

Application d'un modèle parallèle de la méthode PSO au problème de transport d'électricité

Electrical power transmission optimization based on a new version of PSO algorithm

Maria Zemzami¹, Norelislam Elhami², Abderahman Makhloufi³, Mhamed Itmi⁴, Nabil Hmina⁵

^{1,4} LITIS, Université de Normandie, INSA, Rouen, maria.zemzami@gmail.com, abderahman.makhloufi@insa-rouen.fr, mhamed.itmi@insa-rouen.fr

^{2,5} LGS, Université Ibn Tofail, ENSA, Kenitra, Maroc, norelislam@outlook.com, hmina@univ-ibntofail.ac.ma

RÉSUMÉ. Reconnue pendant plusieurs années comme une métaheuristique stochastique performante dans la résolution des problèmes d'optimisation difficiles, la méthode d'optimisation par essaim particulière « PSO » présente cependant des points faibles : le temps de calcul considérable et la convergence prématurée. Plusieurs études ont été menées pour trouver le jeu de paramètres qui conduit à de bonnes performances de l'algorithme.

Dans ce papier, nous proposons une version de l'algorithme PSO, permettant d'améliorer ses performances en introduisant sa parallélisation associée à la notion du voisinage évolutif.

L'algorithme proposé a été testé afin d'améliorer la performance et la fiabilité des structures mécaniques « le problème de transport d'électricité » ; plus précisément l'optimisation de la durée de vie du pylône d'une ligne de transport d'électricité ; l'objectif est de maximiser la résistance à la charge tout en réduisant le coût « la minimisation de l'utilisation des matériaux ».

Dans nos expérimentations, les tests effectués sur le programme ont donné des résultats satisfaisants du modèle parallèle par rapport au modèle séquentiel.

ABSTRACT. In this paper, we suggest a new version of PSO algorithm, that allows the amelioration of its performance by introducing its parallelization associated to the concept of evolutionary neighborhood. The main objective of our approach is to overcome to the two essential disadvantages of PSO: high running time and premature convergence. The proposed algorithm was tested in order to improve the performance and reliability of mechanical structures; more precisely on the electricity pylon example; the objective is to maximize resistance to load while reducing material usage and cost. Experimental results demonstrate that the proposed method is effective and outperforms basic PSO in terms of solution quality, accuracy, constraint handling, and time consuming.

MOTS-CLÉS. Optimisation, PSO, Parallélisation, Voisinage évolutif, Transport d'électricité, Pylône électrique.

KEYWORDS. Optimization, PSO, Parallelization, Evolutionary neighborhood, Electrical transmission, Electricity pylon.

1. Introduction

Suite à l'évolution continue des ressources matérielles dans le domaine informatique, les ordinateurs ont vu leur nombre de processeurs/cœurs augmentés ces dernières années afin de compenser les limites d'augmentation de puissance d'un seul processeur et d'obtenir un facteur d'accélération, puisqu'avec plus de puissance de calcul on pourrait résoudre plus rapidement un problème. Afin d'exploiter pleinement cette puissance de calcul, il convient de réaliser des applications capables d'effectuer plusieurs tâches en parallèle (processus Unix, threads Java par exemple). Les threads, sont la technologie utilisée en Java pour rendre les applications multitâches. Nous nous sommes intéressés à cette technologie afin de profiter des avantages du parallélisme en terme de diminution du temps de calcul, et de la bonne exploitation des ressources matérielles de la machine.

La méthode PSO est une méta-heuristique qui permet de trouver l'optimum d'une fonction en un temps de traitement raisonnable, sauf pour les instances de grande taille où le calcul scientifique est intensif ce qui nécessite un temps de calcul considérable. L'utilisation de modèles parallèles appropriés diminue le temps de calcul et donne des résultats plus satisfaisants que les modèles séquentiels [ZHO 09], [WAI 00]. Fuir la convergence prématurée de la méthode, est aussi un point primordial sur lequel plusieurs chercheurs ont mené leurs études et ont proposé plusieurs versions [TIN 03], [LIU 07], [PAQ 03], [PAR 01].

Le modèle que nous proposons dans ce papier, est une version basée sur l'algorithme PSO utilisant des threads pour le calcul parallèle, et une notion de voisinages dynamiques pour éviter la convergence prématurée de la méthode ; ce modèle a été appliqué pour améliorer la fiabilité du pylône électrique.

Le reste de ce papier est organisé comme suit : dans la section 2, nous présentons la méthode d'optimisation PSO. Dans la section 3, nous décrivons notre approche de parallélisation. La section 4 sera consacrée à la description de la problématique du pylône électrique. Les tests effectués et l'interprétation des résultats obtenus feront l'objet de la section 5, suivie d'une conclusion.

2. L'optimisation par essaim particulaire

Les méta-heuristiques, sont des méthodes d'optimisation permettant d'obtenir une valeur approchée de la solution optimale en un temps raisonnable. Elles ont pour objectif la résolution d'un ensemble de problèmes dans différents domaines sans avoir à modifier le principe de base de l'algorithme de la méthode. Un intérêt particulier est apporté à la méthode d'optimisation approchée PSO. Elle est basée sur les « interactions sociales » entre des « agents » appelés « particules », dans le but d'atteindre un objectif donné dans un espace de recherche commun où chaque particule a une certaine capacité de mémorisation et de traitement de l'information. Cette méta-heuristique d'optimisation stochastique a été proposée en 1995 par James Kennedy socio-psychologue et Russell Eberhart ingénieur électricien.

2.1. Définition de la méthode

Dans PSO le comportement social est modélisé par une équation mathématique permettant de guider les particules durant leur processus de déplacement [KEN 95]. Le déplacement d'une particule est influencé par trois composantes : la composante d'inertie, la composante cognitive et la composante sociale. Chacune de ces composantes reflète une partie de l'équation, figure 2.1 [COO 08] :

- 1) La composante d'inertie : la particule tend à suivre sa direction courante de déplacement ;
- 2) La composante cognitive : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;
- 3) La composante sociale : la particule tend à se diriger vers le meilleur site atteint par ses voisines.

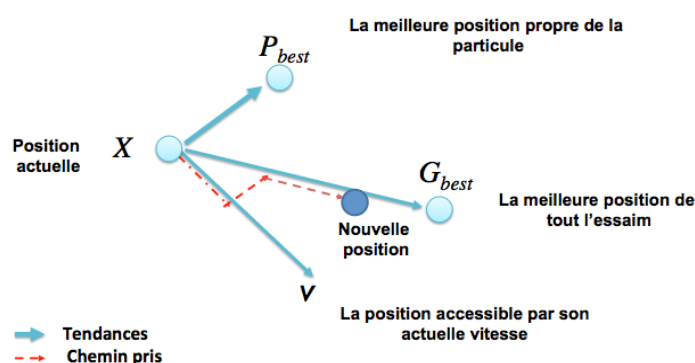


Figure 2.1. Déplacement d'une particule

2.2. Formalisation

Une particule i de l'essaim dans un espace de dimension D est caractérisée, à l'instant t , par :

- X : sa position dans l'espace de recherche ;
- V : sa vitesse ;
- P_b : la position de la meilleure solution par laquelle elle est passée ;
- P_g : la position de la meilleure solution connue de tout l'essaim ;

- $f(P_b)$: la valeur de fitness de sa meilleure solution ;
- $f(P_g)$: la valeur de fitness de la meilleure solution connue de tout l'essai.

Le déplacement de la particule i entre les itérations t et $t+1$ se fait selon les deux équations [1] et [2] [KEN 95] :

$$V(t+1) = V(t) + C1r1(P_b(t) - X(t)) + C2r2(P_g(t) - X(t)) \quad [1]$$

$$X(t+1) = X(t) + V(t+1) \quad [2]$$

- $C1$ et $C2$: deux constantes qui représentent les coefficients d'accélération, elles peuvent être non constantes dans certains cas selon le problème d'optimisation posé ;
- $r1$ et $r2$: deux nombres aléatoires tirés de l'intervalle $[0,1]$.

2.3. Algorithme

L'algorithme de base de la méthode PSO proposé par [KEN 95], commence par une initialisation aléatoire des particules dans leur espace de recherche, en leurs attribuant une position et une vitesse initiales. À chaque itération de l'algorithme les particules se déplacent selon les équations (1) et (2) et les fonctions objectif (fitness) des particules sont calculées afin de pouvoir calculer la meilleure position de toutes P_g . La mise à jour des P_b et P_g est faite à chaque itération suivant l'algorithme cité en figure 2.2 [GHE 13]. Le processus est répété jusqu'à satisfaction du critère d'arrêt.

Début

*Initialiser les paramètres et la taille S de l'essai;
Initialiser les vitesses et les positions aléatoires des particules
dans chaque dimension de l'espace de recherche;
Pour chaque particule, $P_b = X$;
Calculer $f(X)$ de chaque particule;
Calculer P_g ; // la meilleure P_b*

Tant que (la condition d'arrêt n'est pas vérifiée) **faire**

Pour (i allant de 1 à S) **faire**

Calculer la nouvelle vitesse à l'aide de l'équation (1);

Trouver la nouvelle position à l'aide de l'équation (2);

Calculer $f(X)$ de chaque particule;

Si $f(X)$ est meilleure que $f(P_b)$ **alors**

$P_b = X$;

Si $f(P_b)$ est meilleure que $f(P_g)$ **alors**

$P_g = P_b$;

Fin pour

Fin tant que

Fin

Afficher la meilleure solution trouvée P_g .

Figure 2.2. Pseudo code de l'algorithme de base de la méthode PSO

2.4. Configuration de la méthode

Il existe plusieurs paramètres qui interviennent et influencent la performance de la méthode PSO. Le choix de ces paramètres reste critique et dépend généralement du problème posé [PAR 02], [HYA 10], mais a une grande influence sur la convergence de l'algorithme. Parmi les paramètres qui rentrent en ligne de compte citons :

- La dimension du problème ;
- Le nombre des particules ;
- La disposition des particules ;
- Les coefficients de confiance ;
- La vitesse maximale ;
- Le facteur d’inertie ;
- Le facteur de construction ;
- La notion du voisinage ;
- Le critère d’arrêt.

Nous allons par la suite nous intéresser plus particulièrement aux quatre derniers paramètres.

2.4.1. Le coefficient d’inertie

Le coefficient d’inertie w a été introduit par [SHI 98] pour contrôler l’influence de la direction de la particule sur le déplacement futur. Le but de l’introduction de ce paramètre est de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). La formule [1] de calcul de la vitesse devient :

$$V(t+1) = wV(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t))$$

La valeur de w est généralement constante, mais peut être variable dans certains cas, une grande valeur de w est synonyme d’une grande amplitude de mouvement et donc d’exploration globale de l’espace de recherche. Les études menées par [SHI 98] indiquent une meilleure convergence pour w entre 0.8 et 1.2. La détermination de la meilleure valeur de ce paramètre pour chaque algorithme se fait à travers des expérimentations numériques. Dans [SHI 99], les auteurs ont proposé un coefficient d’inertie dynamique qui varie au cours du temps, et diminue linéairement au cours du processus de l’optimisation. Il commence par une valeur proche de 0.9 et descend linéairement pour arriver à 0.4.

2.4.2. Le facteur de constriction

Le facteur de constriction k a été proposé par [CLE 02], dans le but d’améliorer la convergence de l’algorithme, de prévenir l’explosion de l’essaim et de contrôler la vitesse des particules afin d’échapper au problème de la divergence de l’essaim qui cause la convergence prématurée de l’algorithme. La formule de la vitesse [1] devient alors :

$$V(t+1) = kV(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t)) \quad [3]$$

$$\text{où : } k = \frac{2}{|2 - \sigma - \sqrt{\sigma^2 - 4\sigma}|} \quad \text{et : } \begin{aligned} \sigma &= \sigma_1 + \sigma_2 \geq 4.0 \\ \sigma_1 &= C1r1 \\ \sigma_2 &= C2r2 \end{aligned}$$

Selon d’autres études menées par [RUS 00], dans certains cas, le coefficient de constriction seul ne permet pas la convergence vers la solution optimale pour un nombre d’itérations donné. Pour résoudre ce problème, il peut être intéressant de fixer $V_{\max} = X_{\max}$ en plus du coefficient de constriction, ce qui permet d’améliorer les performances globales de l’algorithme.

2.4.3. La notion du voisinage

Le voisinage constitue la structure du réseau social. Le voisinage d’une particule représente avec qui chacune des particules va pouvoir communiquer. Il existe deux principaux types de voisinages :

Le voisinage géographique : ce type de voisinage représente la proximité géographique, c'est la notion la plus naturelle du voisinage pour les essais particuliers, les voisins sont considérés comme les particules les plus proches. Cependant, à chaque itération, les nouveaux voisins doivent être recalculés à partir d'une distance prédéfinie dans l'espace de recherche. C'est donc un voisinage dynamique qu'il convient de définir et d'actualiser à chaque itération. C'est ce type de voisinage qui a été retenu dans notre approche.

Le voisinage social : ce type de voisinage représente la proximité sociale, les voisinages ne sont plus l'expression de la distance mais l'expression de l'échange d'informations, les voisins sont définis à l'initialisation et ne sont pas modifiés par la suite. Une fois le réseau des connexions sociales établi, il n'y a pas besoin de le réactualiser. C'est donc un voisinage statique.

La modification de la formule de vitesse (1) est réalisée en utilisant un nouveau terme dans l'équation. Il a été introduit par [BOC 06], son illustration paraît dans la figure 2.3 [ELH 13] :

$$V(t+1) = V(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t)) + C3r3(Pn(t) - X(t))$$

où :

P_n : la meilleure position du voisinage ;

C_3 : le coefficient d'accélération, appelé aussi paramètre social ;

r_3 : nombre aléatoire tiré de l'intervalle $[0,1]$.

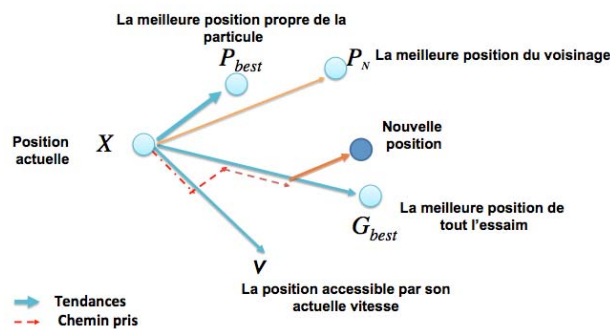


Figure 2.3. Déplacement d'une particule

Le réseau de rapports entre toutes les particules est connu sous le nom de "la topologie de l'essaim". Le choix d'une topologie du voisinage à une importance cruciale, plusieurs études de topologies ont été menées à ce propos [MEN 04], différentes combinaisons ont été proposées dont les plus utilisées sont susmentionnées ci-dessous [ELH 13] :

Topologie en anneau (figure 2.4 (a)) : chaque particule est reliée à n particules, (généralement $n = 3$), chaque particule tend à se diriger vers la meilleure dans son voisinage locale.

Topologie en rayon (figure 2.4 (b)) : la communication entre les particules est faite via une particule centrale, seule cette dernière ajuste sa position vers la meilleure, s'il y a amélioration de sa position, l'information est alors propagée à ses congénères.

Topologie en étoile (figure 2.4 (c)) : chaque particule est reliée à toutes les autres, le réseau social est complet, ie. L'optimum du voisinage est l'optimum global.

C'est le réseau de communication "anneau" qui a été utilisé comme topologie pour notre modèle.

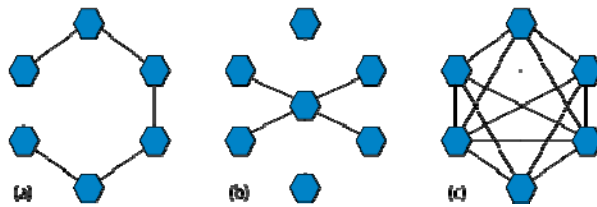


Figure 2.4. Différents types de topologies pour un essaim de particules

2.4.4. Le critère d'arrêt

Le critère d'arrêt diffère suivant le problème d'optimisation posé et les contraintes de l'utilisateur, il est fortement conseillé de doter l'algorithme d'une porte de sortie puisque la convergence vers la solution optimale globale n'est pas garantie dans tous les cas de figure même si les expériences dénotent la grande performance de la méthode. De ce fait plusieurs études ont été menées dans ce sens [ZIE 08], différentes propositions ont eu lieu : l'algorithme doit alors s'exécuter tant que l'un des critères de convergence n'a pas été atteint cela peut être : le nombre maximum d'itérations ; l'optimum global est connu a priori, on peut définir une "précision acceptable". D'autres critères d'arrêt peuvent être utilisés selon le problème d'optimisation posé et des contraintes utilisateurs. Trois critères sont proposés pour notre modèle et que nous présenterons dans la section 3.

3. Proposition d'une approche de parallélisation de la méthode PSO

Dans l'implémentation de l'algorithme classique de la méthode PSO, tous les calculs se font d'une manière séquentielle, c'est de là que vient l'idée de la parallélisation pour améliorer les performances de l'algorithme. Plusieurs scénarios s'imposent [ZHO 09] [WAI 00], celui que nous avons adopté pour notre implémentation, permet de paralléliser les calculs en lançant un ensemble de threads sur des lots de particules se positionnant dans des voisinages différents.

Les threads, sorte de processus, s'exécutent en parallèle pour chaque itération de l'algorithme. Chaque thread exécute le traitement d'une itération de son lot de particules, et attend que les autres threads finissent leurs traitements pour mettre à jour les voisinages et commencer une nouvelle itération. Ce scénario se répète jusqu'à l'obtention d'une solution satisfaisante : "atteinte du critère d'arrêt".

Nos voisinages ont la forme de sphères, qui sont mises à jour à chaque itération : leurs centres évoluent et la valeur du rayon change suivant des conditions relatives au nombre de voisinages.

La particularité de l'approche consiste à profiter de la robustesse de l'algorithme PSO dans le choix du bon paramétrage, plus particulièrement la notion de voisinage dynamique, dans le but de créer la diversité dans la recherche et dans le partage de l'information pour une convergence plus optimale. Aussi bien le calcul parallèle qui permet d'accélérer les calculs afin d'avoir une solution optimale dans un temps de calcul optimisé. La figure 3.1 est une représentation de l'approche proposée.

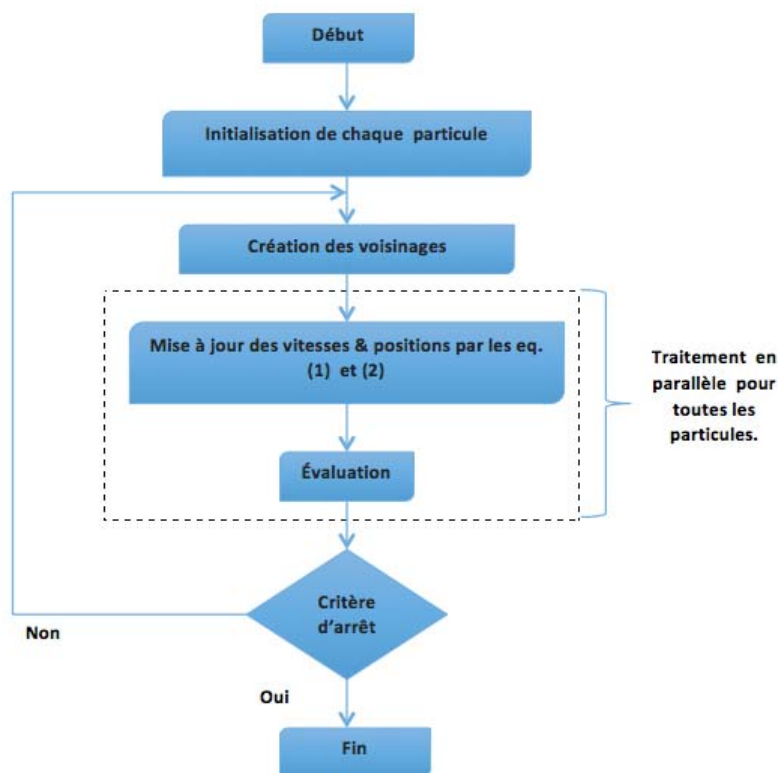


Figure 3.1. Représentation de l'approche implémentée

3.1. Paramétrage de la méthode

Chaque paramètre de l'algorithme PSO a une influence importante sur le comportement des particules et donc de la convergence de l'algorithme ; et même si la méthode PSO présente des résultats satisfaisants, le choix du bon paramétrage de la méthode reste un point critique en tant qu'une des clés de succès pour tout algorithme PSO. Dans la section précédente, nous avons présenté quelques paramètres qui influencent le comportement des particules dans leurs déplacements à la recherche de l'optimum. Le jeu de paramètres que nous avons élaboré dans notre modèle consiste en l'utilisation de plusieurs paramètres variables, que l'on peut modifier depuis l'interface utilisateur dédiée pour cela, tout dépend des exigences du problème d'optimisation posé. Pour notre problématique, une expérimentation massive a été effectuée, pour trouver le jeu de paramètres adéquat ; et qui a donné des résultats que nous estimons satisfaisants.

À noter, qu'un simple changement de la valeur d'un paramètre peut changer grandement le résultat, et peut même mener à une convergence prématurée. Le principe de création des voisinages, le traitement parallèle, le critère d'arrêt et l'algorithme vont être précisés ci-dessous.

3.1.1. Création des voisinages

Les voisinages sont de sphères dynamiques, à chaque itération le nombre de particules comprises dans les différentes sphères change selon les nouvelles positions des particules et la valeur du rayon.

La création des sphères se fait de la manière suivante : On initialise les positions des particules, on spécifie la valeur initiale du rayon, on considère une première particule P_c . Elle représente alors le centre de la sphère S de rayon r . Une particule P_a est voisine de la particule P_c si la distance euclidienne de P_a à P_c est inférieure ou égale à la valeur du rayon r . Sinon, elle devient centre d'une nouvelle sphère. Toute nouvelle particule a son appartenance étudiée par rapport aux diverses sphères créées avant de décider de la création d'une nouvelle sphère.

Par ailleurs, si le nombre de sphère est réduit (nombre prédéfini) alors le rayon commun aux sphères est diminué sensiblement. La particularité des voisinages de notre modèle est qu'on bénéficie des

avantages de la notion du voisinage dans le partage de l'information et la coopération entre les sous-essaims, sans avoir à tomber dans le piège d'une convergence prématurée. Dans le modèle de l'algorithme PSO avec voisinage, le partage de P_n "la meilleure de chaque voisinage" se fait à chaque itération, et sur la base d'une comparaison de toutes les P_n obtenues, on définit la meilleure de tout l'essaim P_g . Ceci dit, si une particule d'un voisinage se dirige vers un site promoteur (contenant une bonne solution), et qu'elle se déclare meilleure de son voisinage, à la fin de l'itération l'information sera propagée, et cette particule sera déclarée meilleure de tout l'essaim, donc elle influencera la formule de déplacement de toutes les particules, qui vont se diriger vers ce site. Nous supposons que ce site contient un optimum local, et qu'il y a bien évidemment la solution optimale quelque part dans l'espace global de recherche, mais vue l'influence de l'information propagée à chaque itération dans le déplacement des particules, ces dernières se dirigent vers la mauvais chemin, ce qui mène à une convergence prématurée.

Ce que nous proposons dans notre modèle, est que les différents voisinages cherchent la solution indépendamment de la valeur du G_b . Chaque particule se déplace en fonction de sa meilleure valeur P_b , et la meilleure de son voisinage P_n . Notre modèle respecte toujours le principe de base de l'algorithme PSO basé sur la coopération entre les particules, et le partage de l'information qui existent toujours, puisque les voisinages sont dynamiques. A chaque itération les particules changent de voisinages et donc elles diffusent leurs informations dans les nouveaux voisinages. Le non partage de la P_g (meilleure position connue globale) à chaque itération permet une meilleure exploitation de l'espace de recherche et donne plus de chance aux particules d'éviter l'anomalie de la convergence prématurée de l'algorithme.

3.1.2. *Le calcul parallèle*

Notre approche basée sur l'algorithme PSO, consiste à lancer un ensemble de traitements (threads) en parallèle. Chaque thread s'occupe du traitement d'un lot de particules pour toutes les itérations jusqu'à atteinte du critère d'arrêt. À la fin de chaque itération, une synchronisation des threads se fait afin d'évaluer les résultats obtenus pour chaque voisinage et mettre à jour les voisinages pour commencer une nouvelle itération.

3.1.3. *Le critère d'arrêt*

Le choix d'un critère optimisé n'est généralement pas simple. Pour plus de diversité dans le programme, nous avons opté pour trois critères d'arrêt :

- 1) le nombre maximum d'itérations sans amélioration ;
- 2) une précision relative au rayon ;
- 3) une précision relative à la distance du G_{best} .

En ce qui concerne le premier critère, nous spécifions un nombre d'itérations au bout duquel sans amélioration remarquable de la solution on arrête le programme. Le deuxième critère est relatif à une valeur précisant le rayon minimal accepté, si cette valeur est atteinte alors l'exécution du programme s'arrête. Le troisième critère concerne la valeur de la meilleure position de tout l'essaim, si la distance entre la valeur de la meilleure position à l'itération t et la meilleure position à l'itération $t+1$ est égale à une précision spécifiée, c'est à dire qu'il n'y a pas d'amélioration notable de la solution alors il y a arrêt des calculs. Tous ces critères sont implémentés, paramétrables depuis l'interface utilisateur, et leurs valeurs varient selon le problème à optimiser. Dans notre cas, nous avons lancé les tests en utilisant ces trois critères, en changeant à chaque fois leurs valeurs, afin de trouver les valeurs les plus optimales à notre problématique. Après avoir trouvé cette combinaison de valeurs, on lance l'exécution et une fois l'un des critères est atteint, le programme s'arrête.

3.1.4. *Les étapes de l'algorithme*

Les principales étapes de l'algorithme de notre approche sont les suivantes :

Étape 1 : Attribuer les particules pour l'ensemble de l'espace de recherche en générant leurs positions, vitesses et topologie de communication.

Étape 2 : Créer les voisinages via la valeur initiale du rayon.

Étape 3 : Diviser le traitement de l'algorithme PSO sur un ensemble de traitements : pour chaque traitement on attribue un thread.

Étape 4 : Attribuer les lots de particules aux threads

Étape 5 : Lancer les traitements de tous les threads en parallèle pour une itération.

Étape 6 : Mettre à jour les voisinages selon les nouvelles positions des particules et la nouvelle valeur du rayon s'il y a lieu.

Étape 7 : Si le critère d'arrêt est satisfait, arrêter, sinon passer à l'étape 5.

Étape 8 : Le résultat est la meilleure solution obtenue parmi les threads.

4. Problématique « Pylône électrique »

La production, le transport et la distribution de l'électricité induisent des pertes ; qui sont dues à plusieurs raisons. Pour notre étude on effectue une analyse structurale du pylône d'une ligne de transport d'électricité qu'on assimile à un treillis plan (figure 4.1). Deux charges identiques F de 1,8 KN sont appliquées aux deux extrémités supérieures du pylône suivant un angle de $\theta=15^\circ$. Les barres formant la structure sont en acier dont le module d'élasticité $E=210$ GPa et le coefficient de Poisson $\sigma=0.27$. La section de chaque barre vaut $A=27.90$ cm². On fait l'hypothèse que le poids de chacune des barres du treillis est négligeable devant les efforts appliqués.

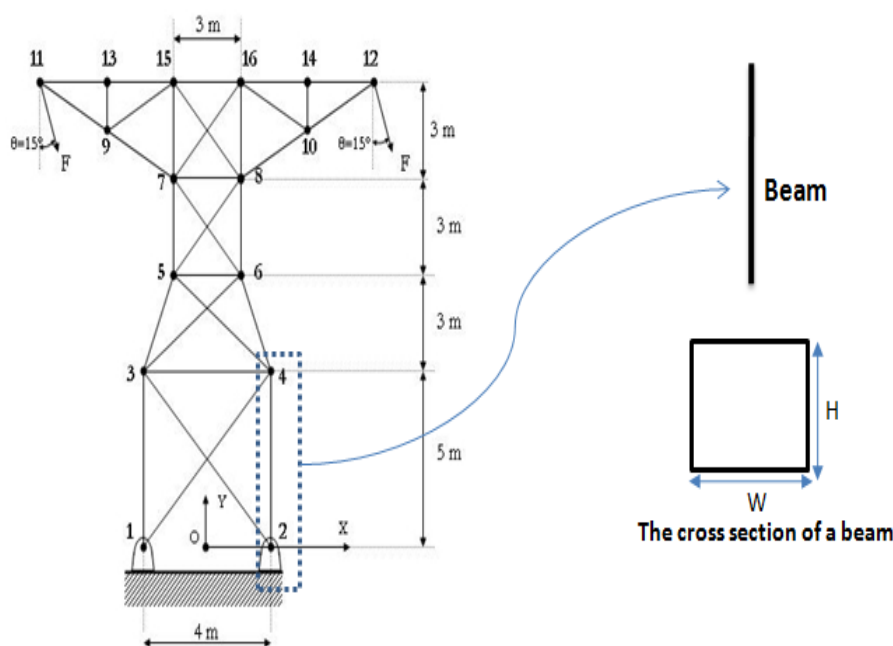


Figure 4.1. Géométrie du pylône électrique

Le but de cette analyse est de déterminer les efforts, les contraintes dans les différents éléments du treillis et le déplacement maximal engendré par les charges appliquées et de vérifier si certains éléments du treillis sont sujets au flambement (voir figure 4.2). Il s'agira également de faire l'optimisation de conception pour trouver la section droite optimale de chaque barre qui minimise le déplacement maximal dans la barre afin d'améliorer la fiabilité du pylône. Les variables d'optimisation sont $X= \{H,W\}$, où H et W sont respectivement la hauteur et la largeur de la section droite de la barre, et qui sont les composantes principales de notre fonction objectif « la fonction à optimiser ».

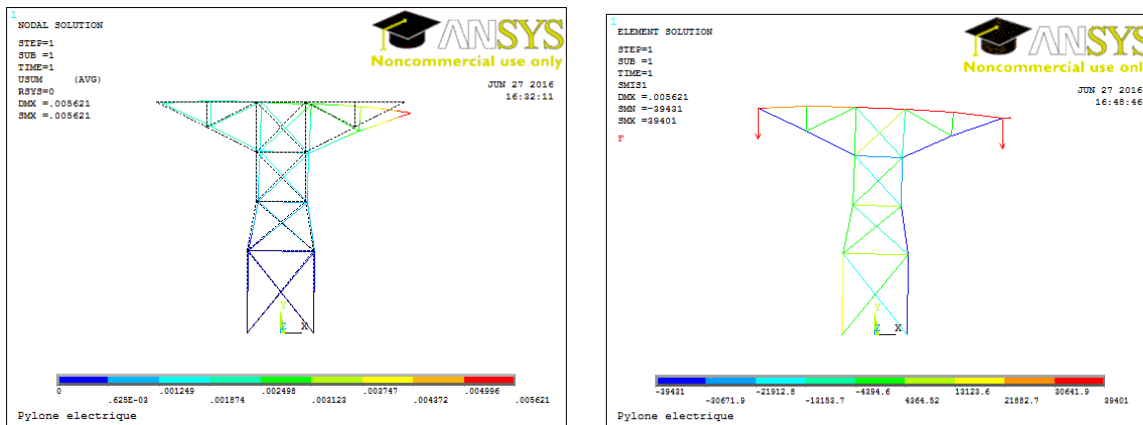


Figure 4.2. L'analyse du déplacement maximal du pylône

5. Description des expérimentations et résultats

La qualité des méthodes d'optimisation est fréquemment évaluée en utilisant des fonctions de test standard. Cette série de problèmes créés spécifiquement pour tester les performances des algorithmes d'optimisation, telles que : la vitesse de la convergence, la précision, la robustesse, la performance générale, etc. En testant notre programme sur un ensemble de fonctions tests, et à travers les résultats obtenus, nous constatons que notre modèle présente une efficacité remarquable en terme de diminution du temps de calcul de l'optimalité de convergence. Mais il ne faut pas oublier, la raison pour laquelle ces méthodes d'optimisation ont vu le jour, c'est dans cette optique que nous nous sommes intéressés à un problème réel, celui du transport de l'électricité, en optimisant la durée de vie d'un pylône électrique.

La modification de l'algorithme de base de la méthode PSO pour notre approche concerne trois points essentiels : la notion du voisinage, l'adaptation des paramètres, et le calcul parallèle. Ces modifications de l'algorithme améliorent sa performance. L'adaptation des paramètres à la problématique posée, a été faite suivant une politique d'expérimentation très aigüe, en changeant à chaque exécution la valeur d'un seul paramètre, pour voir son influence sur la convergence de l'algorithme, cette étape est primordiale pour trouver le jeu de paramètres adéquat à notre problématique.

5.1. Paramètres expérimentaux

Comme tous les algorithmes évolutionnaires, PSO a un ensemble de paramètres qui doit être défini par l'utilisateur, certains de ces paramètres sont mentionnés dans la section 2. Comme mentionné précédemment, notre programme possède une interface utilisateur où l'utilisateur peut sélectionner et modifier tous les paramètres PSO en fonction de ses besoins, y compris : la définition de l'espace de recherche, la taille de la population, la fonction objectif, la topologie de communication, le nombre de traitements (ce dernier dépend de la puissance de la machine exécutant le programme), le critère d'arrêt, etc. (voir figure 5.1).

Paramètres d'exécution	Valeurs	Paramètres de déplacement	Valeurs	Conditions d'arrêt	Valeurs
Nombre d'exécution:	1	C1:	1,25	Valeur maximale du rayon	120,0
Nombre de Threads:	3	C2:	2,25	10 ^{-x} :	
Nombre de particules:	30	C3:	0,0	Arrêt après X itérations	30
Nombre d'itérations:	60	Inclure le best local :	<input type="checkbox"/>	sans amélioration:	
Fonction:	Mecanique	Facteur d'inertie	0,2	Vmax:	100,0
Transformation :	<input checked="" type="checkbox"/>	<Important>			
Dimension:	2				
Valeur initiale du rayon:	0,7				
Chercher		Vider la liste			
Résultats :		Exporter			

Figure 5.1. Capture de l'interface graphique pour la partie des paramètres PSO

Notre algorithme a été programmé en JAVA 1.8, et les expérimentations ont été faites sur une machine MacBook Pro OS X 10.8.15, Core i7, 4 Go. Les expérimentations réalisées reposent sur le lancement de traitements PSO parallèles sur des lots de particules se positionnant dans des voisinages dynamiques à la recherche de l'optimum "minimum" de la fonction objectif (celle que nous avons déduit lors de notre étude de la problématique). D'après les résultats obtenus, nous pouvons dire que le PPSO permet d'obtenir la solution optimale avec une probabilité plus élevée, et aussi le temps de calcul dans PPSO est plus faible que celui du PSO séquentiel. Ainsi que les valeurs obtenus pour le PPSO permettent la minimisation du déplacement de la barre du pylône, ce qui maximise sa durée de vie et donc améliore sa fiabilité.

6. Conclusion et perspectives

Le présent papier contient le descriptif de l'implémentation d'une approche parallèle avec voisinage évolutif de la méthode d'optimisation approchée PSO, appliquée au problème du transport de l'électricité. La méthode PSO est une métaheuristique stochastique à base de population de solutions. Elle recherche des solutions optimales basées sur les notions de coopération et de voisinages. De nombreuses variantes et améliorations du modèle PSO classique ont été proposées en adaptant des paramètres dans l'algorithme mais une bonne accélération de l'algorithme s'avère nécessaire avec une approche de parallélisation. L'algorithme de la méthode PSO a été implémenté en séquentiel, puis en parallèle, en se basant sur l'échange entre les sous-essaims. En testant la version parallèle de la méthode PSO sur cette problématique, et à travers les résultats obtenus, nous constatons que notre modèle présente une efficacité remarquable en terme de diminution du temps de calcul de l'optimalité de convergence.

Enfin, nous envisageons dans le futur de tester le programme sur une machine plus performante, massivement parallèle, pour de meilleurs résultats, et puis de poursuivre l'étude d'autres variantes du modèle parallèle.

Bibliographie

- [AOU 11] AOUAD M.I., *Conception d'algorithmes hybrides pour l'optimisation de l'énergie mémoire dans les systèmes embarqués et de fonctions multimodales*. Thèse de Doctorat, Université Henri Poincaré- Nancy 1, France. 2011.
- [BOC 06] BOCHNEK B., FORY'S P., *Structural optimization for post buckling behavior using particle swarms*. Struct Multidisc Optim, p. 521-531. 2006.
- [CLE 02] CLERC M., KENNEDY J., *The particle swarm - explosion, stability, and convergence in a multidimensional complex space*. In : IEEE Trans. Evolutionary Computation, 6(1) : pp.58-73. an international journal, 1(2-3), pp.235-306. 2002.
- [COO 08] COOREN Y., *Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire. Applications en génie médical et en électronique*. Thèse de Doctorat, Université de Paris 12 Val de Marne, France. 2008.
- [ELH 13] ELHAMI N., *Contribution aux méthodes hybrides d'optimisation heuristiques : Distribution et application à l'interopérabilité des systèmes d'information*. Thèse de Doctorat, Université Mohammed V Rabat, Maroc & Université de Rouen, France, 2013.
- [GHE 13] GHERBOUDJ A., *Méthodes de résolution de problèmes difficiles académiques*. Thèse de Doctorat, Université de Constantine 2, Algérie, 2013.
- [HYA 10] HYASS M.E., HYASS P., *Good Parameters for Particle Swarm Optimization*. In : Laboratories Technical Report no. HL1001. 2010.
- [KEN 95] KENNEDY J., EBERHART R., "Particle Swarm Optimization," Proceedings of the IEEE International Joint Conference on Neural Networks, IEEE Press, vol. 8, no. 3, pp. 1943-1948. 1995.
- [LIU 07] LIU H., ABRAHAM A., ZHANG W., *A Fuzzy Adaptive Turbulent Particle Swarm Optimization*. International Journal of Innovative Computing and Applications, vol. 1, n° 1, p. 39-47. 2007.
- [MEN 04] MENDES R., *Population Topologies and Their Influence in Particle Swarm Performance*. Thèse de doctorat. Université de Minho, Portugal, 2004.

- [PAQ 03] PAQUET U., ENGELBRECHT A.P., *A New Particle Swarm Optimizer for Linearly Constrained Optimization*. In : Proceedings of the IEEE Congress on Evolutionary Computation, vol. (1), pp. 227–233. 2003.
- [PAR 01] PARSOPOULOS K.E., PLAGIANAKOS V.P., MAGOULUS G.D., VRAHATIS M.N., *Objective Function “Stretching” to Alleviate Convergence to Local Minima*. Nonlinear Analysis, Theory, Methods and Applications, vol. 47, n° 5, p. 3419–3424. 2001.
- [PAR 02] PARSOPOULOS K.E., VRAHATIS M.N., *Recent approaches to global optimization problems through particle swarm optimization*. In : Natural Computing : an international journal, vol. 1, n° 2-3, pp.235-306. 2002.
- [RUS 00] RUSS C., EBERHART SHI R., *Comparing inertia weights and constriction factors in particle swarm optimization*. In : Proceedings of the Congress on Evolutionary Computing, pp. 84-89. 2000.
- [SHI 98] SHI Y., EBERHART R.C., *Parameter selection in particle swarm optimization*. In : Proceedings of the Seventh Annual Conference on Evolutionary Programming, pp. 591-600. 1998.
- [SHI 99] SHI Y., EBERHART R.C., *Empirical study of particle swarm optimization*. In : Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp.1945-1950. 1999.
- [TIN 03] TING T.-O., RAO M.V.C., LOO C.K., NGU S.-S., *A New Class of Operators to Accelerate Particle Swarm Optimization*. In : Proceedings of the IEEE Congress on Evolutionary Computation, vol. (4), pp. 2406–2410. 2003.
- [WAI 09] WAINTRAUB M., SCHIRRU R., PEREIRA C., *Parallel Particle Swarm Optimization Algorithm in Nuclear Problems*. In : International Nuclear Atlantic Conference – INAC. 2009.
- [ZHO 09] ZHOU Y., TAN Y., *GPU-Based Parallel Particle Swarm Optimization*. In : Proceedings of the IEEE Congress on Evolutionary Computation, vol. (9), pp.1493–1500. 2009.
- [ZIE 08] ZIELINSKI K., LAUR R., *Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimization*. In : Advanced in Differential Evolution, the series Studies in Computational Intelligence, vol. 143, pp. 111-138 Springer, Berlin Heidelberg. 2008.