

# Diverses techniques d'optimisation inspirées de la théorie de l'auto-organisation dans les systèmes biologiques

Johann Dréo, Patrick Siarry

Université de Paris XII Val-de-Marne

Laboratoire d'Etude et de Recherche en Instrumentation, Signaux et Systèmes

(LERISS, EA 412)

61, avenue du Général de Gaulle, 94010 Créteil

E-mail : [dreo@univ-paris12.fr](mailto:dreo@univ-paris12.fr), [siarry@univ-paris12.fr](mailto:siarry@univ-paris12.fr)

24 octobre 2003

*Mots clefs :*

métaheuristique, intelligence en essaim, auto-organisation, programmation à mémoire adaptative.

## 1 Introduction

La biologie des organismes est la source d'inspiration de nombreuses métaheuristiques. Ainsi, les théories de l'évolution ont inspiré les algorithmes évolutionnaires, les phénomènes de suivi de piste chez les fourmis ont conduit à l'élaboration des algorithmes de colonies de fourmis, l'étude de l'organisation de groupes d'animaux a donné naissance aux méthodes d'optimisation par essais particuliers. Il existe, en outre, d'autres algorithmes, moins connus que ceux que nous venons de citer, qui découlent de la biologie : en particulier, des algorithmes inspirés du fonctionnement du système immunitaire, des algorithmes pour l'allocation dynamique de tâches, s'appuyant sur des modèles d'organisation du travail chez les fourmis, des algorithmes de classification suggérés par les bancs de poissons.

Une contribution importante de la biologie dans ce domaine vient de la *théorie de l'auto-organisation*, qui permet d'analyser les propriétés de plusieurs métaheuristiques issues des métaphores biologiques. Cette théorie — encore peu connue, en dehors de la communauté de l'éthologie — décrit les conditions d'apparition de phénomènes complexes à partir de systèmes distribués, dont les agents font l'objet d'interactions simples, mais nombreuses. La théorie met en avant des concepts tels que la communication, l'hétéarchie dense, les rétroactions, l'amplification des fluctuations et l'émergence. L'intelligence en essaim est

ainsi née sur deux fronts : via une approche “systèmes auto-organisés” (ayant donné lieu aux algorithmes de colonies de fourmis) et via une approche “systèmes socio-cognitifs” (ayant conduit à l’optimisation par essaim particulière).

Nous proposons de mettre la théorie de l’auto-organisation en relation avec le concept de *programmation à mémoire adaptative*, qui tente de décrire les points clefs des métaheuristiques modernes, en insistant notamment sur le rôle de la mémoire et des mécanismes d’intensification et de diversification.

En nous plaçant dans le cadre des théories de l’auto-organisation et de la programmation à mémoire adaptative, nous avons conçu un algorithme inspiré des colonies de fourmis : l’algorithme CIAC (“Continuous Interacting Ant Colony”), qui insiste sur l’importance des canaux de communication. Dans cet algorithme, les fourmis artificielles possèdent deux moyens d’échange des informations : via des pistes de phéromone (principe de base des algorithmes de colonies de fourmis), et via un échange direct (très similaire à l’échange d’information dans l’optimisation par essaim particulière). Plus généralement, nous pensons que la théorie de l’auto-organisation donne des clefs pour concevoir les composants de base de métaheuristiques relevant de l’intelligence en essaim.

Ce document est structuré comme suit. Nous présentons d’abord dans la section 2 quelques métaheuristiques qui, comme l’optimisation par essaim particulière, gagnent à être placées dans le cadre de la théorie de l’auto-organisation. Nous décrivons ensuite dans la section 3 les principaux concepts mis en avant dans les théories de l’auto-organisation et de la programmation à mémoire adaptative, et nous dégagons les relations entre ces deux théories, du point de vue des métaheuristiques. Nous exposons dans la section 4 un exemple d’algorithme inspiré par cette démarche : l’algorithme CIAC, mentionné plus haut. En conclusion, nous tentons de dégager quelques concepts généraux susceptibles de constituer des “briques de base”, pour l’élaboration de nouvelles métaheuristiques relevant des théories de l’auto-organisation et de la programmation à mémoire adaptative.

## 2 Métaheuristiques inspirées de la biologie

Une partie de cette présentation est tirée de notre ouvrage [Dréo et al., 2003].

### 2.1 Optimisation par essaim particulière

L’optimisation par essaim particulière (“Particle Swarm Optimization”, *PSO*) [Kennedy and Eberhart, 1995, Eberhart et al., 2001] est issue d’une analogie avec les comportements collectifs de déplacements d’animaux. La métaphore a de plus été largement enrichie de notions de socio-psychologie. En effet, chez certains groupes d’animaux, comme les bancs de poissons, on peut observer des dynamiques de déplacements relativement complexes, alors que les individus eux-mêmes n’ont accès qu’à des informations limitées, comme la position et la vitesse de leurs plus proches voisins. On peut par exemple observer qu’un banc de poissons est capable d’éviter un prédateur : d’abord en se divisant en deux

groupes, puis en re-formant le banc originel, tout en maintenant la cohésion du banc.

Ces comportements collectifs s’inscrivent tout à fait dans la théorie de l’auto-organisation décrite plus loin. Pour résumer, chaque individu utilise l’information locale à laquelle il peut accéder sur le déplacement de ses plus proches voisins pour décider de son propre déplacement. Des règles très simples comme “rester proche des autres individus”, “aller dans la même direction”, “aller à la même vitesse” suffisent pour maintenir la cohésion du groupe tout entier, et pour susciter des comportements collectifs complexes et adaptés.

Les auteurs de la méthode d’optimisation par essaim particulaire se sont inspirés de ces comportements en s’appuyant sur la théorie de la socio-psychologie du traitement de l’information et des prises de décisions dans les groupes sociaux. Fait exceptionnel et remarquable, cette métaheuristique a été conçue d’emblée dans le cas continu, et c’est toujours dans ce domaine que se situent la majorité des applications à ce jour. La méthode en elle-même met en jeu des groupes de *particules* sous forme de vecteurs se déplaçant dans l’espace de recherche. Chaque particule est caractérisée par sa *position* et un vecteur de changement de position (appelé *vélocité*). À chaque itération, la particule se déplace. La socio-psychologie suggère que des individus se déplaçant sont influencés par leur comportement passé et par celui de leurs voisins (voisins dans le réseau social et non nécessairement dans l’espace). On tient donc compte, dans la mise à jour de la position de chaque particule, de la direction de son mouvement, sa vitesse, sa meilleure position et la meilleure position de ses voisins.

On trouvera un état de l’art complet sur l’optimisation par essaim particulaire et les concepts qui lui sont associés dans [Eberhart et al., 2001], ainsi qu’une synthèse en français dans [Clerc, 2002].

## 2.2 Algorithmes évolutionnaires

Les algorithmes évolutionnaires (AEs) sont des techniques de recherche inspirées par l’évolution biologique des espèces, apparues à la fin des années 1950 [Fraser, 1957]. Parmi plusieurs approches [Holland, 1962][Fogel et al., 1966] [Rechenberg, 1965], les algorithmes génétiques (AGs) en constituent certainement l’exemple le plus connu, à la suite de la parution en 1989 du célèbre livre de D. E. Goldberg [Goldberg, 1989] : *Genetic Algorithms in Search, Optimization and Machine Learning* (voir en [Goldberg, 1994] la traduction française).

Le principe d’un algorithme évolutionnaire se décrit simplement. Un ensemble de  $N$  points dans un espace de recherche, choisis a priori au hasard, constituent la *population* initiale; chaque individu  $x$  de la population possède une certaine performance, qui mesure son degré d’*adaptation* à l’objectif visé : dans le cas de la minimisation d’une fonction objectif  $f$ ,  $x$  est d’autant plus performant que  $f(x)$  est plus petit. Un AE consiste à faire évoluer progressivement, par *générations* successives, la composition de la population, en maintenant sa taille constante. Au cours des générations, l’objectif est d’améliorer globalement la performance des individus; on s’efforce d’obtenir un tel résultat en mimant les deux principaux mécanismes qui régissent l’évolution des êtres vivants, selon

la théorie de C. Darwin :

- la *sélection*, qui favorise la reproduction et la survie des individus les plus performants,
- et la *reproduction*, qui permet le brassage, la recombinaison et les variations des caractères héréditaires des parents, pour former des descendants aux potentialités nouvelles.

En pratique, une représentation doit être choisie pour les individus d’une population. Classiquement, un individu pourra être une liste d’entiers pour des problèmes combinatoires, un vecteur de nombres réels pour des problèmes numériques dans des espaces continus, une chaîne de nombres binaires pour des problèmes booléens, ou pourra même combiner ces représentations dans des structures complexes, si le besoin s’en fait sentir. Le passage d’une génération à la suivante se déroule en quatre phases : une phase de sélection, une phase de reproduction (ou de variation), une phase d’évaluation des performances et une phase de remplacement. La phase de sélection désigne les individus qui participent à la reproduction. Ils sont choisis, éventuellement à plusieurs reprises, a priori d’autant plus souvent qu’ils sont performants. Les individus sélectionnés sont ensuite disponibles pour la phase de reproduction. Celle-ci consiste à appliquer des opérateurs de variation sur des copies des individus sélectionnés pour en engendrer de nouveaux ; les opérateurs les plus utilisés sont le *croisement* (ou *recombinaison*), qui produit un ou deux descendants à partir de deux parents, et la *mutation*, qui produit un nouvel individu à partir d’un seul individu. La structure des opérateurs de variation dépend étroitement de la représentation choisie pour les individus. Les performances des nouveaux individus sont ensuite mesurées, durant la phase d’évaluation, à partir des objectifs fixés. Enfin, la phase de remplacement consiste à choisir les membres de la nouvelle génération : on peut, par exemple, remplacer les individus les moins performants de la population par les meilleurs individus produits, en nombre égal. L’algorithme est interrompu après un certain nombre de générations, selon un critère d’arrêt à préciser.

### 2.3 Systèmes immunitaires

Le terme “système immunitaire artificiel” (“Artificial Immune System”, *AIS*) s’applique à une vaste gamme de systèmes différents, notamment aux métaheuristiques d’optimisation inspirées du fonctionnement du système immunitaire des vertébrés. Un grand nombre de systèmes ont été conçus dans plusieurs domaines différents tels que la robotique, la détection d’anomalies ou l’optimisation (voir [De Castro and Von Zuben, 2000] pour un aperçu de différentes applications).

Le système immunitaire est responsable de la protection de l’organisme contre les “agressions” d’organismes extérieurs. La métaphore dont sont issus les algorithmes *AIS* met l’accent sur les aspects d’*apprentissage* et de *mémoire* du système immunitaire dit *adaptatif* (par opposition au système dit *inné*), notamment via la discrimination entre le *soi* et le *non-soi*. En effet, les cellules vivantes disposent sur leurs membranes de molécules spécifiques dites “anti-

gènes”. Chaque organisme dispose ainsi d’une identité unique, déterminée par l’ensemble des antigènes présents sur ses cellules. Les *lymphocytes* (un type de globule blanc) sont des cellules du système immunitaire qui possèdent des *récepteurs* capables de se lier spécifiquement à un antigène unique, permettant ainsi de reconnaître une cellule étrangère à l’organisme. Un lymphocyte ayant ainsi reconnu une cellule du non-soi va être stimulé à proliférer (en produisant des clones de lui-même) et à se différencier en cellule permettant de garder en mémoire l’antigène, ou en cellule permettant de combattre les agressions. Dans le premier cas, il sera capable de réagir plus rapidement à une nouvelle exposition à l’antigène : c’est le principe même de l’efficacité des vaccins. Dans le second cas, le combat contre les agressions est possible grâce à la production d’anticorps. La figure 1 résume ces principales étapes. Il faut également noter que la diversité des récepteurs dans l’ensemble de la population des lymphocytes est quant à elle produite par un mécanisme d’*hyper-mutation* des cellules clonées.

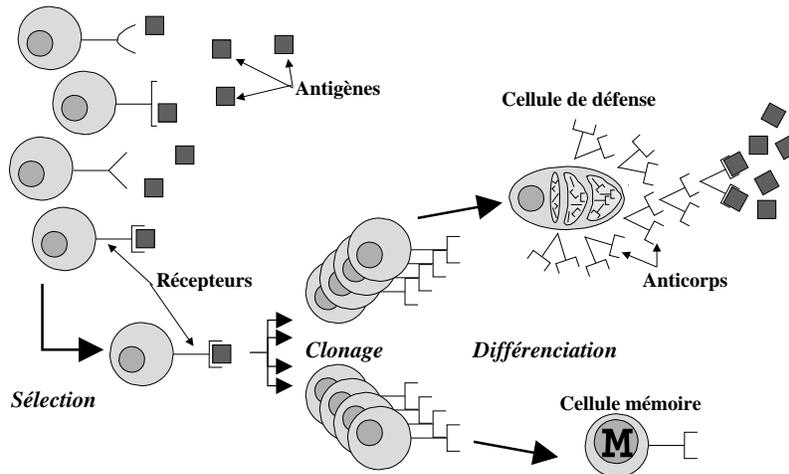


FIG. 1 – La sélection par clonage : des lymphocytes, présentant des récepteurs spécifiques d’un antigène, se différencient en cellule mémoire ou en cellule participant à la défense active de l’organisme par le biais d’anticorps.

Les principales idées utilisées pour la conception de la métaheuristique sont les sélections opérées sur les lymphocytes, accompagnées par les rétroactions positives permettant la multiplication et la mémoire du système. En effet, ces caractéristiques sont capitales pour maintenir les propriétés auto-organisées du système (voir le paragraphe 3.1 pour une définition plus précise de l’auto-organisation).

L’approche utilisée dans les algorithmes *AIS* est voisine de celle des algorithmes évolutionnaires, mais a également été comparée à celle des réseaux de neurones. On peut, dans le cadre de l’optimisation difficile, considérer les *AIS* comme une forme d’algorithme évolutionnaire présentant des opérateurs parti-

culiers. Pour opérer la sélection, on se fonde par exemple sur une mesure d'affinité entre le récepteur d'un lymphocyte et un antigène; la mutation s'opère quant à elle via un opérateur d'hyper-mutation directement issu de la métaphore.

Une description des fondements théoriques et de nombreuses applications des systèmes immunitaires artificiels peut être trouvée dans [De Castro and Von Zuben, 1999], [De Castro and Von Zuben, 2000] [Dasgupta and Attouh-Okine, 1997], et dans [Dasgupta, 1999].

## 2.4 Algorithmes de colonies de fourmis

### 2.4.1 Optimisation naturelle : pistes de phéromones

Les fourmis ont la particularité d'employer pour communiquer des substances volatiles appelées *phéromones*. Elles sont très sensibles à ces substances, qu'elles perçoivent grâce à des récepteurs situés dans leurs antennes. Ces substances sont nombreuses et varient selon les espèces. Les fourmis peuvent déposer des phéromones au sol, grâce à une glande située dans leur abdomen, et former ainsi des pistes odorantes, qui pourront être suivies par leurs congénères.

Les fourmis utilisent les pistes de phéromones pour marquer leur trajet, par exemple entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [Goss et al., 1989, Beckers et al., 1992], sans que les individus aient une vision *globale* du trajet.

En effet, comme l'illustre la figure 2, les fourmis le plus rapidement arrivées au nid, après avoir visité la source de nourriture, sont celles qui empruntent les deux branches les plus courtes. Ainsi, la *quantité* de phéromone présente sur le plus court trajet est légèrement plus importante que celle présente sur le chemin le plus long. Or, une piste présentant une plus grande concentration en phéromone est plus attirante pour les fourmis, elle a une *probabilité* plus grande d'être empruntée. La piste courte va alors être davantage renforcée que la longue, et, à terme, sera choisie par la grande majorité des fourmis.

### 2.4.2 Optimisation par colonies de fourmis et problème du voyageur de commerce

Le problème du voyageur de commerce consiste à chercher le trajet le plus court reliant  $n$  villes données, chaque ville ne devant être visitée qu'une seule fois. Le problème est plus généralement défini dans un graphe complètement connecté  $(N, A)$ , où les villes sont les noeuds  $N$  et les trajets entre ces villes, les arêtes  $A$ .

Dans l'algorithme du "Ant System" (*AS*) [Colormi et al., 1992], à chaque itération  $t$  ( $1 \leq t \leq t_{max}$ ), chaque fourmi  $k$  ( $k = 1, \dots, m$ ) parcourt le graphe et construit un trajet complet de  $n = |N|$  étapes (on note  $|N|$  le cardinal de l'ensemble  $N$ ). Pour chaque fourmi, le trajet entre une ville  $i$  et une ville  $j$  dépend de :

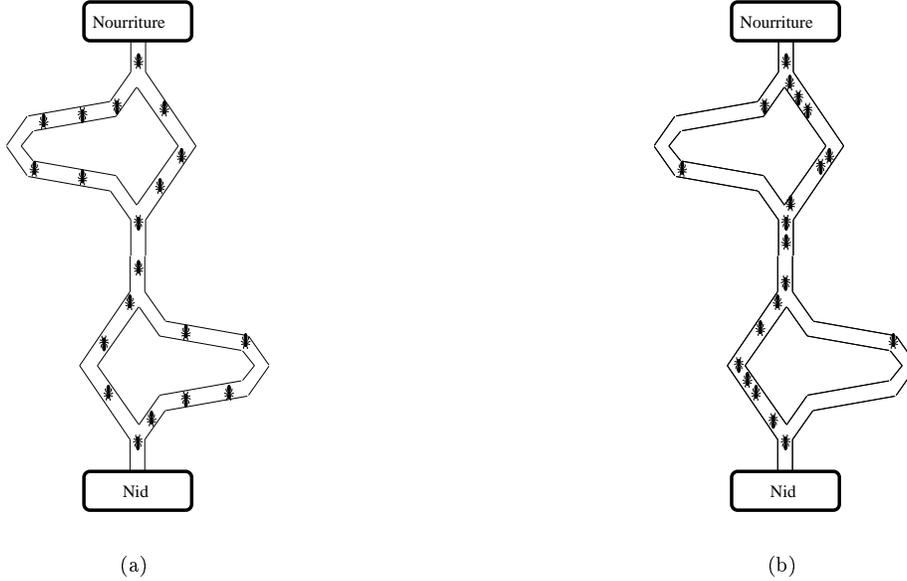


FIG. 2 – Expérience de sélection des branches les plus courtes par une colonie de fourmis : (a) au début de l’expérience, (b) à la fin de l’expérience.

1. la liste des villes déjà visitées, qui définit les mouvements possibles à chaque pas, quand la fourmi  $k$  est sur la ville  $i$  :  $J_i^k$ ,
2. l’inverse de la distance entre les villes :  $\eta_{ij} = \frac{1}{d_{ij}}$ , appelée *visibilité*. Cette information statique est utilisée pour diriger le choix des fourmis vers des villes proches,
3. la quantité de phéromone déposée sur l’arête reliant les deux villes, appelée *l’intensité de la piste*. Ce paramètre définit l’attractivité d’une partie du trajet global et change à chaque passage d’une fourmi. C’est, en quelque sorte, une mémoire globale du système, qui évolue par apprentissage.

La règle de déplacement (appelée “règle aléatoire de transition proportionnelle” par les auteurs [Bonabeau et al., 1999]) est la suivante :

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (1)$$

où  $\alpha$  et  $\beta$  sont deux paramètres contrôlant l’importance relative de l’*intensité* de la piste,  $\tau_{ij}(t)$ , et de la *visibilité*,  $\eta_{ij}$ . Avec  $\alpha = 0$ , seule la visibilité de la ville est prise en compte; la ville la plus proche est donc choisie à chaque pas. Au contraire, avec  $\beta = 0$ , seules les pistes de phéromone jouent. Pour éviter une sélection trop rapide d’un trajet, un compromis entre ces deux paramètres,

jouant sur l'importance relative de la *diversification* et de l'*intensification*, est nécessaire.

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromones  $\Delta\tau_{ij}^k(t)$  sur l'ensemble de son parcours, quantité qui dépend de la *qualité* de la solution trouvée :

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases} \quad (2)$$

où  $T^k(t)$  est le trajet effectué par la fourmi  $k$  à l'itération  $t$ ,  $L^k(t)$  la longueur du tour et  $Q$  un paramètre fixé.

L'algorithme ne serait pas complet sans le processus d'*évaporation* des pistes de phéromone. En effet, pour éviter d'être piégé dans des solutions sous-optimales, il est nécessaire de permettre au système "d'oublier" les mauvaises solutions. On contrebalance donc l'additivité des pistes par une décroissance constante des valeurs des arêtes à chaque itération. La règle de mise à jour des pistes est donc :

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3)$$

où  $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$  et  $m$  est le nombre de fourmis. La quantité initiale de phéromone sur chaque arête suit une distribution uniforme d'une petite quantité  $\tau_0 \geq 0$ .

La figure 3 présente un exemple simplifié de problème du voyageur de commerce optimisé par un algorithme AS.

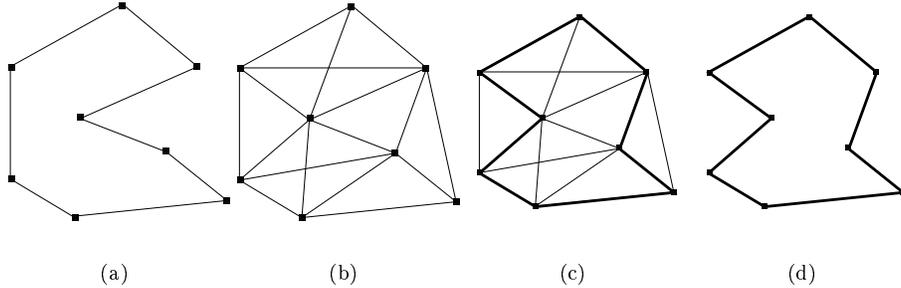


FIG. 3 – Le problème du voyageur de commerce optimisé par l'algorithme AS, les points représentent les villes et l'épaisseur des arêtes la quantité de phéromone déposée. (a) exemple de trajet construit par une fourmi, (b) au début du calcul, tous les chemins sont explorés, (c) le chemin le plus court est davantage renforcé que les autres, (d) l'évaporation permet d'éliminer les moins bonnes solutions.

### 2.4.3 Autres algorithmes

Il existe des métaheuristiques inspirées du comportement des insectes sociaux qui ne sont pas explicitement liées aux — plus connus — algorithmes de colonies de fourmis. En effet, les comportements de ces espèces sont complexes et riches ; et les caractéristiques auto-organisées que présentent ces groupes d’insectes sont une source d’inspiration intéressante.

Un bon exemple est celui d’un algorithme inspiré des modèles d’organisation du travail chez les fourmis [Campos et al., 2000, Cicirello and Smith, 2001, Nouyan, 2002]. Le partage des tâches chez certaines espèces fait apparaître des individus qui accomplissent des tâches spécifiques, ce qui permet d’éviter les coûts (en temps et en énergie par exemple) liés aux ré-attributions de tâches. Cependant, la spécialisation des individus n’est pas rigide, ce qui pourrait être préjudiciable à la colonie, mais elle s’adapte en fonction des nombreux stimulus internes et externes perçus par les individus [Wilson, 1984].

Des modèles de comportement ont été proposés pour expliquer ce phénomène [Bonabeau et al., 1996, Bonabeau et al., 1998, Theraulaz et al., 1998]. Ces modèles mettent en jeu pour chaque type de tâche des seuils de réponse représentant le niveau de spécialisation de l’individu. Ces seuils sont soit fixés [Bonabeau et al., 1998], soit mis à jour en fonction de l’accomplissement des tâches par les individus [Theraulaz et al., 1998].

Ces modèles ont inspiré des algorithmes pour l’allocation dynamique de tâches, où chaque machine se voit associée à un individu disposant d’un jeu de seuils de réponse  $\Theta_a$ , où  $\Theta_{a,j}$  représente le seuil de l’agent  $a$  pour la tâche  $j$ . La tâche  $j$  envoie aux agents un stimulus  $S_j$  représentant le temps d’attente de la tâche. L’agent  $a$  aura une probabilité d’effectuer la tâche  $j$  de :

$$P(\Theta_{a,j}, S_j) = \frac{S_j^2}{S_j^2 + \Theta_{a,j}^2}$$

L’algorithme dispose ensuite de règles de mise à jour des seuils et de règles de décision, au cas où deux agents tenteraient d’effectuer la même tâche (voir [Cicirello and Smith, 2001] pour plus de détails). Des améliorations à cet algorithme de base [Nouyan, 2002] ont permis d’augmenter sa rapidité et son efficacité sur des problèmes complexes.

## 3 Théories de l’auto-organisation et de la programmation à mémoire adaptative

### 3.1 Auto-organisation

L’*auto-organisation* est un phénomène décrit dans plusieurs disciplines, notamment en physique et en biologie. Une définition claire a été proposée [Camazine et al., 2000, p.8] :

L’auto-organisation est un processus dans lequel un *modèle* de niveau global *émerge* uniquement d’un grand nombre d’interactions

entre les composants de *bas niveau* du système. De plus, les règles spécifiant les interactions entre composants du système sont suivies en utilisant uniquement des informations locales, sans référence au *modèle* global.

Deux termes sont à préciser pour une bonne compréhension, “modèle” et “émerger”. Le mot *modèle* est une traduction approximative du mot anglais “pattern”, qui déborde la notion de modèle, et peut signifier aussi structure, configuration générale, forme, schéma, type. D’une manière générale, il s’applique à un “arrangement organisé d’objets dans l’espace ou le temps”. Une propriété *émergente* d’un système est quant à elle une caractéristique qui apparaît à l’improviste (sans avoir été *explicitement* déterminée), de par les interactions entre les composants de ce système.

La question cruciale est donc de comprendre comment les composants d’un système interagissent entre eux pour produire un modèle complexe (au sens relatif du terme, i.e. *plus* complexe que les composants eux-mêmes). Un certain nombre de phénomènes nécessaires ont été identifiés : ce sont les processus de *rétroaction* et la gestion des *flux d’informations*.

Les *rétroactions positives* sont des processus dont le résultat renforce l’action, par exemple par amplification, facilitation, auto-catalyse, etc. Les *rétroactions positives* sont capables d’amplifier les *fluctuations* du système, permettant la mise à jour d’informations peu apparentes. De tels processus peuvent facilement entraîner une divergence du système, s’ils ne sont pas maintenus sous contrôle par des *rétroactions négatives*, qui jouent ainsi le rôle de stabilisateurs du système. Lorsqu’ils sont couplés, de tels processus de *rétroaction* sont de puissants générateurs de modèles.

Dans le cadre de la biologie du comportement, il est intuitif que les interactions entre les composants d’un système vont très souvent mettre en jeu des processus de *communication*, de transfert d’informations entre individus. D’une manière générale, les individus peuvent communiquer, soit par le biais de “signaux”, c’est-à-dire en utilisant un moyen spécifique pour porter une information, soit par le biais d’“indices”, où l’information est portée accidentellement. De même, l’information peut provenir directement d’autres individus, ou bien passer par le biais de l’état d’un travail en cours. Cette deuxième possibilité d’échange d’informations, par le biais de modifications de l’environnement, se nomme la *stigmergie*.

D’une manière générale, tous ces processus sont plus ou moins interconnectés, permettant à un système constitué d’un grand nombre d’individus agissant de concert de résoudre des problèmes trop complexes pour un individu unique.

### 3.2 Programmation à mémoire adaptative

Le concept de programmation à mémoire adaptative (*PMA*) [Taillard, 1998, Taillard et al., 1998] est né de l’observation que les métaheuristiques récentes tendaient à devenir proches. Ainsi, du point de vue de la programmation à

mémoire adaptative, certaines métaheuristiques partagent maintenant une démarche commune, présentée dans l’algorithme 1.

---

**Algorithme 1** Démarche employée par un programme à mémoire adaptative.

---

1. Mémorisation d’un jeu de solutions ou une structure de données rassemblant les particularités des solutions produites par la recherche,
  2. construction d’une solution provisoire sur la base des données mémorisées,
  3. amélioration de la solution par un algorithme de recherche locale,
  4. mémorisation de la nouvelle solution ou de la structure de données associée.
- 

La notion de programmation à mémoire adaptative insiste sur trois concepts fondamentaux : la *mémoire*, l’*intensification* et la *diversification*.

La mémoire représente ici l’information récoltée par l’algorithme, sur laquelle il va s’appuyer pour effectuer sa recherche. La mémoire est présente sous de nombreuses formes possibles de structures mémorielles, de la plus simple (une population de points) à des structures plus complexes (les pistes de phéromones des algorithmes de colonies de fourmis). De notre point de vue, la mémoire joue le même rôle que la modélisation du problème employé lorsque les mécanismes de celui-ci sont bien connus, elle est un moyen de décrire la fonction objectif.

L’intensification (parfois appelée exploitation) consiste en l’utilisation des informations disponibles pour améliorer la pertinence de celles-ci. Du point de vue des métaheuristiques, il s’agit généralement tout simplement de recherche locale. Les algorithmes de recherche locale sont maintenant souvent employés en association avec d’autres métaheuristiques plus complexes, donnant lieu à des algorithmes *hybrides*. On rencontre ainsi souvent l’algorithme du “simplexe” de Nelder-Mead, mais des métaheuristiques plus complexes, comme la recherche avec tabous, sont parfois employés.

La diversification (ou exploration) est la recherche de nouvelles informations, afin d’augmenter la connaissance du problème. Ce sont souvent des méthodes stochastiques, et il est pour le moment difficile de dégager des idées générales, tant la diversité d’approche de cette composante des métaheuristiques est grande.

En pratique, les trois composantes de la PMA sont liées, et il est parfois difficile de distinguer où elles se situent dans les métaheuristiques proposées. De fait, les métaheuristiques tentent d’équilibrer la balance entre diversification et intensification, et bien souvent les améliorations à une métaheuristique existante consistent à faire pencher la balance dans un sens ou dans l’autre.

### 3.3 Bases communes

La théorie de l’auto-organisation nous renseigne sur la structure à employer pour concevoir des métaheuristiques flexibles et capables de s’adapter à un problème donné. En effet, la grande qualité de tels systèmes est de construire

des comportements complexes à partir de règles simples, en se fondant sur une architecture fortement *décentralisée* (on rencontre également le terme de *parallèle*), maintenant reconnue et utilisée pour sa flexibilité et son efficacité [Rudolph, 1992, Pardalos et al., 1996].

La programmation à mémoire adaptative nous renseigne quant à elle sur les méthodes employées par des métaheuristiques efficaces. Les auteurs insistent d’ailleurs sur les qualités de la PMA : parallélisme et flexibilité [Taillard et al., 1998, p. 17].

Les points communs entre ces deux théories sont donc nombreux et, du point de vue de la conception des métaheuristiques, il existe une relation simple entre elles : la PMA décrit le but à atteindre, et la théorie de l’auto-organisation un moyen pour atteindre ce but.

## 4 Exemple d’algorithme inspiré par l’auto-organisation

### 4.1 CIAC

Nous avons élaboré un algorithme de colonie de fourmis, se focalisant sur les principes de *communication*, mis en avant dans les précédentes sections [Dréo and Siarry, 2002]. Ainsi, une formalisation des échanges d’informations est proposée autour de la notion de canaux de communication. En effet, il existe plusieurs moyens de faire passer de l’information entre deux groupes d’individus, par exemple par dépôts de pistes de phéromone ou par échanges directs. On peut définir de la sorte différents *canaux de communication* représentant l’ensemble des caractéristiques du transport de l’information. Du point de vue des métaheuristiques, il y a trois caractéristiques principales (voir figure 4) :

**Portée** : le nombre d’individus mis en cause dans l’échange d’information. L’information peut, par exemple, être émise par un individu, et être perçue par plusieurs autres, et inversement.

**Mémoire** : la persistance de l’information dans le système. L’information peut rester un certain temps dans le système, ou n’être que transitoire.

**Intégrité** : les modifications engendrées par l’utilisation du canal de communication. L’information peut varier dans le temps, ou être biaisée lors de sa transmission.

De plus, l’information passant par un canal de communication peut bien sûr être n’importe quelle information d’intérêt, comme par exemple la valeur et/ou la position d’un point de l’espace de recherche.

L’algorithme *CIAC* (acronyme pour “Continuous Interacting Ant Colony”) utilise deux canaux de communication :

1. Le canal stigmergique (du nom donné aux processus de communication indirects employés par les fourmis, tels que les pistes) fait appel à des spots de phéromone, déposés sur l’espace de recherche, qui vont être plus ou moins attractifs pour les fourmis artificielles, selon leurs concentrations et leurs distances. Les caractéristiques du canal stigmergique sont donc les

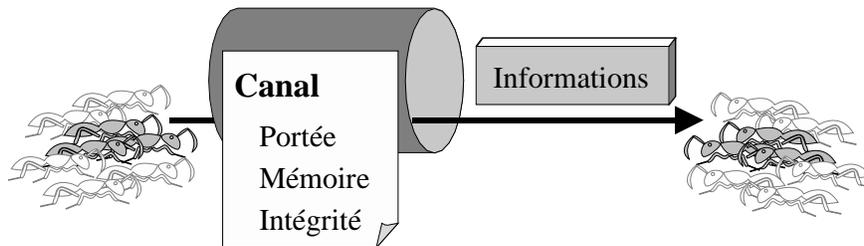


FIG. 4 – Un canal de communication structure les caractéristiques de la transmission de l’information : portée, mémoire et intégrité.

suivantes : la portée est à son maximum, toutes les fourmis pouvant potentiellement prendre en compte l’information ; il y a utilisation de mémoire, puisque les spots persistent sur l’espace de recherche ; enfin l’information évolue avec le temps, puisque les spots s’évaporent. L’information portée par un spot contient implicitement la position d’un point et explicitement la valeur de l’amélioration trouvée par la fourmi ayant déposé le spot.

2. Le canal direct est implémenté sous la forme d’échange de messages entre deux individus. Une fourmi artificielle possède une pile de messages reçus et peut en envoyer à une autre fourmi. La portée de ce canal est de un, puisque seule une fourmi reçoit le message ; la mémoire est implémentée dans la pile de messages que la fourmi mémorise ; enfin l’information (ici un couple position/valeur d’un point) n’est pas altérée au cours du temps.

L’algorithme possède certaines caractéristiques intéressantes, tirant parti des propriétés auto-organisées des algorithmes de colonies de fourmis, notamment une certaine capacité à osciller entre un processus d’intensification et un processus de diversification, quand les deux canaux de communication (stigmergique et direct) sont utilisés en synergie.

Cependant, les résultats ne sont comparables qu’à ceux produits par les autres algorithmes de colonies de fourmis pour le domaine continu, donc en deçà des meilleurs résultats procurés par d’autres métaheuristiques adaptées au cas continu.

## 4.2 Version hybride de CIAC

Les problèmes rencontrés par CIAC ont donné lieu à une hybridation avec l’algorithme de recherche locale de Nelder-Mead [Dréo and Siarry, 2003]. Cette variante de l’algorithme, appelée *HCIAC*, utilise donc deux canaux de communication, et exploite en outre une recherche locale et des processus décisionnels stochastiques. Ce dernier point est implémenté par l’utilisation de fonctions de type stimulus/réponse, qui permettent de définir un seuil de choix pour une action. Concrètement, on utilise une fonction sigmoïde  $p(x) = \frac{1}{1+e^{\delta\omega-\omega x}}$  pour tester un choix fonction d’un état  $x$  d’une fourmi, d’un seuil  $\delta$  déterminant la

position du point d’inflexion et d’une puissance  $\omega$  caractérisant l’inflexion de la sigmoïde. On tire un nombre aléatoire  $r$  dans une distribution uniforme, on a donc deux choix possibles :  $r < p(x)$  ou  $r \geq p(x)$ . Pour  $\delta = 0.5$  et  $\omega = +\infty$  on obtient par exemple un simple choix binaire. L’utilisation de ce type de fonction permet de s’affranchir d’un paramétrage délicat, par exemple en distribuant les seuils selon une loi normale sur toute la population. De même, on peut mettre en place, par ce biais, un système très simple d’apprentissage, en faisant varier les seuils.

L’implémentation de la recherche locale est, comme le suggèrent les théories de la PMA et de l’auto-organisation, fortement décentralisée. Ainsi, chaque fourmi dispose d’une “motivation” à effectuer une recherche locale à un instant donné, la probabilité d’effectuer la recherche locale étant évaluée à l’aide d’une fonction de type stimulus/réponse. La motivation de chaque fourmi va de même évoluer en fonction des informations sur la fonction objectif perçues localement par la fourmi (par l’intermédiaire des deux canaux de communication), ce qui permet à l’algorithme de s’adapter au problème rencontré.

L’hybridation a — comme souvent avec les algorithmes de colonies de fourmis — permis d’atteindre des résultats satisfaisants, comparables à ceux procurés par d’autres métaheuristiques concurrentes pour les problèmes continus.

## 5 Conclusion

Nous avons présenté différentes métaheuristiques fondées sur des métaphores biologiques, ayant en commun d’être inspirées de systèmes auto-organisés. Puis nous avons mis en relation les théories de l’auto-organisation et de la programmation à mémoire adaptative en vue de la conception des métaheuristiques. Enfin nous avons proposé un algorithme inspiré de cette démarche.

Les deux théories présentées permettent de mieux comprendre le fonctionnement des métaheuristiques existantes et d’orienter la conception de nouvelles métaheuristiques. Les concepts importants à retenir sont l’utilisation par les métaheuristiques modernes de la mémoire, de l’intensification et de la diversification, ainsi que l’aspect distribué et flexible de ces algorithmes. Cependant il faut souligner la difficulté de conception d’un système auto-organisé, ce qui explique que l’inspiration vienne de la biologie, où de tels systèmes sont relativement courants.

Les difficultés principales sont les suivantes :

- concevoir une structure mémorielle décrivant correctement le problème et dont il est aisé d’extraire l’information pertinente pour orienter la recherche,
- équilibrer la balance entre des méthodes d’intensification et de diversification,
- maintenir la flexibilité de l’algorithme, de façon à ce qu’il s’adapte au problème.

Les perspectives ouvertes par les points de vue des théories de la programmation à mémoire adaptative et de l’auto-organisation permettront peut-être la concep-

tion de nouvelles métaheuristiques. Il apparaît de plus en plus que les multiples variantes proposées dans la littérature, jouant sur des points cruciaux, comme la balance diversification/intensification, gagneraient à s'appuyer sur une analyse plus rigoureuse qu'une simple démarche empirique par essais et erreurs.

## Références

- [Beckers et al., 1992] Beckers, R., Deneubourg, J. L., and Goss, S. (1992). Trails and U-Turns in the Selection of a Path by the Ant *Lasius Niger*. *J. Theor. Biol.*, 159 :397–415.
- [Bonabeau et al., 1999] Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence, From Natural to Artificial Systems*. Oxford University Press.
- [Bonabeau et al., 1996] Bonabeau, E., Theraulaz, G., and Deneubourg, J.-L. (1996). Quantitative Study of the Fixed Threshold Model for the Regulation of Division of Labour in Insect Societies. In *Proceedings Roy. Soc. London B*, volume 263.
- [Bonabeau et al., 1998] Bonabeau, E., Theraulaz, G., and Deneubourg, J.-L. (1998). Fixed Response Thresholds and the Regulation of Division of Labor in Insect Societies. *Bulletin of Mathematical Biology*, (60) :753–807.
- [Camazine et al., 2000] Camazine, S., Deneubourg, J., Franks, N., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2000). *Self-Organization in Biological Systems*. Princeton University Press.
- [Campos et al., 2000] Campos, M., Bonabeau, E., Theraulaz, G., and Deneubourg, J.-L. (2000). Dynamic Scheduling and Division of Labor in Social Insects. In *Adaptive Behavior 2000*, pages 83–96.
- [Cicirello and Smith, 2001] Cicirello, V. and Smith, S. (2001). Wasp-like Agents for distributed Factory Coordination. Technical Report CMU-RI-TR-01-39, Robotics Institute, Carnegie Mellon University, Pittsburgh.
- [Clerc, 2002] Clerc, M. (2002). L'optimisation par essaim particulaire : principes, modèles et usages. *RSTI-TSI*, 21 :941–964.
- [Colorni et al., 1992] Colorni, A., Dorigo, M., and Maniezzo, V. (1992). Distributed Optimization by Ant Colonies. In Varela, F. and Bourgine, P., editors, *Proceedings of ECAL'91 - First European Conference on Artificial Life*, pages 134–142, Paris, France. Elsevier Publishing.
- [Dasgupta, 1999] Dasgupta, D. (1999). *Artificial Immune Systems and their applications*. Springer Verlag.
- [Dasgupta and Attoh-Okine, 1997] Dasgupta, D. and Attoh-Okine, N. (1997). Immune-based systems : A survey. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 369–374, Orlando. IEEE Press.

- [De Castro and Von Zuben, 1999] De Castro, L. and Von Zuben, F. (1999). Artificial Immune Systems : Part I : Basic Theory and Applications. Technical Report TR-DCA 01/99, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil.
- [De Castro and Von Zuben, 2000] De Castro, L. and Von Zuben, F. (2000). Artificial Immune Systems : Part II- A Survey of Applications. Technical Report DCA-RT 02/00, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil.
- [Dréo et al., 2003] Dréo, J., Pétrowski, A., Siarry, P., and Taillard, E. D. (2003). *Métaheuristiques pour l'optimisation difficile*. Eyrolles.
- [Dréo and Siarry, 2002] Dréo, J. and Siarry, P. (2002). A New Ant Colony Algorithm Using the Heterarchical Concept Aimed at Optimization of Multimimima Continuous Functions. In Dorigo, M., Di Caro, G., and Sampels, M., editors, *Proceedings of the Third International Workshop on Ant Algorithms (ANTS'2002)*, volume 2463 of *Lecture Notes in Computer Science*, pages 216–221, Brussels, Belgium. Springer Verlag.
- [Dréo and Siarry, 2003] Dréo, J. and Siarry, P. (2003). Un algorithme de colonie de fourmis en variables continues hybridé avec un algorithme de recherche locale. In *5ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2003)*, Avignon, France.
- [Eberhart et al., 2001] Eberhart, R., Kennedy, J., and Shi, Y. (2001). *Swarm Intelligence*. Evolutionary Computation. Morgan Kaufmann.
- [Fogel et al., 1966] Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. Wiley.
- [Fraser, 1957] Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Sciences*, 10 :484–491.
- [Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Wesley.
- [Goldberg, 1994] Goldberg, D. E. (1994). *Algorithmes génétiques. Exploration, optimisation et apprentissage automatique*. Addison-Wesley France.
- [Goss et al., 1989] Goss, S., Aron, S., Deneubourg, J. L., and Pasteels J. M. (1989). Self-Organized Shortcuts in the Argentine Ant. *Naturwissenschaften*, 76 :579–581.
- [Holland, 1962] Holland, J. H. (1962). Outline for logical theory of adaptive systems. *J. Assoc. Comput. Mach.*, 3 :297–314.
- [Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. In *Proc. IEEE Int. Conf. on Neural Networks*, volume IV, pages 1942–1948, Piscataway, NJ : IEEE Service Center.
- [Nouyan, 2002] Nouyan, S. (2002). Agent-Based Approach to Dynamic task Allocation. In Dorigo, M., Di Caro, G., and Sampels, M., editors, *Proceedings of*

- the Third International Workshop on Ant Algorithms (ANTS'2002)*, volume 2463 of *Lecture Notes in Computer Science*, pages 28–39, Brussels, Belgium. Springer Verlag.
- [Pardalos et al., 1996] Pardalos, P., Xue, G., and Panagiotopoulos, P. D. (1996). Parallel Algorithms for Global Optimization Problems. In *Solving Combinatorial Optimization Problems in Parallel*, pages 232–247.
- [Rechenberg, 1965] Rechenberg, I. (1965). *Cybernetic Solution Path of an Experimental Problem*. Royal Aircraft Establishment Library Translation.
- [Rudolph, 1992] Rudolph, G. (1992). *Parallel approaches to stochastic global optimization*, pages 256–267. IOS Press, Amsterdam.
- [Taillard, 1998] Taillard, E. D. (1998). *Programmation à mémoire adaptative et algorithmes pseudo-gloutons : nouvelles perspectives pour les méta-heuristiques*. Thèse d’habilitation à diriger les recherches, Université de Versailles Saint Quentin en Yvelines, France.
- [Taillard et al., 1998] Taillard, E. D., Gambardella, L. M., Gendreau, M., and Potvin, J.-Y. (1998). Adaptive Memory Programming : A Unified View of Meta-Heuristics. *European Journal of Operational Research*, 135(1) :1–16.
- [Theraulaz et al., 1998] Theraulaz, G., Bonabeau, E., and Deneubourg, J.-L. (1998). Response Threshold Reinforcement and Division of Labour in Insect Societies. In *Proceedings of the Royal Society of London B*, volume 265, pages 327–335.
- [Wilson, 1984] Wilson, E. (1984). The relation between Caste Ratios and Division of Labour in the Ant Genus *Pheidole* (Hymenoptera : Formicidae). *Behav. Ecol. Sociobiol.*, 16 :89–98.

## Sites internet

- **Livre sur les métaheuristiques :**  
<http://www.eyrolles.com/php.informatique/Ouvrages/9782212113686.php3>
- **Livre sur la théorie de l’auto-organisation :**  
<http://www.scottcamazine.com/personal/selforganization/S0Main.html>
- **Thèse sur la programmation à mémoire adaptative :**  
<http://ina.eivd.ch/collaborateurs/etd/articles.dir/IDSIA-52-98.ps>
- **Méthode des essais particuliers :**  
<http://www.particleswarm.net/>
- **Groupe de travail sur les algorithmes évolutionnaires :**  
<http://www.afia.polytechnique.fr/node.php?lang=fr&node=285>
- **Méthode des colonies de fourmis :**  
<http://iridia.ulb.ac.be/~mdorigo/ACO/>