

Optimisation du coût de l'emballage industriel par les algorithmes d'optimisations PSO, SA et GA

Optimization of the cost of industrial packaging by PSO, SA and GA optimization algorithms

Sara Rhouas¹, Norelislam El Hami¹

¹ Laboratoire Sciences et Ingénierie, Ecole Nationale des Sciences Appliquées, Université Ibn Tofail, Kénitra, Maroc

RÉSUMÉ. La métaheuristique connue sous le nom d'algorithme d'optimisation ; une résolution de problèmes difficiles de minimisation ou de maximisation d'une fonction afin de trouver des solutions quasi optimales. Il existe une grande variété de métaheuristiques, mais dans cet article de recherche, nous ne parlerons que de trois algorithmes d'optimisation qui vont nous aider à optimiser le cout de l'emballage d'une industrie en utilisant le logiciel MATLAB pour les programmer. Le premier algorithme est l'optimisation des essaims de particules le plus connu dans le domaine d'optimisation, qui est inspiré par le mouvement de simulation d'un groupe d'oiseaux, le second est le recuit simulé inspiré du recuit en métallurgie, une technique de traitement thermique impliquant également refroidissement contrôlé d'un matériau qui affecte à la fois la température et l'énergie. Et le dernier est l'algorithme génétique qui est couramment utilisé pour donner des résultats de haute qualité aux problèmes d'optimisation en s'appuyant sur des opérateurs bio-inspirés tels que la mutation, le croisement et la sélection. Nous comparerons la performance de chacun d'entre eux à l'aide des fonctions tests en fonction de leur durée de fonctionnement et de leur convergence et seront appliquer sur notre problème d'optimisation industriel.

ABSTRACT. The metaheuristic known as the optimization algorithm; a resolution of difficult problems of minimization or maximization of a function in order to find almost optimal solutions. There are a wide variety of metaheuristics, but in this research article we will only talk about three optimization algorithms that will help us optimize the cost of packaging an industry by using MATLAB software to program them. The first algorithm is the best-known particle swarm optimization in the optimization field, which is inspired by the simulation movement of a group of birds, the second is the simulated annealing inspired by annealing in metallurgy, a Heat treatment technique also involving controlled cooling of a material which affects both temperature and energy. And the last is the genetic algorithm which is commonly used to give high quality results to optimization problems by relying on bio-inspired operators such as mutation, crossing and selection. We will compare the performance of each of them using the test functions according to their uptime and convergence and will apply to our industrial optimization problem.

MOTS-CLÉS. Optimization, Industrie, Procédé de fabrication, Optimisation par essaims particulaires algorithme, Recuit simulé, Algorithme génétique.

KEYWORDS. Optimization, Industry, Manufacturing process, Particle Swarm optimization, Simulated annealing, Genetic algorithm.

1. Introduction

Une métaheuristique est le nom donné aux algorithmes de haut niveau utilisée en informatique et en optimisation mathématique pour trouver, générer ou fournir en recherche partielle heuristique, une solution suffisamment bonne aux problèmes d'optimisations et spécialement ceux dont les informations sont incomplètes, imparfaites ou ayant une capacité de calcul limitée, avec un ensemble de solutions trop vaste pour être échantillonné complètement. Sur certains problèmes classiques, les métaheuristiques ne garantissent pas la génération d'un optimum global. De nombreuses métaheuristiques utilisent l'optimisation stochastique, ayant une réponse qui est déterminée par un ensemble de variables aléatoires générées par l'optimisation combinatoire, ce qui implique une recherche parmi un grand nombre de solutions possibles [ELH 15].

Un problème d'optimisation complexe, qui contient énormément de solutions possibles et qui doit satisfaire une résolution optimale avec des contraintes ajustées au problème, Nous laisse souvent à utiliser une métaheuristique puissante, et qui sont en majorité des temps des métaheuristiques à

population afin de résoudre ce type de problème d'optimisation. Ces métaheuristiques peuvent souvent identifier de bons résultats avec moins de travail de traitement que les méthodes itératives ou les heuristiques de base. Dans la littérature la majorité des métaheuristiques ont des caractères expérimentaux basées sur des phénomènes naturels et sur la convergence et le potentiel d'atteindre l'optimum global. Le but d'une métaheuristique est d'explorer efficacement l'espace de recherche afin de trouver des solutions optimales, allant de simples procédures de recherche locale à des processus d'apprentissage complexes pour les méta-algorithmes qui sont approximatifs et généralement non déterministes.

L'objectif principal de cet article est d'étudier la performance de trois métaheuristiques dans le domaine industriel. Cet article est consacrée dans la première section à rappeler et décrire les trois algorithmes métaheuristiques, le premier est l'optimisation par essaims particulaires connu comme ayant une haute performance de résolution et inspiré par la simulation du comportement social des oiseaux, ensuite nous allons décrire le deuxième algorithme recuit simulé qui est également inspiré par un phénomène naturel qui est le recuit en métallurgie et enfin nous allons brièvement introduire l'algorithme génétique qui se base sur le phénomène génétique dans la nature et qui fait partie des algorithmes évolutionnaires. La deuxième section est consacrée à montrer les performances de ces algorithmes avec l'aide de certaines fonctions de test. Et la dernière section est consacrée à l'application de ces algorithmes sur une problematique industriel.

1.1. Algorithmes d'optimisation

1.1.1. Optimisation par l'algorithme d'essaims de particules

La méta-heuristique d'optimisation par essaims de particules ou en anglais Particle Swarm Optimization (PSO) a été développée par deux scientifiques le socio-psychologue James Kennedy et l'ingénieur électricien Russel Eberhart en 1995 pour la résolution des problèmes à variables continues. Cet algorithme a été modélisé en se basant sur le comportement social ou pour plus de précision sur les interactions sociales entre des "agents" nommées "particules" et qui représente un "essaim" [KEN 95-EBE 01].

L'optimisation par essaim particule optimise les problèmes en utilisant l'itération de la solution candidate, puis nous changeons la position et la vitesse et vérifions à quel point nous sommes proches pour résoudre le problème et pour connaître la meilleure solution, puis mettons à jour l'autre particule au fur et à mesure que nous optimisons le problème, cet algorithme peut rechercher un grand espace de solutions candidates [KEN 02-ELH 10]. Tout d'abord, l'algorithme commence par générer des particules initiales avec leurs vitesses aléatoirement comme vous pouvez le voir dans l'organigramme (figure 1), afin d'initialiser une population dans notre domaine de recherche. Ensuite, à l'intérieur d'une boucle qui se répète jusqu'à convergence on détermine les meilleures positions de la particule, du voisin et aussi de l'essaim qui va nous permettre de mettre à jour la vitesse et la position de chaque particule pour trouver la meilleure solution dans notre domaine de recherche [MAN 19].

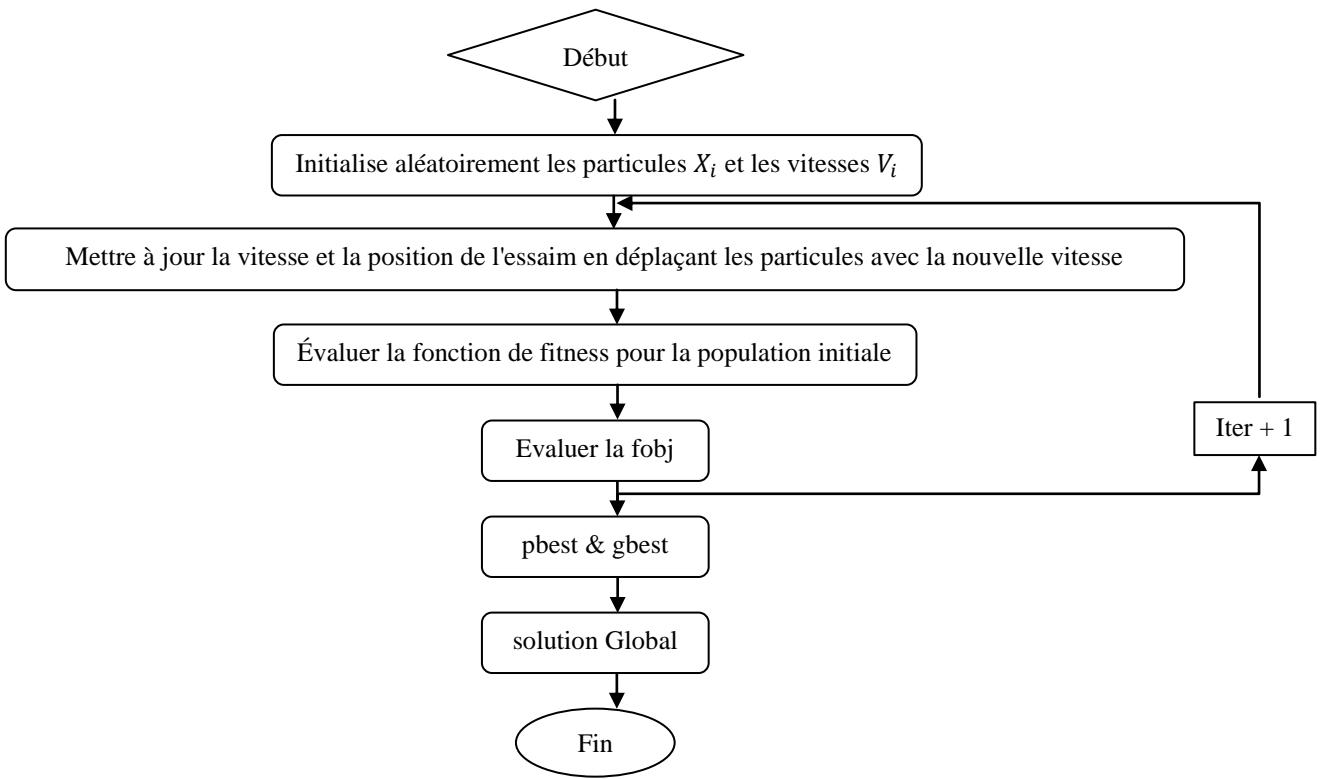


Figure 1. Organigramme PSO

1.1.2. Algorithme du recuit simulé

Le recuit simulé ou Simulated Annealing (SA) est un algorithme permettant de résoudre les problèmes d'optimisation limités. Cette méthode se base sur le processus physique de métallurgie qui consiste de chauffer un matériau, ensuite de diminuer lentement la température pour minimiser les défauts, qui permet de diminuer ainsi l'énergie du système [MIN 08]. Dans cet algorithme cette lente diminution de la température va permettre au système d'explorer le voisinage de tout notre domaine de recherche afin de trouver le meilleur optimum global avant de refroidir et s'arrêter sur l'un des optimums locaux [YUE 18-LAM 08].

L'organigramme figure 2 nous montre qu'au début de l'algorithme on génère la population initiale aléatoirement appartenant à notre domaine de recherche, puis on initialise la meilleure solution de notre population et on fixe la température de départ. À chaque itération de l'algorithme de recuit simulé, on génère un nouveau point qui sera comparer avec le point actuel et qui sera accepté si elle est meilleure avec une certaine probabilité dite Boltzmann pour éviter d'être coincé sur les optimums locaux et d'explorer globalement pour trouver optimum global [GHO 17-PO 09].

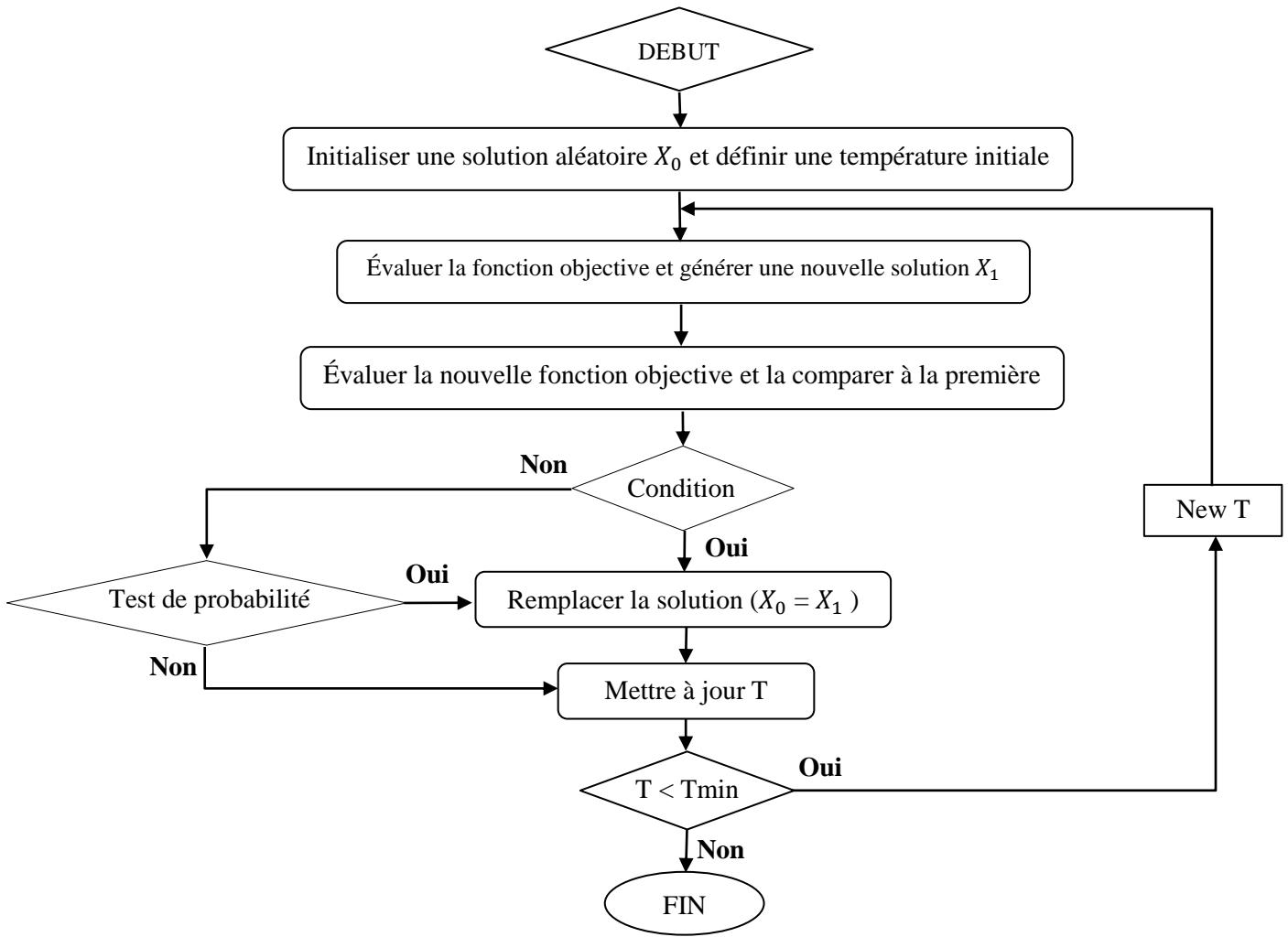


Figure 2. Organigramme SA

1.1.3. Algorithme Génétique

L'algorithme génétique est un algorithme évolutionnaire qui a pour but de trouver une solution approchée des problèmes d'optimisation dont la solution est inconnue dans un temps raisonnable. De nos jours les scientifiques et les ingénieurs ont un énorme volume de données, et qu'ils ont besoin d'une solution dans le plus bref délai c'est pourquoi ils ont créé cet algorithme basé sur l'évolution génétique dans la nature. L'algorithme génétique ne vérifie pas toutes les possibilités mais c'est un moyen rapide de trouver une très bonne solution pour des problèmes qui ont beaucoup de données [LILI 21- IVAN 19].

L'algorithme génétique se résume sur trois étapes essentielles sélection-crossover-mutation comme vous pouvez le constater dans l'organigramme figure 3. D'abord une population de données est générée aléatoirement dans l'espace de recherche chaque individu appelés chromosome, a ses propre caractéristiques nommés gènes. Ensuite, nous passons à la première étape essentielle de l'algorithme qui est la sélection qui contient deux critères de sélection ; l'ordonnancement qui consiste de classifier la population par ordre de performance en utilisant la fonction de fitness de la problématique étudiée pour évoluer les chromosomes, et la roue de fortune qui consiste à attribuer pour chaque chromosome une probabilité proportionnelle à sa fonction d'adaptation [XUN 11-TRU 15]. Après avoir choisi les meilleurs parents de notre population on passe à la deuxième étape qui est le croisement et qui consiste de générer deux nouveaux points en échangeant un certain nombre de gènes entre les parents chromosomes en se basant sur une probabilité de croisement. Enfin nous passons à la dernière étape de l'algorithme génétique qui est la mutation qui consiste de faire changer aléatoirement certaines caractéristiques ou certain gène du nouveau chromosome en se

basant sur la probabilité de mutation afin de créer la nouvelle génération, ce processus se répète jusqu'à atteindre le nombre de génération fixé ou atteindre l'optimum souhaité [FAR 15].

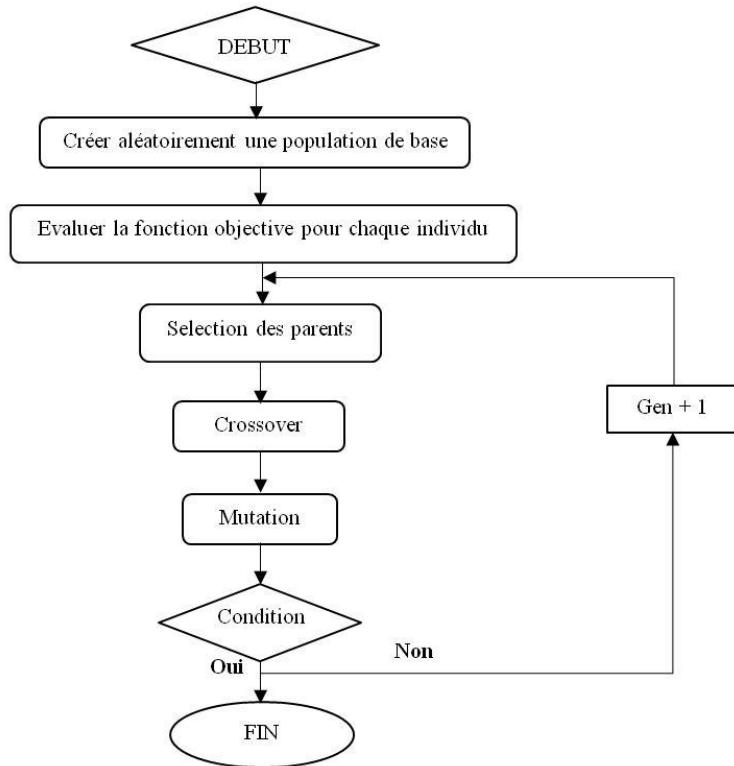


Figure 3. Organigramme GA

2. Résultats de la performance des algorithmes d'optimisation

2.1. Simulation des fonctions tests

Cette simulation a été appliquées sur un ordinateur HP Processeur Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz, 2.40 GHz , 8,00 Go, 64 bits, processeur x64 sur le logiciel Matlab version R2018a.

En utilisant les fonctions tests définies sur le tableau 1 nous allons comparer la performance des trois algorithmes d'optimisation étudié au début PSO, SA et GA avec une précision d'erreur acceptée de 0.01, en se basant sur trois caractéristiques de performance le taux de réussite, le nombre d'évaluation et temps d'exécution qui sont présentait sur la partie suivante.

Fonction Test	Nom de la Fonction	Domaine de recherche	Résultats de la fonction objectif souhaités	Nombre de dimension Xi
BO1	Bohachevsky1	[-10,10]	0	2
BO2	Bohachevsky2	[-10,10]	0	2
BR	Branin	[-5,15]	0.397887	2
CA	Camel	[-5,5]	-1.0316	2
CM	Cosine_Mixture	[-1,1]	-0.4	4
DE	Dejoung	[-5.12,5.12]	0	3
GP	Goldstein_Price	[-2,2]	3	2
GR	griewank	[-600,600]	0	50
HA3	Hartman3	[0,1]	-3.862782	3
HA6	Hartman6	[0,1]	-3.322368	6
HAN	Hansen	[-10,10]	-176.541793	2
RA	Rastrigin	[-5.12,5.12]	0	2
RO	Rosenbrock	[-10,10]	0	10
SH3	Shubert 3	[-10,10]	-2709.07	3
SH4	Shubert 4	[-10,10]	-39303.17	4
SK5	Shekel4_5	[0,10]	-10.1532	4
SK7	Shekel4_7	[0,10]	-10.40294	4
SK10	Shekel4_10	[0,10]	-10.53641	4

Tableau 1. Paramètres des fonctions tests

Sur la figure 4 nous avons les résultats en forme d'histogramme du taux de succès des trois algorithmes d'optimisations PSO, SA et GA, et vous pouvez remarquer que l'algorithme d'optimisation par essaim de particules a le plus grand nombre de succès sur les 18 fonctions tests et en majorité d'un taux de 100% de succès de trouver l'optimum global sur la majorités des fonctions tests, et la suit l'algorithme du recuit simule sauf pour le cas des fonctions tests RA SK5 SK7 et SK10 que l'algorithme génétique a été plus performant.

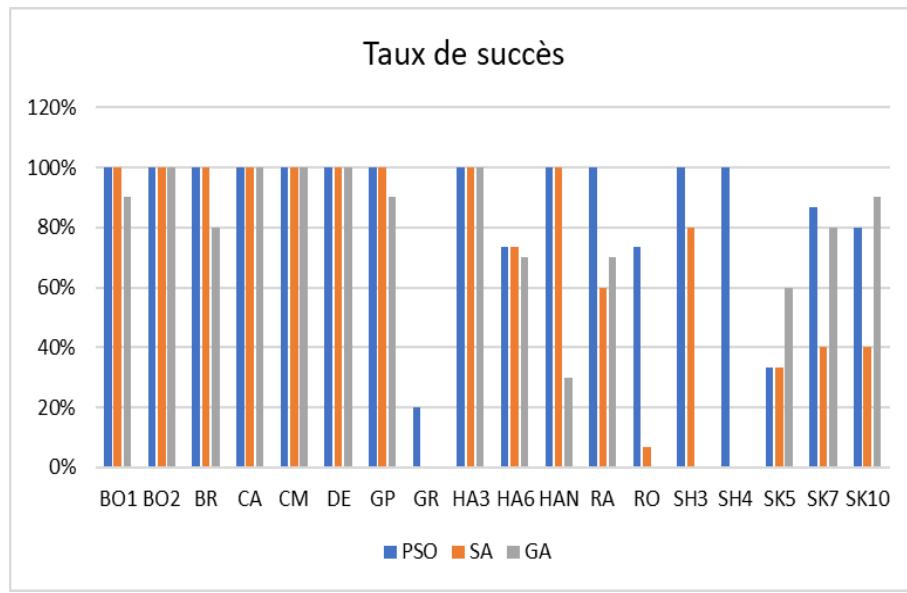


Figure 4. Taux de succès des algorithmes

Sur la figure 5 nous avons l'histogramme de la durée en seconde qui a fallu à chaque algorithme pour trouver la solution optimale, et comme vous pouvez le remarquer l'algorithme du recuit simulé prend plus de temps à chercher de solutions par rapport aux deux autres algorithmes et la suit avec une grande marge de différence l'algorithme d'optimisation par essaim de particules. Et ensuite nous pouvons voir l'algorithme génétique en 3ème place vu qu'elle est connu comme étant un moyen rapide pour avoir des résultats c'est pour cela qu'on remarque qu'il a le plus petit temps d'exécution.

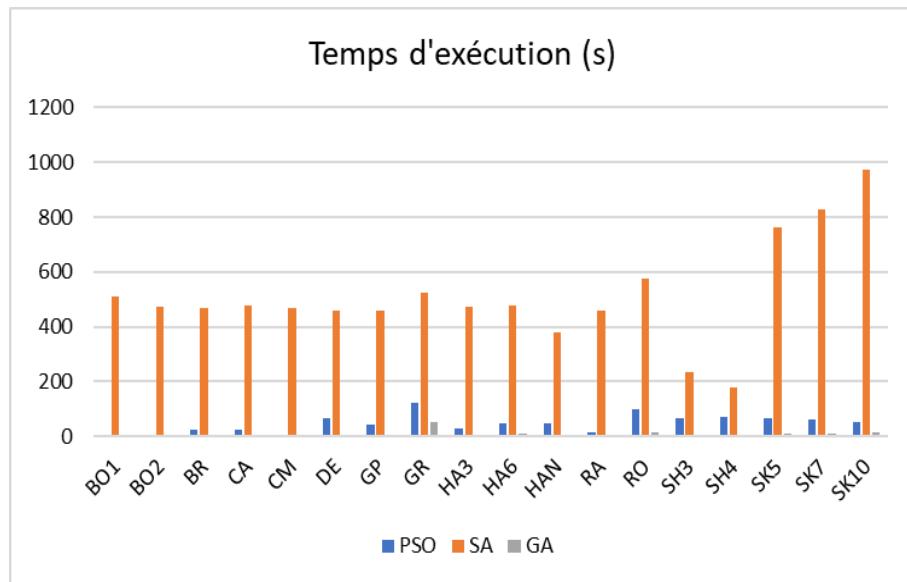


Figure 5. Temps d'exécution

Sur la figure 6 nous avons étudiez le nombre d'évaluation totale des trois algorithmes de recherches PSO, SA et GA pour les 18 fonctions test. On remarque que l'algorithme du recuit simulé a le plus grand nombre d'évaluation de la fonction objectif et le suit avec une grande différence l'algorithme d'optimisation par essaim de particules.

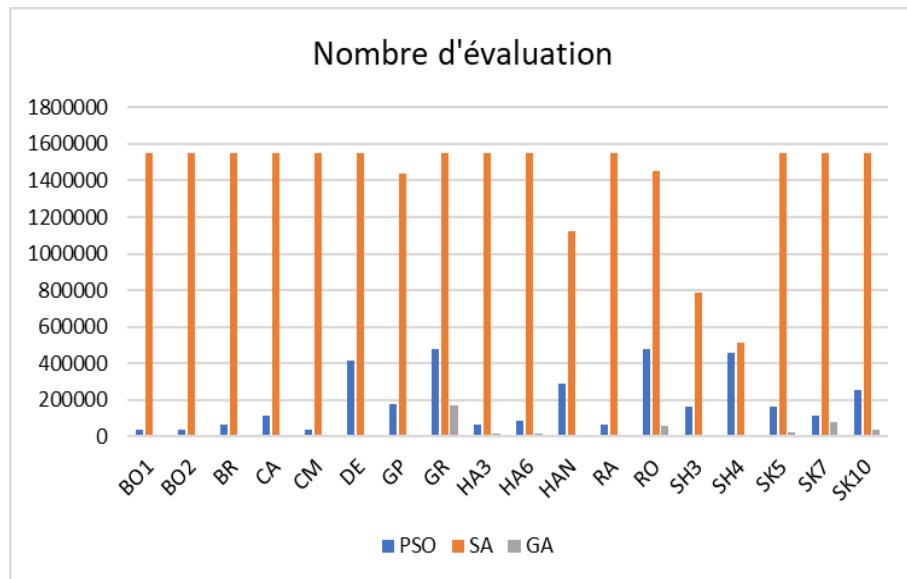


Figure 6. Nombre d'évaluation

Vu les résultats obtenus dans la simulation des algorithmes précédemment, on peut constater que l'algorithme d'optimisation par essaim particule PSO a des meilleurs résultats donc une meilleure performance contrairement aux autres algorithmes. Par contre pour les autres algorithme nous remarquons que SA a des meilleurs résultats comparés à l'algorithme génétique par rapport aux taux de succès, par contre nous pouvons apercevoir que le temps d'exécution et le nombre d'évaluation est plus grande chez l'algorithme du recuit simule vu que l'algorithme génétique est un algorithme qui permet d'avoir des résultats proche de la solution optimale dans les plus bref délai c'est pour cela qu'il a le plus court temps d'exécution parmi les trois algorithmes mais il a le plus faible taux de succès.

3. Application

Une industrie de fabrication d'un nouveau produit qui se caractérise de 28cm de longueur et qui veut concevoir son emballage en forme de boite rectangulaire, et qui ne doit pas dépasser 31cm de longueur afin de limiter le mouvement du produit. Notre objectif est de créer avec un cout optimal, une boite sans couvercle avec un volume qui ne dépasse pas 27000cm^3 et que ces trois dimensions ne doivent pas être inférieure a 1cm, et en utilisant un matériau spécial pour les cotes de la boite pour mieux protéger le produit dont le cout est de $0.0012\text{€}/\text{cm}^2$ et le reste avec un autre matériau conventionnel avec un cout de $0.0003\text{€}/\text{cm}^2$. Et aussi, avec un volume qui ne dépasse pas 27000cm^3 et que ces trois dimensions ne doivent pas être inférieure a 1cm [SIM 16].

Modélisation de la fonction objective :

$$\begin{cases} \min(X_{Lar}, X_{Lon}, X_{Hau}) = 0.0012 * (2*X_{Lar} * X_{Lar}) + 0.0003 * (2*X_{Lon} * X_{Hau} + X_{Lar} * X_{Lon}) \\ 28 \leq X_{Lon} \leq 31 \\ X_{Lar} * X_{Lon} * X_{Hau} \geq 27000 \\ X_{Lar}, X_{Lon}, X_{Hau} \geq 1 \end{cases}$$

Avec :

X_{Lar} : Largeur de la boite.

X_{Lon} : Longueur de la boite

X_{Hau} : Hauteur de la boite.

Dimensions	PSO	SA	GA	Solveur
Largeur	41,9715	40,2471	41,7309	41,74
Longueur	30,9843	30,8841	31	31
Hauteur	20,775	21,7807	20,8711	20,87
Cout	2,869	2,8804	2,8666	2,87

Tableau 2. Résultats de l'application sur les algorithmes d'optimisation

Le tableau 1 contient les résultats de minimisation du cout total de l'emballage d'un produit en utilisant les trois algorithmes d'optimisation, nous pouvons conclure que ces algorithmes traite plus de possibilité comparée à celle d'un solveur simple, et que c'est pour cela quel dure plus de temps à trouver une solution vu qu'elles visent les problèmes les plus complexe et prend plus de temps à engendrer et évaluer les solutions possibles. En plus nous pouvons conclure que l'algorithme par essaims des particulaires a eu le meilleur résultat et le plus précis comparée aux autres algorithmes d'optimisation.

4. Conclusion

La métaheuristique est un terme qui représente une grande variété d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile lorsque les techniques exactes s'avèrent insuffisantes et qui sont souvent issus des domaines de recherches opérationnelle, l'ingénierie ou de l'intelligence artificielle. Ces métaheuristiques sont dans la plupart du temps des algorithmes stochastiques qui progressent vers un optimum global, quelques-uns sont inspirés par des phénomènes naturels utilisant des méthodes implicites explicites ou directe et d'autres qui contient ou non une mémoire. Dans cet article nous nous sommes concentré sur trois algorithmes d'optimisation qui sont l'optimisation par essaims de particules, le recuit simulé et l'algorithme génétique nous avons étudié leur performance de chacun d'entre eux sur Matlab en utilisant des fonctions tests les plus connu en littérature et après comparaison nous avons constaté que l'algorithme d'optimisation par essaims particulaires a des meilleurs résultats et une meilleure performance comparée aux autres algorithmes même s'il n'est pas le plus rapide par rapport à l'algorithme génétique mais vu sa performance et le nombre d'évaluation son retard est justifiée vu que l'algorithme PSO engendre plus de solution potentiel à chaque itération et prend en considération tous les optimums de chaque particule et voisin et aussi dans l'ensemble et cela pour trouver l'optimum global contrairement à l'algorithme génétique qui se base plus sur le croissement des optimums potentiels sur un nombre de génération précis. Nous avons pu aussi remarquer sa performance dans le domaine industriel vu que l'algorithme PSO nous a donné le meilleur optimum donc le meilleur cout d'emballage comparé aux deux autres algorithmes d'optimisation.

Acknowledgement : Rhouas Sara remercie le Centre National de la Recherche Scientifique et Technologique (CNRST) de lui avoir accordé une bourse nationale de recherche doctorale.

Bibliographie

- [ELH 15] N. EL HAMI, M. ITMI, A. EL HAMI, « Simulation and Optimization of an Actuator », *Advanced Materials Research*, Vol 1099, pp. 110-119, Apr. 2015.
- [ELH 15] N. EL HAMI, M. ITMI, A. EL HAMI, « Hybrid Evolutionary Optimization Algorithm for Structures », *Advanced Materials Research*, Vol 1099, pp. 102-109, Apr. 2015.
- [KEN 95] J. KENNEDY AND R. EBERHART, « Particle Swarm Optimization », *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 8, no. 3, pp. 1943-1948, 1995.
- [EBC 01] R.C. EBERHART AND Y. SHI, « Particle Swarm Optimization: Developments Applications and Resources», Proc. IEEE Congr. Evolutionary Computation, vol. 1, pp. 27-30, 2001.
- [KEN 02] J. KENNEDY AND R. MENDES « Population Structure and Particle Swarm Performance », *Proceedings of the IEEE Congress on Evolutionary Computation*, 2002.
- [ELH 10] EL HAMI, N., ELLAIA, R., ITMI, M, « Hybrid evolutionary optimization algorithm MPSO-SA», *International Journal for Simulation and Multidisciplinary Design Optimization*, 4(1), 27-32, 2010.
- [MAN 19] MANOELA KOHLER, MARLEY M.B.R. VELLASCO, RICARDO TANSCHEIT, « PSO+: A new particle swarm optimization algorithm for constrained problems», *Applied Soft Computing*, Volume 85, 2019.
- [MIN 08] MIN-CHIE CHIU, YING-CHUN CHANG, « Shape optimization of multi-chamber cross-flow mufflers by SA optimization», *Journal of Sound and Vibration*, Volume 312, Issue 3, Pages 526-550, 2008.
- [YUE 18] YUEFENG QI, CAILING LI, PENG JIANG, CUI JIA, YANYAN LIU, QI ZHANG, « Research on demodulation of FBGs sensor network based on PSO-SA algorithm », *Optik*, Volume 164, Pages 647-653, 2018.
- [LAM 08] L. LAMBERTI, «An efficient simulated annealing algorithm for design optimization of truss structures», *Computers & Structures*, Volume 86, Issues 19–20, Pages 1936-1953, 2008.
- [GHO 17] GHOLAMHOSSEIN SODEIFIAN, SEYED ALI SAJADIAN, NEDASADAT SAADATI ARDESTANI,« Experimental optimization and mathematical modeling of the supercritical fluid extraction of essential oil from Eryngium billardieri: Application of simulated annealing (SA) algorithm », *The Journal of Supercritical Fluids*, Volume 127, Pages 146-157, 2017.
- [PO 09] PO-HAN CHEN, SEYED MOHSEN SHAHANDASHTI, « Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints», *Automation in Construction*, Volume 18, Issue 4, Pages 434-443, 2009.
- [LILI 21] Lili Xu, « Research on computer interactive optimization design of power system based on genetic algorithm», *Energy Reports*, Volume 7, Supplement 7, Pages 1-13, 2021.
- [HYU 19] HYUKGEUN CHOI, JINHYUN KIM, YOURIM YOON, BYUNG-RO MOON, « Investigation of incremental hybrid genetic algorithm with subgraph isomorphism problem», *Swarm and Evolutionary Computation*, Volume 49, Pages 75-86, 2019.
- [IVAN 19] IVAN VLAŠIĆ, MARKO ĐURASEVIĆ, DOMAGOJ JAKOBOVIĆ,, « Improving genetic algorithm performance by population initialisation with dispatching rules », *Computers & Industrial Engineering*, Volume 137, 106030, 2019.
- [XUN 11] XUNBO SHUAI, XIANGGUANG ZHOU, « A Genetic Algorithm Based on Combination Operators », *Procedia Environmental Sciences*, Volume 11, Part A, Pages 346-350, 2011.
- [TRU 15] MS. TRUPTI BHOSKAR, MR. OMKAR K. KULKARNI, MR. NINAD K. KULKARNI, MS. SUJATA L. PATEKAR, G.M. KAKANDIKAR, V.M. NANDEKAR, « Genetic Algorithm and its Applications to Mechanical Engineering: A Review », *Materials Today: Proceedings*, Volume 2, Issues 4–5, Pages 2624-2630, 2015.
- [FAR 15] FARSHAD KIYOMARSI, « Mathematics Programming based on Genetic Algorithms Education », *Procedia - Social and Behavioral Sciences*, Volume 192, Pages 70-76, 2015.
- [SIM 16] DR. SIMON TAMAYO, M. RONAN CARDIN, DR. SEBASTIAN ECHEVERRI, DR. DANIEL PINO, « Recherche opérationnelle appliquée à la gestion industrielle : Apprentissage par l'exemple basé sur l'utilisation de logiciels tableurs de calcul - Un manuel de mathématiques appliquées aux problèmes de gestion des opérations et de la supply chain », 2016.