HAL
open science

Procédure décentralisée d'affectation d'individus à des activités

Maxime Morge, Antoine Nongaillard

To cite this version:

Maxime Morge, Antoine Nongaillard. Procédure décentralisée d'affectation d'individus à des activités. Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle, 2018, 32 (5-6), pp.627-658. hal-01946000

HAL Id: hal-01946000

https://inria.hal.science/hal-01946000v1

Submitted on 6 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Procédure décentralisée d'affectation d'individus à des activités

Maxime Morge[†] and Antoine Nongaillard[†]

[†]Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France
{maxime.morge,antoine.nongaillard}@univ-lille.fr

Résumé

Dans cet article, nous présentons une modélisation multi-agents pour la formation de coalitions qui correspond à notre cas d'usage. Nous proposons deux procédures d'affectation qui retournent des appariements valides. La première vise à maximiser le bien-être global des individus. La seconde s'assure que tous les individus sont affectés autant que possible à une activité qu'ils préfèrent. Nos expérimentations montrent que les solutions atteintes par ces procédures sont de meilleure qualité que celles obtenues avec des techniques classiques de résolution de problème. De plus, la décentralisation des procédures d'appariement permet d'accélérer leur exécution.

Abstract

In this paper, we introduce an agent-based model for coalition formation which is suitable for our usecase. We propose two clearinghouses mechanisms that return sound matchings. The first aims at maximizing the global welfare of the individuals. The second ensures that all individuals are assigned as much as possible to a preferred activity. Our experiments show that the outcome of our algorithms are better than those obtained with the classical searc/optimization techniques. Moreover, their distribution speeds up their runtime.

Mots clefs. système multi-agents, résolution distribuée de problème, négociation, comportement d'agents, formation de coalition.
Keywords. multi-agent system, distributed problem solving, negotiation, agent behavior, coalition formation.

1 Introduction

Les systèmes multi-agents (SMA) constituent un paradigme de premier ordre pour l'analyse, la conception et l'implémentation de systèmes composés d'entités autonomes en interaction. Afin de concevoir des environnements socio-techniques médiateurs ou simulés, les SMA permettent de modéliser des boucles de rétroaction entre des acteurs hétérogènes dont les prises de décision locales font émerger des phénomènes globaux. Un des défis auquel la communauté SMA fait face consiste à faciliter l'élicitation des préférences des utilisateurs.

Nos travaux s'inscrivent dans un projet de recherche qui vise à comprendre et modéliser les dynamiques de rétroaction se produisant dans un collectif en interaction à la fois au sein d'un réseau social virtuel et d'un réseau social réel. Le cas d'usage sur lequel nous nous concentrons ici concerne un collectif de seniors qui entrent en relation afin d'effectuer des activités ponctuelles. Ce terrain d'expérimentation implique plusieurs milliers d'individus. L'objectif est ici de maximiser les activités partagées afin d'améliorer la cohésion sociale et lutter contre l'isolement chez certains seniors. Nous souhaitons proposer un réseau social numérique afin que les utilisateurs forment des groupes pour pratiquer ensemble des activités. Ce système a pour objectif de suggérer à chaque utilisateur des individus avec lesquels pratiquer ces activités.

Dans cet article, nous présentons une modélisation multi-agents pour la formation de coalitions qui correspond à notre cas d'usage. Un ensemble d'individus doivent être appariés pour pratiquer une des activités qui leur est proposée en fonction des préférences des individus pour leurs alter ego et pour les activités. Nous définissons des règles de décision pour l'élicitation des préférences individuelles et collectives vis-à-vis des coalitions en termes d'efficacité économique, d'équité, de stabilité et de cohésion sociale. En s'appuyant sur la structure bilatérale du problème formalisé, nous proposons ici deux procédures qui retournent des appariements où aucune activité n'est surchargée. La première procédure, dite « sélective », vise à maximiser la satisfaction globale des individus, le bien-être utilitaire. La seconde procédure, dite « inclusive », s'assure que tous les individus sont affectés autant que possible à une activité qu'ils préfèrent. La modélisation multi-agents nous permet de distribuer l'exécution de ces algorithmes décrits sous la forme de comportements d'agents. Nos expérimentations montrent que les solutions atteintes par ces procédures sont de meilleures qualité en termes d'utilitarisme (respectivement d'égalitarisme) que celles obtenues avec des techniques classiques de recherche locale. De plus, la décentralisation des procédures d'appariement permet d'accélérer leur exécution (jusqu'à 5 fois).

Cet article est une version étendue de [13]. D'une part, nous proposons une nouvelle expérience qui montre que la procédure sélective proposée précédemment permet d'atteindre un appariement proche du maximum utilitariste. D'autre part, nous introduisons une procédure inclusive qui retourne un appariement socialement cohésif qui est, en pratique, égalitariste.

La section 2 motive notre approche et la compare aux travaux connexes. Nous introduisons notre problème d'affectation dans la section 3. Nous proposons différentes procédures d'affectation dans la section 4. La distribution de

ces algorithmes est présentée sous la forme de comportements d'agents dans la section 5. La section 6 exhibe nos résultats empiriques. Après un bref retour d'expérience sur notre application pratique (cf section 7), nous synthétisons nos travaux puis nous dressons quelques perspectives (cf section 8).

2 Modélisation multi-agents du processus d'affectation

La théorie du choix social vise à construire et à analyser des processus pour la décision collective où un ensemble d'agents sélectionnent ou classent conjointement un sous-ensemble d'alternatives parmi celles disponibles. L'informatique se préoccupe des questions algorithmiques afin de rendre opérationnelles ces procédures. Nous nous focalisons ici sur une classe de problème d'appariement en particulier. Afin d'illustrer notre cas d'usage, nous considérons en fil rouge le scénario jouet suivant.

Dans le problème de la formation de coalition hédonique, qui a été formalisé dans [7], chaque joueur est doté d'une seule relation de préférence sur l'ensemble des coalitions qui contiennent ce joueur. Notre problème est une spécialisation de ce problème général. Nous pouvons donc représenter notre problème comme un jeu hédoniste. Pour ce faire, nous associons : i) un joueur à chaque activité tel que toutes les coalitions valides sont également préférées ; et ii) un joueur à chaque individu dont les préférences sont déduites des préférences de l'individu. Les préférences qui en résultent ne sont pas additivement séparables [1] comme dans [1], mais nécessitent la génération de listes de coalitions rationnelles dont la représentation est exponentielle [2]. À l'inverse, la représentation succincte des préférences des individus (les préférences sur les activités et sur les pairs sont indépendantes) et la structure bilatérale de notre problème, i.e. le fait que les activités soient des points focaux [18], nous permet de proposer des procédures traitables.

Le problème de sélection des activités de groupe a été proposé dans [6]. Chaque joueur participe à au plus une activité et dans un tel problème, même simplifié [5], ses préférences portent uniquement sur les activités. C'est une généralisation d'un jeu hédonique anonyme. Même si ce problème a été étendu dans [11] pour prendre en compte les relations d'accointances entre individus, ces relations sont encodées par un réseau social, i.e. un graphe non orienté où les nœuds correspondent aux individus et les arêtes représentent des liens de communication entre eux. À l'inverse, les individus sont munis, dans notre problème, d'une relation de préférence sur leur alter ego.

Le problème des hôpitaux/internes a été introduit dans [9]. Ce problème peut être considéré comme une spécialisation du problème de formation de coalition hédonique où un ensemble de résidents doivent être affectés à des hôpitaux étant donnés les préférences des résidents vis-à-vis des hôpitaux et les préférences des hôpitaux pour les résidents. Ce problème a fait l'objet de nombreuses extensions répertoriées dans [12]. À notre connaissance, aucune n'est adaptée à notre cas d'usage.

1. L'utilité pour une coalition est la somme des utilités pour les pairs dans cette coalition.

Comment un agent évalue ses penchants vis-à-vis des alternatives ? Afin de réduire l'effort cognitif de l'utilisateur pour l'élicication des préférences, nous optons ici pour des fonctions d'utilité, i.e. des préférences cardinales. De plus, nous supposons que les préférences sont additivement séparables et que l'évaluation des activités et celle des groupes sont comparables. Bien que leur expressivité soit limitée, nos préférences sont linéaires par rapport au nombre d'individus et d'activités.

Quelle est la « meilleure » solution à un problème de choix collectif ? Dans la littérature, on trouve principalement deux types de règles qui dérivent le choix collectif des préférences individuelles : les premières se basent sur les propriétés désirables de la solution (e.g. la stabilité ou l'efficacité économique) alors que les secondes se basent sur l'agrégation de la satisfaction individuelle des agents (e.g. le bien-être égalitaire). Dans cet article, nous balayons ces deux approches.

Comment atteindre un appariement maximisant le bien-être social ? Les techniques classiques d'optimisation ne sont pas toujours adéquates (cf A). C'est la raison pour laquelle, nous pouvons considérer les algorithmes de Problème d'Optimisation sous Contraintes Distribué (en anglais, *DCOP*) ou les méthodes de recherche locale (en anglais, *LST*). Il a été montré dans [8] que les algorithmes de *DCOP* ne passent pas nécessairement à l'échelle pour les problèmes d'appariement [4]. Nous montrons ici que les techniques de *LST* ne conviennent pas car les fonctions à optimiser possèdent de nombreux optima locaux. C'est la raison pour laquelle nous avons adopté une méthode de résolution multi-agents et notamment une modélisation multi-niveaux comme préconisé par [15].

3 Affectation d'individus à des activités

Nous nous abstrayons ici de l'application pratique visée pour introduire un cadre formel pour la formation de coalition qui modélise dans notre cas d'usage : le problème, les solutions potentielles et leur qualité d'un point de vue individuel et collectif.

3.1 Préférences vis-à-vis des individus et des activités

Nous introduisons ici le problème des individus/activités (IA). Dans une instance de problème IA, les individus privilégient les activités qui leur plaisent avec les partenaires qu'ils apprécient.

Définition 1 (Problème IA): Un **problème d'individus/activités (IA)** de taille (m, n), tel que $m \geq 2$ et $n \geq 1$, est un couple $IA = \langle I, A \rangle$ avec m individus and n activités, où :

— $A = \{a_1, \ldots, a_n\}$ est un ensemble de n activités. Chaque activité a_j a une capacité maximale $c_j \in \mathbb{N}^+$;

— $I = \{1, \ldots, m\}$ est un ensemble de m individus. Chaque individu i est muni,

1. d'une fonction d'intérêt vis-à-vis des activités $v_i : A \cup \{\theta\} \to [-1, 1]$

2. d'une fonction d'affinité vis-à-vis des pairs $w_i : I \setminus \{i\} \to [-1, 1]$

Intuitivement, l'activité nulle (notée θ), qui correspond à ne rien faire, a une capacité infinie : tous les individus peuvent y être affectés quitte à être seul.

Dans la suite de cet article, nous considérons une instance quelconque du problème $IA = \langle I, A \rangle$ de taille (m, n).

Les fonctions d'intérêt (respectivement d'affinité) permettent d'exprimer la répulsion (< 0), l'attirance (≥ 0), voire l'indifférence (0) d'un individu vis-à-vis des activités (respectivement de ses pairs). Par convention, l'intérêt d'un individu pour l'activité nulle est neutre :

$$\forall i \in I \ v_i(\theta) = 0 \tag{1}$$

On déduit de chaque fonction d'intérêt v_i une relation de préférence individuelle sur les activités, c'est-à-dire une relation d'ordre réflexive, totale et transitive sur $A \cup \{\theta\}$. Cette relation (notée $\trianglerighteq_i$) est définie telle que :

$$\forall i \in I \ \forall a_j, a_k \in A \cup \{\theta\} \ (v_i(a_j) \geq v_i(a_k) \Rightarrow a_j \trianglerighteq_i a_k) \tag{2}$$

La relation de préférence stricte correspondante est notée $\triangleright_i$.

Afin d'apprécier la satisfaction pour un individu i d'appartenir à un groupe g, nous prolongeons sa fonction d'affinité sur l'ensemble des groupes auxquels il appartient, noté $G(i) = \{g \subseteq I \mid i \in G\}$. Ainsi la satisfaction pour un individu d'appartenir à un groupe dépend uniquement des autres membres de ce groupe et non pas des autres groupes constitués par les individus restants. En d'autres termes, les préférences sur les groupes sont des préférences purement hédoniques. Pour l'instant, nous définissons de manière abstraite l'affinité d'un individu i pour un groupe :

$$\begin{gathered} w_i : G(i) \to [-1, 1] \\ \forall i \in I \ \forall g \in G(i) \ w_i(g) = \circledcirc_{j \in g \setminus \{i\}} \ w_i(j) \end{gathered} \tag{3}$$

Cette affinité, qui est normalisée ($w_i(g) \in [-1, 1]$), évalue les préférences d'un individu vis-à-vis des 2^{m-1} groupes. Cette fonction d'affinité sera instanciée dans la section 6.

On déduit de chaque fonction d'affinité w_i une relation de préférence individuelle sur les groupe de pairs, c'est-à-dire une relation d'ordre réflexive, totale et transitive sur $G(i)$. Cette relation (notée $\succsim_i$) est définie telle que :

$$\forall i \in I \ \forall g, g' \in G(i) \ (w_i(g) \geq w_i(g') \Rightarrow g \succsim_i g') \tag{4}$$

La relation de préférence stricte correspondante est notée $\succ_i$ et celle d'équivalence $\sim_i$.

La satisfaction pour un individu i d'appartenir à un groupe g pour pratiquer une activité a dépend : (i) de son intérêt pour l'activité ; (ii) de son affinité vis-à-vis du groupe. Pour l'instant, nous définissons de manière abstraite cette utilité :

$$\begin{gathered} u_i : G(i) \times A \cup \{\theta\} \to [-1, 1] \\ \forall i \in I \ \forall g \in G(i) \ \forall a \in A \cup \{\theta\}, \ u_i(g, a) = w_i(g) \oplus v_i(a) \end{gathered} \tag{5}$$

	a	**b**
$\mathbf{v_1}$	.50	.25
$\mathbf{v_2}$	.50	.25
$\mathbf{v_3}$	.50	.25
$\mathbf{v_3}$	.50	.25

j =	**1**	**2**	**3**	**4**
$\mathbf{w_1(j)}$		1.0	−0.5	−1.0
$\mathbf{w_2(j)}$	1.0		0.5	−1.0
$\mathbf{w_3(j)}$	1.0	0.5		−1.0
$\mathbf{w_4(j)}$	1.0	1.0	−1.0	

$$a \succeq_1 b \qquad \{2,1\} \succ_1 \{1\} \succ_1 \{3,1\} \succ_1 \{4,1\}$$
$$a \succeq_2 b \qquad \{1,2\} \succ_2 \{3,2\} \succ_2 \{2\} \succ_2 \{4,2\}$$
$$a \succeq_3 b \qquad \{1,3\} \succ_3 \{2,3\} \succ_3 \{3\} \succ_3 \{4,3\}$$
$$a \succeq_4 b \qquad \{1,4\} \succ_4 \{2,4\} \succ_4 \{4\} \succ_4 \{3,4\}$$

Fig. 1: Préférences cardinales (en haut) et préférences ordinales déduites (en bas) des individus vis-à-vis des activités (à gauche) et des pairs (à droite)

Nou supposons que cette utilité est normalisée ($u_i(g,a) \in [-1,1]$). Cette fonction d'utilité sera instanciée dans la section 6.

Dans la suite de cet article, les fonctions utilités ne sont pas définies pour correspondre à des préférences ordinales comme dans [3] mais nous supposons un accès direct aux utilités comme c'est le cas dans notre application pratique (voir par exemple les préférences représentées dans la figure 3.1).

3.2 Appariement

Nous sommes maintenant en mesure de définir les solutions potentielles au problème d'appariement présenté précédemment.

Nous souhaitons former des coalitions autour des activités.

Définition 2 (Coalition): Une **coalition** est couple $C = \langle a_C, g_C \rangle$ où $a \in A \cup \{\theta\}$ et $g \subseteq I$. La capacité de la coalition (noté c_C) correspond à la capacité de son activité (noté a_C) si $a_C \in A$ et 1 sinon [2]. La taille de la coalition correspond à la cardinalité de son groupe (noté g_C). Une coalition est dit **valide** si sa taille n'excède pas sa capacité : $\text{card}(g_C) \leq c_C$. Une coalition non-vide C est telle que $g_C \neq \emptyset$. C est pour i si $i \in g_C$.

Nous considérons une instance de problème où le nombre d'individus est considérablement plus important que le nombre d'activités ($m >> n$).

Définition 3 (Appariement): Un **appariement** M est représenté par les fonctions $a_M : I \to A \cup \{\theta\}$ et $g_M : I \to \mathcal{P}(I)$ telles que :

$$\forall i \in I,\ a_M(i) = \theta \Rightarrow g_M(i) = \{i\} \tag{6}$$
$$\forall i \in I\ \forall j \in g_M(i),\ a_M(j) = a_M(i) \tag{7}$$
$$\forall i \in I\ \forall j \in I \setminus \{i\},\ a_M(i) = a_M(j) \neq \theta \Rightarrow g_M(i) = g_M(j) \tag{8}$$

2. Les individus sans activité sont seuls.

Par définition, l'affectation d'un individu est une activité, éventuellement l'activité nulle, et chaque individu est associé à un groupe auquel il appartient. Tous les individus qui sont affectés à l'activité nulle sont seuls (cf équation 6). Tous les individus associés les uns aux autres ont la même activité (cf équation 7) et réciproquement tous les individus qui sont affectés à la même activité, à l'exception de l'activité nulle, sont associés les uns aux autres (cf équation 8).

Pour simplifier les notations, la fonction de **participation** d'un appariement M détermine l'ensemble des individus qui pratique chaque activité :

$$\begin{aligned} p_M &: A \cup \{\theta\} \to \mathcal{P}(I) \\ p_M(a) &= \{i \in I \mid a_M(i) = a\} \end{aligned} \tag{9}$$

L'ensemble des pratiquants d'une activité peut être vide. Si $a_M(i) = \theta$, on dit que i est inactif. Dans un appariement M, une activité $a \in A \cup \{\theta\}$ est :

1. surchargée si $\text{card}(p_M(a)) > c_a$;
2. sous-chargée si $\text{card}(p_M(a)) < c_a$;
3. pleine (noté $\text{full}_M(a)$) sinon.

Un appariement est dit **valide** si aucune activité n'y est surchargée.

Un appariement M est une structure de coalition, i.e. une partition d'individus :

$$\forall a \in \ A \cup \{\theta\},\ \langle a, p_M(a)\rangle \text{ est une coalition} \tag{10}$$

$$\bigcup_{a \in A \cup \{\theta\}} p_M(a) = I \tag{11}$$

$$\forall a_i \in A \cup \{\theta\}\ \forall a_j \in A \cup \{\theta\} \setminus \{a_i\},\ p_M(a_i) \cap p_M(a_j) = \emptyset \tag{12}$$

En effet, les équations 10, 11 et 12 se déduisent directement de l'équation 9 et des définitions 2 et 3. On note $C_M(i)$ la coalition dans M qui contient i. Pour clarifier les définitions de stabilité proposées dans la section suivante, nous introduisons la fonction de **ciblage** qui retourne les partenaires rejoints en sélectionnant l'activité a dans l'appariement M :

$$\begin{aligned} o_M &: A \cup \{\theta\} \to \mathcal{P}(I) \\ o_M(a) &= \begin{cases} p_M(a) & \text{si } a \in A, \\ \emptyset & \text{sinon.} \end{cases} \end{aligned} \tag{13}$$

3.3 Rationalité individuelle

Nous sommes maintenant en mesure d'évaluer les satisfactions individuelles vis-à-vis des solutions potentielles.

Chaque individu évalue la coalition à laquelle il appartient et donc l'appariement en fonction de son groupe et de son activité. Un individu i considère qu'une coalition valide C pour lui ($i \in g_C$) est individuellement rationnelle si

et seulement si $u_i(g_C, a_C) \geq 0$. On dit qu'un appariement valide M est **individuellement rationnel** (noté IR) si et seulement si :

$$\forall i \in I,\ u_i(g_M(i), a_M(i)) \geq 0 \tag{14}$$

L'existence d'un appariement valide IR est garantie, e.g. l'appariement trivial (noté M_0) où tous les individus sont inactifs.

On déduit de chaque fonction d'utilité u_i une relation de préférence individuelle sur les coalitions, c'est-à-dire une relation d'ordre réflexive, totale et transitive sur $A \cup \{\theta\} \times G(I)$. L'individu i préfère C à C' (dénoté $C \succsim_i C'$) si $u_i(g_C, a_C) \geq u_i(g_{C'}, a_{C'})$. Un individu préfère un appariement à un autre s'il préfère sa coalition dans le premier. Considérons M et M' deux appariements pour notre instance de problème IA. L'individu i préfère M à M' (noté $M \succsim_i M'$) si et seulement si $C_M(i) \succsim_i C_{M'}(i)$. Les relations de préférence stricte sur les coalitions/appariements sont notées $\succ_i$.

Exemple 1 (Rationalité): En considérant le problème IA représenté dans la figure 3.1, nous nous attarderons sur 3 des 63 appariements valides possibles :

— M_0 avec $p_{M_0}(a) = \emptyset$, $p_{M_0}(b) = \emptyset$ et $p_{M_0}(\theta) = \{1, 2, 3, 4\}$;
— M_1 avec $p_{M_1}(a) = \{1, 2\}$, $p_{M_1}(b) = \{4\}$ et $p_{M_1}(\theta) = \{3\}$;
— M_2 avec $p_{M_2}(a) = \{1, 2\}$, $p_{M_2}(b) = \{3, 4\}$ et $p_{M_2}(\theta) = \emptyset$;

Contrairement à M_0 et M_1, M_2 n'est pas individuellement rationnel car la coalition valide $\langle b, \{3, 4\}\rangle$ n'est rationnelle ni pour 3, ni pour 4. Ces deux individus préfèrent strictement être inactif plutôt que mal accompagné pour l'activité b. À l'inverse, ils préfèrent strictement être ensemble pour faire l'activité a plutôt qu'inactif :

$$\langle a, \{3, 4\}\rangle \succ_3 \langle \theta, \{3\}\rangle \succ_3 \langle b, \{3, 4\}\rangle \tag{15}$$

$$\langle a, \{3, 4\}\rangle \succ_4 \langle \theta, \{4\}\rangle \succ_4 \langle b, \{3, 4\}\rangle \tag{16}$$

3.4 Évaluation sociale

Nous sommes maintenant en mesure d'évaluer les appariements valides du point de vue du collectif.

Une première propriété désirable est la stabilité. Intuitivement, les individus d'une coalition dite « bloquante » souhaitent se séparer pour former leur propre coalition, ce qui rend l'appariement sous-jacent instable.

Une coalition C valide et non-vide **bloque fortement** un appariement M si et seulement si tous les individus de la coalition la préfèrent strictement plutôt que d'être affecté selon M :

$$\forall i \in g_C,\ C \succ_i C_M(i) \tag{17}$$

Un appariement valide est **stable de cœur** (noté CS) si et seulement s'il n'y a pas de coalition valide et non-vide qui le bloque fortement. Par définition, un appariement qui est stable de cœur est individuellement rationnel.

Une coalition C valide et non-vide **bloque faiblement** un appariement M si et seulement si tous les individus de la coalition la préfèrent plutôt que d'être affecté selon M et au moins un individu de la coalition la préfère strictement :

$$\forall i \in g_C,\ C \succsim_i C_M(i) \wedge \exists i \in g_C,\ C \succ_i C_M(i) \tag{18}$$

Un appariement valide est **strictement stable de cœur** (noté SCS) s'il n'y a pas de coalition valide et non-vide qui le bloque faiblement. Par définition, tous les appariements SCS sont *a fortiori* CS.

Un appariement valide M est **stable au sens de Nash** (noté NS) si aucun individu n'a intérêt à dévier unilatéralement vers une autre activité sous-chargée :

$$\begin{aligned} &\forall i \in I\ \forall a \in A \cup \{\theta\} \setminus \{a_M(i)\} \\ &\text{full}_M(a) \vee C_M(i) \succsim_i \langle a, o_M(a) \cup \{i\} \rangle \end{aligned} \tag{19}$$

Un tel appariement est à l'abri des mouvements individuels car les coalitions sont rationnelles pour tous les individus et au moins aussi bonnes que les autres coalitions valides atteignables par un seul mouvement.

Un appariement valide M est **individuellement stable** (noté IS) si et seulement si aucun individu n'a intérêt à dévier unilatéralement vers une autre coalition valide ou si c'est le cas, l'individu qui dévie n'est pas unanimement apprécié par ses nouveaux partenaires :

$$\begin{aligned} &\forall i \in I\ \forall a \in A \cup \{a\} \setminus \{a_M(i)\} \\ &\text{full}_M(a) \\ &\vee \big[\langle a, o_M(a) \cup \{i\} \rangle \succ_i C_M(i) \Rightarrow \exists j \in o_M(a),\ C_M(j) \succ_j \langle a, o_M(a) \cup \{i\} \rangle\big] \end{aligned} \tag{20}$$

Par définition, un appariement IS est *a fortiori* NS.

Un appariement valide M est **contractuellement individuellement stable** (noté CIS) si et seulement aucun individu ne peut bénéficier de son seul mouvement d'une coalition à une autre coalition sans aggraver la satisfaction de l'une ou de l'autre :

$$\begin{aligned} &\forall i \in I\ \forall a \in A \cup \{a\} \setminus \{a_M(i)\} \\ &\text{full}_M(a) \\ &\vee \big[\langle a, o_M(a) \cup \{i\} \rangle \succ_i C_M(i) \Rightarrow \quad \exists j \in o_M(a),\ C_M(j) \succ_j \langle a, o_M(a) \cup \{i\} \rangle \\ &\qquad\qquad \vee \exists j' \in g_M(i) \setminus \{i\}\ , C_M(i) \succ_{j'} \langle a_M(i), g_M(i) \setminus \{i\} \rangle\big] \end{aligned} \tag{21}$$

Par définition, un appariement CIS est *a fortiori* IS. On peut remarquer que, contrairement à un appariement IS, un appariement CIS n'est pas nécessairement IR.

Nous formulons maintenant d'autres propriétés désirables pour mesurer l'efficacité économique d'un appariement. Comme les précédentes, elles ne s'appuient pas sur des comparaisons inter-personnelles.

Un appariement valide M est **parfait** (noté P) si la coalition pour chaque individu est l'une de ses préférés :

$$\forall i \in I\ \forall a \in A \cup \{\theta\}\ \forall g \in G(i),\ C_M(i) \succsim_i \langle a, g \rangle \tag{22}$$

Un appariement M' **domine au sens de Pareto** M si et seulement M' est strictement préféré à M pour au moins un individu et pas pire pour tous les autres :

$$\exists i \in I, C_{M'}(i) \succ_i C_M(i) \tag{23}$$

$$\wedge \quad \forall i \in I, C_{M'}(i) \succsim_i C_M(i) \tag{24}$$

Un appariement est **optimal au sens de Pareto** (noté PO) s'il n'est pas dominé au sens de Pareto par un appariement valide. En d'autres termes, un appariement est optimal au sens de Pareto s'il n'existe pas d'alternative valide pour laquelle tous les agents sont dans une position meilleure ou équivalente.

La stabilité stricte de cœur est une condition suffisante pour l'optimalité au sens de Pareto.

Proposition 1 (Efficacité vs. Stabilité): SCS $\subseteq$ PO

Preuve 1 (Efficacité vs. Stabilité): Nous prouvons par contradiction qu'un appariement valide et strictement stable de cœur est optimal au sens de Pareto. Nous supposons que M est un appariement valide et strictement stable de cœur qui n'est pas optimal au sens de Pareto. Ainsi, il existe un appariement valide M' qui domine M au sens de Pareto. Considérons la coalition $C_{M'}(i)$ pour i qui satisfait les équations 23 et 24. L'équation 18 est vérifiée. En conséquence on conclut que $C_{M'}(i)$ est valide et bloque faiblement M. Ceci est en contradiction avec notre hypothèse.

Contrairement à l'approche parétienne, la théorie du choix social analyse les comparaisons inter-personnelles [19]. Cette théorie réconcilie l'utilitarisme de Jeremy Bentham et la justice distributive prônée par John Rawls [14].

Définition 4 (Bien-être social): Considérons un appariement valide M.

— Le **bien-être utilitaire** de M est défini tel que :

$$U(M) = \frac{1}{m} \sum_{i \in I} u_i(g_M(i), a_M(i)) \tag{25}$$

— Le **bien-être égalitaire** de M est défini tel que :

$$E(M) = \min_{i \in I}(u_i(g_M(i), a_M(i))) \tag{26}$$

Les bien-être sont normalisés ($U(M), E(M) \in [-1, 1]$). Plus ils sont élevés, plus l'appariement est bon. Un appariement valide est un **maximum utilitariste** (respectivement **maximum égalitariste**) - noté MaxUtil (respectivement MaxEgal) s'il maximise le bien-être utilitaire (resp. égalitaire). On peut noter que, si tous les individus et activités sont attractifs, le bien-être égalitaire est nul dès qu'un individu est inactif. Par définition, un appariement valide MaxUtil est *a fortiori* PO. De même, un appariement valide MaxEgal est *a fortiori* IR.

Finalement, en s'inspirant de la psychologie sociale [20], nous pouvons considérer que la cohésion est plus étroitement liée à l'intérêt des activités qu'aux

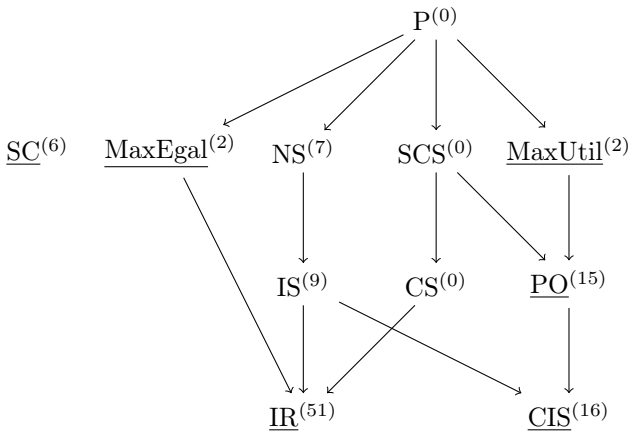

Fig. 2: Relations d'inclusion entre les concepts de stabilité, d'optimalité, d"équité et de cohésion pour le problème IA. L'existence des concepts soulignés est garantie. Les exposants indiquent la répartition des 63 appariements valides de notre exemple (cf figure 3.1) en fonction de leurs propriétés

affinités interpersonnelles. Formellement, un appariement valide M est **socialement cohésif** (SC) si et seulement si tous les individus sont affectés autant que faire se peut à une activité qu'ils préfèrent :

$$\forall i \in I \ \forall a \in \{a \in A \setminus \{a_M(i)\} \mid v_i(a) \geq 0 \wedge v_i(a) > v_i(a_M(i))\}, \ \text{full}_M(a) \quad (27)$$

L'existence d'un appariement valide SC est garantie, e.g. l'appariement trivial quand toutes les activités sont rebutantes. Ce concept est orthogonale aux précédents, et contrairement à la plupart des autres, un appariement valide SC n'est pas nécessairement valide.

Inspirée de [1], la figure 3.4 représente sous la forme d'un treillis les relations d'inclusion entre les concepts présentés dans cette section. Par exemple, l'arc entre SCS et PO indique que le premier concept est une condition nécessaire du second mais, comme l'illustre notre exemple, pas suffisante.

Exemple 2 (Évaluation sociale): En considérant le problème IA représenté dans la figure 3.1, les appariements valides MaxUtil sont :

— M_1 avec $p_{M_1}(a) = \{1, 2\}$, $p_{M_1}(b) = \{4\}$ et $p_{M_1}(\theta) = \{3\}$;
— M'_1 avec $p_{M'_1}(a) = \{1, 2\}$, $p_{M'_1}(b) = \{3\}$ et $p_{M'_1}(\theta) = \{4\}$.

Les appariements valides MaxEgal sont :

— M_3 avec $p_{M_3}(a) = \{1, 4\}$, $p_{M_3}(b) = \{2, 3\}$ et $p_{M_3}(\theta) = \emptyset$;
— M'_3 avec $p_{M'_3}(a) = \{3, 4\}$, $p_{M'_3}(b) = \{1, 2\}$ et $p_{M'_3}(\theta) = \emptyset$.

Les appariements M_1 et M'_1 sont PO mais pas SCS. Contrairement à M_1 et M'_1, les appariements M_3 et M'_3 sont SC mais pas IR.

Comme l'illustre notre exemple, si la stabilité est une propriété désirable, il n'existe pas nécessairement de telles solutions. De plus, l'optimalité au sens de Pareto ne semble pas être suffisamment discriminante. En revanche, les concepts de cohésion et de bien-être sociaux sont de meilleurs candidats pour évaluer la qualité d'un appariement du point de vue de la société d'individus.

4 Procédure centralisée d'appariement

En s'appuyant sur la structure bilatérale du problème IA, nous proposons ici deux algorithmes qui retournent des appariement valides.

Nos algorithmes s'inspirent de l'algorithme d'acceptation différée orienté individu [9] - en anglais, *student-proposing deferred acceptance algorithm*. Chaque individu i est associé à une liste de concessions, i.e. la liste des activités qui l'attirent classées par intérêt décroissant ($\trianglerighteq_i$). D'un côté, l'individu i se propose aux activités par ordre de préférence quitte à concéder en cas de rejet. De l'autre côté, la coalition $\langle a, g\rangle$, en tant qu'hôte, explicite sa contrainte de capacité c_a et les objectifs des individus hébergés, c'est-à-dire une agrégation des bien-être individuels qui peut être ici :

— soit une règle sociale utilitaire

$$\operatorname*{argmax}_g \{\sum_{i\in g} u_i(g,a) \mid card(g) \leq c_a\}, \tag{28}$$

— soit une règle sociale égalitaire

$$\operatorname*{argmax}_g \{\min_{i\in g} u_i(g,a) \mid card(g) \ \leq c_a\}. \tag{29}$$

Dans le premier algorithme représenté dans la figure 4, l'appariement initial est l'appariement trivial, i.e. tous les individus sont inactifs. Tour à tour, chaque individu libre i considère l'activité qui l'attire et qu'il préfère a. Si la coalition est vide, i est affecté. Sinon, cet algorithme tente d'améliorer la satisfaction de cette coalition, éventuellement en excluant les individus dont la présence pénalise la règle sociale (utilitaire ou égalitaire) du groupe. Les individus qui sont exclus, éventuellement i, doivent concéder et donc considérer l'activité attractive suivante. On peut noter que cet algorithme est dit **sélectif** car la règle sociale est évaluée à chaque proposition d'intégration à un groupe existant (ligne 20). Tant que la capacité de a n'est pas atteinte alors le groupe peut croître (ligne 28). Finalement, un individu qui est rejeté par toutes ses activités attrayantes est définitivement inactif (ligne 8).

Comme un groupe de k individus comporte $2^k - 1$ sous-groupes non-vides, nous proposons un algorithme d'approximation qui consiste à n'exclure qu'au plus un individu à chaque étape (ligne 22).

Cet algorithme retourne toujours un appariement valide.

Proposition 2 (Terminaison): Notre algorithme sélectif (éventuellement d'approximation) appliqué à une instance quelconque d'un problème IA se termine et l'appariement retourné est valide.

```
1  val m = new Matching(ia)
2  var free = ia.individuals
3  var concessions = ia.individuals.map{ i =>
4    (i, ia.activities.sortWith((a1, a2) => i.v(a1) > i.v(a2) =>0))
5  }
6  while(! free.isEmpty){// Tq. les inds. ne sont pas tous affectés
7    free.foreach { i: Individual =>// Pour chaque individu libre
8      if (concessions(i).isEmpty){// S'il est désespéré
9        m.a+=(i -> Activity.VOID)
10       m.g+=(i -> new Group(i))
11       free-=i
12     } else {// Sinon il choisit l'activité préféré
13       val a = concessions(i).head
14       val g = m.p(a)
15       val ng = g + i
16       if (g.isEmpty){// Si la coalition est vide
17         m.a+=(i -> a)
18         m.g+=(i -> new Group(i))
19         free-=i
20       } else { // Sinon la coalition est non-vide
21         var sbgs = Set[Group]()
22         if (approximation){// Soit au plus un individu est exclu
23           ng.foreach { j => sbgs+=ng.filterNot(_.equals(j)) }
24           sbgs+=ng
25         } else { // Soit tous les groupes potentiels sont évalués
26           sbgs = ng.sbgs().filterNot(o => o.equals(Group()))
27         }
28         if (a.c == g.size) sbgs-=ng
29         var umax = Double.MinValue
30         var bestG = new Group()
31         sbgs.foreach{ sg =>
32           val u = rule match {// Règle sociale du groupe
33             case Utilitarian => sg.usum(a) //∑_{j∈sg} u_j(sg,a)
34             case Egalitarian => sg.umin(a) //min_{j∈sg} u_j(sg,a)
35           }
36           if (umax < u) {
37             umax = u
38             bestG = sg
39           }
40         }// Le meilleur sous-groupe est sélectionné
41         bestG.foreach { j => m.g+=(j -> bestG) }
42         // Des individus peuvent être exlus
43         (g.diff(bestG)).foreach{j : Individual =>
44           m.a+=(j -> Activity.VOID)
45           m.g+=(j -> new Group(j))
46           free+=j
47           concessions+=(j -> concessions(j).tail)//et concéder
48         }
49         if (bestG.contains(i)){//L'individu i peut être intégré
50           m.a+=(i -> a)
51           free-=i
52         } else concessions+=(i -> concessions(i).tail)//ou concède
53       }
54     }
55   }
56 }
```

Fig. 3: Algorithme sélectif pour le calcul d'un appariement où la coalition est ré-évaluée à chaque fois qu'un individu est intégré

Preuve 2 (Terminaison): Nous considérons l'invariant de boucle $\Sigma_{i \in I}|concessions(i)|+|free|$. Cet invariant est positif ou nul. Il décroît strictement à l'issue de chacune des boucles car :

1. un individu affecté est retiré de $free$;
2. un individu qui n'est pas affecté concède jusqu'à être éventuellement inactif ;
3. un individu, qui est exclu, concèdent et au moins un autre individu est affecté (donc retiré de $free$).

L'appariement obtenu est valide car les activités n'y sont jamais surchargées.

Exemple 3 (Algorithme sélectif): Considérons le problème IA représenté dans la figure 3.1. Quelque soit la règle sociale adoptée, notre algorithme sélectif permet d'atteindre M_1 avec $p_{M_1}(a) = \{1, 2\}$, $p_{M_1}(b) = \{4\}$ et $p_{M_1}(\theta) = \{3\}$. On vérifie bien que cet appariement est optimal au sens de Pareto mais pas socialement cohésif. La partie gauche de la figure 3 présente une trace de notre algorithme où l'individu 3 termine inactif.

Dans le cas général, on peut noter que l'algorithme sélectif ne maximise pas nécessairement le bien-être utilitaire (respectivement égalitaire) en appliquant la règle sociale utilitaire (respectivement égalitaire) car les individus donnent la priorité à l'intérêt des activités plutôt qu'à l'affinité des groupes. Malgré cette priorité, l'appariement résultant n'est pas nécessairement socialement cohésif. C'est la raison pour laquelle nous proposons l'algorithme inclusif représenté dans la figure 4. Contrairement à l'algorithme précédent, la règle sociale d'une coalition, en tant qu'hôte, n'est évaluée que lorsque sa contrainte de capacité est violée (ligne 16) et l'exclusion d'un seul individu est envisagé à chaque intégration (ligne 22).

L'algorithme inclusif retourne toujours un appariement valide qui est socialement cohésif quelque soit la règle sociale employée.

Proposition 3 (Terminaison): L'algorithme inclusif appliqué à une instance quelconque d'un problème IA se termine et l'appariement retourné est valide.

Preuve 3 (Terminaison): Idem à la preuve 2.

Proposition 4 (Cohésion sociale): L'algorithme inclusif appliqué à une instance quelconque d'un problème IA retourne un appariement socialement cohésif.

Preuve 4 (Cohésion sociale): Nous montrons par contradiction que le résultat M de l'algorithme inclusif est un appariement socialement cohésif. Nous supposons qu'il existe un individu i et une activité $a \neq a_M(i)$ telle que $v_i(a) \geq 0$, $v_i(a) > v(a_M(i))$ et qui est sous-chargée (cf équation 27). Comme i est attirée par a, cet individu s'est proposé à cette activité qui est dans liste de concessions (ligne 4) avant d'être définitivement inactif. Il a donc été exclu par cette activité qui est pleine car les groupes potentiels envisagés sont de taille c_a (ligne 22). Ceci est une contradiction avec notre hypothèse.

```
M: a(2): [] b(2): [] θ: [1, 2, 3, 4]
Comme a est inoccupé,
4 est affecté à a

M: a(2): [4] b(2): [] θ: [1, 2, 3]
3 choisit a avec [4]
[3].usum(a)=0,250
[4].usum(a)=0,250
[4, 3].usum(a)=0,167
Comme [3] est sélectionnée,
4 est exclut de a
3 est affecté à a

M: a(2): [3] b(2): [] θ: [1, 2, 4]
2 choisit a avec [3]
[2].usum(a)=0,250
[3].usum(a)=0,250
[3, 2].usum(a)=0,667
Comme [3, 2] est sélectionnée,
2 est affecté à a

M: a(2): [2, 3] b(2): [] θ: [1, 4]
1 choisit a avec [2, 3]
[3, 2].usum(a)=0,667
[1].usums(a)=0,250
[3].usum(a)=0,250
[2, 1].usum(a)=0,833
[2].usum(a)=0,250
[3, 1].usum(a)=0,583
Comme [2, 1] est sélectionnée,
3 est exclut de a
1 est affecté à a

M: a(2): [1, 2] b(2): [] θ: [3, 4]
Comme b est inoccupé,
3 est affecté à b

M: a(2): [1, 2] b(2): [3] θ: [4]
4 choisit b avec [3]
[4].usum(b)=0,125
[3].usum(b)=0,125
[3, 4].usum(b)=-0,083
Comme [4] est sélectionnée,
3 est exclut de b
4 est affecté à b

M: a(2): [1, 2] b(2): [4] θ: [3]
3 est désespéré et donc inactif

M: a(2): [1, 2] b(2): [4] θ: [3]
```

```
a(2): [] b(2): [] θ: [1, 2, 3, 4]
4 choisit a avec []
Comme a n'est pas plein,
4 est affecté à a

a(2): [4] b(2): [] θ: [1, 2, 3]
3 choisit a avec [4]
Comme a n'est pas plein,
3 est affecté à a

a(2): [3, 4] b(2): [] θ: [1, 2]
2 choisit a avec [3, 4]
[4, 3].umin(a)=0,083
[4, 2].umin(a)=0,083
[3, 2].umin(a)=0,333
Comme [3, 2] est sélectionnée,
4 est exclut de a
2 est affecté à a

a(2): [2, 3] b(2): [] θ: [1, 4]
1 choisit a avec [2, 3]
[3, 2].umin(a)=0,333
[3, 1].umin(a)=0,167
[2, 1].umin(a)=0,417
Comme [2, 1] est sélectionnée,
3 est exclut de a
1 est affecté à a

a(2): [1, 2] b(2): [3] θ: [4]
4 choisit b avec [3]
Comme b n'est pas plein,
4 est affecté à b

a(2): [1, 2] b(2): [3, 4] θ: []
```

Fig. 4: Trace d'exécution pour l'algorithme exact appliquant la règle utilitaire (à gauche) et pour l'algorithme inclusif appliquant la règle égalitaire (à droite)

```
val m = new Matching(ia)
var free = ia.individuals
var concessions = ia.individuals.map{ i =>
  (i, ia.activities.sortWith((a1, a2) => i.v(a1) > i.v(a2) =>0))
}
while (! free.isEmpty){// Tq. les inds. ne sont pas tous affectés
  free.foreach { i: Individual =>// Pour chaque individu libre
    if (concessions(i).isEmpty){// S'il est desespéré
      m.a+=(i -> Activity.VOID)
      m.g+=(i -> new Group(i))
      free-=i
    } else {// Sinon il choisit l'activité préféré
      val a = concessions(i).head
      val g = m.p(a)
      val ng = g + i
      if (!a.isFull){// Si la capacité n'est pas atteinte
        m.a+=(i -> a)
        m.g+=(i -> new Group(i));
        free-=i
      } else {// Sinon la capacité est atteinte
        var sbgs = Set[Group]()
        ng.foreach { j => sbgs+=ng.filterNot(_.equals(j)) } // L'exclusion d'un seul individu est envisagée
        var umax = Double.MinValue
        var bestG = new Group()
        sbgs.foreach{ sg =>
          val u= rule match {// Règle social du groupe
            case Utilitarian => sg.usum(a) //$\sum_{j\in sg} u_j(sg,a)$
            case Egalitarian => sg.umin(a) //$\min_{j\in sg} u_j(sg,a)$
          }
          if (umax < u) {
            umax = u
            bestG = sg
          }
        }// Le meilleur sous-groupe est sélectionné
        bestG.foreach { j => m.g+=(j -> bestG) }
        // Au plus un individu peut être exlu
        (g.diff(bestG)).foreach {j : Individual =>
          m.a+=(j -> Activity.VOID)
          m.g+=(j -> new Group(j))
          free+=j
          concessions+=(j -> concessions(j).tail)//et concéder
        }
        if (bestG.contains(i)){//L'individu i peut être affecté
          m.a+=(i -> a)
          free-=i
        } else concessions+=(i -> concessions(i).tail)//ou concède
      }
    }
  }
}
```

Fig. 5: Algorithme inclusif pour le calcul d'un appariement où seules les coalitions pleines sélectionnent les individus

Exemple 4 (Algorithme inclusif): Considérons le problème IA représenté dans la figure 3.1. Quelque soit la règle sociale adoptée, l'algorithme inclusif permet d'atteindre M_2 avec $p_{M_2}(a) = \{1, 2\}$, $p_{M_2}(b) = \{3, 4\}$ et $p_{M_2}(\theta) = \emptyset$. On vérifie bien que cet appariement est socialement cohésif mais pas individuellement rationnel. La partie droite de la figure 3 présente une trace de notre algorithme inclusif où tous les activités sont pleines.

5 Comportement d'agent

Nous décrivons ici les procédures d'appariement présentées précédemment sous la forme de comportement d'agents afin de distribuer leur exécution.

Nous adoptons le modèle de programmation concurrente par passage de messages asynchrones, appelé modèle d'acteur, proposé dans [10]. Les primitives de ce modèle sont les agents et les événements : un agent est un programme indépendant qui s'exécute sur son propre processeur ; un événement correspond à la création d'un agent ou à l'émission/la réception d'un message. On suppose que les canaux de communication sous-jacents sont fiables (un message est délivré une et une seule fois) et que les messages peuvent arriver dans un ordre différent de celui de l'émission.

Afin de proposer un moteur d'appariement distribué à partir de ce modèle, nous distinguons trois types d'agents :

1. l'agent « solveur » qui crée les autres agents et enregistre les affectations ;
2. les agents « individu » tous munis d'un même comportement mais avec des préférences hétérogènes ;
3. les agents « coalition » tous munis d'un même comportement mais qui gèrent des contraintes de capacité différentes et des objectifs de groupe dissemblables.

Le comportement de l'agent « solveur » consiste à : i) créer les agents ; ii) lancer la résolution ; iii) enregistrer les affectations et les exclusions ; iv) et renvoyer l'appariement quand tous les individus sont affectés ou définitivement inactifs.

Que la procédure soit sélective ou inclusive, le comportement d'un agent « individu » est identique (cf figure 5). Il consiste à se proposer à l'activité préférée qui l'attire. Quand l'agent est affecté ou exclu, il en informe l'agent « solveur ». Comme nous le verrons dans la suite, lors d'une exclusion, l'agent « coalition » attend une confirmation de la part de l'agent exclu avant de procéder effectivement à l'intégration d'autres agents afin que l'appariement ne soit pas retourné prématurément par l'agent « solveur ». Si un agent « individu » devient libre, il concède (ligne 19), puis il se propose à l'activité suivante qui l'attire jusqu'à être définitivement inactif. Conformément au principe de subsidiarité, la reconfiguration d'une coalition peut-être soumis à l'arbitrage de l'agent « coalition » qui requiert l'opinion des agents « individu » concernés (ligne 23).

Afin de décrire le comportement d'un agent « coalition », nous utilisons le patron de conception de programmation concurrente FSM (*Finite State Machine*) avec trois états :

```
def receive = {
  case Start => { // Le comportement est activé
    if (! concessions.isEmpty) concessions.head ! Propose(i)
    else solver ! Assignement(i, VOID)
  }
  case Accept => { // L'affectation est confirmée
    solver ! Assignement(i, concessions.head)
  }
  case Reject => {// L'affectation est infirmée
    concessions = concessions.tail
    if (concessions.isEmpty) solver ! Assignement(i, VOID)
    else concessions.head ! Propose(i)
  }
  case Eject => {// L'agent est exclu
    solver ! Disassignement(i)
  }
  case Acknowledge => {// L'exclusion confirmée par le solveur
    concessions.head ! Confirm
    concessions = concessions.tail
    if (concessions.isEmpty) solver ! Assignement(i, VOID)
    else concessions.head ! Propose(i)
  }
  case Query(g,a) => {// L'utilité est demandé
    sender ! Reply(g,a,i.u(g,a))
  }
}
```

Fig. 6: Comportement de l'agent « individu » i. L'émission (noté `!`) de message est déclenchée par la réception d'une locution (`Start`, `Accept`, etc.)

1. l'état initial de disposant (`Disposing`), où chaque proposition est traitée ;
2. l'état de reconfiguration (`Casting`), où l'opinion d'agents « individu » est sollicitée afin de déterminer le périmètre du groupe conformément à la règle sociale ;
3. l'état de liquidation (`Firing`), où les agents « individu » exclus sont notifiés.

Les transitions entre ces états sont du type :
`Évenement if conditions ➔ actions`.
Elles sont déclenchées par un événement externe, e.g. la réception d'un message. L'automate est déterministe, i.e. les conditions de déclenchement des transitions sortantes d'un même état sont mutuellement exclusives. Ce patron de conception de comportement d'agent permet de gérer des dialogues multi-parties concurrents. Les propositions reçues par un agent « coalition » sont traitées une-à-une, quitte à être mise de côté (`stash()`) pour être évalués plus tard (`unstash()`), mais leur acceptation/rejet dépendent de l'opinion des autres agents « individu » concernés.

Par exemple, la figure 5 décrit le comportement de l'agent représentant l'activité $\langle a, g\rangle$ pour la procédure sélective sans approximation. Selon ce comportement, une proposition d'un individu i reçue dans l'état `Disposing` est acceptée si la coalition est vide. Sinon, l'agent évalue cette proposition dans l'état `Casting` en sélectionnant un sous-groupe non-vide de $g \cup \{i\}$ qui respecte la contrainte de capacité et l'objectif de la coalition, i.e. la règle sociale. Ensuite, si le proposant n'est pas membre du groupe choisit alors la proposition est rejetée. Dans le cas contraire, la proposition est acceptée dès que, si des membres de g ont été écartés, ceux-ci confirment la notification de l'agent « solveur » (dans l'état `Firing`). Pour les autres procédures d'affectation, le comportement des agents « coalition » est similaire à l'exception des conditions et des actions pour les transitions sortantes de l'état `Disposing`. Par exemple, pour la procédure inclusive, la reconfiguration de la coalition n'a lieu que lorsque la capacité est atteinte.

Finalement, on peut remarquer qu'une résolution peut atteindre des résultats différents à partir de la même configuration. Le non-déterminisme observable est nécessaire pour distribuer la résolution de problèmes au travers de programmes indépendants.

6 Évaluation empirique

Nos expérimentations [3] visent à évaluer : i) l'exploitation des propriétés structurelles du problème par nos procédures ; ii) la qualité des solutions atteintes ; et iii) l'accélération due à leur distribution.

Notre prototype ScaIA (`https://github.com/cristal-smac/ScaIA`) a été implémenté avec le langage de programmation Scala et la boîte à outils Akka [4].

3. Les expérimentations on été réalisées avec un PC doté de 8 cœurs Intel (R) i7 2.8GHz et 16GB de RAM.
4. http ://akka.io

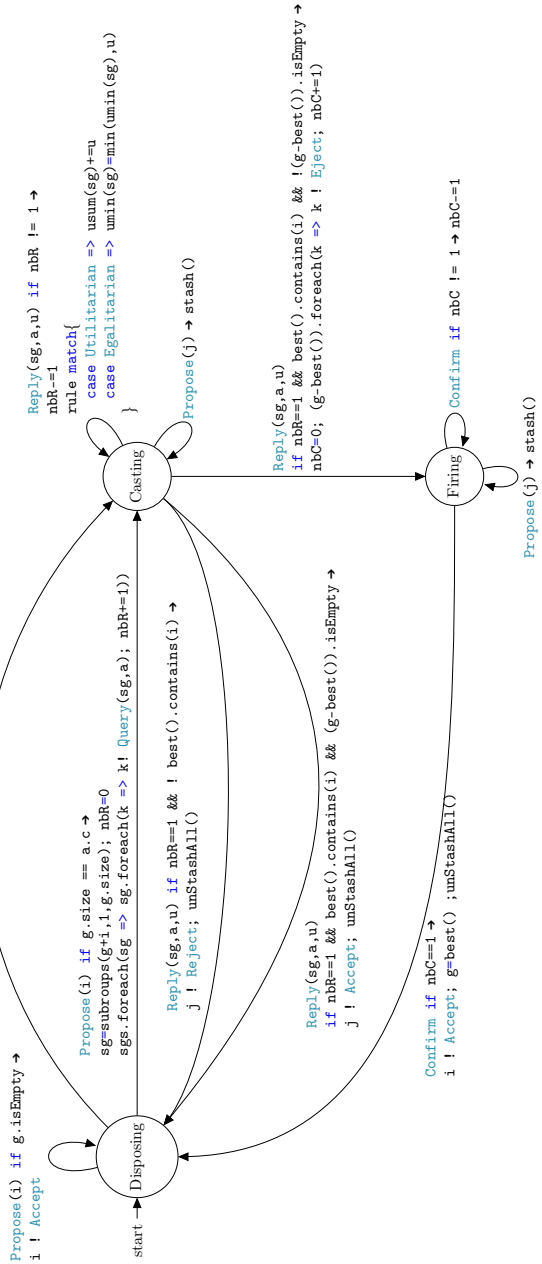

Fig. 7: Comportement de l'agent représentant la coalition $\langle a, g \rangle$ dans la procédure sélective sans approximation. `subgroups(g,min,max)` retourne les sous-groupes de `g` dont la taille est comprise entre `min` et `max`

Cette dernière, en s'appuyant sur le modèle d'acteur [10], nous permet de réduire la distance entre les spécifications de nos comportements d'agent et leur implémentation.

Nos procédures exploitent la structure du problème IA qui, contrairement au jeu hédonique, est bilatérale. Afin d'en mesurer les mérites, d'une part, nous traduisons une instance de problème IA en un jeu hédonique comme indiqué dans la Section 2 et nous en calculons une coalition contractuellement stable (CIS). D'autre part, nous traduisons une instance de problème IA sous la forme d'un problèmes quadratiques mixtes (en anglais, *Mixed Integer Quadratic Problem*) comme indiqué dans l'A pour sa résolution à l'aide de l'outil IBM® ILOG® CPLEX®. Nous considérons des instances de problème IA avec 2 activités et $m \in [2, 20]$ individus. Par souci de simplicité, toutes les activités ont la même capacité ($c = \lceil m/2 \rceil$). Nous générons de manière (pseudo)-aléatoire 100 instances pour chaque valeur de m.

Tout d'abord, nous instancions la définition de la fonction d'affinité et celle de la fonction d'utilité. Inspiré par notre application pratique, nous axiomatisons le principe *quo plus est eo hilarior, sed quo paucior eo melior cibus*[5], en calculant l'affinité d'un individu i pour un groupe de la manière suivante :

$$w_i : G(i) \to [-1, 1]$$
$$\forall i \in I \ \forall g \in G(i) \ w_i(g) = \frac{1}{m-1} \Sigma_{j \in g \setminus \{i\}} \ w_i(j) \tag{30}$$

Cette affinité, qui est normalisée ($w_i(g) \in [-1, 1]$), croit avec la taille du groupe à condition que i soit attiré par tous les membres du groupe. Cette fonction, qui évalue les préférences d'un individu vis-à-vis des 2^{m-1} groupes, est additivement séparable et donc sa représentation est linéaire par rapport au nombre d'individus. Par souci de simplicité, nous calculons la fonction d'utilité de la manière suivante :

$$u_i : G(i) \times A \cup \{\theta\} \to [-1, 1]$$
$$\forall i \in I \ \forall g \in G(i) \ \forall a \in A \cup \{\theta\}, \ u_i(g, a) = \frac{w_i(g) + v_i(a)}{2} \tag{31}$$

Comme chaque facteur $w_i(g), v_i(a) \in [-1, 1]$, cette utilité est normalisée ($u_i(g, a) \in [-1, 1]$).

La figure 6 compare les bien-être utilitaires obtenus par l'algorithme d'approximation sélectif, par l'algorithme CIS proposé dans [2] et par la résolution quadratique (MIQP). Les figures 6 et 6 comparent les temps d'exécution moyens de ces algorithmes avec une échelle de temps logarithmique et en distinguant le temps d'exécution du temps pour la traduction : i) du problème IA en jeu hédonique ou en programme mathématique ; et ii) de la solution. Contrairement à l'optimisation mathématique, le temps d'exécution de notre procédure bénéficie des propriétés structurelles du problème IA. De plus, l'algorithme CIS est pénalisé par la génération des préférences (RLC) dont la représentation n'est pas

5. En français « plus on est de fous, plus on rit » et en anglais *the more the merrier, but the fewer the better fare.*

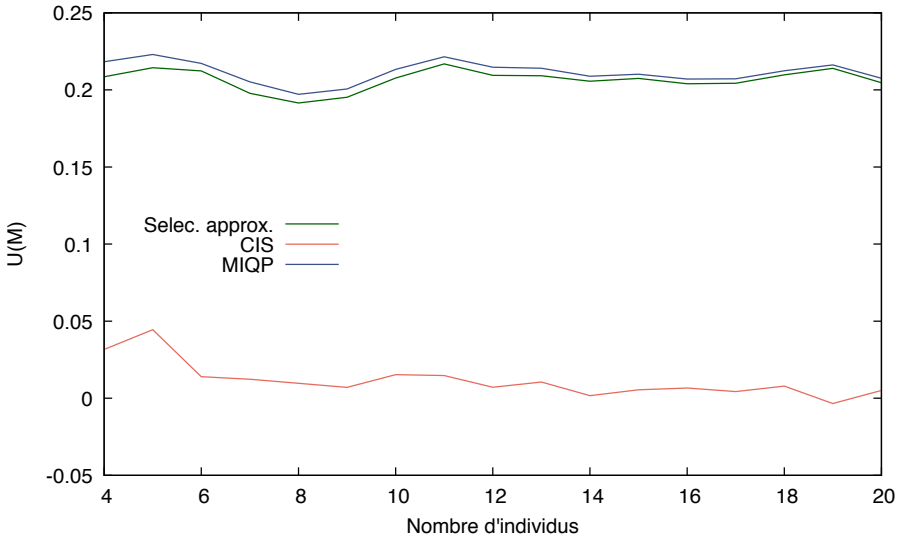

Fig. 8: Bien-être utilitaire obtenu par notre algorithme d'approximation sélectif comparé aux algorithmes CIS et MIQP

linéaire comme dans un problème IA mais exponentielle. Même si notre procédure ne maximise pas toujours le bien-être utilitaire, nous constatons qu'elle est proche du maximum utilitariste calculé par optimisation mathématique (MIQP) tout en étant (25 fois) plus rapide (pour 20 individus).

De manière similaire, les figures **??** et 6 comparent les temps d'exécution et bien-être égalitaires obtenus par l'algorithme inclusif et par l'algorithme d'approximation de résolution quadratique (MIQP). Le bien-être égalitaire obtenu par notre procédure est proche mais surpasse celui approximé par optimisation mathématique (MIQP) tout en étant (120 fois) plus rapide (pour 20 individus).

Même si nous n'avons pas de résultats théoriques pour la stabilité du résultat de la procédure sélective, la figure 6 montre que 95 % des résultats de la procédure sélective avec approximation sont optimaux au sens de Pareto et 96 % d'entre eux sont individuellement rationnels. Nous nous sommes limité à $n = 2$ et $m \leq 13$ car le problème de décision pour la Pareto-optimalité est n'est pas tractable.

Comme nous avons évalué dans [13] la qualité des solutions atteintes par l'algorithme sélectif et son accélération due à sa distribution, nous nous focalisons ici sur la procédure inclusive. Nous considérons des instances particulières de problème IA. Comme dans notre application pratique, nous considérons que les n activités et les m individus sont attractifs et que ces derniers peuvent tous

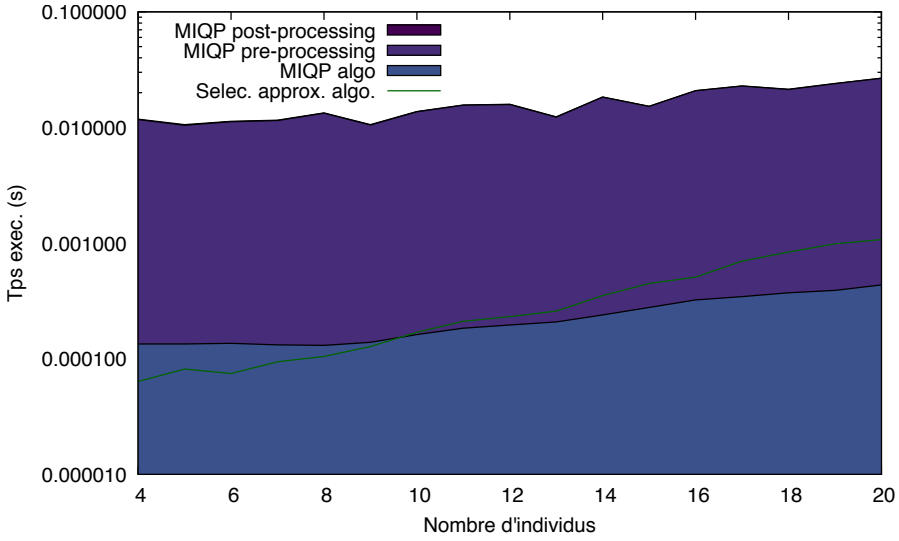

Fig. 9: Temps d'exécution obtenu par notre algorithme d'approximation sélectif comparé à l'algorithme MIQP

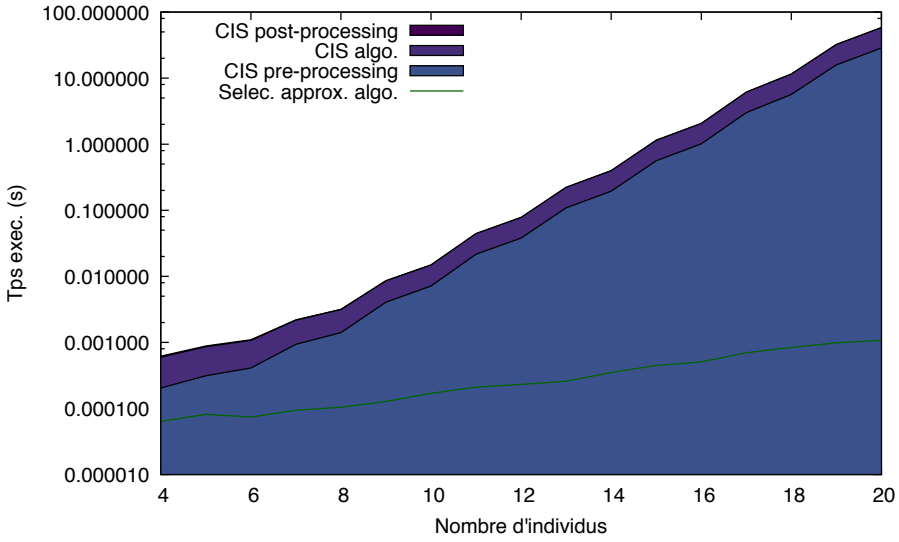

Fig. 10: Temps d'exécution obtenu par notre algorithme d'approximation sélectif comparé à l'algorithme CIS

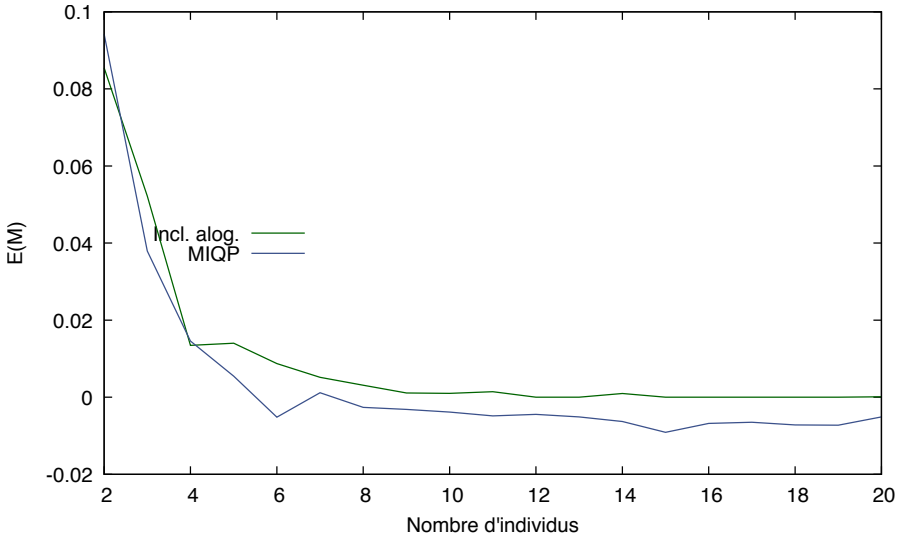

Fig. 11: Bien-être égalitaire obtenu par notre algorithme inclusif comparé à l'algorithme MIQP

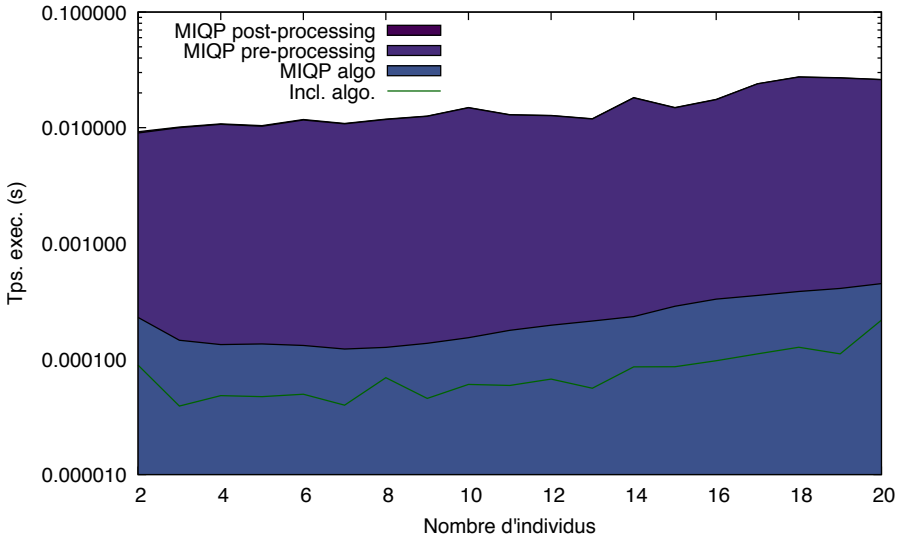

Fig. 12: Temps d'exécution obtenu par notre algorithme inclusif comparé à l'algorithme MIQP

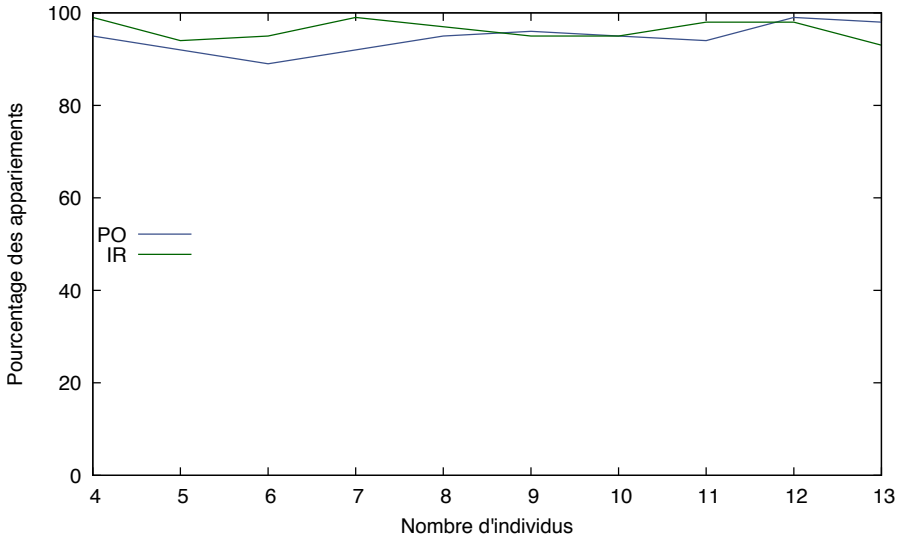

Fig. 13: Pourcentage des appariements calculés par la procédure sélective avec approximation qui sont optimaux au sens de Pareto (PO) et individuellement rationnels (IR)

être affectés :

$$\forall i \in I \ \forall j \in I \setminus \{i\} \ \forall a \in A, \ w_i(j) > 0 \wedge v_i(a) > 0 \tag{32}$$

$$\Sigma_{a \in A} c_a \geq m \tag{33}$$

Par souci de simplicité, toutes les activités ont la même capacité ($c = \lceil m/n \rceil$).

Comme l'algorithme MIQP ne passe pas l'échelle, nous avons implémenté un algorithme de recherche locale [16] pour le comparer à notre procédure d'affectation. Cet algorithme de montée en gradient, qui commence à partir d'un appariement valide et aléatoire (où tous les individus sont affectés à une activité), cherche à améliorer itérativement le bien-être égalitaire. Deux appariements sont voisins s'ils sont identiques à l'exception d'un individu affecté à un activité différente, éventuellement nulle. Si cette dernière est pleine, tous les échanges d'individus avec les participants de cette activité sont considérés.

Tout d'abord, nous comparons le bien-être égalitaire de l'appariement atteint par notre algorithme avec celui atteint par la recherche locale. La figure 6 présente le bien-être égalitaire moyen obtenu pour 100 instances avec chaque jeu de paramètres (tel que $2 \leq n \leq 10$ and $2 \times n \leq m \leq 100$). Notre procédure inclusive surpasse l'algorithme de recherche locale. En effet, le bien-être égalitaire, comme le bien-être utilitaire, est pour le problème IA une fonction qui possède de nombreux optima locaux.

Ensuite, nous comparons le temps d'exécution de l'algorithme inclusif de notre algorithme avec sa version décentralisée. La figure 6 montre les temps

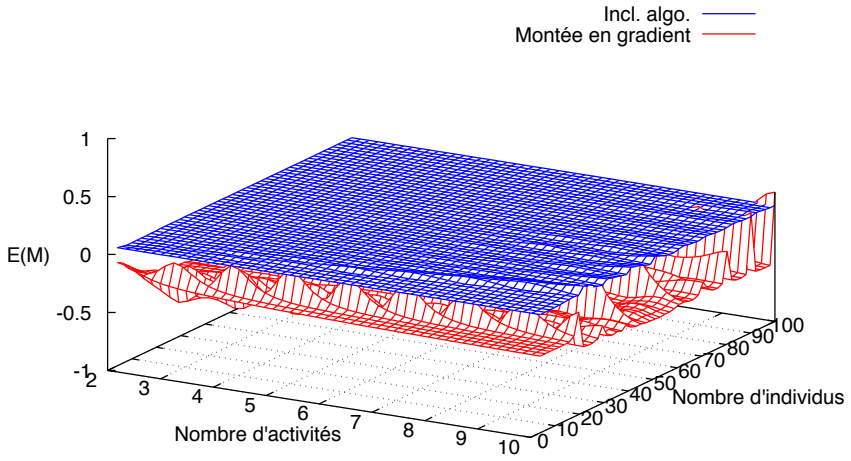

Fig. 14: Bien-être égalitaire moyen pour notre procédure inclusive et pour l'algorithme de recherche locale

d'exécution moyens pour chaque jeu de paramètres (avec $2 \leq n \leq 10$ and $2 \times n \leq m \leq 400$). Bien que l'algorithme centralisé soit plus rapide quand le nombre d'individus est faible (~ 40), son temps d'exécution augmente rapidement avec le nombre d'individus (13 ms pour 100 individus et 10 activités) alors que le temps d'exécution de la version décentralisé est inférieure (9 ms dans ce dernier cas). De plus, le temps d'exécution de la montée en gradient est trop important pour être représenté (2 969 ms pour 100 individus et 10 activités). On peut s'attendre à un temps d'exécution encore plus important si on adopte une méthode de recherche locale de type recuit simulé sans pour autant garantir d'obtenir un optimum global.

En résumé, la solution atteinte par notre procédure inclusive semble proche du maximum égalitariste. De plus, cet algorithme est distribuable, ce qui permet d'accélérer jusqu'à 5 fois son exécution.

7 Application pratique

Notre application pratique concerne un collectif de seniors qui souhaitent entrer en relation afin d'effectuer des activités. L'objectif est ici de maximiser les activités partagées afin d'améliorer la cohésion sociale et lutter contre l'isolement chez certains seniors.

Selon la figure 7, qui représente le modèle conceptuel de données que nous avons adopté pour construire la base de données, chaque utilisateur (`HolidayMaker`) est décrit (`describesHolidayMakers`) par un ensemble de caractéristiques

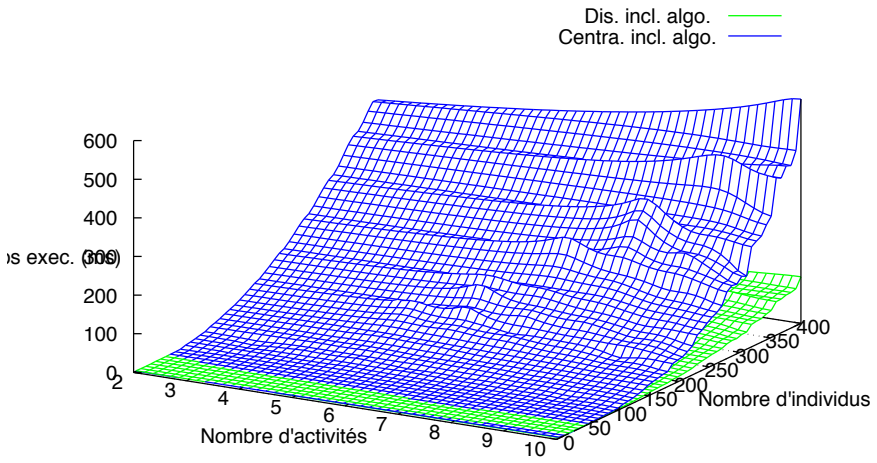

Fig. 15: Temps d'exécution moyen pour notre procédure (décentralisée) inclusive

(`HolidayMakerFeature`) : son sexe, son âge, s'il est fumeur, ses hobbies, etc. Ainsi chaque utilisateur indique (ou non) les valeurs qui le caractérisent (`HolidayakerFeatureValue`) et il peut signaler son attraction voire sa répulsion (`attractionPartner`) vis-à-vis de ces valeurs pour ses co-vacanciers. C'est à partir de ces connaissances que sont agrégées les affinités entre utilisateurs (`computedPreferencePartner`). De manière similaire, les voyages (`Tour`) sont décrits (`describesTour`) par des valeurs (`TourFeatureValue`) de caractéristique (`TourFeature`) sur lesquels les utilisateurs expriment des préfèrences (`likesTourFeature`). C'est à partir de ces connaissances que sont agrégés les intérêts des utilisateurs pour les voyages (`computedPreferenceTour`).

Bien que ce terrain d'expérimentation implique plusieurs milliers d'individus, nous avons préalablement testé notre prototype avec 102 utilisateurs et 10 voyages. La table 7 résume les temps d'exécution et les bien-être atteints par tous les algorithmes, à l'exception des algorithmes par optimisation mathématique trop lents. Que cela soit pour le bien-être utilitaire ou le bien-être égalitaire, nos algorithmes s'avèrent plus efficaces et plus rapides que la montée en gradient. Les temps d'exécution des versions distribuées de nos algorithmes sont plus rapides quand le nombre d'intégration est suffisant.

8 Discussion

Nous avons abordé ici le problème générique où des individus doivent être affectés à des activités en privilégiant les activités qui leur plaisent avec les partenaires qu'ils apprécient. Bien que souhaitable, la stabilité d'un appariement

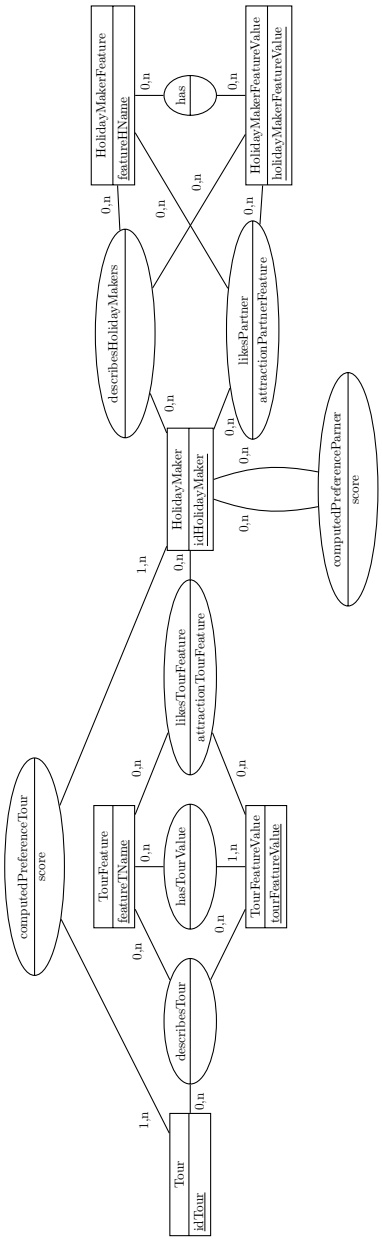

Fig. 16: Modèle conceptuel pour les données réelles collectées et traitées par nos algorithmes d'appariement. Les entités sont représentées par des rectangles et les associations par des rectangles à bord arrondis

Algorithme	Bien-être	Temps d'exécution (ms)
Sélec. avec appr.	$U(M) = 0,154$	263
Dis. sélec. avec appr.	$U(M) = 0,156$	210
Montée en gradient	$U(M) = -0,066$	11 221
Inclusif	$E(M) = 0,0$	22
Dis. inclusif	$E(M) = 0,0$	44
Montée en gradient	$E(M) = -0,433$	20 643

Tab. 1: Bien-être et temps d'exécution des différents algorithmes d'appariement sur les données réelles

n'est pas garantie. À l'inverse, l'optimalité au sens de Pareto semble ne pas être suffisamment discriminante. C'est la raison pour laquelle nous avons évalué la qualité d'un appariement du point de vue de la société d'individus à l'aide des concepts issus de la théorie du choix social. D'autres concepts sont envisageables, qu'ils soient issus de l'économie comme l'absence d'envie justifiée (en anglais, *envy-freeness*), la non-manipulabilité (en anglais, *strategy-proofness*) ou issus de la psychologie sociale comme le concept de cohésion sociale esquissé ici.

Même si une étude de la complexité dans le pire des cas est nécessaire, nos expérimentations, notamment celles concernant des données réelles, montre que les bien-être sociaux possèdent de nombreux optima locaux pour ce problème. C'est pourquoi nous avons proposé deux heuristiques, une pour chacune des notions de bien-être social, qui retournent des appariement valides, selon une règle sociale (utilitaire ou égalitaire) pour la première et socialement cohésifs pour la seconde. Ces procédures s'appuient sur l'aspect organisationnel de la modélisation multi-agents en introduisant explicitement un niveau intermédiaire entre l'individu et la société, i.e. les coalitions d'individus qui partagent un intérêt commun pour une activité. Conformément au principe de subsidiarité, la reconfiguration d'une coalition est soumise à l'arbitrage d'un agent représentant les objectifs et contraintes d'un groupe d'individus.

Comme l'illustrent nos résultats empiriques, les méthodes de résolution multi-agents peuvent être plus efficaces que les méthodes générales pour l'optimisation de problèmes dont la formulation est inadaptée. De plus, au-delà d'une seule distribution qui comme ici accélère son exécution, la modélisation des comportements d'agents permet d'introduire des stratégies variées pour mettre en œuvre les contraintes et les objectifs des différentes coalitions d'agents.

Dans de futurs travaux, nous envisageons de prendre en compte l'aspect relationnel de la modélisation multi-agents pour distribuer les négociations entre accointances dans le réseau social formé.

Remerciements

Ce travail s'inscrit dans le projet PartENS soutenu par le programme chercheur-citoyen de la région Nord Pas de Calais. Nous remercions le comité de pro-

gramme des JFSMA qui, par ses remarques, nous a permis d'améliorer cet article.

Références

[1] H. Aziz, F. Brandt, and H. G. Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence Journal*, 195 :316–334, 2013.

[2] C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1) :1–30, 2004.

[3] C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. D. Procaccia, and O. Sheffet. Optimal social choice functions : A utilitarian view. *Artificial Intelligence Journal*, 227 :190–213, 2015.

[4] I. Brito and P. Meseguer. Distributed stable matching problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 152–166. Springer, 2005.

[5] A. Darmann, J. Döcker, B. Dorn, J. Lang, and S. Schneckenburger. On simplified group activity selection. In *Proceedings of 5th International Conference on Algorithmic Decision Theory*, pages 255–269, Luxembourg, Luxembourg, 2017. Springer International Publishing.

[6] A. Darmann, E. Elkind, S. Kurz, J. Lang, J. Schauer, and G. Woeginger. Group activity selection problem. In *Proceedings of the 8th International Conference on Internet and Network Economics*, pages 156–169, Liverpool, UK, 2012. Springer Berlin Heidelberg.

[7] J. Dreze and J. Greenberg. Hedonic coalitions : Optimality and stability. *Econometrica*, 48 :987–1003, 1980.

[8] P. Everaere, M. Morge, and G. Picard. Casanova : un comportement d'agent respectant la privacité pour des mariages stables et équitables. *Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle*, 26(5) :471–494, 2012.

[9] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69 :9–14, 1962.

[10] C. Hewitt, P. Bishop, and R. Steiger. A universal modular ACTOR formalism for artificial intelligence. In *Proc. of the 3rd International Joint Conference on Artificial Intelligence*, pages 235–245, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc.

[11] A. Igarashi, D. Peters, and E. Elkind. Group activity selection on social networks. In *Proc. of 31th AAAI Conference on Artificial Intelligence*, pages 565–571, San Francisco, California USA, 2017.

[12] D. F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2014.

[13] M. Morge and A. Nongaillard. Affectation distribuée d'individus à des activités avec des préférences additivement séparables. In *Journées Francophones sur les Systèmes Multi-Agents*, pages 19–28, Caen, France, 2017. Cépaudès édition.

[14] H. Moulin. *Fair Division and Collective Welfare*. The MIT Press, 2002.

[15] A. Nongaillard and S. Picault. Modélisation multi-niveaux du bien-être social dans un SMA : application aux problèmes d'affectation et d'appariement. *Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle*, 31(6) :709–734, 2017.

[16] S. Russell and P. Norvig. *Artificial Intelligence : A Modern Approach*, chapter Chapter 4 : Informed Search and Exploration, pages 94–136. Pearson Education, 2003. 2nd edition.

[17] S. Sahni. Computationally related problems. *SIAM Journal on Computing*, 3(4) :262–279, 1974.

[18] T. C. Schelling. *The strategy of conflict*. Harvard university press, 1980.

[19] A. K. Sen. *Collective Choice and Social Welfare*. North-Holland, 1970.

[20] S. J. Zaccaro and C. A. Lowe. Cohesiveness and performance on an additive task : Evidence for multidimensionality. *The Journal of Social Psychology*, 128(4) :547–558, 1988.

A Programmation linéaire

Considérons une instance de problème IA de taille (m, n).

Afin de trouver un appariement maximum utilitariste, nous pouvons considérer la programmation quadratique, i.e. une méthode d'optimisation avec un modèle mathématique représentable par :

1. $n \times m$ variables de décision $x_{ia} \in \{0, 1\}$ telles que,

$$x_{ia} = \begin{cases} 1 & \text{si } i \text{ est affecté à } a, \\ 0 & \text{sinon}\,; \end{cases} \tag{34}$$

2. m contraintes représentant l'exclusion mutuelle de l'affectation d'un même individu aux activités,

$$\Sigma_{a \in A} x_{ia} \leq 1; \tag{35}$$

3. n contraintes garantissant la validité de l'appariement,

$$\Sigma_{i \in I} x_{ia} \leq c_a. \tag{36}$$

En considérant les fonctions d'intérêts et d'affinités, la fonction objectif à maximiser est :

$$\frac{1}{m} \sum_{i \in I} \sum_{a \in A} \left[\frac{1}{2} \Big(x_{ia} v_i(a) + \frac{1}{m-1} \sum_{j \neq i} x_{ja} w_i(j) \Big) \right] \tag{37}$$

Quand on écrit sous une forme standard ces problèmes, i.e. en minimisant une fonction objectif de la forme $\frac{1}{2}xQx^T + c^T x$ avec une matrice symétrique Q, on peut remarquer que cette dernière n'est pas nécessairement définie positive et donc le problème peut être NP-difficile [17].

Afin de trouver un appariement maximum égalitariste, nous pouvons considérer le programme linéaire suivant :

1. $n \times m$ variables de décision $x_{ia} \in \{0, 1\}$ telles que,

$$x_{ia} = \begin{cases} 1 & \text{si } i \text{ est affecté à } a, \\ 0 & \text{sinon}\,; \end{cases} \tag{38}$$

2. une variable de décision $uMin \in [-1; 1]$ correspondant à l'utilité minimale d'un individu ;
3. m contraintes représentant l'exclusion mutuelle de l'affectation d'un même individu aux activités,

$$\Sigma_{a \in A} x_{ia} \leq 1; \tag{39}$$

4. n contraintes garantissant la validité de l'appariement,

$$\Sigma_{i \in I} x_{ia} \leq c_a. \tag{40}$$

5. m contraintes correspondant à l'utilité minimale qui s'appuient sur les fonctions d'intérêts et d'affinités :

$$\sum_{a \in A} \left[\frac{1}{2} \Big(x_{ia} v_i(a) + \frac{1}{m-1} \sum_{j \neq i} x_{ja} w_i(j) \Big) \right] \geq uMin \tag{41}$$

La fonction objectif à minimiser est $uMin$. Ce programme permet de mettre en œuvre un algorithme d'approximation qui calcule une solution approchée.