

Optimisation métaheuristique et en application mécatronique

Metaheuristic optimization and mechatronic application

Hakima Reddad¹, Maria Zemzami², Norelislam El Hami³, Nabil Hmina¹

¹ Laboratoire LGS, Université Moulay Sultan Slimane, Béni Mellal, Maroc

² Laboratoire LGS, Ecole nationale des sciences appliquées, Béni Mellal, Maroc

³ Laboratoire des sciences et ingénierie, Ecole nationale des sciences appliquées, Université Ibn Tofail, Kénitra, Maroc

RÉSUMÉ. Cet article est composé de deux parties, la première partie est consacrée à une présentation de cinq algorithmes métaheuristiques d'optimisation, l'algorithme SSA (l'algorithme d'essaim des salpes) qui se base sur le déplacement linéaire des salpes dans les océans pour trouver leur cible, l'algorithme CSA (l'algorithme de recherche des coucous) qui se base sur le comportement des coucous pour trouver et pondre leurs œufs dans les nids d'hôtes, l'algorithme GWO (l'optimisation des loups gris) qui est basé sur le comportement des loups gris dans la nature à travers une hiérarchie sociale, l'algorithme FA (l'algorithme des lucioles) qui est basé sur l'attractivité à la lumière des lucioles les unes envers les autres dans la nature et enfin l'algorithme BA (l'algorithme des chauves-souris) qui est basé sur le processus de chasse des micro chauves-souris à travers l'écholocation.

Et la deuxième partie concerne la résolution d'un problème mécatronique d'optimisation réel sous des contraintes, il s'agit d'un calcul numérique par les algorithmes qu'on a discuté dans la première partie, de la géométrie optimale d'un support d'une plaque de commercialisation, sous des contraintes bien déterminées, ce support doit garantir une résistance mécanique contre des efforts de compression et de flambage engendrés par la plaque, et selon les résultats obtenues par chaque algorithme on va déduire la solution optimale du problème.

ABSTRACT. This article is composed of two parts, the first part is devoted to a presentation of five metaheuristic optimization algorithms, the SSA algorithm (the salp swarm algorithm) which is based on the linear displacement of salps in the oceans to find their target, the CSA algorithm (the cuckoo search algorithm) which is based on the behavior of cuckoos to find and lay their eggs in host nests, the GWO algorithm (the optimization of gray wolves) which is based on the behavior of gray wolves in nature through a social hierarchy, the FA algorithm (the firefly algorithm) which is based on the light attractiveness of fireflies to each other in nature and finally the BA algorithm (the bat algorithm) which is based on the process of hunting micro bats through echolocation.

And the second part concerns the resolution of a constrained real mechatronic optimization problem, it is about a numerical computation by the algorithms which one discussed in the first part, of the optimal geometry of a support of a marketing plate, under well-defined constraints, this support must guarantee mechanical resistance against compressive and buckling forces generated by the plate, and according to the results obtained by each algorithm we will deduce the optimal solution of the problem.

MOTS-CLÉS. algorithme métaheuristique, optimisation, SSA, FA, GWO, CSA, BA, mécatronique.

KEYWORDS. Meta-heuristic algorithms, optimization, SSA, FA, GWO, CSA, BA, mechatronics.

1. Introduction

Chaque être vivant a sa propre façon de vivre dans un essaim organisé par des lois qu'on l'appelle l'intelligence de l'essaim (SI), comme les abeilles, les fourmis, les oiseaux, les poissons, les loups... etc. [VAS 20]

Ce phénomène naturel a permis de développer de nouveaux algorithmes d'optimisation appelés algorithmes d'optimisation méta-heuristiques ; leur principe de fonctionnement est inspiré du comportement grouillant dans la nature de ces êtres vivants, que ça soit lors de leur processus de chasse ou bien leurs méthodes pour maximiser leur survie, ou leurs façon de s'organiser sous une hiérarchie sociale...[VAS 20]

En général, un algorithme d'optimisation est un processus mathématique informatique permettant de trouver le minimum ou le maximum d'une fonction objectif (problème d'optimisation). Les algorithmes méta-heuristiques traitent les problèmes d'optimisation comme une boîte noire, ils sont

caractérisés par des recherches aléatoires dans un domaine de recherche contrôlées par des distributions qui décrivent mathématiquement le déplacement naturel et aléatoire des êtres vivants dans la nature comme le vol de Lévy, le pas aléatoire, et choisir membre qui a la meilleure position dans le groupe de recherche (l'essaim) cette position dirige vers la solution du problème d'optimisation. [ELH 15] [ZEM 19]

Ces algorithmes ont montré leurs performances dans divers domaines d'optimisation, ils pouvaient conduire vers la résolution des problèmes d'optimisation qu'ils soient contraints ou non contraints, complexes, multimodales, discontinus, sans avoir de connaissances particulières sur la fonction objectif ou sur ses informations différentielles. [VAS 20] [ELH 15] [ZEM 19]

Les algorithmes abordés dans cet article sont :

- SSA (algorithme d'essaim des salpes)
- CSA (algorithme de recherche de coucou)
- GWO (algorithme d'optimisation du loup gris)
- FA (algorithme de luciole)
- BA (algorithme des chauves-souris)

2. Les algorithmes d'optimisation métaheuristiques

2.1. l'algorithme GWO

GWO (Grey Wolf optimisation algorithm) est une méthode d'optimisation métaheuristique, proposée par Seyd Ali Mirjalili, Seyed Mohammed Mirjalili et Andrew Lewis en 2014, c'est une méthode d'optimisation inspirée de la hiérarchie sociale et des techniques de chasse des loups gris dans la nature. [MIR 14] [WAN 19]

Les loups gris vivent en groupe ; chaque groupe est organisé par un ordre hiérarchique. La hiérarchie sociale des loups gris est subdivisée en quatre catégories, qui contiennent les loups alpha, les loups bêta, les loups delta et les loups oméga.

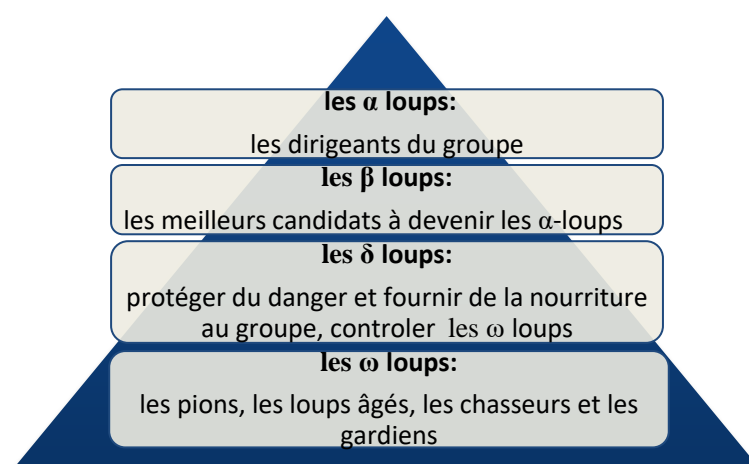


Figure 2.1. La hiérarchie des loups gris

L'algorithme GWO est le modèle mathématique du comportement social des loups gris, les nouvelles solutions générées par l'algorithme dans chaque itération sont générées en voisinage des positions du loup alpha X_1 (meilleure solution générée à l'itération précédente), du loup bêta X_2 (deuxième meilleure solution générée à l'itération précédente) et le loup delta X_3 (troisième

meilleure solution générée à l'itération précédente) en utilisant l'équation suivante : [MIR 14][WAN 19]

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad [1]$$

X_1, X_2, X_3 Sont calculés à l'aide des équations suivantes : [MIR 14] [WAN 19]

$$X_1 = |X_\alpha - A_1 D_\alpha|; D_\alpha = |C_1 X_\alpha - X(t)| \quad [2]$$

$$X_2 = |X_\beta - A_2 D_\beta|; D_\beta = |C_2 X_\beta - X(t)| \quad [3]$$

$$X_3 = |X_\delta - A_3 D_\delta|; D_\delta = |C_3 X_\delta - X(t)| \quad [4]$$

$X(t+1)$ Est la nouvelle position du loup oméga, $X(t)$ est la position actuelle du loup oméga; D_α , D_β et D_δ sont les voisinages de la position du loup alpha, du loup bêta et du loup delta respectivement.

A_i, C_i ($i=1, 2, 3$) sont des vecteurs de coefficients calculés comme suit: [MIR 14] [WAN 19]

$$A_i = 2a.r_1 - a \quad ; \quad r_1 \in [0,1] \quad [5]$$

$$C_i = 2a; \quad (i=1, 2, 3) \quad [6]$$

$$a = 2 - \frac{t}{Maxt} \quad [7]$$

Maxt est le nombre maximal d'itérations ; t est le numéro de l'itération actuelle.

- L'organigramme de l'algorithme GWO est représenté comme suit :

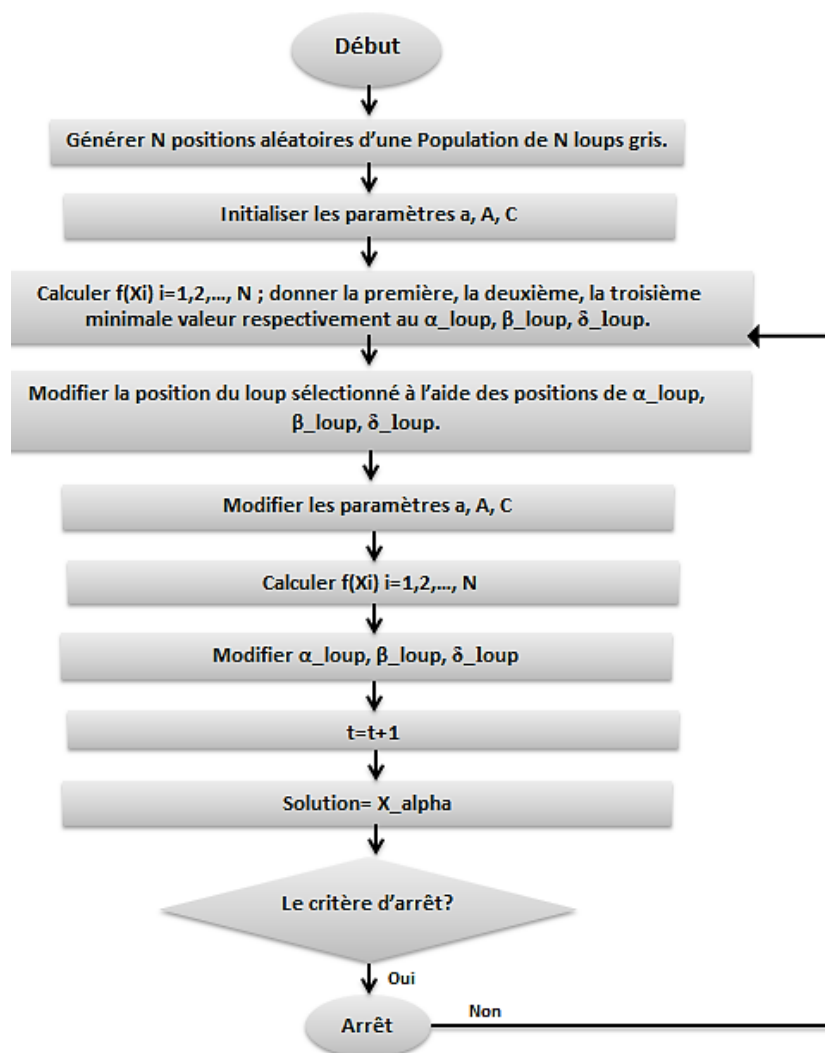


Figure 2.2. L'organigramme de l'algorithme GWO

2.2. L'algorithme SSA

SSA (Salp Swarm Algorithm) est une méthode d'optimisation métaheuristique, proposée par SeydAli Mirjalili, Seyed Mohammed Mirjalili, Amir H.Gandomi, Shahrazad Saremi, Seyedeh Zahra Mirjalili et Hossam Faris en 2017, c'est une méthode d'optimisation inspirée du mécanisme d'essaimage des salpes lors de la recherche de nourriture dans les océans. [MIR 17] [HEG 20]

Les salpes recherchent ensemble en formant des colonies linéaires ou circulaires dirigées par la salpe à l'avant de la chaîne, cette salpe est appelée la salpe leader, et les salpes restantes sont les suiveuses.

Ce mécanisme est transformé en modèle mathématique dans l'algorithme SSA par l'équation suivante :

1- Premièrement, la salpe leader pourrait changer de position selon l'équation suivante : [MIR 17] [HEG 20]

$$X_1^i = \begin{cases} A_i + r_1((X_{\max}^i - X_{\min}^i)r_2) + X_{\min}^i; r_3 \geq 0 \\ A_i + r_1((X_{\max}^i - X_{\min}^i)r_2) - X_{\min}^i; r_3 < 0 \end{cases} \quad i=1,2,\dots,D \quad [8]$$

$$\text{Avec : } r_1 = 2e^{-\frac{4t}{Max}} \quad [9]$$

r_1 et r_2 Sont deux nombres aléatoires entre 0 et 1, D est la dimension du problème. r_1 est un coefficient de l'algorithme SSA qui permet d'équilibrer entre le processus d'exploration et le processus d'exploitation.

Maxt est le nombre maximum d'itérations ; t est le numéro de l'itération actuelle. [MIR 17] [HEG 20]

X_1^i est la i-ème dimension de la position de la salpe leader, A_i est la i-ème dimension de la position de la proie, X_{min}^i et X_{max}^i sont les extremums de la i-ème dimension du domaine de recherche.

2- Et en ce qui concerne la position des salpes suiveurs, ils pourraient changer leur position selon l'équation suivante : [MIR 17] [HEG 20]

$$X_m^i = \frac{1}{2}(X_m^i + X_{m-1}^i); m = 1, 2, \dots, N \quad [10]$$

Avec : X_m^i et X_{m-1}^i sont les positions de deux salpes adjacentes ; N est le nombre des salpes dans la population.

- L'organigramme de l'algorithme SSA est représenté comme suit :

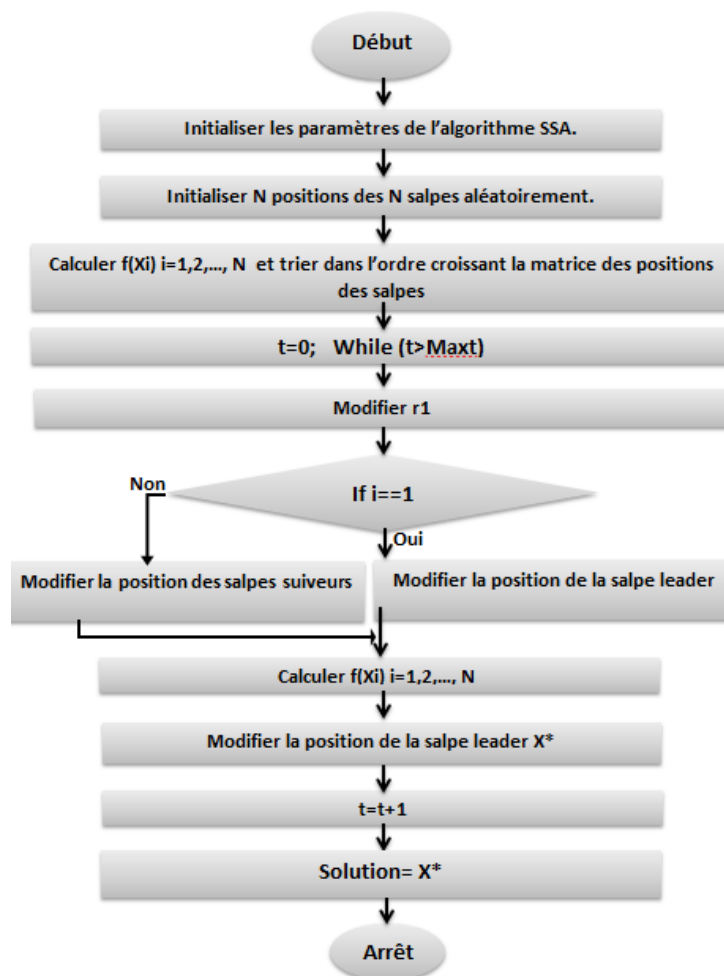


Figure 2.3. L'organigramme de l'algorithme SSA

2.3. l'algorithme CSA

CSA (Cuckoo Search Algorithm) est une méthode d'optimisation métaheuristique, proposée par Xin-She Yang et Suash Deb en 2009, c'est une méthode d'optimisation inspirée du comportement d'une espèce d'oiseaux appelés les coucous. [YAN 09]

Ces oiseaux pondent leurs œufs dans le nid d'oiseaux d'autres espèces, afin de maximiser leur survie et leur productivité ; ce comportement a conduit vers un nouvel algorithme appelé CSA. [PAY 05]

Les œufs sont les solutions du problème et l'œuf de coucou est la nouvelle solution générée, cette solution est calculée via la distribution Lévy Flight comme suit : [BRO 07] [PAV 07]

$$X_i(t+1) = X_i(t) + \alpha \oplus \text{levy}(\lambda) \quad [11]$$

$$\text{levy}(\lambda) = S * (X_i(t) - X_{best}) \quad [12]$$

$$i = 1, 2, \dots, N$$

α est une constante, S est la taille du pas, λ est l'exposant de Levy, et \oplus est un opérateur de multiplication d'entrée; X_{best} est la meilleure solution actuelle. [BRO 07] [PAV 07]

La taille du pas est calculée avec l'équation suivante : [BRO 07] [PAV 07] $S = \frac{\sigma_u * u}{|v|^{1/\beta}} \quad [13]$

u et v suivent une loi normale. [BRO 07] [PAV 07] $\sigma_u = \left(\frac{\Gamma(1+\beta) \cdot \sin(\pi \frac{\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \beta \cdot 2^{(\beta-1)/2}} \right)^{1/\beta} \quad [14]$

Lorsque l'oiseau hôte découvre l'œuf de coucou, cela signifie que la condition $r_1 < Pa$ est vraie, cet œuf de coucou est jeté (la nouvelle solution générée est modifiée) et il est remplacé par une nouvelle solution calculée comme suit : [YAN 09]

$$X_i(t+1) = X_i(t) + \text{rand} * (n_1 - n_2); n_1, n_2 = 1, 2, \dots, N \quad [15]$$

- L'organigramme de l'algorithme CSA est représenté comme suit :

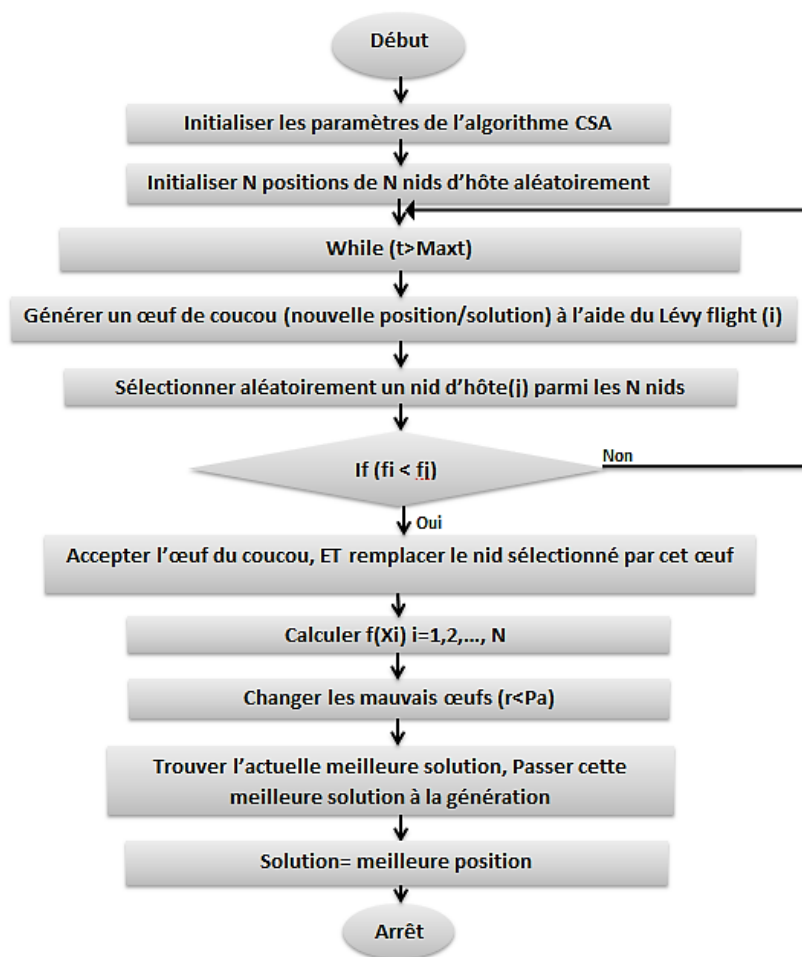


Figure 2.4. L'organigramme de l'algorithme CSA

2.4. L'algorithme FA

FA (firefly algorithm) est une méthode d'optimisation métaheuristique, proposée par Amir H.Gandomi et al. en 2011, c'est une méthode d'optimisation inspirée du phénomène d'attractivité des lucioles l'une par l'autre dans la nature. [AKH 19] [BRA 10]

Pour modéliser le comportement des lucioles dans la nature et construire l'algorithme FA, la position de chaque luciole est représentée avec une solution du problème, et la nouvelle position de chaque luciole est générée avec l'équation suivante : [AKH 19] [BRA 10]

$$X_i(t+1) = X_i(t) + \beta(X_j(t) - X_i(t)) + \alpha\varepsilon_i; i = 1, 2, \dots, N \quad [16]$$

$X_j(t)$ et $X_i(t)$ sont les positions de deux lucioles attirées l'une par l'autre, $\alpha\varepsilon_i$ est la taille du pas aléatoire de la luciole i et β est la fonction d'attractivité de la luciole calculée comme suit : $\beta = \beta_0 e^{-\gamma r^2}$ [17]

γ est le coefficient d'absorption de lumière, β_0 est la valeur de base du coefficient d'attraction, r est la distance entre les deux lucioles et elle est calculée comme suit : [AKH 19] [BRA 10] $r_{ij} = \sqrt{(X_j(t) - X_i(t))^2}$ [18]

2.4.1. L'organigramme de l'algorithme FA

L'organigramme de l'algorithme FA est représenté comme suit :

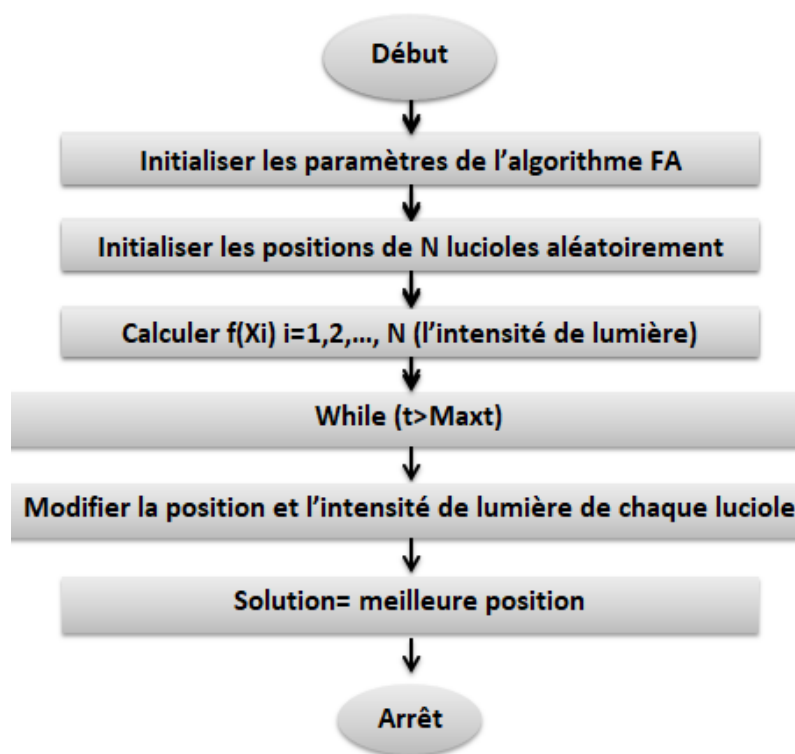


Figure 2.5. L'algorithme FA

2.5. L'algorithme BA

BA (Bats algorithm) est une méthode d'optimisation métaheuristique, proposée par Xin-She Yang en 2010, c'est une méthode d'optimisation basée sur le comportement d'écholocation des micro-chauves-souris dans le processus de chasse. Les micro-chauves-souris produisent une vibration sonar qui localise la position d'une proie dans l'espace de recherche, les micro-chauves-souris pourraient localiser une cible grâce à un écho reçu ce phénomène est appelé écholocation. [FIS 14] [ZEB 20]

Chaque micro chauve-souris est caractérisé par cinq variables, la fréquence f_i , la vitesse v_i , la position x_i , le volume du sonar A_i , le taux d'émission d'impulsions r_i . [FIS 14] [ZEB 20]

Ces variables sont calculées à l'aide des équations suivantes: [FIS 14] [ZEB 20]

- La fréquence: $f_i = f_{\min} + (f_{\max} - f_{\min})\beta$ [19]

Avec f_{\min} et f_{\max} dépendent de la nature du problème, β est un nombre aléatoire entre 0 et 1.

- La vitesse:

$$v_i(t+1) = v_i(t) + (X_i(t) - X_{best})f_i \quad [20]$$

$v_i(t+1)$ est la nouvelle vitesse du chauve-souris i , et $X_i(t)$ est sa actuelle position, X_{best} est l'actuelle meilleure position.

- La position: $X_i(t+1) = X_i(t) + v_i(t+1)$ [21]

- Le volume du Sonar:

Initialement il est calculé à l'aide de l'équation suivante: $A_i = rand \times (A_{\max} - A_{\min}) + A_{\min}$ [22]

Par la suite il décroît dans chaque itération selon l'équation suivante : $A_i = \alpha A_i^t$ [23]

- Le taux d'émission d'impulsions:

Initialement il est calculé à l'aide de l'équation suivante: $r_i = rand \times (r_{max} - r_{min}) + r_{min}$ [24]

Par la suite il croit dans chaque itération selon l'équation suivante : $r_i = r_i^t [1 - e^{-\gamma^t}]$ [25]

- L'organigramme de l'algorithme BA est représenté comme suit :

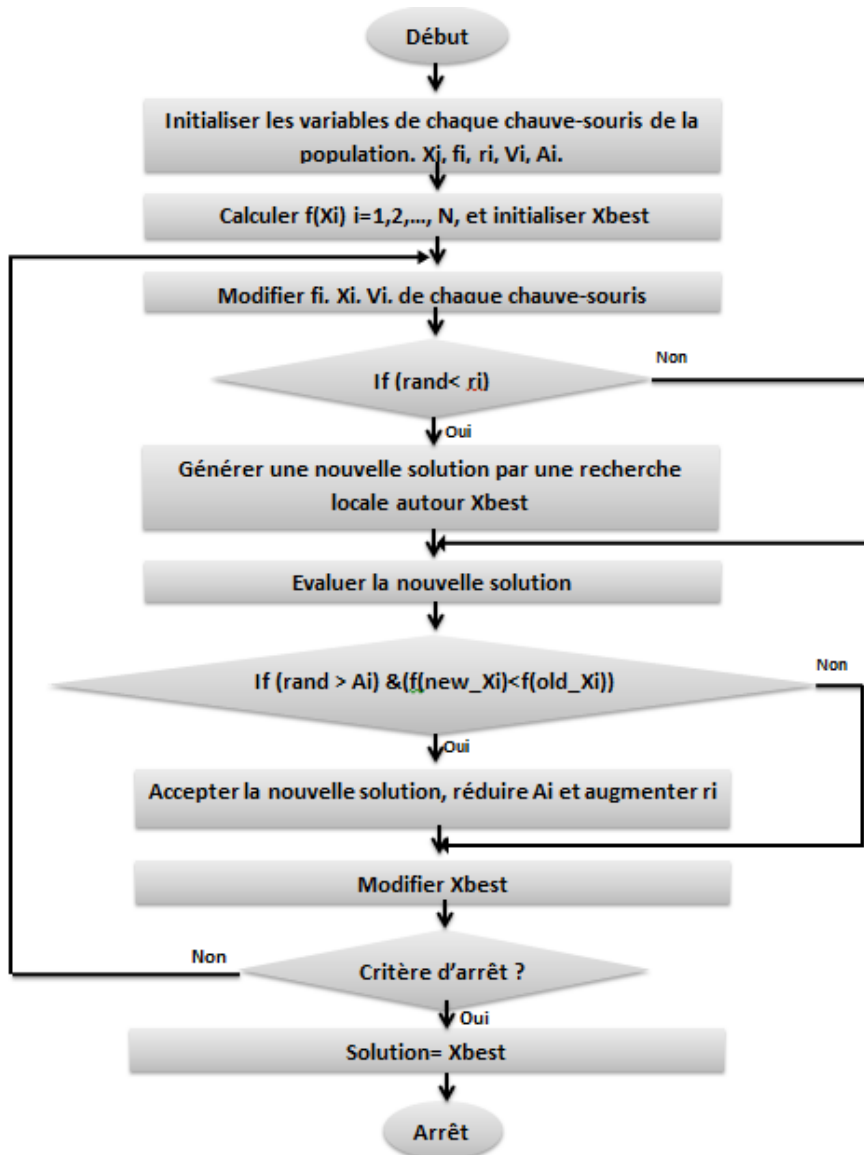


Figure 2.6. L'organigramme de l'algorithme BA

3. Matériel et méthode

Les algorithmes SSA, GWO, CSA, FA, BA sont élaborés à l'aide du Pseudo-code et des paramètres présentés dans le tableau (3.2) de chaque algorithme, ils sont développés et testés dans le logiciel Matlab.

Les algorithmes sont testés sur le même appareil (processeur : i7-6600U CPU 2.60GHz/2.81 GHz, système d'exploitation : 64bits).

4. Application

Un fabricant de panneaux de commercialisation veut produire un soutien sous forme d'un tuyau qui sera le support du panneau. Suite à une étude de marché, la hauteur souhaitée du support du panneau est $L=1,6$ m, le poids maximum à supporter est 132 Kg qui est équivalent d'un effort de $P=1294,92$ N.

Objectif: Optimiser la géométrie du tuyau. => Minimiser le coût => Minimiser la masse du tuyau.

▪ Modèle mécanique

Choix de matériau : Acier

Propriétés mécaniques	Module de Yong :	Limite d'élasticité :	Densité :
	$E=200$ GPa	$\sigma_{LE} = 294$ MPa	$\rho = 7870$ Kg / m ³

Tableau 3.1. Les propriétés mécaniques d'acier

Les efforts mécaniques que doit résister le tuyau :

✚ L'effort de compression :

Le poids de la plaque de commercialisation exige une pression sur l'aire transversale du tuyau ce qui engendre un effort de compression sur le tuyau, pour que le tuyau peut résister ces efforts il faut garantir que la pression appliquée soit inférieure à la limite maximale permmissible du matériau.

$$\text{On a : } \sigma_{\text{compression}} = \frac{P_{\text{maison}}}{S_{\text{trans}}} = \frac{P}{2\pi R t} \quad [26]$$

$$\text{Donc respecter la contrainte suivante : } \frac{P}{2\pi R t} \leq \sigma_{LE}$$

✚ L'effort de flambage

A cause de l'effort de compression soumis au tuyau, le tuyau peut s'échapper de cette charge et donc se fléchir et se déformer dans la direction perpendiculaire à l'axe de la compression.

Pour éviter ce phénomène la force de compression (P) doit être inférieure à la charge critique de flambage donnée par la formule d'Euler comme suivant :

$$F = \frac{\pi^2 EI}{(2L)^2} \quad [27]$$

Avec I est le moment quadratique du tuyau, la section transversale du tuyau est une section annulaire donc le moment du tuyau est donné par :

$$I = \pi R t \left(2R^2 + \frac{t^2}{2} \right) \quad [28]$$

$$\text{D'où la contrainte suivante : } P \leq \frac{E\pi^2 R^3 t}{(2L)^2}$$

Les équations de résistance de matériaux qu'on a utilisé sont valables pour les poutres donc pour que ces équations seront valides pour ce tuyau ses dimensions doivent respecter la contrainte suivante : $t \leq \frac{R}{10}$

▪ **Modèle mathématique :**

Problème :

On vise à minimiser la masse du tuyau=fonction objective :

$$\left\{ \begin{array}{l} \min M(R,t) = 2\pi R t * L * \rho = 2\pi R t * 1,6 * 7870 \\ \text{s.c :} \\ R \in [0.01;0.05]; t \in [0.001;0.005] \\ 1) 10 \leq \frac{R}{t} \leq 50 \\ 2) \frac{4L^2 P}{\pi^3 E} \leq R^3 t \leq 6,25 \times 10^{-7} \end{array} \right.$$

Variables de décision :

$R \in [0.01;0.05]$: Le rayon du tuyau (unité : m)

$t \in [0.001;0.005]$: L'épaisseur du tuyau (unité : m)

Résultats :

Les valeurs optimales de la masse, l'épaisseur et le rayon du tuyau obtenues par les algorithmes SSA, FA, CSA, BA, GWO sont présentés dans le tableau ci-dessous :

		Solveur	SSA	FA	CSA	BA	GWO
f_objective	Masse	1,0193Kg	1,0292069Kg	1,021296Kg	1,082332Kg	1,179999Kg	1,16369209Kg
variables	R	0,0129m	0,01297828m	0,011701m	0,012m	0,01433271m	0,01365297m
	t	0,001m	0,00100233m	0,0011032m	0,00114m	0,00104058m	0,001077297m

Tableau 3.2. Les résultats obtenus par les 5 algorithmes et le Solveur

On a défini la fonction objective (la masse du tuyau), les variables de décision du problème (le rayon et l'épaisseur) et les contraintes mécaniques et de fabrication qui contrôlent le choix des dimensions du tuyau par la suite on a exécuté les algorithmes d'optimisation.

Selon les résultats obtenues on peut conclure que les algorithmes SSA et FA ont pu convergé vers la solution optimale du problème et de trouver le tuyau le plus léger possible en tenant compte des contraintes qui va nous permettre de soutenir la plaque de commercialisation de 132Kg à 1,6m de hauteur .

Conclusion

Cet article avait présenté les cinq algorithmes d'optimisation métaheuristique, l'algorithme SSA qui se base sur le déplacement linéaire des salpes dans les océans pour trouver leur cible, l'algorithme CSA qui se base sur le comportement des coucous pour trouver et pondre leurs œufs

dans les nids d'hôtes, l'algorithme GWO qui est basé sur le comportement des loups gris dans la nature à travers une hiérarchie sociale, l'algorithme FA qui est basé sur l'attractivité à la lumière des lucioles les unes envers les autres dans la nature et enfin le BA qui est basé sur le processus de chasse des micro chauves-souris à travers l'écholocation.

Par la suite ces algorithmes avaient été développés dans le logiciel Matlab, et afin d'évaluer chaque algorithme on les a testé sur une application, il s'agit d'un problème mécanique d'optimisation réel contraint, l'objectif de ce problème est de trouver la géométrie optimale d'un support d'une plaque de commercialisation, ce support doit garantir une résistance mécanique contre des efforts de compression et de flambage engendrés par la plaque sous des contraintes bien déterminées, et selon les résultats obtenues par chaque algorithme on va déduire la solution optimale du problème.

L'algorithme FA a trouvé le minimum global de la fonction dans un premier temps par les itérations par rapport aux autres algorithmes, et à chaque fois il a trouvé la solution, mais l'algorithme FA prend le plus de temps pour calculer la solution, l'algorithme SSA a trouvé d'autre part le minimum global de la fonction après quelques itérations par rapport aux autres algorithmes, et à chaque fois qu'il a trouvé la solution avec un taux de réussite plus élevé, l'algorithme SSA avait également le troisième meilleur temps de calcul de la solution, et l'algorithme GWO est le dernier algorithme qui a trouvé le minimum global de la fonction par les itérations par rapport aux autres algorithmes avec un taux de réussite inférieur, de plus l'algorithme GWO est l'algorithme le plus rapide, il faut le minimum de temps pour calculer la solution.

De ces résultats on déduit que l'algorithme SSA a montré le meilleur compromis de la convergence vers le minimum global de la fonction par les itérations et un temps de calcul minimum.

Bibliography

- [ELH 15] N. EL HAMI, M. ITMI, A. EL HAMI, « Simulation and Optimization of an Actuator », *Advanced Materials Research*, Vol 1099, Apr. 2015.
- [ELH 15] N. EL HAMI, M. ITMI, A. EL HAMI, « Hybrid Evolutionary Optimization Algorithm for Structures », *Advanced Materials Research*, Vol 1099, Apr. 2015.
- [ZEM 19] Zemzami, Maria, El hami, Norelislam, Itmi Mhamed, Hmina, Nabil, « An evolutionary hybrid algorithm for complex optimization problems », *International Journal of Advanced Trends in Computer Science and Engineering*. (2019).
- [ZEM 19] Zemzami Maria «Variations sur PSO: approches parallèles, jeux de voisinages et applications. »(2019).
- [MIR 14] S. Mirjalili, S. M. Mirjalili, A. Lewis, « Grey Wolf Optimizer, *Advances in Engineering Software*», vol. 69, pp. 46-55, 2014, DOI: <http://dx.doi.org/10.1016/j.advengsoft.2013.12.007>.
- [WAN 19] Wang, JS., Li, SX, « An Improved Grey Wolf Optimizer Based on Differential Evolution and Elimination Mechanism», *Sci Rep* 9, 7181 (2019). <https://doi.org/10.1038/s41598-019-43546-3>
- [MIR 17] S. Mirjalili et al., « Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software* », (2017),6-9, <http://dx.doi.org/10.1016/j.advengsoft.2017.07.002>
- [HEG 20] Ah. E. Hegazy, M.A. Makhlof, Gh. S. El-Tawel, « Improved salp swarm algorithm for feature selection », *Journal of King Saud University - Computer and Information Sciences*, Volume 32, Issue 3, 2020, Pages 335-338, ISSN 1319-1578, <https://doi.org/10.1016/j.jksuci.2018.06.003>.
- [YAN 09] Yang X-S, Deb S, « Cuckoo search via Lévy flights», In: *Proceedings of World Congress on Nature & Biologically Inspired Computing*. IEEE Publications, USA, pp 210–214, (2009)
- [PAY 05] Payne RB, Sorenson MD, Klitz K, « *The Cuckoos* », Oxford University Press, New York, (2005)
- [BRO 07] Brown C, Liebovitch LS, Glendon R, « Lévy flights in Dobe Ju/hoansi foraging patterns », *Human Ecol* 35:129–138, (2007)
- [PAV 07] Pavlyukevich I, « Lévy flights, non-local search and simulated annealing », *J Comput Phys* 226:1830–1844

- [AKH 19] Akhoondzadeh, Mehdi & Azizi, Kamal, « Firefly Algorithm (FA) (2019).
- [BRA 10] Eds M. Bramer, R. Ellis, M. Petridis, « Research and Development in Intelligent Systems XXVI », Springer London, pp. 209-218 (2010).
- [FIS 14] Fister jr, Iztok & Fister, Iztok & Yang, Xin-She & Fong, Simon & Zhuang, Yanan, « Bat algorithm: Recent advances», CINTI 2014 - 15th IEEE International Symposium on Computational Intelligence and Informatics, Proceedings. 163-166. 10.1109/CINTI.2014.7028669.
- [ZEB 20] Amar Yahya Zebari, Saman M. Almufti, Chyavan Mohammed Abdulrahman, « Bat algorithm (BA): review, applications and modifications», 2-4, International Journal of Scientific World; Vol 8, No 1 (2020), <http://dx.doi.org/10.14419/ijsw.v8i1.30120>
- [VAS 20] A Vasuki, « Nature inspired optimization algorithms », London, 2020, 143-203.
- [MOM 13] Momin Jamil and Xin-She Yang, « A literature survey of benchmark functions for global optimization problems, Int. Journal of Mathematical Modelling and Numerical Optimisation», Vol. 4, No. 2, pp. 150–194 (2013). DOI: 10.1504/IJMMNO.2013.055204