

Optimisation des Structures Mécaniques à Variables Mixtes Utilisant l'Algorithme CoDE

A. Hammoudi ^(*), H. Selmani ^{(*)2}, F. Djddou ^{(*)3}, B. Keskes ^{(*)4}

**Laboratoire de mécanique de précision appliquée
Institut d'Optique et Mécanique de Précision
Université Ferhat Abbas Sétif 1, Algérie.
BP19000 Sétif, Algérie.*

¹ hammoudiabderazek@gmail.com

² housseem_2008_real@hotmail.fr

³ djeddouferhat@yahoo.fr

⁴ b_keskes@yahoo.fr

Abstract - The performance of the differential evolution algorithm DE in solving complex optimization problems with or without constraints is directly related to the choice of trial vector generation strategies and the value of its control parameters such: crossover control parameter CR, scaling-factor F, population size NP, etc. In the present work, we applied a recent optimization method, called composite DE (CoDE). This method uses three trial vector generation strategies and three control parameter settings. It randomly combines them to generate trial vectors. To investigate the robustness of CoDE in solving mechanical problems with mixed type variables continuous / discrete highly constrained, we have chosen to optimize the dimensions of a helical spring subjected to a constant compressive load axially guided. In this case the objective function is to minimize the volume of the spring. A program using CoDE has been developed and implemented in MATLAB. The results obtained in this paper were compared with those existing in the literature, they demonstrate the effectiveness, robustness of the applied method.

Index Terms - Composite differential evolution, helical spring, mechanical problems, mixed variables, trial vector generation strategy.

I. INTRODUCTION

L'évolution différentielle DE est un algorithme d'optimisation qui a connu énormément de succès depuis son apparition et à l'origine il est conçue pour les problèmes d'optimisation continus et sans contraintes, ses extensions actuelles peuvent permettre de traiter les problèmes à variables mixtes et gèrent les contraintes non linéaires. Cependant, la performance de DE dépend de deux facteurs essentiels. L'un est la stratégie de la génération des vecteurs d'essai, schéma de mutation et l'opérateur de croisement, et l'autre facteur est la valeur des paramètres de contrôle de l'algorithme (la taille de la population NP, le facteur de mutation F et la probabilité de croisement CR) [1].

En effet, pour améliorer les performances de DE plusieurs variantes ont été développées dans ces dernières décennies. Certains travaux sont focalisés principalement sur les stratégies de la génération de vecteur d'essai. Fan et Lampinen [2] ont proposés un opérateur de mutation trigonométrique qui permet d'accélérer la convergence de DE. Des nouveaux schémas de mutation « current-to-best/1 » et « current-

to-pbest/1 » ont été développés respectivement par Das & al. [3] et Zhang and Sanderson [4]. D'autres travaux portent sur l'optimisation des paramètres de contrôle de DE (NP, F, CR). Storn et Price [5] ont suggéré que le NP doit être compris entre 5D et 10D, F doit avoir la valeur de 0,5 et CR peut être fixé entre 0,1 et 0,9. Ronkkonen et al. [6] ont suggéré que NP doit être compris entre 2D et 4D, F doit être choisi dans l'intervalle [0.4, 0.95] et CR devrait être compris entre 0,2 et 1,0. Das et al. [7] ont introduit deux schémas pour adapter le facteur de mutation F dans DE. Un schéma varie F d'une manière aléatoire, et l'autre réduit F d'une manière linéaire de la valeur maximale prédéfinie à une valeur minimale. Yong Wang & al [1], ont développé une méthode, appelé composite DE (CoDE), elle combine trois schéma de mutation (rand/1/bin, rand/2/bin et current-to-rand/1) avec trois valeurs de F et CR (F = [1 1 0.8] et CR = [0.1 0.9 0.2]) d'une manière aléatoire. La performance de CoDE pour résoudre les problèmes d'optimisation a été validée seulement par les 25 fonctions Benchmark de test CEC 2005.

Le présent travail porte sur l'optimisation d'une structure mécanique à variables mixtes discrètes/continues «dimensions d'un ressort hélicoïdal» en utilisant le CoDE. Les trois schémas de mutation utilisés pour générer les vecteurs d'essai sont (rand/1/bin, Best/1/bin et rand/2/bin) et les valeurs des paramètres sont ($F = [0.1 \ 0.2 \ 0.1]$ et $CR = [0.9 \ 0.8 \ 0.1]$). Le ressort est soumis à une charge de compression constante guidée axialement. La fonction objectif à minimiser dans ce cas est le volume du ressort. Le diamètre du fil d , le diamètre d'enroulement moyen D et le nombre de spires actives N ont été choisis comme des variables de décisions à optimisées. Ce problème d'optimisation est soumis à huit contraintes de conception de type d'inégalités non linéaires.

Le reste de l'article sera organisé comme suit. La deuxième section décrit le principe et le déroulement de CoDE. Dans la troisième section, le modèle mathématique du problème d'optimisation du ressort hélicoïdal a été formulé. La quatrième section est réservée pour les résultats numériques. Nous terminerons le présent travail par une conclusion sur la méthode utilisée.

II. COMPOSITE DE (CoDE)

Le DE est un algorithme d'optimisation itératif qui suit le schéma classique des algorithmes évolutionnaires. Il peut être écrit en quatre phases: l'initialisation, la mutation, le croisement et la sélection. L'idée principale de la variante CoDE, utilisée dans cet article, est de combiner aléatoirement à la fois trois schémas de mutation de DE les plus connues (rand/1/bin, Best/1/bin et rand/2/bin) avec trois valeurs de facteur de mutation $F = [0.1 \ 0.2 \ 0.3]$ et de probabilité de croisement $CR = [0.9 \ 0.8 \ 0.7]$ les plus recommandées dans des travaux antérieurs, afin de générer les vecteurs d'essai des individus de la population pour chaque génération. Cette stratégie permet de garantir une diversification des solutions compétant pour atteindre les meilleurs valeurs de la fonction objectif. Le meilleur individu est sélectionné pour la population de la prochaine itération si la valeur de la fonction correspond est la plus petite dans les trois schémas. La figure 1 explique le principe de fonctionnement de CoDE.

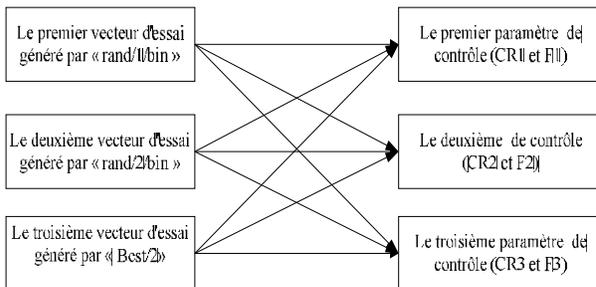


Fig. 1 Le schéma explicatif de CoDE

II.1 INITIALISATION

Comme dans le cadre de ce travail, le problème traité est à variables mixtes (continue et discret). À la génération $G = 0$, les vecteurs initiaux sont créés d'une manière aléatoire à partir de l'espace de solution possible pour chaque variable, de la manière suivante:

A. Pour les variables continues

$$x_{i,j}^{G=0} = x_i^{(l)} + rand_{i,j} * (x_i^{(s)} - x_i^{(l)})$$

(1)

Où $j = 1, 2, \dots, NP$, $i = 1, 2, \dots, D$. D est le nombre de variables, NP est la taille de la population, $x_i^{(l)}$ et $x_i^{(s)}$ sont respectivement les limites inférieures et supérieures de $i^{\text{èmes}}$ variable du problème. L'opérateur $rand_{i,j}[0,1]$ permet de créer une valeur aléatoire uniformément distribuée dans l'intervalle $[0,1]$.

B. Pour les variables discrètes

Pour traiter ce type de variable (discrète) l'idée est de considérer comme variable, l'indice de la variable ou bien son rang dans le tableau ou le vecteur des valeurs normalisées. Si l'indice crée est par exemple 2 alors c'est la deuxième valeur normalisée qui sera prise. Les indices sont créés aléatoirement comme suit:

$$x_{i,j}^{G=0} = INT \left(x_i^{(l)} + rand_{i,j} * (x_i^{(s)} - x_i^{(l)}) \right)$$

(2)

Où INT est une fonction qui permet de transformer la valeur réelle de $x_{i,j}^{(G=0)}$ en valeur entière la plus proche et donne ainsi l'indice de la variable dans l'intervalle connue.

II.2 MUTATION ET CROISEMENT

Pour chaque vecteur courant $x_{i,j}^G$, nous générons trois vecteurs qui peut être créé en utilisant trois stratégies de mutation les plus connues:

- Rand/1/bin

$$u_{i,j}^{G+1} = \begin{cases} v_{i,j}^{G+1} = x_{i,r1}^G + F_1 (x_{i,r2}^G - x_{i,r3}^G) \\ \text{si } (rand_i(0,1) \leq CR_1) \text{ or } i = i_{rand} \\ x_{i,j}^G \quad \text{autrement} \end{cases} \quad (3)$$

- Best/1/bin

$$u_{i,j}^{G+1} = \begin{cases} v_{i,j}^{G+1} = x_{i,best}^G + F_2 (x_{i,r1}^G - x_{i,r2}^G) \\ \text{si } (rand_i(0,1) \leq CR_2) \text{ or } i = i_{rand} \\ x_{i,j}^G \quad \text{autrement} \end{cases}$$

(4)

- Rand/2/bin

$$u_{i,j}^{G+1} = \begin{cases} v_j^{G+1} = x_{i,r_1}^G + \text{rand}(x_{i,r_2}^G - x_{i,r_3}^G) + F_3(x_{i,r_4}^G - x_{i,r_5}^G) \\ \text{si } (\text{rand}_i(0,1) \leq CR_3) \text{ \& } i \neq \text{rand} \\ x_{i,j}^G \quad \text{autrement} \end{cases}$$

(5)

Les indices r_1, r_2, r_3, r_4, r_5 aléatoirement choisis dans l'intervalle $[1, 2, \dots, NP]$, tous différents et différents de l'indice courant j . $x_{i,best}^G$ est le meilleur individu à la $G^{\text{ème}}$ génération. Les facteurs de mutation F_1, F_2, F_3 et les probabilités de croisement CR_1, CR_2 et CR_3 sont aléatoirement choisis respectivement dans les ranges $F = [0.1 \ 0.2 \ 0.1]$ et $CR = [0.9 \ 0.8 \ 0.1]$. i_{rand} est un nombre aléatoire uniforme appartenant à l'intervalle $[1, D]$.

II.3 SELECTION

La sélection consiste à comparer seulement la valeur de la fonction objective des deux vecteurs x_j^G et u_j^{G+1} , le meilleur individu est sélectionné pour la population de la prochaine itération si la valeur de la fonction correspond est la plus petite dans les trois schémas. La sélection peut être formulée mathématiquement comme suit:

$$x_j^{G+1} = \begin{cases} u_j^{G+1} & \text{if } f(u_j^{G+1}) \leq f(x_j^G) \\ x_j^G & \text{autrement} \end{cases} \quad (6)$$

II.3 TRAITEMENT DES CONTRAINTES

A. Contraintes limite

Toute valeur du vecteur mutant trouvée infaisable c'est-à-dire $v_{i,j}^{G+1} \notin [x_i^{(l)}, x_i^{(s)}]$, est remplacée par une valeur générée comme suit :

$$v_{i,j}^{G+1} = \begin{cases} \min\{I_j, 2S_j - u_{i,j}^{G+1}\} & \text{if } v_{i,j}^{G+1} < I_j \\ \max\{I_j, 2S_j - u_{i,j}^{G+1}\} & \text{if } v_{i,j}^{G+1} > S_j \end{cases}$$

(7)

B. Contraintes fonctions

La méthode de pénalité statique a été utilisée pour résoudre ce problème. Elle permet de transformer la résolution du problème avec contraintes en un problème sans contrainte de la façon suivante:

$$F(x) = f(x) + \lambda_k \cdot \sum_{k=1}^8 \langle g_k(x)^2 \rangle \quad (8)$$

Nous avons utilisé un coefficient de pénalité différent (λ_k) pour chaque contrainte, $F(x)$ est la fonction objective pénalisée, g_k les contraintes d'inégalité.

III. ETUDE EXPERIMENTALE

Afin d'étudier la performance de CoDE à résoudre les problèmes mécaniques à variables mixtes fortement contraint, nous avons choisi d'optimiser la conception d'un ressort hélicoïdal soumis à une charge de compression constante guidée axialement Fig1 [8]. La fonction objectif à minimiser dans ce cas est le volume

du ressort. Le diamètre du fil d , le diamètre d'enroulement moyen D et le nombre de spires actives N ont été choisis comme des variables de décisions à optimisées, D est un continu choisit entre $0.25 \leq D \leq 1.3$ (inch), n est un entier choisit entre $2 \leq N \leq 15$ et d peut prendre l'une des 42 valeurs discrètes énumérées dans le Tableau 1. Ce problème est soumis à huit contraintes d'inégalités non linéaires.

- Les variables d'optimisation choisies sont:

$$X = (x_1, x_2, x_3)^T = (d, D, N)^T$$

- La fonction du coût à minimiser est le volume de ressort, elle s'exprime sous la forme suivante:

$$f(X) = \frac{\pi D d^2 (N + 2)}{4}$$

(9)

Sous les fonctions contraintes suivantes:

- La contrainte au cisaillement provoquée par la force de compression doit être inférieure à la contrainte maximale admissible de cisaillement S du matériau:

$$g_1(X) = \frac{8C_f F_{\max} D}{3.14156d^3} - S \leq 0$$

(10)

- La longueur libre du ressort l_f doit être inférieure à la valeur maximale l_{\max}

$$g_2(X) = l_f - l_{\max} \leq 0 \quad (11)$$

- Le diamètre du fil ne doit pas être inférieur au diamètre minimal indiqué d_{\min}

$$g_3(X) = d_{\min} - d \leq 0$$

(12)

- Le diamètre extérieur d'enroulement doit être inférieur au diamètre maximal D_{\max}

$$g_4(X) = d + D - D_{\max} \leq 0 \quad (13)$$

- Le diamètre de l'enroulement interne doit être au moins trois fois moins que le diamètre du fil pour éviter un ressort légèrement enroulé :

$$g_5(X) = 3 - \frac{D}{d} \leq 0$$

(14)

- La flèche sous la charge δ donnée, doit être inférieure à la flèche maximale spécifiée sous précontrainte δ_{\max}

$$g_6(X) = \delta - \delta_{\max} \leq 0$$

(15)

- La flèche totale doit être compatible avec la longueur de la bobine libre l_f

$$g_7(X) = \delta_p + \frac{F_{\max} - F_p}{K} + 1.05(N + 2)d - l_{free} \leq 0 \quad (16)$$

- La flèche de précontrainte à la charge maximale, doit être supérieure à la flèche de fonctionnement indiquée

$$g_8(X) = \delta_w - \frac{F_{\max} - F_p}{K} \leq 0 \quad (17)$$

- Les paramètres utilisés ci-dessus sont comme suit:

$$C_f = \frac{4(S_i) - 1}{4(S_i) - 4} - \frac{0.165}{S_i} \quad (18) \quad S_i = \frac{D}{d} \quad (19)$$

$$K = \frac{Gd^4}{8ND^3} \quad (20) \quad \delta_p = \frac{F_p}{K} \quad (21)$$

$$l_f = \frac{F_{\max}}{K} + 1.05(N + 2)d \quad (22)$$

- Les données de problème sont comme suit

Variable	Description	Valeur
S	Contrainte maximale admissible de cisaillement	$189 \times 10^3 \text{ psi}$
F_{\max}	Force maximale	1000 lb
d_{\min}	Diamètre du fil minimal	0.2 inch
D_{\max}	Diamètre extérieur d'enroulement maximal	3 inch
F_p	Force précontrainte de compression	300 lb
δ_{\max}	Flèche maximale spécifiée sous précontrainte	6 inch
δ_w	Flèche de fonctionnement indiquée	1.25 inch
E	Module d'élasticité de matériau	30×10^6
G	Module de cisaillement de matériau	11.5×10^6

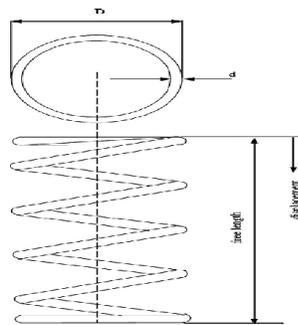


Fig. 2 Schéma du problème de conception.

Tableau 1
VALEURS DISCRETES DES DIAMETRES DE FIL
A RESSORT

Diamètre du fil d (inch)						
0.009	0.0095	0.0104	0.0118	0.0128	0.0132	0.014
0.015	0.0162	0.0173	0.0180	0.0200	0.0230	0.025
0.028	0.0320	0.0350	0.0410	0.0470	0.0540	0.063
0.072	0.0800	0.0920	0.1050	0.1200	0.1350	0.148
0.162	0.1770	0.1920	0.2070	0.2250	0.2440	0.263
0.283	0.3070	0.3310	0.3620	0.3940	0.4375	0.500

IV. RESULTATS NUMERIQUES

Les paramètres de contrôle de CoDE considérés pour le présent cas d'étude sont les suivants: la taille de population est NP = 100, le nombre maximum de générations Gmax= 200, le facteur de mutation F et la probabilité CR, ils peuvent prendre aléatoirement l'une des 3 valeurs suivantes F = [0.1 0.1 0.3] et CR = [0.9 0.8 0.7].

Afin de confirmer les résultats obtenus par le CoDE et de juger la stabilité du programme nous avons choisi d'effectuer 50 exécutions. Les résultats statistiques des exécutions sont présentés dans le tableau 2. L'évaluation de la fonction objectif et les fonctions contraintes au cours de deux cents générations sont représentées respectivement sur les figures 3, 4, 5, 6 et 7.

Pour les paramètres de contrôle considérés, la valeur minimale de la fonction objective est de $f(x) = 2.1344 \text{ inch}^3$, pour un diamètre du fil $d = 0.2630 \text{ inch}$, un diamètre d'enroulement moyen $D = 1.1369 \text{ inch}$ et un nombre de spires actives $N = 9$. Ces résultats ont été obtenus avec une satisfaction totale des contraintes du problème (voir tableau 3) et un temps d'exécution faible.

Tableau 2
LES RESULTATS STATISTIQUES DES
EXECUTIONS

Meilleur résultat	Moyen	Pire	Ecart-type
2.1344	1.8252	2.2373	0.4834

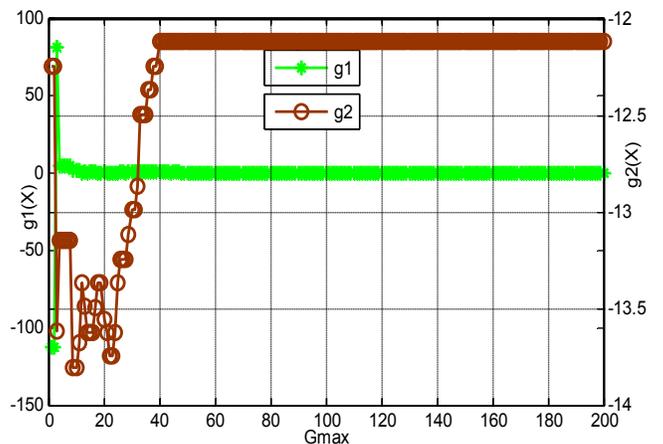


FIG.3 Evaluation des fonctions contraintes g_1 et g_2 .

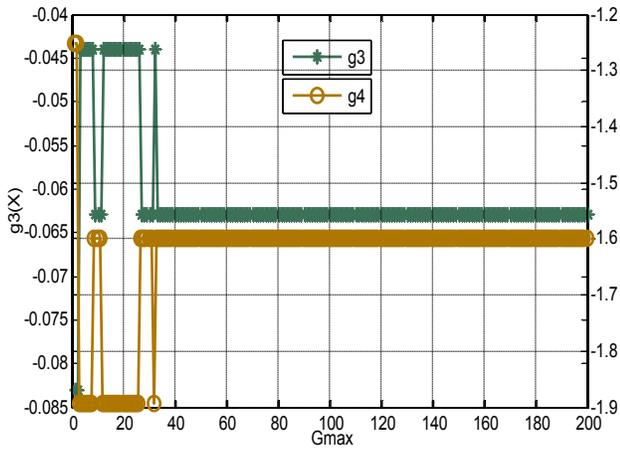


FIG.4 Evaluation des fonctions contraintes g_3 et g_4 .

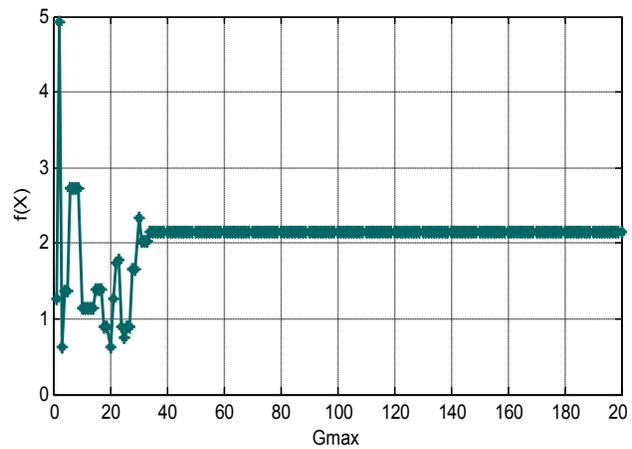


FIG.7 Evaluation de la fonction objectif.

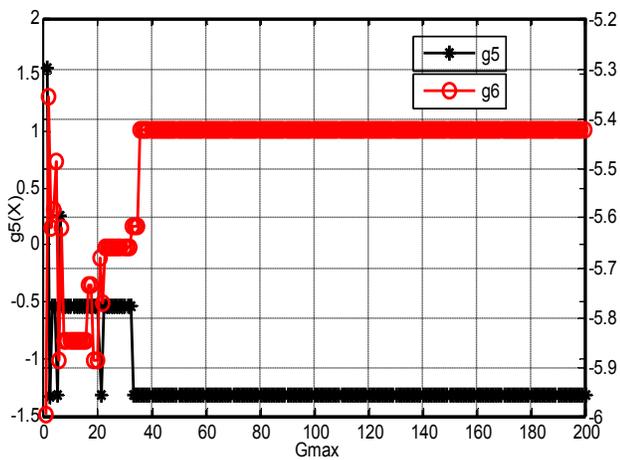


FIG.5 Evaluation des fonctions contraintes g_5 et g_6 .

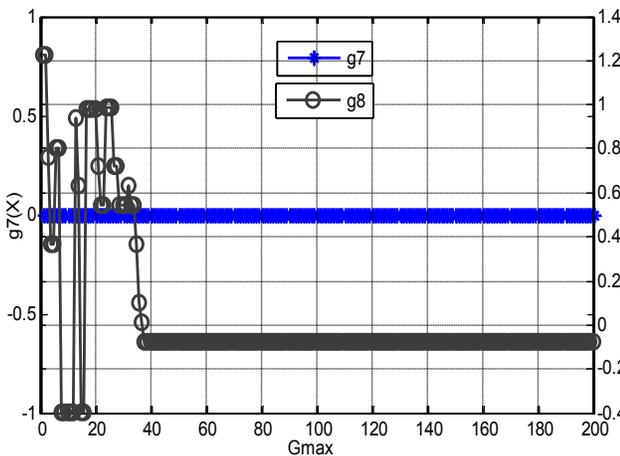


FIG.6 Evaluation des fonctions contraintes g_7 et g_8 .

IV.1 ETUDE COMPARATIVE

Le problème considéré a été traité dans la littérature par plusieurs auteurs, utilisant différentes techniques d'optimisation tels que, FA: Firefly Algorithm [9], PSO: Particle Swarm Optimization [10], GA: Genetic Algorithms [11], [8]. Le tableau 3 présente alors les résultats d'optimisation que nous avons obtenus par le CoDE et ceux des auteurs sus-cités.

Comme le montre le tableau 3, les résultats de l'optimisation indiquent que le CoDE est plus efficace que les autres algorithmes méta-heuristiques tels que FA, PSO et GA. En effet, le plus petit volume de ressort est obtenu par l'algorithme que nous avons proposé, tout en satisfaisant les contraintes.

Tableau 3
COMPARAISON DES RESULTATS DEA AVEC LA LITTERATURE

Variables	Présente article	[10]	[8]	[11]	[9]
d	0.2630	0.2830	0.283	0.263	0.2830
D	1.1369	1.223	1.2274	1.1096	1.223049
N	9	9	9	9	9
g_1	-7.5808	-1008.8	-550.9	25184.82	-1008.02
g_2	-9.0394	-8.964	-8.926	-9.1745	-8.9460
g_3	-0.0630	-0.083	-0.083	-0.0630	-0.0830
g_4	-1.600	-1.7769	-1.772	-1.8900	-1.7770
g_5	-1.3230	-1.3217	-1.337	-1.2190	-1.3220
g_6	-5.4230	-5.4643	-5.448	-5.464	-5.4640
g_7	0	0	0	0	0
g_8	-0.0963	0.0001	-0.013	-0.0014	0.00000
f_{\min}	2.1344	2.659	2.665	2.0831	2.6586

V. CONCLUSION

Dans cet article, un problème d'optimisation mécanique à variables mixtes « dimensionnement d'un ressort hélicoïdal à traction » en utilisant CoDE a été résolu. Un modèle mathématique a été établi avec trois variables de conception et huit contraintes d'inégalité, la fonction objectif à minimiser est le volume du ressort. Les résultats de l'optimisation indiquent que le CoDE est plus efficace que des autres algorithmes.

REFERENCES

- [1] Y. Wang, Z. Cai, and Q. Zhang, "Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters," *IEEE Trans. Evol. Comput.*, VOL. 15, n0. 1, pp.55-66, Feb. 2011.
- [2] H. Y. Fan and J. Lampinen, "A trigonometric mutation operator to differential evolution," *J. Global Optim.*, vol. 27, no. 1, pp. 105-129, 2003.
- [3] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526-553, Jun. 2009.
- [4] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945-958, Oct. 2009.
- [5] R. Storn and K. V. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Opt.*, vol. 11, no. 4, pp. 341-359, Dec. 1997.
- [6] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real parameter optimization with differential evolution," in *Proc. IEEE CEC*, vol. 1. 2005, pp. 506-513.
- [7] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Proc. GECCO*, Jun. 2005, pp. 991-998.
- [8] S. Wu and P. Chow P, "Genetic algorithms for nonlinear mixed discrete-integer optimization problems via metagenetic parameter optimizations," *Eng. Optimiz.* vol. 24, pp. 137-159, 1995.
- [9] A. H. Gandomi, X. Yang and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm", *Computers and Structures.*, Vol. 89, pp. 2325-2336, 2011.
- [10] C. Guo, H. JS, B. Ye and Y. Cao, "Swarm intelligence for mixed-variable design optimization," *J. Zhejiang Univ SCL*, vol. 5, no. 7, pp. 851-60, 2004.
- [11] K. Deb and M. Goyal, "Optimizing engineering designs using a combined genetic search," In: *Back IT*, editors. 7th international conference on genetic algorithms, pp. 512-528, 1997.