

Auto-organisation par coopération pour l'adaptation

Principes et applications

Gauthier Picard

SMA / G2I / École Nationale Supérieure des Mines de Saint-Étienne (ENSM.SE)

`picard@emse.fr`

24 janvier 2008



Introduction

Motivations

- ▶ Environnements des logiciels de plus en plus complexes (ouverture, indéterminisme, dynamique, ...)
- ▶ Distribution logique/géographique/temporelle
- ▶ Fonctionnalité difficile à spécifier et besoin d'adaptation globale (non-linéarité, organisation inconnue, ...)
- ▶ Les systèmes sont décomposables et leurs parties spécifiabiles

Inspiration naturelle ou artificielle

- ▶ Mécanismes d'*auto-organisation*
- ▶ Systèmes *multi-agents*, intelligence par essaim
- ▶ La *coopération* comme mécanisme de micro-niveau

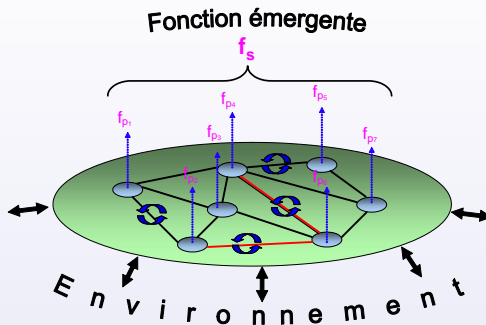


Sommaire

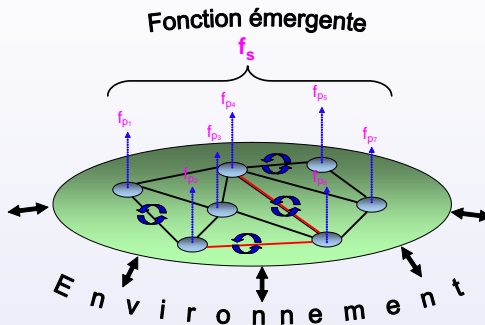
- 1 Systèmes multi-agents adaptatifs (AMAS)
- 2 Résolution de problèmes par auto-organisation
- 3 Applications
- 4 Conclusion



Auto-organisation



Auto-organisation



- ▶ 1 organisation \Rightarrow 1 fonction
- ▶ Changer d'organisation \Rightarrow changer de fonction
- ▶ Adaptation et apprentissage
- ▶ **Besoin d'un critère *local* de réorganisation non *explicitement* informé de la connaissance de la fonction globale**



Auto-organisation

Exemples de systèmes

- ▶ Physiques : cellules de convection de Bénard, réaction BZ, dunes, etc.
- ▶ Biologiques : pelages, coquillages, empreintes digitales, etc.
- ▶ Sociaux : hardes, volées, insectes sociaux, réseaux sociaux humains



Video source : Guy Theraulaz (CNRS, CRCA)



La coopération comme moteur de l'auto-organisation

Adaptation par auto-organisation coopérative [Caperla et al., 2003]

- ▶ L'auto-organisation est un moyen de construire des systèmes *adaptatifs*
- ▶ Le critère d'auto-organisation est la *coopération*

Coopération

Un agent est coopératif si c_{per} et c_{dec} et c_{act} :

c_{per} Les signaux perçus par l'agent sont compris sans ambiguïté

c_{dec} L'information reçue est utile au raisonnement de l'agent

c_{act} Le raisonnement de l'agent mène à des actions utiles aux autres



La coopération comme moteur de l'auto-organisation

Adaptation par auto-organisation coopérative [Caperla et al., 2003]

- ▶ L'auto-organisation est un moyen de construire des systèmes *adaptatifs*
- ▶ Le critère d'auto-organisation est la *coopération*

Coopération

Un agent est coopératif si c_{per} et c_{dec} et c_{act} :

c_{per} Les signaux perçus par l'agent sont compris sans ambiguïté

c_{dec} L'information reçue est utile au raisonnement de l'agent

c_{act} Le raisonnement de l'agent mène à des actions utiles aux autres



La coopération comme moteur de l'auto-organisation

Adaptation par auto-organisation coopérative [Capera et al., 2003]

- ▶ L'auto-organisation est un moyen de construire des systèmes *adaptatifs*
- ▶ Le critère d'auto-organisation est la *coopération*

Coopération

Un agent est coopératif si c_{per} et c_{dec} et c_{act} :

c_{per} Les signaux perçus par l'agent sont compris sans ambiguïté

c_{dec} L'information reçue est utile au raisonnement de l'agent

c_{act} Le raisonnement de l'agent mène à des actions utiles aux autres

➡ Approche proscriptive : les agents, pour bien fonctionner, doivent éviter et résoudre les situations non coopératives (SNC) : $\neg c_{per}$ ou $\neg c_{dec}$ ou $\neg c_{act}$



La coopération comme moteur de l'auto-organisation

Adaptation par auto-organisation coopérative [Capera et al., 2003]

- ▶ L'auto-organisation est un moyen de construire des systèmes *adaptatifs*
- ▶ Le critère d'auto-organisation est la *coopération*

Coopération

Un agent est coopératif si c_{per} et c_{dec} et c_{act} :

c_{per} Les signaux perçus par l'agent sont compris sans ambiguïté

c_{dec} L'information reçue est utile au raisonnement de l'agent

c_{act} Le raisonnement de l'agent mène à des actions utiles aux autres

➡ Approche proscriptive : les agents, pour bien fonctionner, doivent éviter et résoudre les situations non coopératives (SNC) : $\neg c_{per}$ ou $\neg c_{dec}$ ou $\neg c_{act}$

Raisonnement

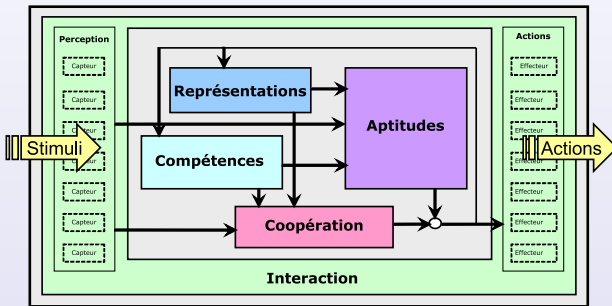
- ▶ Mesure de difficulté
- ▶ Prise en compte de la difficulté des voisins
- ▶ Action en réponse aux difficultés voisines



Un modèle d'agent coopératif

Architecture

- ▶ Cycle de vie "perception-décision-action"
- ▶ Plusieurs modules représentent une partition des capacités physiques, cognitives ou sociales
- ▶ Attitude sociale coopérative



Méthode ADELFE (Projet RNTL)

Atelier de Développement de Logiciels à Fonctionnalité Émergente

1 Un modèle d'agent coopératif

2 Un processus

- ▶ Extension du *Rational Unified Process*
- ▶ Activités spécifiques à la conception d'agents coopératifs
- ▶ Des besoins au développement
- ▶ Utilisation du SPEM

3 Des notations

- ▶ Extension d'A-UML
- ▶ Définition de stéréotypes UML

4 Des outils

- ▶ Support du processus, AdelfeToolkit
- ▶ Support des notations, OpenTool
 - ▶ Prise en charge des stéréotypes et vérification à la volée
 - ▶ Ajout des diagrammes de protocole A-UML
 - ▶ Transformation de protocoles en machines à états finis pour simulation
- ▶ Générateur de code Java
- ▶ Outil d'aide à la conception par simulation



Méthode ADELFE (Projet RNTL)

Atelier de Développement de Logiciels à Fonctionnalité Émergente

- ❶ Un modèle d'agent coopératif
- ❷ Un processus
 - ▶ Extension du *Rational Unified Process*
 - ▶ Activités spécifiques à la conception d'agents coopératifs
 - ▶ Des besoins au développement
 - ▶ Utilisation du SPEM
- ❸ Des notations
 - ▶ Extension d'A-UML
 - ▶ Définition de stéréotypes UML
- ❹ Des outils
 - ▶ Support du processus, AdelfeToolkit
 - ▶ Support des notations, OpenTool
 - ▶ Prise en charge des stéréotypes et vérification à la volée
 - ▶ Ajout des diagrammes de protocole A-UML
 - ▶ Transformation de protocoles en machines à états finis pour simulation
 - ▶ Générateur de code Java
 - ▶ Outil d'aide à la conception par simulation



Méthode ADELFE (Projet RNTL)

Atelier de Développement de Logiciels à Fonctionnalité Émergente

- ❶ Un modèle d'agent coopératif
- ❷ Un processus
 - ▶ Extension du *Rational Unified Process*
 - ▶ Activités spécifiques à la conception d'agents coopératifs
 - ▶ Des besoins au développement
 - ▶ Utilisation du SPEM
- ❸ Des notations
 - ▶ Extension d'A-UML
 - ▶ Définition de stéréotypes UML
- ❹ Des outils
 - ▶ Support du processus, AdelfeToolkit
 - ▶ Support des notations, OpenTool
 - ▶ Prise en charge des stéréotypes et vérification à la volée
 - ▶ Ajout des diagrammes de protocole A-UML
 - ▶ Transformation de protocoles en machines à états finis pour simulation
 - ▶ Générateur de code Java
 - ▶ Outil d'aide à la conception par simulation



Méthode ADELFE (Projet RNTL)

Atelier de Développement de Logiciels à Fonctionnalité Émergente

- ❶ Un modèle d'agent coopératif
- ❷ Un processus
 - ▶ Extension du *Rational Unified Process*
 - ▶ Activités spécifiques à la conception d'agents coopératifs
 - ▶ Des besoins au développement
 - ▶ Utilisation du SPEM
- ❸ Des notations
 - ▶ Extension d'A-UML
 - ▶ Définition de stéréotypes UML
- ❹ Des outils
 - ▶ Support du processus, AdelfeToolkit
 - ▶ Support des notations, OpenTool
 - ▶ Prise en charge des stéréotypes et vérification à la volée
 - ▶ Ajout des diagrammes de protocole A-UML
 - ▶ Transformation de protocoles en machines à états finis pour simulation
 - ▶ Générateur de code Java
 - ▶ Outil d'aide à la conception par simulation



Sommaire

- 1 Systèmes multi-agents adaptatifs (AMAS)
- 2 Résolution de problèmes par auto-organisation
 - Principes
 - Exemple 1 : n reines
 - Exemple 2 : affectation de fréquences
 - Types de problèmes
- 3 Applications
- 4 Conclusion



Principes

CSP distribués (DCSP/DCOP)

Avec distribution

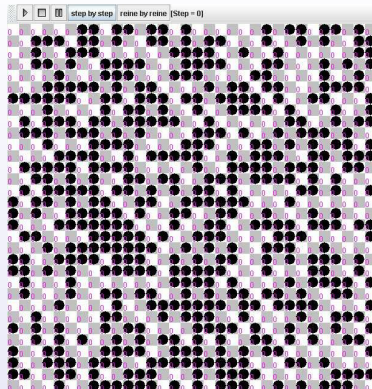
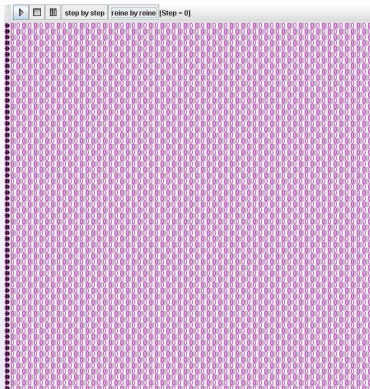
- ▶ de variables, de contraintes, de traitement, de fonctions, de disciplines...

Avec adaptation

- ▶ Adaptation à la dynamique de l'environnement : ouverture, utilisateur
- ▶ Minimisation des changements endogènes après modification des données



Exemple : n reines



- 1 reines sur même case → case perçue la moins contrainte
- 2 reines peuvent attaquer même case → déplacement impliquant le moins de violations OU laisser les plus contraintes se déplacer
- 3 reine perçoit une case moins contrainte → déplacement vers la case perçue la moins contrainte



Exemple 2 : affectation de fréquences

Problème d'affectation de fréquences (FAP) [Defaix, 2000]

Objectif Allouer un couple (*fréquence, polarisation*) à chaque *trajet* d'un réseau de télécommunication

Contraintes

$$\left\{ \begin{array}{l} |f_i - f_j| = \epsilon_{ij} \text{ ou } |f_i - f_j| \neq \epsilon_{ij} \text{ (CI)} \\ p_i = p_j \text{ ou } p_i \neq p_j \text{ (CI)} \\ |f_i - f_j| \geq \gamma_{ij}(p_i, p_j, k) \text{ avec } k = 0 \text{ à } 10 \text{ (CEM)} \end{array} \right.$$

Problème d'affectation de fréquence avec polarisation (FAPP)

Objectif Satisfaire les contraintes impératives et *minimiser* le coût des CEM

Propriétés Différents types de contraintes *binaires*, *larges* domaines fréquentiels (≈ 200), *grand* nombre de trajets (de 100 à 3000)

Méthodes efficaces et centralisées

- arc-consistance (pour réduire les domaines) et méthode tabou (pour résoudre le problème) [Vasquez, 2002] [Galinier et al., 2005] [Hertz et al., 2005]



Résolution multi-agent de CSP

Problèmes distribués de satisfaction de contraintes (DCSP)

- ▶ CSP classique
- ▶ répartition des variables parmi les agents
- ▶ organisation induite par les contraintes

Algorithmes connus

- ▶ ABT, AWCS, ADOPT, ERA, DynDBA, etc.
- ▶ des propriétés diverses : complétude, asynchronisme, type d'organisation (arbre, pseudo-arbre, etc.) [Yokoo et al., 1998] [Liu et al., 2002] [Modi et al., 2005]



Résolution multi-agent de CSP

Problèmes distribués de satisfaction de contraintes (DCSP)

- ▶ CSP classique
- ▶ répartition des variables parmi les agents
- ▶ organisation induite par les contraintes

Algorithmes connus

- ▶ ABT, AWCS, ADOPT, ERA, DynDBA, etc.
- ▶ des propriétés diverses : complétude, asynchronisme, type d'organisation (arbre, pseudo-arbre, etc.) [Yokoo et al., 1998] [Liu et al., 2002] [Modi et al., 2005]

➡ *notre approche : mécanismes locaux, asynchrone et non complet*



Description

Agent

- ▶ un agent est responsable de l'affectation de valeur à un trajet
- ▶ caractéristiques principales : valeur et difficulté
- ▶ vue limitée

Environnement

- ▶ social : le voisinage partageant des contraintes
- ▶ physique : les domaines et les contraintes

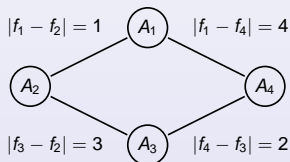
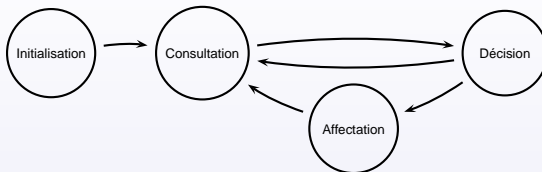


Figure: Exemple avec 4 agents et 4 contraintes impératives



Comportement des agents



Décision

- ▶ chaque agent compare sa difficulté avec ses voisins
- ▶ le plus en difficulté au sein d'un voisinage est élu

Affectation

- ▶ ne concerne que l'élu
- ▶ seule action physique pour un agent
- ▶ le *meilleur* changement de valeur selon l'agent



Communication

Description

- ▶ par messages et asynchrone
- ▶ permet l'échange des valeurs et des difficultés entre voisins

Session d'affectation

- ▶ bloque le voisinage de l' élu (point de synchronisation)
- ▶ l' élu invite ses voisins, et attend leur confirmation
- ▶ met à jour les vues de l' élu pour une meilleure action
- ▶ les voisins peuvent *annuler* une session
- ▶ résout le conflit pouvant survenir lors de la décision

Agents inéligibles

- ▶ deux élections consécutives (tabou au niveau des agents)
- ▶ toutes les contraintes sont satisfaites



Mesure de difficulté

- ▶ met en avant l'éloignement à une bonne solution selon l'agent
- ▶ déterminée de manière empirique (à la manière d'heuristiques)
- ▶ tuple trié de sous-critères

$$d_x^y = [Im_x^y, Po_x^y, NS_x^y, Ol_x^y]$$

Im Amélioration, étude du domaine des valeurs

$$\forall x \in A, Im_x = NS_x - \min\{NS_x(f, p), (f, p) \in F(x) \times P(x) \setminus (f, p)_x\}$$

Po Possibilité, difficulté des contraintes

$$\forall x \in A, Po_x = \min\{|FPS(c^x)|, c^x \in Cl_x \text{ et } c^x = false\}$$

NS Nombre de contraintes insatisfaites

Ol Nombre d'affectations dans le voisinage depuis la dernière satisfaction de la contrainte

$$\forall x \in A, Ol_x = \max\{TI_x(c^x), c^x \in Cl_x\}$$



Choix des valeurs

(1/2)

1^e réduction du domaine des valeurs : Critère discriminant

- ▶ lors de la comparaison des difficultés de l'élu et de ses voisins : un des critères permet de les distinguer
- ▶ choix de valeurs basé sur le critère qui a permis l'élection de l'agent
 - Im* : valeurs qui maximisent l'amélioration
 - Po* : ensemble de contraintes avec le minimum de possibilités
 - NS* : ensemble de contraintes avec le minimum de possibilités
 - Ol* : les contraintes les plus "âgées"
 - Eq* : les contraintes partagées avec les agents de difficulté égale



Choix des valeurs

(2/2)

2^e réduction du domaine des valeurs : dialogue avec les voisins

- ▶ amélioration *immédiate* de la situation du voisin
- ▶ amélioration *possible* de la situation du voisin
- ➔ utilité d'une valeur proposée par un élu pour un de ses voisins

$$u_x^y(f_x, p_x) = \max\{|S_y(f_y, p_y)|, (f_y, p_y) \in F_y \times P_y \text{ et } (f, p)_x^y = (f_x, p_x)\}$$

- ▶ attitude utilitariste : choix des valeurs qui maximisent la somme des utilités

3^e réduction du domaine des valeurs : choix de l'élu

- ▶ choix personnel en calculant l'amélioration immédiate de chaque valeur
- ▶ cas d'égalité de valeurs : choix aléatoire parmi les valeurs jugées équivalentes par l'élu



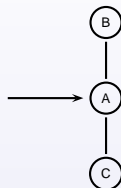
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



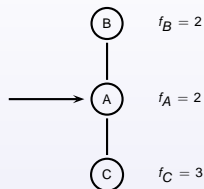
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



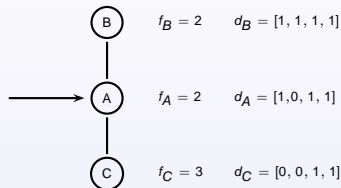
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



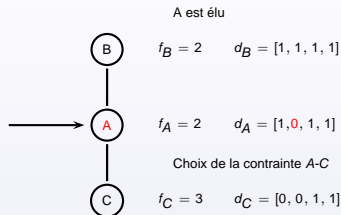
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



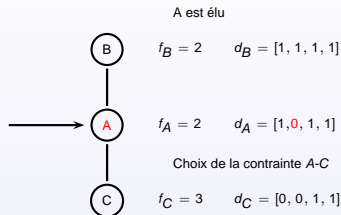
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$

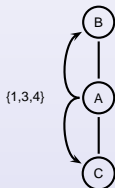


Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$

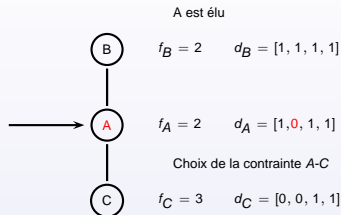


Dialogue avec les voisins

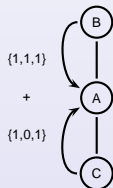


Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$

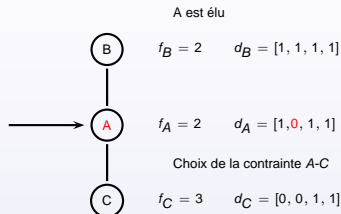


Dialogue avec les voisins

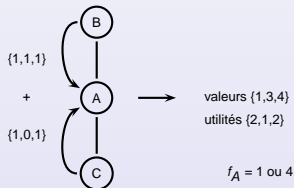


Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$

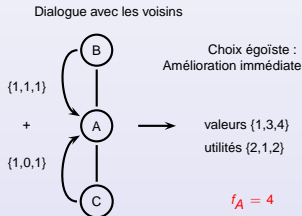
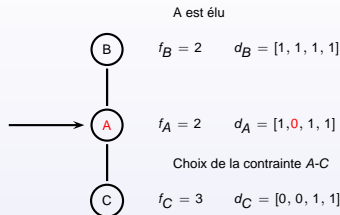


Dialogue avec les voisins



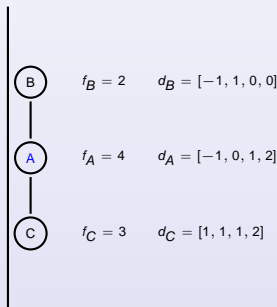
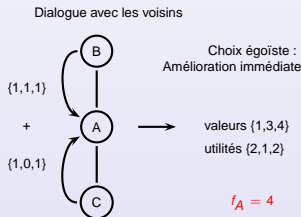
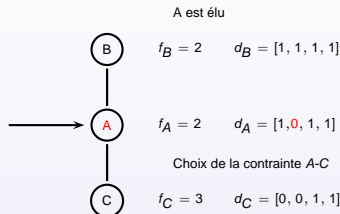
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



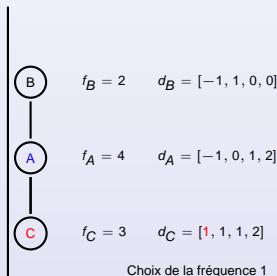
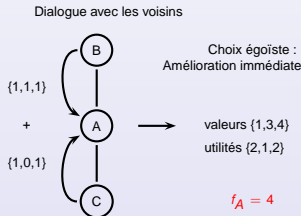
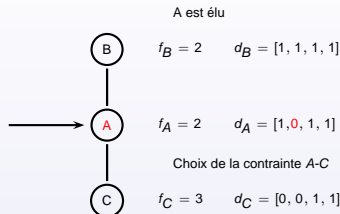
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



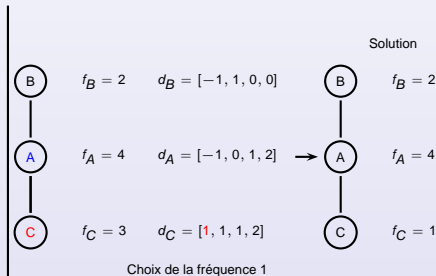
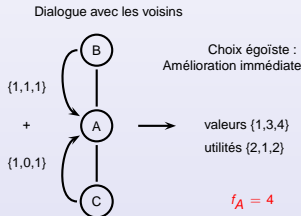
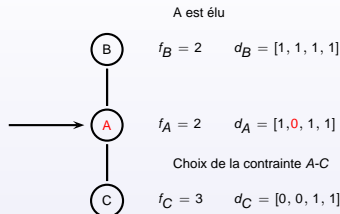
Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



Exemple

- ▶ 3 trajets : A, B et C
- ▶ $D_f = \{1, 2, 3, 4\}$
- ▶ pas de polarisation
- ▶ $|f_A - f_B| = 2$
- ▶ $|f_A - f_C| = 3$



Convergence

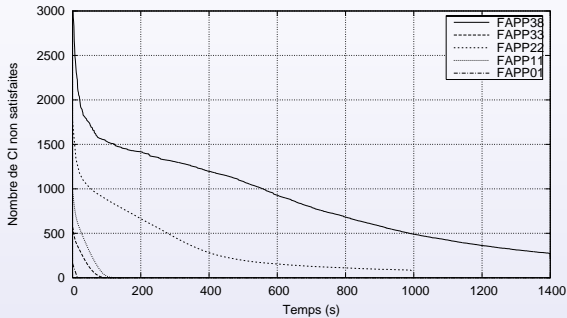


Figure: Convergence vers des solutions



Affectations

Table: Pourcentage de CI insatisfaites

Instance FAPP	01	11	22	33	38
CI	168	978	1799	578	3112
% non satisfaites	0	0,015	3,49	0,004	6,82

Table: Affectations

Instance FAPP	01	11	22	33	38
Affect.	107.24	547.02	964.81	338.35	1146.18
Affect./CI	0.64	0.56	0.54	0.59	0.37
Affect./Temps(s)	17.87	3.42	0.96	2.26	0.82
σ	5.33%	6.89%	5.36%	2.28%	3.88%

- ✗ peu de changements de valeurs par seconde
- ✓ des changements de valeurs bien choisis
- ✓ indépendance vis-à-vis des situations initiales



Trafic des messages

Table: Trafic des messages

Instances FAPP	01	11	22	33	38
Agents	200	1,000	1,750	650	2,500
Messages	2,335	15,841	33,247	8,348	48,727
Messages/Agent	11.7	15.8	19	12.8	19.5

Table: Pourcentage de messages d'annulation

Instance FAPP	01	11	22	33	38
% annulation	4.21	6.02	13.56	5.22	10.28



Pertinence des critères de décision

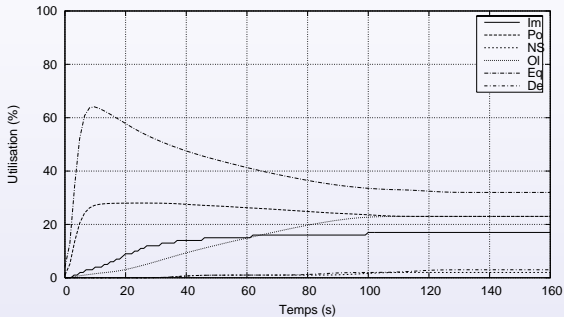


Figure: Distribution des critères discriminant pour l'instance FAPP11



Discussion

Communication

- ✓ les mises à jour d'informations peuvent être réduites
- ✓ les sessions d'affectation peuvent être réduites
 - ▶ limitation du temps d'invitation
 - ▶ limitation du nombre d'invités : préférences
 - ▶ agents invités à plus d'une session à la fois

Évaluation de la difficulté

- ✓ les critères sont génériques
 - ▶ l'ordonnancement des critères reste une question ouverte

Contraintes souples et optimisation

- ✗ CEM sont plus nombreuses que les CI
 - ▶ 2 types de voisinages peuvent être considérés
 - ▶ comment inclure les CEM dans la mesure de difficulté ?

Comparaison avec d'autres approches

- ✓ pas d'ajout de voisinage virtuel (APO, ABT)
- ✓ pas de hiérarchie (ADOPT) → ouverture
 - ▶ le critère de terminaison est une limite de temps
- ✗ 2 fois moins efficace qu'ADOPT sur de la coloration de graphe (mais topologies différentes)



Types de problèmes

- ▶ Emploi du temps/Scheduling
 - ▶ Emploi du temps universitaire
 - ▶ Emploi du temps aéroportuaire
- ▶ Manufacturing control
 - ▶ Affectation de tâches
 - ▶ Navigation
- ▶ Conception aéronautique
 - ▶ Conception de mécanismes
 - ▶ Conception avion préliminaire
- ▶ ...



Sommaire

- 1 Systèmes multi-agents adaptatifs (AMAS)
- 2 Résolution de problèmes par auto-organisation
- 3 Applications**
 - Emploi du temps universitaire
 - Robotique collective
 - Conception aéronautique
- 4 Conclusion



Emploi du temps universitaire

Cahier des charges

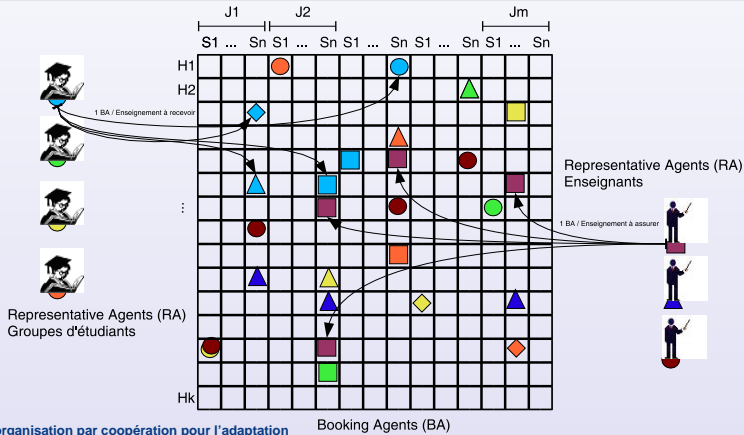
- ▶ Solveur dynamique d'emplois du temps universitaires
- ▶ Les contraintes des acteurs sont modifiables à tout moment
- ▶ Les résolutions ne doivent pas se reconstruire à chaque modification



Emploi du temps universitaire

Cahier des charges

- Solveur dynamique d'emplois du temps universitaires
- Les contraintes des acteurs sont modifiables à tout moment
- Les résolutions ne doivent pas se reconstruire à chaque modification



Les agents dans ETTO

RepresentativeAgents

- ▶ Représentent les acteurs humains (enseignants, étudiants)
- ▶ Regroupent les contraintes de disponibilité, d'équipements, ...
- ▶ Délèguent la recherche de créneaux et de salles aux BookingAgents

Les BookingAgents

- ▶ Représentent un enseignement d'un acteur (à suivre ou à donner)
- ▶ Réservent les créneaux horaires et les salles de manière coopérative
- ▶ Explorent la grille de l'emploi du temps (n-dimensions) à la recherche de créneaux, de salles et de partenaires
- ▶ Modifient l'environnement en posant des marqueurs dans les cases de la grille
- ▶ Négocient avec les autres BAs pour trouver une organisation adéquate
- ▶ La cohérence de leurs réservations est assurée par leur RepresentativeAgent



Les agents dans ETTO

RepresentativeAgents

- ▶ Représentent les acteurs humains (enseignants, étudiants)
- ▶ Regroupent les contraintes de disponibilité, d'équipements, ...
- ▶ Délèguent la recherche de créneaux et de salles aux BookingAgents

Les BookingAgents

- ▶ Représentent un enseignement d'un acteur (à suivre ou à donner)
- ▶ Réservent les créneaux horaires et les salles de manière coopérative
- ▶ Explorent la grille de l'emploi du temps (n-dimensions) à la recherche de créneaux, de salles et de partenaires
- ▶ Modifient l'environnement en posant des marqueurs dans les cases de la grille
- ▶ Négocient avec les autres BAs pour trouver une organisation adéquate
- ▶ La cohérence de leurs réservations est assurée par leur RepresentativeAgent



Coopération

Règles de coopération

- ▶ Incompétence de partenariat → déplacement + mémorisation
- ▶ Incompétence de réservation → déplacement + mémorisation
- ▶ Improductivité de message → retour du message
- ▶ Conflit de partenariat → négociation (déplacement ou partenariat)
- ▶ Conflit de réservation → négociation (déplacement ou réservation)
- ▶ Inutilité de réservation → déplacement



Coopération

Règles de coopération

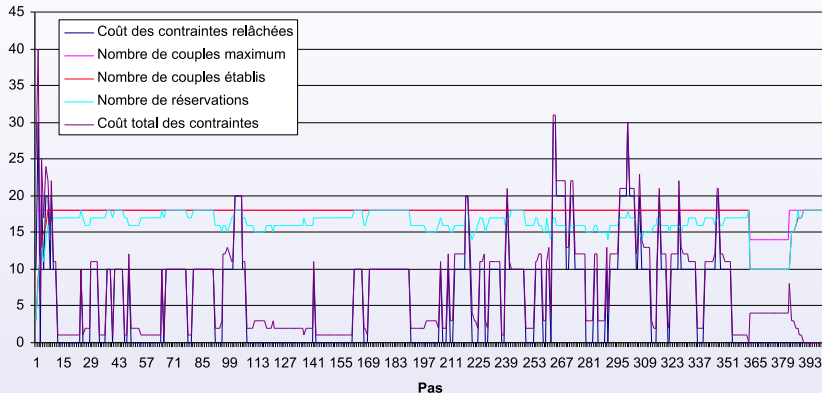
- ▶ Incompétence de partenariat → déplacement + mémorisation
- ▶ Incompétence de réservation → déplacement + mémorisation
- ▶ Improductivité de message → retour du message
- ▶ **Conflit de partenariat** → négociation (déplacement ou partenariat)
- ▶ **Conflit de réservation** → négociation (déplacement ou réservation)
- ▶ Inutilité de réservation → déplacement

Mesure de difficulté

- ▶ Contraintes pondérées
- ▶ Calcul du coût d'un partenariat
- ▶ Calcul du coût d'une réservation
- ▶ Prise en compte du temps de recherche



Expérimentations



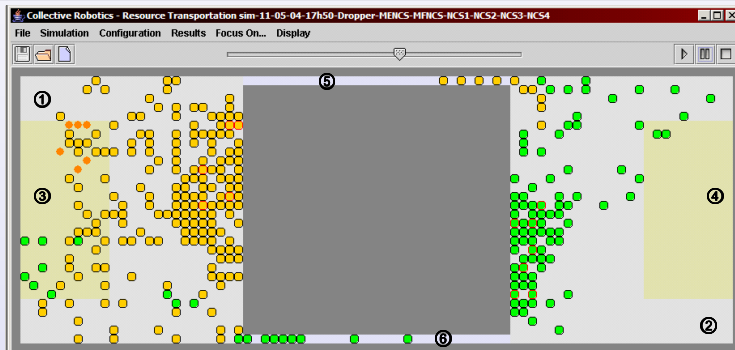
Variation des contraintes en cours de résolution, avec suppression de 8 BookingAgents après stabilisation du système

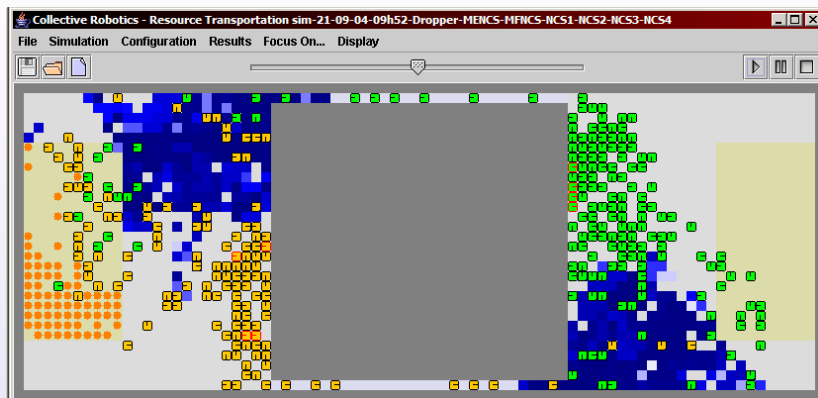


Navigation sous environnement contraint

Robots

- ▶ Autonomes
- ▶ Tâche de transport de ressources
- ▶ Entités de micro-niveau
- ▶ Non communicants

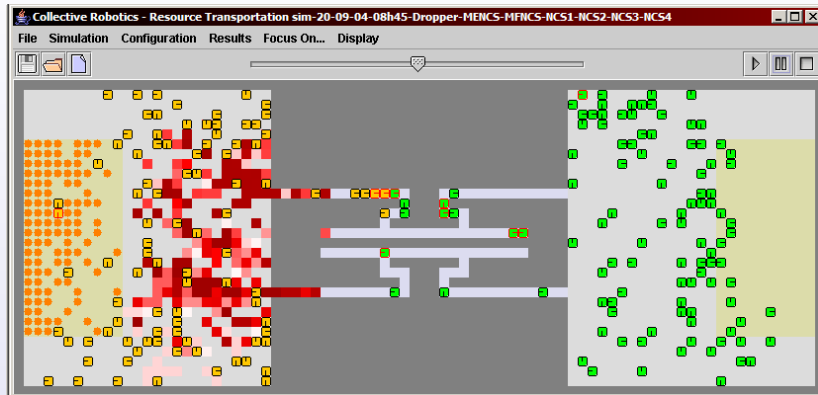




Positionnement de tous les marqueurs virtuels (carrés sombres) pour tous les robots et les deux buts



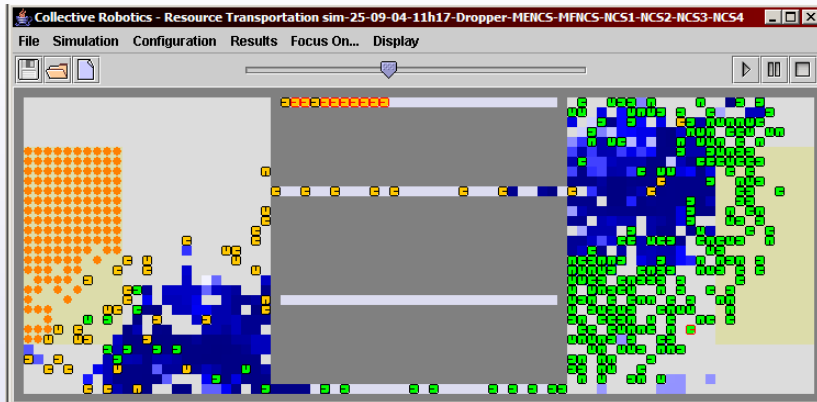
Marqueurs virtuels



Positionnement de tous les marqueurs virtuels (carrés sombres) pour le but *atteindre la zone de dépôt* dans un environnement difficile (avec impasses)



Environnement dynamique



Positionnement de tous les marqueurs virtuels (carrés sombres) dans un environnement dynamique avec deux couloirs fermés



Conception aéronautique

Présentation

- ▶ Besoin d'outil d'aide à la conception aéronautique
- ▶ Interaction avec les concepteurs
- ▶ Besoin d'adaptation à la dynamique
- ▶ Domaines continus

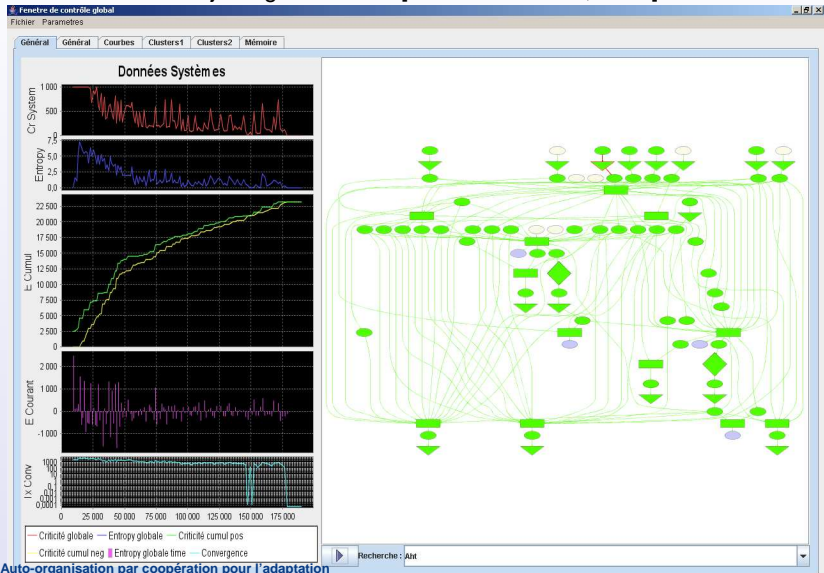
Exemples

- ▶ Conception préliminaire
 - ▶ Agents = disciplines (géométrie, aérodynamique, masse, etc.)
 - ▶ Coopération = négociation des paramètres des fonctions
 - ▶ Objectif = satisfaction puis optimisation des contraintes
- ▶ Conception de mécanismes
 - ▶ Agents = composants (châssis, barres, ressorts, etc.)
 - ▶ Coopération = négociation des dimensions et propriétés entre composants voisins
 - ▶ Objectif = suivi de trajectoire



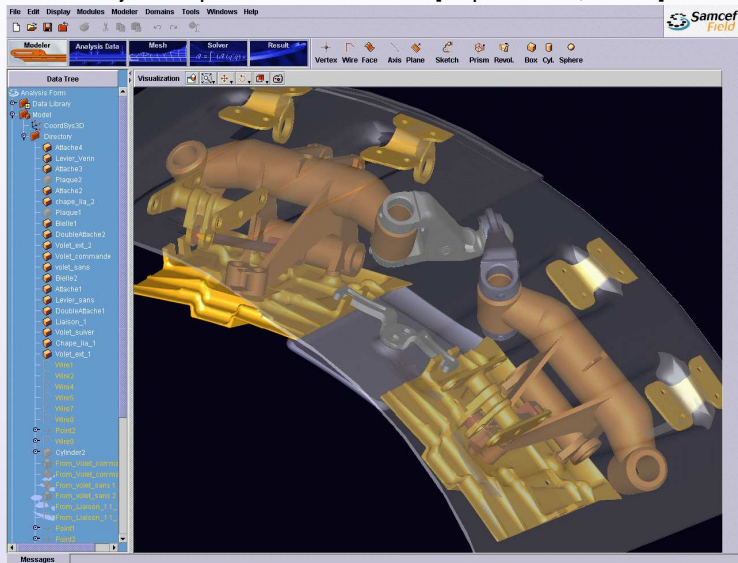
Conception préliminaire

Projet région ATOCA [Welcomme et al., 2007]



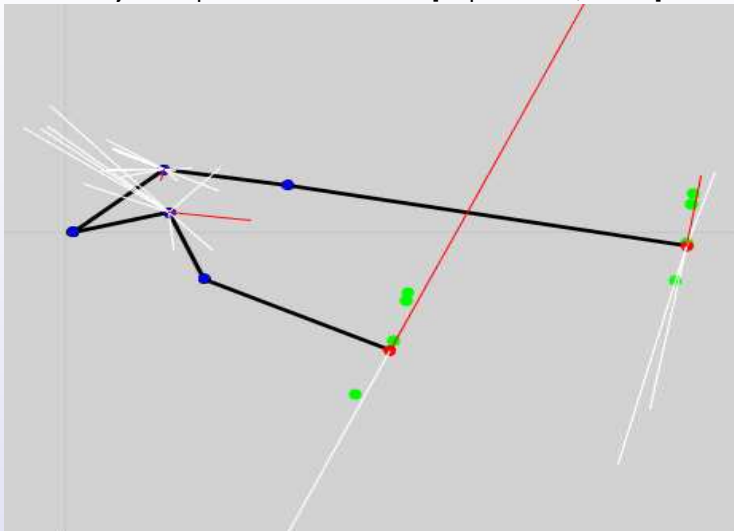
Conception de mécanismes

Projet européen FP5 SYNAMEC [Capera et al., 2004a]



Conception de mécanismes

Projet européen FP5 SYNAMEC [Capera et al., 2004a]



Sommaire

- 1 Systèmes multi-agents adaptatifs (AMAS)
- 2 Résolution de problèmes par auto-organisation
- 3 Applications
- 4 Conclusion**



Conclusion

Bilan

- ▶ Domaine aux inspirations nombreuses
- ▶ Méthodologie de développement de SMA adaptatifs
 - ▶ Approche **bottom-up**
 - ▶ Modélisation/conception de l'environnement
 - ▶ Conception des agents
 - ▶ Conception des règles d'auto-organisation
- ▶ Résolution/optimisation distribuée
 - ▶ Avantages
 - ✓ **Adaptation** à la dynamique
 - ✓ **Traçage** des problèmes
 - ✓ **Anytime**
- ▶ Simulation / Modélisation
- ▶ Applications nombreuses



Conclusion (cont.)

Principales pistes d'études

- ▶ Ingénierie
 - ✗ Pas de contrôle/émergence → pas de preuve de convergence, de stabilité, etc.
 - ▶ Besoin de **preuves** formelles
 - ▶ **Comment choisir les règles** de coopération pour un problème donné ?
- ▶ De l'émergence faible à l'émergence forte : **bouclage organisationnel**
 - 1- auto-organisation
 - 2- réflexion organisationnelle et mémorisation
 - 3- perturbation → auto-organisation (pas de réflexion organisationnelle) or ré-organisation (réflexion organisationnelle)

...
- ▶ Questions de fonds
 - ▶ Du global au local
 - ▶ Comment déduire un critère local d'optimisation à partir du critère global ?
 - ▶ Ce critère doit être coopératif et le moins dépendant du problème
 - ▶ Du local au global
 - ▶ Quel critère local implique l'émergence d'un comportement adéquat (optimum) ?
 - ▶ Quels protocoles sûrs pour supporter de telles résolutions distribuées ?



Projets en cours de montage

ID4CS (ANR)

- ▶ Integrative Design for Complex Systems
- ▶ Appel : ANR COSINUS
- ▶ Partenaires : IRIT, EADS, Airbus, IMT, LGMT, Artal, EMSE
- ▶ Ingénierie aéronautique

➡ Séminaire et réunion de préparation le 14/02/08 au G2I

AmlCriM (FP7)

- ▶ Ambient Intelligence for Crisis Management
- ▶ Appel : FP7 ICT Call 3 - FP7-ICT-2007-8.5
- ▶ Partenaires : IRIT, Telefonica, EMSE, University of Trento, Fraunhofer IAIS, University of Antwerp, Loughborough University
- ▶ Auto-organisation de services pour l'aide aux personnes en cas de catastrophe en milieu urbain



Des questions ?



Références

- C. Bernon, M.-P. Gleizes, S. Peyruqueou, and G. Picard. ADELFE: a Methodology for Adaptive Multi-Agent Systems Engineering. In P. Petta, R. Tolksdorf, and F. Zambonelli, editors, *Third International Workshop on Engineering Societies in the Agents World (ESAW'02)*, volume 2577 of *Lecture Notes in Computer Science (LNCS)*, pages 156–169, Madrid, Spain, September 2002a. Springer-Verlag. ISBN 3-540-14009-3.
- C. Bernon, M.-P. Gleizes, G. Picard, and P. Glize. The ADELFE Methodology For an Intranet System Design. In P. Giorgini, Y. Lespérance, G. Wagner, and E. Yu, editors, *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*, volume 57, Toronto, Canada, May 2002b. CAiSE'02, CEUR Workshop Proceedings.
- C. Bernon, V. Camps, M.-P. Gleizes, and G. Picard. Designing Agents' Behaviours within the Framework of ADELFE Methodology. In A. Omicini, P. Petta, and J. Pitt, editors, *Fourth International Workshop on Engineering Societies in the Agents World (ESAW'03)*, volume 3071 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 311–327, Imperial College London, UK, October 2003a. Springer-Verlag.
- C. Bernon, V. Camps, M.-P. Gleizes, and G. Picard. Tools for Self-Organizing Applications Engineering. In G. Di Marzo Serugendo, A. Karageorgos, O. F. Rana, and F. Zambonelli, editors, *Engineering Self-Organizing Applications – First International Workshop (ESOA) at the Second International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'03)*, volume 2977 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 283–298, Melbourne, Australia, 2003b. Springer-Verlag. ISBN 3-540-21201-9.
- C. Bernon, V. Camps, M.-P. Gleizes, and G. Picard. *Engineering Self-Adaptive Multi-Agent Systems : the ADELFE Methodology*, chapter 7, pages 172–202. Idea Group Publishing, 2005.
- C. Bernon, M.-P. Gleizes, and G. Picard. Enhancing Self-Organising Emergent Systems Design with Simulation. In *Seventh International Workshop on Engineering Societies in the Agents World (ESAW'06)*, Dublin, Ireland from the 6th - 8th September, 2006, 2006.
- D. Capera, JP. Georgé, M.-P. Gleizes, and P. Glize. The AMAS theory for complex problem solving based on self-organizing cooperative agents. In *1st Int. TAPOCS Workshop at 12th IEEE WETICE*, pages 383–388. IEEE, 2003.
- D. Capera, M.-P. Gleizes, and P. Glize. Mechanism Type Synthesis based on Self-Assembling Agents. *Journal of Applied Artificial Intelligence*, 18(9-10):921–936, 2004a.
- D. Capera, G. Picard, M.-P. Gleizes, and P. Glize. Applying ADELFE Methodology to a Mechanism Design Problem. In *Third Joint Conference on Multi-Agent System (AAMAS'04)*, pages 1508–1509, New York, USA, July 2004b. IEEE Computer Society. ISBN 1-58113-864-4.



Références (cont.)

- D. Capera, G. Picard, M.-P. Gleizes, and P. Glize. A Sample Application of ADELFE Focusing on Analysis and Design : The Mechanism Design Problem. In M.-P. Gleizes, A. Omicini, and F. Zambonelli, editors, *Fifth International Workshop on Engineering Societies in the Agents World (ESAW'04), 20-22 October 2004, Toulouse, France*, volume 3451 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 231–244. Springer-Verlag, 2005.
- T. Deafx. FAPP: Frequency Assignment with Polarization Problem - ROADEF Challenge 2001. Technical Report Revision 2, CELAR/TCOM, 2000.
- P. Galinier, M. Gendreau, and P. Soriano. Solving the Frequency Assignment Problem with Polarization by Local Search and Tabu. *4OR*, 3(1):59–78, 2005.
- J.-P. Georgé, G. Picard, M.-P. Gleizes, and P. Glize. Living Design for Open Computational Systems. In M. Fredriksson, A. Ricci, R. Gustavsson, and A. Omicini, editors, *International Workshop on Theory And Practice of Open Computational Systems (TAPOCS) at 12th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03)*, pages 389–394, Linz, Austria, June 2003. IEEE Computer Society. ISBN 0-7695-1963-6.
- A. Hertz, D. Schindl, and N. Zufferey. Lower bounding and tabu search procedures for the frequency assignment problem with polarization constraints. *4OR*, 3(2):139–161, 2005.
- J. Liu, H. Jing, and Y. Y. Tang. Multi-agent Oriented Constraint Satisfaction. *Artificial Intelligence*, 136(1):101–144, 2002.
- P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees. *Artificial Intelligence*, 161(2):149–180, 2005.
- K. Ottens, G. Picard, and V. Camps. Transformation de modèles d'agents dans la méthode ADELFE : Des stéréotypes de conception à l'implémentation. *Revue Technique et Science Informatique – L'objet*, 11(3), 2006.
- G. Picard. ADELFE, une méthode de conception de systèmes multi-agents adaptatifs. In *Colloque des Doctorants de l'Ecole Doctorale Informatique et Télécommunication (EDIT'03)*, pages 7–11, Université Paul Sabatier, Toulouse, France, Avril 2003a.
- G. Picard. UML Stereotypes Definition and AUML Notations for ADELFE Methodology with OpenTool. In *The First European Workshop on Multi-Agent Systems (EUMAS'03), St Catherine College, Oxford, 18th and 19th December 2003*, 2003b.
- G. Picard. Cooperative Agent Model Instantiation to Collective Robotics in ADELFE. In M.-P. Gleizes, A. Omicini, and F. Zambonelli, editors, *Fifth International Workshop on Engineering Societies in the Agents World (ESAW'04), 20-22 October 2004, Toulouse, France*, volume 3451 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 209–221. Springer-Verlag, 2005a.



Références (cont.)

- G. Picard. Résolution d'emploi du temps dynamique et distribuée par auto-organisation coopérative. In *7èmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA'05)*, Plate-forme AFIA, Nice, pages 127–140. Presses Universitaires de Grenoble (PUG), 2005b. ISBN 2-7061-1285-9.
- G. Picard and M.-P. Gleizes. An Agent Architecture to Design Self-Organizing Collectives: Principles and Application. In D. Kazakov, D. Kudenko, and E. Alonso, editors, *AISB'02 Symposium on Adaptive Multi-Agent Systems (AAMASII)*, volume 2636 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 141–158, Univerity of London, UK, April 2002. Springer-Verlag. ISBN 3-540-40068-0.
- G. Picard and M.-P. Gleizes. Outils pour la réalisation de systèmes multi-agents adaptatifs dans le cadre de la méthode ADELFE. In J.P. Briot and K. Ghedira, editors, *Déploiement des systèmes multi-agents – Vers un passage à l'échelle (Journées Francophones sur les Systèmes Multi-Agents, JFSMA'03)*, pages 249–254, Hammamet, Tunisia, 27-29 November 2003. Hermès - Lavoisier (Revue RSTI Hors-série). ISBN 2-7462-0810-5.
- G. Picard and M.-P. Gleizes. *The ADELFE Methodology – Designing Adaptive Cooperative Multi-Agent Systems*, chapter 8, pages 157–176. Kluwer Publishing, 2004. ISBN 1-4020-8057-3.
- G. Picard and M.-P. Gleizes. Cooperative Self-Organization to Design Robust and Adaptive Collectives. In *Second International Conference on Informatics in Control, Automation and Robotics (ICINCO'05)*, 14-17 September 2005, Barcelona, Spain, Volume I, pages 236–241. INSTICC Press, 2005.
- G. Picard and M.-P. Gleizes. Auto-organisation coopérative pour la conception de collectifs adaptatifs et robustes. In *7ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06)*, 6, 7 et 8 Février 2006, Lille, France, pages 385–400. Presses Universitaires de Valenciennes, 2006.
- G. Picard and P. Glize. Model and Experiments of Local Decision Based on Cooperative Self-Organization. In Prasad, B., editor, *Second International Indian Conference on Artificial Intelligence (IICAI'05)*, 20-22 December 2005, Pune, India, pages 3009–3024, 2005a. ISBN 0-9727412-1-6.
- G. Picard and P. Glize. Modélisation et expérimentations d'une décision locale basée sur l'auto-organisation coopérative. In *Journées Francophones sur les Systèmes Multi-Agents (JFSMA'05)*, à Calais du 23 au 25 novembre 2005, pages 161–174. Hermès-Lavoisier, 2005b.
- G. Picard and P. Glize. Model and Analysis of Local Decision Based on Cooperative Self-Organization for Problem Solving. *Multiagent and Grid Systems – An International Journal (MAGS)*, 2(3):253–265, 2006.
- G. Picard, C. Bernon, V. Camps, and M.-P. Gleizes. ADELFE : Atelier de développement de logiciels à fonctionnalité émergente. In J.P. Briot and K. Ghedira, editors, *Démonstration lors des Journées Francophones sur les Systèmes Multi-Agents, JFSMA'03*, pages 387–392, Hammamet, Tunisia, 27-29 November 2003. Hermès - Lavoisier (Revue RSTI Hors-série). ISBN 2-7462-0810-5.



Références (cont.)

- G. Picard, C. Bernon, and M.-P. Gleizes. Cooperative Agent Model within ADELFE Framework: An Application to a Timetabling Problem. In *Third Joint Conference on Multi-Agent System (AAMAS'04)*, pages 1506–1507, New York, USA, July 2004. IEEE Computer Society. ISBN 1-58113-864-4.
- G. Picard, C. Bernon, and M.-P. Gleizes. Emergent Timetabling Organization. In *Multi-Agent Systems and Applications IV - 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05)*, 15-17 September 2005, Budapest, Hungary, volume 3690 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 440–449. Springer-Verlag, 2005a.
- G. Picard, C. Bernon, and M.-P. Gleizes. ETTO : Emergent Timetabling by Cooperative Self-Organization. In *Engineering Self-Organizing Applications – Third International Workshop (ESOA) at the Fourth International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'05)*, July 2005, Utrecht, Netherlands, volume 3910 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 31–45. Springer-Verlag, 2005b.
- G. Picard, S. Mellouli, and M.-P. Gleizes. Techniques for Multi-Agent System Reorganization. In O. Dikenelli, M.-P. Gleizes, and A. Ricci, editors, *Sixth International Workshop on Engineering Societies in the Agents World (ESAW'05)*, 26-28 October 2005, Kuşadası, Aydın, Turkey, volume 3963 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer-Verlag, 2005c.
- G. Picard, M.-P. Gleizes, and P. Glize. Distributed Frequency Assignment Using Cooperative Self-Organization. In *First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07)*, Boston, Mass., USA, July 9-11, 2007, 2007.
- M. Vasquez. Arc-consistency and tabu search for the frequency assignment problem with polarization. In *CP-AI-OR'02*, pages 359–372, 2002.
- J.-B. Welcomme, M.-P. Gleizes, and R. Redon. A Self-Organising Multi-Agent System Managing Complex System Design Application to Conceptual Aircraft Design. In *International Conference on Complex Open Distributed Systems (CODS)*, Chengdu, 22/07/2007-24/07/2007, 2007.
- M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. The Distributed Constraint Satisfaction Problem : Formalization and Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, 1998.

