

# Minimisation du coût d'affectation du personnel en utilisant les algorithmes HHO et ACO

## Minimizing the cost of staff assignment using HHO and ACO algorithms

El Attaoui Anas<sup>1</sup>, Norelislam El Hami<sup>1</sup>

<sup>1</sup>Laboratoire sciences et ingénierie, Ecole Nationale des sciences appliquées, Université Ibn Tofail, Kenitra, Maroc

**RÉSUMÉ.** Dans cette recherche, deux paradigmes d'optimisation basés sur la population et inspirés de la nature ont été décrits, appelés "Harris Hawks Optimization" (HHO) et "Ant Colony Optimization" (ACO). La véritable inspiration de HHO est le comportement coopératif et la technique de poursuite des buses de Harris dans la nature, appelée "bond surprise". Dans le même temps, l'ACO s'inspire de l'observation du comportement de fourmis réelles. Ces deux mouvements naturels ont été modélisés mathématiquement pour créer des algorithmes d'optimisation. L'efficacité des optimiseurs HHO et ACO est vérifiée à travers des comparaisons qui montrent que l'algorithme HHO fournit des résultats prometteurs et compétitifs au niveau des fonctions tests, tandis que pour l'étude cas traitée, qui est la planification des horaires et la minimisation du coût d'affectation du personnel, ACO est meilleur.

**ABSTRACT.** In this research, two population-based, nature-inspired optimization paradigms were described, dubbed "Harris Hawks Optimization" (HHO) and "Ant Colony Optimization" (ACO). The real inspiration of HHO is the cooperative behavior and pursuing technique of Harris' hawks in nature, termed "surprise pounce." At the same time, ACO is inspired by observing the behavior of actual ants. Those two natural movements were mathematically modeled to create optimization algorithms. The effectiveness of HHO and ACO optimizers is checked throughout comparisons that show that the HHO algorithm provides better results when the comparison is based on test functions, while for the case study treated, which is the planning of schedules and the minimization of the cost of staff assignment, ACO is better.

**MOTS-CLÉS.** ACO-HHO-Algorithm-Optimisation-Métaheuristique-Affectation du personnel.

**KEYWORDS.** ACO-HHO-Algorithm-Optimization-Metaheuristic-Staff assignment.

## 1. Introduction

De nombreux problèmes d'optimisation du monde réel sont continus, discrets, contraints ou non contraints [Abb 19, Far 19]. En raison de ces caractéristiques, certaines classes de problèmes sont difficiles à résoudre à l'aide des méthodes traditionnelles de programmation mathématique, telles que le gradient conjugué, la programmation quadratique séquentielle, la méthode de la plus forte pente rapide et la méthode quasi-Newton [Noc 06, Wu 16]. Plusieurs formes d'études ont montré que ces approches sont inefficaces ou inefficaces pour traiter diverses situations multimodales, non-continues et non-différentiables à plus grande échelle dans le monde réel [Wu 15]. Or, en raison de leur simplicité et de leur facilité de mise en œuvre, les algorithmes métaheuristiques ont été conçus et utilisés comme solutions alternatives compétitives pour un large éventail de problèmes. Cependant, la majorité des métaheuristiques ont un défaut commun : ils présentent souvent une sensibilité délicate à l'ajustement des paramètres fournis par l'utilisateur. Un autre inconvénient est que les algorithmes métaheuristiques n'atteignent pas toujours l'optimum global [Dré 06]. Dans ce papier, nous allons présenter deux métaheuristiques qui sont ACO et HHO, une comparaison de performances à l'aide de plusieurs fonctions tests et finalement un étude de cas qui est la minimisation du coût d'affectation du personnel.

## 2. Harris Hawks optimization -HHO

La méthode « Harris hawks optimization » (HHO) est un nouveau paradigme d'optimisation par intelligence d'essaim introduit en 2019 par Ali Asghar Heidari et al, qui s'inspire des comportements d'équipe et des schémas de poursuite des buses de Harris dans la nature. La buse de Harris est

connue pour chasser de manière coopérative en groupes, bien que d'autres rapaces chassent fréquemment seuls. L'attitude sociale de la buse de Harris a été attribuée à son intelligence, ce qui en fait un oiseau facile à dresser et populaire pour la fauconnerie [Moa 19]. Plusieurs buses bondissent sur une cible à partir de différentes directions pour tenter de la prendre au dépourvu. Les buses de Harris peuvent présenter divers comportements de poursuite en fonction de la nature dynamique des événements et des mouvements de fuite de la proie [Bed 88].

## 2.1. Algorithme HHO

L'algorithme de HHO se découpe en 3 phases principales qui sont :

- phase d'exploration ;
- phase de transition de l'exploration à l'exploitation ;
- phase d'exploitation.

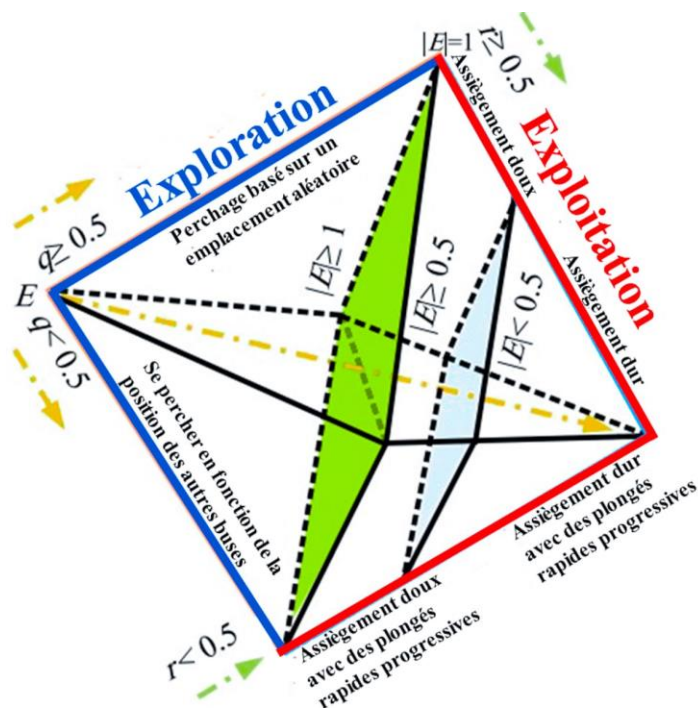


Figure 1.1. Les étapes de HHO

### 2.1.1. Phase d'exploration

Lorsque nous analysons la nature des buses de Harris, nous constatons que leur forte vision permet de suivre et d'identifier les proies, bien que celles-ci ne soient pas toujours immédiatement visibles. Par conséquent, les buses attendent, observent et surveillent le site du désert pendant plusieurs heures avant de voir une cible. Les faucons de Harris sont les solutions candidates dans HHO, et la meilleure solution candidate à chaque phase est considérée comme la proie visée (optimum). Dans HHO, les buses de Harris s'assoient au hasard dans diverses zones et attendent leurs proies en utilisant deux tactiques [Hei 19].

### 2.1.2. Phase de transition de l'exploration vers l'exploitation

L'algorithme HHO peut passer de l'exploration à l'exploitation et basculer entre différentes actions d'exploitation en fonction de l'énergie de fuite de la proie. L'énergie de la proie diminue de manière significative tout au long des activités de fuite [Hei 19].

### 2.1.3. Phase d'exploitation

Les buses de Harris exécutent le bondissement surprise dans cette phase en attaquant la proie découverte dans la phase précédente. D'autre part, les proies tentent souvent d'échapper à des circonstances néfastes. Par conséquent, différentes techniques de poursuite apparaissent dans le monde réel. Le HHO propose quatre façons différentes d'imiter la phase d'attaque en se basant sur les comportements de fuite des proies et les stratégies de poursuite des buses de Harris [Hei 19].

## 3. Ant Colony Optimization (ACO)

Ant Colony optimization a été inspirée par l'examen du comportement réel des fourmis. Dans les années 1940 et 1950, l'entomologiste français Pierre-Paul Grassé a découvert que certaines espèces de termites réagissaient à ce qu'il appelait des "stimuli importants" [Gra 46]. Il a constaté que les répercussions de ces réactions pouvaient servir de nouveaux stimuli importants à la fois pour l'insecte qui les a générées et pour les autres insectes de la colonie. Grassé a utilisé le mot stigmergie [Gra 59] pour caractériser ce type de communication dans lequel « les travailleurs sont stimulés par les performances qu'ils ont réalisées ». Les colonies de fourmis sont un excellent endroit pour voir la stigmergie en action. Les fourmis qui se rendent à une source de nourriture et en reviennent laissent tomber au sol une matière appelée phéromone. D'autres fourmis détectent la présence de phéromones et ont tendance à suivre les pistes où la concentration de phéromones est plus élevée. Les fourmis utilisent ce processus pour acheminer la nourriture vers leur nid de manière très efficace [Den 90], et peuvent utiliser ce comportement au niveau de la colonie, qui est basé sur l'utilisation de la rétroaction positive, pour découvrir le chemin le plus court entre une source de nourriture et leur nid [Gos 89].

### 3.1. Algorithme ACO pour les problèmes combinatoire

Le Premier algorithme ACO appelé « Système de fourmi » a été proposé par Dorigo et al., et il a été dédié principalement à la résolution des problèmes d'optimisation combinatoires [Dor 06]. L'une des meilleures façons d'illustrer le fonctionnement de la métaheuristique système de fourmi est son application au problème du voyageur de commerce [Dor 97], qui consiste en un ensemble de lieux (villes) et d'un voyageur de commerce qui doit visiter tous les lieux une fois et une seule. Les distances entre les lieux sont données et la tâche consiste à trouver une tournée hamiltonienne de longueur minimale.

La caractéristique principale de l'algorithme c'est que à chaque itération, la valeur de phéromone entre chaque deux points (villes) se mets à jour par toutes les fourmis qui ont construit une solution dans l'itération elle-même. Dans la construction d'une solution, les fourmis sélectionnent la ville à visiter par un mécanisme stochastique, pour que finalement on obtient le chemin optimal. La mise à jour de la phéromone et la construction des solutions sont modélisées mathématiquement [Dor 96].

### 3.2. Algorithme ACO pour les problèmes continues

L'algorithme ACO a permis la résolution de plusieurs problèmes combinatoires [Dor 06]. Quoique, il y'a une grande catégorie de problèmes qui ne sont pas combinatoires mais continue. Dès les premiers jours de l'ACO en tant que technique d'optimisation combinatoire, les scientifiques ont cherché à l'utiliser pour résoudre également des problèmes continus. Cependant, l'extension de la métaheuristique ACO aux domaines continus n'était pas évidente. Les nombreuses approches proposées variaient souvent de la formulation de base de l'ACO. Une version étendue de l'ACO pour les domaines continus, intitulée  $ACO_{\mathbb{R}}$  a été conçue par M. Dorigo et Socha en 2008 pour résoudre les problèmes d'optimisation continues [Soch 08].

Dans l'ACO pour les problèmes combinatoires, les valeurs des phéromones sont associées à un ensemble fini de valeurs discrètes liées aux décisions des fourmis, ce qui permet d'exprimer les valeurs des phéromones sous la forme d'un tableau de phéromones. Ceci n'est pas réalisable dans la situation continue, car les valeurs potentielles ne sont pas finies. Par conséquent,  $ACO_{\mathbb{R}}$  utilise plutôt une archive de solutions comme moyen de représenter la distribution des phéromones dans l'espace de recherche.

L'archive de solutions offre plusieurs solutions appropriées au problème. Alors qu'un modèle de phéromone en optimisation combinatoire peut être considéré comme une mémoire implicite de l'historique de recherche, une archive de solutions est une mémoire explicite [Soch 08].

Le fonctionnement fondamental du modèle  $ACO_{\mathbb{R}}$  algorithme est le suivant : Dans un premier temps, l'archive des solutions est initialisée. Ensuite, à chaque itération, de nombreuses solutions sont générées de manière probabiliste par les fourmis. Ces solutions peuvent être améliorées par n'importe quel processus d'amélioration (par exemple, recherche locale ou approches par gradient). Enfin, l'archive des solutions est mise à jour avec les solutions créées [Soch 08].

#### 4. Test de performance des algorithmes

Dans cette section on va tester l'algorithme HHO et  $ACO_{\mathbb{R}}$  grâce aux fonctions tests suivantes :

Fonctions objectives	Acronyme	Xmin	Xmax	Dimension	Minimum global
Branin	BR	-5	15	2	0,3978878
Bohachevsky 1	BO1	-100	100	2	0
Bohachevsky 2	BO2	-100	100	2	0
Six-Hump Camel	CA	-5	5	2	-1,0316
Cosine mixtrure	CM	-1	1	4	-0.4
De jong	DE	-5.12	5.12	3	0
Goldstein Price	GP	-2	2	2	3
Griewank	GR	-600	600	50	0
Hartmann 3	HA3	0	1	3	-3,862782
Hartmann 6	HA6	0	1	6	-3,322368
Hansen	HAN	-10	10	2	- 176,541793
Rastrigin	RA	-5.12	5.12	2	0
Rosenbrock	RO	-10	10	10	0
Shekel 5	SK5	0	10	4	-10,1532
Shekel 7	SK7	0	10	4	-10,40294
Shekel 10	SK10	0	10	4	-10,53641

Tableau 4.1. Fonctions tests

On lance le programme HHO et  $ACO_{\mathbb{R}}$  **100** fois.

Les paramètres du programme HHO sont :

- le nombres de candidats (Buses) = 30 ;
- l’erreur mathématique = 0.01.

On a abouti aux résultats suivants :

Fonctions objectives	Acronyme	Temps moyen d’exécution (s)	Moyenne des minimums globaux obtenus	Taux de succès
Branin	BR	0.27491	0,3978878	100%
Bohachevsky 1	BO1	0.29202	0	100%
Bohachevsky 2	BO2	0.30219	0	100%
Six-Hump Camel	CA	0.10646	-1,0316	100%
Cosine mixtrure	CM	0.08334	-0.4	100%
De jong	DE	0.29278	0	100%
Goldstein Price	GP	0.16501	3	100%
Griewank	GR	0.38316	0.5913	0%
Hartmann 3	HA3	0.35413	-3,862782	100%
Hartmann 6	HA6	0.43698	-3,2725	60%
Hansen	HAN	0.3291	-176,541793	100%
Rastrigin	RA	0.26607	0	100%
Rosenbrock	RO	0.47709	0	100%
Shekel 5	SK5	0.70456	-10,1532	100%
Shekel 7	SK7	0.77932	-10,40294	100%
Shekel 10	SK10	0.8887	-10,53641	100%

**Tableau 4.2.** Résultat du programme HHO

Les paramètres du programme  $ACO_{\mathbb{R}}$  sont :

- le nombres de candidats (Fourmis) = 40 ;
- la taille de l’archive est  $K=10$  ;
- la vitesse de convergence  $\xi = 1$  ;
- localité du processus de recherche  $q = 0.5$  ;
- l’erreur mathématique = 0.01.

Les résultats sont comme suit :

Fonctions objectives	Acronyme	Temps moyen d'exécution (s)	Moyenne des minimums globaux obtenus	Taux de succès
Branin	BR	0.26012	0,39797	99%
Bohachevsky 1	BO1	0.24409	0	100%
Bohachevsky 2	BO2	0.24461	0.0021	100%
Six-Hump Camel	CA	0.22856	-1,0316	100%
Cosine mixtrure	CM	0.36307	-0.4	100%
De jong	DE	0.29041	0	100%
Goldstein Price	GP	0.30116	3	100%
Griewank	GR	22.408	0,016529	88%
Hartmann 3	HA3	0.2776	-3,862782	100%
Hartmann 6	HA6	0.51438	-3,2973	79%
Hansen	HAN	0.2615	-167,2415	72%
Rastrigin	RA	0.28672	0,12934	90%
Rosenbrock	RO	11.7997	0,03143	85%
Shekel 5	SK5	0.47213	-5,7163	40%
Shekel 7	SK7	0.47588	-6,8995	52%
Shekel 10	SK10	0.47274	-8,3972	71%

**Tableau 4.3.** Résultat du programme  $ACO_{\mathbb{R}}$

**REMARQUE.** Au niveau des fonctions test, l'algorithme HHO donne des résultats très pertinents par rapport à  $ACO_{\mathbb{R}}$ .

## 5. Application : Minimisation du coût d'affectation du personnel

Le directeur d'une usine souhaite organiser le planning de ses ouvriers de manière à assurer une présence minimum d'un certain nombre d'ouvriers à toute période de la journée.

Une journée est découpée par tranche de 4h et le besoin minimum des ouvriers est indiqué dans le tableau suivant :

Période	0h-4h	4h-8h	8h-12h	12h-16h	16h-20h	20h-24h
Nombre d'ouvriers	8	13	15	22	15	12

**Tableau 5.1.** Besoin minimum des ouvriers

- les ouvriers prennent leur service une fois par jour, pour une durée de 8h successives ;
- les horaires de prise en service sont au début de chaque période, soit 0h,4h,8h,12h,16h,20h ;
- le coût horaire est de 10 €/h.



Le directeur a la possibilité de demander aux ouvriers de faire des heures supplémentaires qui ne dépassent pas 4h, à la suite de leur service. Le paiement des heures supplémentaires est à 150% du paiement initial, soit 15 €/h.

Si on ne considère pas les règles particulières quant au cumul des heures supplémentaires, et l'on considère une journée type (pas d'absence, ni congés), **quelle serait l'affectation de coût minimum permettant la présence d'un nombre d'ouvriers minimum par période sur une journée ?**

### 5.1. Modélisation mathématique

Formalisons la problématique en un modèle mathématique. Nous devons connaître les horaires de présence des ouvriers par période. Ceux-ci ne peuvent effectuer que deux types de services : un service normal de 8h ou un service long de 12h.

Nous considérons deux types de variables de décision : Le nombre d'ouvriers en service normal commençant à une période donnée et le nombre d'ouvriers en service long commençant à une période donnée.

- |  |  |
|--|--|
| $X_1$ = Ouvrier service normal débutant à 0h | $X_7$ = Ouvrier service normal débutant à 12h    |
| $X_2$ = Ouvrier service long débutant à 0h   | $X_8$ = Ouvrier service long débutant à 12h      |
| $X_3$ = Ouvrier service normal débutant à 4h | $X_9$ = Ouvrier service normal débutant à 16h    |
| $X_4$ = Ouvrier service long débutant à 4h   | $X_{10}$ = Ouvrier service long débutant à 16h   |
| $X_5$ = Ouvrier service normal débutant à 8h | $X_{11}$ = Ouvrier service normal débutant à 20h |
| $X_6$ = Ouvrier service long débutant à 8h   | $X_{12}$ = Ouvrier service long débutant à 20h   |

Pour un taux horaire de 10 € en temps normal et 15 € en temps supplémentaires, les coûts des services normal et long sont respectivement 80 € et 140 €. Et donc, la fonction objectif du salaire total par jours s'écrit de la manière suivante :

$$\text{Min } f(X_j) = 80*(X_1 + X_3 + X_5 + X_7 + X_9 + X_{11}) + 140*(X_2 + X_4 + X_6 + X_8 + X_{10} + X_{12})$$

Les contraintes concernent seulement le respect du nombre minimal des ouvriers par période, pour les écrire il est important de visualiser la présence des ouvriers des différents services. Pour se faire nous pouvons utiliser une vue matricielle : Une période par ligne et un type de service par colonne. Dans le tableau suivant, chaque valeur de 1 indique que les ouvriers du service concerné (colonne) sont présents dans la plage horaire (ligne) :

Période	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$
0h-4h	1	1								1	1	1
4h-8h	1	1	1	1								1
8h-12h		1	1	1	1	1						
12h-16h				1	1	1	1	1				
16h-20h						1	1	1	1	1		
20h-24h								1	1	1	1	1

Tableau 5.2. Matrice de présence des ouvriers par service

Les contraintes s'écrivent donc comme suit :

$$X_1 + X_2 + X_{10} + X_{11} + X_{12} \geq 8$$

$$X_1 + X_2 + X_3 + X_4 + X_{12} \geq 13$$

$$X_2 + X_3 + X_4 + X_5 + X_6 \geq 15$$

$$X_4 + X_5 + X_6 + X_7 + X_8 \geq 22$$

$$X_6 + X_7 + X_8 + X_9 + X_{10} \geq 15$$

$$X_8 + X_9 + X_{10} + X_{11} + X_{12} \geq 12$$

$$X_{ij} \geq 0 ;$$

## 5.2. Résultats

On a fait la résolution de ce problème par les deux métaheuristiques  $ACO_{\mathbb{R}}$  et HHO, avec les mêmes paramètres utilisés précédemment.

Les résultats sont les suivants :

- Pour HHO, la valeur de la fonction objectif et les variables de décision sont :

$$f(X_j) = 3700 \text{ €}$$

$$\begin{array}{llllll} X_1 = 3 ; & X_2 = 1 ; & X_3 = 4 ; & X_4 = 1 ; & X_5 = 9 ; & X_6 = 0 ; \\ X_7 = 7 ; & X_8 = 5 ; & X_9 = 4 ; & X_{10} = 0 ; & X_{11} = 0 ; & X_{12} = 4 ; \end{array}$$

- Pour  $ACO_{\mathbb{R}}$ , la valeur de la fonction objectif et les variables de décision sont :

$$f(X_j) = 3580 \text{ €}$$

$$\begin{array}{llllll} X_1 = 6 ; & X_2 = 0 ; & X_3 = 3 ; & X_4 = 3 ; & X_5 = 9 ; & X_6 = 0 ; \\ X_7 = 5 ; & X_8 = 5 ; & X_9 = 5 ; & X_{10} = 0 ; & X_{11} = 1 ; & X_{12} = 1 ; \end{array}$$

**REMARQUE.** La solution obtenue par  $ACO_{\mathbb{R}}$  est moins coûteuse, et du coup on peut conclure que pour cette application  $ACO_{\mathbb{R}}$  est plus performant que HHO

## 6. Conclusion

Une métaheuristique est un algorithme d'optimisation qui a pour but la résolution des problèmes d'optimisation complexes d'une manière rapide et efficace. Dans ce papier, nous avons présenté deux métaheuristiques basées sur la population nommés ACO et HHO. L'étude a montré que HHO est très performant au niveau des fonctions tests. En effet, après avoir lancé les deux programmes 100 fois et dans les mêmes conditions, HHO a donné des résultats remarquables avec des taux de succès élevés par rapport à ACO. En revanche, pour le cas du problème de minimisation du coût d'affectation du personnel, ACO a retourné un coût minimal égal à 3580 € optimal par rapport à la valeur donnée par HHO, qui est 3700 €. Dans ce contexte, plusieurs questions et perspectives se posent, notamment est ce que pourquoi HHO, qui est meilleur au niveau des fonctions tests n'a pas pu trouver un minimum optimal de la fonction objectif du problème traité qui est désormais linéaire, et ACO a donné un résultat meilleur. Cette perspective fera l'objet d'une nouvelle recherche.



## Bibliographie

- [Abb 19] R. Abbassi, A. Abbassi, A.A. Heidari, S. Mirjalili, *An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models*, *Energy Convers. Manage.* 179 (2019) 362–372.
- [Far 19] H. Faris, A.M. Al-Zoubi, A.A. Heidari, I. Aljarah, M. Mafarja, M.A. Hassonah, H. Fujita, *An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks*, *Inf. Fusion* 48 (2019) 67–83.
- [Noc 06] J. Nocedal, S.J. Wright, *Numerical Optimization*, 2nd ed., 2006.
- [Wu 16] G. Wu, *Across neighborhood search for numerical optimization*, *Inform. Sci.* 329 (2016) 597–618.
- [Wu 15] G. Wu, W. Pedrycz, P.N. Suganthan, R. Mallipeddi, *A variable reduction strategy for evolutionary algorithms handling equality constraints*, *Appl. Soft Comput.* 37 (2015) 774–786.
- [Dré 06] J. Dréo, A.P. étrowski, P. Siarry, E. Taillard, *Metaheuristics for Hard Optimization: Methods and Case Studies*, Springer Science & Business Media, 2006.
- [Moa 19] H. Moayedi, A. Osouli, H. Nguyen and A. Rashid, “*A novel Harris hawks’ optimization and k-fold cross-validation predicting slope stability*”, *Engineering with Computers*, 2019
- [Bed 88] J.C. Bednarz, *Cooperative hunting in harris’ hawks (parabuteo unicinctus)*, *Science* 239 (1988) 1525.
- [Hei 19] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “*Harris hawks optimization: Algorithm and applications*,” *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi : 10.1016/j.future.2019.02.028.
- [Gra 46] P.-P. Grassé, *Les Insectes Dans Leur Univers*, Paris, France : Ed. du Palais de la découverte, 1946.
- [Gra 59] P.-P. Grassé, “*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, pp. 41–81, 1959.
- [Den 90] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, “*The self-organizing exploratory pattern of the Argentine ant*,” *Journal of Insect Behavior*, vol. 3, p. 159, 1990.
- [Gos 89] S. Goss, S. Aron, J.-L. Deneubourg, and J.-M. Pasteels, “*Self-organized shortcuts in the Argentine ant*,” *Naturwissenschaften*, vol. 76, pp. 579–581, 1989.
- [Dor 06] M. Dorigo, M. Birattari, et T. Stutzle, “*Ant colony optimization*,” *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.
- [Dor 97] M. Dorigo and L. M. Gambardella, “*Ant colony system: a cooperative learning approach to the traveling salesman problem*,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997, doi: 10.1109/4235.585892.
- [Dor 96] M. Dorigo, V. Maniezzo, and A. Coloni, “*Ant System: Optimization by a colony of cooperating agents*,” *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 26, no. 1, pp. 29–41, 1996.
- [Soch 08] K. Socha et M. Dorigo, “*Ant colony optimization for continuous domains*”, *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, mars 2008, doi: 10.1016/j.ejor.2006.06.046.