

Casanova : un comportement d'agent pour l'équité des mariages préservant la privacité

P. Everaere^a
patricia.everaere@univ-lille1.fr

M. Morge^a
maxime.morge@univ-lille1.fr

G. Picard^b
picard@emse.fr

^aLaboratoire d'Informatique Fondamentale de Lille,
Université Lille 1, France

^bDépartement ISCOD / Institut Henri Fayol
École Nationale Supérieure des Mines de Saint-Etienne, France

Résumé

Cet article prône une approche orientée individu pour la résolution du problème classique des mariages stables. Selon cette approche, la solution émerge des négociations entre agents. Par exemple, l'algorithme de Gale-Shapley (GS) distingue deux comportements d'agents (proposant et disposant) qui négocient pour aboutir à une solution stable, pareto-optimale mais inéquitable : la communauté des proposant est favorisée. Nous proposons ici l'algorithme Casanova qui met en œuvre une stratégie de concession minimale. Les agents jouent simultanément le rôle de disposant et de proposant dans une multitude de négociations bilatérales. De plus, les solutions qui émergent ne peuvent être atteintes par GS et elles sont plus équitables. Notre implémentation est décentralisée et préserve la privacité.

Mots-clés : *Modèle de comportement d'agents, Théorie des jeux, Résolution décentralisée*

Abstract

This paper promotes an individual-based approach for solving a well-known problem, the stable marriage problem. In this approach, a solution is the output of an emergent phenomena due to the negotiation between the agents. For instance, the Gale-Shapley (GS) algorithm implies two distinct behaviours (the proposer and the responder) to reach a stable matching which is pareto-optimal even if it is in favour of the proposers rather than the responders. We propose here the Casanova algorithm which applies the minimal concession strategy for negotiation. Each agents play the role of proposer and responder in multi bi-lateral negotiations. Moreover, the emergent solutions, which are more equitable, cannot be reached by GS. Our implementation is privacy-preserving and decentralized.

Keywords: *Agent behaviour, Game Theory, Decentralized Problem Solving*

1 Introduction

Les systèmes multi-agents proposent un cadre conceptuel et des outils informatiques particulièrement adaptés à la modélisation et la simulation de phénomènes complexes observés en Physique, en Biologie ou en Economie [13]. L'approche orientée individu fait intervenir différentes entités en interaction dont le comportement constitue un modèle explicatif de phénomènes émergents.

Nous nous intéressons dans cet article à un problème bien connu en Informatique [6] et en Economie [12], celui des mariages stables - en anglais, *Stable Marriage Problem* (SM). L'enjeu de ce problème est simple. En considérant deux communautés dont chaque membre à des préférences vis-à-vis des individus de l'autre communauté, l'objectif est de former des couples en prenant en compte leurs préférences. Ce problème abstrait a de nombreuses applications. Par exemple, pour affecter des projets à des étudiants, chaque étudiant classe par ordre de préférence les projets et réciproquement le responsable de chaque projet classe lui aussi les candidats potentiels. Le co-voiturage, l'échange de domicile ou les réseaux d'entreprise sont d'autres exemples de domaines d'applications.

Dans la perspective d'une approche orientée individu, l'algorithme séminal de Gale-Shapley [5] permet de résoudre le problème SM en distinguant deux comportements d'agents distincts : une communauté de proposant et une communauté de disposants. Les négociations entre ces agents aboutissent à une solution stable mais inéquitable : la communauté des proposant est favorisée. Dans cet article, nous proposons l'algorithme Casanova où les agents jouent simultanément le rôle de disposant et de proposant dans une multitude de négociations bilatérales. Notre approche fait émerger des solutions stables qui ne peuvent être at-

teintes par l’algorithme de Gale-Shapley, et qui sont plus équitables entre les deux communautés. Notre implémentation est décentralisée et elle préserve la confidentialité, en effet les préférences des agents ne sont pas révélées à une entité centrale qui les agrège.

Nous commençons cet article par un aperçu de la littérature (cf. section 2). Nous formalisons ensuite le problème SM dans la section 3 puis nous présentons l’algorithme de Gale-Shapley dans une perspective orientée individu (cf. section 4). Afin d’évaluer les solutions à un problème SM nous utilisons la théorie des jeux (cf. section 5) et la théorie du choix social (cf. section 6). Notre algorithme Casanova est décrit dans la section 7. Nous présentons le résultat de nos expérimentations dans la section 8. Nous terminons cet article par une discussion.

2 Travaux connexes

Comme l’indique la Table 1, l’algorithme de Gale-Shapley est un algorithme centralisé qui favorise une communauté au détriment d’une autre. Comme nous le présenterons dans la section 4, cet algorithme peut être naturellement envisagé selon une approche centrée individu (DisEGS). [2, 3] propose également une méthode distribuée de résolution de contrainte où la formulation des contraintes respecte la confidentialité (dans la version PKC) mais n’assure pas l’équité à cause de l’ordre total entre hommes et femmes, nécessaire à la bonne conduite de l’algorithme. L’algorithme Casanova [8] qui s’inspire de DisEGS vise à améliorer l’équité entre les communautés.

Méthode	Décentralisé	Privacité	Équité
GS [5]	✗	✗	✗
DisEGS [2]	✓	✓	✗
DisFC [2, 3]	✓	✓	✗
Casanova [8]	✓	✓	✓

TABLE 1 – Méthode de résolution du SM

3 Problème du mariage stable

Nous étudions ici le problème des mariages stables, en anglais *Stable Marriage Problem*, qui a été introduit par Gale et Shapley [5]. Ce problème s’énonce simplement. On considère n individus d’une communauté (e.g. la gent féminine) et n individus d’une autre communauté (e.g. la gent masculine). Chaque individu a des penchants pour les individus de l’autre partition avec lesquels il souhaite être apparié.

Définition 1 (SM) Un *problème de mariage stable* de taille n (avec $n \geq 1$) est un couple $SM = \langle X, Y \rangle$ avec $|X| = |Y| = n$, où :

- $X = \{x_1, \dots, x_n\}$ est un ensemble de n relations de préférence sur Y , représentant les préférences des hommes sur l’ensemble des femmes.
- $Y = \{y_1, \dots, y_n\}$ est un ensemble de n relations de préférence sur X , représentant les préférences des femmes sur l’ensemble des hommes.

On note $y_2 \succ_{x_1} y_3$ le fait que l’homme x_1 préfère strictement la femme y_2 à la femme y_3 . Dans cet article, on suppose que les relations de préférences sont des ordres stricts, ce qui signifie que chaque individu est en mesure de choisir entre deux partenaires (il n’y a ni ex æquo, ni indifférence). On appelle problème de mariage stable avec liste complète (SMC) un problème SM où toutes les préférences sont complètes (tous les individus sont classés). Par opposition, dans un problème de mariage stable incomplet (SMI), au moins l’une des listes de préférence est incomplète. Le fait qu’une liste de préférences soit incomplète signifie que l’individu préfère rester seul plutôt que d’être mal accompagné.

Résoudre un problème SM consiste à assortir ces communautés. On appelle appariement, en anglais *matching*, un ensemble de n mariages monogames et hétérosexuels.

Définition 2 (Appariement) Un *appariement* M pour le problème $SM = \langle X, Y \rangle$ de taille n est une application $\mu_M : X \cup Y \rightarrow X \cup Y \cup \{\theta\}$ ¹ telle que :

- $\forall x \in X, \mu_M(x) \in Y \cup \theta$ et $\forall y \in Y, \mu_M(y) \in X \cup \theta$
- $\forall z \in X \cup Y$, si $\mu_M(z) \neq \theta$ alors $\mu_M(\mu_M(z)) = z$

$\mu_M(z)$ représente le partenaire de z selon l’appariement M . Dans un appariement complet, chaque individu est marié et l’application μ est involutive et donc bijective. $\mu_M(z) \succ_z \mu_{M'}(z)$ signifie que z préfère (son partenaire dans) l’appariement M à (son partenaire dans) M' . $\mu_M(z) \succeq_z \mu_{M'}(z)$ si $\mu_M(z) \succ_z \mu_{M'}(z)$ ou $\mu_M(z) = \mu_{M'}(z)$. Dans le cas d’un problème SMC, seuls les appariements complets sont intéressants. À l’inverse, pour un problème SMI, certains appariements complets sont irrationnels car ils consistent à marier un individu à un partenaire qui n’est pas dans sa liste de préférences.

1. θ dénote l’individu fantôme

Un appariement est stable si il n'existe pas de couple qui préférerait être ensemble plutôt qu'avec leur partenaire actuel. En d'autres termes, il n'existe pas de relation extra-conjugale menaçant l'appariement. Ces relations prennent la forme de couples hypothétiques bloquants :

Définition 3 (Stabilité) Soient un problème $SM = \langle X, Y \rangle$ de taille n et M un appariement pour SM . Un couple $(x_i, y_i) \in X \times Y$ est **bloquant** si $y_i \succ_{x_i} \mu_M(x_i)$ et $x_i \succ_{y_i} \mu_M(y_i)$. M est **instable** s'il existe un couple bloquant.

Un appariement stable est une solution admissible à un problème SMC. Dans le cas d'un problème SMI, un appariement stable est admissible s'il est rationnel, c.-à-d. si chacun des individus est marié à un partenaire qui est dans sa liste de préférences.

Exemple 1 Soit les relations de préférence suivantes $SM = \langle \{x_1, x_2, x_3\}, \{y_1, y_2, y_3\} \rangle$ avec :

$$\begin{array}{ll} y_2 \succ_{x_1} y_1 \succ_{x_1} y_3 & x_2 \succ_{y_1} x_1 \succ_{y_1} x_3 \\ y_3 \succ_{x_2} y_2 \succ_{x_2} y_1 & x_3 \succ_{y_2} x_2 \succ_{y_2} x_1 \\ y_1 \succ_{x_3} y_3 \succ_{x_3} y_2 & x_1 \succ_{y_3} x_3 \succ_{y_3} x_2 \end{array}$$

Pour ce SMC, l'appariement M ($\mu_M(x_1) = y_2$, $\mu_M(x_2) = y_1$, $\mu_M(x_3) = y_3$) est instable car (x_2, y_2) est un couple bloquant. À l'inverse, les trois appariements M_1 , M_2 et M_3 sont stables :

$$\begin{array}{l} - \mu_{M_1}(x_1) = y_1, \mu_{M_1}(x_2) = y_2, \mu_{M_1}(x_3) = y_3 \\ - \mu_{M_2}(x_3) = y_1, \mu_{M_2}(x_1) = y_2, \mu_{M_2}(x_2) = y_3 \\ - \mu_{M_3}(x_2) = y_1, \mu_{M_3}(x_3) = y_2, \mu_{M_3}(x_1) = y_3 \end{array}$$

4 Algorithme de Gale-Shapley

L'algorithme de Gale-Shapley [5] (GS) est une preuve constructive de l'existence d'une solution admissible pour n'importe quelle instance de SM (complet ou non).

L'algorithme GS peut être décrit comme un cérémonial au cours duquel les femmes attendent dans une salle et les hommes qui sont à l'extérieur, rentrent à tour de rôle dans un ordre arbitraire. Quand un homme entre, il se présente à la femme qu'il préfère. Si la femme est libre et qu'elle le considère comme acceptable (i.e. il est dans sa liste de préférences), il devient son partenaire. Si elle a déjà un partenaire, il y a conflit et elle choisit l'homme qu'elle préfère. L'élu reste et l'autre, rejeté, sort de la pièce. Ce dernier se souvient de cet échec et reviendra plus tard pour se présenter à la fille qui suit dans l'ordre de ses préférences. On s'arrête quand

tous les hommes qui ne sont pas libres sont désespérés (leur liste de préférences est vide).

Dans l'algorithme GS (cf Algo. 1) tel que nous le présentons, les hommes proposent et les femmes disposent. Une exécution consiste en une séquence de propositions des hommes envers les femmes. En d'autres termes, les hommes jouent le rôle de proposant et les femmes jouent le rôle de répondant. On dit que GS est orienté homme. Il peut être facilement adapté pour être orienté femme, en inversant leur rôle respectif. Dans ce cas, les femmes proposent et les hommes disposent.

Exemple 2 En considérant le SM de l'exemple 1, le résultat de l'algorithme GS orienté homme (respectivement orienté femme) est l'appariement M_2 (respectivement M_3). L'appariement M_1 ne peut être atteint par aucune exécution de GS qu'il soit orienté homme ou femme.

5 Optimalité sociale

Une fois qu'un appariement est établi, il est essentiel de déterminer à quel point cet appariement est acceptable pour la société. Dans cette section, nous utilisons la théorie des jeux [12] pour évaluer les solutions admissibles à un problème SM. Pour la clarté, nous nous restreignons dans la suite de cet article au problème avec liste complète (SMC). Nous évaluons alors les solutions admissibles, i.e. celles qui sont stables.

Un appariement stable est le résultat d'un jeu entre deux communautés. Il peut être évalué du point de vue d'une communauté ou de l'autre.

Définition 4 (Préférence communautaire)

Soit un problème $SM = \langle X, Y \rangle$ de taille n . Soient M et M' deux appariements stables pour SM et $E \subseteq (X \cup Y)$. M **pareto-domine** M' sur E ssi $\forall z \in E \mu_M(z) \succeq_z \mu_{M'}(z)$ et $\exists z \in E$ tq $\mu_M(z) \succ_z \mu_{M'}(z)$. Un appariement est **pareto-optimal** si c'est un appariement stable qui n'est pas pareto dominé sur $X \cup Y$. Un appariement est pareto **mâle-optimal** (respectivement pareto **femelle-optimal**) si c'est un appariement stable qui n'est pas pareto-dominé sur X (respectivement Y).

Donald Knuth a démontré dans [6] que l'exécution de l'algorithme GS orienté homme aboutit à une solution –un appariement stable– qui est mâle-optimale et que cette solution est une des pires du point de vue des femmes.

Données : SM

Résultat : M

Tous les individus sont libres;

tant que *il existe un homme libre x qui ne soit pas désespéré faire*

$y \leftarrow$ la première femme de la liste de x ;

 // x propose à y ;

$x_2 \leftarrow$ est le partenaire courant de y éventuellement θ ;

si y considère x comme acceptable **alors**

 └ Marier x et y ; // y dispose;

si x_2 n'est pas un fantôme (i.e. θ) **alors**

 └ $x_2 \leftarrow$ libre;

pour chaque successeur x_3 de x dans la liste de y **faire**

 └ Supprimer x_3 de la liste de y ; // y ne veut plus concéder;

 └ Supprimer y de la liste de x_3 ; // y informe ceux concernés;

Algorithme 1 – GS orienté homme

Théorème 1 (Conflit global [6]) *Soit un problème SM.*

- *Il y a toujours un appariement qui est mâle-optimal, i.e. le résultat de l'algorithme GS orienté homme.*
- *Il y a toujours un appariement qui est femelle-optimal, i.e. le résultat de l'algorithme GS orienté femme.*
- *Tous les appariements qui sont mâle-optimaux sont pareto-dominés sur les femmes par les autres appariements stables.*

Comme l'espace des versions bien connu en apprentissage automatique, l'espace des solutions stables est un treillis. L'espace que l'on envisage ici ne s'appuie pas sur une relation de généralisation mais sur la relation de pareto-dominance (e.g. sur les femmes) et la frontière supérieure est constituée des mâle-optimaux alors que la frontière inférieure est l'ensemble des solutions femelle-optimales. Chacune des solutions frontières peut être atteinte par l'algorithme GS en modifiant l'ordre d'apparition des proposants. Toutes les autres solutions, hors des frontières, ne peuvent être atteintes par l'algorithme GS mais par un algorithme centralisé qui passe d'un appariement stable à un autre appariement stable en dégradant la satisfaction d'une communauté ou de l'autre communauté.

Dans une situation stable, si un individu souhaite divorcer pour un partenaire qu'il préfère alors ce coup s'effectue au détriment de son partenaire courant. Ces dilemmes sont capturés par le théorème suivant.

Théorème 2 (Conflit local [6]) *Soient M et M' deux appariements stables pour un problème*

SM. Si $\mu_M(x) = y$ et $\mu_{M'}(x) \neq y$, alors

- *soit $\mu_{M'}(x) \succ_x \mu_M(x)$ et $\mu_M(y) \succ_y \mu_{M'}(y)$;*
- *soit $\mu_M(x) \succ_x \mu_{M'}(x)$ et $\mu_{M'}(y) \succ_y \mu_M(y)$.*

La stabilité est une condition nécessaire à la pareto-optimalité mais ce n'est pas une condition suffisante.

Théorème 3 (Pareto optimalité) *Tout appariement stable est un optimum de Pareto.*

Preuve 1 (Pareto optimalité) *Par l'absurde, supposons qu'un appariement stable M n'est pas pareto-optimal. Alors, il existe un appariement M' qui pareto-domine M sur $X \cup Y$. En conséquence,*

$$\exists z \in X \cup Y \text{ t.q. } \mu_{M'}(z) \succ_z \mu_M(z) \quad (1)$$

et

$$\forall z' \neq z, \mu_{M'}(z') \succeq_{z'} \mu_M(z') \quad (2)$$

Notons $h = \mu_M(z)$ et $h' = \mu_{M'}(z)$. Avec 1, on a $\mu_{M'}(z) \succ_z \mu_M(z)$ et on peut déduire que $h' \succ_z h$. Comme $h' \neq z$, grâce à 2, $\mu_{M'}(h') \succeq_{h'} \mu_M(h')$, donc $z \succeq_{h'} \mu_M(h')$. Comme l'ordre considéré est strict et complet et que $\mu_M(h') \neq z$, on a $z \succ_{h'} \mu_M(h')$. M n'est pas stable car (z, h') est bloquant.

Exemple 3 *En considérant le SM de l'exemple 1, l'appariement M_2 (respectivement M_3) est mâle optimal (respectivement femelle optimal) car les hommes (respectivement femmes) sont mariés avec leur partenaire préféré. L'appariement M_1 , où tous les individus sont mariés avec leur second partenaire préféré, est un optimum de pareto comme M_2 et M_3 .*

En revanche, un appariement pareto-optimal n'est pas nécessairement stable.

Théorème 4 *Tout appariement pareto-optimal n'est pas stable.*

Preuve 2 Soit le $SM = \langle \{x_1, x_2\}, \{y_1, y_2\} \rangle$ avec $y_2 \succ_{x_1} y_1$, $y_2 \succ_{x_2} y_1$, $x_2 \succ_{y_1} x_1$ et $x_2 \succ_{y_2} x_1$. Considérons l'appariement M tel que $\mu_M(x_1) = y_2$ et $\mu_M(x_2) = y_1$. M est pareto-optimal mais cet assignement n'est pas stable car le couple (x_2, y_2) est bloquant.

En résumé, l'optimalité au sens de Pareto ne nous permet pas de distinguer parmi les solutions stables lesquelles sont les meilleures du point de vue de la société globale.

6 Bien-être social

Afin d'évaluer la qualité d'une solution du point de vue collectif, la théorie du choix social [9] propose d'autres outils que la théorie des jeux. Cette évaluation repose sur une mesure individuelle de la satisfaction des individus, qui est ensuite agrégée afin d'estimer la satisfaction de toute (ou partie de) la société.

Pour le problème SM , chaque individu (quelle que soit sa communauté) évalue sa satisfaction en fonction de ses préférences via son regret qui est : 0 s'il est marié avec le premier individu de sa liste ; 1 s'il est marié avec le deuxième individu de sa liste, etc. Moins un individu a de regret, plus grande est sa satisfaction.

Définition 5 (Satisfaction individuelle) Soient un problème $SM = \langle X, Y \rangle$ de taille n et un individu z . Considérons l'appariement M . Si le rang de $\mu_M(z)$ dans z est k (avec $0 \leq k < n$), alors le **regret de z dans l'appariement M** (noté $r_z(\mu_M(z))$) est k . La **fonction d'utilité de l'individu z** est la fonction $u_z : X \cup Y \rightarrow [0, 1]$ définie telle que :

$$u_z(\mu_M(z)) = \begin{cases} \frac{(n-1)-k}{n-1} & \text{si } \mu_M(z) \neq \theta \\ 0 & \text{sinon} \end{cases} \quad (3)$$

[9] propose de nombreuses fonctions pour agréger les utilités des individus. Dans la cadre du problème SM , nous considérons le point de vue utilitaire, celui des hommes et celui des femmes. De plus, nous introduisons ici la mesure de bien-être social équitable afin d'évaluer les disparités dans une solution.

Définition 6 (Satisfaction collective) Soit un problème SM de taille n . Considérons un appariement M .

- Le **bien-être utilitaire** considère le bien-être de toute la société : $sw_u(X \cup Y) = \frac{1}{2n} \sum_{z \in X \cup Y} u_z(\mu_M(z))$.
- Le **bien-être masculin** considère la satisfaction des hommes : $sw_u(X) = \frac{1}{n} \sum_{x \in X} u_x(\mu_M(x))$.
- Le **bien-être féminin** considère la satisfaction des femmes : $sw_u(Y) = \frac{1}{n} \sum_{y \in Y} u_y(\mu_M(y))$.
- Le **bien-être équitable** considère l'équité entre les hommes et les femmes : $sw_e(X \cup Y) = 1 - |sw_u(X) - sw_u(Y)|$.

Notons que ces notions de bien-être ont été normalisées (entre 0 et 1). Nous avons choisi d'envisager l'équité entre les deux communautés plutôt que le bien-être égalitaire qui considère le bien-être de l'agent le plus insatisfait indépendamment de la communauté à laquelle il appartient.

Exemple 4 Si nous considérons de nouveau l'exemple 1, les bien-être pour les différents appariements stables sont les suivants :

Bien-être	M_1	M_2	M_3
$sw_u(X \cup Y)$.5	.5	.5
$sw_u(X)$.5	1	0
$sw_u(Y)$.5	0	1
$sw_e(X \cup Y)$	1	0	0

L'appariement M_1 , qui ne peut pas être atteint par l'algorithme GS, est la solution la plus équitable.

7 Algorithme Casanova

Contrairement à GS qui décrit un cérémonial, l'algorithme que nous proposons, Casanova, décrit le comportement d'agents négociant. Dans GS, chaque communauté se voit attribuer un rôle, proposant ou disposant, alors que dans Casanova, les agents jouent simultanément le rôle de disposant et de proposant dans une multitude de négociations bilatérales. Casanova réalise alors une stratégie de concession minimale [7, 11], basée sur le regret des agents, pour permettre aux agents de faire émerger un appariement stable plus équitable que la solution proposée par GS. Selon cette stratégie, les agents se proposent à leurs partenaires préférés. En cas d'échec, un agent concède, ce qui se traduit par

l'abandon de ses exigences vis-à-vis des partenaires potentiels. Ainsi, au début, un agent envoie une proposition à son unique partenaire préféré, et s'il refuse il concède et donc envoie une nouvelle proposition aux deux agents qu'il préfère, etc.

7.1 Comportement des agents

Dans Casanova, chaque individu est représenté par un agent dont l'état interne est défini comme suit.

Définition 7 Soient $SM = \langle X, Y \rangle$ un problème de taille n et un agent $a \in \mathcal{A}$ représentant l'individu $z \in X \cup Y$. À chaque instant, l'agent a est représenté par un tuple $\langle \sigma, \pi, \kappa, \mu, \lambda, \omega, \alpha \rangle$ où :

- $\sigma \in \{\top, \perp\}$ est le *statut marital* (\top si marié avec un autre agent, \perp sinon) ;
- π est la *liste de préférences* de l'agent, ordonnée grâce à la relation \succ_z (cf. Définition 1) ;
- $\kappa \in [0, n[$ est le *niveau de concession* ou *niveau de regret courant* ;
- $\mu \in X \cup Y \cup \{\theta\}$ est le *partenaire courant* ;
- $\lambda \in X \cup Y \cup \{\theta\}$ est l'*amant courant* ;
- $\omega \in [0, n]$ est un *compteur de messages sans réponse* ;
- α est la *liste des acceptants*.

Par souci de concision, on considère $z \equiv a$ et $X \cup Y \equiv \mathcal{A}$. On définit le niveau de concession comme la position maximale dans la liste de préférences que l'agent considère comme acceptable. On note qu'un agent n'est pas forcément marié avec son partenaire courant μ : il peut être fiancé ($\sigma = \perp$ et $\mu \neq \theta$). De plus, un amant λ est un agent qui fait une proposition acceptable ($r_a(\lambda) < \kappa$) à un agent a déjà marié ayant un niveau de concession κ . Le compteur ω sert à déterminer si l'agent attend encore des réponses aux propositions qu'il a lancées. La liste des acceptants α correspond à l'ensemble des agents ayant accepté une des propositions de l'agent.

Initialement, $\sigma = \perp$, $\kappa = 1$, $\mu = \theta$, $\lambda = \theta$, $\omega = 0$, $\alpha = []$. Les préférences π varient d'un agent à l'autre.

Nous pouvons représenter le comportement des agents grâce à un automate fini déterministe composé de 5 états : libre (*free*), hésitant (*hesitating*), fiancé (*engaged*), marié fidèle (*faithful*), et marié infidèle (*unfaithful*). Chacun de ces états dépend des valeurs de certains attributs de l'agent (μ , λ , σ et ω), comme présenté ci-dessous :

état	μ	λ	σ	ω
free	θ	θ	\perp	> 0
hesitating	$z \in \mathcal{A}$	θ	\perp	> 0
engaged	$z \in \mathcal{A}$	θ	\perp	$= 0$
faithful	$z \in \mathcal{A}$	θ	\top	$= 0$
unfaithful	$z \in \mathcal{A}$	$z' \in \mathcal{A}$	\top	$= 0$

Les transitions entre états et leurs conditions sont représentées dans la figure 1. Les actions consécutives aux transitions sont exprimées dans le tableau 2, pour plus de lisibilité.

Un agent peut envoyer les messages suivants :

- **propose** pour faire une proposition à un autre agent ;
- **accept** (resp. **reject**) pour répondre positivement (resp. négativement) à une proposition émanant d'un autre agent ;
- **confirm** (resp. **withdraw**) pour confirmer (resp. infirmer) le mariage, suite à une acceptation ;
- **divorce** pour briser un mariage.

7.2 Algorithme

L'algorithme Casanova correspond à l'exécution en parallèle des automates finis déterministes correspondants à la figure 1, pour chacun des agents des deux communautés. Tous les agents ont donc le même comportement, mais des préférences différentes. Chaque agent mène trois activités de front : (i) la proposition ; (ii) la disposition ; et (iii) la remise en question.

Proposition. Au début de l'algorithme, tous les agents sont libres (*free*). Chaque agent lance des propositions aux agents qu'il considère comme acceptables ($b \in \pi$, t.q. $r_a(b) < \kappa$). Lorsqu'il reçoit toutes les réponses et au moins une acceptation avant toute autre proposition, il envoie un message de confirmation au meilleur et il devient marié (*faithful*). En l'absence d'acceptation parmi toutes les réponses et en l'absence d'autre proposition, l'agent augmente son niveau de concession κ . Il recommence alors une phase de proposition avec un niveau de concession supérieur. Dès qu'un agent reçoit une proposition, il passe en phase de disposition (*hesitating*). Par contre, si à l'issue de cette activité de proposition, il est marié (donc il a reçu toutes les réponses à sa proposition, et n'a reçu aucune autre proposition), il peut se remettre en question.

Disposition. Lorsqu'un agent reçoit une proposition alors qu'il est libre et donc en attente de

origine	destination	condition	action
free	free	reject & $\omega=0$ & all are reject	$\kappa++$; send(propose) to all m peers at concession level κ ; $\alpha \leftarrow []$; $\omega \leftarrow m$;
free	free	(accept or reject) & $\omega \neq 0$	$\mu \leftarrow \text{best}(\alpha)$; send(confirm) to μ ; send(withdraw) to every $z \in \alpha \setminus \mu$; $\sigma \leftarrow \top$
free	hesitating	propose & $\text{level}(y) < \kappa$	$\mu \leftarrow y$
free	faithful	(accept or reject) & $\text{best}(\alpha) \neq \theta$ & $\omega=0$	$\mu \leftarrow \text{best}(\alpha)$; send(confirm) to μ ; send(withdraw) to every $z \in \alpha \setminus \{\mu\}$; $\sigma \leftarrow \top$
hesitating	hesitating	propose	send(reject) to y
hesitating	engaged	(accept or reject) & $\omega = 0$ & $\text{best}=\mu$	send(accept) to μ ; send(withdraw) to every $z \in \alpha$
hesitating	faithful	(accept or reject) & $\omega=0$ & $\text{best}(\alpha) <_x \mu$	send(reject) to μ ; $\mu \leftarrow \text{best}(\alpha)$; send(confirm) to μ ; send(withdraw) to every $z \in \alpha \setminus \mu$; $\sigma \leftarrow \top$
engaged	engaged	propose	send(reject) to y
engaged	free	withdraw & $y=\mu$	$\sigma \leftarrow \perp$; $\mu \leftarrow \theta$; $\kappa++$; send(propose) to all m peers at concession level κ ; $\alpha \leftarrow []$; $\omega \leftarrow m$
engaged	faithful	confirm & $\mu=y$	$\sigma \leftarrow \top$
faithful	faithful	propose & $\mu <_x y$	send(reject) to y
faithful	free	divorce	$\sigma \leftarrow \perp$; $\mu \leftarrow \theta$; $\kappa++$; send(propose) to all m pairs at concession level κ ; $\alpha \leftarrow []$; $\omega \leftarrow m$
faithful	unfaithful	propose & $y <_x \mu$	$\lambda \leftarrow y$; send(accept) to y
unfaithful	unfaithful	propose	send(reject) to y
unfaithful	engaged	divorce	$\mu \leftarrow \lambda$; $\lambda \leftarrow \theta$; $\sigma \leftarrow \perp$
unfaithful	faithful	confirm & $\lambda=y$	send(divorce) to μ ; $\mu \leftarrow \lambda$; $\lambda \leftarrow \theta$

TABLE 2 – Actions consécutives aux transitions d'états

réponses, il va hésiter (hesitating) entre cette proposition et les éventuelles réponses positives à celle qu'il a émise. Il attend donc toutes les réponses des agents auxquels il a fait une proposition puis il va : (i) soit accepter la proposition si elle est préférable au meilleur des acceptants - dans ce cas l'agent est fiancé (engaged) et attend une confirmation pour se marier; (ii) soit confirmer au meilleur des acceptants et l'agent devient alors marié. Dans tous les cas, les autres agents reçoivent des réponses négatives (reject ou withdraw). À la fin de cette phase l'agent est : (i) soit marié et va donc pouvoir se remettre en question; (ii) soit libre s'il a accepté la proposition mais que cette dernière est annulée par un withdraw et l'agent augmente donc son niveau de concession pour recommencer le processus de proposition.

Remise en question. Bien sûr, ce n'est pas parce qu'un agent est marié, qu'il va le rester tout au long de l'exécution de l'algorithme. En effet, il peut recevoir des propositions plus intéressantes que son partenaire actuel. Dans ce cas, l'agent continue à être marié mais fait une « infidélité » (unfaithful) en acceptant la proposition. Si cette dernière aboutit sur une confirmation, l'agent divorce de son partenaire actuel (μ) et se marie avec le proposant (l'amant λ), sinon l'agent reste fidèle. Lorsqu'un agent reçoit un message de divorce, il devient : (i) soit libre s'il n'a pas d'amant, et donc il recommence une phase de proposition; (ii) soit fiancé avec

son amant et donc en attente d'une confirmation dans une phase de disposition.

Arrêt. L'algorithme se termine lorsque tous les agents sont mariés, qu'ils soient fidèles ou non.

Lorsque l'algorithme s'arrête, la solution obtenue pour le problème SM avec liste complète est stable.

Théorème 5 *L'appariement obtenu par Casanova est stable.*

Preuve 3 *Supposons le contraire. Dans ce cas, Casanova termine sur un appariement M dans lequel tous les agents sont mariés (c'est la condition d'arrêt de l'algorithme) et il existe un couple bloquant, noté (x, y) . Comme x est marié à $\mu(x)$ (avec $y \succ_y \mu_M(x)$) et selon le déroulement de Casanova, x a envoyé une proposition à y . Comme x n'est pas marié à y cela signifie que y a rejeté sa proposition. C'est possible pour deux raisons : (i) soit le niveau de concession de y n'était pas suffisant à ce moment pour accepter x ; (ii) soit y était engagé à ce moment là avec un partenaire préféré à x . Dans les deux cas, comme y est marié avec $\mu_M(y)$ et que $x \succ_y \mu_M(y)$, cela implique que le niveau de concession de y lui a permis plus tard de faire une proposition à x ou que son partenaire l'a quitté et donc qu'il a fait une proposition à x . Comme $y \succ_x \mu(x)$, x n'a pas pu refuser cette offre : contradiction.*

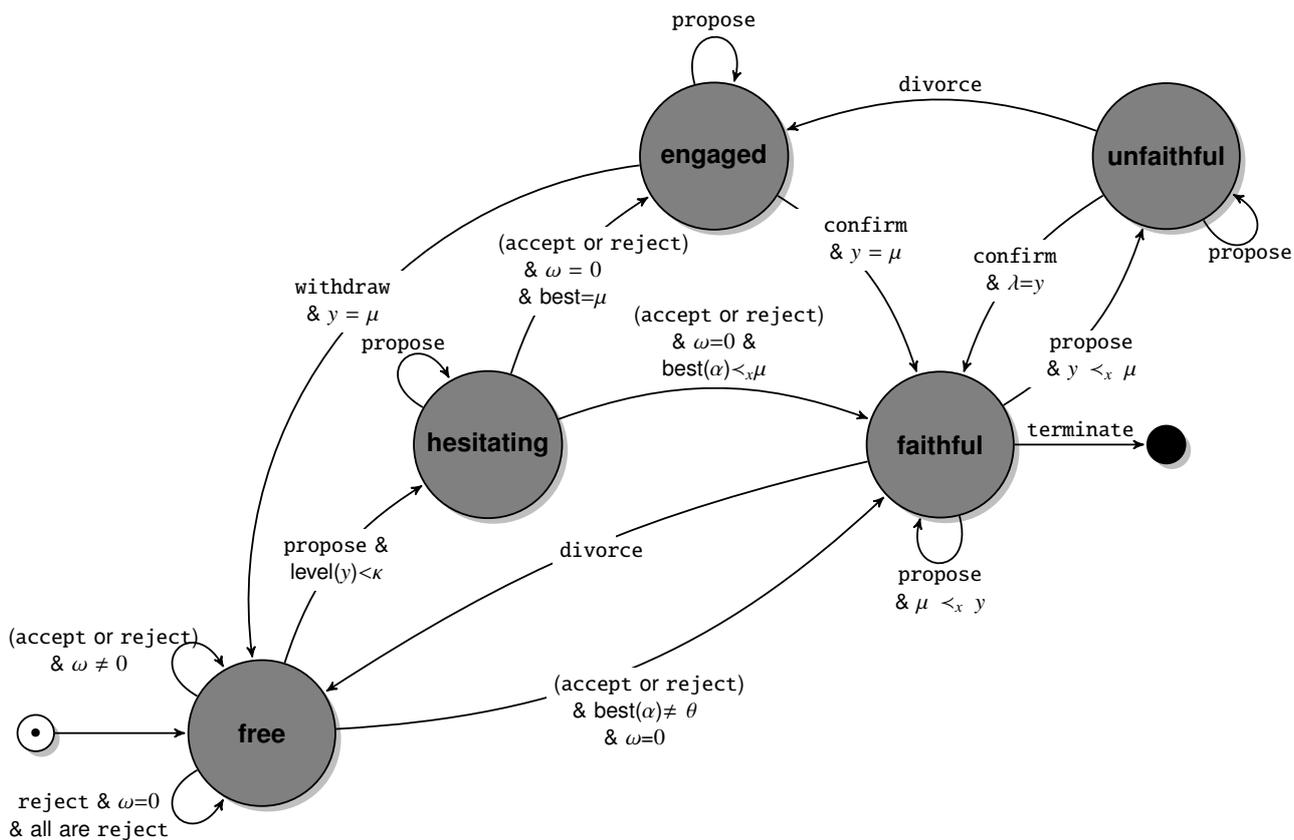


FIGURE 1 – Comportement des agents représenté comme un automate fini déterministe

Exemple 5 Considérons l'exemple 1 du point de vue de y_3 . Les messages envoyés ou reçus par y_3 sont représentés dans la figure 2. Pour rappel, les préférences de y_3 sont $x_1 \succ_{y_3} x_3 \succ_{y_3} x_2$. Initialement, y_3 est *free* et son niveau de concession $\kappa = 0$. Ainsi, le seul partenaire acceptable est x_1 . Dans notre exemple, les négociations conduisent à l'appariement M_1 t.q. $\mu_{M_1}(x_1) = y_1$, $\mu_{M_1}(x_2) = y_2$, $\mu_{M_1}(x_3) = y_3$ avec $sw_e(X \cup Y) = 1$. Tout d'abord, y_3 reçoit une proposition d'un partenaire non acceptable, x_2 (message 1). Ainsi, y_3 rejette cette proposition (message 2). Indépendamment de cette proposition, y_3 fait une proposition à son partenaire préféré, x_1 (message 3). Entre temps, y_3 reçoit une deuxième série de propositions non acceptables de x_2 et de x_3 (messages 4 et 6), qu'elle rejette immédiatement (messages 5 et 7). Malheureusement, la proposition de y_3 est rejetée par x_1 . Par conséquent, elle concède ($\kappa \leftarrow 2$) et elle avance dans sa liste de préférences. y_3 envoie une proposition à x_1 (message 9) et x_3 (message 10), qui sont toutes deux rejetées (messages 11 et 12). Par conséquent, y_3 concède à nouveau et envoie une proposition à ses trois partenaires

acceptables, x_1 (message 13), x_3 (message 14) et x_2 (message 15). x_2 et x_3 acceptent (messages 16 et 17) et ils se considèrent fiancés en attente de confirmation. x_1 rejette la proposition (message 18). y_3 choisit le meilleur acceptant (x_3), confirme le mariage (message 19), et annule sa proposition à x_2 (message 20).

8 Évaluation

Casanova est implémenté à l'aide de la plateforme multi-agent Jason [1]. Jason est une version étendue du langage AgentSpeak [10] qui implémente la sémantique opérationnelle de ce langage. Afin de stopper les négociations lorsqu'une solution stable est atteinte, nous avons implémenté un processus asynchrone inspiré de [4]. Tous les agents sont placés dans un graphe orienté constitué d'un unique circuit et ils échangent des jetons de terminaison.

Afin d'évaluer Casanova, on l'exécute pour des instances de SM dont la taille est comprise de 2 à 100. Pour chaque instance de problème, nous exécutons 10 fois Casanova. Dans la figure 3, on

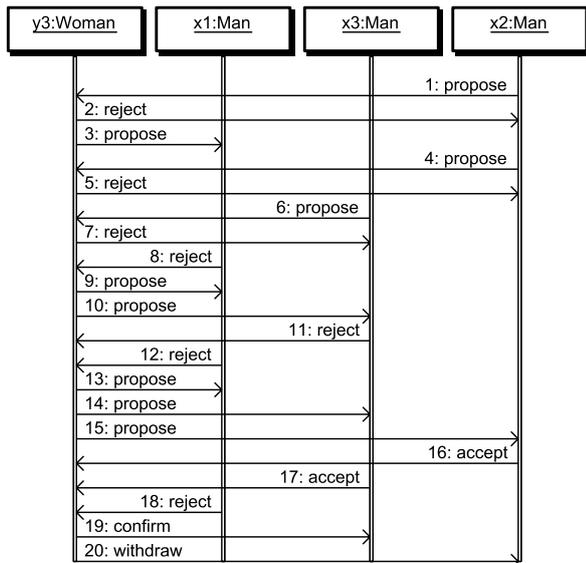


FIGURE 2 – Trace d'exécution de Casanova du point de vue de l'agent y_3

compare la valeur moyenne des différents bien-être à celle obtenue par *GS*.

Tout d'abord, on observe sans surprise que les solutions de Casanova sont moins favorables à la gent masculine mais plus équitables entre les deux communautés. De plus, nous constatons que le bien-être utilitaire est en moyenne meilleur pour Casanova que pour *GS*. Casanova améliore la satisfaction globale de la société. En effet, les solutions optimales ne sont pas nécessairement dans les frontières du treillis des solutions. Ce dernier résultat nous encourage à poursuivre nos investigations.

9 Conclusion

Dans cet article nous avons adopté une approche centrée individu pour la résolution du problème des mariages stables. Dans cette perspective, l'algorithme de Gale-Shapley convient dans la mesure où : (i) il est distribuable ; (ii) la privacité des préférences est préservée ; (iii) deux comportements d'agents –disposant et proposant– interagissent pour atteindre un appariement stable. Toutefois, de part la dichotomie entre ces deux comportements, l'algorithme *GS* favorise les proposant aux détriments des disposants. L'algorithme aboutit donc à des solutions inéquitables qui ne sont pas nécessairement optimales du point de vue global. Nous avons proposé ici Casanova, un comportement par lequel chaque agent joue simultanément le rôle de proposant et de disposant. L'interaction

entre les agents munis de ce comportement permet d'aboutir à une solution stable, plus équitable voire plus optimale d'un point de vue global.

Grâce à Casanova, les agents interagissent dans une multitude de négociations bilatérales. Ils mettent en œuvre une stratégie de concession minimale. Ainsi, au début, les agents envoient une proposition à leur partenaire préféré, et s'il refuse ils concèdent et donc envoient une nouvelle proposition aux deux agents qu'ils préfèrent, etc. De plus, les agents mariés peuvent être amenés à divorcer si l'un des époux reçoit une proposition préférable. Nous avons réalisé notre implémentation en Jason qui fournit un langage permettant de prototyper rapidement des comportements d'agents, et de les distribuer dans un réseau grâce à une infrastructure Jade, par exemple. Toutefois, l'outillage disponible d'analyse et de contrôle de l'exécution peut s'avérer insuffisant lorsque l'on souhaite passer à l'échelle et/ou contrôler de nombreuses expérimentations, notamment les tours de paroles. À cette intention, notre travail actuel consiste à proposer une implémentation de Casanova en Java.

Le problème du mariage stable est un problème abstrait qui peut être instancié par une large gamme d'applications pratiques comme l'affectation des projets à des étudiants. On peut penser également au co-voiturage qui est un problème d'appariement entre deux communautés : des conducteurs et des passagers. D'une manière plus générale, le problème *SM* est un cas particulier du problème d'appariement stable (en anglais, *Stable Matching Problem*). Parmi les extensions possibles, on compte : (i) le problème de co-localisation (en anglais, *roomate problem*) qui consiste à partitionner un ensemble fini d'individus en couples ou singletons ; (ii) la formation de coalitions où les individus sont partitionnés en fonction de leurs préférences sur les coalitions ; (iii) les marchés sociaux où des liens peuvent être établis et les individus ont des préférences sur ces liens (voire sur la structure entière du réseau). Afin de résoudre ce genre de problèmes, l'approche centrée individu permet d'envisager des algorithmes adaptatifs, *anytime* et préservant la privacité des préférences.

Remerciements

Nous remercions Jean-François Vion pour son travail d'implémentation et les relecteurs pour leurs commentaires.

Références

- [1] Rafael H. Bordini, Jomi Fred Hubner, and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons Ltd, 2007.
- [2] I. Brito and P. Meseguer. Distributed stable marriage problem. In *6th Workshop on Distributed Constraint Reasoning at IJCAI 05*, pages 135–147, 2005.
- [3] I. Brito and P. Meseguer. Distributed stable matching problems with ties and incomplete lists. In *Proc. of CP-06*, volume 6057 of *LNCS*, pages 675–680. Springer Verlag, Berlin, Germany, 2006.
- [4] Edsger W. Dijkstra and Carel S. Scholten. Termination detection for diffusing computations. *Inf. Process. Lett.*, 11(1) :1–4, 1980.
- [5] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69 :9–14, 1962.
- [6] Donald Knuth. *Mariage stables*. Les Presses de l'Université de Montréal, 1971.
- [7] Maxime Morge and Paolo Mancarella. Assumption-based argumentation for the minimal concession strategy. In *Proc. of ArgMAS 2009*, volume 6057 of *LNCS*, pages 114–133. Springer Verlag, Berlin, Germany, 2010.
- [8] Maxime Morge and Gauthier Picard. Privacy-preserving strategy for negotiating stable, equitable and optimal matchings. In *Proc. of PAAMS*, pages 97–102, 2011.
- [9] Hervé Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.
- [10] Anand S. Rao. AgentSpeak(L) : BDI agents speak out in a logical computable language. In *Proc. of MAAMAW*, 1996.
- [11] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of encounter : designing conventions for automated negotiation among Computers*. The MIT press series of artificial intelligence. MIT press, 1994.
- [12] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-sided Matching ; A Study in Game-theoretic Modeling and Analysis*. Cambridge University Press, 1990.
- [13] Thomas Schelling. *Micromotives and macrobehavior*. Norton & Co., 1978.

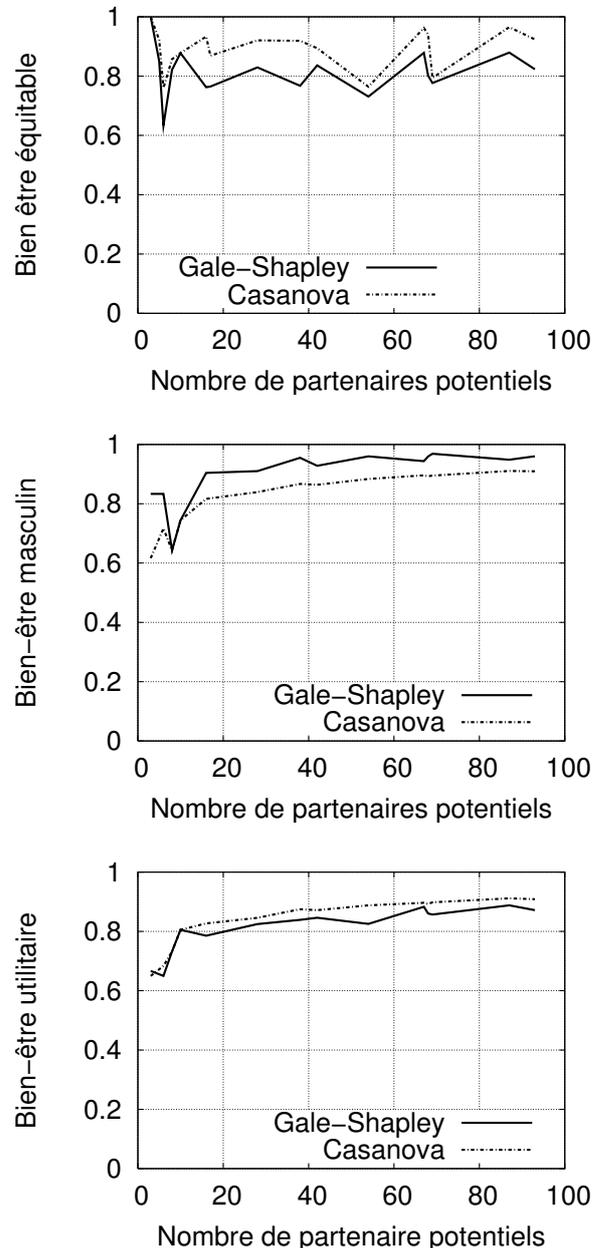


FIGURE 3 – Comparaison des solutions de Casanova avec celles de GS (ou DisEGS).