
Métaheuristiques pour l'optimisation et auto-organisation dans les systèmes biologiques

Johann Dréo — Patrick Siarry

Université de Paris XII Val-de-Marne

Laboratoire d'Étude et de Recherche en Instrumentation, Signaux et Systèmes (LERISS, EA 412)

61, avenue du Général de Gaulle, 94010 Créteil

{dreo,siarry}@univ-paris12.fr

RÉSUMÉ. Cet article propose de mettre en relation les concepts d'auto-organisation et de programmation à mémoire adaptative (PMA) dans le domaine des métaheuristiques d'optimisation. L'auto-organisation est décrite dans le cadre de la biologie et la PMA est définie. Les principales catégories de métaheuristiques utilisant ces deux notions sont décrites et les utilisations de l'auto-organisation et de la PMA sont mises en valeur.

ABSTRACT. This paper proposes to link the concepts of self-organization and adaptive memory programming (AMP) in the field of optimization metaheuristics. The self-organization is described within the framework of biology and the AMP is defined. The main categories of the metaheuristics using these two concepts are described and uses of the self-organization and the AMP are highlighted.

MOTS-CLÉS : optimisation, métaheuristique, intelligence en essaim, auto-organisation, programmation à mémoire adaptative.

KEYWORDS: optimization, metaheuristic, swarm intelligence, self-organization, adaptive memory programming.

1. Introduction

La biologie est la source d'inspiration de nombreuses métaheuristiques. Ainsi, les théories de l'évolution ont inspiré les algorithmes évolutionnaires [GOL 89, GOL 94], les phénomènes de suivi de piste chez les fourmis ont conduit à l'élaboration des algorithmes de colonies de fourmis [COL 92, BON 99], l'étude de l'organisation de groupes d'animaux a donné naissance aux méthodes d'optimisation par essais particuliers [KEN 95, KEN 01]. 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 [DEC 99], des algorithmes pour l'allocation dynamique de tâches [CIC 01], s'appuyant sur des modèles d'organisation du travail chez les fourmis, des algorithmes de classification suggérés par les essais d'insectes [AUP 03].

Une contribution importante de la biologie dans ce domaine vient de la *théorie de l'auto-organisation* [CAM 00a, p.8], qui permet d'analyser les propriétés de plusieurs métaheuristiques issues des métaphores biologiques. Cette théorie (notamment étudiée en dehors de la biologie [NIC 77]) 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, 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 particulier).

Nous proposons de mettre la théorie de l'auto-organisation en relation avec le concept de *programmation à mémoire adaptative* [TAI 98a], 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.

Plus généralement, nous pensons que la théorie de l'auto-organisation combinée à la programmation à mémoire adaptative donne des clefs pour concevoir les composants de base de métaheuristiques relevant de l'intelligence en essaim.

Cet article est structuré comme suit. Nous décrivons d'abord dans la section 2 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 présentons ensuite dans la section 3 quelques métaheuristiques qui gagnent à être placées dans le cadre de la théorie de l'auto-organisation. Nous exposons en particulier dans la section 3.6 un exemple d'algorithme inspiré par cette démarche : l'algorithme CIAC. 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. Théories de l'auto-organisation et de la programmation à mémoire adaptative

2.1. Auto-organisation

L'*auto-organisation* est un phénomène décrit dans plusieurs disciplines, notamment en physique [PRI 71, NIC 77] et en biologie. Étant donné que les métaheuristiques présentées dans cet article sont inspirées de phénomènes biologiques, nous avons choisi de considérer une définition dans ce cadre. Une définition claire a été proposée [CAM 00a, p.8] :

L'auto-organisation caractérise un processus au cours duquel une *structure* émerge au niveau global uniquement d'un grand nombre d'interactions entre les composants de *niveau local* du système. De plus, les règles spécifiant les interactions entre composants du système sont suivies en utilisant uniquement des informations locales, sans référence à la *structure* globale.

Deux termes sont à préciser pour une bonne compréhension, "structure" et "émerger". Le mot *structure* est une traduction approximative du mot anglais "pattern", qui déborde la notion de structure, et peut signifier aussi configuration générale, forme, schéma, type [MEI 82]. 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. Nous parlerons donc d'émergence pour souligner le caractère non-déterminé d'une propriété, sans faire référence au fait que cette propriété soit organisée en *structure* ni qu'elle soit située à un niveau différent des interactions.

La question cruciale est donc de comprendre comment les composants d'un système interagissent entre eux pour produire une structure 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 [CAM 00a].

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 uti-

lisant un moyen spécifique pour porter une information, soit par le biais d’“indices”, où l’information est portée accidentellement [SEE 89]. 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* [GRA 59, THE 95].

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.

2.2. Programmation à mémoire adaptative

Le concept de programmation à mémoire adaptative (*PMA*) [TAI 98a, TAI 98b] 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*. Dans la littérature des algorithmes évolutionnaires, ces deux dernières notions sont souvent désignées par les termes *exploitation* et *exploration*, ayant un sens similaire. Nous avons choisi ici d’utiliser les termes de la définition originale de la programmation à mémoire adaptative, employés plus généralement dans la littérature de la recherche avec tabous ou du recuit simulé.

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 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 est une modélisation des solutions, elle est un moyen de décrire la fonction objectif. Cette modélisation s’opère par une distribution sur l’espace de recherche, biaisée vers les meilleures solutions, qui évolue vers les optima du problème. Cette mémoire peut être définie comme globale (par rapport au problème dans son ensemble) ou inter-individuelle (d’une solution relativement à une autre).

L’intensification consiste en l’utilisation des informations disponibles pour améliorer la pertinence de celles-ci. Du point de vue des métaheuristiques, il s’agit géné-

ralement 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* [TAL 02]. On rencontre ainsi souvent l'algorithme du "simplexe" de Nelder-Mead [NEL 65], mais des métaheuristiques plus complexes, comme la recherche avec tabous, sont parfois employées.

La diversification 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'approches 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 d'une métaheuristique existante consistent à faire pencher la balance dans un sens ou dans l'autre.

Le concept de programmation à mémoire adaptative se veut une forme de généralisation du mode de fonctionnement des métaheuristiques. Certaines métaheuristiques ont un mode de fonctionnement qui semble d'emblée très proche de la PMA, c'est le cas par exemple de la méthode GRASP ("Greedy Randomized Adaptive Search Procedure") [RES 00] ou encore des algorithmes à estimation de distribution (EDA) [LAR 02]. Cependant, la PMA propose une approche généraliste, sans entrer dans les détails de l'implémentation, qui induisent souvent des a priori sur la façon d'aborder tel ou tel problème. C'est le cas par exemple des EDA où la phase de diversification utilise un tirage aléatoire dans une loi donnée, il y a donc un a priori fort sur le type de loi utilisée, qui dépend en pratique du problème abordé. La programmation à mémoire adaptative tente d'unifier différentes métaheuristiques, et déborde le cadre des algorithmes utilisant l'auto-organisation décrits dans cet article.

2.3. Bases communes

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* ou *distribuée*), maintenant reconnue et utilisée pour sa flexibilité et son efficacité [RUD 92, PAR 96].

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 [TAI 98b].

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. Ainsi, une métaheuristique efficace devrait, selon la programmation à mémoire adaptative, mettre en place des mécanismes de mémoire, d'intensification et de diversification, reste la question des moyens à utiliser pour mettre en place ces mécanismes. L'auto-organisation propose un modèle de réalisation : un algorithme à base de population définissant des interactions simples au niveau local, permettant l'émergence d'un comportement complexe au niveau global.

3. Métaheuristiques inspirées de la biologie

Cette section présente quelques métaheuristiques inspirées de la biologie et utilisant des phénomènes d'auto-organisation. Nous avons délibérément choisi de restreindre la liste des métaheuristiques présentées à des classes d'algorithmes parmi les plus connues, afin d'éviter une énumération peu pertinente et forcément incomplète. De plus, certaines classes de métaheuristiques se recoupent entre elles, la classification proposée ici est celle la plus communément admise.

Une partie de cette présentation est tirée de notre ouvrage [DRE 03a].

3.1. *Optimisation par essaim particulaire*

L'optimisation par essaim particulaire ("Particle Swarm Optimization", *PSO*) [KEN 95, KEN 01] 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 reformant 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 précédemment. 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 [KEN 01]. 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* ou *vecteur vitesse*). 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.

Ici, les rétroactions positives sont mises en place au niveau de l'attraction des particules les unes pour les autres. Les limitations de déplacement de chaque particule ou les éventuelles ré-initialisations entre deux itérations forment les rétroactions négatives. La mémoire est structurée au niveau local, entre particules voisines, à chaque itération chaque particule n'évolue qu'en fonction de ses proches voisins, et non pas selon l'état global de la population à l'itération précédente.

On trouvera un état de l'art complet sur l'optimisation par essaim particulaire et les concepts qui lui sont associés dans [KEN 01], ainsi qu'une synthèse en français dans [CLE 02].

3.2. 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 [FRA 57]. Parmi plusieurs approches [HOL 62, FOG 66, REC 65], 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 [GOL 89] : *Genetic Algorithms in Search, Optimization and Machine Learning* (voir en [GOL 94] 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, au besoin, combiner ces représentations dans des structures complexes. 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.

Dans cette famille de métaheuristiques [BAE 00a, BAE 00b], les rétroactions sont parfois difficiles à cerner, tant les variantes sont nombreuses. D'une façon générale, les rétroactions positives sont implémentées sous la forme d'opérateurs de type sélection, alors que les rétroactions négatives sont typiquement mises en place par des opérateurs de mutation. La mémoire est située au niveau local, l'évolution de chaque individu d'une itération à l'autre étant liée à l'évolution des individus voisins.

3.3. *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 [DEC 00] 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 "antigènes". Chaque organisme possède

ainsi 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 incité 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.

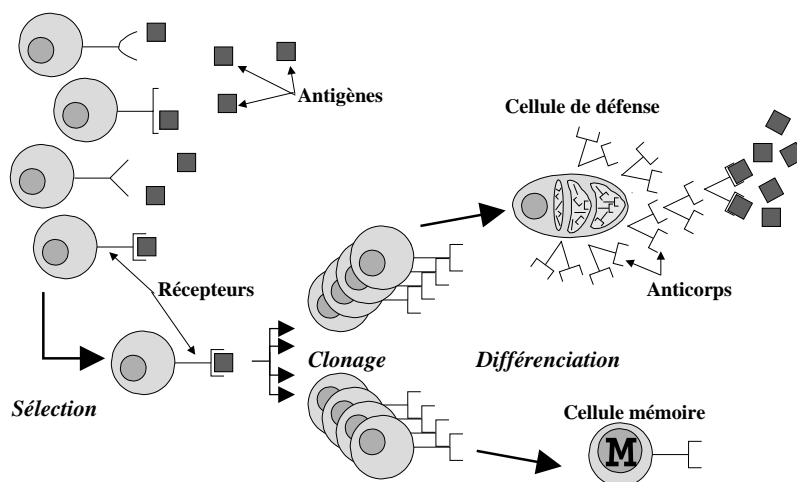


Figure 1. La sélection par clonage : des lymphocytes, présentant des récepteurs spécifiques d'un antigène, se différencient en cellule mémoire ou en cellule participant à la défense active de l'organisme par le biais d'anticorps.

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.

L'approche utilisée dans les algorithmes *AIS* est très voisine de celle des algorithmes évolutionnaires [GAS 99], mais a également été comparée à celle des réseaux de neurones [DAS 97a]. On peut, dans le cadre de l'optimisation difficile, considérer les *AIS* comme une forme d'algorithme évolutionnaire présentant des opérateurs particuliers. Pour opérer la sélection (exemple de rétroaction positive, voir la section précédente), on se fonde par exemple sur une mesure d'affinité entre le récepteur d'un lym-

phocyte et un antigène ; la mutation (exemple de rétroaction négative) 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 [DEC 99], [DEC 00, DAS 97b], et dans [DAS 99].

3.4. Algorithmes de colonies de fourmis

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 [GOS 89, BEC 92], sans que les individus aient une vision *globale* du trajet.

Le premier algorithme de colonies de fourmis [COL 92] a été conçu pour optimiser le problème du voyageur de commerce. Ce problème consiste à chercher le trajet le plus court reliant n villes données, chaque ville ne devant être visitée qu'une seule fois.

Dans l'algorithme du "Ant System" (AS) [COL 92], à chaque itération, chaque fourmi parcourt le graphe et construit un trajet complet de n étapes. Pour chaque fourmi, le trajet entre une ville i et une ville j dépend de :

- 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 : $\eta_{ij} = \frac{1}{d_{ij}}$, appelé *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 [BON 99]) contrôle l'importance relative de l'*intensité* de la piste et de la *visibilité*.

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromone sur l'ensemble de son parcours, quantité qui dépend de la *qualité* de la solution trouvée.

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 figure 2 présente un exemple simplifié de problème du voyageur de commerce optimisé par un algorithme AS.

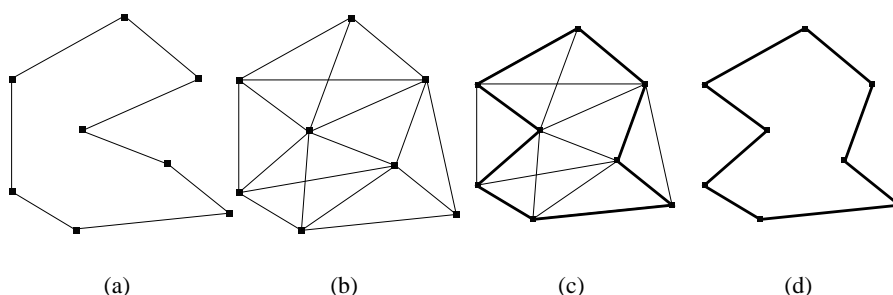


Figure 2. Le problème du voyageur de commerce optimisé par l'algorithme AS, les points représentent les villes et l'épaisseur des arêtes la quantité de phéromone déposée. (a) exemple de trajet construit par une fourmi, (b) au début du calcul, tous les chemins sont explorés, (c) le chemin le plus court est davantage renforcé que les autres, (d) l'évaporation permet d'éliminer les moins bonnes solutions.

Dans ces algorithmes, les rétroactions positives sont formées par l'attrait des fourmis pour les pistes les plus concentrées. Le principe de l'évaporation concrétise les rétroactions négatives. La mémoire est mise en place au niveau global, par les pistes de phéromones.

Les algorithmes de colonies de fourmis sont pour l'essentiel appliqués à des problèmes combinatoires, notamment du type du voyageur de commerce [GAM 95, DOR 96, DOR 97b, DOR 97a]. Cependant, devant le succès rencontré par ces algorithmes, d'autres pistes commencent à être explorées : par exemple, l'utilisation de ces algorithmes dans des problèmes *continus* [DRE 04b] et/ou *dynamiques* [DIC 98], ou encore l'exploitation de ce type d'algorithmes dans un cadre d'*intelligence en essaim* [BON 99] et avec d'autres métaheuristiques [MON 99, MON 00a, DRE 03b, DRE 04a]. Le livre de référence sur les algorithmes de colonies de fourmis [BON 99] insiste sur les aspects biologiques du domaine et présente un grand nombre d'algorithmes. Un autre livre, plus récent, se restreint aux algorithmes de colonies de fourmis [DOR 04].

3.5. Autre algorithme inspiré des insectes sociaux

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 [CAM 00b, CIC 01, NOU 02]. 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 [WIL 84].

Des modèles de comportement ont été proposés pour expliquer ce phénomène [BON 96, BON 98, THE 98]. 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 [BON 98], soit mis à jour en fonction de l'accomplissement des tâches par les individus [THE 98].

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 Θ_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 [CIC 01] pour plus de détails). Des améliorations à cet algorithme de base [NOU 02] ont permis d'augmenter sa rapidité et son efficacité sur des problèmes d'allocation dynamique de tâches.

Ici, les rétroactions positives sont liées à la spécialisation des individus, alors que les rétroactions négatives sont mises en place sous la forme des changements de tâches. La mémoire est locale, au niveau des seuils des différents individus.

3.6. Algorithme de colonie de fourmis auto-organisé

Nous avons élaboré un algorithme de colonie de fourmis, se focalisant sur les principes de *communication*, mis en avant plus haut [DRE 02]. Cet algorithme est apparenté aux algorithmes de colonies de fourmis pour les problèmes *continus* [BIL 95,

WOD 97, MAT 00, MON 00b], qui manipulent une population à la manière des algorithmes évolutionnaires [LIN 02].

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 [HOL 90, DRE 01]. On peut définir 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 : la *portée* (le nombre d'individus mis en cause dans l'échange d'information), la *mémoire* (la persistance de l'information dans le système) et l'*intégrité* (les modifications engendrées par l'utilisation du canal de communication). De plus, l'information passant par un canal de communication peut 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 tels que les pistes employés par les fourmis) 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.

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.

Le manque d'efficacité de *CIAC* dans le domaine de l'intensification a donné lieu à une hybridation [DRE 03c] avec l'algorithme de recherche locale de Nelder-Mead [NEL 65]. Cette variante de l'algorithme, appelée *HCIAC* ("Hybrid Continuous Interacting Ant Colony"), présente une meilleure efficacité. Elle utilise deux canaux de communication, et exploite en outre une recherche locale et des processus décisionnels stochastiques, implémentés à l'aide de fonctions de type stimulus/réponse, qui permettent de définir un seuil de choix pour une action. L'implémentation de la recherche locale est, comme le suggèrent les théories de la PMA et de l'auto-organisation, fortement décentralisée.

Les rétroactions positives sont ici mises en place par l'attrait des fourmis pour les spots de phéromone existants et les échanges de messages, les rétroactions négatives par l'évaporation et les choix stochastiques. La mémoire est à la fois globale (spots de phéromone) et locale (messages entre deux fourmis, seuils de choix).

4. Conclusion

Les deux théories présentées devraient permettre 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 métaheu-

ristiques 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 mémoire échantillonnant correctement le problème et dont il est aisé d'extraire l'information pertinente pour orienter la recherche,
- équilibrer la balance entre des techniques 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 conception 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.

5. Bibliographie

- [AUP 03] AUPETIT S., MONMARCHÉ N., SLIMANE M., GUINOT C., VENTURINI G., « Clustering and Dynamic Data Visualization with Artificial Flying Insect », *Genetic and Evolutionary Computation Conference (GECCO)*, 2003.
- [BAE 00a] BAECK T., FOGEL D. B., MICHALEWICZ Z., *Evolutionary Computation 1 : Basic Algorithms and Operators*, Institute of Physics Publishing, 2000.
- [BAE 00b] BAECK T., FOGEL D. B., MICHALEWICZ Z., *Evolutionary Computation 2 : Advanced Algorithms and Operators*, Institute of Physics Publishing, 2000.
- [BEC 92] BECKERS R., DENEUBOURG J. L., GOSS S., « Trails and U-Turns in the Selection of a Path by the Ant *Lasius Niger* », *J. Theor. Biol.*, vol. 159, 1992, p. 397–415.
- [BIL 95] BILCHEV G., PARMEE I., « The Ant Colony Metaphor for Searching Continuous Design Spaces », *Lecture Notes in Computer Science*, vol. 993, 1995, p. 25–39.
- [BON 96] BONABEAU E., THERAULAZ G., DENEUBOURG J.-L., « Quantitative Study of the Fixed Threshold Model for the Regulation of Division of Labour in Insect Societies », *Proceedings Roy. Soc. London B*, vol. 263, 1996.
- [BON 98] BONABEAU E., THERAULAZ G., DENEUBOURG J.-L., « Fixed Response Thresholds and the Regulation of Division of Labor in Insect Societies », *Bulletin of Mathematical Biology*, n° 60, 1998, p. 753–807.
- [BON 99] BONABEAU E., DORIGO M., THERAULAZ G., *Swarm Intelligence, From Natural to Artificial Systems*, Oxford University Press, 1999.
- [CAM 00a] CAMAZINE S., DENEUBOURG J., FRANKS N., SNEYD J., THERAULAZ G., BONABEAU E., *Self-Organization in Biological Systems*, Princeton University Press, 2000.
- [CAM 00b] CAMPOS M., BONABEAU E., THERAULAZ G., DENEUBOURG J.-L., « Dynamic Scheduling and Division of Labor in Social Insects », *Adaptive Behavior*, 2000, p. 83–96.
- [CIC 01] CICIRELLO V., SMITH S., « Wasp-like Agents for distributed Factory Coordination », rapport n° CMU-RI-TR-01-39, décembre 2001, Robotics Institute, Carnegie Mellon University, Pittsburgh.

- [CLE 02] CLERC M., « L'optimisation par essaim particulaire : principes, modèles et usages », *Technique et Science Informatiques*, vol. 21, 2002, p. 941–964.
- [COL 92] COLORNI A., DORIGO M., MANIEZZO V., « Distributed Optimization by Ant Colonies », VARELA F., BOURGINE P., Eds., *Proceedings of ECAL'91 - First European Conference on Artificial Life*, Paris, France, 1992, Elsevier Publishing, p. 134–142.
- [DAS 97a] DASGUPTA D., « Artificial Neural Networks Vs. Artificial Immune Systems », *Sixth International Conference on Intelligent Systems*, Boston, juin 1997.
- [DAS 97b] DASGUPTA D., ATTOH-OKINE N., « Immune-based systems : A survey », *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, Orlando, octobre 1997, IEEE Press, p. 369–374.
- [DAS 99] DASGUPTA D., *Artificial Immune Systems and their applications*, Springer Verlag, 1999.
- [DEC 99] DE CASTRO L., VON ZUBEN F., « Artificial Immune Systems : Part I : Basic Theory and Applications », rapport n° TR-DCA 01/99, décembre 1999, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil.
- [DEC 00] DE CASTRO L., VON ZUBEN F., « Artificial Immune Systems : Part II - A Survey of Applications », rapport n° DCA-RT 02/00, février 2000, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil.
- [DIC 98] DI CARO G., DORIGO M., « AntNet : Distributed stigmergic control for communications networks », *Journal of Artificial Intelligence Research*, vol. 9, 1998, p. 317–365.
- [DOR 96] DORIGO M., MANIEZZO V., COLORNI A., « The Ant System : Optimization by a Colony of Cooperating Agents », *IEEE Trans. Syst. Man Cybern.*, vol. B, n° 26, 1996, p. 29–41.
- [DOR 97a] DORIGO M., GAMBARDELLA L. M., « Ant Colonies for the Traveling Salesman Problem », *BioSystems*, vol. 43, 1997, p. 73–81.
- [DOR 97b] DORIGO M., GAMBARDELLA L. M., « Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem », *IEEE Trans. Evol. Comp.*, vol. 1, 1997, p. 53–66.
- [DOR 04] DORIGO M., STÜTZLE T., *Ant Colony Optimization*, MIT Press, 2004.
- [DRE 01] DREO J., « Modélisation de la mobilisation chez les fourmis », Mémoire de DEA, Université Paris7 & Université Libre de Bruxelles, 2001.
- [DRE 02] DREO J., SIARRY P., « A New Ant Colony Algorithm Using the Heterarchical Concept Aimed at Optimization of Multim minima Continuous Functions », DORIGO M., DI CARO G., SAMPELS M., Eds., *Proceedings of the Third International Workshop on Ant Algorithms (ANTS'2002)*, vol. 2463 de *Lecture Notes in Computer Science*, Brussels, Belgium, septembre 2002, Springer Verlag, p. 216–221.
- [DRE 03a] DREO J., PÉTROWSKI A., SIARRY P., TAILLARD E. D., *Métaheuristiques pour l'optimisation difficile*, Eyrolles, septembre 2003.
- [DRE 03b] DREO J., SIARRY P., « Diverses techniques d'optimisation inspirées de la théorie de l'auto-organisation dans les systèmes biologiques », *Journée optimisation par essaim particulaire (OEP'2003)*, 2 octobre 2003.

- [DRE 03c] DREO J., SIARRY P., « Un algorithme de colonie de fourmis en variables continues hybridé avec un algorithme de recherche locale », *5ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2003)*, Avignon, France, février 2003.
- [DRE 04a] DREO J., SIARRY P., « Algorithmes à estimation de distribution et colonies de fourmis », *11ème journée évolutionnaire (JET11)*, 12 mars 2004.
- [DRE 04b] DREO J., SIARRY P., « Continuous Interacting Ant Colony Algorithm Based on Dense Heterarchy », *Future Generation Computer Systems*, vol. 20, n° 5, 2004, p. 841–856.
- [FOG 66] FOGEL L. J., OWENS A. J., WALSH M. J., *Artificial Intelligence through Simulated Evolution*, Wiley, 1966.
- [FRA 57] FRASER A. S., « Simulation of genetic systems by automatic digital computers », *Australian Journal of Biological Sciences*, vol. 10, 1957, p. 484–491.
- [GAM 95] GAMBARDELLA L. M., DORIGO M., « Ant-Q : A Reinforcement Learning Approach to the Travelling Salesman Problem », *Proceedings Twelfth International Conference on Machine Learning*, vol. ML-95, Palo Alto, 1995, Morgan Kaufmann, p. 252–260.
- [GAS 99] GASPAR A., COLLARD P., « From GAs to artificial immune systems : improving adaptation in time dependent optimization », ANGELINE P., MICHALEWICZ Z., SCHONAUER M., YAO X., ZALZALA A., Eds., *Proceedings of the Congress on Evolutionary Computation*, vol. 3, Washington D.C., 1999, p. 1859–1866.
- [GOL 89] GOLDBERG D. E., *Genetic Algorithms in Search, Optimization and Machine learning*, Addison-Wesley, 1989.
- [GOL 94] GOLDBERG D. E., *Algorithmes génétiques. Exploration, optimisation et apprentissage automatique*, Addison-Wesley France, 1994.
- [GOS 89] GOSS S., ARON S., DENEUBOURG J. L., PASTEELS J. M., « Self-Organized Shortcuts in the Argentine Ant », *Naturwissenschaften*, vol. 76, 1989, p. 579–581.
- [GRA 59] GRASSÉ P.-P., « La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs », *Insectes sociaux*, vol. 6, 1959, p. 41–83.
- [HOL 62] HOLLAND J. H., « Outline for logical theory of adaptive systems », *J. Assoc. Comput. Mach.*, vol. 3, 1962, p. 297–314.
- [HOL 90] HOLLOBLER B., WILSON E., *The Ants*, Springer Verlag, 1990.
- [KEN 95] KENNEDY J., EBERHART R. C., « Particle swarm optimization », *Proc. IEEE Int. Conf. on Neural Networks*, vol. IV, Piscataway, NJ : IEEE Service Center, 1995, p. 1942–1948.
- [KEN 01] KENNEDY J., EBERHART R., SHI Y., *Swarm Intelligence*, Evolutionary Computation, Morgan Kaufmann, avril 2001.
- [LAR 02] LARRAÑAGA P., LOZANO J., *Estimation of Distribution Algorithms, A New Tool for Evolutionary Computation*, Genetic Algorithms and Evolutionary Computation, Kluwer Academic Publishers, 2002.
- [LIN 02] LING C., JIE S., LING Q., HONGJIAN C., « A Method for Solving Optimization Problems in Continuous Space Using Ant Colony Algorithm », DORIGO M., DI CARO G., SAMPELS M., Eds., *Proceedings of the Third International Workshop on Ant Algo-*

- gorithms (ANTS'2002)*, vol. 2463 de *Lecture Notes in Computer Science*, Brussels, Belgium, septembre 2002, Springer Verlag, p. 288-289.
- [MAT 00] MATHUR M., KARALE S. B., PRIYE S., JYARAMAN V. K., KULKARNI B. D., « Ant Colony Approach to Continuous Function Optimization », *Ind. Eng. Chem. Res.*, vol. 39, 2000, p. 3814-3822.
- [MEI 82] MEINHARDT H., *Models of Biological Pattern Formation*, Academic Press, London, 1982.
- [MON 99] MONMARCHÉ N., RAMAT E., DROMEL G., SLIMANE M., VENTURINI G., « On the similarities between AS, BSC and PBIL : toward the birth of a new meta-heuristics », *E3i* n° 215, 1999, Université de Tours.
- [MON 00a] MONMARCHÉ N., RAMAT N., DESBARAT L., VENTURINI G., « Probabilistic search with genetic algorithms and ant colonies », WU A., Ed., *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop*, 2000, p. 209-211.
- [MON 00b] MONMARCHÉ N., VENTURINI G., SLIMANE M., « On how *Pachycondyla apicalis* ants suggest a new search algorithm », *Future Generation Computer Systems*, vol. 16, 2000, p. 937-946.
- [NEL 65] NELDER J. A., MEAD R., « A simplex method for function minimization », *Computer Journal*, vol. 7, 1965, p. 308-313.
- [NIC 77] NICOLIS G., PRIGOGINE I., *Self-organization in Non-equilibrium Systems*, New York, 1977.
- [NOU 02] NOUYAN S., « Agent-Based Approach to Dynamic task Allocation », DORIGO M., DI CARO G., SAMPELS M., Eds., *Proceedings of the Third International Workshop on Ant Algorithms (ANTS'2002)*, vol. 2463 de *Lecture Notes in Computer Science*, Brussels, Belgium, septembre 2002, Springer Verlag, p. 28-39.
- [PAR 96] PARDALOS P., XUE G., PANAGIOTOPOULOS P. D., « Parallel Algorithms for Global Optimization Problems », *Solving Combinatorial Optimization Problems in Parallel*, 1996, p. 232-247.
- [PRI 71] PRIGOGINE I., GLANDSDORF P., *Thermodynamic Theory and Structure, Stability and Fluctuations*, Wiley and Sons, New York, 1971.
- [REC 65] RECHENBERG I., *Cybernetic Solution Path of an Experimental Problem*, Royal Aircraft Establishment Library Translation, 1965.
- [RES 00] RESENDE M., « Greedy randomized adaptive search procedures (GRASP) », rapport n° TR 98.41.1, 2000, AT&T Labs-Research.
- [RUD 92] RUDOLPH G., « *Parallel approaches to stochastic global optimization* », p. 256-267, IOS Press, Amsterdam, 1992.
- [SEE 89] SEELEY T. D., « The honey bee colony as a superorganism », *American Scientist*, vol. 77, 1989, p. 546-553.
- [TAI 98a] TAILLARD E. D., « 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, 1998.
- [TAI 98b] TAILLARD E. D., GAMBARDELLA L. M., GENDREAU M., POTVIN J.-Y., « Adaptive Memory Programming : A Unified View of Meta-Heuristics », *European Journal of Operational Research*, vol. 135, n° 1, 1998, p. 1-16.

- [TAL 02] TALBI E.-G., « A Taxonomy of Hybrid Metaheuristics », *Journal of Heuristics*, vol. 8, n° 5, 2002, p. 541–564.
- [THE 95] THERAULAZ G., BONABEAU E., « Coordination in distributed building », *Science*, vol. 269, 1995, p. 686–688.
- [THE 98] THERAULAZ G., BONABEAU E., DENEUBOURG J.-L., « Response Threshold Reinforcement and Division of Labour in Insect Societies », *Proceedings of the Royal Society of London B*, vol. 265, 1998, p. 327–335.
- [WIL 84] WILSON E., « The relation between Caste Ratios and Division of Labour in the Ant Genus *Pheidole* (Hymenoptera : Formicidae) », *Behav. Ecol. Sociobiol.*, vol. 16, 1984, p. 89–98.
- [WOD 97] WODRICH M., BILCHEV G., « Cooperative distributed search : the ant's way », *Control and Cybernetics*, vol. 26, n° 3, 1997.

Sites internet

– **Livre sur les métaheuristiques :**

<http://www.eyrolles.com/Informatique/Livre/9782212113686/>

– **Livre sur la théorie de l’auto-organisation :**

<http://www.scottcamazine.com/personal/selforganization/SOMain.html>

– **Thèse sur la programmation à mémoire adaptative :**

<http://ina.eivd.ch/collaborateurs/etd/articles.dir/IDSIA-52-98.ps>

– **Méthode des essaims particulaires :**

<http://www.particleswarm.info/>

– **Groupe de travail sur les algorithmes évolutionnaires :**

<http://afia.lri.fr/node.php?lang=fr&node=285>

– **Méthode des colonies de fourmis :**

<http://iridia.ulb.ac.be/~mdorigo/ACO/>

– **Systèmes immunitaires artificiels :**

<http://www.cs.kent.ac.uk/people/staff/jt6/aisbook/ais-researchers.htm>

***Johann Dréo** est docteur en informatique, et biologiste de formation. Ses recherches portent sur les métaheuristiques d'optimisation inspirées des insectes sociaux.*

***Patrick Siarry** est ingénieur de l'École Supérieure d'Électricité et professeur à l'université Paris XII Val-de-Marne où il dirige des travaux de recherche sur les méthodes heuristiques récentes pour l'optimisation difficile.*