

Un algorithme de colonie de fourmis hybridé avec un algorithme de recherche locale ; applications à l'optimisation en variables continues.

Johann Dréo et Patrick Siarry

Université de Paris XII, Laboratoire d'Étude et de Recherche en Instrumentation, Signaux et Systèmes (LERISS, E.A.412), 61 avenue du Général de Gaulle, 94010 CRÉTEIL Cedex

*dreo@univ-paris12.fr, 01.45.17.14.95
siarry@univ-paris12.fr, 01.45.17.15.67*

Mots-clés : optimisation continue, méthode des colonies de fourmis, recrutement, communication directe, hybridation, simplex.

Le premier algorithme inspiré des colonies de fourmis (le “Ant System” [1]), a été appliqué à divers problèmes d'optimisation combinatoire avec succès. Par la suite, un petit nombre de travaux ont été consacrés à l'adaptation au cas continu de cette nouvelle métaheuristique. Le premier algorithme conçu pour l'optimisation de fonctions continues est l'algorithme CACO (“Continuous Ant Colony Algorithm” [2]). Tous ces algorithmes retiennent un trait particulier du comportement des fourmis réelles : le dépôt de piste de phéromone. En effet, la colonie de fourmis est souvent décrite comme un système distribué capable de résoudre des problèmes complexes de par la mise en jeu de processus “stigmergiques”, définis comme une forme de communication indirecte, par le biais de modifications de l'environnement.

Mais le dépôt de piste est aussi rencontré dans certains processus de “recrutement”, défini par les biologistes comme un mode de communication conduisant quelques individus à se réunir à un endroit où un travail est nécessaire. Une autre méthode d'optimisation s'inspirant de cette définition a été développée dans le cas continu : l'algorithme API, inspiré par le comportement de recrutement d'une fourmi primitive [3]. Cependant, l'algorithme API utilise peu les structures mémorielles qui caractérisent généralement les systèmes de colonies de fourmis [4], à savoir la présence d'une mémoire externe, partagée par les agents.

Notre point de vue est que la métaphore des colonies de fourmis peut être définie comme un modèle mettant en jeu non seulement des phénomènes stigmergiques, mais aussi, plus largement, des processus de recrutement. Nous avons développé, dans cette optique, une méthode d'optimisation continue, inspirée des colonies de fourmis, qui exploite le concept de communication interindividuelle [5]. Cette méthode, dénommée CIAC (“Continuous Interacting Ant Colony”), s'applique à la fois à la recherche globale et à la recherche locale d'un optimum. Ce nouvel algorithme a été testé sur un large ensemble de fonctions analytiques, comprenant un grand nombre de variables (jusqu'à 100), et comparé avec les meilleures méthodes publiées (autres algorithmes fondés sur les colonies de fourmis, algorithmes génétiques, recherche tabou et recuit simulé).

Comme le “Ant System” original, notre algorithme est moins compétitif dans le domaine de la recherche locale, malgré une recherche globale relativement efficace. Pour

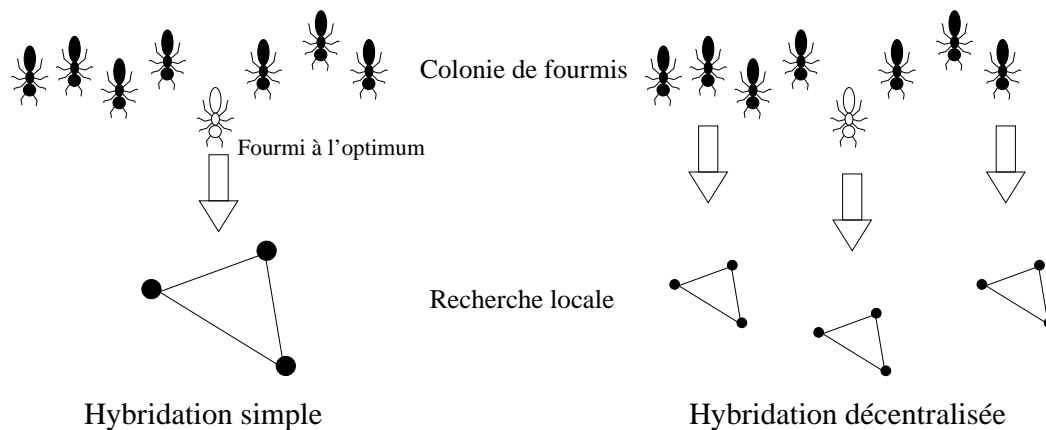


Figure 1: Deux hybridations possibles entre l’algorithme CIAC et le “simplex” de Nelder-Mead

pallier à ce problème, nous avons choisi d’implémenter une hybridation avec l’algorithme du “simplex” de Nelder-Mead [6], nous avons appelé cet algorithme hybridé “HCIAC”. Plusieurs versions de l’algorithme hybridé ont été implémentées (cf. figure 1), du fait des différentes possibilités de relations entre les deux méthodes : relation temporelle (la recherche locale est lancée à intervalles réguliers à partir de la meilleure solution) ou relation spatiale (la recherche locale est lancée indépendamment par chaque fourmi). Nous qualifierons le premier algorithme d’hybridation “simple”, puisqu’il s’agit simplement d’améliorer la meilleure solution à un temps donné. La seconde hybridation permet de maintenir l’aspect fortement décentralisé qui caractérise les algorithmes de colonies de fourmis. En effet, chaque agent décide de pratiquer une recherche locale uniquement sur les informations dont il dispose, et non pas (comme dans le premier cas) sur des informations globales. On peut qualifier cette hybridation de “décentralisée”.

L’algorithme hybride HCIAC est aujourd’hui testé sur le même jeu de fonctions que l’algorithme CIAC et il présente des améliorations notables sur certains points.

- [1] A. Coloni, M. Dorigo, V. Maniezzo. “Distributed Optimization by Ant Colonies”. *Proceedings of ECAL91 - European Conference on Artificial Life*, 134-142. Elsevier Publishing, 1991.
- [2] G. Bilchev, I.C. Parmee, “The Ant Colony Metaphor for Searching Continuous Design Spaces”, *Lecture Notes in Computer Science*, 993, 25-39, 1995.
- [3] N. Monmarché, G. Venturini, M. Slimane, “On how *Pachycondyla apicalis* ants suggest a new search algorithm”, *Future Generation Computer Systems* , 16, 937-946, 2000.
- [4] E.D. Taillard, L. Gambardella, M. Gendreau, J-Y. Potvin, “Adaptive Memory Programming: A Unified View of Metaheuristics”, *EURO XVI Conference Tutorial and Research Reviews booklet*, Brussels, July 1998.
- [5] J. Dréo & P. Siarry, “Un nouvel algorithme de colonie de fourmis exploitant la communication directe entre individus ; application à l’optimisation en variables continues”, *ROADEF 2002 - 4ème congrès de la société française de recherche opérationnelle et d’aide à la décision*, Paris ENST, 120-121, février 2002.
- [6] J.A. Nelder & R. Mead. “A simplex method for function minimization.” *The Computer Journal*, 7(4):308-313, 1965.