

# Problème d'optimisation de contraintes distribuées et choix social en systèmes multi-agents

Rapport de Master 2 Recherche  
*Web Intelligence*



l'Université Jean Monnet et l'École des Mines de Saint-Étienne

14 juin 2010

**Santiago Villarreal**

Laboratoire d'accueil : École Nationale Supérieure des Mines de Saint-Étienne (ENSMSE),  
Centre Génie Industriel et Informatique (G2I)

Responsables de stage :

Gauthier Picard,

Maxime Morge

Equipe d'accueil : Département Systèmes Multi-Agents

**Mots-clés** : Problème de satisfaction de contraintes, systèmes multi-agents, comportement d'agents, optimisation de contraintes distribuée, choix social, bien-être.

---

**Résumé** : Dans le domaine de l'intelligence artificielle, certains problèmes sont formalisés comme des problèmes de satisfaction de contraintes (CSP). Dans un système multi-agents, nous cherchons à résoudre un CSP de manière distribuée (DisCSP), optimisée (DisCOP), et on peut également prendre en compte le bien-être de la société d'agents (évaluation collective).

À partir de la littérature, nous présentons les caractéristiques et les solutions à de tels problèmes. Nous proposons, comme perspective, une nouvelle solution prenant en compte le bien-être des agents, contrairement aux approches classiques.

Ce travail est illustré au travers du problème de mariage stable.

---



---

# Remerciements

J'adresse mes remerciements à l'équipe du Centre Génie Industriel et Informatique de l'École des Mines de Saint-Étienne pour m'avoir permis d'effectuer mon stage master recherche 2.

Je remercie plus particulièrement et spécialement :

Monsieur Picard Gauthier et monsieur Morge Maxime mes encadrants, qui ont fait preuve d'une grande disponibilité à mon égard, pour m'avoir fait confiance et m'a guidé dans la réalisation de mon stage recherche 2, surtout la patience au moment de poser un question.

Monsieur Olivier Boissier, responsable du département SMA pour m'avoir accueilli au sein de son équipe, pour l'aide et les conseils concernant théorie et technique de Jason évoquées dans ce rapport, aussi la motivation obtenue.

Je remercie également l'ensemble du personnel du Laboratoire d'accueil : École Nationale Supérieure des Mines de Saint-Étienne, et chaque professeur de mon master pour les connaissances acquises.

J'ai grandes souvenirs de cette expérience.

Agradesco desde el fondo de mi corazón a mis padres Rafael Villarreal y Gladys Narvaez por la oportunidad de poder vivir esta experiencia que le he catalogado como la mas enriquecedora de mi vida. ellos siempre han sido mi inspiración y mi valentía en cada paso que doy, siempre han apoyado toda decisión que he tomado. Me siento orgullo de representar aquí a mi familia y por la confianza que ellos han puesto en mi, todo esfuerzo realizado les dedico ha ellos. también nombro a mi hermano que ha sabido poner en mi una ambición sana, la fraternidad, y conocimiento de diversas filosofía, a los nuevos integrantes de este gran equipo como es Rafael Mateo Villarreal y Susana Sanchez. pero mi familia no se limita a ellos 4, son muchas personas que espero me escucen por no nombrarlas, que con una llamada telefónica o un email alegraron mi dia, muchas gracias a toda la gente especial para mi.

Agradesco a Fernanda Loayza, mi preciosa que ha sabido acompañarme largas noches y ser un empuje motivador, escuchar ser un oído y un hombro, levantarme con un beso en las mañanas, brindarme su carino incondicionalmente, gracias preciosa por ser parte de esta tesis, gracias por sacarme sonrisas en medio de mi seriedad y muchas cosas mas que llenan mi corazón día a día.

Agradezco también a mis amistades aquí en Francia, sobre todo a Stalin Salazar y Nancy Betancourt, fuimos tres ecuatorianos que llegamos con una meta, y les deseo mucha suerte para que se haga realidad, sus companias y momentos de apoyo y diversión.

Con mucho cariño a mis seres queridos.



---

# Table des matières

<b>Remerciements</b>	<b>iii</b>
<b>Introduction</b>	<b>1</b>
<b>I État de l’art</b>	<b>3</b>
<b>1 Le problème du mariage stable modélisé sous forme de CSP</b>	<b>5</b>
1.1 Constraint Satisfaction Problem (CSP)	5
1.2 Illustration avec le Problème Mariage Stable (SMP)	5
1.2.1 Le Problème Mariage Stable (SMP)	5
1.2.2 Variantes et extensions de SMP	7
1.2.2.1 SM avec liste de préférences complète (SMP)	7
1.2.2.2 SM avec liste de préférences incomplète (SMPI)	7
1.2.2.3 Extensions du SMP	7
1.2.3 Modélisation de SMP en CSP	7
1.2.4 Algorithme extension Gale-Shapley (EGS)	8
<b>2 Approches de résolution par SMA</b>	<b>11</b>
2.1 Agent et système multi-agent (SMA)	11
2.1.1 Agent	11
2.1.2 SMA	11
2.2 Distributed Constraint Satisfaction Problem (DisCSP)	11
2.3 Distributed Constraint Optimization Problem (DisCOP)	12
2.4 Algorithme pour DisSMP	13
2.5 Choix social et bien être social	15
2.5.1 Théorie du choix social	15
2.5.2 Théorie du bien-être social	16
2.5.2.1 Bien-être utilitaire	16
2.5.2.2 Bien-être égalitaire	16
2.5.2.3 Le produit de Nash	17
2.5.2.4 Bien-être élitiste	17
<b>II Vers une nouvelle approche</b>	<b>19</b>
<b>3 Analyse de l’algorithme DisEGS</b>	<b>21</b>
3.1 Algorithme DisEGS version agent	21
3.1.1 Les procédures	21
3.2 Analyse de DisEGS	25
3.2.1 L’égalité entre agent	25

3.2.2	privacité . . . . .	26
3.2.3	Bien-être social . . . . .	26
<b>4</b>	<b>Présentation de notre approche</b>	<b>29</b>
4.1	Algorithme Casanova . . . . .	29
4.1.1	Les procédures . . . . .	30
4.2	Analyse de Casanova . . . . .	31
4.2.1	L'égalité entre agents . . . . .	31
4.2.2	Privacité . . . . .	31
4.2.3	Bien-être social . . . . .	31
<b>5</b>	<b>Validation</b>	<b>33</b>
5.1	Exemple pour l'expérimentation . . . . .	33
5.2	Expérimentation DisEGS . . . . .	33
5.3	Expérimentation Casanova . . . . .	34
5.4	Comparaison entre DisEGS et Casanova . . . . .	35
	<b>Conclusion</b>	<b>37</b>
	<b>Bibliographie</b>	<b>39</b>







---

# Introduction

Ce rapport présente le travail effectué lors de mon stage Master 2 Recherche, dans le cadre de ma deuxième année de Master Web Intelligence, cohabilité par l'Université Jean Monnet de Saint-Étienne et l'École Nationale Supérieure des Mines de Saint-Étienne. j'ai réalisé ce travail dans l'équipe ISCOD du centre Génie Industriel et Informatique (G2I) de l'École des Mines de Saint-Étienne et suis encadré par Gauthier Picard (ISCOD/ENSMSE) et Maxime Morge (SMAC/LIFL).

L'idée est de modéliser des problèmes de décision collectifs (par exemple planifier un rendez-vous avec des collègues) comme des problèmes d'optimisation sous contraintes distribuées (DisCOP). Les agents doivent trouver un accord sur un ensemble de décisions. Chaque décision réside dans l'affectation d'une valeur à une variable sujette à des contraintes. Des algorithmes tels que ABT et AWC, AAS, DPOP, ADOPT [Yokoo, 2001] [Modi *et al.*, 2005] permettent de calculer une solution optimale.

Toutefois, ces derniers s'appuient sur deux hypothèses fondamentales comme : (i) l'objectif consiste à maximiser le bien-être social utilitaire, c'est-à-dire la somme des utilités des agents et (ii) les agents sont égoïstes, c'est-à-dire ils n'acceptent des accords que si ceux-ci augmentent strictement leur bien-être individuel. Nous souhaitons nous appuyer sur des critères comme le bien-être social élitiste, le bien-être social égalitaire ou le produit de Nash [Moulin, 1986]. Par exemple, le bien-être social égalitaire revient à maximiser le bien-être individuel de l'agent le plus pauvre. À cette intention, les agents proposés pourront être altruistes, c'est-à-dire prêt à faire des concessions afin d'accroître le bien-être individuel de leur accointances. De tels comportements nécessitent la connaissance par les agents des préférences de leur accointances.

Nous avons sélectionné le Problème de Mariage Stable comme cas d'étude. Nous avons étudié les propositions existantes, et nous avons construit notre algorithme.

Dans une première partie de ce rapport, des définitions de base, importantes pour bien comprendre le travail proposé et une illustration avec un exemple de choix social (problème mariage stable) sont présentées. Suivent une étude des solutions existantes de manière centralisée et distribuée (approches de résolution par SMA) avec des définitions nécessaires pour le développement du travail présenté. Enfin, nous décrivons notre nouvelle approche.



*Première partie*

---

## **État de l'art**



# 1

## Le problème du mariage stable modélisé sous forme de CSP

Dans ce chapitre nous décrirons notre cas d'étude, puis à partir des définitions de CSP et SMP, nous cherchons modéliser SMP comme un CSP.

### 1.1 Constraint Satisfaction Problem (CSP)

Un problème de satisfaction de contraintes (CSP) est une structure générale qui peut formaliser divers problèmes dans l'application de l'intelligence artificielle [Yokoo, 2001].

**Définition 1.** *Un CSP est défini de la manière suivante. Soit un ensemble de variables  $X = \{X_1, \dots, X_n\}$ . Chaque variable  $X_i$  peut prendre une valeur parmi un ensemble  $D_i$ . Cet ensemble constitue le domaine de la variable. Formellement, on a  $D = \{D_1, \dots, D_n\}$ . On a également un ensemble de contraintes  $C = \{C_1, \dots, C_p\}$  qui est un ensemble de contraintes spécifiant les combinaisons de valeurs acceptables pour les variables. Chaque contrainte  $C_i$  peut également s'exprimer par un prédicat  $p_i(Y_1, \dots, Y_j)$  défini pour un sous-ensemble de variables  $\{Y_i, \dots, Y_j\} \subseteq X$ . On cherche à donner une valeur à chaque variable de manière à ce que chaque contrainte soit satisfaite.*

Une solution à un CSP est une affectation totale qui satisfait toutes les contraintes de  $C$ . Il existe des algorithmes pour résoudre de CSP, par exemple Iterative improvement, Consistency, ou Backtracking qui est un algorithme recherche basique, systématique et complet pour résoudre un CSP [Yokoo, 2001].

### 1.2 Illustration avec le Problème Mariage Stable (SMP)

Nous avons choisi le SMP comme cas de étude pour ses caractéristiques. Il nous donne une idée claire pour démarre notre travail. Nous utilisons la définition classique de [Brito & Meseguer, 2005] :

#### 1.2.1 Le Problème Mariage Stable (SMP)

Un SMP est un problème combinatoire. Il consiste à chercher un ensemble de mariages stables entre  $n$  hommes et  $n$  femmes (figure 1.1). Chaque individu a ses préférences représentées sous forme de liste ordonnée sur les membres du groupe de l'autre sexe (figure 1.2).

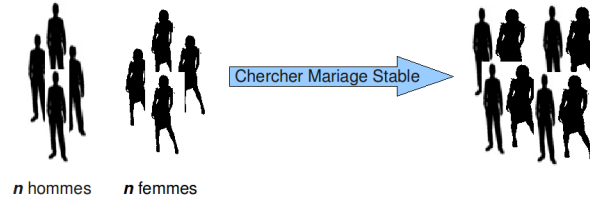


FIG. 1.1 – Ensemble des agents

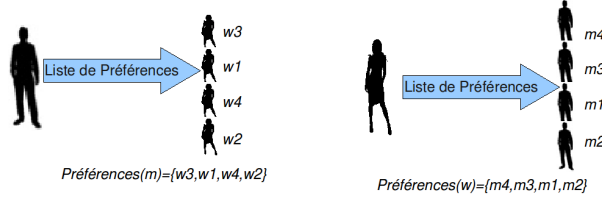


FIG. 1.2 – Liste de préférences

Un mariage  $M$  n'est pas stable si et seulement si pour une personne  $m$  mariée à une personne  $w$ , il existe une tierce personne  $f$  telle que (figure 1.3) :

1.  $m$  préfère  $f$  par rapport  $w$
2. et  $f$  préfère  $m$  par rapport à son partenaire actuel

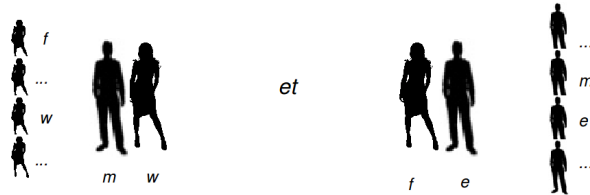


FIG. 1.3 – Liste de préférences

**Définition 2.** Formellement, on dit qu'un mariage est stable ssi :

$$\forall i, 1 \leq i \leq n$$

$M = \{m_i\}$  ensemble des hommes

$F = \{f_i\}$  ensemble des femmes

$$\forall m_i \exists LP(m_i) = F$$

$$\forall f_i \exists LP(f_i) = M.$$

$$\forall m_i = f_k, \neg \exists f_j \prec_{LP(m_i)} f_k \text{ et } f_j = m_x, m_i \prec_{LP(f_j)} m_x.$$

### 1.2.2 Variantes et extensions de SMP

#### 1.2.2.1 SM avec liste de préférences complète (SMP)

Une personne a des préférences sur toutes les autres personnes de l'autre sexe, donc pour tout individu  $i$ ,  $|LP(i)| = n$

Exemple :

$m_1 : w_2 w_3 w_1$	$w_1 : m_1 m_2 m_3$
$m_2 : w_1 w_2 w_3$	$w_2 : m_1 m_3 m_2$
$m_3 : w_2 w_1 w_3$	$w_3 : m_2 m_1 m_3$

Il n'y a qu'un seul mariage stable dans ce cas :  $M = \{(m_1, w_2)(m_2, w_1)(m_3, w_3)\}$

#### 1.2.2.2 SM avec liste de préférences incomplète (SMPI)

Une personne a des préférences sur un sous-ensemble des personnes de l'autre sexe. Dans le condition de SMP, on dit que  $m$  appartient à la liste de préférences de  $w$  et  $w$  appartient à la liste de préférence de  $m$ . Une instance de SMI peut être complète ou incomplète, donc  $\forall i, 1 \leq |LP(i)| \leq n$

#### 1.2.2.3 Extensions du SMP

Dans [Brito & Meseguer, 2005], on trouve des variations pour SMP : le problème de colocataire stable (en anglais SRP), Mariage sans spécification du sexe (généralisation), etc. Par contre, EGS ne résout pas ces problèmes.

### 1.2.3 Modélisation de SMP en CSP

[Brito & Meseguer, 2005] modélise le SMP comme le CSP suivant :

Ils distinguent deux types de variables dans le SMP,  $N$  hommes ( $X_1, \dots, X_n$ ) et  $N$  femmes ( $Y_1, \dots, Y_n$ ). La  $LP(q)$  est la liste de préférence de  $q$ , et les domaines sont :

- $D(X_i) = \{j : w_j \in LP(m_i)\} \forall i, 1 \leq i \leq n$
- $D(Y_j) = \{i : m_i \in LP(w_j)\} \forall j, 1 \leq j \leq n$

Dans ce CSP, quand la variable  $X_i$  prend le valeur  $j$ , on dit que l'homme  $m_i$  est marié avec la femme  $w_j$ . Soit  $d_i^m = |D(X_i)|$  et  $d_j^w = |D(Y_j)|$ . Les contraintes sont définies à partir d'un homme et d'une femme, pour chaque paire  $i$  et  $j$  où  $(1 \leq i, j \leq n)$ , la contrainte  $X_i/Y_j$  entre  $X_i$  et  $Y_j$  est représentée par une matrice de conflits (figure 1.4)  $C_{i,j}$  de taille  $d_i^m \times d_j^w$ , et  $k, l (k \in D(x_i) \text{ et } l \in D(y_j))$ , donc :

- $C_{i,j}[k, l] = \text{Allows}$ , indique que cette relation est accepté
- $C_{i,j}[k, l] = \text{Illegal}$ , assure la monogamie du mariage
- $C_{i,j}[k, l] = \text{Blocking}$ , si et seulement si une paire  $m_i$  préfère  $w_j$  par rapport  $w_k$  et  $w_j$  préfère  $m_i$  par rapport à  $m_l$

- $C_{i,j}[k,l] = \text{Support}$ , dans les autres cas

	$m_1$	$m_2$	$m_3$		$m_1$	$m_2$	$m_3$
$w_2$	S	S	I	$w_2$	1	1	0
$w_1$	I	I	A	$w_1$	0	0	1
$w_3$	S	S	I	$w_3$	1	1	0

FIG. 1.4 – matrice de conflits entre  $m_3$  et  $w_1$

#### 1.2.4 Algorithme extension Gale-Shapley (EGS)

EGS résoudre SMP de manière centralisée car on connaît la liste de préférences de tous les individus. L'algorithme de Gale-Shapley montre qu'il existe au moins un mariage stable pour tout SMP en  $O(n^2)$  [Gale & Shapley, 1962]. EGS est un extension de l'algorithme de Gale-Shapley avec une étape supplémentaire ( 1.5 ) pour supprimer les partenaires incompatibles de la liste des préférences. EGS implique une séquence de proposition des hommes vers les femmes (ou inversement). L'algorithme 1 présente une version orienté homme [Brito & Meseguer, 2006].

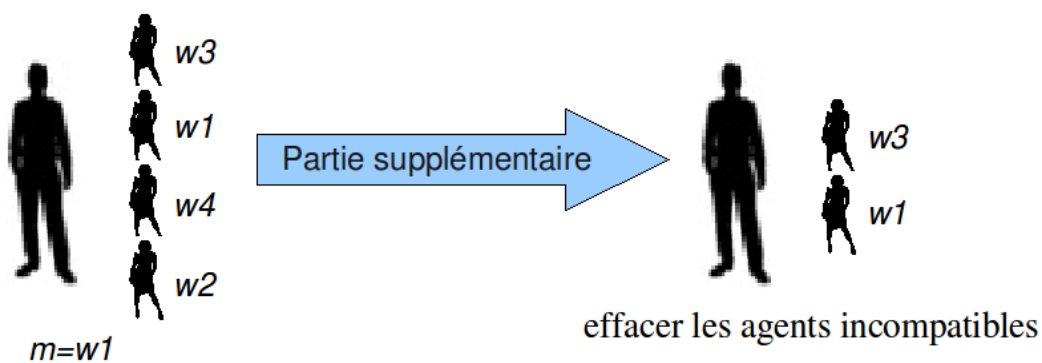


FIG. 1.5 – Étape supplémentaire



Chaque personne est initialement libre ;

**Tant que** (il existe un homme  $m$  libre et  $m$  a une liste de préférences non vide) **faire**

    [3.]  $w :=$  la première femme sur la liste de  $m$  ;  $\{m$  **propose** à  $w\}$

**Si** ( $m$  n'est pas sur la liste des préférences de  $w$ ) **Alors**

        supprimer  $w$  de la liste des préférences de  $m$  ;

        aller à la ligne 3 ;

**Fin Si**

**Si** (il existe homme  $p$  est engagé avec  $w$ ) **Alors**

        rendre  $p$  libre ;

**Fin Si**

    engager  $m$  et  $w$  l'un à l'autre ;

**Pour** chaque successeur  $p$  de  $m$  sur la liste de  $w$  **faire**

        supprimer  $p$  de la liste  $w$  ;

        supprimer  $w$  de la liste  $p$  ;

**Fin Pour**

**Fait**

Algorithme 1: Algorithme EGS, homme orienté



# 2

---

## Approches de résolution par SMA

Dans cette chapitre, on donnera des définitions importantes pour comprendre la résolution de SMP distribué. Il est important de remarquer que DisEGS donne des mariages stables, par contre il ne fait pas une analyse de la satisfaction de chaque agent ou de la satisfaction de tous les participants. Une amélioration de cet algorithme peut être l'inclusion de l'analyse et l'évaluation de la satisfaction des agents.

### 2.1 Agent et système multi-agent (SMA)

Nous présentons dans cette section les concepts d'agent et système multi-agent.

#### 2.1.1 Agent

Selon la définition de [Wooldridge & Jennings, 1995] :

**Dfinition 3.** *Un agent est un système informatique situé dans un environnement qui est :*

- autonome, il agit de lui-même sans ordres extérieurs ;
- sociable, il peut communiquer avec d'autres agents ;
- réactif, il perçoit son environnement et réagit à ses changements ;
- proactif, il ne fait pas que réagir à l'environnement, il peut décider seul d'agir pour atteindre ses objectifs.

#### 2.1.2 SMA

Il existe plusieurs définitions pour un SMA. On peut dire que un SMA est une société d'agents. On considère la définition de [Wooldridge & Jennings, 1995].

**Dfinition 4.** *Un système multi-agent (SMA) est un ensemble d'agents qui sont dans un même environnement. Il fournit des moyens de communication aux agents, qui interagissent afin d'effectuer les tâches générales (communes).*

### 2.2 Distributed Constraint Satisfaction Problem (DisCSP)

Un DisCSP est un CSP devant être résolu par un ensemble d'agents  $A = \{A_1, \dots, A_k\}$ . Chaque variable appartient à un agent. Un agent peut affecter une valeur uniquement aux variables qu'il possède.

Il existe deux types de contraintes, les contraintes intra-agents qui ne sont connues que par un agent et les contraintes inter-agents qui sont connues par tous les agents ayant une variable impliquée dans cette contrainte [Gaudreault, 2009].

Yokoo a proposé différents algorithmes pour résoudre les DisCSP [Yokoo, 2001]. Les critères pour comparer les algorithmes sont généralement (i) le temps nécessaire pour résoudre le problème (ii) le nombre de messages échangés entre les agents [Gaudreault, 2009]. L'algorithme est d'ABT (Asynchronous Backtracking) décrit suite.

ABT est le premier algorithme de résolution de DisCSP [Yokoo, 2001] qui permet aux agents de travailler de manière simultanée. On parle d'asynchronisme car les agents peuvent changer la valeur de leur variable à tout moment. Quand un agent change la valeur de sa variable, il communique aux agents de priorité inférieure. Quand un agent reçoit la valeur d'un agent de priorité supérieure, il donne une nouvelle valeur à sa variable pour avoir une valeur compatible dans les contraintes. S'il existe un "blocage"<sup>1</sup> dans une contrainte, un message (nogood) est envoyé aux agents impliqués pour qu'ils changent la valeur de leur variable respective. Nous souhaitons utiliser les caractéristiques d'ABT pour construire notre propre solution.

## 2.3 Distributed Constraint Optimization Problem (DisCOP)

**Définition 5.** Un DisCOP est une extension d'un DisCSP. On a un ensemble de variables  $X = \{X_1, \dots, X_n\}$ . Un agent  $A_i \in A = \{A_1, \dots, A_n\}$  qui a pour responsabilité de donner une valeur à la variable  $X_i$ . Chaque variable prend la valeur d'un domaine  $D_i = \{D_1, \dots, D_n\}$ . On dispose également d'un ensemble de contraintes binaires  $C = \{C_1, \dots, C_m\}$ . Chaque contrainte  $C_j$  s'exprime par une fonction  $f_j(y_\alpha, y_\beta) : D_\alpha \times D_\beta \rightarrow \mathbb{R}^+$  définie pour un couple de variables  $\{y_\alpha, y_\beta\} \subseteq X$  avec  $\alpha \neq \beta$ . On cherche à donner une valeur à chaque variable de manière à minimiser la fonction  $F = \sum_{j=1, \dots, m} f_j$  [Gaudreault, 2009].

L'idée n'est pas forcément de minimiser le nombre de contraintes violées dans le DisCSP, contrairement du problème MaxSAT.

L'algorithme Asynchronous Distributed Constraint Optimization (ADOPT) est un algorithme à la fois asynchrone et complet pour les DisCOP [Modi *et al.*, 2005]. Il donne la garantie de la qualité d'une solution obtenue grâce au fait que les agents calculent à tout instant, donc on peut avoir une réponse partielle.

La communication est locale. Les agents peuvent envoyer seulement aux voisins<sup>2</sup>. ADOPT a un agent racine unique qui indique le coût global lié et la terminaison du calcul; ADOPT a un degré de centralisation car il y a information partagée entre des agents et l'agent racine, mais il a également des caractéristiques distribué, car les agents calculent en parallèle.

---

<sup>1</sup>blocage quand une valeur d'une variable n'est pas valide pour la contrainte

<sup>2</sup>un voisin est un agent qui a une contrainte commune

ADOPT a une organisation d'agents sous forme d'un pseudo arbre où la racine est l'agent qui a plus de priorité pour choisir le valeur de sa variable que ses fils. Chaque agent choisit une valeur minimisant le coût total des contraintes violées.

ADOPT suppose que DisCOP a les caractéristiques (hypothèses de travail) suivants :

- toutes les contraintes sont binaires
- la fonction objectif s'exprime comme la somme des valeurs des contraintes
- la fonction objectif est monotone

## 2.4 Algorithme pour DisSMP

Le DisSMP est un SMP résolu de manière distribuée où chaque agent garde et modifie ses préférences. L'algorithme DisEGS est l'algorithme de référence pour résoudre le DisSMP [Brito & Meseguer, 2006].

DisEGS est une adaptation d'EGS pour le DisSMP décrit dans [Brito & Meseguer, 2006]. Il consiste en deux phases : (i) proposition et (ii) disposition. Il existe deux types d'orientation homme (ou femme), où (i) les agents hommes (ou les femmes) font des proposition aux agents de sa préférence du group de l'autre sexe pour se marier avec lui, et (ii) les agents femmes(hommes) donne une réponse à la proposition ; la repense peut être l'acceptation ou le refus (l'agent supprime alors du refus de sa liste de préférences). Dans les algorithmes 2 et 3 [Brito & Meseguer, 2006], nous présentons la version orientée homme.

Les messages échangés entre les agents sont :

*propose* :  $x$  envoie ce message pour proposer l'engagement à  $y$  ; *accepte* :  $y$  envoie ce message pour communiquer l'acceptation à  $x$  après qu'elle a reçu une message de proposition. *rejet* :  $y$  envoie ce message pour communiquer que  $y$  n'est pas disponible pour  $x$ . *arrêt* : c'est un message spécial pour communiquer que l'exécution est terminée (fin), ce message est envoyé par un agent special.

```

Procédure homme( $m$ )
   $m \leftarrow libre$ ;
   $fin \leftarrow faux$ ;
  Tant que ( $\neg fin$ ) faire
    Si (( $m = libre$ ) ET ( $préférences(m) \neq \emptyset$ )) Alors
       $w \leftarrow premier(préférences(m))$ ;
      envoieMsg(propose, $m,w$ );
       $m \leftarrow w$ ;
    Fin Si
     $msg \leftarrow lireMsg()$ ;
    Selon que
      accepte : rien à faire ;
      supprime :  $préférences(m) \leftarrow préférences(m) - msg.sender$ ;
        Si ( $msg.envoyer = w$ ) Alors
           $m \leftarrow libre$ ;
        Fin Si
      arrêt :  $fin \leftarrow vrai$ ;
    Fin Selon que
  Fait
Fin

```

Algorithme 2: La procédure homme dans la version orienté homme de l'algorithme DisEGS

```

Procédure femme( $w$ )
   $w \leftarrow \text{libre};$ 
   $\text{fin} \leftarrow \text{faux};$ 
  Tant que ( $\neg \text{fin}$ ) faire
     $\text{msg} \leftarrow \text{lireMsg}();$ 
    Selon que
      propose :  $m \leftarrow \text{msg.sender};$ 
      Si ( $\neg(m \in \text{préférences}(w))$ ) Alors
         $\text{envoieMsg}(\text{supprime}, w, m);$ 
      Sinon
         $\text{envoieMsg}(\text{accepte}, w, m);$ 
         $w \leftarrow m;$ 
        Pour  $p \prec_w m$  faire
           $\text{envoieMsg}(\text{supprime}, w, m);$ 
           $\text{préférences}(w) \leftarrow \text{préférences}(w) - p;$ 
        Fin Pour
      Fin Si
    arrêt :  $\text{fin} \leftarrow \text{vrai};$ 
  Fin Selon que
Fait
Fin

```

Algorithme 3: La procédure femme la version orienté homme de l'algorithme DisEGS

## 2.5 Choix social et bien être social

### 2.5.1 Théorie du choix social

On considère une société d'agents. Chaque agent a ses préférences de choix d'action. Il est nécessaire d'avoir une décision collective et de prendre en compte les préférences des agents pour arriver à un accord collectif. Chaque agent prend en compte le choix des autres agents pour avoir une décision sur son choix. Formellement, on cite la définition de fonction de choix social de [Shoham & Brown, 2009].

**Définition 6. (Fonction de choix social)** Soient  $A$  : un ensemble d'agents,  $O$  : un ensemble de choix sociaux,  $(\succ_i, O)_{i \in A}$  : les préférences individuelles (transitive asymétrique strict). On appelle **fonction de choix social**  $\succ^*$  la préférence collective. Elle doit satisfaire les propriétés suivantes :

$\succ^*$  existe quelque soit les préférences individuelles ;

$(\succ^*, O)$  est asymétrique et transitive ;

la règle d'unanimité (efficacité parétienne) :

$$\text{si } \forall i \in A, o \succ_i o' \text{ alors } o \succ^* o'$$

la règle d'indépendance vis à vis des alternatives non-pertinentes :

$$\text{si } \forall o, o' \in O, o \succ_i o' \Leftrightarrow o \succ'_i o' \text{ alors } \succ'^* \Leftrightarrow \succ^*$$

la règle de non-dictature :

$$\neg \exists i \in A, o \succ_i o' \Rightarrow o \succ^* o'$$

## 2.5.2 Théorie du bien-être social

L'évaluation collective d'une allocation de ressources est toujours un problème important à résoudre. On peut définir un ensemble d'outils prenant en compte le bien-être de tous ses agents d'une société d'agents. On utilise les critères de bien-être utilitaire, bien-être égalitaire, ou le produit de Nash pour l'analyse du bien-être collectif. [Delahaye & Mathieu, 2009] :

### 2.5.2.1 Bien-être utilitaire

Le concept plus utilisée pour évaluer une allocation de ressources est le bien-être utilitaire. On évalue le bien-être collectif comme la somme des bien-être de chaque agents de la société. Ce critère est utilisé pour maximiser le bien-être moyen des agents de la société. Quand on pense à l'égalité entre les agents, dans ce cas on doit utiliser le bien-être égalitaire car le bien-être utilitaire n'est pas adapté dans ce cas car on cherche maximiser le bien-être individuel. Selon la définition de [Nongaillard, 2009] :

**Définition 7.** Le bien-être utilitaire d'une allocation de ressource  $A$ , que l'on notera  $sw_u(A)$ , correspond à la somme des utilités individuelles.

$$sw_u(A) = \sum_{i \in P} u_i(R_i), A \in A.$$

### 2.5.2.2 Bien-être égalitaire

Le bien-être égalitaire considère les bien-être individuels d'agents les plus pauvres de la société. L'idée est de réduire les inégalités entre les agents, rechercher un traitement équitable. Selon la définition de [Nongaillard, 2009] :

**Définition 8.** Le bien-être égalitaire d'une allocation de ressources  $A$ , que l'on notera  $sw_e(A)$ , correspond à l'utilité individuelle de l'agent le plus pauvre.

$$sw_e(A) = \min_{i \in P} u_i(R_i), A \in A.$$



### 2.5.2.3 Le produit de Nash

Le produit de Nash est un concept qui peut être vu comme un compromis entre le bien-être utilitaire et bien-être égalitaire. On ne doit pas considérer l'utilisation du produit de Nash que où dans le cas toutes les valeurs d'utilité sont positives. Selon la définition de [Nongaillard, 2009] :

**Dfinition 9.** *Le produit de Nash d'une allocation de ressources  $A$ , que l'on notera  $sw_n(A)$ , correspond au produit des utilités individuelles.*

$$sw_n(A) = \prod_{i \in P} u_i(R_i), A \in \mathbf{A}.$$

### 2.5.2.4 Bien-être élitiste

Le bien-être élitiste est le contraire du bien-être égalitaire. Le bien-être élitiste se centre sur le bien-être de l'agent le plus riche dans la société. On utilise ce concept quand tous les agents ont un objectif commun. Selon la définition de [Nongaillard, 2009] :

**Dfinition 10.** *Le bien-être élitiste d'une allocation de ressource  $A$ , que l'on notera  $sw_{el}(A)$ , correspond à l'utilité individuelle de l'agent le plus riche dans la population.*

$$sw_{el}(A) = \max_{i \in P} u_i(R_i), A \in \mathbf{A}.$$



*Deuxième partie*

---

## **Vers une nouvelle approche**



# 3

---

## Analyse de l'algorithme DisEGS

Dans ce chapitre nous présentons les modifications que nous avons apportées et une analyse à l'algorithme DisEGS de [Brito & Meseguer, 2006].

### 3.1 Algorithme DisEGS version agent

Les algorithmes 5 et 6 sont la version agent que nous avons implémentées avec le langage Jason à partir des algorithmes 2 et 3. On y utilise les mêmes messages échangés, par contre nous avons implémenté un autre mécanisme d'arrêt pour éliminer l'agent special(totalement descentralisé) dans les algorithmes 2 et 3, donc le message *arrêt* n'est plus envoyé par un agent spécial ; ce processus sera plus détaillé dans la suite.

#### 3.1.1 Les procédures

```
Procédure démarre()
   $w \leftarrow premier(préférences(m));$ 
  propose( $m, w$ );
Fin
Procédure propose( $m, w$ )
  envoieMsg propose( $m, w$ );
Fin
Procédure mariage( $m, w$ )
   $w \leftarrow m;$ 
  Pour  $p$  après  $m$  dans  $préférences(w)$  faire
    envoieMsg rejet( $w, m$ );
    effacer( $m, préférences(w)$ )
  Fin Pour
Fin
```

```

Procédure homme(m)
  m ← libre;
  fin ← faux
  Tant que (  $\neg fin$  ) faire
    Si ((m == libre) ET (préférences(m)  $\neq \emptyset$ )) Alors
      démarre();
    Fin Si
    msg ← lireMsg ();
    Selon que
      accepte(m, w) : m ← w;
      rejet(m, w) : effacer(w, préférences(m))
        Si (m == w) Alors
          m ← libre;
        Fin Si
        démarre();
      arrêt : fin ← vrai;
    Fin Selon que
  Fait
Fin

```

Algorithme 4: La procédure homme dans la version orientée homme et agent de l'algorithme DisEGS.

```

Procédure femme( $w$ )
   $w \leftarrow libre$ ;
   $fin \leftarrow faux$ ;
  Tant que ( $\neg fin$ ) faire
     $msg \leftarrow lireMsg()$ ;
    Selon que
      propose( $m, w$ ) : Si ( $\neg(m \in list(w))$ ) Alors
        envoieMsg rejet( $m, w$ );
      Sinon
        envoieMsg accepte( $w, m$ );
        mariage( $m, w$ );
      Fin Si
    arrêt:  $fin \leftarrow vrai$ ;
  Fin Selon que
Fait
Fin

```

Algorithme 5: La procédure femme dans la version orientée homme et agent de l'algorithme DisEGS.

```

Procédure arrêt(m)
  satisfaction ← faux;
  Tant que (  $\neg fin$  ) faire
    Si ((m¬libre) ET (listeMessage ≠ ∅)) Alors
      | satisfaction ← vrai;
    Fin Si
    msg ← lireMsg ();
    Selon que
      | jeton(couleur): Si (couleur == blanc ET satisfait) Alors
        | | envoieMsg jeton(blanc);
        | Sinon
        | | envoieMsg jeton(noir);
        | Fin Si
    Fin Selon que
  Fait
Fin

```

Algorithme 6: Mécanisme d'arrêt dans l'algorithme DisEGS.

**Mécanisme d'arrêt.** C'est un processus asynchrone pour arrêter la procédure homme et la procédure femme. Il existe un unique jeton qui circule d'agent en agent ordonnés par un numéro et qui forment un cycle. Le premier agent homme commence le cycle, il envoie le jeton blanc dès lors qu'il est satisfait<sup>1</sup>. Il demande l'arrêt<sup>2</sup> à tous les autres agents quand il reçoit le jeton blanc du dernier agent. Dans l'algorithme, on montre la procédure d'un agent *m*, si tous les agents précédents sont satisfaits, l'agent *m* reçoit un jeton blanc sinon noir. Si l'agent *m* a reçu un jeton blanc et qu'il est satisfait, il envoie un jeton blanc à l'agent successeur, dans les autre cas envoie un jeton noir. Comme indique la figure 3.1, ce mécanisme est inspiré de [Dijkstra & Scholten, 1988]

Le message échangé *jeton (couleur)* indique que l'agent a reçu le jeton de couleur indiquée (blanc ou noir).

<sup>1</sup>il est satisfait si et seulement s'il est marié et si la liste de messages en attente de réponse est vide.

<sup>2</sup>il envoie le message arrêt à tous les autres agents.



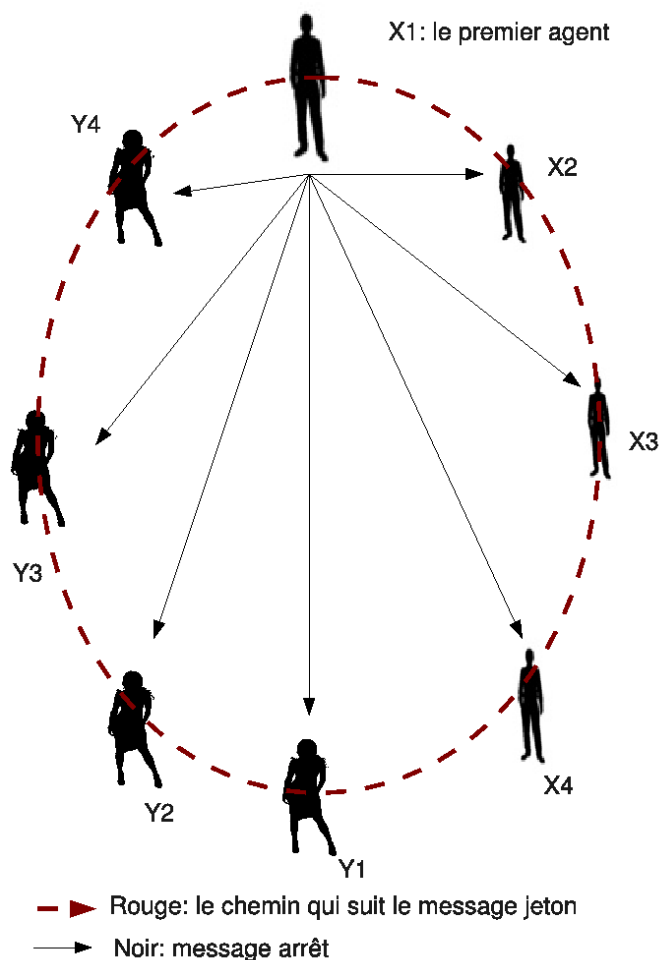


FIG. 3.1 – Mécanisme d'arrêt

## 3.2 Analyse de DisEGS

Pour l'analyse de DisEGS, nous avons pris en compte trois points comme l'égalité, la *privacité*<sup>3</sup> et le bien-être social.

### 3.2.1 L'égalité entre agent

Dans DisEGS, il existe deux types de comportements d'agents. Des agents proposent et les autres agents disposent (figure 3.2). Nous avons trouvé que cet algorithme sou tend une inégalité car il existe un avantage pour les agents du sexe orienté qui font des proposition, les autres agents sont limités à répondre à ces propositions.

<sup>3</sup>nous définirons *privacité* comme l'établissement des limites qui permettraient de partager la liste des préférences avec un autre individu ou un groupe d'individus (agents)

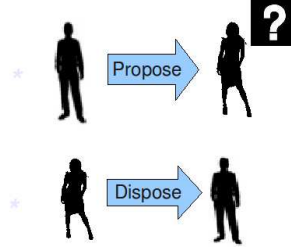


FIG. 3.2 – DisEGS Homme orientée : deux phases

### 3.2.2 privacité

Nous nous fixons comme objectif de privacité que les agents ne partagent pas leur listes de préférences avec les autres agents. Dans EGS, c'est un algorithme centralisé, tous les agents communiquent leur liste de préférences pour trouver un partenaire stable, c'est-à-dire un mariage stable. À la différence de DisEGS où tous les agents garde sa propre liste de préférence et chaque agent est libre de manipuler et la modifier lors du le processus (figure 3.3).

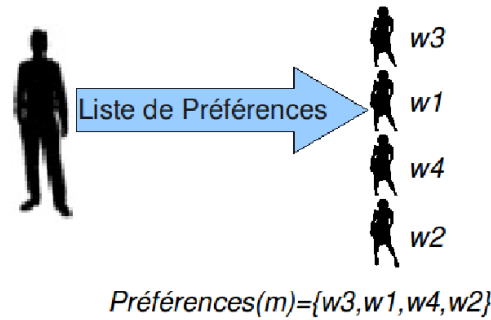


FIG. 3.3 – liste des préférences

### 3.2.3 Bien-être social

Pour le bien-être social dans DisEGS, nous avons considéré la satisfaction individuelle donc le bien-être individuel. Pour évaluer le niveau de satisfaction (bonheur) de chaque agent, nous examinons la position de son partenaire dans la liste des préférences, donc la fonction de satisfaction pour l'agent  $x$  :

$$u_x = \frac{1}{\|position_{LP(x)}(partenaire(x))\|}$$

On doit évaluer de manière disjonctive car il existe deux type d'orientation. c'est-à-dire orientée homme (ou femme) où les agents hommes (femmes) font des proposition aux femmes (hommes) Nous avons utilisé le bien-être utilitaire (figure 3.4). on dit disjonctive car soit les hommes dans le cas orientée homme, soit les femmes maximisent leur bien-être.

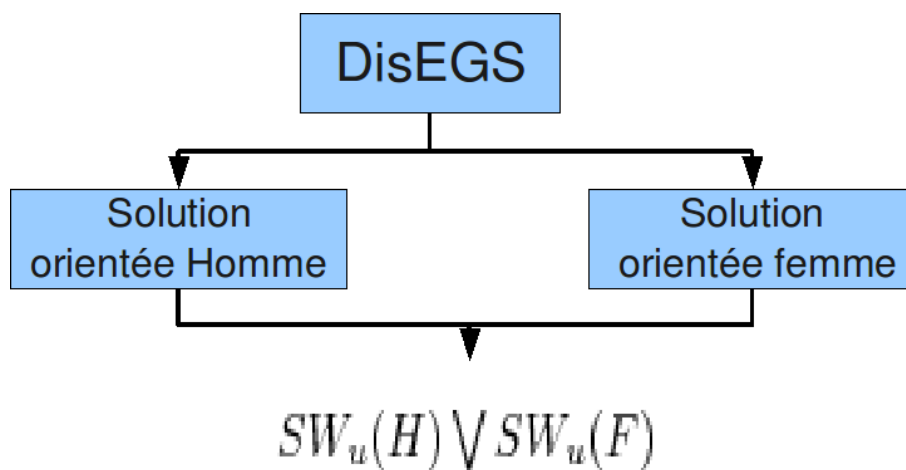


FIG. 3.4 – Bien-être social DisEGS



# 4

---

## Présentation de notre approche

Dans ce chapitre nous présentons notre proposition d'algorithme. Nous avons essayé une option pour résoudre le problème de SMP ; cette option a consisté à unifier les procédures *proposer* et *disposer* de l'algorithme DisEGS, mais les résultats observés ont été que certains agents ont vidé leur liste de préférences ; sans sa liste de préférences, l'agent reste seul, donc il n'y avait pas solution. Finalement nous avons trouvé un algorithme que nous l'avons appelé *Casanova* qui résout SMP en d'accord avec nos objectifs (égalité, privacité, bien-être social).

### 4.1 Algorithme Casanova

Nous avons appelé notre algorithme Casanova car le comportement de l'agent est "*symbole de séduction*" comme Giacomo Girolamo Casanova (figure 4.1 <sup>1</sup>). Dans Casanova, l'agent cherche à ce que leur partenaire soit dans les premiers positions de sa liste de préférences. Au moment qu'il est marié, il fait des proposition aux autres personnes plus préférées que son actuel partenaire. Soit les figures 4.3 et 4.4 un exemple de comportement de l'agent  $a$  avec  $\text{préférences}(a)=[e,f,d]$ .



FIG. 4.1 – Photo de Casanova

---

<sup>1</sup>[http : \www.occultopedia.com\images\\_casanova.jpg](http://www.occultopedia.com/images_casanova.jpg)

## 4.1.1 Les procédures

```

Procédure Casanova(a)
  partenaire  $\leftarrow$  libre;
  fin  $\leftarrow$  faux;
  niveauAcceptable  $\leftarrow$  1;

  propose(niveauAcceptable);
  Tant que ( ( $\neg$ fin) OU (listeMessage  $\neq \emptyset$ ) ) faire
    msg  $\leftarrow$  lireMsg (b);
    j  $\leftarrow$  position de b en préférences(a);
    Selon que
      propose(b,a) : Si (j  $\preceq$  niveauAcceptable) Alors
        envoieMsg accepte (a, b);
        mariage(a, b);
      Sinon
        envoieMsg rejet(a, b);
      Fin Si
      accepte(a,b) : Si (j  $\preceq$  niveauAcceptable) Alors
        mariage(a,b);
      Sinon
        envoieMsg(rejet, a, b);
      Fin Si
      rejet(a,b) : Si (j == niveauAcceptable ) Alors
        partenaire  $\leftarrow$  libre;
        niveauAcceptable ++;
        propose(niveauAcceptable);
      Fin Si
    arrêr : fin  $\leftarrow$  vrai;
  Fin Selon que
Fait
Fin

```

Algorithme 7: Algorithme Casanova de l'agent  $a$

```

Procédure propose(k)
  Pour (x=1 ; x ≤ k ; x++) faire
    v = bx ∈ préférences(a);
    [ x est la position dans la liste de préférences]
    envoieMsg(propose, a, v);
  Fin Pour
Fin

Procédure mariage(a, b)
  envoieMsg(rejet, a, partenaire);
  partenaire ← b;
  niveauAcceptable ← j;
Fin

```

## 4.2 Analyse de Casanova

Comme en DisEGS notre analyse est basé sur l'entre les sexe, privacité et le bien-être social.

### 4.2.1 L'égalité entre agents

Nous avons intégré la caractéristique d'être égalitaire à notre solution, tous les agents ont la avantage de proposer et de disposer.

### 4.2.2 Privacité

Comme DisEGS, il existe la *privacité* sur la liste de préférences, tous les agents manipulent leur liste de préférences, on a continué avec cette approche.

### 4.2.3 Bien-être social

On utilise la même fonction  $u_x$ , mais à différence de DisEGS nous utilisons le bien-être égalitaire pour évaluer (figure 4.2).

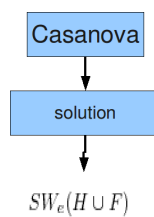


FIG. 4.2 – bien-être égalitaire

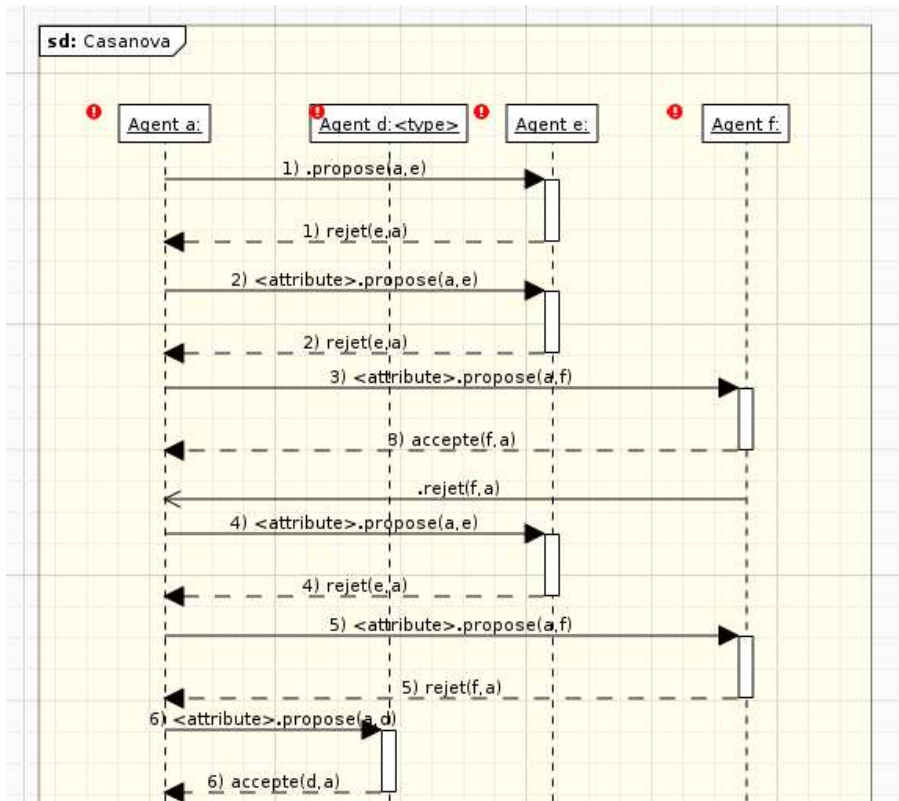


FIG. 4.3 – diagramme de séquence *propose agent a*

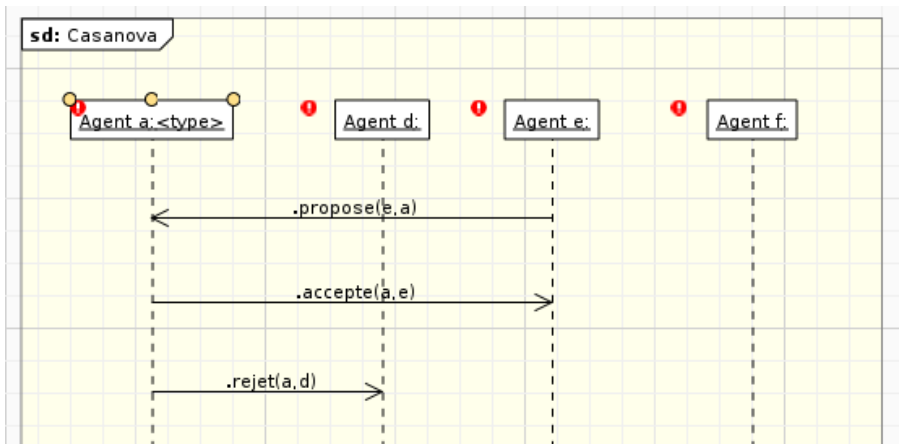


FIG. 4.4 – diagramme de séquence *dispose agent a*



# 5 Validation

---

Dans ce chapitre nous décrirons des expérimentations que nous avons réalisé pour valider. Nous avons cherché à comparer les approches (DisEGS et Casanova), nous avons pris en compte des critères comme le temps de résolution, le nombre de messages.

Premièrement nous avons implémenté EGS centralise en Prolog. Cette implémentation donne effectivement des mariages stables de références. Pour faire des tests, nous avons développé un générateur d'exemples.

## 5.1 Exemple pour l'expérimentation

Nous avons pris l'exemple suivant pour l'expérimentation :

Soit 3 hommes ( $X_1, X_2, X_3$ ) et 3 femmes ( $Y_1, Y_2, Y_3$ )

préférences( $X_1$ ) : $Y_2 Y_1 Y_3$	préférences( $Y_1$ ) : $X_1 X_2 X_3$
préférences( $X_2$ ) : $Y_1 Y_3 Y_2$	préférences( $Y_2$ ) : $X_3 X_1 X_2$
préférences( $X_3$ ) : $Y_3 Y_1 Y_2$	préférences( $Y_3$ ) : $X_2 X_1 X_3$

mariages stables :  $M = \{[(X_2, Y_1)(X_1, Y_2)(X_3, Y_3)]; [(X_1, Y_1)(X_3, Y_2)(X_2, Y_3)]\}$ .

## 5.2 Expérimentation DisEGS

Nous avons implémenté DisEGS en Jason<sup>1</sup>. Aussi, nous avons implémenté un générateur d'exemples aléatoirement suivant les contraintes exprimées dans , on donne le nombre des agents la complétude des préférences et il produit le code Jason.

Nous avons expérimenté notre exemple avec la version orientée homme, où les agents hommes ont un rôle de "propose.asl" et les agents femmes ont un rôle de "dispose.asl". Nous avons trouvé la première réponse aux 25e cycle, et la deuxième aux 125e cycle, il prend un temps de 4 secondes (figure 5.1).

---

<sup>1</sup>[http : \jason.sourceforge.net\JasonWebSite\Jason%20Home.php](http://\jason.sourceforge.net\JasonWebSite\Jason%20Home.php)

### 5.3 Expérimentation Casanova

En Casanova, les agents ont un seul comportement ("codeCasanova.asl"), pour notre exemple, nous avons regardé que Casanova prend un temps de 7 secondes et, il trouve la première réponse aux 51<sup>e</sup> cycle et la deuxième aux 271<sup>e</sup> cycle (figure 5.2).

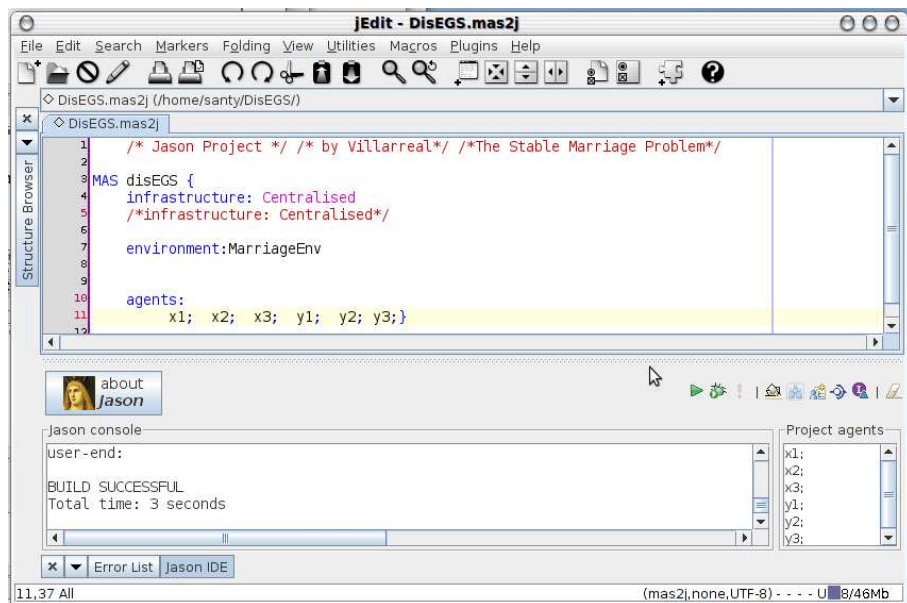


FIG. 5.1 – DisEGS en Jason

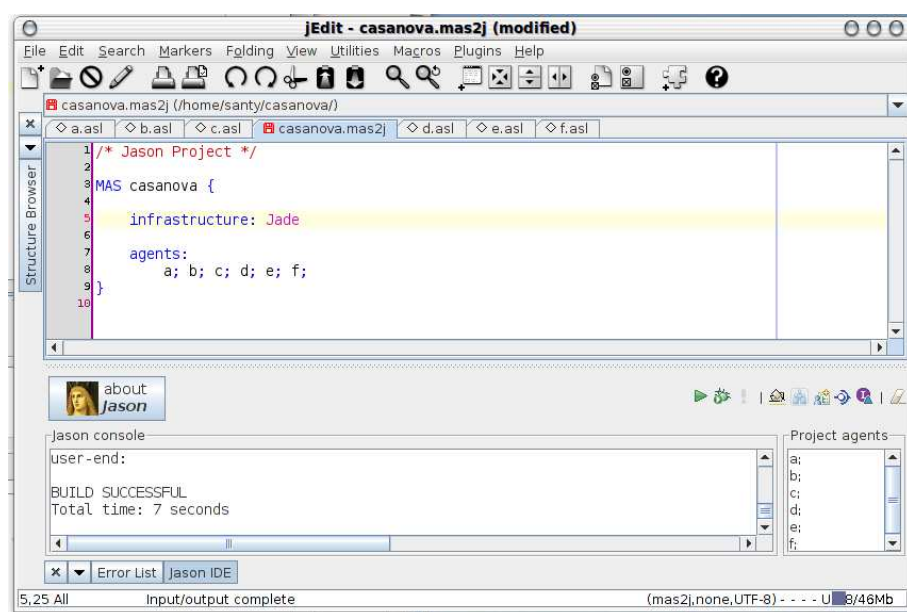


FIG. 5.2 – Casanova en Jason

## 5.4 Comparaison entre DisEGS et Casanova

Pour une comparaison entre DisEGS et Casanova, nous avons pris en compte le nombre de cycles pour obtenir une réponse, on a regardé qu'il existe plus cycles dans Casanova, aussi il prend plus de temps pour *s'arrêter*.

C'est clair que les agents envoient plus message dans Casanova que DisEGS, par contraire nous n'avons pas réalisé un conteur de message pour DisEGS ni pour Casanova, donc on ne peut pas être précis le pourcentage de différences.



---

# Conclusion

Dans ce document, nous parlons de l'analyse de la littérature nécessaire pour mon stage Master Recherche que j'ai réalisé dans l'équipe ISCOD de ENSMSE. Nous avons abordé des définitions de base comme : un CSP, un agent, un SMA. Ces définitions, nous permettons définir un DisCsP, un DisCOP, le choix social et le bien-être social. Nous avons illustré notre problème à résoudre avec un exemple (SMP).

Nous avons étudié SMP et les solutions existantes (EGS et DisEGS). Nous avons trouvé que ces solutions ont des inégalités pour certaines agents, ces solutions ne garantis pas de *privacité* sur les préférences des agents. Surtout, nous avons trouvé qu'il n'existe pas une évaluation du bien-être dans leurs réponses.

Notre travail a consisté à proposer un algorithme capable de maximiser le bien-être social, de respecter la *privacité* sur la liste de préférences d'agents et leur égalité. Notre approche, Casanova a ces caractéristiques, nous avons utilisé le produit de Nash ou le bien-être égalitaire pour l'évaluation du bien-être, pour le bien-être, aussi nous considérons que l'égalité de sexe (égalité entre les agents) est important.

Les pistes futures, nous avons proposé, implémenté et comparé le algorithme Casanova, mais nous n'avons pas évalué la complexité algorithmique des agents ni complexité de communication (le nombre des message envoyés). Aussi, il manque un processus additionnel que a la fin il donne le résultat qui réfléchit maximiser le bien-être social quand il y a des solutions.



---

# Bibliographie

- BRITO I. & MESEGUER P. (2005). Distributed stable matching problems. In *Principles and Practice of Constraint Programming - CP 2005*, number 3709 in Lecture Notes in Computer Science, p. 152–166 : Springer Berlin / Heidelberg.
- BRITO I. & MESEGUER P. (2006). Distributed stable matching problems with ties and incomplete lists. In *In Proceedings of CP '06, LNCS*, p. 675–679 : Springer.
- DELAHAYE J.-P. & MATHIEU P. (2009). La répartition idéale des biens existe-t-elle ? *Pour La Science*, **381**, 88–93.
- DIJKSTRA E. & SCHOLTEN C. (1988). Termination detection for diffusing computations. *INFO. PROC. LETT.*, **11**(1), 1–4.
- GALE D. & SHAPLEY L. (1962). College admissions and the stability of the marriage. *American Mathematical Monthly*, **69**, 9–15.
- GAUDREAU J. (2009). *Algorithmes pour la prise de décision distribuée en contexte hiérarchique !* PhD thesis, Université de Montréal.
- MODI P. J., SHEN W.-M., TAMBE M. & YOKOO M. (2005). Adopt : asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, **161**, 149–180.
- MOULIN H. (1986). *Choosing from a Tournamen*. City.
- NONGAILLARD A. (2009). *An agent-based approach for distributed resource allocations*. PhD thesis, Concordia University and University Lille 1.
- SHOHAM Y. & BROWN K. L. (2009). *Multiagent systems algorithmic, game theoretic, logical*. Cambridge.
- WOOLDRIDGE M. & JENNINGS N. (1995.). *Intelligent Agents*.
- YOKOO M. (2001). *Distributed Constraint Satisfaction*. Springer.