

# Rapport de Projet MOSIMA

## Confiance et Réputation

Alexandre Bontems – Gualtiero Mottola

### Table des matières

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Étude préliminaire</b>                           | <b>1</b>  |
| <b>2</b> | <b>Étude progressive du modèle FIRE</b>             | <b>2</b>  |
| 2.1      | Modèle avec composante IT seule (FIRE-IT)           | 2         |
| 2.2      | Modèle avec composante IT et WR (FIRE-IT-WR)        | 5         |
| 2.3      | Modèle avec composante IT, WR et CR (FIRE-IT-WR-CR) | 7         |
| 2.4      | Analyses de sensibilité du modèle FIRE-IT-WR-CR     | 9         |
| 2.5      | Extensions  | 11        |
|          | <b>Références</b>                                   | <b>13</b> |

## 1 Étude préliminaire

**1.1** Les auteurs de [1] cherchent à construire un modèle de confiance et de réputation dans le cadre de systèmes multi-agents *ouverts*, c'est-à-dire des environnements que les agents peuvent joindre et quitter à tout moment. Les agents considérés ont alors des appartenances et des buts différents, et interagissent entre eux pour satisfaire leurs buts respectifs. Dans de telles conditions il est difficile pour un individu d'évaluer la fiabilité d'un interlocuteur. Le modèle FIRE propose donc quatre mesures distinctes permettant à tout agent une telle évaluation.

- **Interaction Trust (IT)** : confiance dérivée des relations directes entre agents. À chaque interaction, les performances sont enregistrées et peuvent ensuite être agrégées en donnant plus de poids aux interactions récentes.
- **Witness Reputation (WR)** : un agent cherchant à évaluer un autre est capable d'interroger les acteurs ayant déjà interagi avec lui.
- **Certified Reputation (CR)** : chaque agent est capable de pourvoir des certificats de performance délivrés par les agents avec lesquels il a interagi.
- **Role-based Trust (RT)** : confiance définie par la relation entre les agents. Par exemple pour des agents appartenant au gouvernement.

L'agrégation de ces mesures permet d'évaluer la confiance que l'on peut accorder à un individu, c'est-à-dire à quel point il est susceptible de fournir un bon service/bien. Ainsi, lorsqu'un agent cherche à utiliser un service il sera capable de choisir le plus prometteur selon cette mesure. Le modèle prévoit également de donner des importances différentes à chacune des composantes et de prendre en compte la fiabilité des informations.

**1.2** Les quatre composantes produisent un modèle robuste grâce à la pluralité des sources d'information considérées. Si l'une fait défaut (e.g. on a peu d'interactions enregistrées lorsqu'un agent arrive dans l'environnement pour la première fois), alors les autres pourront compléter. Par ailleurs, les auteurs cherchent à produire un modèle fidèle aux systèmes existants et les mesures proposées répliquent des comportements observés dans de tels systèmes.

## 2 Étude progressive du modèle FIRE

### 2.1 Modèle avec composante IT seule (FIRE-IT)

**2.1.1** Dans cette première version du modèle, l'initialisation des agents et leurs interactions sont implémentées. Comme dans l'évaluation expérimentale détaillée dans [1], deux types d'agents sont considérés : consommateur et fournisseur. Ces derniers donnent accès à un unique type de service que les consommateurs vont chercher à utiliser selon un niveau d'activité  $\alpha$  propre à chacun.

À chaque itération de la simulation, la procédure **UpdateConsumer** (algorithme 1) est donc exécutée pour tous les consommateurs. Ils possèdent un rayon d'action  $r_0$ , commun à tous, et c'est les fournisseurs dont la distance au consommateur est comprise dans ce rayon qui seront considérés pour interaction. La procédure construit donc ce voisinage et lance une interaction. Il est important de noter que les agents sont placés sur une sphère mais que nous calculons la distance euclidienne entre deux agents et non la distance sur la sphère.

---

**Algorithme 1** Mise à jour des consommateurs.

---

**Input :**  $c$  consommateur,  $\mathcal{F}$  liste des fournisseurs  
**procedure** UPDATECONSUMER( $c, \mathcal{F}$ )  
     $r \leftarrow$  réel aléatoire entre 0 et 1  
    **if**  $r \leq \alpha$  **then**  
         $\mathcal{F}_v \leftarrow \{f \mid f \in \mathcal{F} \text{ et } \text{dist}(f, c) \leq r_0\}$   
        INTERACT( $c, \mathcal{F}_v$ )  
    **end if**  
**end procedure**

---

La procédure **Interact** (algorithme 2) décrit le processus d'interaction, dépendant d'un paramètre de température  $T$  décroissant avec le temps. En effet, cette version du modèle utilise uniquement la composante IT dans laquelle un historique des interactions passées est gardé en mémoire et est utilisé pour calculer une valeur de confiance pour chaque fournisseur. Par conséquent, certains fournisseurs ne pourront pas être évalués en début de simulation. La procédure a pour but d'évaluer les fournisseurs qui peuvent l'être et de distinguer ceux qui ne le peuvent pas. La fonction **ChooseProvider** (algorithme 3) est ensuite utilisée pour choisir le fournisseur en privilégiant les fournisseurs non-évalués lorsque  $T$  est grand ou en choisissant celui avec la meilleure évaluation lorsque  $T$  est plus petit. Cette température est décrémentée à chaque interaction.

---

**Algorithme 2** Interaction entre consommateur et fournisseur.

---

**Input :**  $c$  consommateur,  $\mathcal{F}_v$  liste des fournisseurs voisins de  $c$ .  
**procedure** INTERACT( $c, \mathcal{F}_v$ )  
    **if**  $\mathcal{F}_v = \emptyset$  **then**  
        **stop** ▷ Pas d'interaction si aucun fournisseur à proximité.  
    **end if**  
     $NoTrustValues \leftarrow \{f \mid f \in \mathcal{F}_v \text{ et } \text{GETLOCALRATINGS}(c, f) = \emptyset\}$   
     $bestTrust \leftarrow -\infty$   
    **for all**  $f \in \mathcal{F}_v \setminus NoTrustValues$  **do**  
         $\mathcal{R}_l \leftarrow \text{GETLOCALRATINGS}(c, f)$  ▷ Évaluations de  $c$  pour  $f$ .  
         $t \leftarrow \text{GETTRUST}(\mathcal{R}_l)$  ▷ Confiance accordée à  $c$ .  
        **if**  $t > bestTrust$  **then**  
             $bestProvider \leftarrow f$   
             $bestTrust \leftarrow t$   
        **end if**  
    **end for**  
     $chosenProvider \leftarrow \text{CHOOSEPROVIDER}(c, NoTrustValues, bestProvider)$

---

---

```

     $p \leftarrow \text{PERFORM}(\text{chosenProvider})$ 
     $\text{history} \leftarrow \text{history} \cup \{(\text{chosenProvider}, p, \text{time})\}$ 
     $T \leftarrow \max(1, T - 1)$ 
end procedure

```

---

Puisque seule la composante IT est incluse, la valeur de confiance donnée par la fonction **GetTrust** correspond ici à  $\mathcal{T}_I$ , c'est-à-dire une valeur comprise dans  $[-1, 1]$  calculée à partir de l'historique du consommateur. On ne prend pas en compte les poids  $W_K$  ni la notion de fiabilité  $\rho_K$  pour la même raison.

La fonction **ChooseProvider** choisit à partir de cette valeur le fournisseur avec lequel interagir. Deux actions sont possibles : choisir le meilleur agent selon IT ( $a_1$ ) ou choisir un agent aléatoirement parmi les non-évalués ( $a_2$ ). La probabilité de choisir l'une des actions est définie par

$$P(a_k) = \frac{e^{\frac{ER(a_k)}{T}}}{\sum_{a_i} e^{\frac{ER(a_i)}{T}}}$$

où  $ER(a_1)$  est la meilleure valeur de confiance parmi les fournisseurs évalués et  $ER(a_2)$  est la moyenne d'utilité enregistrée par le consommateur au cours de sa vie. Grâce au paramètre  $T$ , les agents vont plutôt choisir des fournisseurs non-évalués en début de simulation puis choisir le meilleur selon IT lorsqu'ils ont « exploré » leur environnement.

---

**Algorithme 3** Choix du fournisseur.

---

```

Input :  $c$  consommateur,  $NoTrustValues$  liste des fournisseurs sans évaluation,
 $bestProvider$  fournisseur avec la meilleur évaluation.
function CHOOSEPROVIDER( $c, NoTrustValues, bestProvider$ )
    if  $bestProvider = \emptyset$  then                                 $\triangleright$  Aucun fournisseur voisin n'a d'évaluation.
        return fournisseur aléatoire parmi  $NoTrustValues$ 
    end if
    if  $|NoTrustValues| = 0$  then                                 $\triangleright$  Tous les fournisseurs sont évalués.
        return  $bestProvider$ 
    end if
    Choisir  $a_k$  avec probabilité  $P(a_k)$ 
    if  $a_1$  then
        return  $bestProvider$ 
    else
        return fournisseur aléatoire parmi  $NoTrustValues$ 
    end if
end function

```

---

**2.1.2** Les agents sont initialisés selon les mêmes valeurs de paramètres que dans l'évaluation expérimentale des auteurs. Deux populations d'agents (consommateurs et fournisseurs) sont placés aléatoirement sur une sphère de rayon  $s$ . Parmi les fournisseurs on compte  $N_P$  agents de plusieurs types :

- $N_{PG}$  fournisseurs de type GOOD
- $N_{PO}$  fournisseurs de type OK
- $N_{PI}$  fournisseurs de type INTERMITTENT
- $N_{PB}$  fournisseurs de type BAD

Chacun fournira un service dont l'utilité dépend de son type. En effet, les fournisseurs GOOD, OK et BAD sont tous créés avec un profil différent, défini par une moyenne de performance  $\mu_P$  et un écart type  $\sigma_P$ . La moyenne est choisie uniformément aléatoirement dans un intervalle en fonction du type. Les tables 1a et 1b détaillent les valeurs utilisées. L'utilité gagnée par

un consommateur utilisant le service d'un de ces agents est alors tirée selon une loi normale  $\mathcal{N}(\mu_P, \sigma_P)$ . Les fournisseurs **INTERMITTENT** quant à eux produisent une utilité tirée uniformément aléatoirement dans l'intervalle  $[\text{PL\_BAD}, \text{PL\_GOOD}]$ .

| Niveau de performance | Utilité |
|-----------------------|---------|
| PL_PERFECT            | 10      |
| PL_GOOD               | 5       |
| PL_OK                 | 0       |
| PL_BAD                | -5      |
| PL_WORST              | -10     |

(a) Constantes de niveau de performance

| Profil | Intervalle de $\mu_P$                   | $\sigma_P$ |
|--------|---|------------|
| GOOD   | $[\text{PL\_GOOD}, \text{PL\_PERFECT}]$ | 1.0        |
| OK     | $[\text{PL\_OK}, \text{PL\_GOOD}]$      | 2.0        |
| BAD    | $[\text{PL\_WORST}, \text{PL\_OK}]$     | 2.0        |

(b) Profil des fournisseurs

TABLE 1 – Paramètres de profil des fournisseurs

En ce qui concerne les consommateurs, un nombre  $N_C$  sont évalués, tous utilisant FIRE-IT pour calculer la valeur de confiance d'un fournisseur. Toujours en suivant la méthodologie expérimentale, la taille de leur historique est limité à  $H$  et à chaque interaction enregistrée on associe un poids dépendant de la différence de temps entre l'interaction et le calcul de confiance. Le paramètre  $\lambda$  est utilisé pour le calcul de ce poids. Enfin le niveau d'activité des consommateurs est choisis uniformément aléatoirement dans l'intervalle  $[0.25, 1]$ . Les valeurs fixées pour ces paramètres sont visibles en 2.

| Variable                                   | Symbole    | Valeur          |
|--|------------|-----------------|
| Rayon de la sphère monde                   | $s$        | 22              |
| Rayon d'action des consommateurs           | $r_0$      | 13              |
| Température initiale                       | $T_{init}$ | 100             |
| Nombre d'itérations                        | $N$        | 200             |
| Nombre total de fournisseurs               | $N_P$      | 100             |
| · GOOD                                     | $N_{PG}$   | 10              |
| · OK                                       | $N_{PO}$   | 40              |
| · INTERMITTENT                             | $N_{PI}$   | 5               |
| · BAD                                      | $N_{PB}$   | 45              |
| Nombre de consommateurs dans chaque groupe | $N_C$      | 500             |
| Intervalle du niveau d'activité            | $\alpha$   | $[0.25, 1.00]$  |
| Taille de l'historique des agents          | $H$        | 10              |
| Facteur d'actualisation IT                 | $\lambda$  | $-(5/\ln(0.5))$ |

TABLE 2 – Paramètres utilisés pour produire la figure 1

Le seul paramètre dont la valeur n'est pas spécifiée dans l'article est le rayon d'action des agents consommateurs. Ici on fait l'hypothèse qu'il est le même pour tous et on fixe sa valeur à 13 pour reproduire les résultats des auteurs.

**2.1.3** La figure 1 reproduit la figure 9 de [1] et on y voit bien que l'utilité gagnée par les consommateurs croît avec le nombre d'interactions. Au début de la simulation, l'historique est vide et les fournisseurs sont donc choisis aléatoirement parmi les voisins. L'utilité gagnée reste dans l'intervalle  $[0.0, 1.0]$ . Plus la simulation progresse cependant, plus les agents récupèrent des informations sur leur environnement et sont capables de choisir le fournisseur pour lequel ils éprouvent le plus de confiance. Arrivé au terme des  $N$  itérations, on constate donc une croissance soutenue de l'utilité moyenne chez les consommateurs. On note que le plus gros de la croissance est enregistré dans les vingt premières itérations ; les agents sont capables d'identifier très vite les meilleurs fournisseurs de leur voisinage. Puisque les profils des fournisseurs sont tirés aléatoirement, la valeur moyenne maximale constaté peut varier entre 3 et 4 selon les simulations.

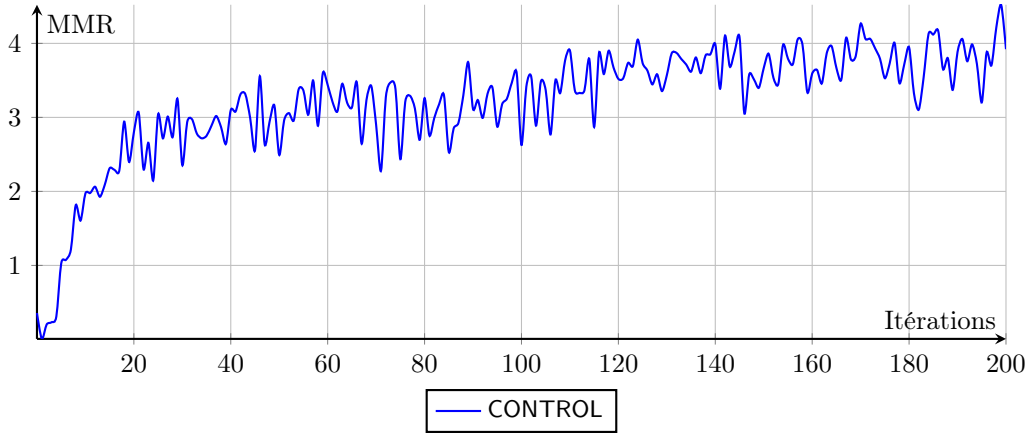


FIGURE 1 – Utilité moyenne des consommateurs avec FIRE-IT

## 2.2 Modèle avec composante IT et WR (FIRE-IT-WR)

**2.2.1** Nous présentons ici la méthode de sélection des fournisseurs lorsque la composante WR est activée. Les consommateurs doivent récupérer la liste des notes de témoins qui leur permettra ensuite de calculer la valeur de confiance selon cette composante. Nous avons choisi pour cela d'implémenter une fonction récursive `GetAllWitnessRatings` (algorithme 4). Elle est appelée par le consommateur qui cherche à évaluer un fournisseur ainsi que par certains de ses voisins lorsqu'ils ne peuvent pas fournir une liste de notes sur le fournisseur. On note que cette fonction utilise deux paramètres  $n_{RL}$  et  $n_{BF}$  qui définissent respectivement la profondeur et la largeur maximale de l'arbre de récursion.

---

**Algorithme 4** Récupération des notes de témoins.

---

**Input :**  $c$  consommateur courant,  $prevCons$  liste des consommateurs précédents,  $prov$  fournisseur évalué,  $currentDepth$  profondeur courante.

**function** `GETALLWITNESSRATINGS`( $c, prevCons, prov, currentDepth$ )

**if**  $currentDepth = n_{RL}$  **then** ▷ Profondeur maximale de l'arbre atteinte.

**return**  $\emptyset$

**end if**

$currentDepth \leftarrow currentDepth + 1$

$prevCons \leftarrow prevCons \cup \{c\}$

$closestCons \leftarrow \text{GETCLOSESTCONSUMERS}(c, prevCons)$

$returnList \leftarrow \emptyset$

**for all**  $x \in closestCons$  **do**

$ratings \leftarrow \text{ASKRATINGS}(x, prov)$

**if**  $ratings = \emptyset$  **then** ▷ Pas de notes correspondant à ce fournisseur.

$ratings \leftarrow \text{GETALLWITNESSRATINGS}(x, prevcons, prov, currentDepth)$

**end if**

$returnList \leftarrow returnList \cup ratings$

**end for**

**return**  $returnList$

**end function**

---

Cette fonction récursive fait usage de deux fonctions auxiliaires `AskRatings` (algorithme 5) et `GetClosestConsumers` (algorithme 6). La première permet de récupérer les notes d'un consommateur pour un fournisseur. La fonction `GetAllWitnessRatings` peut ainsi savoir si un consommateur a des informations utiles sur le fournisseur qui est évalué. La fonction retourne une liste vide si le consommateur ne connaît rien du fournisseur. La fonction `GetClosestConsumers` renvoie une liste des consommateurs voisins c'est-à-dire situés dans le rayon  $r_0$  de l'appelant. Elle ne renvoie cependant qu'au plus  $n_{BF}$  voisins.

---

**Algorithme 5** Demande des notes à un témoin.

---

**Input :**  $c$  consommateur courant,  $prov$  fournisseur évalué  
**function** ASKRATINGS( $c, prov$ )  
     $matched \leftarrow \emptyset$   
    **for all**  $(f, n, t) \in history$  **do**  
        **if**  $f = prov$  **then** ▷ La note porte sur le fournisseur que l'on recherche.  
             $matched \leftarrow matched \cup \{x\}$   
        **end if**  
    **end for**  
    **return**  $matched$   
**end function**

---

---

**Algorithme 6** Récupération des consommateurs les plus proches.

---

**Input :**  $c$  consommateur courant,  $prevCons$  liste des consommateurs précédents,  $\mathcal{C}$  liste des consommateurs  
**function** GETCLOSESTCONSUMERS( $c, prevcons$ )  
     $closestCons \leftarrow \emptyset$   
    **for all**  $cons \in \mathcal{C}$  et  $cons \neq c$  et  $cons \notin prevCons$  **do**  
        **if**  $dist(c, cons) < r_0$  **then** ▷ Ce consommateur est visible pour moi.  
             $closestCons \leftarrow closestCons \cup \{cons\}$   
        **end if**  
    **end for**  
     $nbc \leftarrow 0$   
    **while**  $closestCons \neq \emptyset$  et  $nbc < n_{BF}$  **do**  
         $tmpC \leftarrow \operatorname{argmin}_{x \in closestCons} dist(c, x)$  ▷ Plus proche voisin.  
         $returnConsumers \leftarrow returnConsumers \cup \{tmpC\}$   
         $closestCons \leftarrow closestCons \setminus \{tmpC\}$   
         $nbc \leftarrow nbc + 1$   
    **end while**  
    **return**  $closestCons$   
**end function**

---

Après avoir récupéré les notes des voisins sur un fournisseur donné, on peut directement incorporer le nouveau calcul de confiance dans la procédure **Interact**. En effet, la fonction **GetTrust** prend maintenant en argument les notes locales et les notes de voisins pour produire une valeur de confiance à partir des mesures  $\mathcal{T}_I$  et  $\mathcal{T}_W$ . La formule d'agrégation suivante est utilisée comme spécifiée dans l'article.

$$\mathcal{T}(c, f) = \frac{\sum_{K \in \{I, W\}} \omega_K \cdot \mathcal{T}_K(c, f)}{\sum_{K \in \{I, W\}} \omega_K}$$

Avec  $\omega_K = W_K \cdot \rho_K(c, f)$ . La fiabilité  $\rho_K$  des  $\mathcal{T}_K$  entre donc en jeu et on donne des poids différents aux composantes IT et WR. Comme dans IT, chaque note de témoin se voit attribuer un poids dépendant de son ancienneté.

**2.2.2** Le protocole pour générer la figure est identique à celui utilisé pour la partie 2.1.2. On ajoute la courbe ayant WR pour légende qui montre l'utilité moyenne des consommateurs utilisant IT et WR. On introduit également plusieurs nouveaux paramètres dans la simulation (table 3). Les valeurs de  $\gamma_K$  sont utilisées dans le calcul des fonctions  $\rho_K$ .

**2.2.3** On observe sur la figure 2, que d'une manière générale les agents utilisant WR ont une utilité bien supérieure aux agents utilisant uniquement IT. Cet écart est fortement marqué dans les premières itérations de la simulation : les agents WR atteignent une utilité proche de leur utilité maximale rapidement (une dizaine d'itérations). On peut également noter que plus

| Variable   | Symbole    | Valeur      |
|--|------------|-------------|
| Largeur maximale de récursion                                    | $n_{BF}$   | 2           |
| Profondeur maximale de récursion                                 | $n_{RL}$   | 5           |
| <i>Coefficients des composantes</i>                              |            |             |
| · Interaction Trust  | $W_I$      | 2.0         |
| · Witness Reputation   | $W_W$      | 1.0         |
| <i>Paramètres des fonctions de fiabilité (<math>\rho</math>)</i> |            |             |
| · Interaction Trust  | $\gamma_I$ | $-\ln(0.5)$ |
| · Witness Reputation   | $\gamma_W$ | $-\ln(0.5)$ |

TABLE 3 – Paramètres utilisés pour produire la figure 2

le nombre d'itérations est grand, plus l'écart entre WR et les agents de contrôle se resserre. Cet écart s'explique par le fait que les agents WR exploitent une base de données bien plus grande pour évaluer les fournisseurs. Ils peuvent utiliser les notes de leurs voisins dès le début de la simulation (ce qui explique la croissance très rapide) et obtiennent une connaissance de leur environnement plus complète que les agents IT.

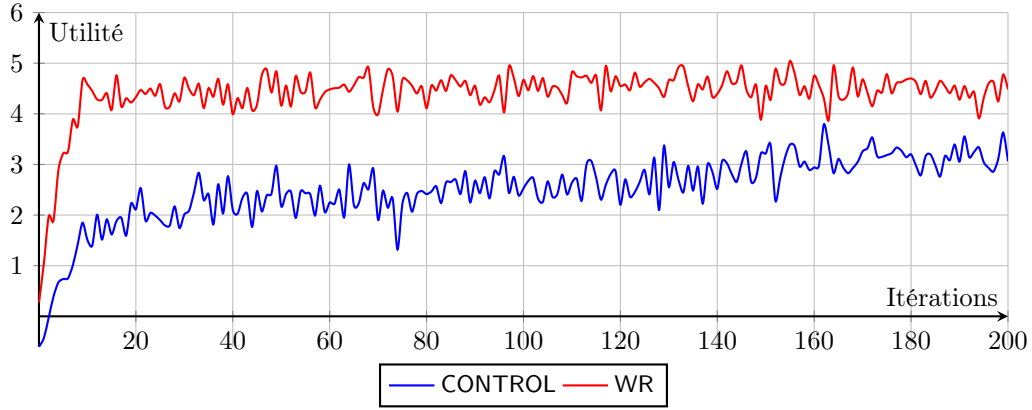


FIGURE 2 – Utilité moyenne des consommateurs avec FIRE-IT comparé a FIRE-IT-WR

## 2.3 Modèle avec composante IT, WR et CR (FIRE-IT-WR-CR)

**2.3.1** Pour pouvoir implémenter la composante CR, les agents fournisseurs se voient dorénavant dotés d'un historique de leurs précédentes transactions. À chaque interaction avec un consommateur, ils sont alors libres d'enregistrer leur performance et de la présenter ensuite aux prochains consommateurs. Dans notre implémentation, les fournisseurs choisissent de sauvegarder leurs  $H$  dernières meilleures performances. L'algorithme 7 détaille cette procédure.

Une fois ces notes enregistrées, si un consommateur souhaite évaluer un fournisseur il pourra le contacter et obtenir cet historique. La confiance du consommateur envers le fournisseur se calcule de la même façon que dans les parties précédentes. On introduit le facteur  $W_C$  qui représente le poids de CR dans le calcul de la valeur de confiance et le paramètre  $\gamma_C$  pour le calcul de fiabilité.

---

### Algorithme 7 Performance des fournisseurs dans CR.

---

**Input :**  $f$  fournisseur courant  
**function** PERFORM( $f$ )  
 $p \leftarrow \text{PERFORMANCE}(f)$   
 $historyF \leftarrow \text{GETHISTORY}(f)$   
**if** SIZE(HISTORYF) <  $H$  **then**  
 $historyF \leftarrow historyF \cup \{p\}$

```

else
     $\minHistory \leftarrow \min_{(f,p',t) \in history_H} p'$ 
    if  $p > \minHistory$  then  $\triangleright$  La nouvelle valeur est meilleure que la pire performance
    enregistrée.
         $historyF \leftarrow historyF \setminus \{\minHistory\}$ 
         $historyF \leftarrow historyF \cup \{p\}$ 
    end if
end if
return  $p$ 
end function

```

**2.3.2** Pour répliquer les figures 10 et 11 de l'article, aucun changement de protocole n'a été nécessaire par rapport aux parties précédentes. On introduit simplement les paramètres manquant selon les valeurs de la table 4. Tous les paramètres précédents sont inchangés.

| Variable   | Symbole    | Valeur      |
|--|------------|-------------|
| <i>Coefficients des composantes</i>                              |            |             |
| · Certified Reputation   | $W_C$      | 0.5         |
| <i>Paramètres des fonctions de fiabilité (<math>\rho</math>)</i> |            |             |
| · Certified Reputation   | $\gamma_C$ | $-\ln(0.5)$ |

TABLE 4 – Paramètres utilisés pour produire la figure 3

**2.3.3** La figure 3 réplique la figure 10 dans laquelle on retrouve une courbe CONTROL pour les agents utilisant IT et une courbe WR pour les agents utilisant cette composante en plus de IT. La courbe CONTROL reste inchangée mais on observe une claire supériorité du modèle FIRE-IT-WR avec un écart allant jusqu'à trois points d'utilité dès les premières itérations. Ce modèle converge également beaucoup plus rapidement que IT et WR vers son utilité maximum. Enfin il est beaucoup moins demandant que WR puisqu'il n'implique aucune récursion et montre une utilité moyenne supérieure à tous les modèles étudiés jusqu'à présent.

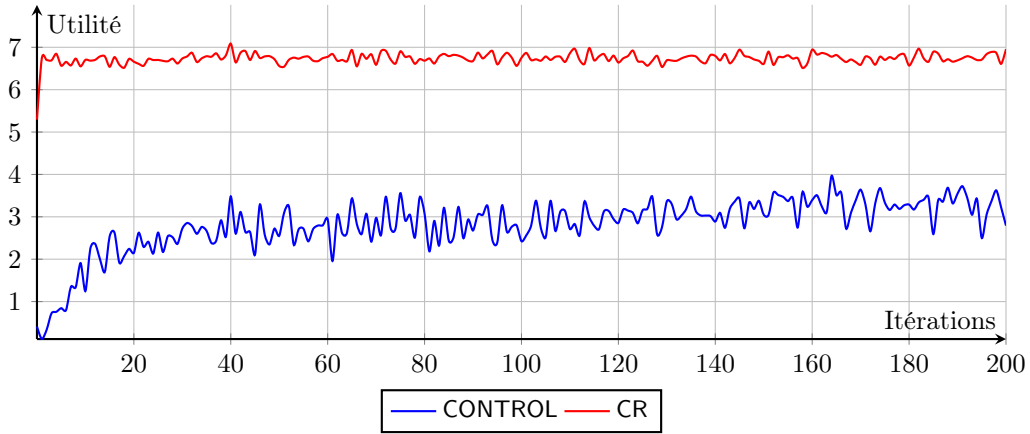


FIGURE 3 – Utilité moyenne des consommateurs avec FIRE-IT comparé a FIRE-IT-CR

La figure 4, quant à elle, montre la comparaison des modèles FIRE-IT-WR-CR et FIRE-IT-WR. On y voit que l'inclusion de la composante CR impacte de beaucoup l'utilité moyenne gagnée par les consommateurs. Si on compare avec la figure 3, on se rend aussi compte que l'inclusion de WR n'apporte pas un grand gain d'utilité. Puisque c'est une composante qui demande plus de temps de calcul que les autres, on peut se demander si elle est bien nécessaire dans le modèle final.



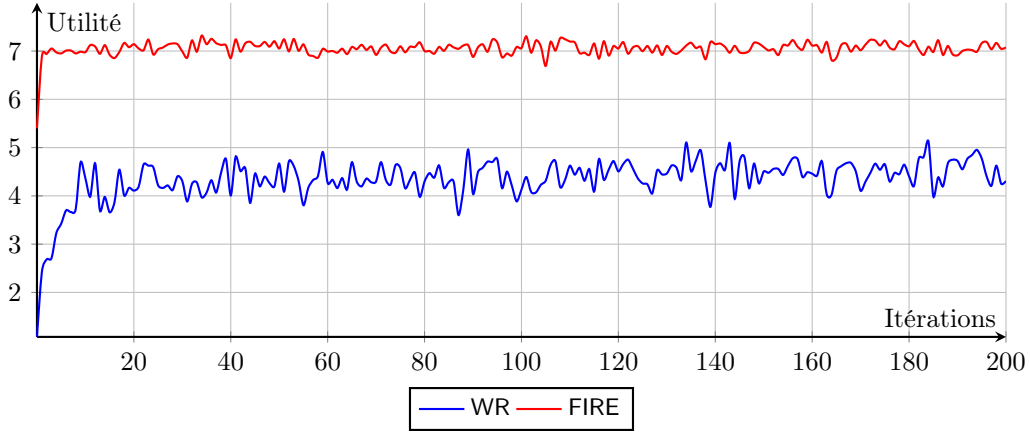


FIGURE 4 – Utilité moyenne des consommateurs avec FIRE comparé a FIRE-IT-WR

**2.3.4** La figure 5 montre l'évolution des mesures  $\rho_K$  au sein des agents consommateurs. On se rend compte que la fiabilité des composantes IT et WR est jugée plutôt basse selon cette mesure et ce tout au long de la simulation. En revanche, la fiabilité de la composante CR reste plutôt bonne avec une certaine croissance en début de simulation. Puisque les agents ont un historique de taille réduite, les composantes IT et WR sont sujettes à une certaine variabilité des performances enregistrées. Par construction la mesure  $\rho_K$  reste donc faible. Mais en ce qui concerne la composante CR, les agents fournisseurs n'enregistrent que leurs meilleures performances et l'historique fourni au consommateur est moins variable.

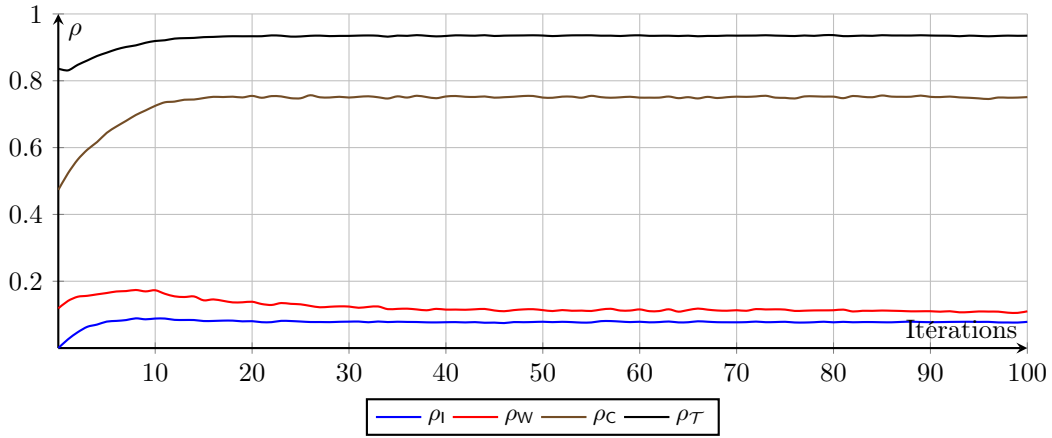


FIGURE 5 – Évolution des mesures  $\rho$

Lors du calcul de la valeur de confiance finale, chaque composante est incluse avec un poids  $\omega_K = \rho_K \cdot W_K$ . Si les poids  $W_I$  et  $W_W$  sont effectivement plus élevés que  $W_C$ , les valeurs de fiabilité donne une grande importance à la composante CR dans la valeur finale de confiance accordée aux fournisseurs. Les fournisseurs exhibant les plus grandes valeurs de performance sont ainsi capables de se démarquer, ce qui explique en partie les performances du modèle FIRE-IT-CR.

## 2.4 Analyses de sensibilité du modèle FIRE-IT-WR-CR

**2.4.1** Le premier paramètre ajouté au modèle est le rayon d'action  $r_0$  des agents consommateurs. Il était fixé à 13 sur toutes les simulations précédentes et on se propose maintenant d'étudier la sensibilité du modèle à ce paramètre. On prend donc 10 valeurs de 5 à 15 et on en-

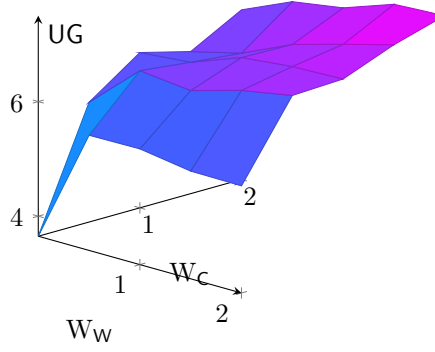


FIGURE 6 – Influence des poids des composantes WR et CR

registre les moyennes d'utilité des consommateurs sur 5 simulations de 50 itérations du modèle FIRE-IT-WR-CR. Puisque les modèles convergent rapidement vers une utilité moyenne stable, on estime que 50 itérations sont suffisantes. Tous les autres paramètres gardent les valeurs fixées jusqu'à présent. La table suivante présente les résultats.

| $r_0$          | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
|----------------|------|------|------|------|------|------|------|------|------|------|------|
| <b>Utilité</b> | 1.30 | 1.62 | 3.24 | 3.92 | 4.56 | 5.56 | 6.22 | 6.44 | 7.25 | 7.32 | 7.72 |

On remarque donc sans surprise que le rayon d'action influe grandement sur l'utilité que peuvent gagner les agents. En effet, plus le rayon est grand plus un consommateur aura de chance de connaître un bon fournisseur dans son entourage et de l'identifier.

Le second paramètre ajouté au modèle est la température initiale  $T_{init}$  fixée à 100 pour tous les tests précédents. On étudie son influence sur le modèle pour des valeurs de 40 à 140 en incrémentant de 10 la valeur entre chaque test. Chaque valeur est une moyenne sur dix simulations de 50 itérations. Tous les autres paramètres gardent les valeurs fixées jusqu'à présent. La table suivante présente les résultats.

| $T_{init}$     | 40   | 50   | 60   | 70   | 80   | 90   | 100  | 110  | 120  | 130  | 140  |
|----------------|------|------|------|------|------|------|------|------|------|------|------|
| <b>Utilité</b> | 4.75 | 5.32 | 5.42 | 4.98 | 4.59 | 5.01 | 4.90 | 4.98 | 4.82 | 5.01 | 4.99 |

Il semble que la température initiale n'ait que peu d'influence sur la valeur de l'utilité pour le modèle FIRE. On ne remarque aucune corrélation entre la valeur étudiée et l'utilité de nos agents.

**2.4.2** La figure 6 présente l'évolution de l'utilité moyenne du modèle en fonction des poids  $W_W$  et  $W_C$ . Ces valeurs sont obtenues sur 5 simulations de 50 itérations (avec 250 agents seulement pour réduire les temps de calcul). Les deux paramètres prennent les valeurs 0, 0.5, 1, 1.5 et 2. On voit donc bien que plus on prend en compte les deux composantes, plus l'utilité moyenne est élevée. Les deux composantes combinées avec un poids de 2 chacune donnent la meilleure performance. En revanche, on peut également voir que l'utilisation de l'une ou l'autre des composantes seule donne une utilité moyenne très proche. Étant donné les temps de calcul nécessaires pour WR on peut se demander ici aussi si son inclusion est bien nécessaire au modèle.

**2.4.3** La figure 7 montre la distribution des valeurs UG des consommateurs et nous permet de déterminer quelles sont les composantes ayant la meilleure distribution de valeurs, c'est-à-dire quelles sont les composantes les plus efficaces pour sélectionner les fournisseurs. D'une manière générale toutes les composantes se concentrent surtout entre 0 et 8 avec les plus grandes concentrations entre 3 et 7. Nous remarquons que les modèles FIRE, FIRE-IT-CR et FIRE-IT-WR semblent être plus efficaces que FIRE-IT avec des pics de valeurs centrés en 5. Cependant

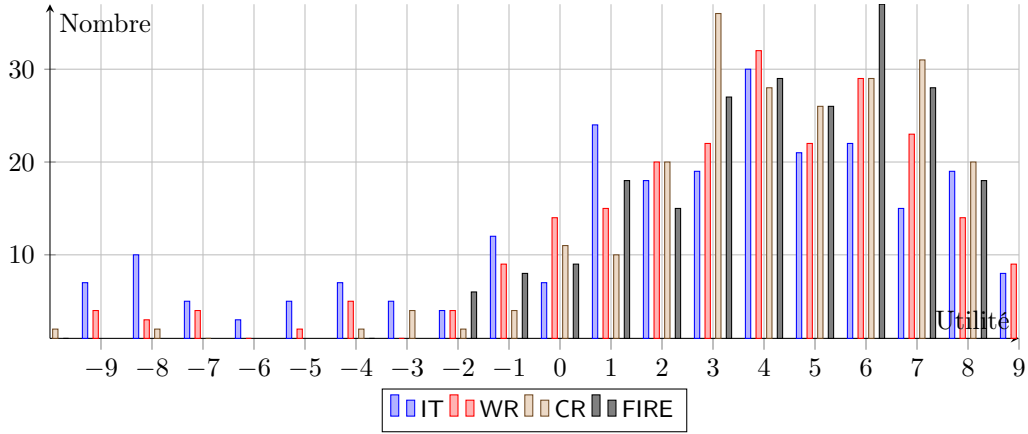


FIGURE 7 – Distribution des valeurs UG des clients

FIRE est le plus performant car moins étalé que ses concurrents (presque aucune valeurs entre -10 et -2). La composante IT, quant à elle, est la plus dispersée avec des concentrations relativement hautes en -8. FIRE étant un mix des trois autres composantes IT, WR et CR, sa courbe devrait être similaire à celle des trois autres modèles.

**2.4.4** On fait maintenant varier les paramètres  $N_{PG}$ ,  $N_{PO}$  et  $N_{PB}$  de la manière suivante : pour chacun la proportion d’agents fournisseurs de ce type varie dans les simulations. Le reste de la population est divisé équitablement entre les types restants. Ainsi si on a 40% de  $N_{PG}$ , alors on aura 30% de  $N_{PO}$  et 30% de  $N_{PB}$ . Les valeurs affichées dans les figures 8 sont calculées sur 5 simulations de 50 itérations.

Ces figures montrent que le modèle résiste aux faibles proportions de bons fournisseurs. Lorsque l’on fait varier  $N_{PB}$  par exemple, l’utilité moyenne enregistrée ne baisse de manière drastique que lorsque le nombre de bons fournisseurs est très réduit. Puisque les fournisseurs n’ont pas de limite sur leur nombre de clients, même un petit nombre de bons agents suffit à garantir une bonne utilité moyenne parmi les consommateurs. On remarque la même tendance pour les figures de  $N_{PG}$  et  $N_{PO}$ .

Les écart types en revanche montrent que la satisfaction des agents est d’autant plus hétérogène que les bons fournisseurs viennent à manquer. On pourra expliquer cela par la disposition géographique des agents : moins on compte d’agents  $N_{PG}$ , moins il est probable qu’un client en connaisse un dans son voisinage.

## 2.5 Extensions

Nous avons choisit comme extension l’ajout d’agents menteurs (2). Certains fournisseurs mentiront donc sur leur performances lorsque des agents consommateurs leur demandent les notes qu’ils ont reçus lors des transactions précédentes. Pour ce faire nous avons ajouté une variable qui détermine le nombre d’agents fournisseurs menteurs. Ces agents gardent leur status pendant toute la durée de la simulation. Lorsqu’un fournisseur est menteur il renverra toujours une liste de scores parfaits au consommateur qui lui demande ses certificats. Ce comportement est illustré dans l’algorithme 8.

Sachant que cette modification a uniquement un effet sur la composante CR, nous avons décidé de d’abord porter notre analyse sur le modèle FIRE-IT-CR uniquement puis d’étudier l’effet de la modification de cette composante sur le modèle FIRE entier. Plusieurs proportions d’agents menteurs ont été testées sur 5 exécutions du modèle FIRE-IT-CR (100 itérations) : une exécution n’ayant aucun agent menteur et les autres ayant respectivement 30%, 70% et 90% d’agents menteurs. Les résultats sont visibles dans la table 5. Le reste des paramètres de la simulation restent inchangés par rapport a la section 2.3.

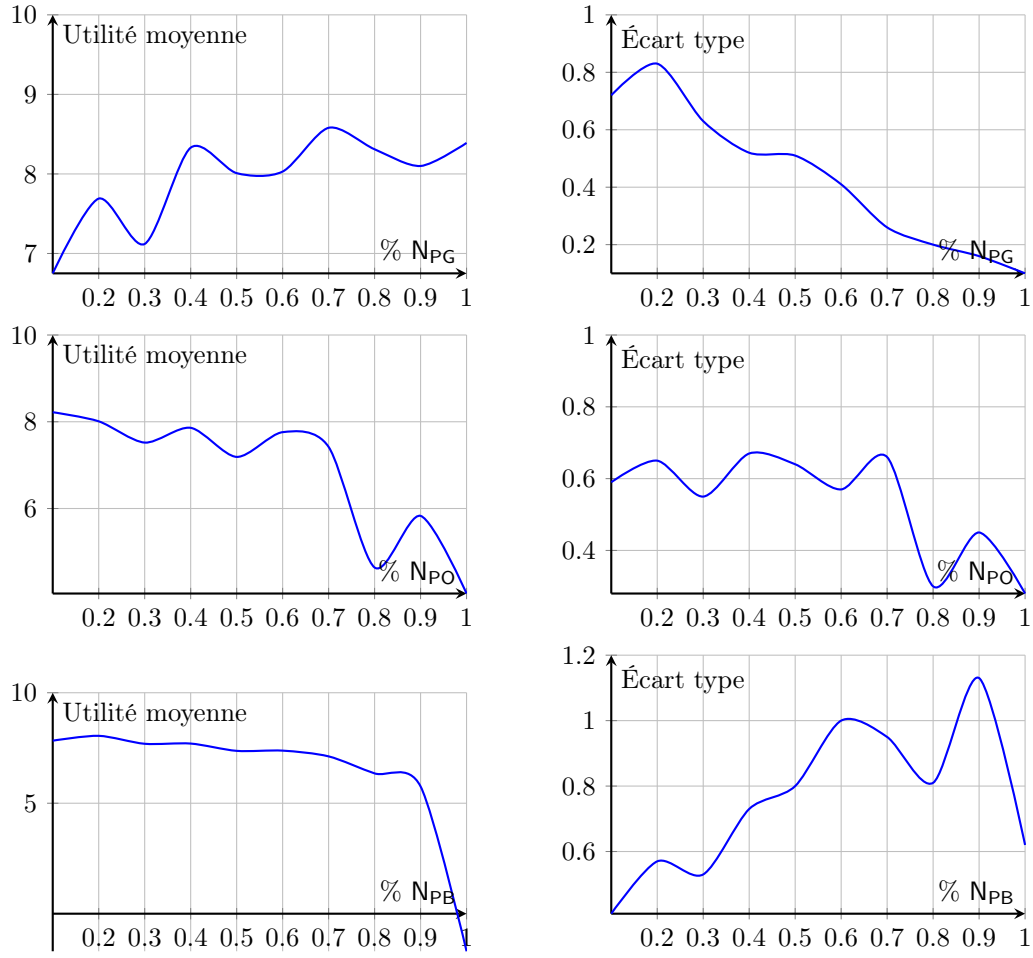


FIGURE 8 – Variation des paramètres  $N_{PG}$ ,  $N_{PO}$  et  $N_{PB}$

---

**Algorithme 8** Récupération des notes certifiées.

---

**Input :** *prov* le fournisseur courant  
**function** GETCERTIFIEDRATINGS(*prov*)  
      $returnValue \leftarrow \text{HISTORY}(prov)$   
     **if** isLiar **then**  
         **for all** (*prov*, *p*, *t*)  $\in returnValue$  **do**  
              $p \leftarrow \text{PL-PERFECT}$   
         **end for**  
     **end if**  
     **return**  $returnValue$   
**end function**

---

| % Menteurs | 0    | 0.3   | 0.6  | 0.9  |
|------------|------|-------|------|------|
| FIRE-IT-CR | 6.05 | 5.25. | 5.15 | 4.98 |
| FIRE       | 6.35 | 6.10. | 6.15 | 5.99 |

TABLE 5 – Utilités moyennes avec agents menteurs

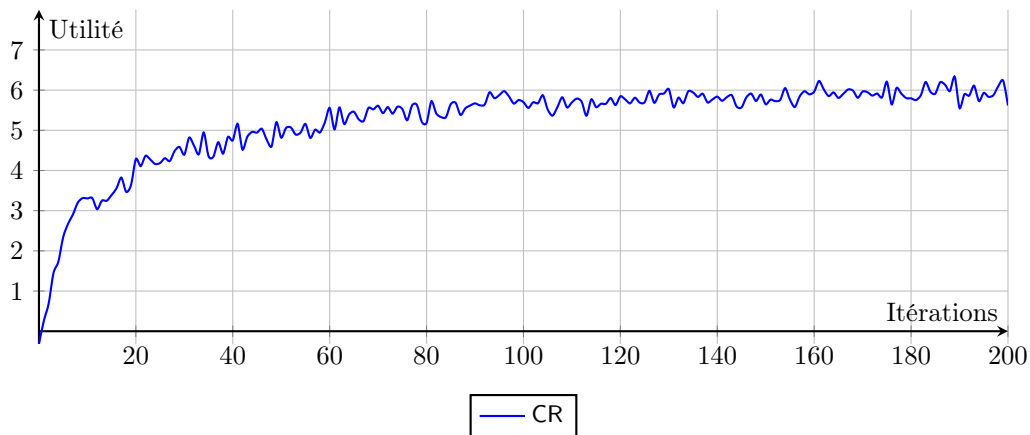


FIGURE 9 – Utilité moyenne des consommateurs avec CR lorsque 50% des fournisseurs mentent.

Il semble d'après ces tests que le pourcentage de menteurs fait effectivement baisser l'utilité moyenne dans les modèles testés (perte d'un point de UG pour FIRE-IT-CR entre 0% et 90% de menteurs). On remarque aussi que dans le modèle FIRE-IT-CR l'influence est forte en début de simulation. En effet on atteint le palier d'utilité beaucoup plus lentement lorsque le pourcentage de menteurs est supérieur à 0 comme le montre la figure 9 comparée à la figure 3. On explique cela par le fait que les consommateurs commencent certainement par choisir des menteurs car ils retournent immédiatement des valeurs de performance parfaites. On peut déduire de ces résultats que le modèle FIRE est relativement robuste vis-à-vis du type de mensonge que nous avons implémenté car la baisse d'utilité est relativement faible (surtout dans le modèle FIRE).

**Bonus** Pour que le modèle puisse être étendu il faut que les agents aient la possibilité de donner des notes à n'importe quel autre agent, sous la contrainte que ceux-ci aient une interaction sociale. On pourrait donc imaginer que le rayon  $r_0$  représente dans ce cas les agents avec qui on a la possibilité d'interagir. De ce fait il est possible de donner une note à un autre agent qui est dans mon rayon d'interaction avec qui j'ai eu une interaction sociale. Nous supposons que dans toute interaction sociale il y a un demandeur et un fournisseur (par exemple un client et un caissier). La performance du fournisseur sera générée à partir de sa note sociale, qui évolue au fur et à mesure du temps en fonction de la qualité des services fournis. On ajoute également le fait que lors d'une interaction, l'agent demandeur a la possibilité de donner une appréciation du service (positive ou négative) et c'est cette appréciation qui fait évoluer la note sociale. Elle peut être consultée par tous les autres agents pour savoir s'ils veulent interagir avec cette personne. Durant la simulation il semble crucial d'implémenter le déplacement des agents pour qu'ils n'aient pas toujours les mêmes voisins. Pour conclure avec la comparaison au modèle FIRE, un agent peut choisir deux façons différentes d'interagir avec un autre : selon ses expériences personnelles précédentes (IT) ou selon la note sociale publique de l'autre agent (WR). Certains agents peuvent donc se voir refuser des interactions si leur note sociale est trop basse.

## Références

- [1] Trung Dong HUYNH, Nicholas R. JENNINGS et Nigel R. SHADBOLT. « An Integrated Trust and Reputation Model for Open Multi-agent Systems ». In : *Autonomous Agents and Multi-Agent Systems* 13.2 (sept. 2006). DOI : 10.1007/s10458-005-6825-4.