnêteté du service d'information. Ce choix peut donc sembler être en contradiction avec nos efforts de rendre le paiement  $h_k(x)$  le plus juste possible. Ce n'est cependant pas le cas. En effet, la fonction  $h_k(x)$  a pour but de rémunérer les services d'information par rapport à l'importance de l'information transmise dans le calcul de la réputation d'un service web. Mais la mise à jour de la réputation d'un service d'information dépend du fait qu'il ait menti ou non. C'est quelque chose de binaire : soit il ment, soit il dit la vérité. Un service d'information peut soit obtenir une note positive, soit obtenir une note négative, soit ne pas obtenir de note. Le système BRS semble donc s'imposer pour donner une réputation aux services d'information, car celui-ci utilise une notation binaire.

### Système public ou privé

L'avantage du système BRS est qu'il peut être public ou privé. Dans notre modèle, nous l'utilisons de manière privée, c'est-à-dire que chaque communauté calcule la réputation d'un service d'information selon ses propres expériences passées avec celui-ci. C'est l'hypothèse que nous prendrons pour nos simulations. Cependant, le système BRS convient aussi pour un système public local, c'est-à-dire où une communauté calcule la réputation d'un service d'information à partir de ses propres expériences et de l'expérience d'autres communautés

ou encore pour un système global, où la réputation d'un service d'information est une valeur unique, la même dans toutes les communautés, comme le système d'Ebay que nous avons présenté dans la section 2.2.

### 7.2.2 Mise à jour de la réputation d'un service d'information

La première stratégie de mise à jour est celle qui a été présentée à travers les trois exemples ci-dessus. Elle consiste à actualiser la réputation d'un service d'information seulement lorsque le service web a été accepté au sein de la communauté et que le maître a pu vérifier sa qualité de service.

Dans le cas où un service web a une bonne qualité de service et qu'il n'y a pas de corruption, les services d'information doivent théoriquement tous dire la vérité (voir section 6.2). Le service web est alors accepté au sein de la communauté et la réputation de chaque service d'information ayant transmis une notation au maître augmente dans cette communauté.

S'il y a corruption, des services d'information pourraient mentir, et le service web pourrait ne pas être accepté. S'il est tout de même accepté, le maître pourra tester lui-même le service web, augmenter la réputation des services d'information ayant dit la vérité, et baisser la réputation de ceux ayant menti. Si le service web n'est pas accepté, le maître ne pourra le tester lui-même et la réputation des services d'information ne sera pas mise à jour. Afin d'empêcher les problèmes de collusion, nous avons introduit dans le chapitre 6 la probabilité  $P$  qu'un service web soit quand même accepté alors que la réputation calculée par le maître de communauté à partir des notations reçues des différents services d'information est mauvaise. Nous espérons que cette probabilité agisse à long terme comme une menace envers les services d'information qui voudraient mentir sans subir aucun contre-coups (pas de baisse de réputation).

Dans le cas où le service web a une mauvaise qualité de service, et qu'une majorité de services d'information dit la vérité, le service web n'est pas accepté et la réputation des services d'information n'est pas mise à jour. Par contre, s'ils mentent en majorité, le service web sera accepté, et la réputation des menteurs diminuera alors que la réputation des autres augmentera.

L'avantage de cette stratégie est que la réputation d'un service d'information n'est mise à jour que lorsque le maître a lui-même pu vérifier la validité de l'information qu'il a reçue de celui-ci. L'inconvénient est que la réputation d'un service d'information n'est pas mise à jour à chaque interaction.

La seconde stratégie de mise à jour est de toujours actualiser la réputation d'un service d'information ayant transmis une information. Si le service web n'est pas accepté, la réputation d'un service d'information est mise à jour selon l'interprétation associée à la réputation calculée par le maître à partir des notations reçues. L'avantage de cette stratégie

## 7.2 La réputation des services d'information

---

est que la réputation d'un service d'information est mise à jour à chaque interaction avec le maître de communauté. L'inconvénient est que la réputation d'un service d'information peut être actualisée de manière incorrecte (par exemple, dans le cas où un service web est bon mais que la majorité des services d'information mente).

Nous pensons qu'il vaut mieux ne pas mettre à jour la réputation d'un service d'information à chaque interaction que de risquer de la mettre à jour de manière incorrecte. C'est pour cela que nous choisissons la première stratégie de mise à jour pour notre modèle.

### 7.2.3 Pondération par la réputation

Afin de donner plus d'importance à la réputation des services d'information, il est possible de pondérer chaque paiement par la valeur de réputation du service d'information. En faisant cela, le maître pourrait augmenter la valeur maximale des paiements  $g_k(x)$  et  $h_k(x)$  qu'il peut accorder, puisque chaque paiement qu'il donnera sera pondéré par la valeur de réputation du service d'information et donc inférieur (car la réputation est une valeur entre 0 et 1). Regardons un petit exemple explicatif.

EXEMPLE 7.4. Admettons que le maître de communauté ait un budget maximal de 1000. Il fait appel à 10 services d'information et peut donc attribuer un paiement maximum de 100 par service d'information. Admettons qu'il détermine  $f_k(x) = 10$  et fixe les valeurs maximales de  $g_k(x)$  et de  $h_k(x)$  à respectivement 30 et 60. Admettons que le service web désirant entrer dans la communauté est bon et que tous les services d'information disent la vérité. Ils recevront donc tous le paiement maximum de 100. Cependant, si les paiements sont pondérés par les valeurs de réputations des services d'information, le maître sait que le paiement maximum de 100 sera dans les faits plus petit. Il connaît la réputation des services d'information et peut donc en calculer la moyenne. Si la moyenne des réputations est de, par exemple, 0,8, le maître ne distribuera que 800 si tous les services d'information obtiennent le paiement maximum. Le maître peut alors fixer le paiement maximum à 120,5. Cela peut avoir deux avantages :

- les services d'information avec une bonne réputation sont mieux rémunérés que les services avec une rémunération moindre, ce qui pousse ces derniers à être honnêtes pour augmenter leur réputation ;
- les services d'information ayant une réputation proche de 1 gagnent plus que dans un système où la réputation n'est pas prise en compte. Les services d'information à bonne réputation ont donc moins d'incitation à se laisser corrompre dans ce système. C'est important, car le mensonge d'un service d'information à haute réputation est plus préjudiciable pour le maître que le mensonge d'un service d'information à basse réputation (car lorsque le maître calcule la moyenne des notations transmises par les services d'information, celle-ci est pondérée par la réputation desdits services).

On peut cependant voir deux défauts à ce système :

- les services d'information ayant une mauvaise réputation gagnent moins dans ce système que dans le système où leur réputation n'est pas prise en compte, ce qui

- peut baisser l'incitant qu'ils ont à dire la vérité ;
- si tous les services d'information ont une réputation proche de 1, ce système ne change rien.

### Le paiement $\pi$

Nous considérons que la réputation d'un système d'information dans une communauté n'est connue que par le maître et le service d'information lui-même. Un service web ne peut donc pas pondérer son paiement  $\pi$  par rapport à la réputation du service d'information qu'il tente de corrompre, parce qu'il ne la connaît pas. Un service web va donc offrir le même  $\pi$  à tous les services d'information qu'il tente de corrompre. Or, il serait bien plus avantageux pour un service web d'offrir un plus gros paiement  $\pi$  aux services d'information réputés (car leurs notations a un plus gros poids dans le calcul de la réputation par le maître, comme dit dans le chapitre 4), et d'abaisser le paiement  $\pi$  offert aux services d'information peu réputés. Nous pensons qu'opter pour un système de réputation privé et local est donc le plus avantageux, car la réputation de chaque service d'information est une information privée détenue uniquement par le maître de communauté, et il serait difficile pour un service web de la connaître. Dans les simulations que nous présenterons dans le chapitre 8, le paiement  $\pi$  n'est pas pondéré par la réputation du service d'information qui le reçoit.

### Notes sur les simulations

Dans les simulations, on fixe les paramètres de chaque fonction de paiement. Le paiement  $h_k(x)$  maximal possible est fixé et ne change pas au cours d'une simulation. La réflexion exposée ci-dessus a un sens dans un contexte « réel » où un maître possède un budget et qu'il doit déterminer les paramètres de ses paiements. Pondérer les paiements par la réputation permet de pouvoir offrir des valeurs de paiement maximales plus élevées sans sortir des limites du budget. Cependant, dans nos simulations, nous fixons nous-mêmes les paiements maximum possibles, ce n'est pas le maître qui les fixe selon un budget. Nous allons fixer les paramètres des paiements et voir dans quels cas les services d'information disent la vérité, et dans quels cas ils mentent. Il est certain qu'en fixant une valeur de  $\pi$  énorme (50 fois la valeur maximale de  $u_k(x)$  par exemple), les services d'information mentiront tous. La réflexion faite ci-dessus permet de relativiser en disant que dans des cas réalistes, le paiement  $\pi$  ne saurait pas être beaucoup plus haut que les paiement du maître. En effet, d'une part, il semble logique de penser que les communautés ont un budget plus conséquent qu'un simple service web. D'autre part, le maître connaissant la réputation des services d'information, il peut adapter ses paiements et augmenter ceux qu'il offre aux services web réputés (mais il diminuera alors les paiements offerts aux services web moins réputés).

### 7.3 Analyse des jeux répétés

#### 7.3.1 Équilibre de Nash

Une première manière d'étudier les jeux sur le long terme est de trouver le(s) équilibre(s) de Nash de la version répétée des jeux. Selon le théorème de Nash, que nous avons vu à la section 3.4.4, tout profil d'utilités faisables peut être un équilibre de Nash. En effet, dans un jeu répété, une stratégie ne consiste pas forcément à répéter la même action à chaque tour (voir le dilemme du prisonnier répété section 3.4). La stratégie consistant par exemple à dire la vérité neuf fois sur dix pourrait très bien faire partie d'un équilibre de Nash. Dans le chapitre 6, nous avons montré que certains profils d'actions sont des équilibres de Nash sous certaines conditions. Le théorème de Nash nous dit que dans un jeu répété, un profil d'utilités faisables peut être un équilibre de Nash (si les joueurs sont assez patients), si les utilités du profil dépassent l'utilité minmax pour chaque joueur. Un raisonnement intuitif nous permet de penser que si un profil d'actions  $a_1, \dots, a_n$  constitue un équilibre de Nash dans un jeu stratégique, alors le profil de stratégies correspondant à jouer ces mêmes actions à chaque tour constitue un équilibre de Nash dans le jeu répété. En fait, la notion d'équilibre de Nash pour les jeux répétés n'est intéressante que si l'on étudie des stratégies plus complexes (comme la stratégie de coopération du dilemme du prisonnier répété vue à la section 3.4). Dans notre cas, une étude théorique des équilibres de Nash des jeux répétés seraient à la fois trop complexe (quelles stratégies étudier ?) et relativement inutile, puisque nous avons la possibilité de faire des simulations. En effet, lors d'une simulation, à chaque fois qu'un service d'information doit choisir une action (mentir ou dire la vérité), il analyse un jeu, stratégique ou répété selon les paramètres entrés. Nous verrons un panel de simulations dans le chapitre 8.

#### 7.3.2 Remarques sur les jeux répétés

Un jeu répété consiste à répéter un même jeu stratégique. Les utilités d'un jeu stratégique sont des constantes déterminées par la fonction d'utilité. Elles ne changent pas d'un tour à l'autre. Or, dans notre cas, les paiements  $g_k(x)$  et  $h_k(x)$  sont calculés à chaque tour, et diffèrent donc (légèrement) d'un tour à l'autre. Dans notre analyse, nous considérons  $\alpha, \beta, \beta_-, \beta^-, \gamma, \gamma^-$ , et  $\pi$  comme des constantes. Leurs valeurs sont identiques d'un tour à l'autre. Notre but ici est d'analyser le comportement à long terme des services d'information lorsque leurs paiements sont pondérés par leur réputation. La réputation va donc pondérer l'utilité, et variera d'un tour à l'autre. C'est le seul facteur qui va varier d'un tour à l'autre. Théoriquement, on peut aussi voir la réputation comme un élément semblable à la patience, c'est-à-dire ne faisant pas partie de l'utilité dans le jeu stratégique, mais intervenant dans le jeu répété.

Il est aussi nécessaire d'avoir en tête que nous répétons le même jeu, alors que dans la pratique, un service d'information peut faire face à une situation  $a$ , puis faire ensuite face à une situation  $b$ , avant d'être à nouveau en face de la situation  $a$ , etc.

### 7.3.3 Analyse intuitive du cas d'un service web malhonnête avec une mauvaise qualité de service

Nous allons essayer de faire une analyse intuitive du cas où un service web avec une mauvaise qualité de service tente de corrompre les services d'information avec un paiement  $\pi$  pour que ceux-ci surévaluent la notation que chacun lui accorde. Voici le tableau représentant le jeu, dans lequel nous ajoutons la réputation pour pondérer les paiements  $\alpha$ ,  $\beta$ ,  $\beta_-$ ,  $\beta^-$ ,  $\gamma$ , et  $\gamma^-$ .

		ASI	
		Dire la vérité	Mentir
SI	Dire la vérité	$R_{SI}((1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma)),$ $R_{ASI}((1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma))$	$R_{SI}(\alpha + \beta^- + \gamma),$ $R_{ASI}(\alpha + \beta_- + \gamma^-) + \pi$
	Mentir	$(1 - P)R_{SI}(\alpha + \beta^-) + P(R_{SI}(\alpha + \beta^- + \gamma^-) + \pi),$ $R_{ASI}((1 - P)(\alpha + \beta_-) + P(\alpha + \beta_- + \gamma))$	$R_{SI}(\alpha + \beta + \gamma^-) + \pi,$ $R_{ASI}(\alpha + \beta + \gamma^-) + \pi$

TABLE 7.1 – Jeu lorsque le service web est malhonnête avec une mauvaise qualité de service.

Nous avons vu que selon la valeur de  $\pi$  par rapport aux autres paiements, le profil (Dire la vérité, Dire la vérité), et le profil (Mentir, Mentir) peuvent être équilibres de Nash. Que se passe-t-il si on répète le jeu ? Lorsqu'un service d'information n'a pas encore eu d'interaction avec un maître de communauté, sa réputation est de 0,5 (voir la formule que nous rappelons dans la section 7.2). Nous posons l'hypothèse qu'un maître de communauté ne désire pas faire confiance à un service d'information qui a plus de chance de mentir que de dire la vérité, c'est à dire qui a une réputation inférieure à 0,5. Un service d'information avec une réputation inférieure à 0,5 reçoit donc une utilité de 0 car en théorie il n'est plus sollicité par le maître.

#### Les services d'information décident de toujours mentir

Les services d'information obtiennent chacun le paiement suivant :

$$\frac{1}{2}(\alpha + \beta + \gamma^-) + \pi$$

Après le premier tour du jeu répété, la réputation d'un service d'information sera descendue à  $\frac{1}{3}$ , trop petite pour qu'il soit sollicité par le maître de communauté. L'utilité totale du jeu répété est donc uniquement celle du premier tour.

#### Les services d'information décident de toujours dire la vérité

Pour un jeu répété à  $N$  tours, l'utilité totale reçue par le service d'information SI est la suivante :

$$\sum_{t=0}^N \delta^t R_{SI}((1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma))$$

$\delta$  est le facteur d'actualisation incarnant la patience de SI.  $R_{SI}$  est sa réputation. Puisque les services d'information disent toujours la vérité, les services web ne sont pas acceptés avec une haute probabilité de  $(1 - P)$ . La réputation d'un service d'information est actualisée seulement lorsque le service web a été accepté au sein de la communauté et que le maître

### 7.3 Analyse des jeux répétés

---

a pu vérifier sa qualité de service. Dans ce cas, lorsque les services d'information disent la vérité, leur réputation n'a qu'une probabilité  $P$  d'augmenter. Si  $P = 0, 1$ , alors la réputation des services d'information n'augmente en moyenne qu'une fois tous les dix tours. L'utilité moyenne d'un service d'information au bout de  $N$  tours est donc de :

$$\sum_{t=0}^N \delta^t \left( \frac{Pt + 1}{Pt + 2} \right) ((1 - P)(\alpha + \beta) + P(\alpha + \beta + \gamma))$$

#### Les services d'information décident de mentir dès qu'ils peuvent

Un service d'information pourrait choisir la stratégie de mentir dès qu'il peut, c'est-à-dire dès que sa réputation a augmenté. Si la probabilité  $P$  est égale à  $0, 1$ , le service d'information ayant choisi cette stratégie ne mentira qu'en moyenne une fois sur 10. On peut penser qu'il préférera augmenter petit à petit sa réputation afin d'obtenir des paiements du maître plus élevés, que de garder une réputation constante de  $0, 5$  et recevoir de temps en temps un paiement  $\pi$ .

#### 7.3.4 Conclusion

Tenter d'analyser les jeux répétés pour trouver les différents équilibres de Nash possibles dans les 4 scénarios nous paraît fastidieux et peu utile. L'analyse intuitive du dernier cas nous permet cependant de comprendre l'importance du système de réputation des services d'information. En effet, un service d'information ayant une réputation inférieure à  $0, 5$  ne sera pas repris (car la probabilité qu'il mente lors de la prochaine interaction est plus grande que la probabilité qu'il dise la vérité). Un service d'information agissant sur le long terme ne mentira pas en risquant de faire baisser sa réputation en dessous de  $0, 5$  même s'il reçoit un paiement plus élevé sur le court terme.

Nous allons, dans le chapitre suivant, présenter des simulations que nous avons effectuées, afin de voir le comportement des services d'information dans certains cas. Dans une simulation, un certain nombre de services web désire entrer dans différentes communautés. Les maîtres font donc appel aux services d'information afin de calculer la réputation des services web désirant entrer. Afin de décider s'il va mentir ou non, chaque service d'information, à chaque sollicitation d'un maître, va analyser un jeu stratégique ou répété (selon les paramètres de la simulation), afin de choisir l'action lui apportant la plus grande utilité. Selon que le nombre de fois qu'un service d'information répète le jeu, et sa patience, l'utilité qu'il calcule peut changer. Par exemple, un service web très patient accordera autant d'importance à un paiement à recevoir possiblement dans le futur qu'un paiement immédiat dans le calcul de l'utilité, ce qui ne sera pas le cas d'un service d'information plutôt impatient. Nous allons donc, dans le chapitre qui suit, présenter un panel de simulations, et tenter de conclure sur l'efficacité de notre modèle : incite-t-il les services d'information à toujours dire la vérité, à dire la vérité dans certaines situations, ou ne les incite-t-il pas du tout ?



## Chapitre 8

# Expérimentations par simulations

Dans ce chapitre, nous allons présenter les résultats de nos expérimentations. Celles-ci ont été effectuées avec un simulateur que nous avons développé à partir d'un existant. Une présentation rapide de ce simulateur est faite dans la section 8.1. L'annexe A le présente de manière plus détaillée. Dans la section 8.2, nous présentons et explicitons les hypothèses que nous avons posées pour les simulations. Enfin, dans la section 8.3, nous analysons un panel de simulations de manière à présenter les résultats les plus importants.

Notre but, en réalisant des simulations, est de voir dans quels cas, ou quels types de cas, les services d'information disent la vérité, et dans quels cas ils mentent. On pourra ainsi voir si le système d'incitation mis en place fonctionne bien dans la plupart des cas, dans certains types de cas, ou ne fonctionne pas du tout. Notre but est de montrer qu'à long terme, les services d'information plutôt patients préféreront être honnêtes. A la fin de ce chapitre, nous critiquerons notre modèle en fonction des résultats des simulations.

### 8.1 Présentation rapide du simulateur

Le simulateur utilisé a été initialement créé lors d'un stage (2010-2011). Nous avons repris la partie graphique, et ré-implémenté la partie technique, étant donné que notre modèle est au final assez différent de l'article [BK11b]. En effet, en plus d'avoir modifié la fonction de paiement (présentée dans le chapitre 5) et ajouté la prise en compte du nouveau cas (présenté dans le chapitre 6), nous avons fait en sorte que les services d'information puissent « penser » sur le long terme en analysant des jeux répétés et non de simples jeux stratégiques (voir chapitre 7).

Nous allons maintenant voir les différents paramètres disponibles pour chaque type d'agents (les communautés, les services web, et les services d'information), ainsi que les hypothèses posées pour la réalisation du simulateur.

### 8.1.1 Les services web

#### Les paramètres

- La qualité de service : cette valeur comprise dans l'intervalle  $]0, 1[$  représente la qualité de service du service web. Plus la valeur est proche de 1, plus la qualité de service est élevée. Il est impossible qu'un service web ait une qualité de service égale à 1, ou égale à 0.
- La variabilité de qualité de service : cette valeur représente le pourcentage de variabilité de la qualité de service. Plus cette valeur est élevée, plus la qualité de service déterminée par un service d'information risque de différer de la valeur réelle. Ce paramètre est utile pour permettre une hétérogénéité dans les notations des services d'information. Chaque service d'information a ainsi une notation légèrement différente.
- Le montant  $\pi$  : cette valeur représente le paiement qu'un service web malhonnête et ayant une mauvaise qualité de service accorde à un service d'information pour qu'il mente.
- Le montant «  $\pi$ -step » : cette valeur représente de combien un service d'information malhonnête va augmenter le montant  $\pi$  pour tenter de corrompre les services d'information à la prochaine tentative d'entrer dans une communauté, s'il a échoué dans la précédente.
- Le nombre maximum de tentatives d'entrer dans une communauté : si le service web n'est toujours pas entré dans une communauté après un certain nombre de tentatives, il abandonne.

#### Les hypothèses

- L'unique but d'un service web est d'entrer dans une communauté.
- Un service web ne désire pas entrer dans une communauté en particulier, il en choisit une au hasard.
- Un service web ne tente pas d'entrer dans une communauté qui l'a refusé.
- Un service web ne peut appartenir qu'à une seule communauté.
- La qualité de service d'un service web ne change pas au cours du temps.
- Un service web ne désire pas quitter une communauté.
- Un service web ne connaît pas la réputation d'un service d'information pour une

## 8.1 Présentation rapide du simulateur

---

communauté.

### 8.1.2 Les communautés

#### Les paramètres

- Nombre de services d'information utilisés : paramètre fixant le nombre de services d'information différents qu'une communauté consulte lorsqu'un service web désire y entrer.
- Seuil d'acceptation d'un service web : seuil qui représente la valeur minimale de réputation pour qu'un service web soit accepté au sein de la communauté.
- Seuil de consultation d'un service d'information : seuil qui représente la valeur minimale de réputation pour qu'un service d'information soit consulté par la communauté.

#### Les hypothèses

- Une communauté considère un service web comme bon si sa réputation calculée égale ou dépasse le seuil d'acceptation. Ce seuil d'acceptation est toujours égal à 0,5 dans nos simulations.
- Une communauté n'a pas de limite de taille.
- Une communauté connaît tous les services d'information du système.
- Une communauté calcule elle-même la réputation de chaque service d'information uniquement à partir des interactions passées qu'elle a eues avec lui. Elle utilise le système BRS pour calculer la réputation d'un service d'information.

### 8.1.3 Les services d'information

#### Les paramètres

Le seul paramètre est la *patience*. Plus celui-ci est proche de 1, plus le service d'information est patient. Ce paramètre prend sa valeur dans l'intervalle  $]0, 1]$ .

#### Les hypothèses

- Un service d'information possède une réputation différente dans chaque communauté. On considère que les communautés ne se consultent pas pour calculer la réputation d'un service d'information.
- Un service d'information connaît sa réputation dans une communauté.
- Un service d'information connaît les seuils d'acceptation et de consultation de chaque communauté.

## 8.2 Paramétrage d'une simulation et hypothèses posées

Nous allons maintenant voir les différents paramètres d'une simulation et les hypothèses posées. Certains paramètres ont déjà été présentés dans la section précédente. Nous les expliquerons plus en détails.

### Population de services web, de communautés, et de services d'information

Il faut tout d'abord fixer les paramètres concernant les populations d'agents, c'est-à-dire les communautés, les services web, et les services d'information.

Nous pouvons fixer le nombre de communautés de services web dans le système. Par défaut, nous mettons 10 communautés. En mettre plus complique l'analyse des simulations (nous avons 4 graphiques par communauté), mais en mettre moins limite fortement les simulations (les communautés n'ont pas assez de choix). Pour chaque communauté, on peut choisir le nombre de services d'information qu'elle consulte pour prendre une décision. Par défaut, nous mettons ce paramètre à 5, soit la moitié des services d'informations disponibles. On peut également fixer la valeur de réputation minimale d'un service web pour que la communauté l'accepte (0,5 par défaut, nous ne le changerons pas), et la réputation minimale d'un service d'information pour que la communauté le consulte (0,5 par défaut, nous ne le changerons pas).

Nous pouvons ensuite fixer le nombre de services web. Le minimum est de 100, et la valeur par défaut est de 1000. 1000 services web suffisent pour faire converger une simulation, en mettre plus ne ferait qu'augmenter inutilement le temps de la simulation. Chaque service web peut essayer d'entrer dans une communauté un certain nombre de fois. Ce nombre peut être également fixé. Lors d'une simulation, chaque service web va tenter d'entrer dans une communauté choisie au hasard. S'il est refusé, il va tenter d'entrer dans une autre communauté (après avoir éventuellement augmenté  $\pi$ ), et ce, jusqu'à que le nombre d'essais maximum fixé soit atteint, ou que le service web ait tenté d'entrer dans chaque communauté. Par défaut, chaque service web ne fait qu'un seul essai. Le problème en mettant plus d'un essai est qu'il est difficile d'analyser les résultats (puisque le  $\pi$  augmente à chaque tentative, on ne sait pas dire en regardant les graphiques, à quel moment  $\pi$  est devenu assez grand). Il est plus facile de faire deux simulations en changeant la valeur de  $\pi$ . Le simulateur permet de créer différentes catégories de services web. La population de services web est répartie entre ces catégories. On peut déterminer le nombre de catégories et la répartition de la population de services web dans chacune d'elles. Par exemple, nous pouvons créer trois catégories et répartir 50% des services web dans la première, 30% dans la deuxième, et 20% dans la troisième. Par défaut, il n'y a qu'une seule catégorie de services web. Pour chaque catégorie de services web, nous pouvons définir l'intervalle de qualité de service (compris entre 0,01 et 0,99), le pourcentage de la population totale de services web que cette catégorie concerne (s'il n'y a qu'une catégorie il sera de 100%), la variabilité de la qualité de service (plus elle est élevée plus les services d'information auront des mesures différentes), et enfin, le paiement  $\pi$  que les services web de la catégorie sont

## 8.2 Paramétrage d'une simulation et hypothèses posées

---

prêts à donner pour entrer dans une communauté (les services web d'une catégorie sont automatiquement malhonnêtes si ce paramètre n'est pas nul). Par défaut,  $\pi = 0$ . Quant à la variabilité, nous la laisserons toujours à 0,3, car au fur et à mesure des tests, c'est cette valeur qui nous semblait optimale (les services d'information ont des notations différentes de chaque services web, mais pas trop).

Enfin, il reste les paramètres concernant les services d'information. Nous pouvons changer le nombre de services d'information, mais nous l'avons fixé par défaut à 10. Moins semble insuffisant pour la diversité des services d'information, plus semble inutile, et complexifie l'analyse des résultats d'une simulation. Chaque service d'information a un facteur de patience. Par défaut, il y a 10 services d'information avec une patience 0,1 à 1, par saut de 0,1.

### Paielements

Parlons à présent des paramètres concernant les différents paiements. Tout d'abord, nous pouvons fixer la valeur d' $\alpha$  (hors pondération par la réputation bien sûr), la valeur de  $\beta$  maximale et la valeur de  $\gamma$  maximale (il s'agit en fait du numérateur des fonctions  $g_k(x)$  et  $h_k(x)$ ). Par défaut,  $\alpha\_max = 5$ ,  $\beta\_max = 20$ , et  $\gamma\_max = 40$ . Nous avons bien  $\beta\_max > \alpha\_max$ , et  $\gamma\_max > \alpha\_max + \beta\_max$ . Pour permettre les paiements négatifs, il faut fixer les valeurs des paramètres  $\beta\_cr$  et  $\gamma\_cr$ , qui déterminent la constante à enlever à chaque paiement. En effet, comme le montrent les formules du chapitre 5, les paiements sont de base toujours positifs, et on retire une constante afin de permettre les paiements négatifs. Cette constante est par défaut la moitié du numérateur (déterminant donc le paiement maximal). Ainsi, en fixant  $\beta\_max = 20$  et  $\beta\_cr = 0$ , les paiements sont toujours positifs ( $]0, 20]$ ). En fixant  $\beta\_cr = 10$ , les paiements peuvent être négatifs ( $] -10, 10]$ ). Il reste le paramètre  $\pi$ -step, qui n'est pas un paiement de la part du maître de communauté, mais la valeur de l'augmentation de  $\pi$  à chaque nouvel essai de corruption de la part d'un service web malhonnête. Si le service web est refusé une première fois par un maître de communauté, il essaye d'entrer dans une autre communauté en payant sa valeur de  $\pi$  précédente augmentée de la valeur de  $\pi$ -step. Ce paramètre est égal à 5 par défaut.

### Autres paramètres

Le dernier écran de paramétrage propose de fixer la probabilité  $P$  (voir chapitre 6), qui est pour rappel la probabilité que le maître de communauté accepte un service web pourtant désigné comme mauvais (afin de vérifier la valeur de vérité des évaluations données par les services d'information). Ce paramètre est fixé à 0,1 par défaut. Nous pensons que 0,1 est un bon compromis, ni trop gros, ni trop petit. Nous verrons dans les simulations ce qui se passe lorsqu'il varie.

Il y a ensuite le paramètre servant à déterminer le nombre de coups à l'avance qu'un service d'information va simuler pour prendre sa décision entre mensonge ou vérité (autrement dit, le nombre de tours des jeux répétés que les services d'information vont analyser pour prendre une décision). Par défaut il est de 5. Dans son processus de prise de décision,

via les jeux répétés, un service d'information se base sur une suite d'interactions (égale au nombre de coups fixés) représentant le même cas que celui actuel, mais avec sa valeur de réputation personnelle changeante. Le nombre de coups à l'avance que peut effectuer un service d'information est le même pour tous les services d'information. Nous n'allons pas au-delà des 10 coups à l'avance car cela ne semble d'une part pas nécessaire pour faire converger vers une préférence (entre mentir ou dire la vérité) et d'autre part parce que le temps pris pour terminer la simulation est exponentiellement croissant.

Le paramètre suivant fixe la probabilité d'existence de corruption par un tiers lorsqu'un service web est bon mais qu'un concurrent ne veut pas le voir rentrer. Par défaut il est de 0. Dans le cas où cette probabilité n'est pas nulle, nous pouvons fixer la valeur du paiement  $\pi$  de ce tiers.

Enfin, il est possible de déterminer si les services d'information disent de petits mensonges (c'est-à-dire qu'ils mentent en s'éloignant le moins possible de la notation réelle), ou des mensonges plus conséquents. Par exemple, admettons qu'un service d'information note un service web à 0,3. Un petit mensonge consiste à transmettre 0,5, autrement dit la moyenne (la note minimale pour qu'un service web soit accepté). Un gros mensonge consisterait à transmettre 0,9. Par défaut, les services d'information mentent en donnant une notation juste au-dessus ou juste en-dessous du seuil d'acceptation de la communauté. Cela permet pour un service d'information de limiter la perte de paiements  $\beta$  et  $\gamma$ , voire d'empêcher un paiement négatif. Cependant, cela limite les chances de recevoir  $\pi$ . En effet, le paiement  $\pi$  n'est donné à un service d'information que s'il a menti, et que son mensonge a fonctionné. Or, dire de petits mensonges risque parfois de ne pas être suffisant lorsque d'autres services d'information disent la vérité.

**EXEMPLE 8.1.** Prenons 5 services d'information ayant notés un service web. Les notations sont les suivantes : 0,3; 0,25; 0,35; 0,28; 0,32. Si les services d'information ont tous la même réputation, la réputation du service web calculée par la communauté est de 0,3 et le service web n'est pas accepté au sein de la communauté. Admettons que 3 des 5 services d'information décident de mentir de manière conséquente. Ils augmentent leur notation de 0,5. Le nouvel ensemble de notations est le suivant : 0,8; 0,75; 0,85; 0,28 : 0,32. La réputation du service web calculée par la communauté est de 0,6, soit suffisante pour que le service web soit accepté au sein de la communauté. Par contre, si les services d'information disent de plus petits mensonges, en n'augmentant chacun leur note que de 0,25, on obtient l'ensemble de notations suivant : 0,55; 0,50; 0,60; 0,28 : 0,32. La réputation calculée par la communauté est de 0,45 et le service web n'est pas accepté. Les mensonges n'ont pas fonctionné et les 3 menteurs ne reçoivent pas de paiements  $\pi$ . Ils recevront par contre de meilleurs paiements  $\beta$  et  $\gamma$ .

Dire de petits mensonges limite donc le risque d'avoir un  $\beta$  ou un  $\gamma$  négatif, mais augmente le risque de ne pas recevoir  $\pi$ . Dans certaines simulations, nous changerons ce paramètre afin d'avoir des mensonges plus conséquents.

### 8.2.1 Remarques sur le fonctionnement des simulations

#### La variabilité de la qualité de service

La qualité de service d'un service web ne change pas au court du temps. Le paramètre « variabilité de la qualité de service » influe sur l'hétérogénéité des notations des services d'information. Par exemple, si ce paramètre a une petite valeur, les services d'information noteront un service web avec une qualité de service à 0,8 entre 0,75 et 0,85, alors que si le paramètre a une valeur plus élevée, un service web avec une qualité de service à 0,8 sera noté entre 0,65 et 0,95. Nous laissons toujours ce paramètre à 0,3. Chaque service d'information aura donc une évaluation assez précise, bien que non parfaite, de chaque réputation des services web.

#### Déterminisme des simulations

Le simulateur est non déterministe. Deux exécutions ayant les mêmes paramètres d'entrée auront des résultats légèrement différents. Cela est normal et est dû aux faits que :

- le maître de communauté accepte un service web désigné comme mauvais avec une probabilité  $P$  ;
- les réputations des services web sont connues des services d'information avec une petite variance aléatoire, les services d'information n'ont donc pas les mêmes notations d'une simulation à l'autre ;
- les services web font leur demande pour entrer dans une communauté dans un ordre différent d'une simulation à l'autre ;
- un service web choisit une communauté au hasard, il ne désirera pas forcément entrer dans la même communauté d'une simulation à l'autre ;
- etc.

#### Mise à jour de la réputation des services d'information

Dans les simulations, le maître de communauté n'opère un changement de réputation sur les services d'information ayant participé au don d'information que dans le cas où il accepte le service web et qu'il peut donc tester lui même sa qualité de service.

#### Limitation due au petit nombre de services d'information

Nous avons vu qu'un maître de communauté consultait les services d'information s'ils avaient une réputation minimale fixée. La réputation d'un service d'information descend s'il ment et qu'il se fait prendre par le maître de communauté lors de ce mensonge. Descendu en-dessous du seuil d'acceptation, il ne sera plus pris sauf si quasi tous les autres services d'information ont une moins bonne réputation. En effet, dans les faits (et pas dans le calcul des coups à l'avance évidemment), lorsque tous les services d'information ont une réputation inférieure à la réputation minimale requise pour être consultés par le maître de communauté (si cela arrive), le maître décide de re-piocher au hasard des services d'information, et de leur donner une nouvelle chance.

### Donnée connue du simulateur et non des agents

Dans les graphiques, nous pourrions voir la répartition des services web dans les communautés : ceux qui n'ont pas été acceptés, et ceux qui l'ont été. Mais nous pouvons voir également si les services web acceptés et refusés étaient bons ou mauvais. Nous pourrions ainsi avoir une idée du pourcentage de bonnes décisions du maître. Le maître de communauté ne sait pas s'il a bien fait ou non de refuser un service web, il ne peut le vérifier. Bien que le graphique illustre cette distinction entre bonne ou mauvaise décision, il ne s'agit que d'une donnée non connue du maître de communauté, mais connue du programme afin de pouvoir illustrer sur un graphique cette distinction, qui nous semblait nécessaire et pertinente.

### Précision quant à l'utilisation des jeux répétés par le simulateur

Nous avons implémenté le simulateur pour qu'il utilise les jeux répétés. Chaque service d'information va décider de l'action qu'il va exécuter (mentir ou dire la vérité en donnant la bonne évaluation d'un service web entrant), en simulant une suite de coups à l'avance afin de voir quelle suite de coups est la plus prometteuse en terme de paiements. Nous avons posé qu'à part un changement de la réputation du service d'information en fonction de ses coups, la situation reste la même. C'est à dire que, par exemple, si la situation est celle d'un mauvais service web, tous les coups joués à l'avance se dérouleront dans cette situation. Le service d'information simule le jeu auquel il est confronté de manière répétée. Bien sûr, le facteur d'actualisation  $\delta$ , représentant la patience, est pris en compte. Dans le chapitre précédent, nous avons vu que si la réputation d'un service d'information tombe en-dessous d'un certain seuil, le service d'information ne gagne plus rien, car il n'est plus pris dans aucune interaction. Lorsqu'un service d'information fait sa propre simulation des jeux répétés, il prend ce fait en compte.

### Rappel sur le paiement $\pi$

Les services d'information ne reçoivent un paiement  $\pi$  que si le service web avec une bonne qualité de service est rejeté dans le cas d'un paiement  $\pi$  venant de services web tiers, ou si le service web avec une mauvaise qualité de service est accepté dans le cas d'un paiement  $\pi$  de sa part pour rentrer dans la communauté.

Pour rappel, nous estimons qu'aucun service web ne paiera un  $\pi$  pour rentrer dans la communauté s'il a une bonne qualité de service, et qu'aucun service web tiers n'interviendra avec un paiement  $\pi$  si le service web entrant a une mauvaise qualité de service. Le premier cas est logique, car le service web sait qu'il sera accepté si le maître de communauté connaît sa bonne évaluation. Bien que des services web tiers pourraient payer un  $\pi$  pour l'empêcher d'entrer dans la communauté, nous posons que le service web entrant ne prend pas cela en compte et qu'il ne paie donc pas un  $\pi$  servant à contrer celui de services web tiers. Le second cas pourrait sembler moins logique, puisque après tout, le service web entrant pourrait demander à un service web tiers de l'aider à payer un  $\pi$  aux services d'information pour entrer. Nous ne prenons tout de même pas en compte cette situation. Pour résumer

### 8.3 Résultats et analyse de cas spécifiques et pertinents

ces deux cas non pris en compte, nous pourrions dire que nous posons qu'il n'existe pas de communication entre les services web entrant et les autres. Un service web entrant ne saura donc jamais si un service web présent dans la communauté paie contre lui, tout comme il ne pourra jamais lui demander de l'aider.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

#### 8.3.1 Les paramètres par défaut

Le tableau suivant montre les paramètres par défaut. Pour chaque simulation, seuls les paramètres ayant été modifiés seront mentionnés.

Nombre de communautés	10
Nombre de services d'information à consulter lors d'une demande d'entrée d'un service web	5
Réputation minimale d'un service web pour qu'il soit accepté	0,5
Réputation minimale d'un service d'information pour qu'il soit consulté	0,5
Nombre de services web	1000
Nombre de tentatives maximum par service web d'entrer dans une communauté	1
Nombre de catégories de services web	1
Intervalle de qualité de service	]0, 1[
Variabilité de la qualité de service	0,3
Valeur de $\pi$ proposée par un mauvais service web	0
Valeur de $\pi$ -step	5
Nombre de services d'information	10
Patience des services d'information	1 pour le premier, 0,9 pour le deuxième, 0,8 pour le troisième, jusqu'à 0,1 pour le dixième
Paiement $f_k(x)$	5
Numérateur du paiement $g_k(x)$	20
Numérateur du paiement $h_k(x)$	40
Constante de retrait du paiement $g_k(x)$	10
Constante de retrait du paiement $h_k(x)$	20
Probabilité P	0,1
Nombre de coups d'un jeu répété	5
Probabilité de corruption par un tiers dans le cas d'un bon service web désirant entrer dans une communauté	0
Valeur de $\pi$ en cas de corruption par un tiers	0
Type de mensonges	petits mensonges, les services d'information qui mentent transmettent de fausses notations proches du seuil d'acceptation.

TABLE 8.1 – Paramètres par défaut d'une simulation.

### 8.3.2 Plan des simulations

Voici notre plan de simulations. Chaque simulation est brièvement décrite mais les détails sont explicités dans les sous-sections de chacune des simulations.

- Simulation 1 : Tester l'honnêteté des services d'information lorsque les services web sont mauvais et donnent un  $\pi$  de 40, sans jeux répétés. Cette simulation a pour but de vérifier que tous les services d'information mentent, comme cela devrait logiquement être, vu la valeur de  $\pi$ .
- Simulation 2 : Tester l'honnêteté des services d'information lorsque les services web sont mauvais et donnent un  $\pi$  de 40, avec jeux répétés de 8 tours. Lorsqu'un service d'information ment, ce mensonge est petit (voir sous-section 8.2). Nous allons comparer cette simulation à la première afin de voir si les services d'informations plutôt patients préfèrent dire la vérité lorsqu'ils réfléchissent sur le long terme.
- Simulation 3 : Tester l'honnêteté des services d'information lorsque les services web sont mauvais et donnent un  $\pi$  de 40, avec jeux répétés de 8 tours. Lorsqu'un service d'information ment, ce mensonge est moyen (voir sous-section 8.2). Nous allons comparer cette simulation à la seconde pour voir le fait que les mensonges soient plus conséquent influe sur les décisions des services d'information.
- Simulation 4 : Tester l'honnêteté des services d'information lorsque les services web sont mauvais et donnent un  $\pi$  de 40, avec jeux répétés de 8 tours. Les services d'information ont tous un facteur de patience de 1. Avec cette simulation, nous aimerions voir si les services d'information disent toujours la vérité lorsqu'ils sont tous extrêmement patient et réfléchissent sur le long terme.
- Simulation 5 : Tester l'honnêteté des services d'information lorsque les services web sont mauvais et ne donnent pas de  $\pi$ , sans jeux répétés. Avec cette simulation, nous voulons vérifier si les services d'information, lorsqu'ils ne réfléchissent pas sur le long terme, peuvent mentir même lorsqu'on ne leur propose pas de  $\pi$ , et ce afin obtenir un  $\gamma^-$  parfois positif (voir section 5.7).
- Simulation 6 : Tester l'honnêteté des services d'information lorsque les services web sont mauvais et ne donnent pas de  $\pi$ , sans jeux répétés. Les punitions sont plus grandes que dans la simulation 5. Nous allons comparer cette simulation avec la précédente afin de voir si les services d'information sont plus honnêtes lorsque la fonction  $h_k(x)$  punit plus facilement ( $\gamma^-$  a plus de chance d'être négatif).
- Simulation 7 : Tester l'honnêteté des services d'information lorsque les services web sont bons, sans jeux répétés. Il n'y a pas de  $\pi$  de services tiers. Cette simulation a pour but de vérifier un résultat théorique : tous les services d'information doivent dire la vérité dans ce cas.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

---

- Simulation 8 : Tester l'honnêteté des services d'information lorsque les services web sont bons, avec jeux répétés de 4 tours à l'avance. Il y a un  $\pi$  de services tiers égal à 40 et proposé avec une probabilité de 0,5. Cette simulation a pour but de voir à quel point la corruption d'un tiers peut influencer les services d'information à mentir en abaissant la notation d'un bon service web.
- Simulation 9 : Tester l'honnêteté des services d'information lorsque les services web sont bons, avec jeux répétés de 8 tours à l'avance. Il y a un  $\pi$  de services tiers égal à 40 et proposé avec une probabilité de 0,5. Avec cette simulation, nous aimerions voir si réfléchir à plus long terme (jeu répété de 8 tours au lieu de 4) change les résultats pour les services d'informations les plus patients.
- Simulation 10 : Tester l'honnêteté des services d'information lors d'une simulation générale, et sans jeux répétés.
- Simulation 11 : Tester l'honnêteté des services d'information lors d'une simulation générale, et avec jeux répétés de 5 tours.
- Simulation 12 : Tester l'honnêteté des services d'information selon la probabilité  $P$ . Les services web sont bons, il y a un  $\pi$  de services tiers égal à 40 et proposé avec une probabilité de 0,5. Prise en compte des jeux répétés de 5 tours. Le paramètre  $P$  est fixé à 0,1.
- Simulation 13 : Tester l'honnêteté des services d'information selon la probabilité  $P$ . Les services web sont bons, il y a un  $\pi$  de services tiers égal à 40 et proposé avec une probabilité de 0,5. Prise en compte des jeux répétés de 5 tours. Le paramètre  $P$  est fixé à 0,01. Ces deux dernières simulations ont pour but de vérifier l'utilité de la probabilité  $P$ , introduite dans le chapitre 6.

### 8.3.3 Simulation 1

#### Paramètres

Les services web sont mauvais, ils ont une qualité de service variant entre 0,01 et 0,49. Ils sont tous malhonnêtes et proposent un  $\pi$  de 40. Il y a 10 services d'information avec une patience de 0,1 à 1 par pas de 0,1. Nous exécutons la simulation en la paramétrant à un tour, ce qui signifie que le choix de chaque service d'information sera déterminé par le résultat d'un jeu stratégique. Tous les autres paramètres sont laissés par défaut.

Le paiement maximal possible que l'on peut obtenir du maître ici est de 35. Avec un  $\pi$  de 40, les services d'information devraient théoriquement tous mentir.

#### Résultats généraux

Voici tout d'abord le graphique exposant le nombre de mensonges et le nombre de vérités lors de cette simulation. Comme prévu, étant donné la grandeur de  $\pi$  et le nombre de coups à l'avance fixé à 1, tous les services d'information mentent. Étant donné qu'il y avait 1 000 services web lors de cette simulation, et que pour chaque demande d'information, la communauté fait appel à 5 services d'information, il y a donc en tout 5 000 demandes d'information au total. Les services d'information ayant tous mentis, il y a eu au total 5 000 mensonges, ce qui explique les grandeurs de l'axe  $y$  du graphique.



FIGURE 8.1 – Évolution du nombre de mensonges/vérités pour la simulation 1

Étant donné tous ces mensonges, les maîtres de communauté ont accepté à tort tous les services web désirant entrer. La figure 8.2 montre que pour chaque communauté, la totalité des services web la composant ont été acceptés à tort.

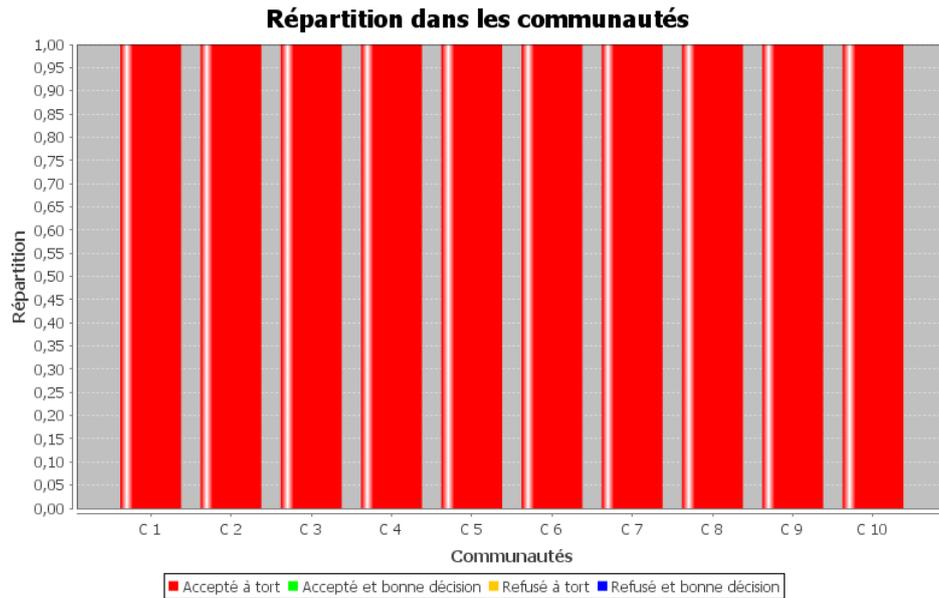


FIGURE 8.2 – Répartition dans les communautés pour la simulation 1. Tous les services web ont été acceptés à tort (ils étaient mauvais).

#### Service d'information 1

Nous voyons que le premier service d'information, le plus patient (facteur d'actualisation égal à 1), ment aussi, car le fait de réfléchir à très court terme empêche la patience de rentrer en jeu. Celle-ci entre en compte à partir du second coup. Nous pouvons voir sur la figure 8.3 que sur les presque 500 sollicitations du service d'information, il a toujours menti.

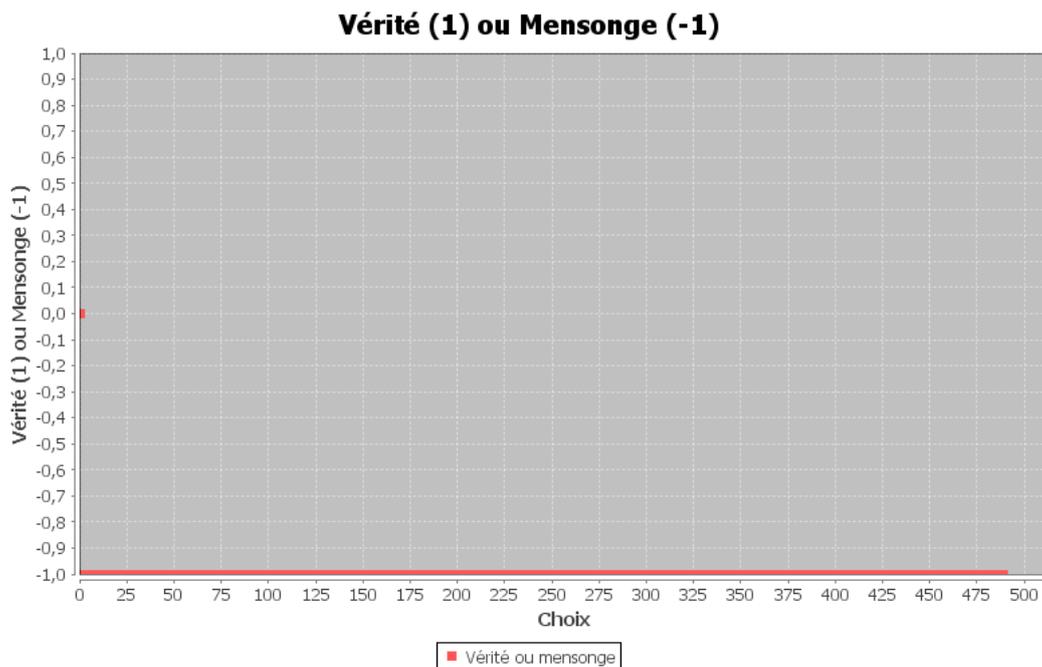


FIGURE 8.3 – Nombre de vérités/mensonges du service d'information le plus patient pour la simulation 1. On peut voir qu'il ment toujours.

La figure 8.4 montre les paiements cumulés par le service d'information. Nous voyons sur ce graphique que les paiements  $\pi$  compensent largement les autres paiements très faibles. Il reçoit un total de près de 20 000 en paiements  $\pi$  ce qui correspond bien à ses presque 500 mensonges multipliés par 40. Les paiements  $\alpha$  et  $\beta$  sont très faibles car, comme nous allons le voir sur la figure suivante, la réputation du service d'information est très basse. Vu la taille de  $\pi$ , il est logique, si le système d'information ne prend en compte ni sa patience, ni l'évolution de sa réputation à long terme, que celui-ci choisisse de mentir. Si le service d'information reçoit un si gros paiement  $\pi$ , c'est aussi parce qu'il continue de conseiller des maîtres de communauté alors qu'il ne devrait plus être choisi vu sa réputation médiocre. Cependant, les services d'information étant tous médiocres, le maître de communauté est obligé d'en choisir de tels, ou sinon, la simulation s'arrêterait prématurément. Il s'agit ici d'une limitation des simulations, où nous testons des cas limites (les services web sont tous mauvais et tous malhonnêtes), et où il n'existe que 10 services d'information au total (échelle réduite).

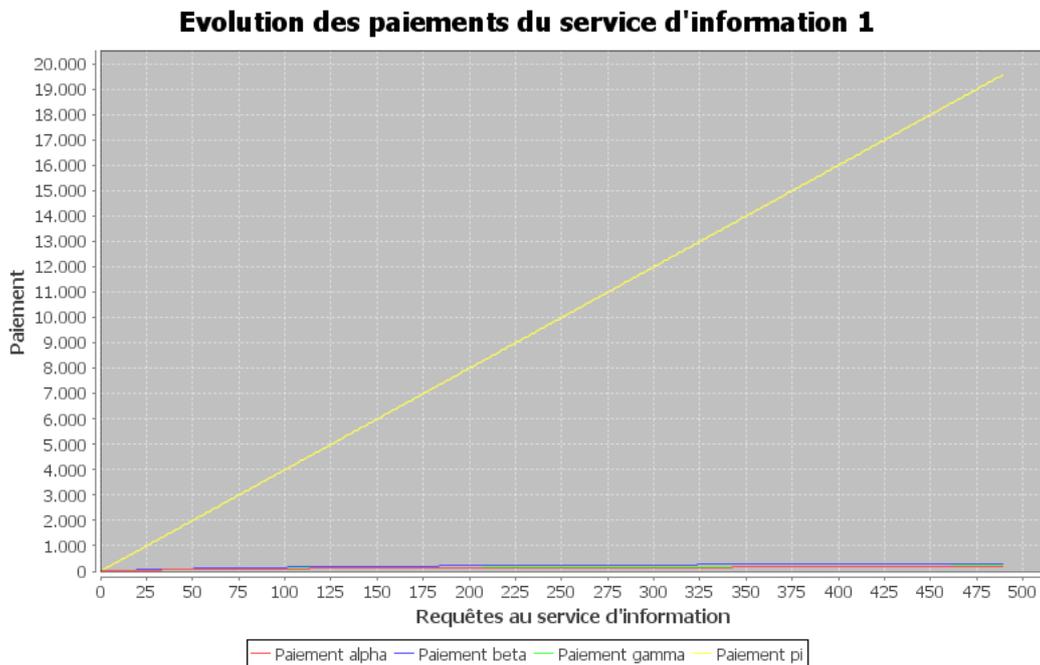


FIGURE 8.4 – Évolution des paiements du service d'information le plus patient pour la simulation 1.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

La figure 8.5 montre une baisse de réputation à chaque mensonge détecté. La communauté classe les services d'information par réputation, et lorsque la réputation d'un service d'information est supérieure au seuil de consultation, il est susceptible d'être consulté par une communauté. Mais lorsque qu'il n'existe pas assez de services d'information ayant une réputation supérieure au seuil, la communauté choisit au hasard dans les services d'information restant. Comme dit plus haut, il s'agit d'une limitation des simulations, due au faible nombre de services d'information. C'est donc pour cela que, malgré sa mauvaise réputation, le service d'information est repris.

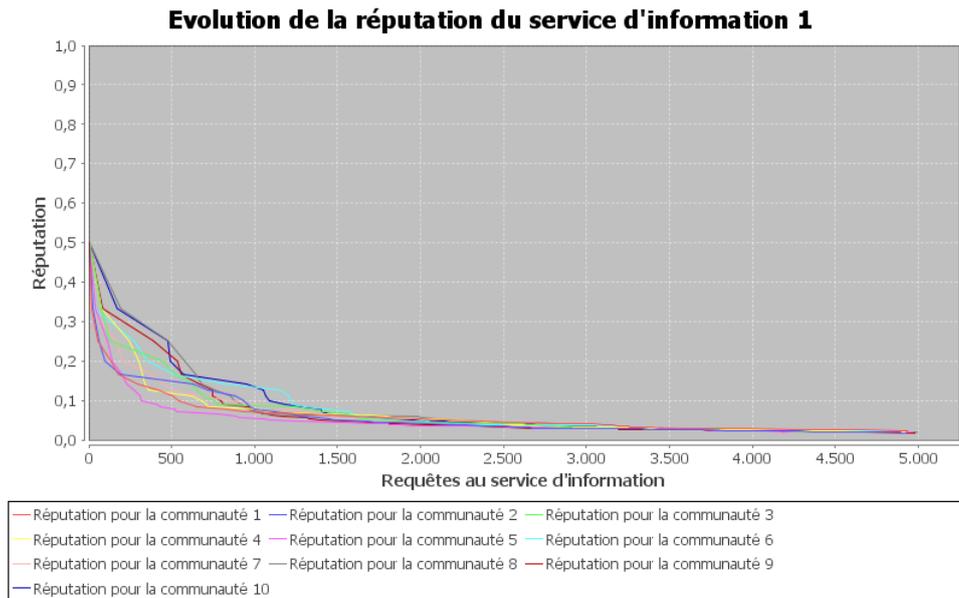


FIGURE 8.5 – Évolution de la réputation du service d'information le plus patient pour la simulation 1.

#### 8.3.4 Simulation 2

##### Paramètres

Les services web sont mauvais, ils ont une qualité de service variant entre 0,01 et 0,49. Ils sont tous malhonnêtes et proposent un  $\pi$  de 40. Il y a 10 services d'information avec une patience de 0,1 à 1 par pas de 0,1. Nous exécutons la simulation en la paramétrant à 8 tours, ce qui signifie que le choix de chaque service d'information sera déterminé par le résultat d'un jeu répété de longueur 8. Les mensonges sont petits. Tous les autres paramètres sont laissés par défaut. Il s'agit donc d'une simulation identique à la première mais où les services d'information agissent en pensant sur le long terme.

##### Résultats généraux

Dans ce cas-ci, avec 8 coups à l'avance, nous observons qu'il y a plus de vérités que de mensonges (voire figure 8.6). Le nombre de mensonges reste cependant assez conséquent (56% de vérités et 44% de mensonges). Les mensonges sont principalement émis par les services d'information non patients.



FIGURE 8.6 – Évolution du nombre de mensonges/vérités pour la simulation 2.

Les communautés acceptent beaucoup moins que dans la simulation précédente. Les résultats montrés par la figure 8.7 semblent cependant très bons lorsqu'on voit le taux de mensonges qui reste relativement élevé. D'autant plus qu'une partie des services web acceptés à tort l'ont été à cause de la probabilité  $P$  fixée à 0,1 par défaut. Cela est dû au fait que dans cette simulation, les mensonges sont petits (les services d'information menteurs transmettent des notations tout juste supérieures aux seuils d'acceptation des communautés). Les mensonges ont donc une influence plus réduite, apparemment trop pour tromper les communautés. Aussi, étant donné que les services d'information ayant une bonne réputation comptent plus lors du calcul de la réputation du service web, les communautés écoutent plus ceux-ci. En effet, lors du calcul de la moyenne de la réputation du service web à partir des notations données par les services d'information, ces notations sont pondérées par leur réputation respective. Un service d'information avec une réputation de 0,9 donnant un bon conseil sera beaucoup plus écouté qu'un service d'information ayant donné une réputation erronée de 0,3 et dont la propre réputation est de 0,1.

### Service d'information 1

Nous observons que le service d'information le plus patient (facteur d'actualisation de 1) est quasiment toujours honnête. Avec une réputation de départ de 0,5, il dit la vérité afin de faire monter sa réputation car si elle est trop basse, il ne sera pas repris. A long terme, malgré le fait que sa réputation est plus grande qu'au départ, et qu'il peut donc se « permettre » de mentir, il préférera la plupart du temps dire la vérité. Comme nous pouvons le voir sur la figure 8.8, le service répond à plus de 900 requêtes sur les 1 000 de la simulation.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

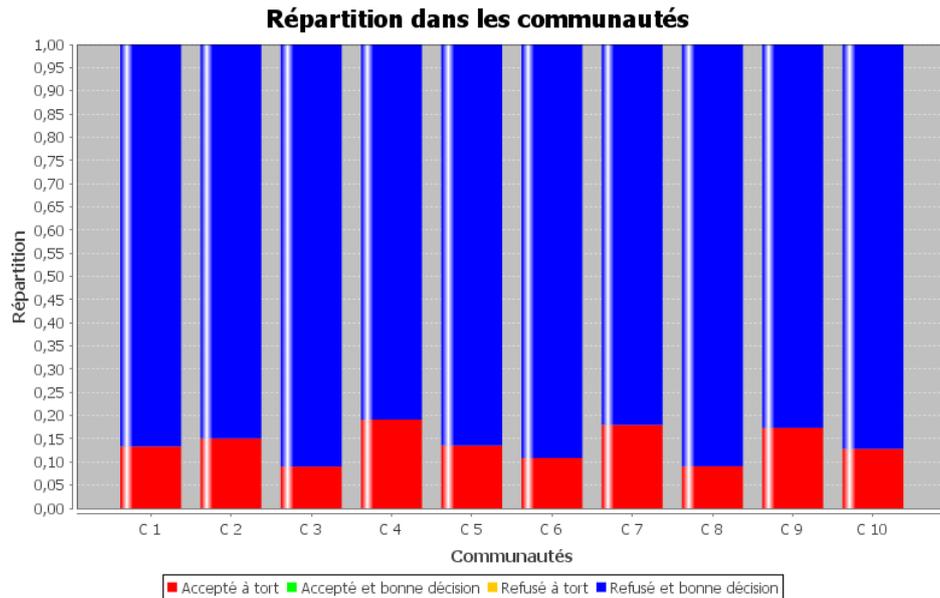


FIGURE 8.7 – Répartition dans les communautés pour la simulation 2. La plupart des services web ont été refusés à raison (ils étaient tous mauvais).

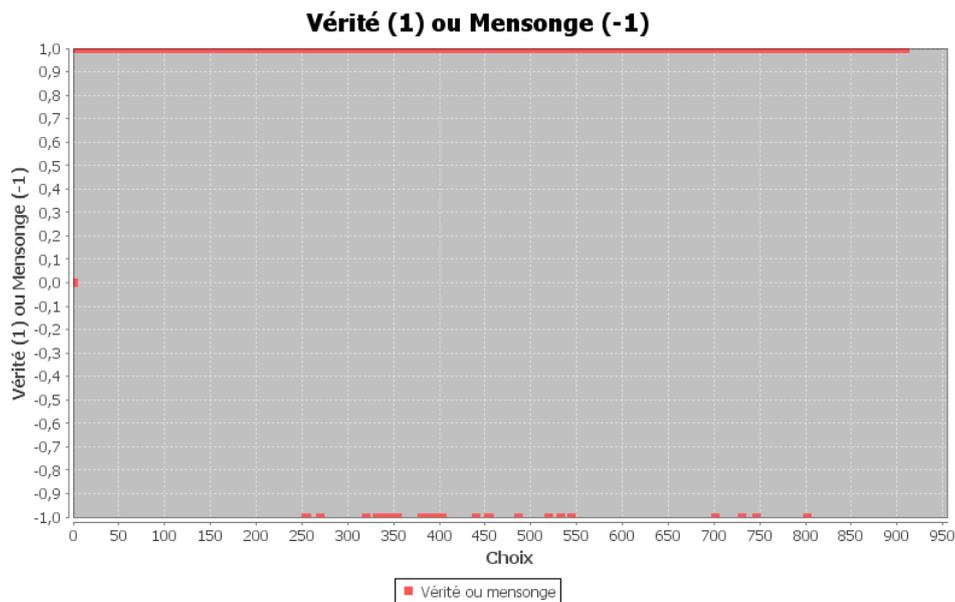


FIGURE 8.8 – Nombre de vérités/mensonges du service d'information le plus patient pour la simulation 2. On peut voir qu'il dit quasiment toujours la vérité.

Sur le graphique d'évolution des paiements (figure 8.9), nous voyons qu'ils augmentent de plus en plus, sans doute grâce à l'augmentation de la réputation du service d'information. Le paiement  $\pi$  ne suffit pas à long terme à surpasser les autres paiements. Ce service d'information reçoit un paiement total de près de 8750. Ce nombre n'est pas comparable avec les 20 000 reçus par le service d'information le plus patient de la première simulation, car comme expliqué, ce nombre énorme est dû à une limitation des simulations.

Sur la figure 8.10, nous observons que la réputation du service d'information augmente

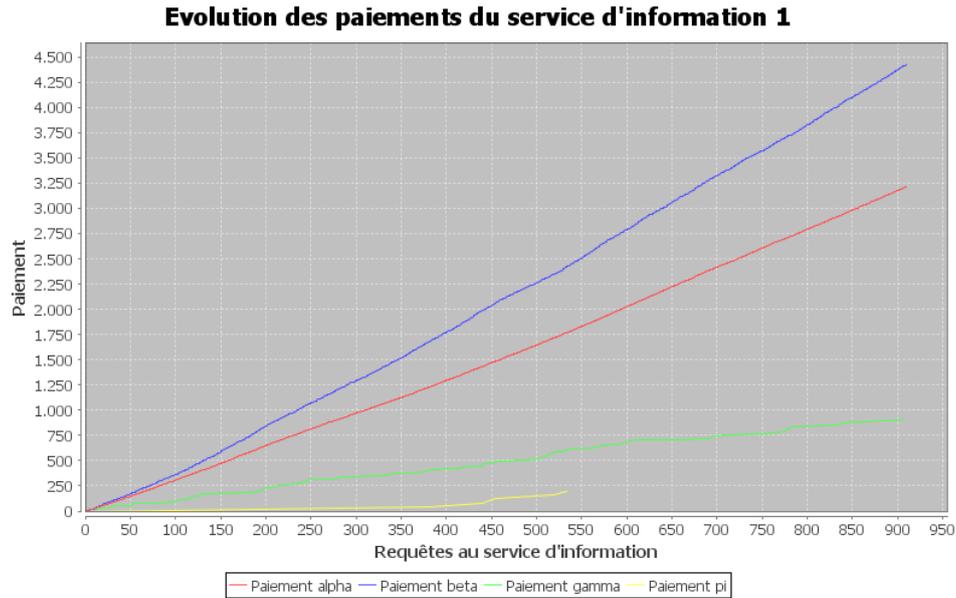


FIGURE 8.9 – Évolution des paiements du service d'information le plus patient pour la simulation 2.

globalement dans chaque communauté. On voit cependant que la réputation baisse dans certaines communautés, ce qui correspond aux moments où le service d'information a menti. Il a d'ailleurs eu une mauvaise réputation dans trois communautés, mais a fini par être repêché. A la fin de la simulation, le service d'information a une bonne réputation dans 9 des 10 communautés.

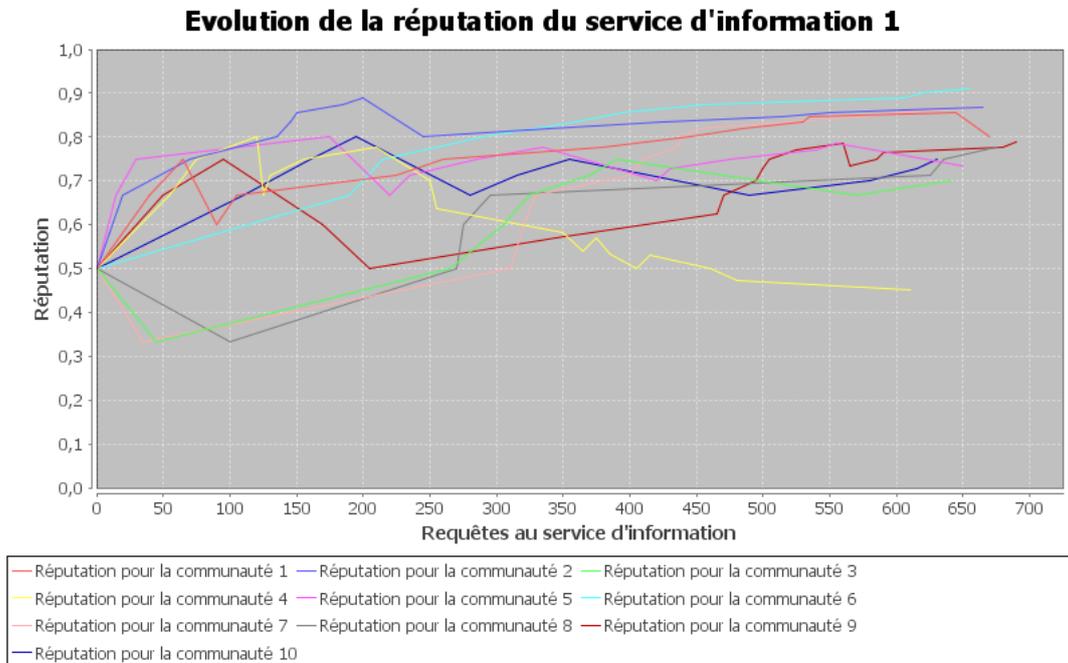


FIGURE 8.10 – Évolution de la réputation du service d'information le plus patient dans chaque communauté pour la simulation 2.

#### Service d'information 5

Ce service d'information a une patience de 0,5. Les résultats pour celui-ci sont plus mitigés. En effet, il ment bien plus souvent, comme nous pouvons le voir sur la figure 8.11. Nous pouvons aussi voir sur cette figure, que ce service d'information ne répondra qu'à 400 requêtes environ, soit deux fois moins que le premier service d'information. Cela est sans aucun doute dû à sa mauvaise réputation.

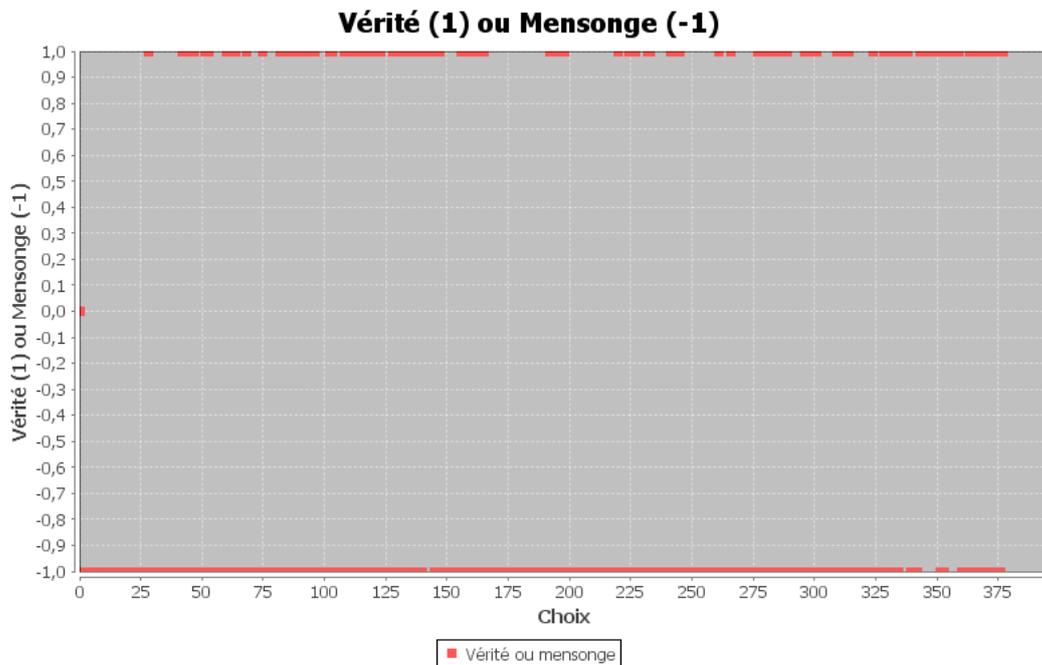


FIGURE 8.11 – Nombre de vérités/mensonges du service d'information 5 pour la simulation 2. On peut voir qu'il ment beaucoup plus que le premier, plus patient.

Sur le graphique des paiements cumulés, nous voyons que le paiement  $\pi$  total obtenu est plus élevé que les autres paiements. Cependant, ce service d'information reçoit un paiement total de seulement 3450 soit deux fois moins que le premier service d'information. La patience paie donc sur le long terme.

Sur la figure 8.13, nous observons que la réputation du service d'information est mauvaise dans toutes les communautés.

Nous pouvons donc voir que l'apport de l'étude des jeux répétés est positif car elle motive les services d'information les plus patients à dire la vérité.

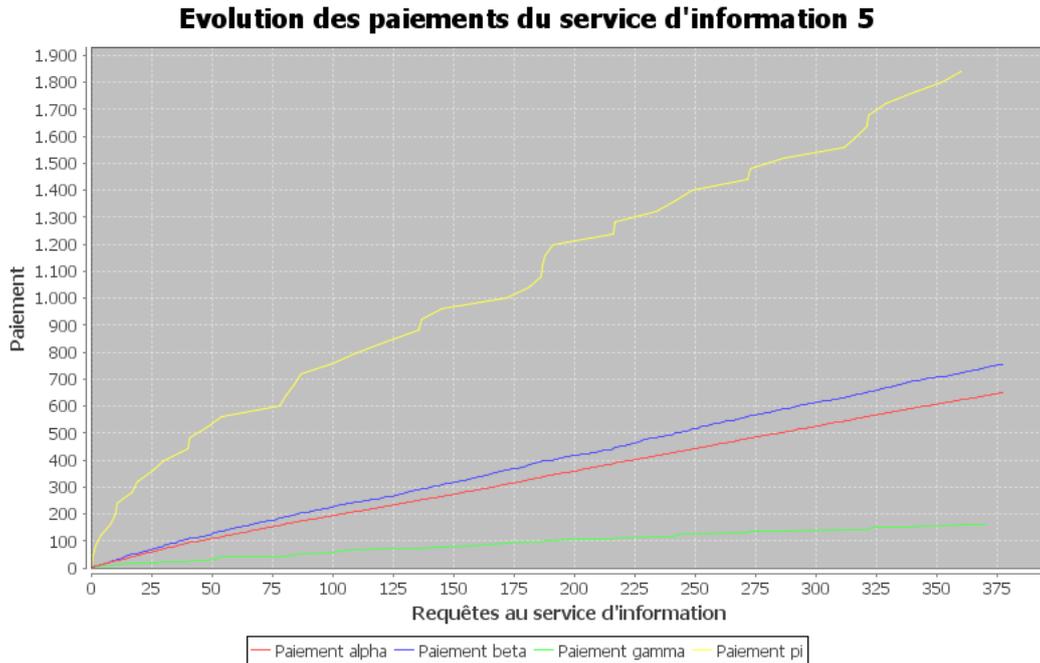


FIGURE 8.12 – Évolution des paiements du service d'information 5 pour la simulation 2. Au total, il gagne moins que le premier service d'information, plus patient.

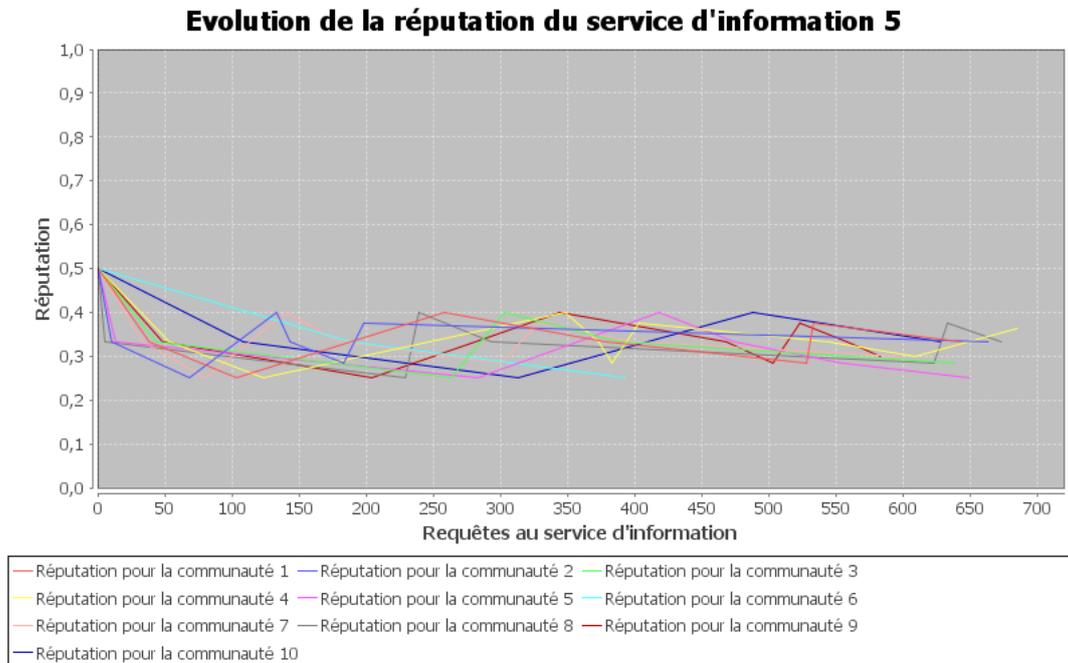


FIGURE 8.13 – Évolution de la réputation du service d'information 5 dans chaque communauté pour la simulation 2.

### 8.3.5 Simulation 3

#### Paramètres

Les services web sont mauvais, ils ont une qualité de service variant entre 0,01 et 0,49. Ils sont tous malhonnêtes et proposent un  $\pi$  de 40. Il y a 10 services d'information avec une patience de 0,1 à 1 par pas de 0,1. Nous exécutons la simulation en la paramétrant à 8 tours, ce qui signifie que le choix de chaque service d'information sera déterminé par le résultat d'un jeu répété de longueur 8. Les mensonges sont moyens. Tous les autres paramètres sont laissés par défaut. Il s'agit donc d'une simulation identique à la précédente (simulation 2) mais où les services d'information malhonnêtes disent de plus « gros » mensonges.

#### Résultats généraux

Le taux de mensonge est de 40%, soit un taux similaire à la simulation précédente. En effet, le paramètre que nous avons changé par rapport à la simulation précédente n'influe pas sur les incitants à mentir ou dire la vérité, mais uniquement sur le mensonge qui sera dit lorsqu'un service d'information choisit de mentir.

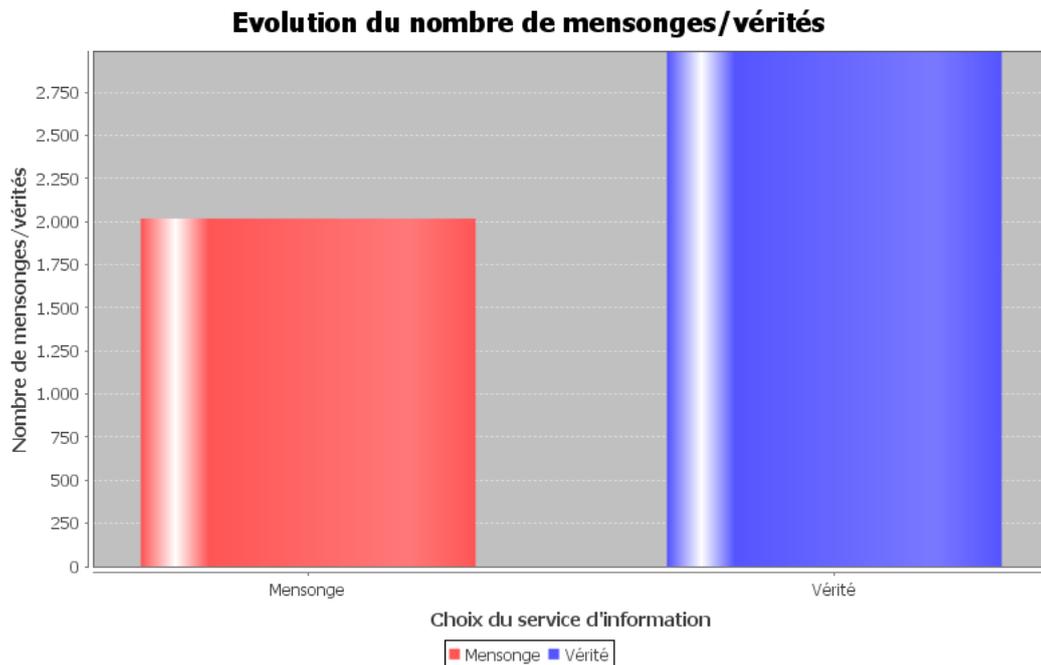


FIGURE 8.14 – Évolution du nombre de mensonges/vérités pour la simulation 3. Il est similaire à la simulation 2.

Ce qui change par contre, c'est le nombre de services web acceptés à tort, beaucoup plus élevé. En effet, les mensonges étant plus conséquents, ils influent plus dans la moyenne pondérée des notations reçues, calculée par la communauté. Les services d'information ayant une meilleure réputation que les autres mentent aussi de temps en temps, ce qui fait pencher la balance.

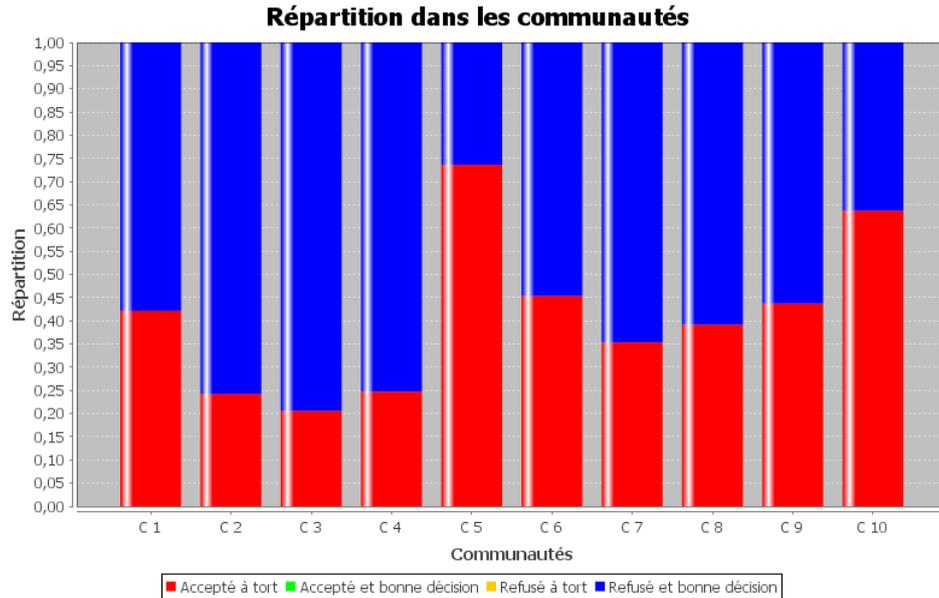


FIGURE 8.15 – Répartition dans les communautés pour la simulation 3. Beaucoup de services web ont été acceptés à tort (ils étaient tous mauvais).

### Service d'information 5 (et 1)

Le service d'information 5 a une patience de 0,5, et le service d'information 1 a une patience de 1. Nous allons comparer les données sur le service d'information 5 avec les données du même service lors de la simulation précédente (simulation 2). On peut observer sur la figure 8.16 que le service d'information répond à un peu moins de 400 requêtes et ment le plus souvent. Il a un comportement similaire au service d'information de même patience de la simulation précédente.

Sur le graphique des paiements cumulés (figure 8.17), nous voyons que le paiement  $\pi$  total obtenu est plus élevé que le paiement  $\pi$  total obtenu par le même service d'information lors de la simulation précédente (4500 au lieu de 1850). Il reçoit un paiement total d'environ 6000. Cela reste plus faible que le service d'information le plus patient qui obtient, dans cette simulation, un total de 12500, soit plus du double (figure 8.18). Si les services d'information reçoivent plus, c'est parce que leurs mensonges « fonctionnent » mieux, comme on a pu le voir sur la figure 8.15. Les services d'information ayant menti ne recevant le paiement  $\pi$  que si le mensonge fonctionne, ils reçoivent un nombre de paiements  $\pi$  plus élevés.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

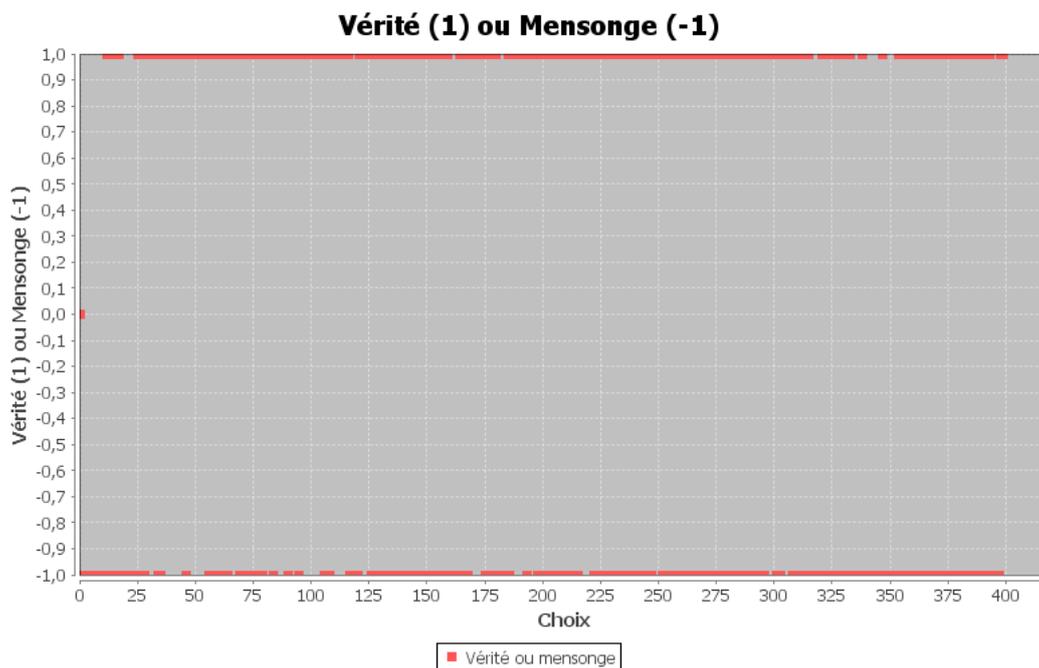


FIGURE 8.16 – Nombre de vérités/mensonges du service d'information 5 pour la simulation 3.

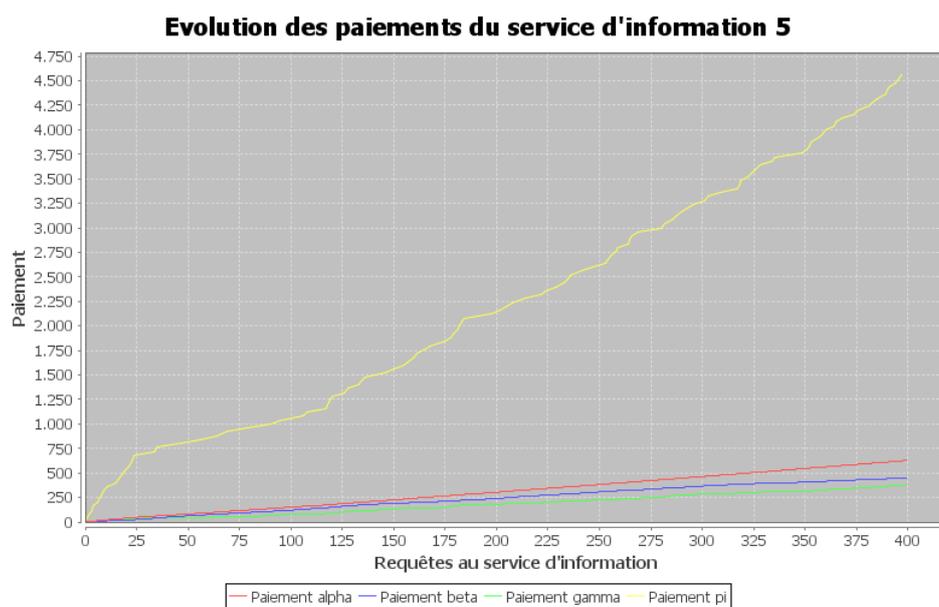


FIGURE 8.17 – Évolution des paiements du service d'information 5 pour la simulation 3.

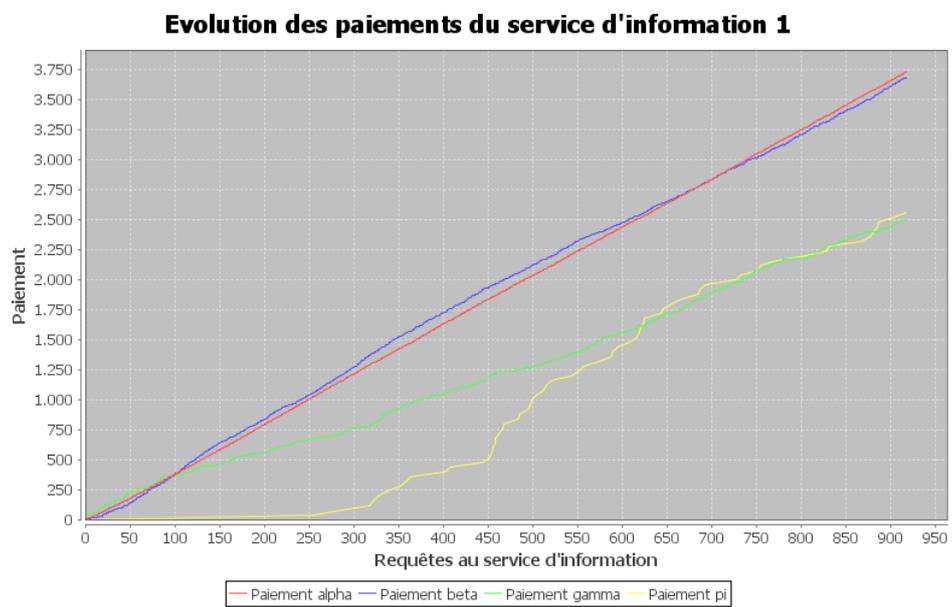


FIGURE 8.18 – Évolution des paiements du service d'information 1 pour la simulation 3.

### 8.3.6 Simulation 4

#### Paramètres

Les services web sont mauvais, ils ont une qualité de service variant entre 0,01 et 0,49. Ils sont tous malhonnêtes et proposent un  $\pi$  de 40. Il y a 10 services d'information, chacun avec une patience de 1. Nous exécutons la simulation en la paramétrant à 8 tours, ce qui signifie que le choix de chaque service d'information sera déterminé par le résultat d'un jeu répété de longueur 8. Tous les autres paramètres sont laissés par défaut. Il s'agit donc d'une simulation identique à la simulation 2 mais où tous les services d'information sont patients, c'est à dire qu'ils ont tous un facteur de patience de 1.

#### Résultats généraux

Comme on peut le voir sur le graphique suivant (figure 8.19), le nombre de mensonges est très proche de 0.

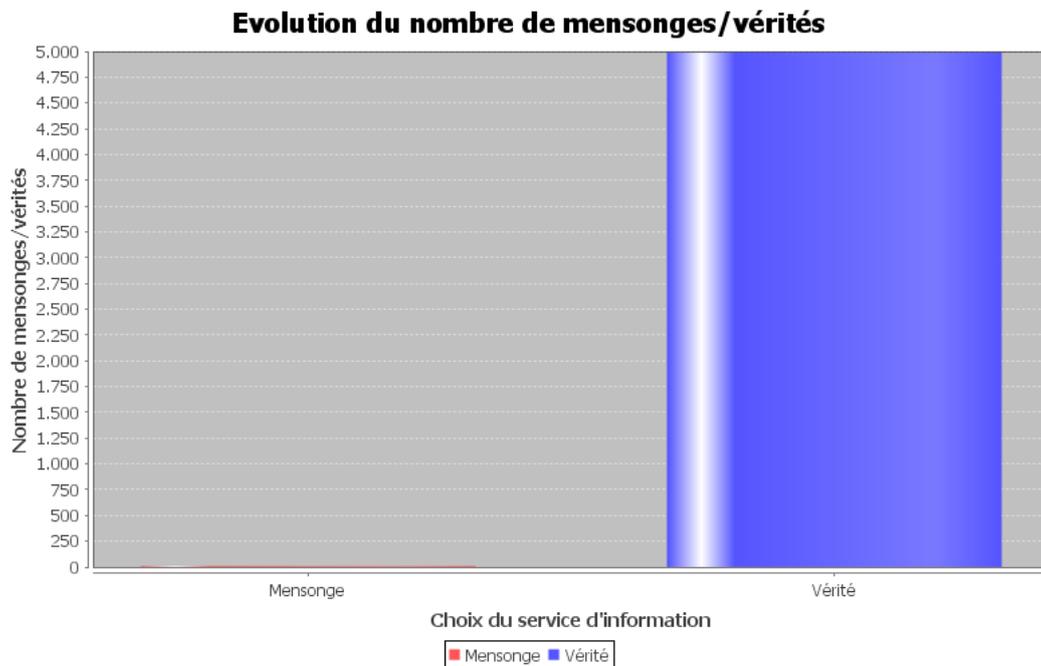


FIGURE 8.19 – Évolution du nombre de mensonges/vérités pour la simulation 4. Il y a très peu de mensonges

### 8.3.7 Simulation 5

#### Paramètres

Les services web sont tous mauvais (et donc leur qualité de service varie entre 0,01 et 0,49), et proposent un  $\pi$  de 0. Les services d'information utilisent des jeux stratégiques.

Le but de cette simulation est de voir si, comme nous l'avons souligné dans la section 5.7, les services d'information pourraient être incités à mentir pour obtenir un paiement  $\gamma^-$  qui peut être positif.

#### Résultats généraux

Nous pouvons voir sur le graphique représentant l'évolution du nombre de mensonges et de vérités, qu'il y a un taux assez important de mensonge (environ 40%). Cela est dû au fait que les services d'information ne prennent pas en compte le futur en calculant plusieurs coups à l'avance. Le  $\gamma^-$  reçu est intéressant à court terme car positif (les mensonges sont petits), mais cela est sans compter qu'il ne pourra que diminuer au fil des interactions, à cause de la diminution de la réputation du service d'information.

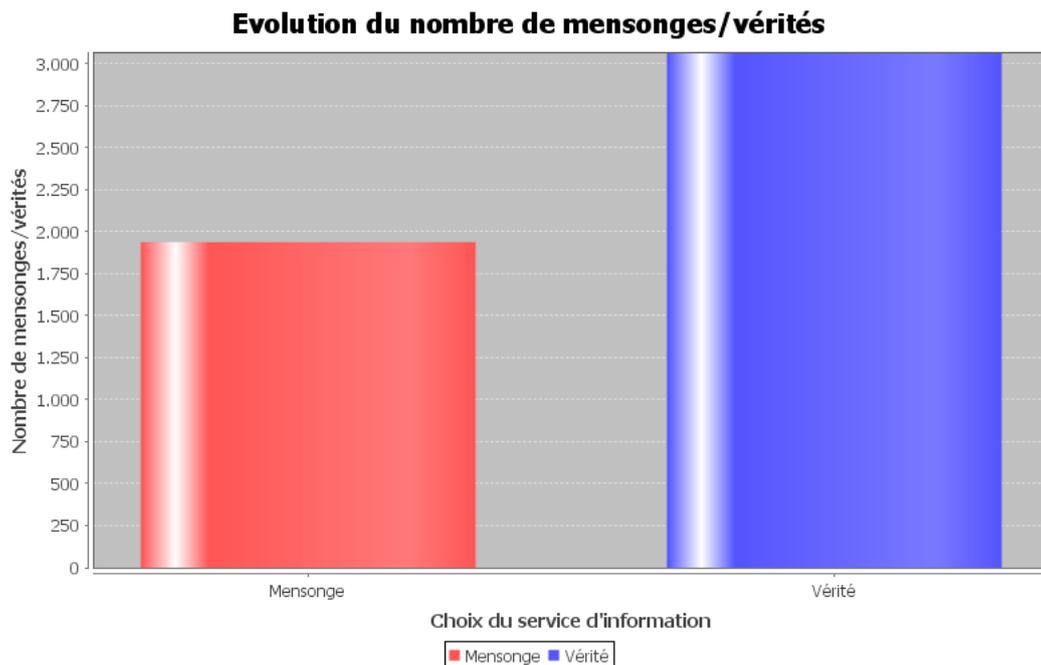


FIGURE 8.20 – Nombre de mensonges/vérités pour la simulation 5.

En choisissant de refaire la même simulation mais en utilisant les jeux répétés, on obtient le résultat montré à la figure 8.21, avec seulement 2 tours, et des services d'information avec une patience de 0,6 à 1.

#### Service d'information 1

Nous exposons ici les graphiques du service d'information 1, ayant, comme tous les autres, un facteur de patience égal à 1. Il ment beaucoup plus souvent qu'il ne dit la vérité.

### 8.3 Résultats et analyse de cas spécifiques et pertinents



FIGURE 8.21 – Nombre de mensonges/vérités pour la simulation 5 bis. Les services d'information pensent un peu plus à long terme.

Cela est dû au fait qu'il ne réfléchit pas à long terme (pas d'utilisation des jeux répétés).

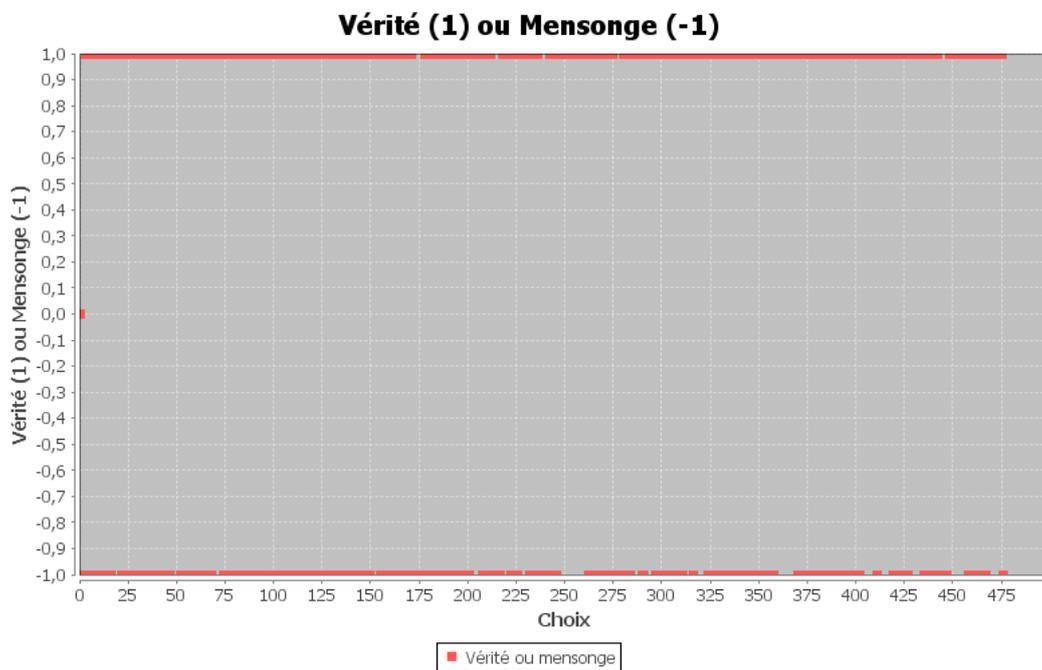


FIGURE 8.22 – Nombre de vérités/mensonges du service d'information 1 pour la simulation 5.

Comme nous pouvions nous y attendre, la réputation du service d'information chute beaucoup à cause de ses mensonges (figure 8.23).

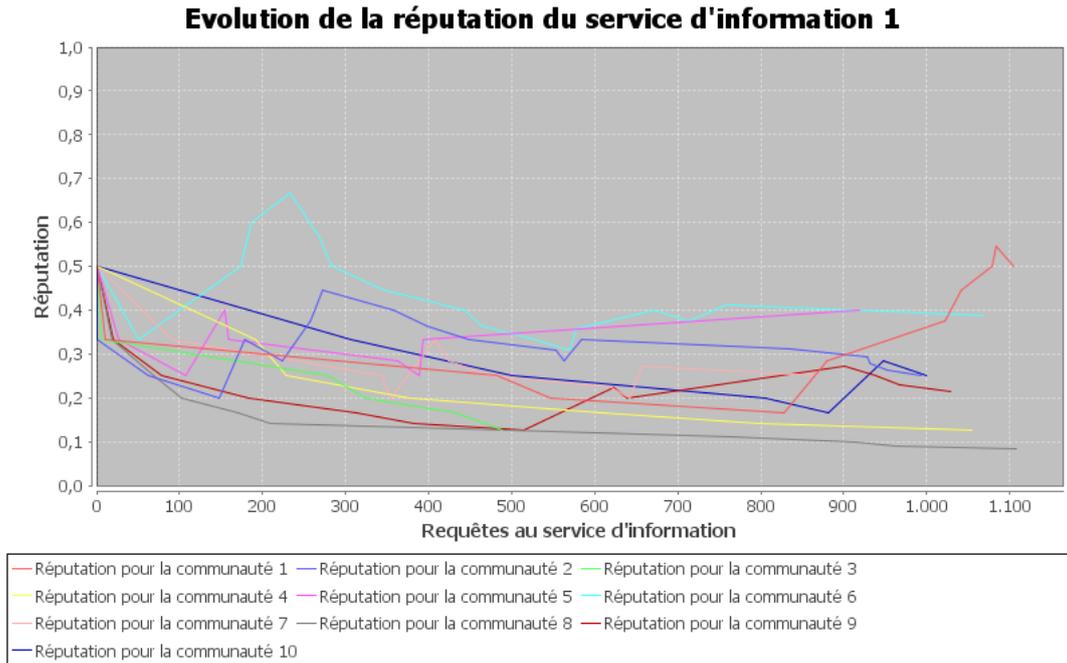


FIGURE 8.23 – Évolution de la réputation du service d'information le plus patient pour la simulation 5

Le service d'information bénéficie d'un paiement  $\beta$  pas trop bas, malgré le fait que sa réputation diminue (on le voit sur le graphique car la courbe croît de moins en moins vite), car il évalue le service web entrant de la même façon que les autres (ils mentent tous). Son paiement  $\gamma$  est moindre que le paiement  $\beta$ , ce qui ne devrait pas être le cas en général, mais il s'agit d'un cas particulier où le  $\gamma$  est donné suite à un mauvais service web accepté à tort. Nous voyons que, étant positif, le service d'information gagne plus à court terme en mentant et en ayant un  $\gamma$ , même faible. S'il avait pris en compte sa réputation à long terme, il aurait sans doute opté pour une meilleure réputation et un  $\beta$  plus conséquent à long terme.

Sur le graphique des paiements, nous voyons que les paiements gamma opposés sont parfois supérieurs aux paiements effectivement reçus. Pourquoi dès lors le service d'information a-t-il choisi cette action? Car lors de son calcul du coup à jouer, il avait pris en compte l'action des autres. Ici, le paiement gamma opposé correspond au paiement qu'il aurait reçu s'il avait changé d'avis mais pas les autres. Lors de sa décision, il ne pouvait néanmoins pas forcer la décision des autres et gagnait sûrement plus en choisissant l'action effectuée.

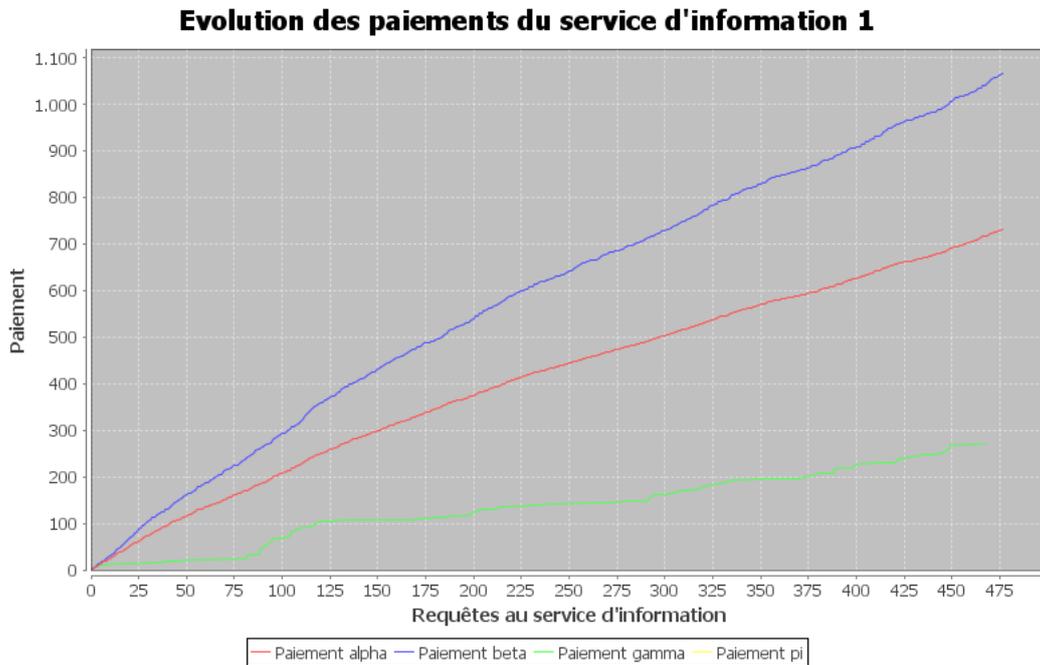


FIGURE 8.24 – Évolution des paiements reçus par le service d'information le plus patient pour la simulation 5.

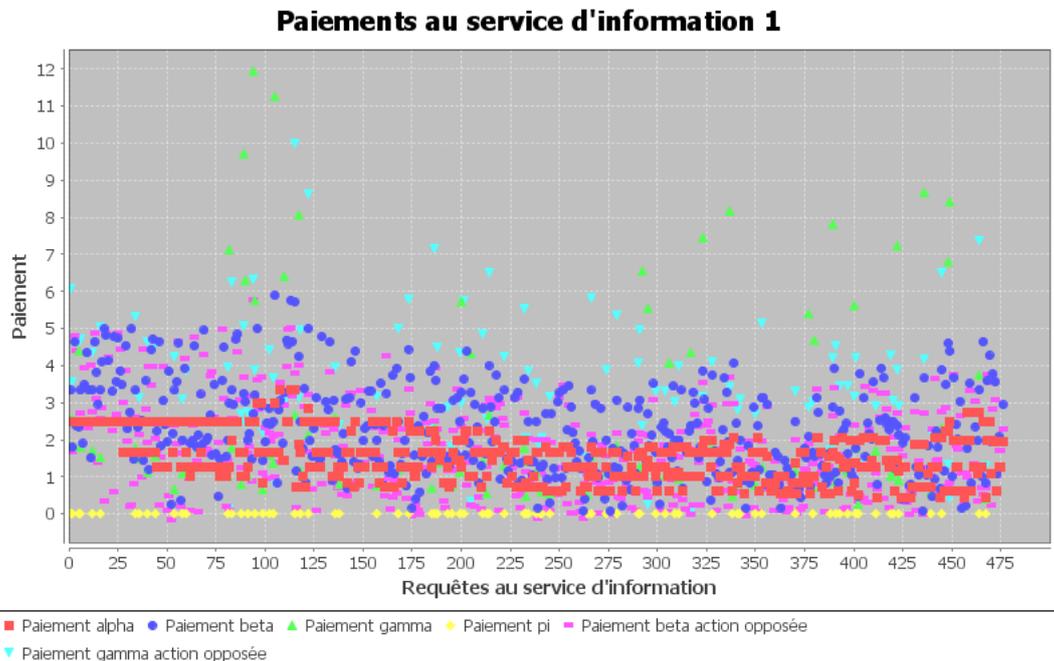


FIGURE 8.25 – Graphique des paiements reçus par le service d'information le plus patient à chaque tout pour la simulation 5.

### 8.3.8 Simulation 6

#### Paramètres

Dans cette simulation, les services web ont une qualité de service comprise entre 0,01 et 0,49 , et proposent un  $\pi$  de 0. Il n'y a toujours pas de prise en compte du jeu répété par les services d'information. Les paiements sont différents des simulations précédentes : Le paiement alpha est maintenant de 2 et les paiements négatifs sont plus élevés : le paiement  $\beta$  négatif est de 15 et le  $\gamma$  négatif est de 30.

#### Résultats généraux

Nous voyons que dans cette simulation, il y a près de deux fois moins de mensonges que dans la simulation précédente. Les paiements plus facilement négatifs influencent donc les services d'information à dire la vérité.

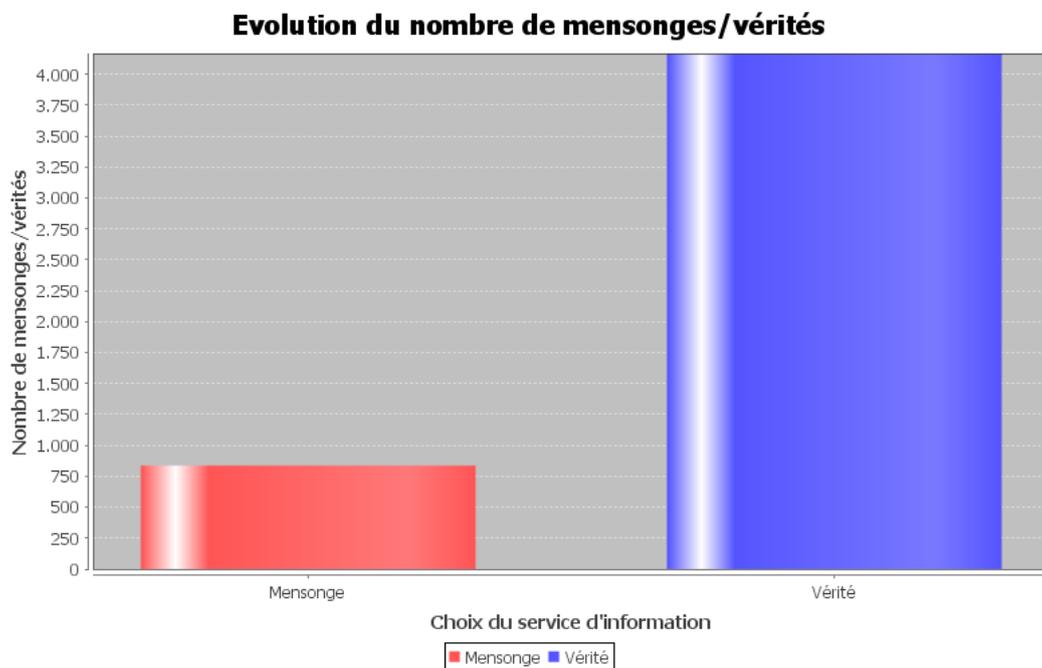


FIGURE 8.26 – Nombre de mensonges/vérités pour la simulation 6.

#### Service d'information 1

Le service d'information 1 est, comme nous pouvons le voir sur le graphique de vérité ou mensonge, beaucoup plus honnête que celui de la simulation précédente.

Étant donné qu'il ment quand même de temps en temps, sa réputation augmente puis diminue continuellement, mais n'est pas aussi catastrophique que lorsqu'il était moins honnête, bien qu'elle ne soit presque jamais au-dessus du seuil d'acceptation de la part d'une communauté.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

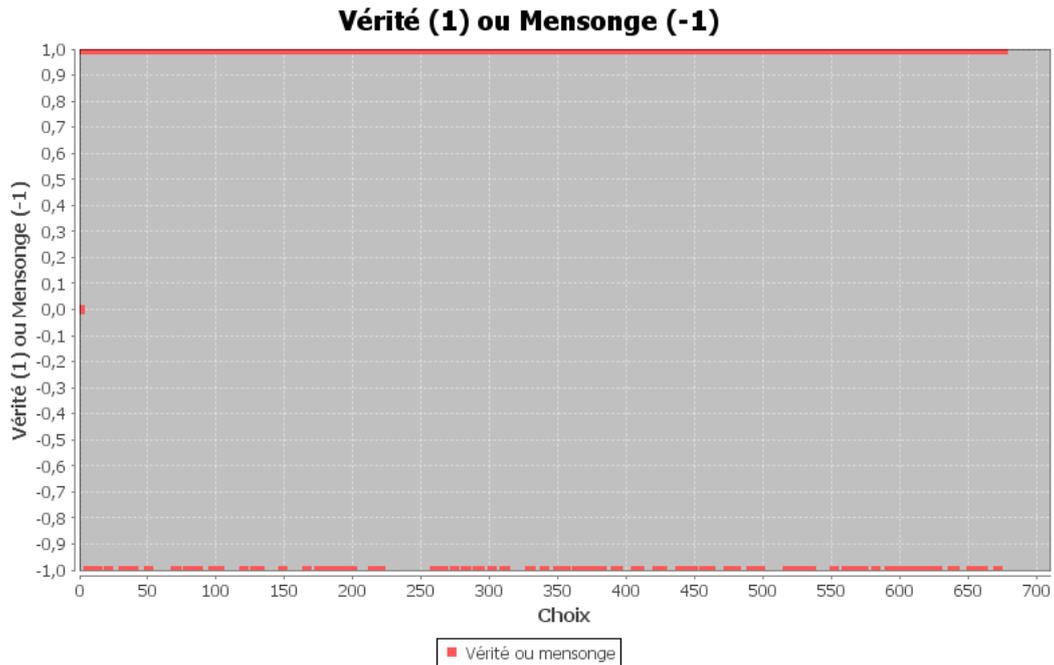


FIGURE 8.27 – Nombre de mensonges/vérités du service d'information le plus patient pour la simulation 6.

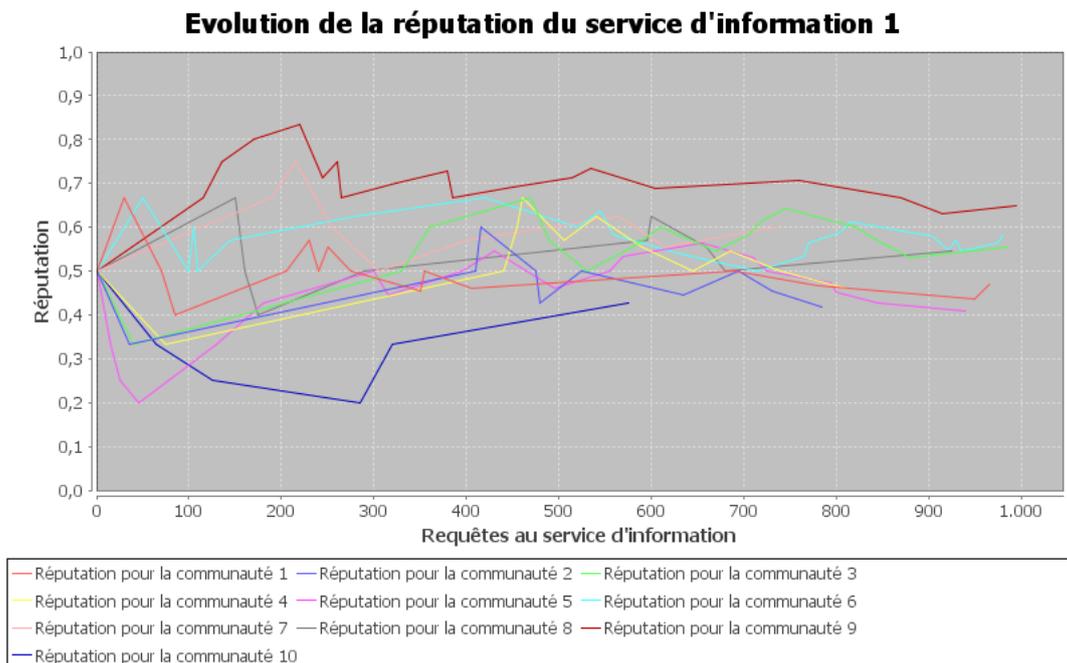


FIGURE 8.28 – Évolution de la réputation du service d'information le plus patient pour la simulation 6.

Sur le graphique de l'évolution des paiements du service d'information, le paiement  $\gamma$  est beaucoup moins élevé que précédemment, cela est dû aux mensonges plus rares, le paiement  $\gamma$  est moins souvent donné car moins de mauvais services web sont acceptés.

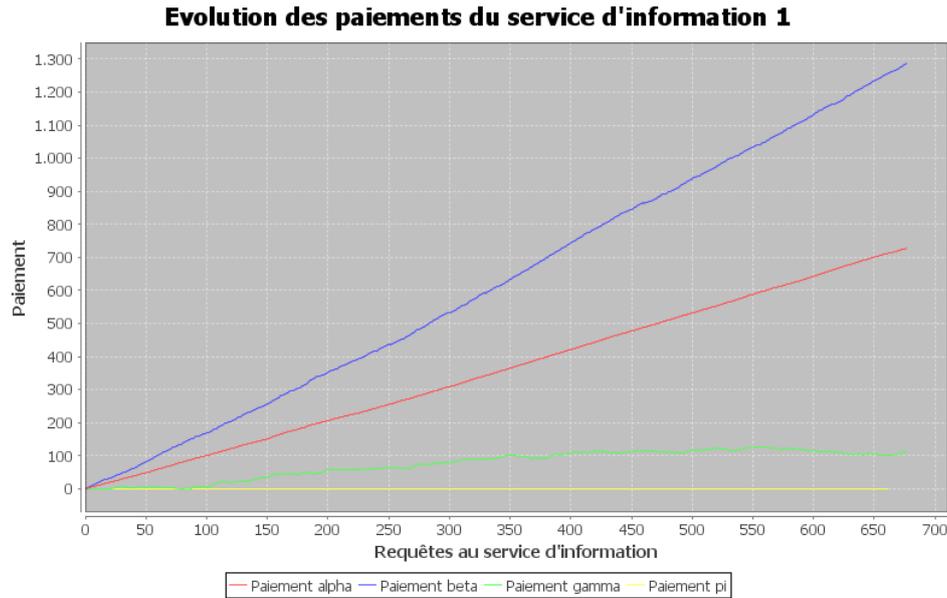


FIGURE 8.29 – Évolution des paiements du service d'information le plus patient pour la simulation 6.

Le graphique des paiements nous montre que le service d'information aurait eu des punitions avec le paiement  $\gamma$  s'il avait choisi l'action opposée, ce paiement étant plus punitif que précédemment. Le paiement  $\beta$  l'étant aussi, le service d'information n'a pas intérêt à diverger de l'opinion générale (allant vers la vérité), qui, heureusement dans ce cas, penche plus du côté de la vérité. Nous voyons donc l'intérêt de bien paramétrer les paiements. En effet, nous n'avons pas eu besoin d'augmenter les paiements donnés par les communautés, mais de modifier légèrement les punitions possibles, afin qu'elles ne soient pas trop négatives mais décourageant néanmoins les menteurs.

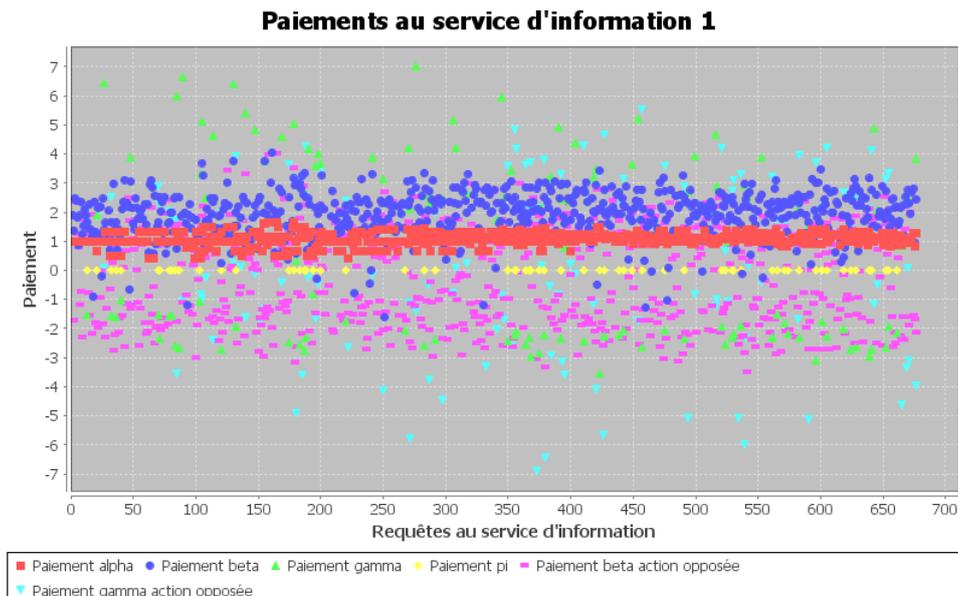


FIGURE 8.30 – Graphique des paiements du service d'information le plus patient pour la simulation 6.

### 8.3.9 Simulation 7

#### Paramètres

Les services web sont à présent bons, ils disposent d'une qualité de service comprise entre 0,5 et 0,99. Les services d'information utilisent des jeux stratégiques. Il n'y a pas de paiement  $\pi$  d'un service tiers voulant empêcher l'acceptation des services web entrants.

Théoriquement, tous les services d'information devraient dire la vérité. Cette simulation a pour but de le vérifier.

#### Résultats généraux

Comme nous nous y attendions, les services d'information disent tous la vérité, ils n'ont pas d'intérêt à mentir étant donné que le  $\gamma$  sera positif à coup sûr, les services web étant bons, et vu qu'il n'y a pas de paiement de tiers.



FIGURE 8.31 – Nombre de mensonges/vérités pour la simulation 7.

Étant donné la trivialité de cette simulation, nous n'exposons pas d'autres graphiques. Le but était juste de montrer que la simulation concorde avec notre analyse théorique du chapitre 6.

### 8.3.10 Simulation 8

#### Paramètres

Dans cette nouvelle simulation, les paramètres de la simulation précédente sont conservés, mis à part qu'il y a maintenant un paiement  $\pi$  de services tiers de 40, donné avec une probabilité de 0,5. Le jeu répété est maintenant pris en compte par les services d'information qui calculent 4 coups à l'avance. Il y a 10 services d'information avec une patience allant de 0,6 à 1.

#### Résultats généraux

La prise en compte de 4 coups à l'avance empêche une trop grosse majorité de mensonges, bien que nous pouvons voir qu'il y en a tout de même beaucoup.

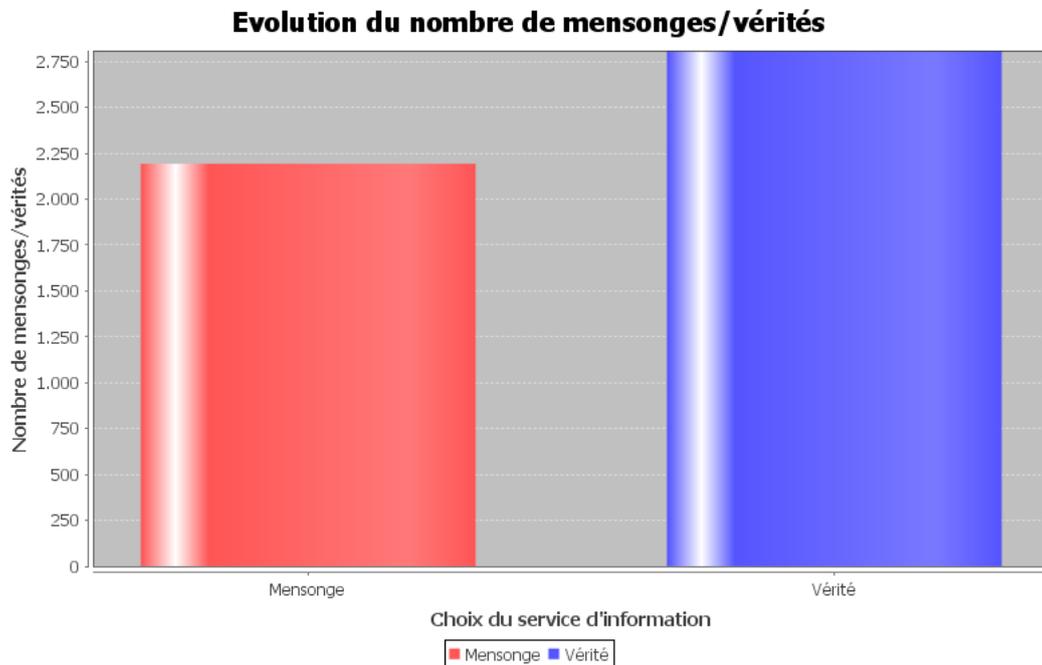


FIGURE 8.32 – Nombre de mensonges/vérités pour la simulation 8.

Quasiment tous les services web ont été acceptés, ce qui est le résultat attendu. Bien que le nombre de mensonges soit élevé, une majorité de vérités suffit à conseiller de manière efficace les communautés. N'oublions pas que les mensonges sont petits ici, si les mensonges étaient plus gros, il y aurait plus de bons services web refusés. La figure 8.34 montre le graphique de répartition dans les communautés pour une simulation similaire mais avec des mensonges moyens. De plus les services d'information ayant une bonne réputation continue à faire pencher la balance grâce à leurs conseils corrects dans la plupart des cas.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

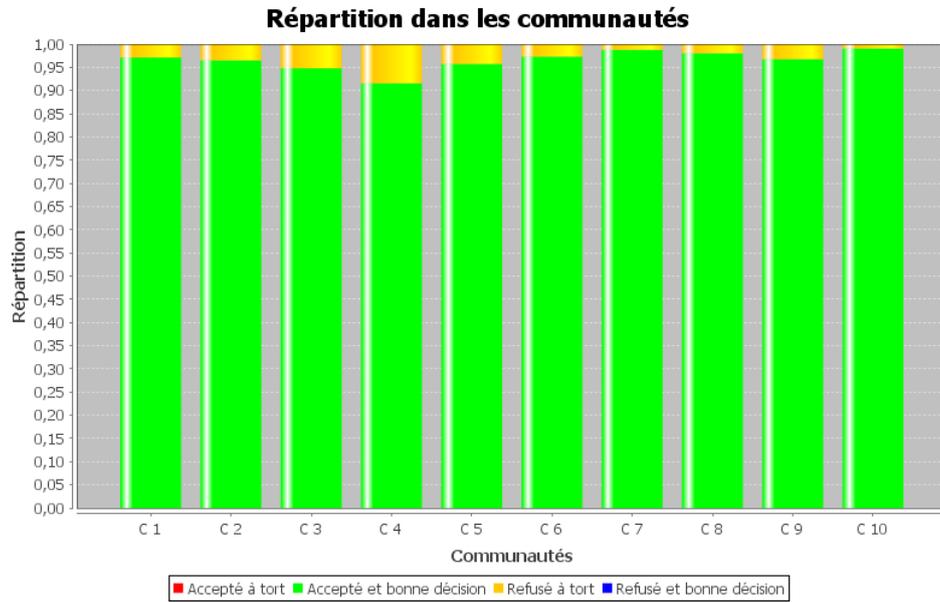


FIGURE 8.33 – Répartition des services web dans les communautés pour la simulation 8.

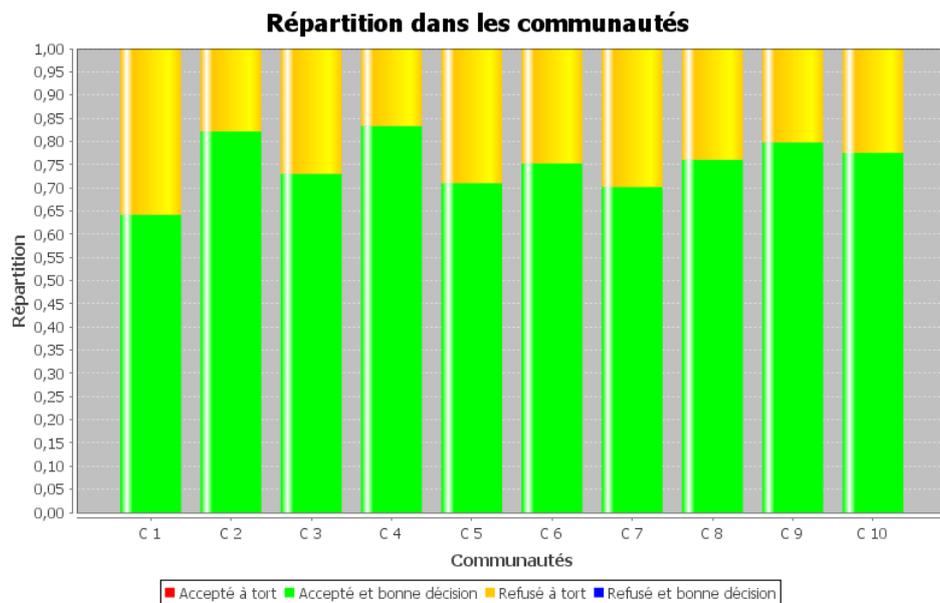


FIGURE 8.34 – Répartition des services web dans les communautés lorsque les mensonges sont plus conséquents.

## Service d'information 1

Ce service d'information est le plus patient (1), et dit majoritairement la vérité.

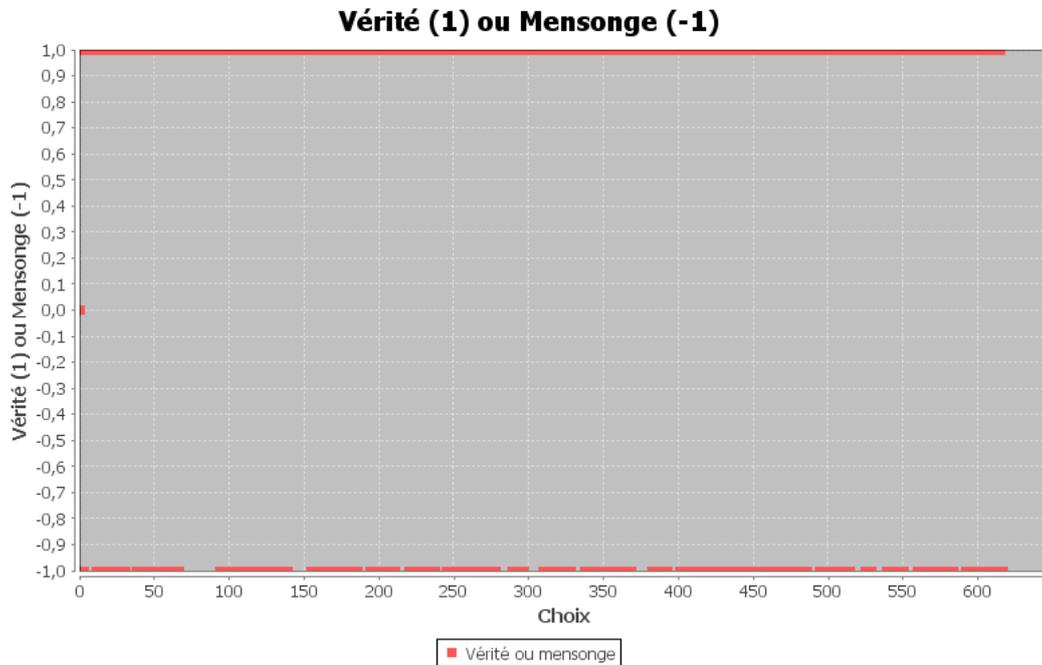


FIGURE 8.35 – Nombre de mensonges/vérités du premier service d'information (patience = 1) pour la simulation 8.

Sa réputation augmente de façon correspondante vis-à-vis de son honnêteté, bien que pour certaines communautés elle diminue au début pour stagner à 0,32.

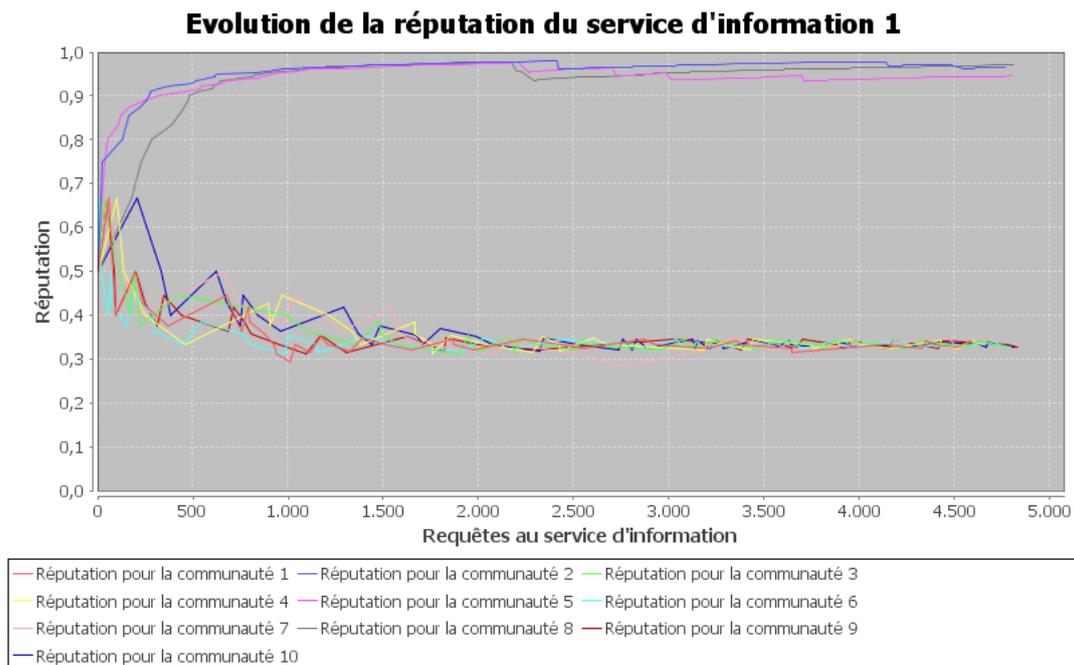


FIGURE 8.36 – Évolution de la réputation du premier service d'information pour la simulation 8.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

Nous observons que le paiement  $\gamma$  à long terme est bien plus élevé que le paiement  $\pi$ , étant donné que le service d'information ne ment pas souvent.

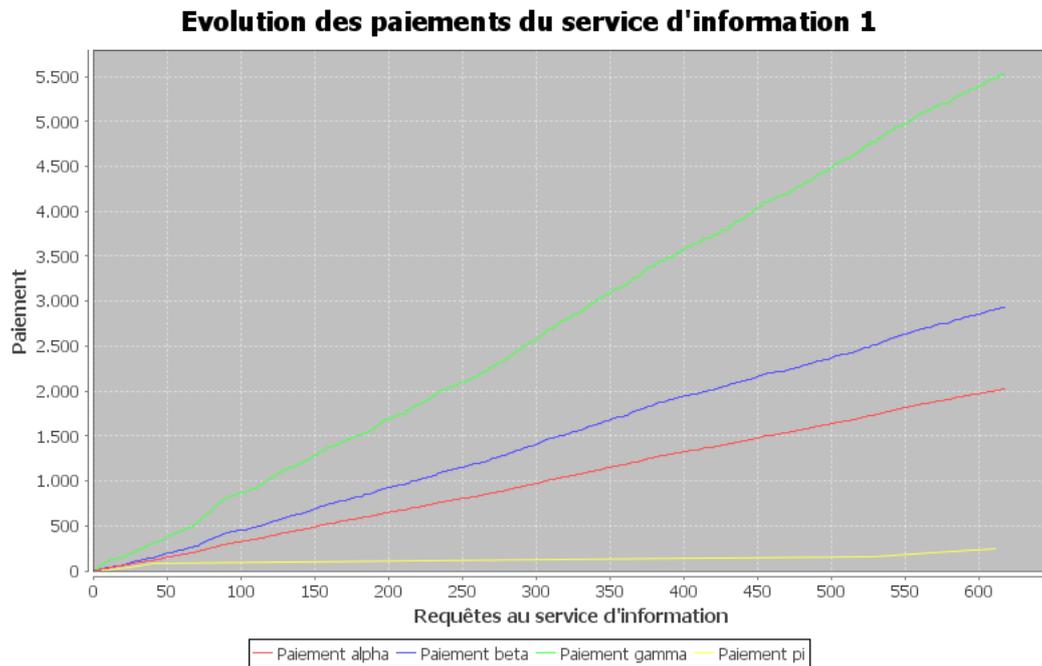


FIGURE 8.37 – Cumulation des différents paiements accordés au premier service d'information pour la simulation 8.

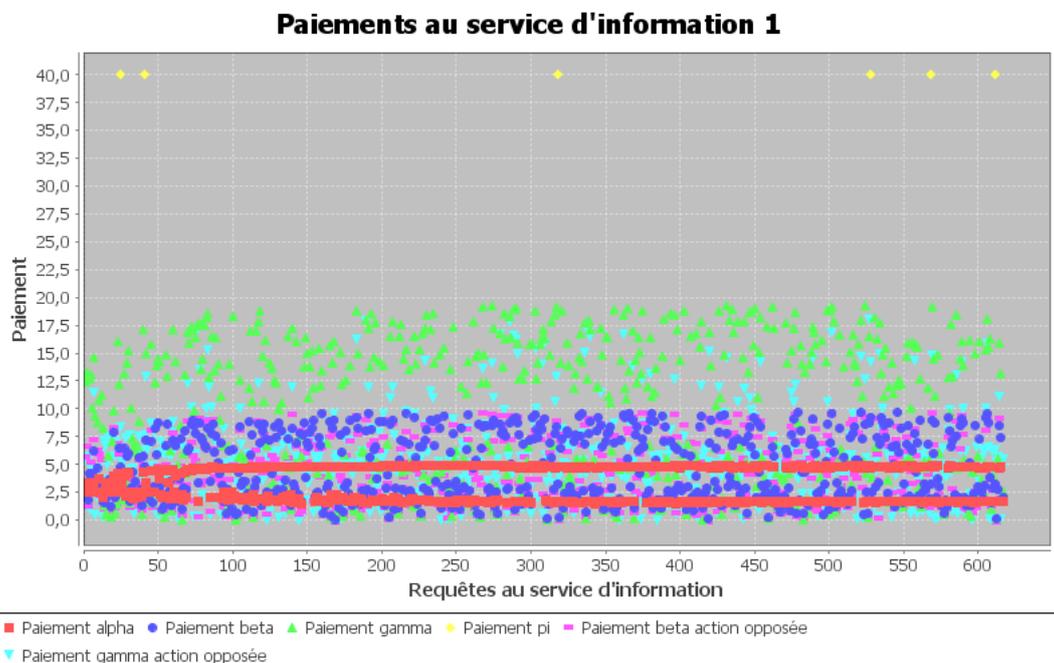


FIGURE 8.38 – Graphique des paiements accordés au premier service d'information pour la simulation 8.

## Service d'information 2

Ce service d'information est aussi que le précédent, mais dit encore plus la vérité.

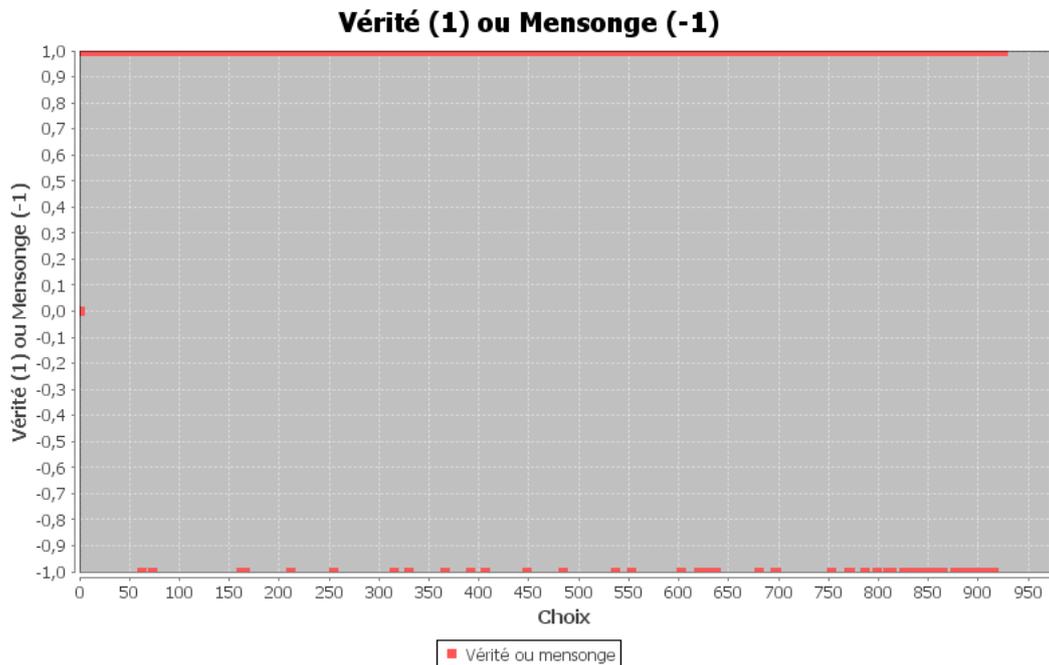


FIGURE 8.39 – Nombre de mensonges/vérités du deuxième service d'information (patience = 1) pour la simulation 8.

Sa réputation augmente donc dans toutes les communautés, sauf la 8.

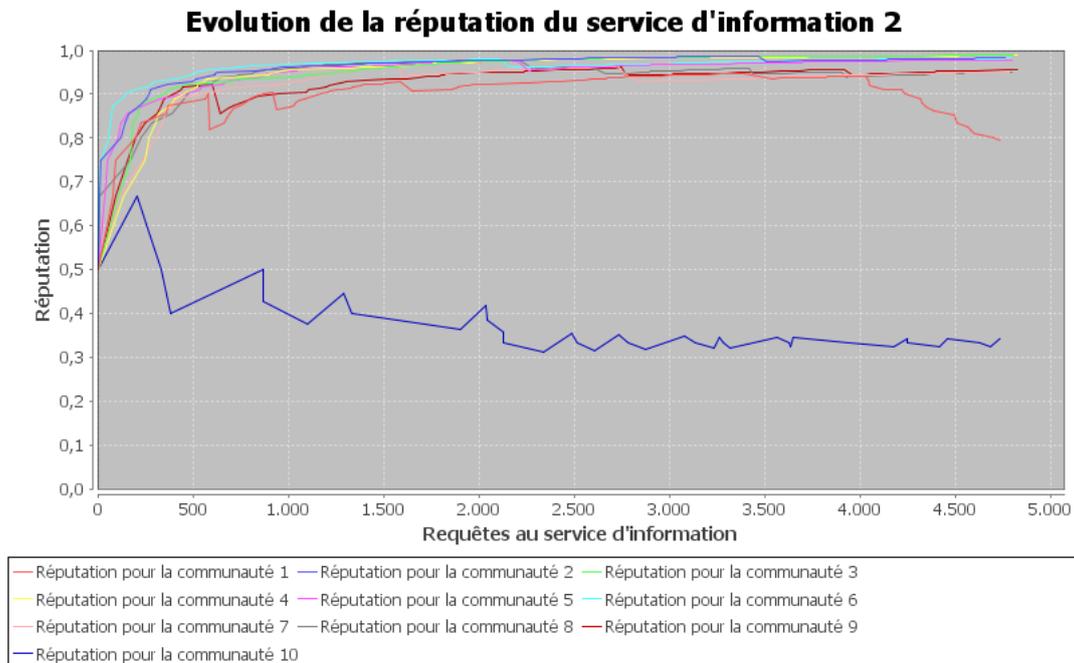


FIGURE 8.40 – Évolution de la réputation du deuxième service d'information pour la simulation 8.

A nouveau le paiement  $\gamma$  amassé est très élevé.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

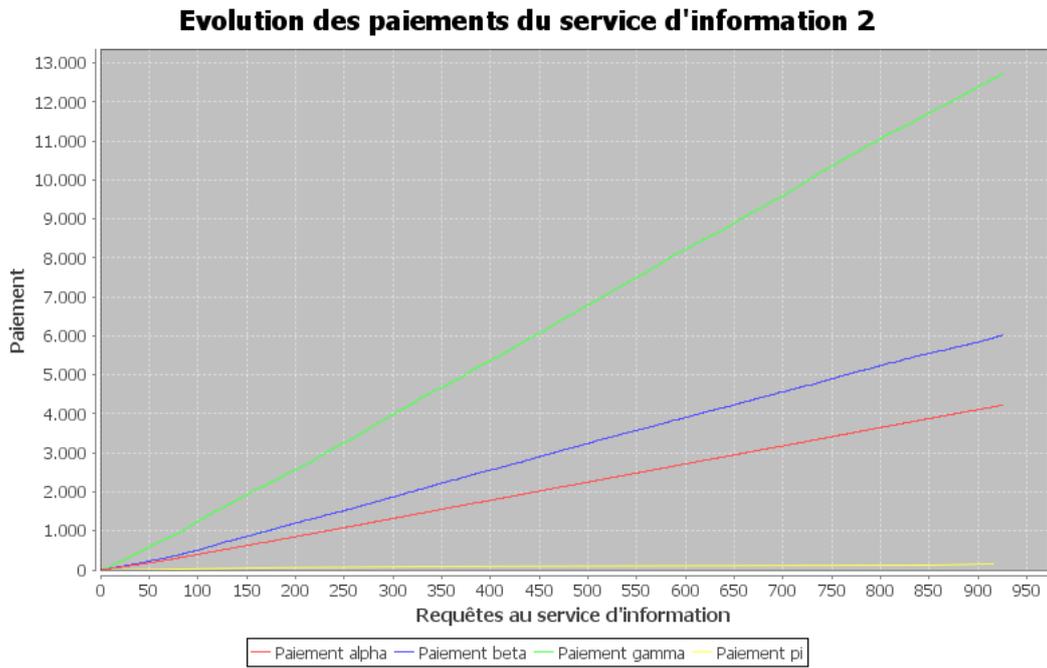


FIGURE 8.41 – Cumulation des différents paiements accordés au deuxième service d'information pour la simulation 8.

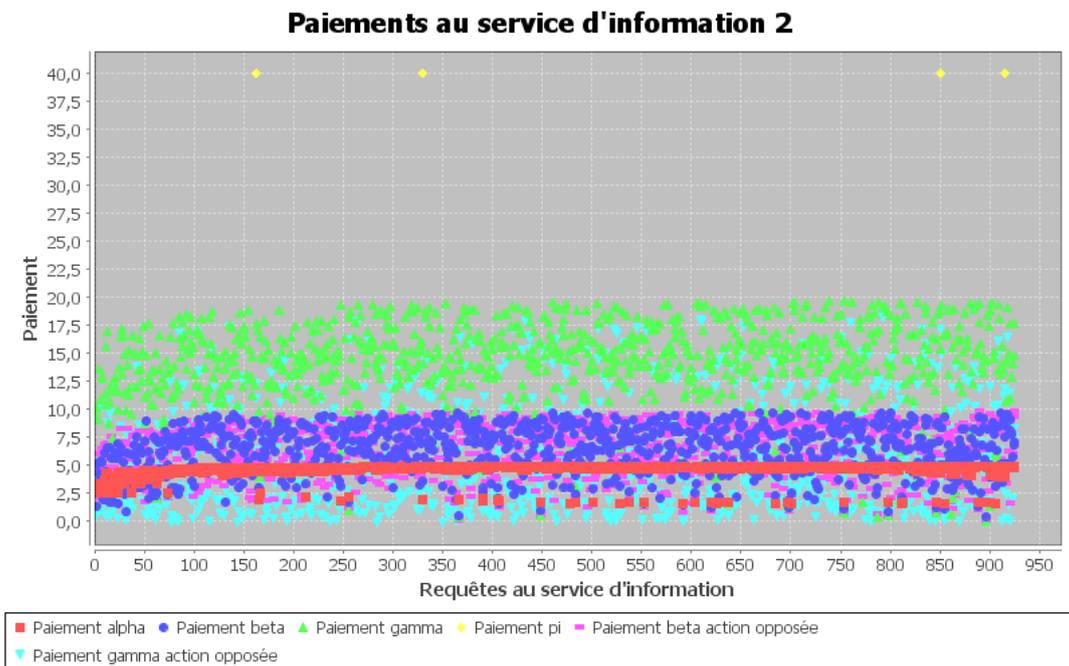


FIGURE 8.42 – Graphique des paiements accordés au deuxième service d'information pour la simulation 8.

## Service d'information 5

Regardons maintenant le service d'information ayant un facteur de patience de 0,8. Celui-ci ment plus de la moitié des interactions. La patience est donc bel et bien un facteur important dans notre modèle. Il semble d'ailleurs logique de penser qu'ils seront sûrement assez patients dans la réalité, nous en reparlerons dans la conclusion de ce chapitre.

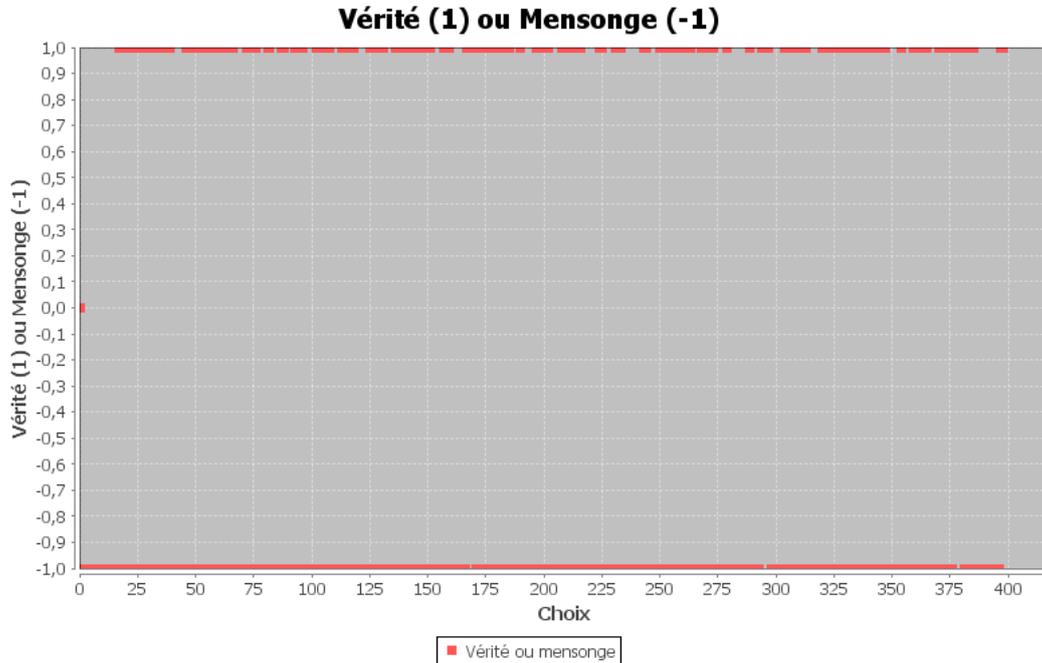


FIGURE 8.43 – Nombre de mensonges/vérités du cinquième service d'information (patience = 0,8) pour la simulation 8.

Sa réputation diminue d'abord pour finalement stagner à 0,3 dans toutes les communautés. Cela est dû au fait qu'il va essayer de conserver une certaine réputation à long terme et parer une trop grosse diminution après plusieurs mensonges par une ou plusieurs vérités.

Son paiement  $\pi$  est d'abord plus élevé que le  $\gamma$ . C'est parce qu'il commence les premières interactions par des mensonges. Ensuite il s'arrange pour augmenter ces deux paiements continuellement. Cependant, en comparant ces paiements avec ceux du deuxième service d'information plus haut par exemple, nous pouvons remarquer que son impatience réduit de beaucoup ses gains sur le long terme. En effet, son paiement  $\gamma$  de 1200, plus son paiement  $\pi$  final de 500 font pâle figure par rapport aux 13000 gagnés rien que par le paiement  $\gamma$  du service d'information 2, plus patient et consulté deux fois plus souvent grâce à sa bonne réputation.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

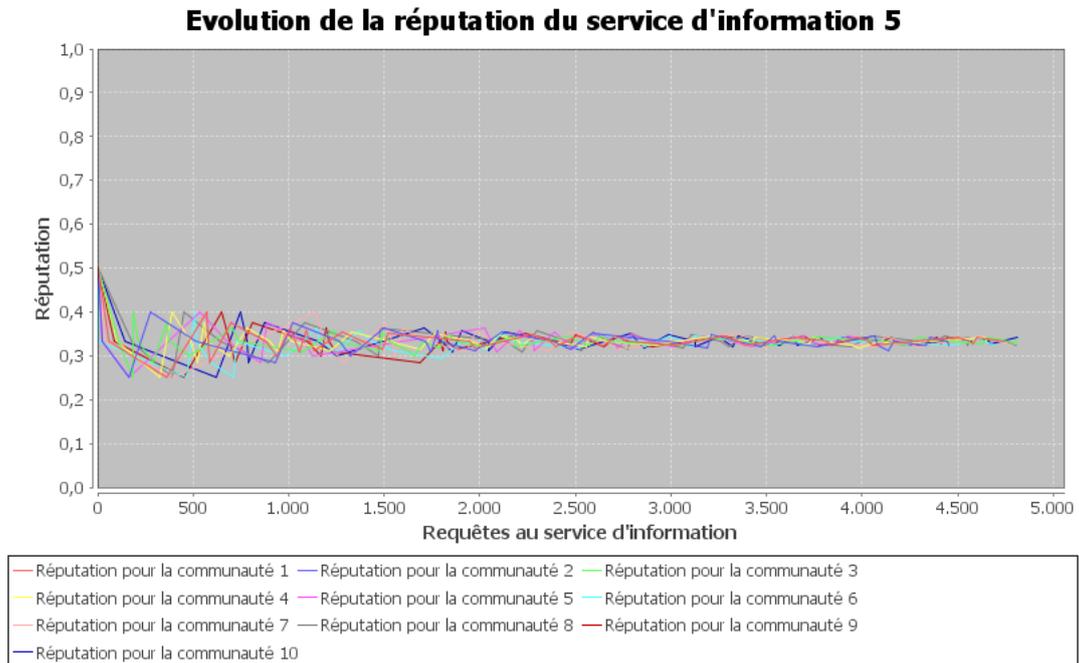


FIGURE 8.44 – Évolution de la réputation du cinquième service d'information pour la simulation 8.

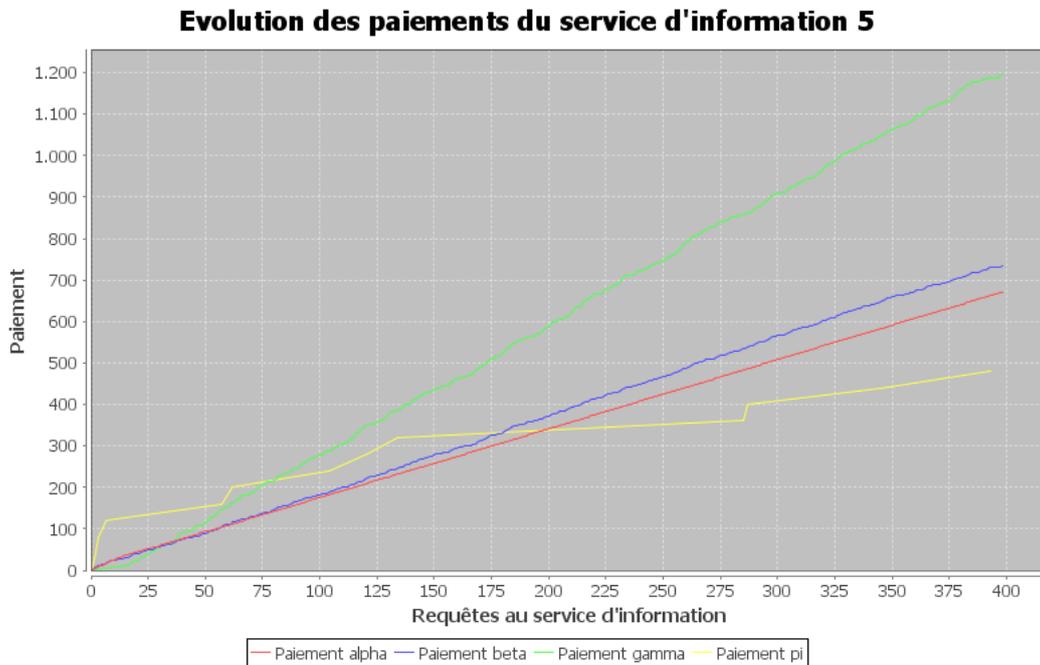


FIGURE 8.45 – Cumulation des différents paiements accordés au cinquième service d'information pour la simulation 8.

### 8.3.11 Simulation 9

#### Paramètres

Le nombre de coups à l'avance pris en compte par les services d'information est maintenant de 8. Les autres paramètres sont les mêmes que ceux de la simulation précédente.

#### Résultats généraux

Grâce au nombre de coups à l'avance pris en compte par les services d'information qui a doublé, nous voyons qu'il n'y a presque plus de mensonges (9 fois moins que de vérités).

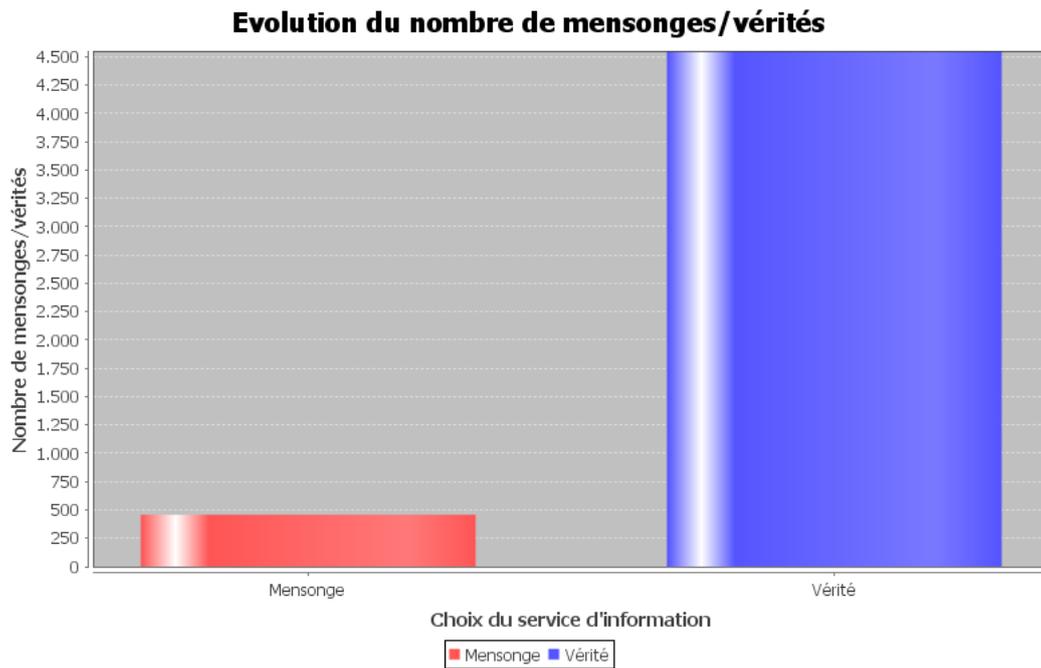


FIGURE 8.46 – Nombre de mensonges/vérités pour la simulation 9.

Les communautés sont donc plus efficacement guidées vers la bonne décision qui est d'accepter les services web de bonne qualité de service.

#### Service d'information 1

Le service d'information le plus patient dit en permanence la vérité.

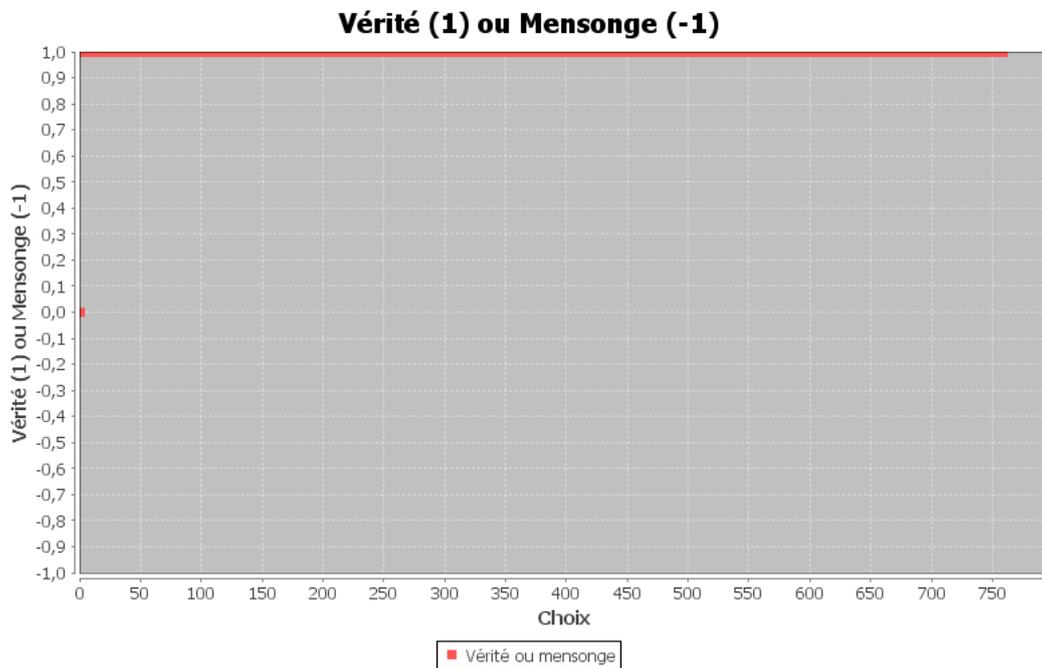


FIGURE 8.47 – Nombre de mensonges/vérités du premier service d'information (patience = 1) pour la simulation 9.

Sa réputation monte donc très rapidement.

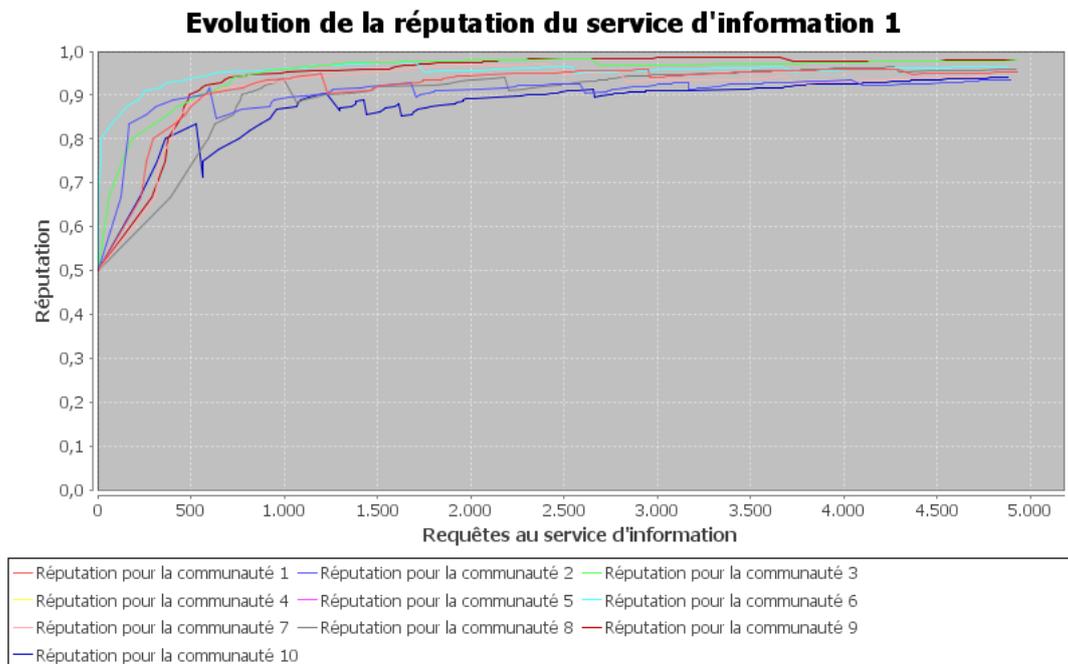


FIGURE 8.48 – Évolution de la réputation du premier service d'information pour la simulation 9.

Ses paiements sont tous très élevés sur le long terme.

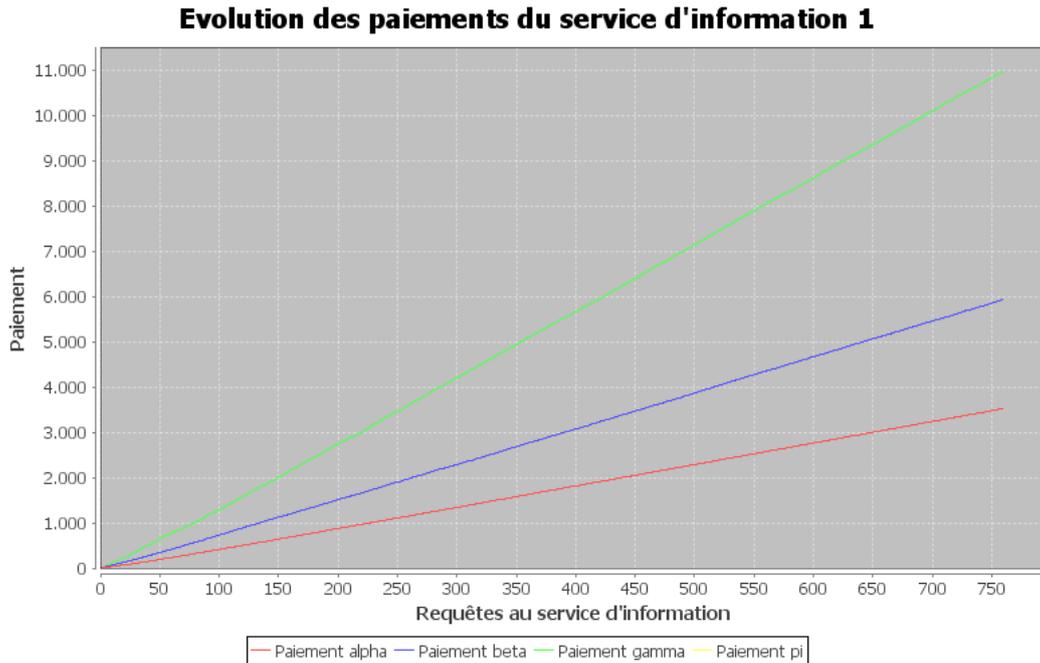


FIGURE 8.49 – Cumulation des différents paiements accordés au premier service d'information pour la simulation 9.

### Service d'information 5

Le service d'information de patience 0,8 ment de temps en temps, mais dit majoritairement la vérité, soit le contraire que lors de la simulation précédente. Un service d'information patient sans l'être extrêmement ment donc moins lorsqu'il réfléchit à plus long terme.

Sa réputation diminue pour les communautés auxquelles il a menti, mais certaines communautés lui associent une bonne réputation.

Bien qu'il ait menti de temps en temps, son paiement  $\pi$  reste très faible car les services web sont presque toujours acceptés (à cause de la faible proportion de mensonges), ce paiement n'est donc pas toujours donné, même lorsque le service d'information a menti.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

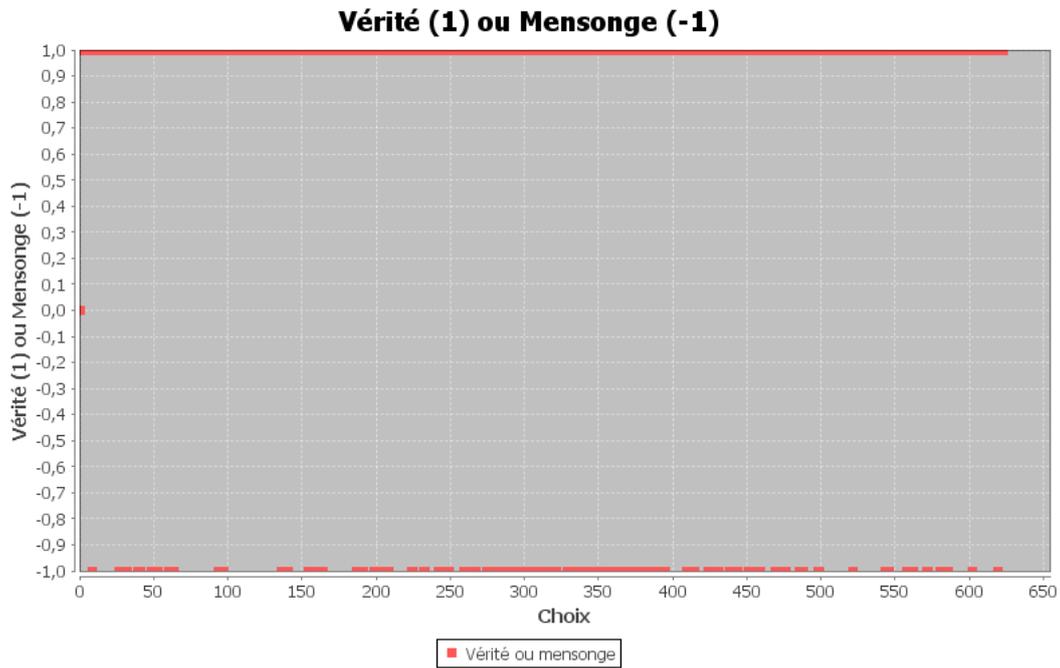


FIGURE 8.50 – Nombre de mensonges/vérités du cinquième service d'information (patience = 0,8) pour la simulation 9.

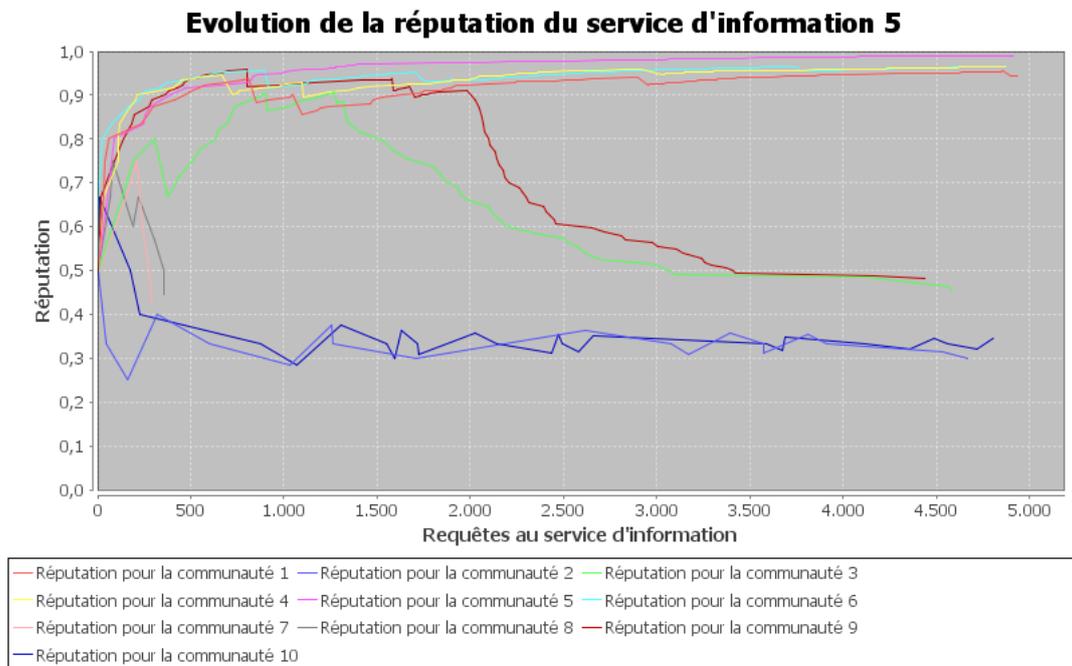


FIGURE 8.51 – Évolution de la réputation du cinquième service d'information pour la simulation 9.

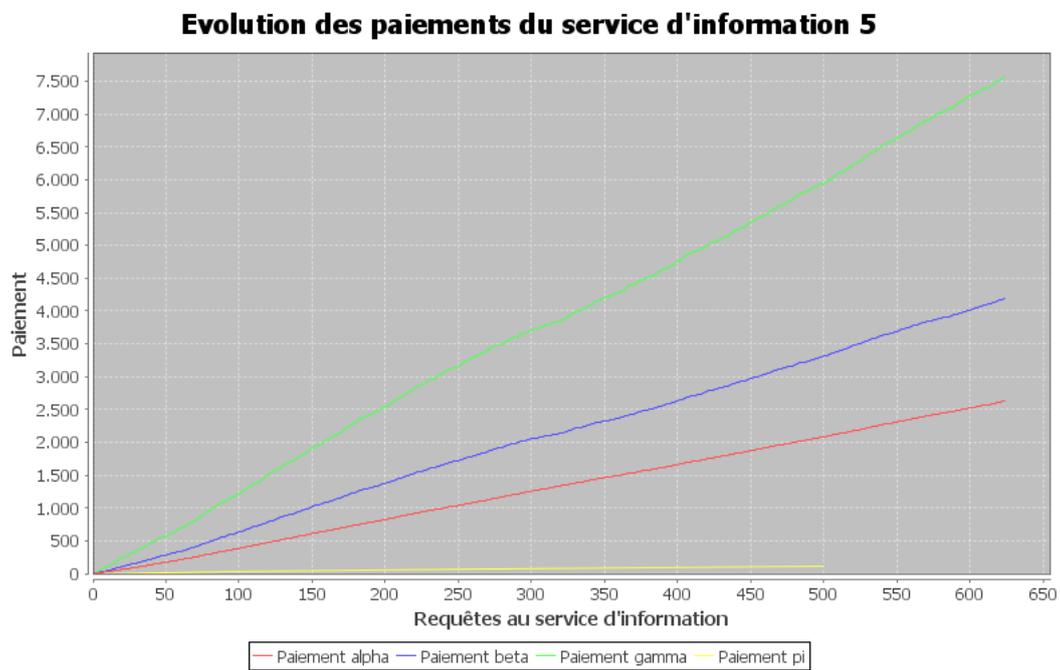


FIGURE 8.52 – Cumulation des différents paiements accordés au cinquième service d'information pour la simulation 9.

### 8.3 Résultats et analyse de cas spécifiques et pertinents

#### Service d'information 10

Ce service d'information est peu patient (0,6) et n'est presque pas consulté. En effet, il commence par mentir plus de 10 fois, avant de dire environ 1 fois sur 3 ou 4 la vérité. Les communautés cessent donc de le consulter lorsque sa réputation devient trop faible, au profit des services d'information plus patients.

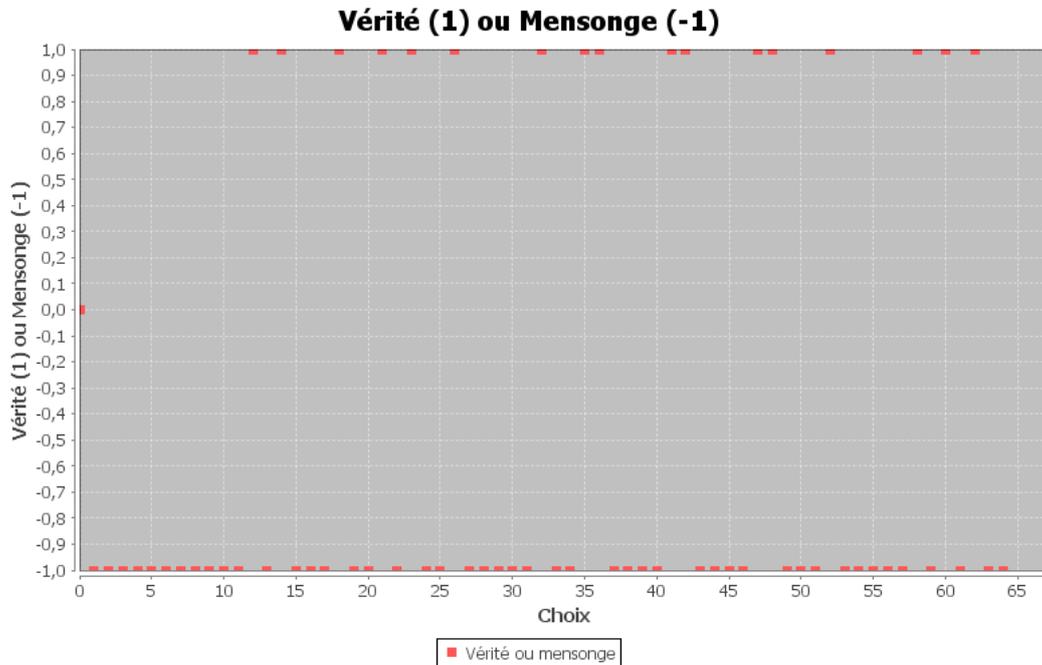


FIGURE 8.53 – Nombre de mensonges/vérités du dixième service d'information (patience = 0,6) pour la simulation 8.

Comme nous pouvons le constater, la réputation de ce service d'information baisse presque en permanence et reste toujours en-dessous du seuil de consultation de 0,5.

Son paiement  $\gamma$  est au final très faible, car il ne sera plus jamais consulté après la 64ème interaction.

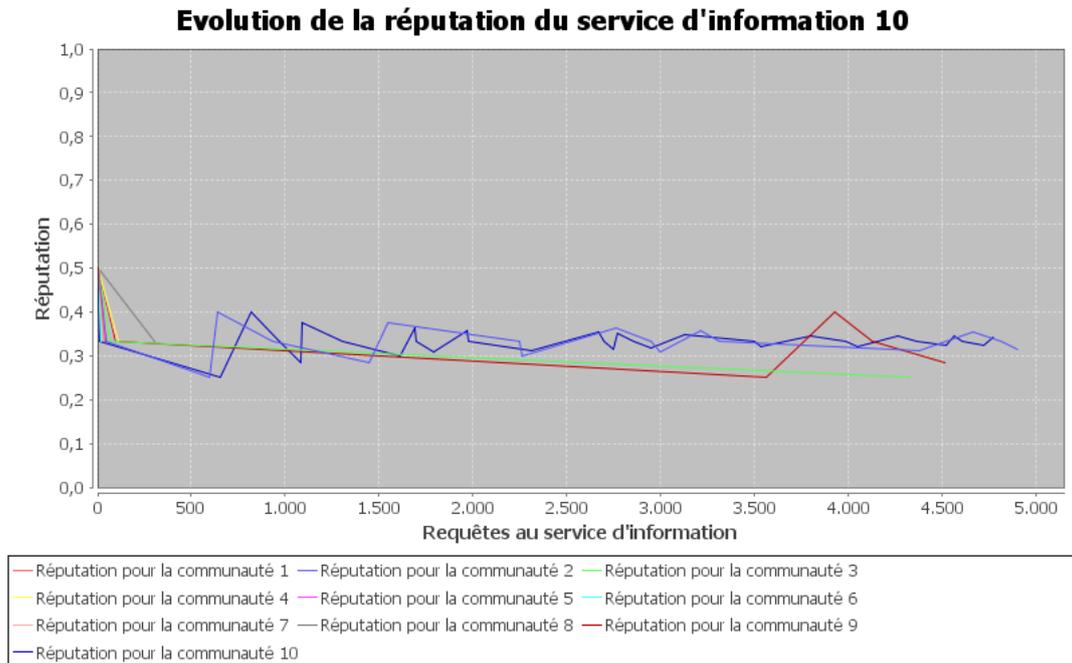


FIGURE 8.54 – Évolution de la réputation du dixième service d'information pour la simulation 8.

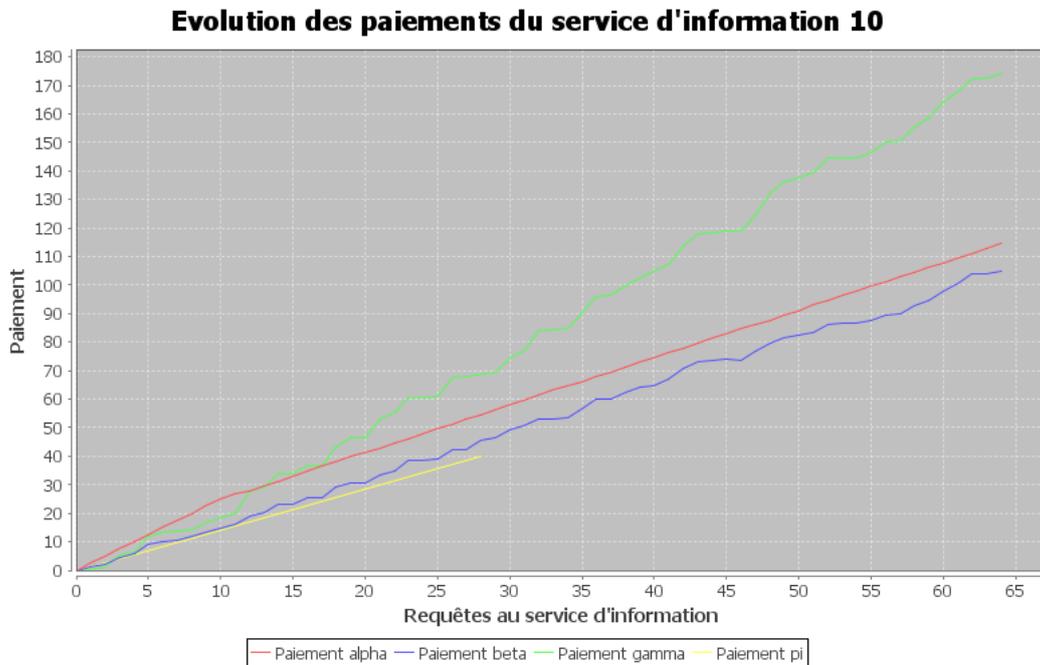


FIGURE 8.55 – Cumulation des différents paiements accordés au dixième service d'information pour la simulation 9.

### 8.3.12 Simulation 10

#### Paramètres

Nous exposons maintenant deux simulations générales. Dans celles-ci, les services d'information analysent des jeux stratégiques, et ne calculent donc pas plusieurs coups à l'avance. Il y a deux catégories de services web, chacune contenant 50% de la population, avec une qualité de service entre 0,01 et 0,99. Une catégorie contient des services web malhonnêtes proposant un  $\pi$  de 40 (s'il sont mauvais). L'autre contient des services web honnêtes. La probabilité de corruption par un tiers est fixée à 0,5, et la valeur du  $\pi$  à 40. Il y a 10 services d'informations avec des patiences entre 0,6 et 1. Les autres paramètres sont laissés par défaut.

#### Résultats généraux

Les résultats exposent qu'il y a environ deux fois plus de mensonges que de vérités. Cela découle du choix de ne pas utiliser les jeux répétés.

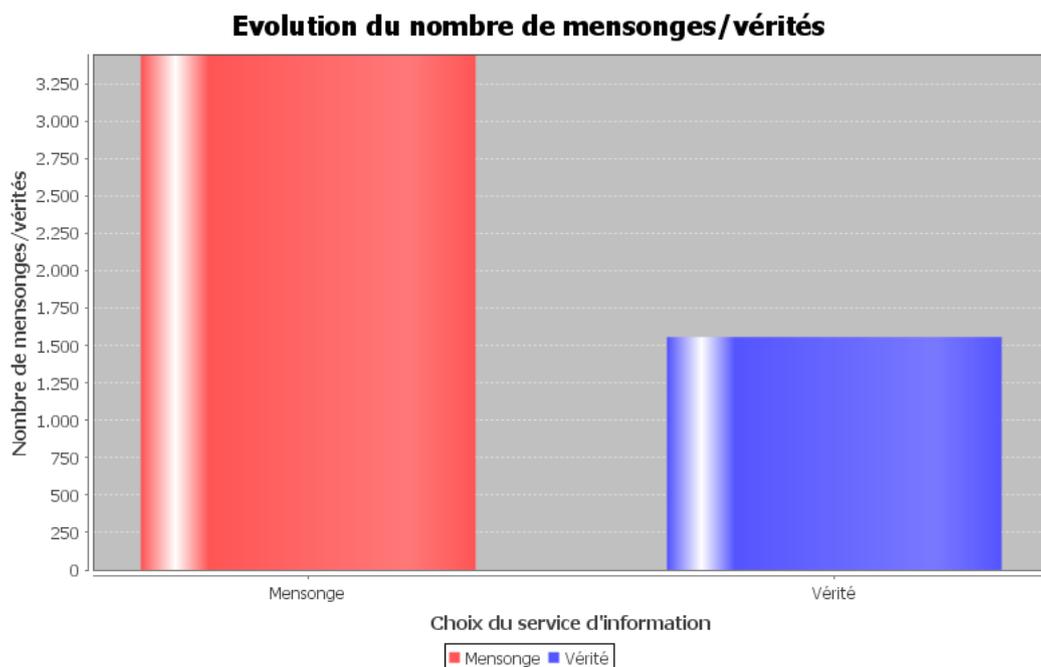


FIGURE 8.56 – Nombre de mensonges/vérités pour la simulation 10.

Cependant, nous pouvons voir sur la figure 8.57 que les bonnes décisions (que ce soit lors d'un refus ou d'une acceptation) sont légèrement plus nombreuses que les mauvaises dans toutes les communautés. Cela est dû au fait que les mensonges sont petits. Sur la figure 8.58, nous voyons la répartition des services web lorsque les mensonges sont plus gros. Les mauvaises décisions sont alors majoritaires. De nouveau les bonnes décisions prises le sont grâce aux "bons" services d'information (ayant une bonne réputation) qui disent souvent la vérité.

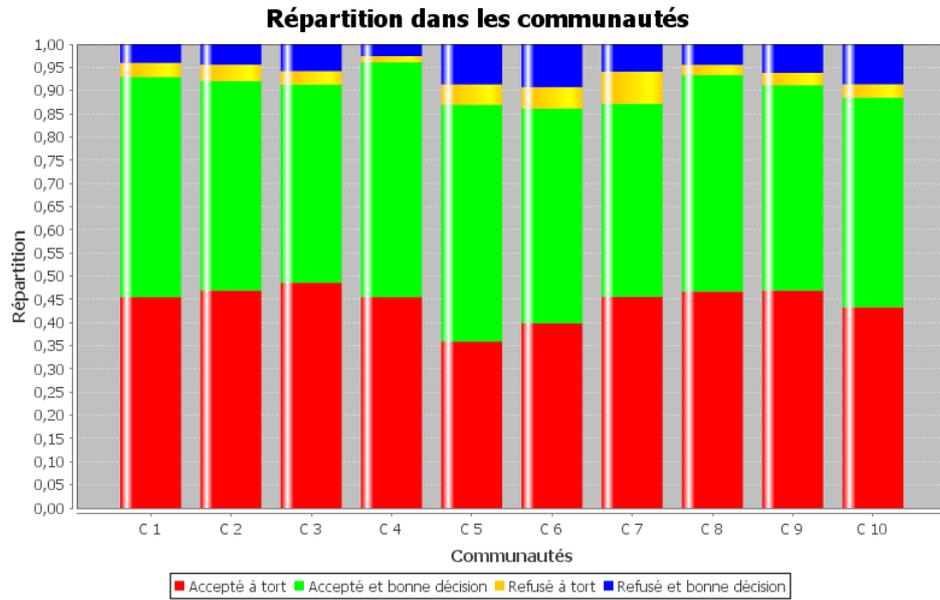


FIGURE 8.57 – Répartition des services web dans les communautés pour la simulation 10 lorsque les mensonges sont petits.

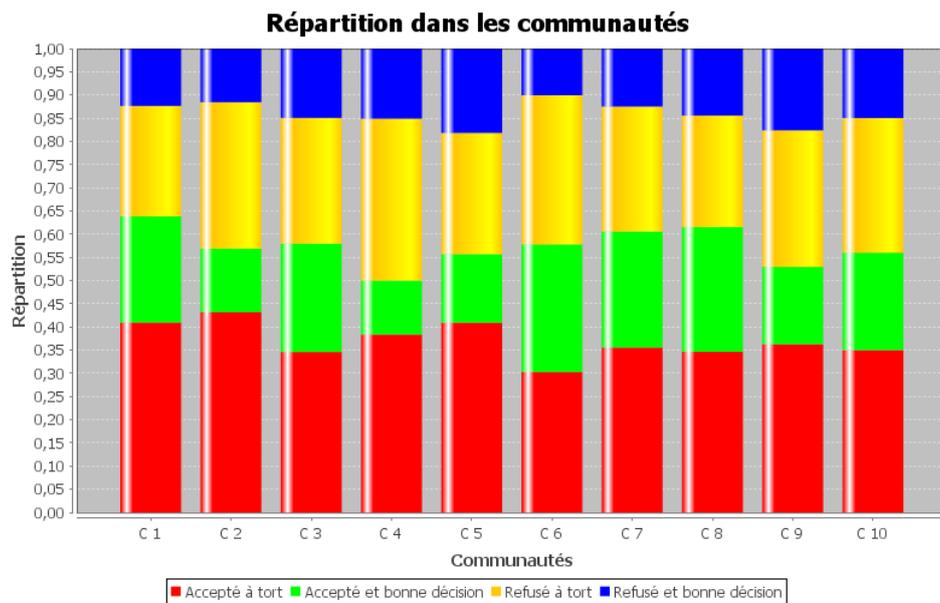


FIGURE 8.58 – Répartition des services web dans les communautés pour la simulation 10 lorsque les mensonges sont gros.

### 8.3.13 Simulation 11

#### Paramètres

Dans cette deuxième simulation générale, les services d'information prennent maintenant en compte 5 coups à l'avance. Les paramètres sont les mêmes que ceux de la simulation précédente.

#### Résultats généraux

Il y a maintenant une forte majorité de vérités. Nous voyons donc une fois de plus l'intérêt de la prise en compte des jeux répétés par les services d'information, dont bénéficient aussi les communautés.



FIGURE 8.59 – Nombre de mensonges/vérités pour la simulation 11.

Les bonnes décisions de la part des communautés sont maintenant très prévalentes sur les mauvaises. Elles ont été bien conseillées.

### 8.3.14 Simulation 12

#### Paramètres

Nous terminons les expérimentations par la mise en valeur de la probabilité d'acceptation d'un service web bon pourtant déconseillé, la probabilité  $P$ . Nous allons montrer deux simulations faisant varier cette valeur et les comparer. Pour ce faire, nous ne considérons que les bons services web, un  $\pi$  tiers de 40 avec une probabilité de 0,5, et une prise en compte de 5 coups à l'avance de la part des services d'information. La première simulation fixe la probabilité  $P$  à 0,1.

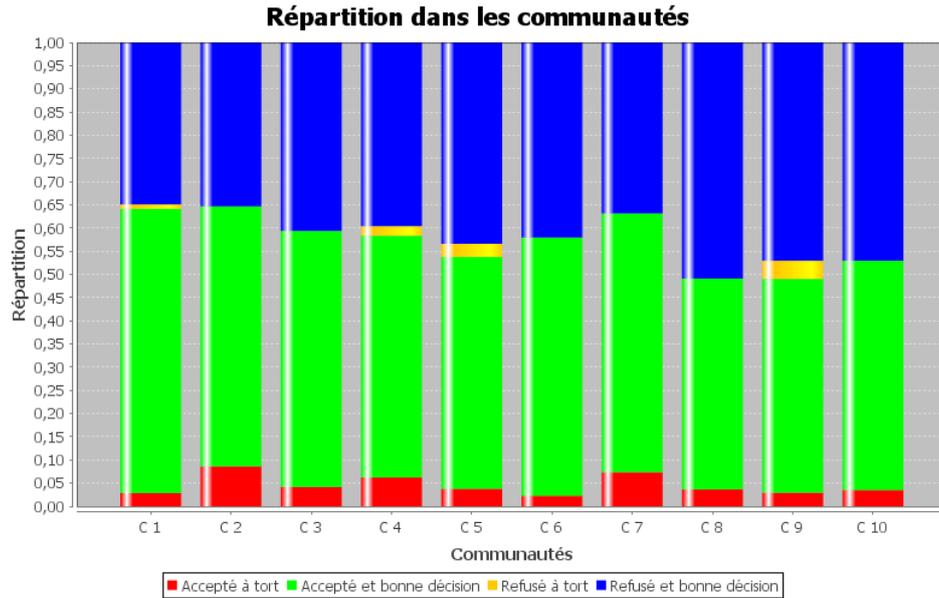


FIGURE 8.60 – Répartition des services web dans les communautés pour la simulation 11.

### Résultats généraux

Étant donné le nombre de coups à l'avance pris en compte par les services d'information, il y a beaucoup plus de vérités que de mensonges.

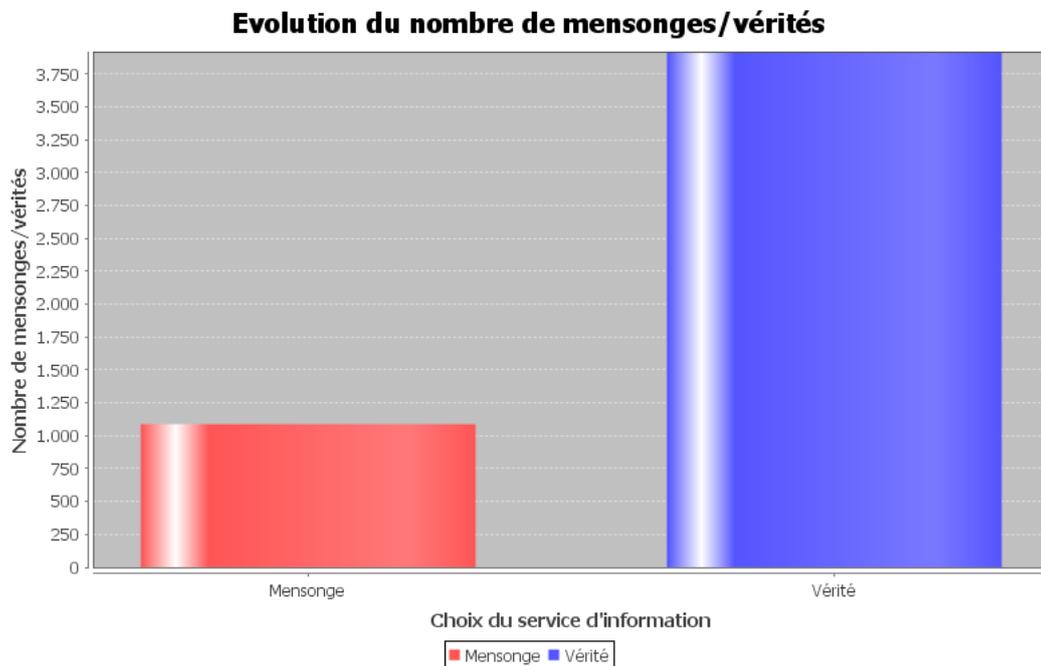


FIGURE 8.61 – Nombre de mensonges/vérités pour la simulation 12.

### 8.3.15 Simulation 13

#### Paramètres

Changeons maintenant la probabilité  $P$  et fixons là à 0,01. Les autres paramètres restent les mêmes que dans la simulation précédente.

#### Résultats généraux

Il y a maintenant presque autant de mensonges que de vérités. Les services d'information ont un sentiment d'impunité car ils savent que les communautés ne vérifieront presque jamais la véracité de leurs conseils.



FIGURE 8.62 – Nombre de mensonges/vérités pour la simulation 13.

Cependant, les bonnes décisions prévalent toujours lorsque les mensonges sont petits. La probabilité  $P$  joue donc sur la décision des services d'information et peut influencer sur le nombre de bonnes décisions prises lorsque les mensonges sont plutôt gros. Il ne faut pas oublier qu'un désavantage de la probabilité  $P$  est qu'elle peut forcer un maître de communauté à accepter des service web avec une mauvaise qualité de service. Si les mensonges sont petits, la probabilité  $P$  s'avère être un désavantage. Si les mensonges sont plutôt gros, la probabilité  $P$  est un avantage. Encore une fois l'importance des conseils des bons services d'information est important pour guider les décisions des communautés et joue aussi.

## 8.4 Conclusion

Dans la dernière section de ce chapitre concernant les simulations, nous allons tout d'abord comparer les différentes simulations de manière synthétique. Ensuite nous parlerons des résultats obtenus et de ce qu'ils impliquent.

### 8.4.1 Comparaisons

Dans les 6 premières simulations, nous avons testé des cas où tous les services web étaient mauvais. Voici la comparaison du ratio de mensonge/vérité pour chacune des simulations dans la figure 8.63. Dans la première simulation où nous pouvons observer qu'il y a 100% de mensonge, les services d'information réfléchissent à court terme (jeu stratégique), et un  $\pi$  de 40 est donné. Les services d'information préfèrent mentir pour obtenir un  $\pi$  élevé. Dans les simulations 2 et 3, les services d'information pensent à long terme mais les mensonges commis dans la simulation 3 sont plus gros que dans la simulation 2. Cependant, la « taille des mensonges » n'influe pas sur la décision de mentir ou non. Les taux de vérités et de mensonges pour les simulations 2 et 3 sont donc semblables. Les vérités dans ces deux simulations proviennent des services d'information les plus patients. Dans la simulation 4, où tous les services d'information sont patients, il n'y a plus que des vérités.

La simulation 5a ne prends pas en compte de paiement  $\pi$ , la différence entre cette simulation et la simulation 5b est que dans cette dernière les services d'information pensent à long terme. (Sans paiement  $\pi$  nous pouvons néanmoins remarquer qu'il y a beaucoup moins de mensonge que lorsqu'il y en a un, comme nous pouvons le voir sur la barre de la simulation 1).

La simulation 6 est le même que la simulation 5a sauf que les paiements sont plus facilement négatifs, nous pouvons donc observer que le nombre de mensonges en est réduit.

Regardons à présent le graphique comparant les pourcentages de vérité/mensonge pour les simulations 7 à 9.

Ces simulations se font avec des services web qui sont bons, la simulation 7 ne prend pas en compte de paiement  $\pi$  de tiers. Comme prévu, les services d'information disent tous la vérité. La simulation 8 ajoute un tel paiement  $\pi$  de 40 avec une probabilité de corruption par un tiers égale à 0,5. Les jeux répétés font 4 tours. La simulation 9 est égale à la précédente sauf qu'il y a 8 tours. La patience des services d'information est entre 0,6 et 0,9 pour toutes ces simulations, car nous ne jugions pas pertinents les autres niveau de patience, après avoir découvert que les plus impatientes ( $\delta < 0,5$ ) ne prennent de toute manière quasiment plus compte de la valeur des gains après 2 ou 3 tours. Nous voyons donc que pour les services d'information assez patients, le nombre de tours joue beaucoup sur le nombre de vérités.

Regardons maintenant le graphique comparant les simulations 12 et 13 (figure 8.65). La seule différence entre ces deux simulations et la valeur de  $P$  qui change. La simulation 12

## 8.4 Conclusion

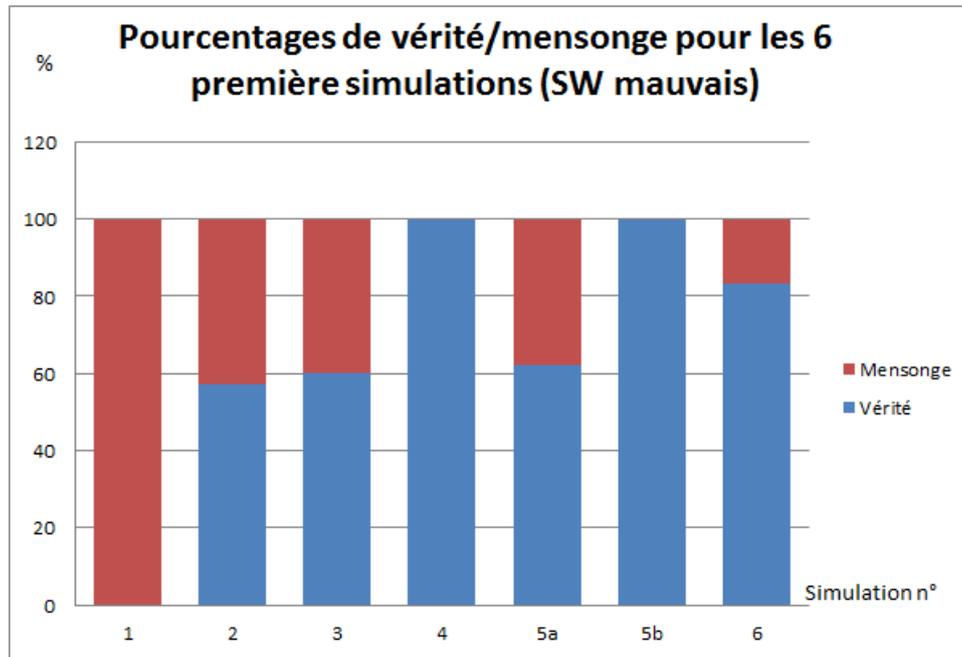


FIGURE 8.63 – Pourcentage de vérité/ mensonge pour les 5 premières simulations.

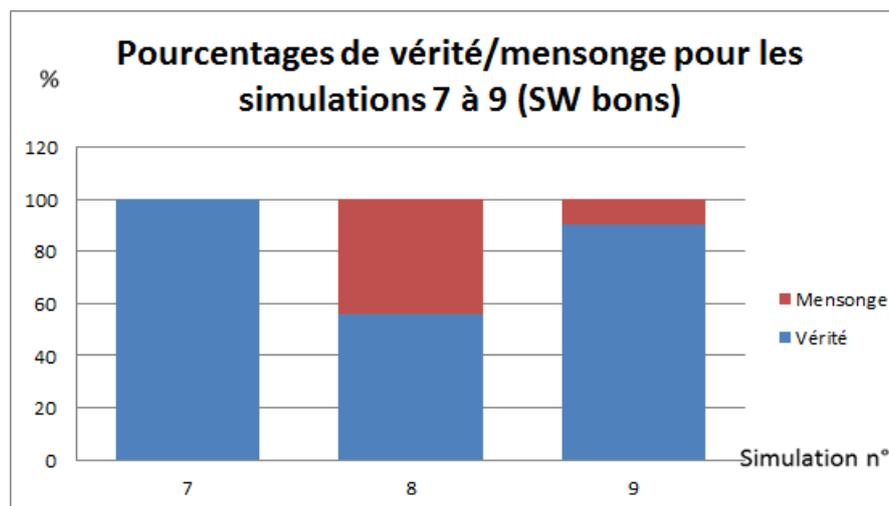


FIGURE 8.64 – Pourcentage de vérité/ mensonge pour les simulations 7 à 9.

possède une probabilité  $P$  de 0,1 et la 13 une probabilité  $P$  de 0,01. Nous voyons qu'il y a plus de mensonges lorsque cette probabilité est faible, les services d'information se sentant immunisés lors d'un mensonge, pensant que la communauté ne le remarquera pas (étant donné qu'elle refuse le service web, refusé à tort).

Reprenons maintenant les simulations 10 et 11. Ce sont des simulations générales. La seule différence est que la 11 utilise les jeux répétés (5 tours), et la 10 les jeux stratégiques (« one-shot »). La simulation 11 a de très bons résultats, bien meilleurs que la simulation 10. Dans la figure 8.66 la simulation 10 est au-dessus. La simulation 11 prend en compte le long terme et les services d'information sont assez patients (0,6 à 1). Les services d'information

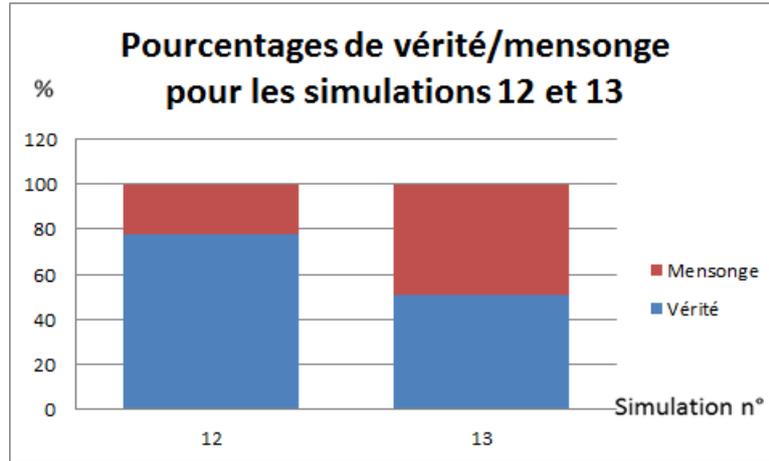


FIGURE 8.65 – Pourcentage de vérité/ mensonge pour les simulations 12 et 13.

assez patients et pensant sur le long terme préfèrent donc dire la vérité.

#### 8.4.2 Implications des résultats

Nous déduisons donc de ces résultats, que notre modèle répond bien aux problèmes. D'une part le respect de 4 des propriétés de justesse pour le paiement a été implémenté, la cinquième propriété apportait des résultats légèrement moins bons, comme nous en avons déjà parlé dans le chapitre 5. La probabilité  $P$  joue bien sur les décisions des services d'information lorsqu'ils choisissent entre le mensonge ou la vérité. La possibilité de rendre les paiements négatifs afin de punir les services d'information malhonnêtes permet de conjurer la corruption (en partie), et d'éviter au maître de communauté de devoir augmenter la valeur de ses paiements maximums. Enfin, l'ajout des jeux répétés et la prise en compte de la patience joue énormément sur l'honnêteté des services d'information. Un service d'information patient et réfléchissant sur le long terme décidera très souvent si pas toujours de dire la vérité. Au final, notre mécanisme d'information nous semble performant lorsque les services d'information agissent sur le long terme et sont plutôt patients.

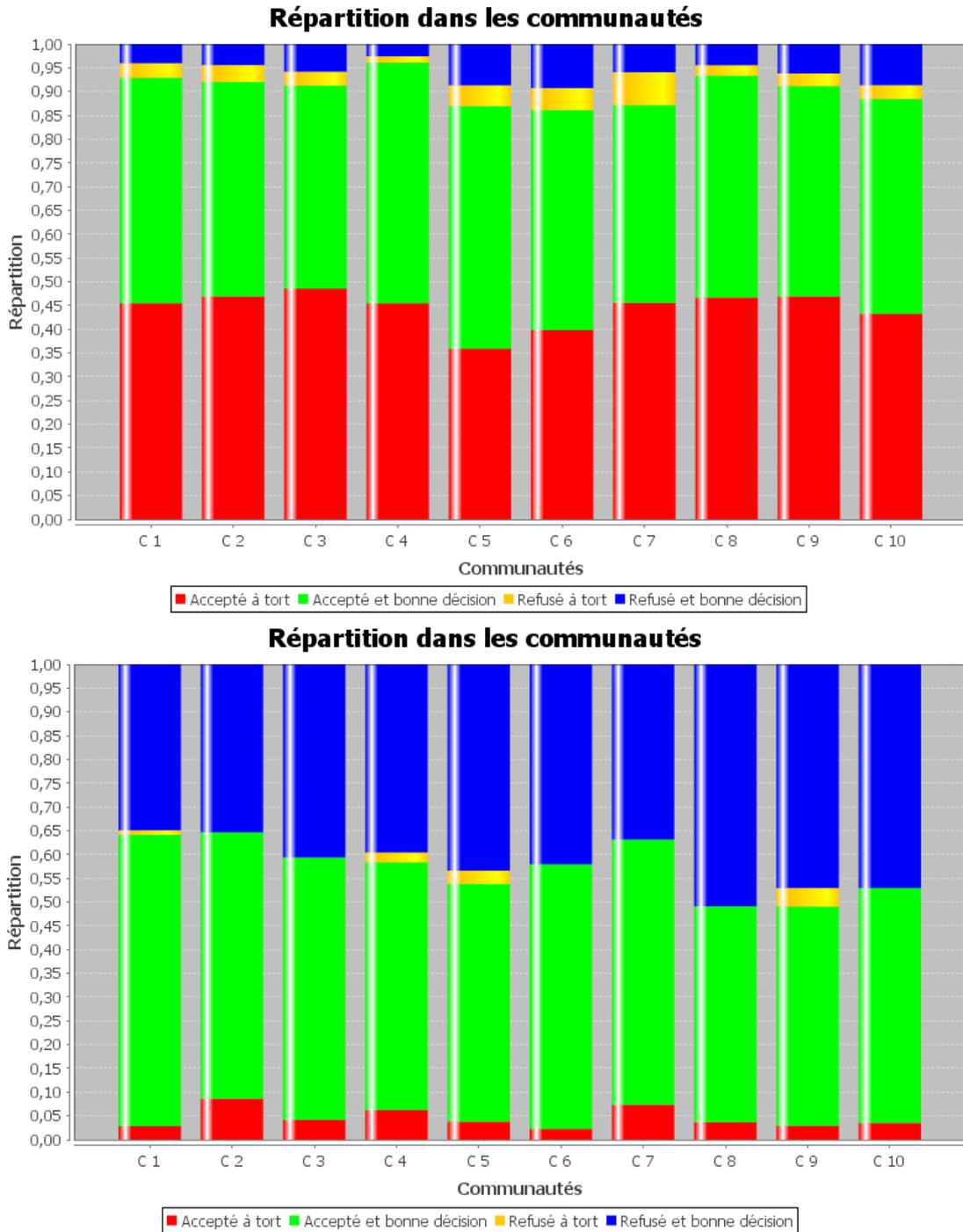


FIGURE 8.66 – Comparaison des décisions des communautés.



## Chapitre 9

# Conclusion

Dans un système où plusieurs services web collaborent en formant des communautés, le problème de la sélection des nouveaux membres est omniprésent. Dans la description du contexte, nous avons vu qu'un service web avait, pour plusieurs raisons, intérêt à rejoindre une communauté. Il est important pour la communauté de n'accepter en son sein que des services web ayant une bonne qualité de service. Afin de savoir cela, le maître de communauté va calculer la réputation du service web en faisant appel à des services d'information. Ceux-ci sont des services spécialisés dans l'évaluation de la qualité de service des services web, et qui désirent vendre cette information essentielle au maître de communauté. Cependant, les services d'information sont des agents rationnels désirant maximiser leurs revenus. Par conséquent, ils pourraient décider de mentir en transmettant une fausse évaluation s'ils venaient à être corrompu. Il est donc nécessaire de mettre en place un mécanisme incitant les services d'information à être honnêtes.

Dans le chapitre 2, nous avons introduit les systèmes de réputation en présentant les techniques déjà existantes. Nous avons notamment vu qu'un simple système de réputation pouvait servir de mécanisme d'incitation. Ce chapitre faisait également office d'un bref état de l'art sur les systèmes de réputation car certaines des techniques présentées (comme *Sporas* et *Histos*) n'ont pas été utilisées dans la suite de ce mémoire.

Ensuite, dans le chapitre 3, nous avons introduit les principaux éléments de la théorie des jeux dont nous nous servons tout au long du mémoire. Le chapitre reprend les jeux stratégiques, les jeux étendus, l'ajout de mouvements simultanés et de mouvements aléatoires, et enfin, les jeux répétés. Nous utiliserons les jeux stratégiques dans les chapitres 4 à 6, les jeux étendus dans le chapitre 6, et les jeux répétés dans le chapitre 7. Les jeux stratégiques et répétés sont aussi utilisés par le simulateur.

Une fois munis de ces outils, nous avons pu présenter et expliquer le modèle sur lequel nous nous basons dans la suite du mémoire. Ce modèle est présenté dans le chapitre 4. Il consiste en une fonction de paiement constituée de trois composantes. La première composante est un paiement fixe donné à un service d'information afin d'obtenir l'information qu'il détient. La deuxième composante est un paiement qui dépend de la différence entre l'information transmise et la réputation moyenne du service web calculée par le maître à

partir de toutes les notations reçues. A partir de cette réputation calculée, le maître décide s'il accepte le service web ou non au sein de la communauté. S'il l'accepte, le maître pourra évaluer la qualité de service de celui-ci. Le maître peut alors accorder le dernier paiement qui dépend de la satisfaction qu'il a de l'information transmise par le service d'information. Le troisième paiement ne peut donc être donné que lorsque le service web est accepté.

En nous inspirant d'un second travail ([GK09]), nous redéveloppons la troisième composante dans le chapitre 5, afin de rendre les paiements accordés aux services d'information les plus justes possibles. Nous en profiterons pour redévelopper la seconde fonction de paiement de manière à ce qu'elle ait une forme semblable à la troisième. Dotés de ces deux nouvelles fonctions, nous créons notre propre modèle, présenté à la fin du chapitre 5.

Dans le chapitre 6, nous avons ajouté un cas afin d'enrichir le modèle : la possibilité de corruption par un tiers. En effet, cette nouvelle possibilité complique la tâche du maître de communauté lorsqu'il désire sélectionner judicieusement ses services web, un service concurrent pouvant désirer empêcher un certain service web d'entrer. Nous avons donc étudié les nouvelles possibilités et les conséquences découlant de ce nouveau cas. Suite à une nouvelle analyse des jeux stratégiques, nous avons déterminé pour chaque cas, les conditions requises pour avoir l'un ou l'autre équilibre de Nash.

Dans le chapitre 7, nous avons décrit comment nous désirions prendre en compte les jeux répétés dans notre modèle. Il en résulte que, lorsque les services d'information devront donner un conseil, leur décision ne se fera plus immédiatement (« one-shot ») mais ils effectueront un calcul de coups à l'avance, afin de prendre en compte le futur, dans lequel les conséquences de leurs actes présents se ressentiront. Nous avons déduit que cet ajout devrait améliorer l'honnêteté des services d'information sauf pour un service d'information totalement impatient, ne se souciant pas du futur et ne prenant pas en compte sa baisse de réputation, mais uniquement son paiement immédiat.

Dans le dernier chapitre, nous avons fait une batterie de tests, ou plutôt de simulations, à l'aide d'un simulateur simplifiant certains aspects de la théorie. Celui-ci a été repris du travail de 2010-2011 pour sa partie graphique et dont on a reprogrammé le fonctionnement, le déroulement interne n'étant plus semblable dû aux changements effectués sur le modèle. Les résultats des simulations sont plutôt positifs. Il en ressort majoritairement que la patience des services d'information et leur prise en compte du futur augmentent énormément leur honnêteté. De plus, la probabilité  $P$  d'accepter un service web pourtant déconseillé empêche effectivement les services d'information de se sentir impunis lors d'un mensonge. Enfin, les paiements négatifs semblent décourager les mensonges.

Les apports de notre modèle paraissent donc positifs quant au comportement des services d'information, lorsqu'ils agissent sur le long terme et sont plutôt patients. Cependant, rien ne dit qu'un cas réel se déroule comme étudié ici. Par exemple, les services d'information pourraient très bien être impatients et ne penser qu'au présent lors de leurs décisions.

---

Comme un modèle a pour but de s'abstraire de la réalité afin de pouvoir représenter une partie de ses aspects, il est normal que beaucoup d'événements non pris en compte, voire inconnus, interviennent dans la réalité. Si nous sommes assez satisfaits des résultats de notre modèle, il possède des limitations.

### **Limitations et travail futur**

Nous n'avons pas pris en compte plusieurs problèmes. Notamment la réputation d'une communauté. En effet, celle-ci aussi pourrait avoir une réputation, par exemple une réputation globale (connue de tous les agents dans le système), qui changerait selon sa justesse de paiement envers les services d'information. Un maître de communauté pourrait aussi être malhonnête et par exemple décider de ne pas donner un paiement en faisant comme si les services d'information lui avaient menti. Il pourrait également ne pas toujours mettre à jour la réputation des bons services d'information afin de ne pas devoir les payer plus cher plus tard. Ces cas dépassent le cadre de ce mémoire.

Nous avons également écarté le cas où un service web interne à une communauté aimerait aider un service web à rentrer dedans. Il y a aussi la possibilité de collusion de la part de plusieurs services web qui s'associeraient pour nuire à un concurrent en corrompant les services d'information. Ce scénario contredirait notre hypothèse que les paiements  $\pi$  ne devraient pas être trop élevés, car il serait illogique qu'un service web ait assez d'argent pour contrecarrer un paiement de la part de la communauté toute entière qui, elle, dispose de plus de fonds.



# Bibliographie

- [BK11a] Jamal Bentahar Philippe Thiran Babak Khosravifar, Mahsa Alishahi. A game theoretic approach for analyzing the efficiency of web services in collaborative networks. *SCC*, 2011.
- [BK11b] Kathleen Clacens Christophe Goffart Philippe Thiran Babak Khosravifar, Jamal Bentahar. Game-theoretic analysis of a web services collaborative mechanism. In *Proceedings of the 9th International Conference on Service Oriented Computing (ICSOC 2011)*, pages 549–556, Springer, Paphos, Cyprus, 2011. Morgan Kaufmann Publishers Inc.
- [ca] Auteurs collaboratifs anonymes. Service web, [http://fr.wikipedia.org/wiki/service\\_web](http://fr.wikipedia.org/wiki/service_web), date de consultation : 05/07/2012.
- [CK11] Goffart Christophe Clacens Kathleen. Game theory based web services collaborative mechanism. Master’s thesis, FUNDP, 2011.
- [Del03] Chrysanthos Dellacoras. The digitization of word of mouth : Promise and challenges of online feedback mechanism. *Management Science*, 49(10) :1407–1424, Octobre 2003.
- [dSAN] Université de Sophia Antipolis (Nice). Www interactive multipurpose server, <http://wims.unice.fr/wims/wims.cgi>, dernière date de consultation : 28/08/2012.
- [eCS05] Jordi Sabater et Carles Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1), pages 33–60, 2005.
- [eRI02] Audun Jøsang et Roslan Ismail. The beta reputation system. *Proceedings of the Fifteenth Bled Electronic Commerce Conference*, pages 324–337, 2002.
- [Fou] The Eclipse Foundation. Eclipse, <http://www.eclipse.org/>, 2012.
- [Gio99] Zacharia Giorgos. Collaborative reputation mechanisms for online communities. Master’s thesis, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences, 1999.
- [GK09] Robin Cohen Georgia Kastidou, Kate Larson. Exchanging reputation information between communities : A payment-function approach. pages 195–200, 2009.
- [jeBF03] Radu jurca et Boi Faltings. An incentive compatible reputation mechanism. *Proceedings of the IEEE Conference on E-Commerce (CEC)*, pages 205–212, 2003.
- [jeBF04] Radu jurca et Boi Faltings. Confess : An incentive compatible reputation mechanism for the online hotel booking industry. *Proceedings of the IEEE Conference on E-Commerce*, pages 205–212, 2004.

- [jeBF07] Radu jurca et Boi Faltings. Collusion-resistant, incentive-compatible feedback paymentsy. *In Proceedings of the Eighth ACM Conference on Electronic Commerce (EC)*, pages 200–209, 2007.
- [Jie09] Zhang Jie. *Promoting Honesty in Electronic Marketplaces : Combining Trust Modeling and Incentive Mechanism Designl*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2009.
- [Jur07] Radu Jurca. *Truthful Reputation Mechanism for Online Systems*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2007.
- [Laf] Yves Lafon. Web services activity, [http ://www.w3.org/2002/ws/](http://www.w3.org/2002/ws/), v 1.312 05/05/2011, date de consultation : 05/07/2012.
- [Osb09] Martin J. Osborne. *An introduction to game theory (international edition)*. Oxford University Press, 2009.

# Annexe A

## Le simulateur

Dans cette annexe, nous allons décrire le fonctionnement du simulateur. Celui-ci est basé sur le simulateur réalisé en 2010-2011 par les étudiants K. Clacens et C. Goffart. Nous y ferons référence par la dénomination « simulateur de base ». L'interface graphique et la structure principale du simulateur de base ont été reprises.

### A.1 Introduction

Le simulateur est codé en Java 1.6 sur la plateforme Eclipse ([Fou]).

### A.2 Paramétrage des agents et hypothèses faites

Voir la section 8.1 : « Présentation rapide du simulateur ».

### A.3 Structure du programme

Le programme est composé de 7 packages :

- *controller* : ce package contient l'unique classe *Controller*. Elle contient la méthode *main*, initialise l'interface au démarrage du programme, et gère les événements déclenchés par l'interface ;
- *exceptions* : comme son nom l'indique, ce package contient les différentes classes d'exceptions ;
- *extRepEntities* : ce package contient les classes représentant une entité, un agent, comme les services web, les services d'information, et les communautés ;
- *extRepGameTheory* : ce package contient les classes permettant de déterminer le comportement d'un service d'information (l'action qu'il va choisir : dire la vérité ou mentir) ;

- observer : ce package contient les classes *Observer* et *Observable* ;
- view : ce package contient les classes en rapport avec l'interface utilisateur ;
- utils : ce package contient les classes ne correspondant à aucun autre package. Ce sont en quelque sorte des classes « outils ».

Nous allons maintenant détailler le contenu des packages *extRepEntities*, *extRepGameTheory*, et *utils*.

### A.3.1 Le package *extRepEntities*

Ce package est composé des classes *ExtRepCommunity*, *ExtRepInformationService*, *ExtRepWebService*, et *ExtRepRepository*.

#### **ExtRepWebService**

Un objet de type *ExtRepWebService* représente un service web. Chaque service web est un thread qui va tenter d'entrer dans une communauté choisie au hasard. S'il échoue, il réessaye avec une autre communauté, et ce, jusqu'à qu'il ait essayé d'entrer dans chaque communauté, ou que le nombre maximum de tentatives soit atteint. Le nombre maximum de tentatives d'entrer dans une communauté est un paramètre fixé par l'utilisateur. Par défaut, il est fixé à une tentative. Lorsque le service web est accepté au sein d'une communauté, ou a épuisé le nombre tentatives possible, le thread s'arrête.

#### **ExtRepCommunity**

Un objet de type *ExtRepCommunity* représente une communauté. Dans le simulateur de base, chaque communauté était un thread qui passait son temps à tester les services web la composant. Afin d'améliorer les performances du simulateur, nous avons changé cette implémentation. Une communauté n'est plus un thread. La classe *ExtRepCommunity* contient une méthode nommée *join*. Cette méthode est appelée par chaque service web voulant rejoindre la communauté. L'appel de cette méthode exécute le code décidant si la communauté va accepter ou refuser le service web (appel des services d'information, calcul de la notation moyenne du service web, etc.). Si la communauté accepte le service web, la méthode permettant de tester le service web est appelée. Cette classe contient aussi les méthodes permettant de calculer les paiements accordés à chaque service d'information.

#### **ExtRepInformationService**

Un objet de type *ExtRepInformationService* représente un service d'information. Dans le simulateur de base, chaque service d'information était aussi un thread. Celui-ci se réveillait toute les 5 secondes pour vérifier si le service d'information avait été utilisé par une communauté (une variable booléenne était mise à *true* lorsqu'un service d'information était utilisé). Si la variable avait la valeur *false*, le service d'information augmentait la valeur de la variable *truthTendency*, puis se rendormait 5 secondes. Par contre, si la

## A.3 Structure du programme

---

variable avait la valeur *true*, il la mettait à *false* et se rendormait 5 secondes en laissant la valeur de la variable *truthTendency* inchangée. La variable *truthTendency* n'étant pour nous pas pertinente, nous l'avons supprimée. En effet, pourquoi décider de l'augmenter lorsqu'il n'y a pas eu d'interaction pendant 5 secondes, et pas 3 secondes ou 8 secondes ? Ce paramètre et l'impact qu'il avait sur le choix du service d'information de mentir ou non étant très flou, nous l'avons donc supprimé. Dans le nouveau simulateur, les services d'information ne sont plus des threads. La classe *ExtRepInformationService* contient une méthode nommée *getWSReputation*. Cette méthode est appelée par une communauté pour que le service d'information transmette une notation. L'appel de cette méthode entraîne l'exécution du code déterminant si le service d'information va mentir ou dire la vérité. La réputation transmise peut donc bien sûr être fausse.

### ExtRepRepository

La classe *ExtRepRepository* est instanciée une seule fois (singleton pattern), et est utilisée pour stocker des informations diverses sur les différentes entités. *ExtRepRepository* possède la liste des tous les services web, tous les services d'information, et toutes les communautés.

#### A.3.2 Le package *extRepGameTheory*

Ce package est composé des classes *Combinaisons*, *DiscountingOutcome*, *ExtRepChoice*, *ExtRepGame*, et *ExtRepOutcome*.

La classe *DiscountingOutcome* contient les méthodes permettant à un service d'information de prendre sa décision (dire la vérité ou mentir) en répétant le jeu représentant sa situation actuelle.

Un objet de type *ExtRepChoice* représente un choix, une action (dire la vérité ou mentir).

Un objet de type *ExtRepGame* représente un jeu.

La classe *ExtRepOutcome* contient une seule méthode, qui calcule l'utilité d'un service d'information selon son choix.

Un objet de type *Combinaisons* représente une combinaison d'actions. Par exemple, (Mentir, Mentir, Dire la vérité).

#### A.3.3 Le package *utils*

Ce package est composé des classes *ISComparator*, *Stats*, et *WebService Starter*.

*WebService Starter* est un thread qui démarre les services web. Il s'assure qu'il y ait toujours un certain nombre de services web « actifs » en même temps. On ne peut pas démarrer

la totalité des services web au début de la simulation (il y aurait trop de threads actifs). On démarre donc uniquement le *WebServiceStarter* qui gèrera l'exécution des services web.

La classe *Stats* contient diverses méthodes utiles pour calculer différentes données de type statistiques, nécessaires pour les graphiques.

La classe *ISComparator* permet de comparer deux services d'information sur base de leurs réputations.

## A.4 Fonctionnement global d'une simulation

Lorsqu'on démarre une simulation, le simulateur commence à vérifier la consistance des données de la simulation : les catégories de services web doivent constituer 100% de la population ensemble, ni plus ni moins. Les valeurs hors du cadre des paramètres ne sont pas possibles à insérer : par exemple il est impossible de mettre un facteur de patience supérieur à 1, un  $\pi$  négatif, etc. La valeur du paramètre sera remise par défaut si une valeur interdite est insérée manuellement. Tous les paramètres ne sont pas vérifiés, le programme étant destiné à des utilisateurs spécifiques, qui savent ce qu'ils font. Le programme va ensuite créer chaque communauté, chaque service d'information et chaque service web. Le graphique montrant la répartition des qualités de service des services web est instancié et exposé.

Le *WebServiceStarter* est démarré. Les itérations du simulateur commencent alors. Il y en aura autant qu'il y aura d'essais d'entrée d'un service web dans une communauté. S'il y a 100 services web et qu'ils n'essaient de rentrer qu'une seule fois dans une communauté, il y aura 100 itérations. S'ils peuvent réessayer une fois, le nombre de simulations variera selon qu'ils sont acceptés directement ou non.

Chaque service web est un thread instancié par le *WebServiceStarter*. Lorsqu'il demande à rentrer dans une communauté, la méthode est appelée et la communauté choisie (de façon aléatoire) va alors appeler des services d'information qu'elle connaît, et par ordre de préférence (selon leur réputation), afin d'obtenir des évaluations sur le service web désirant entrer.

Ces services d'information vont alors appeler la méthode calculant le choix qu'ils vont faire : mentir ou dire la vérité. Cette méthode va, selon le nombre de coups à l'avance pris en compte (fixé lors du paramétrage), simuler un jeu répété (ou un jeu stratégique s'il n'y a qu'un coup de paramétré). Pour ce faire, la méthode prend en compte la situation actuelle, et la répète le nombre de fois nécessaires. Seule la réputation du service d'information appelant change au cours des simulations, et ses paiements totaux. Étant donné qu'il y a à chaque coup (chaque tour) 4 cas possibles (vérité-vérité, vérité-mensonge, mensonge-vérité, mensonge-mensonge, comme dans les tableaux des jeux), le service d'information va à chaque coup simuler ces 4 cas possibles. Après 3 tours, il y aura  $4^3$  possibilités. La

## A.5 Interface utilisateur

---

première sera celle où il y aura 3 « vérité-vérité » d'affilée, la deuxième 2 « vérité-vérité » et 1 « vérité-mensonge », et ainsi de suite. A la fin des tours, chaque branche de l'arbre ainsi formé aura un certain montant total, selon les paiements reçus en fonction des choix et de la réputation courante du service d'information (mise à jour selon les choix réalisés). Le service d'information calcule ces paiements avec ce qu'il connaît : sa propre réputation, la mise-à-jour de celle-ci et la façon de payer de la communauté. Le service d'information choisit son action qui rapporte à plus ou moins long terme (ou à très court terme s'il s'agit d'un « one-shot ») le plus de gains. Il choisit la première action de la branche menant à ces gains.

Ensuite, les services d'information, après avoir choisi leur action, renvoient une évaluation correcte ou fautive à la communauté. Celle-ci calcule alors la moyenne et décide si elle accepte ou non le service web désirant entrer dans sa communauté. S'il est refusé et qu'il n'a pas de possibilité de ré-essai, il abandonne, sinon il réessayera plus tard. Le maître de communauté rémunère alors les services d'information à valeur de leur conseil avec les méthodes de calcul des différentes composantes de la fonction de paiement, et, si le service web est accepté, décide s'il considère que le conseil est un mensonge ou non, et met à jour la réputation des services d'information. Pour rappel, le maître décide qu'une notation transmise est fautive (et donc que le service d'information a menti) si celle-ci ne se trouve pas dans la même zone que l'évaluation faite par le maître. Nous considérons deux zones de notations : la première contient les notations inférieures au seuil d'acceptation (0,5 par défaut), et la seconde contient les notations supérieures ou égales au seuil d'acceptation. Les graphiques de paiements, de réputation, de décision (mensonge ou vérité), et d'évolution des paiements de chaque service d'information interpellé sont mis à jour. Les graphiques de décision de la communauté, du nombre total de mensonges et de vérités sont également mis à jour à chaque décision sur un service web.

## A.5 Interface utilisateur

L'interface graphique permet de paramétrer une simulation, et d'observer les résultats. Nous allons présenter rapidement cette interface. Toutes les valeurs mises dans les paramètres montrés sont laissées par défaut.

### A.5.1 Les paramètres

Tout d'abord il y a le premier écran (figure A.1), celui sur lequel nous arrivons lorsque nous exécutons le programme. A gauche de cet écran, nous avons la paramétrisation des communautés. Nous pouvons fixer le nombre de communautés que l'on désire. Pour chaque communauté, nous fixons le nombre de services d'information qu'elle utilise lors d'une demande de conseil, la réputation minimum d'un service web pour que celui-ci soit acceptable pour la communauté, et la réputation minimum d'un service d'information pour que celui-ci soit consultable par la communauté.

Au milieu de l'écran se trouve les paramètres à propos des services web. Le paramètre concernant le nombre de services web dans la simulation, le nombre maximum de tentatives

que ceux-ci peuvent faire pour rentrer dans la communauté, et le nombre de catégories de services web. Par défaut, il n'y en a qu'une, et dans cet écran nous pouvons voir que l'on peut définir son intervalle de qualité de service, le pourcentage de la population totale de services web appartenant à la catégorie, la variabilité de la qualité de service et enfin la valeur du paiement  $\pi$  donné si cette catégorie concerne des services web malhonnêtes.

Dans la partie droite de l'écran se trouvent les paramètres sur les services d'information. On fixe leur nombre, et ensuite leurs facteurs de patience. Trois boutons supplémentaires sont disponibles :

- tous à 1 : tous les services d'information ont un facteur de patience égal à 1 ;
- de 0,1 à 1 : le facteur de patience est de 1 pour le premier service d'information, 0,9 pour le deuxième et ainsi de suite jusqu'au dixième qui a un facteur de patience égal à 1 ;
- de 0.6 à 1 : les deux premiers services d'information ont un facteur de patience de 1, les deux suivants ont 0.9 et ainsi de suite jusqu'aux deux derniers qui ont une patience de 0,6.

Ces trois boutons ont été ajoutés pour faciliter les expérimentations du mémoire, et ne sont applicables que pour 10 services d'information.

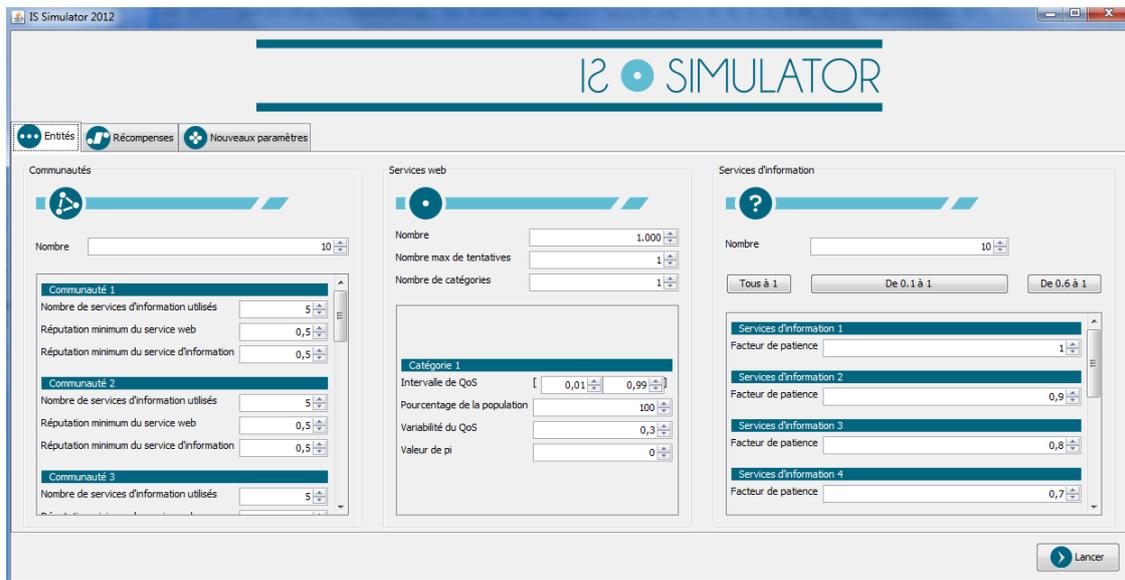


FIGURE A.1 – L'écran des paramètres sur les populations d'agents

Le deuxième écran (figure A.2) sert à paramétrer les récompenses que les maîtres de communauté accordent aux services d'information consultés.  $\alpha$  représente le paiement fixe  $f_k(x)$ .  $\beta$ -max représente le paiement  $g_k(x)$  maximum.  $\gamma$ -max représente le paiement  $h_k(x)$  maximum.  $\beta$ -cr représente la constante de retrait pour  $g_k(x)$ .  $\gamma$ -cr représente la constante de retrait pour  $h_k(x)$ .  $\pi$ -step est l'augmentation du paiement  $\pi$  que propose un service web malhonnête à chaque fois qu'il réessaye d'entrer dans une communauté

Le dernier écran (figure A.3) de configuration est celui des paramètres ajoutés par

## A.5 Interface utilisateur



FIGURE A.2 – L'écran des paramètres sur les paiements

rapport au travail de 2010-2011. La « probabilité P » est la probabilité avec laquelle un maître de communauté décide d'accepter un service web alors qu'il est déconseillé par les services d'information. Le « nombre max de coups » est le nombre de coups joués à l'avance par un service d'information lorsqu'il essaie de prendre une décision, c'est-à-dire le nombre de tours des jeux répétés simulés. La « probabilité de corruption par un tiers » est la probabilité qu'un tiers décide de corrompre les services d'information lorsqu'il ne désire pas qu'un bon service web rentre dans la communauté. La « valeur de  $\pi$  en cas de corruption par un tiers » est, comme son nom l'indique, le paiement  $\pi$  que ce tiers donnera aux services d'information pour les corrompre.

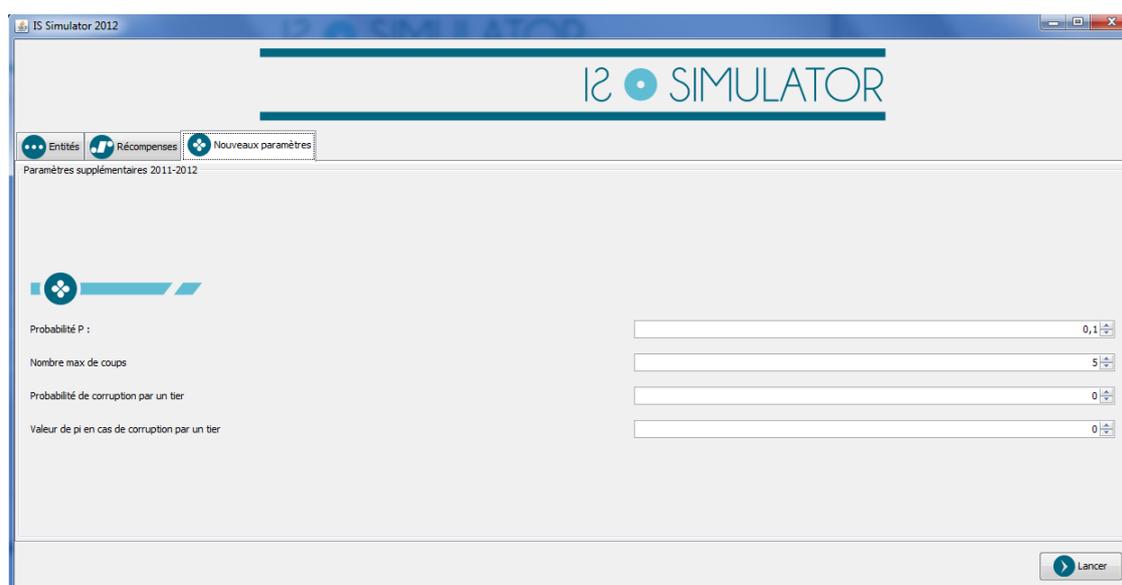


FIGURE A.3 – L'écran des paramètres supplémentaires

## A.5.2 Les graphiques résultants d'une simulation

### Les graphiques généraux

Ce premier écran affiche la répartition de la qualité de service des services web. Les numéros des services web sont en abscisse et la qualité de service en ordonnée. La barre verte sert à afficher la progression de la simulation, ici nous voyons qu'elle est terminée. La zone de texte affiche la création des entités, et l'acceptation ou le rejet de chaque service web, et quand chaque thread représentant un service web se termine.

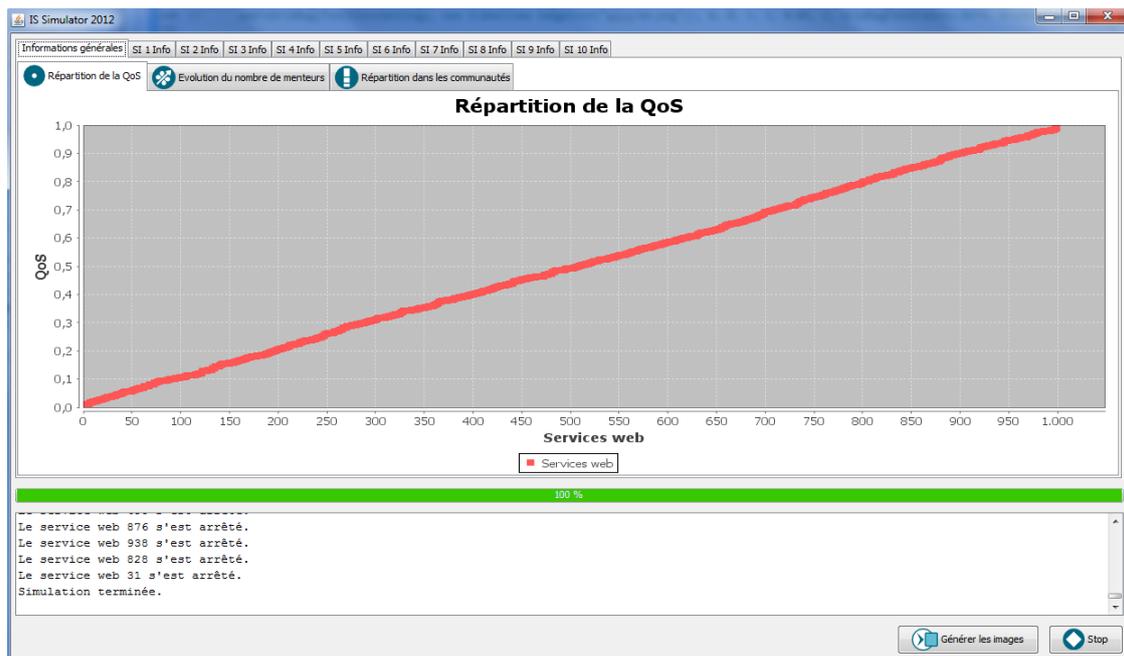


FIGURE A.4 – L'écran des résultats généraux sur les QoS des services web

Le deuxième écran (figure A.5) montre l'évolution du nombre de mensonges et de vérités, avec en abscisse la distinction entre mensonge ou vérité, et en ordonnée le nombre total.

Le troisième et dernier écran concernant les graphiques généraux (figure A.6) affiche la répartition des décisions dans les communautés. Nous voyons en abscisse les différents types de décisions, et en ordonnée leur répartition. Par exemple la communauté 1 à de l'ordre de 10% d'acceptation à tort, 50% d'acceptation correcte, 3% de refus à tort et 37% de refus correct.

### Les graphiques concernant un service d'information

Chaque service d'information a 4 graphiques. Des onglets permettent de sélectionner un service d'information.

Le premier écran affiche ses choix le long des itérations auxquelles il a participé. En abscisse sont numérotées les itérations, et en ordonnée 1 si il a dit la vérité à l'itération

## A.5 Interface utilisateur

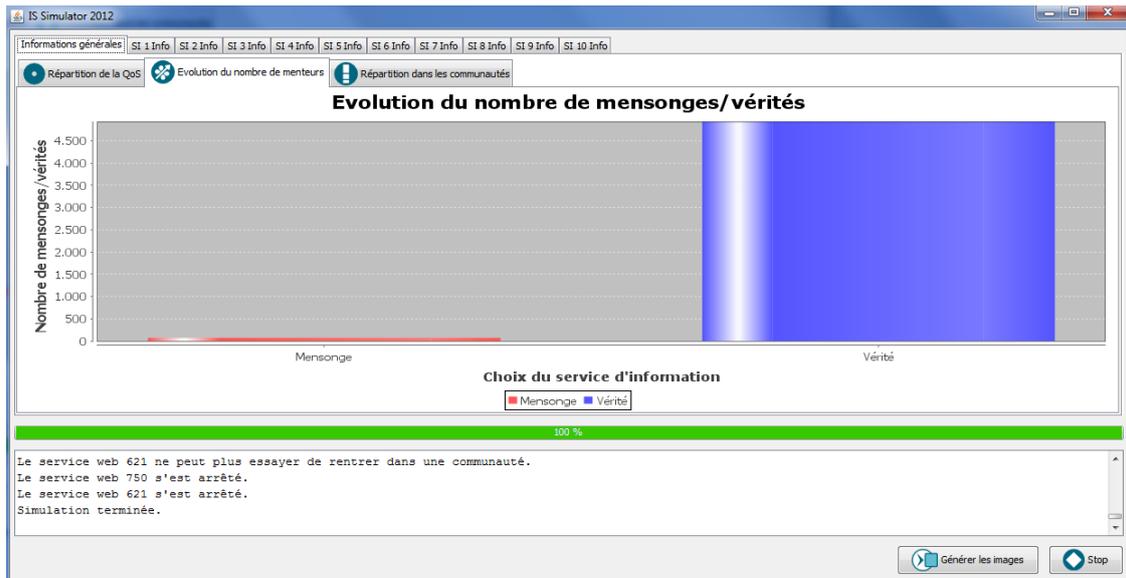


FIGURE A.5 – L'écran des résultats généraux sur le nombre de mensonges et de vérités

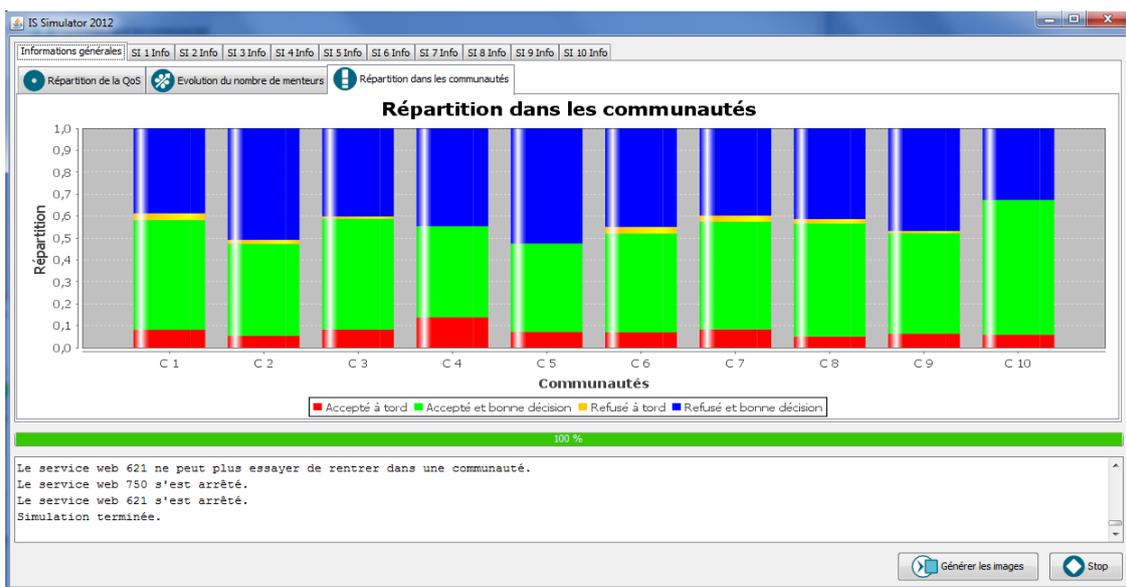


FIGURE A.6 – L'écran des résultats généraux sur les décisions des communautés

correspondante, ou -1 sinon.

Le deuxième écran (figure A.8) affiche l'évolution de la réputation du service d'information, pour chaque communauté. Nous voyons qu'il y a beaucoup plus d'itérations que pour le graphique précédent. C'est parce que ce graphique est généré à chaque itération, même celles auxquelles le système d'information n'a pas participé, elles sont toutes prises en compte et nous pouvons voir les liaisons entre les points, ces points correspondant à des interactions.

Le troisième écran (figure A.9) affiche l'évolution des paiements du service d'informa-

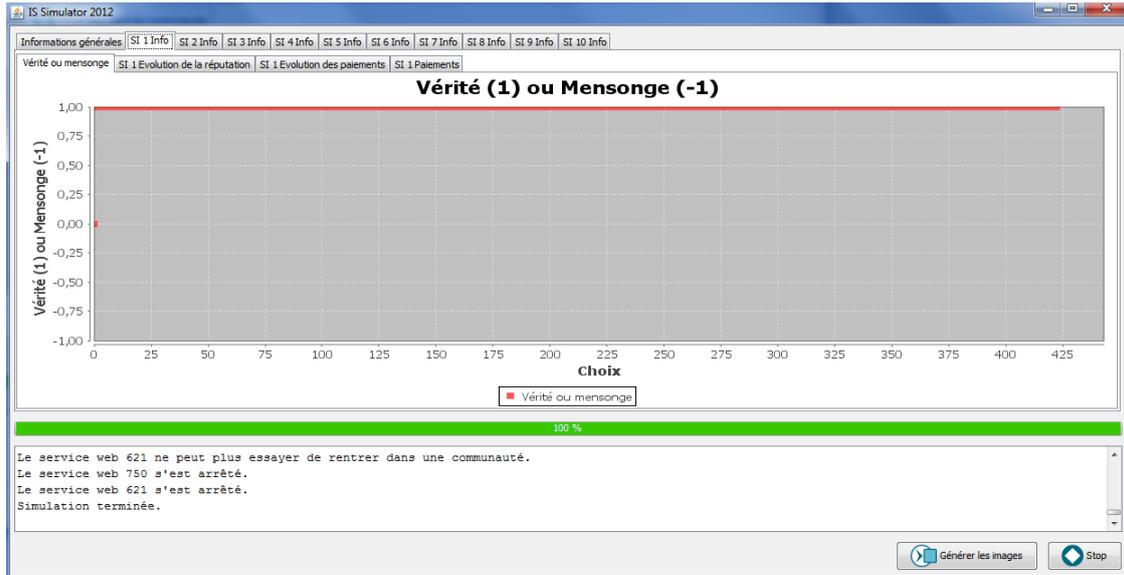


FIGURE A.7 – L'écran d'un service d'information sur les choix à chaque participation au conseil

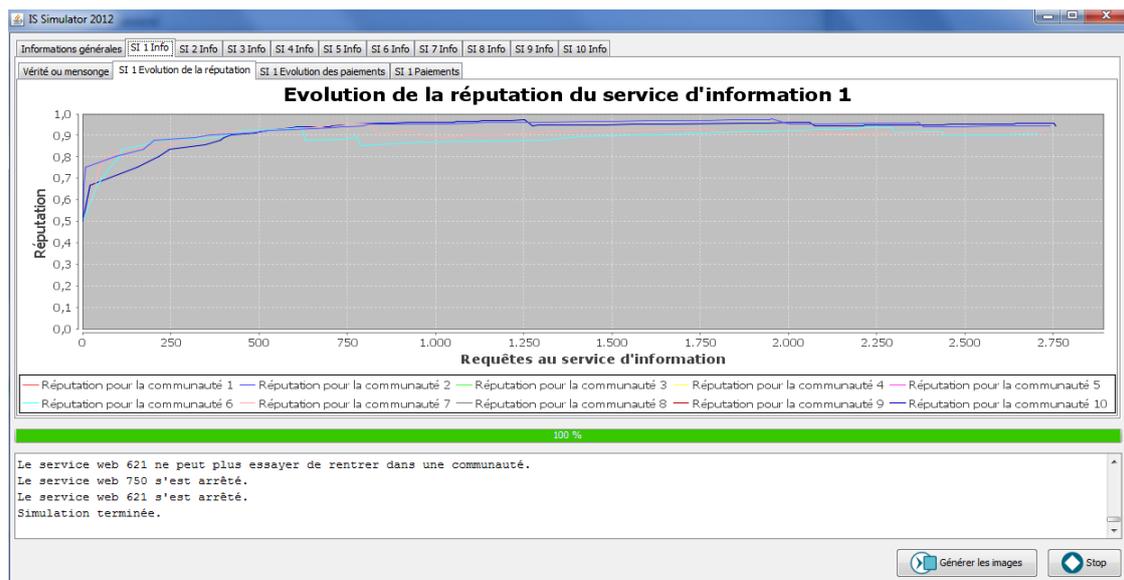


FIGURE A.8 – L'écran d'un service d'information sur l'évolution de sa réputation au fil des itérations et par communauté

tion. En abscisse les numéros des itérations auxquelles il a participé, et en ordonnée le montant des différents paiements. Le paiement  $\beta$  opposé représente le montant  $g_k(x)$  que le service d'information aurait reçu s'il avait choisi l'action contraire. Le paiement  $\gamma$  opposé représente le montant  $h_k(x)$  qu'il aurait reçu en choisissant l'action opposée.

Le dernier écran (figure A.10) affiche la cumulation des paiements que le service d'information a reçu. En abscisse les numéros des itérations auxquelles il a participé, et en ordonnée le montant des paiements cumulés déjà reçus à l'itération correspondante.

## A.5 Interface utilisateur

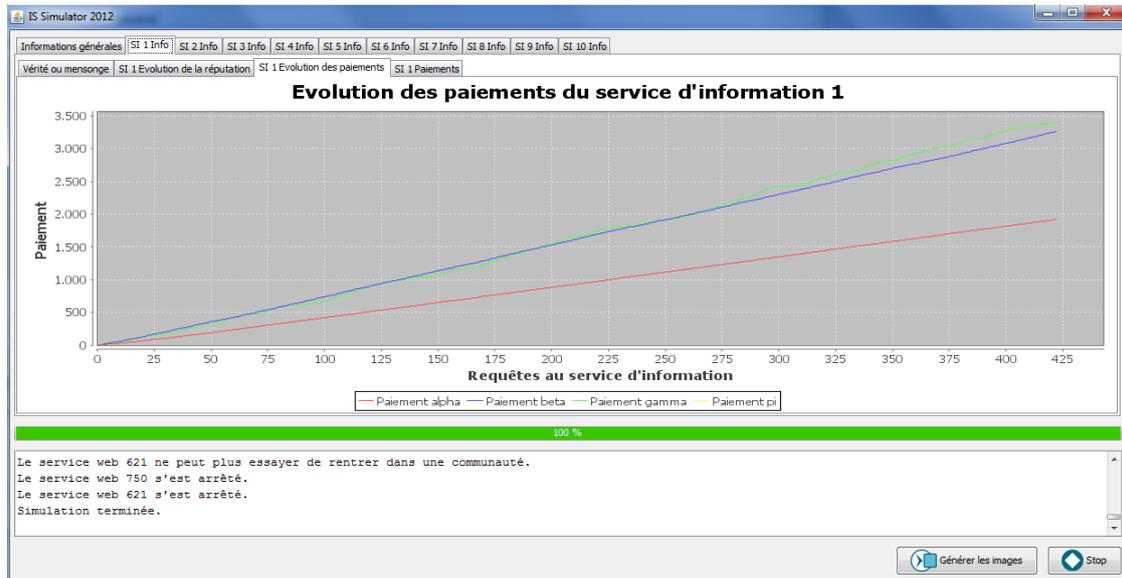


FIGURE A.9 – L'écran d'un service d'information sur l'évolution de ses paiements au fil des itérations

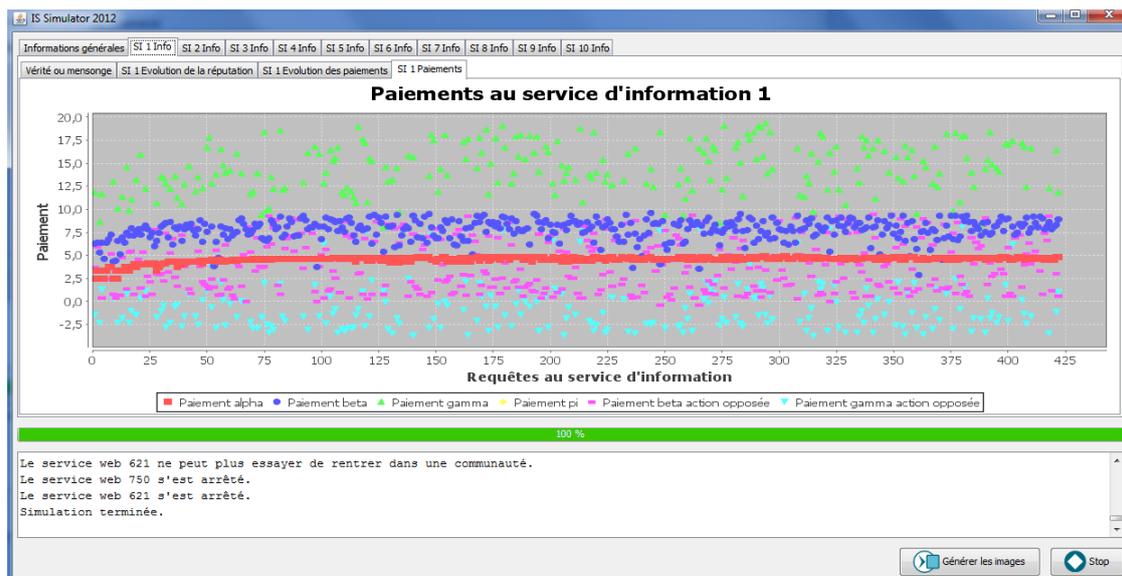


FIGURE A.10 – L'écran d'un service d'information sur ses paiements à chaque itérations

