

Optimisation des conteneurs pour les sphères de Noël, par evolution différentielle

Noemí González-León, Leticia Flores-Pulido, Edgar-Alfredo Portilla-Flores

Université Autonome de Tlaxcala, Faculté des Sciences Fondamentales, de l'Ingénierie et de la Technologie. Mexique et Institut Technologique Supérieur de la Sierra Norte de Puebla, Division des Technologies et des Systèmes D'information. Mexique

Université Autonome de Tlaxcala, Faculté des Sciences Fondamentales, de l'Ingénierie et de la Technologie. Mexique

Institut Polytechnique National, CIDETEC, Mexico

Corresponding Author : Noemí González-León

ABSTRAIT

Dans ce document, une proposition d'un algorithme appelé NOEA pour l'optimisation de la production de conteneurs de sphères de Noël est présentée; dans le but de modéliser le processus d'une machine à blister et à matricer de manière à ce que sa production corresponde à la demande, occupant les ressources nécessaires à la réalisation de tâches et permettant d'obtenir un bénéfice en termes d'économie d'énergie; est basé sur les règles de faisabilité de Deb, ne nécessite pas de ressources informatiques excessives et son temps de réponse est raisonnable. L'importance de l'algorithme proposé réside dans l'amélioration de la conception des couvercles de conteneurs sphères de Noël.

Mots clés-Optimisation conteneur, blisteradora et évolution différentielle

Date of Submission: 25-10-2019

Date Of Acceptance: 10-11-2019

I. INTRODUCTION

Au cours des 40 dernières années, la municipalité de Chignahuapan, Puebla, a été caractérisée par la production de sphères de verre soufflées à la main; Il existe plus de 350 ateliers et 6 usines dédiés au développement de sphères de Noël. Selon Luis Ángel Rojas Lecona [1], une seule usine produit au moins dix millions de sphères par an. En 2012, ils ont produit plus de 300 000 000 de sphères. Ce produit est exporté vers les États-Unis, l'Amérique centrale et l'Amérique du Sud, comme l'Argentine et l'Italie en Europe.

Le processus de production des sphères se termine par l'emballage. Un des grands défis de la petite industrie de la région est le développement du conteneur en verre soufflé pour la distribution, le transport, le stockage, la manipulation et la vente du produit, de manière à ce qu'il puisse être livré au client dans des conditions optimales. L'emballage devient alors un avantage concurrentiel par rapport aux produits concurrents de même qualité.

Il existe deux types d'emballage. Les matériaux utilisés lors du conditionnement des sphères de Noël en verre soufflé sont du carton: l'inconvénient d'utiliser ce type de matériau est de ne pas supporter beaucoup de poids, car il est courant que le produit soit arrimé par les petits producteurs, entraînant des pertes. Trois sphères brisées par boîte. Un autre emballage largement utilisé par les

producteurs de sphères est la combinaison de polychlorure de vinyle appelé PVC et de carton.

Dans la Sierra Norte de Puebla, le climat tempéré, froid et humide prédomine une grande partie de l'année, ce qui nuit à l'utilisation des cartons fabriqués dans la région, car dans ces conditions environnementales, le conteneur devient mou, ce qui provoque la perte du produit. Le processus des récipients cylindriques en PVC pour les sphères est actuellement effectué manuellement, pour les bouchons du récipient, le matériau en PVC est thermoformé; pour l'assemblage du corps du cylindre, il est nécessaire de couper un carré avec une guillotine, un cutter ou des ciseaux, puis de la colle est utilisée pour joindre les côtés du matériau, un cordon est placé sur les couvercles perforés du conteneur afin de maintenir le produit pour le transporter.

L'emballage le plus vendu est celui de 60 sphères, le calibre en PVC est de 3,5 pouces et environ 23 000 emballages ou plus de matériaux sont vendus par atelier ou petit producteur par an. Une personne produit manuellement 50 conteneurs cylindriques en PVC par jour. La présente proposition est basée sur l'amélioration de la qualité du processus de fabrication d'emballages cylindriques permettant de stocker 60 sphères, grâce au fonctionnement d'une blister et d'un mélangeur, étant nécessaire pour générer une

solution optimale garantissant la qualité de l'emballage.

On sait jusqu'ici quelle forme un emballage doit avoir pour stocker, transporter et conserver les produits sphériques de Noël en bon état. Une procédure a déjà été établie, qui permet de construire un emballage approprié pour la production en ligne dudit dessin. Il est donc possible, au moyen d'une conception électronique intelligente et de l'évolution différentielle, de définir la construction et la production en ligne d'un emballage approprié pour le matériau sphérique de Noël en verre soufflé. La production de tels emballages doit respecter les normes et les mesures appropriées pour que la technique intelligente soit correctement conçue.

Les emballages peuvent être classés en trois catégories: souple, semi-rigide, scellants ou adhésifs. Les semi-rigides sont plus épais que 0,127 mm, les flexibles dont l'épaisseur est inférieure à 0,127 mm et les mastics ou adhésifs nécessitent plusieurs couches ensemble, nécessitant généralement chaleur et pression. Actuellement sur le marché américain, les contenants dominent 51 pourcent du marché total.

II. BILAN DE L'ÉTAT DE L'ART

Une revue de littérature a été faite, les boursiers consultés, qui en 2009 [2], Il a utilisé la méthode de thermoformage pour le scellement et le remplissage de récipients rigides, permettant une économie de 10 à 15 pourcent du produit. Dans 2017 [3], Les boursiers ont réussi à réduire leurs coûts en raison de préoccupations liées à l'énergie et à la consommation de matériaux. L'un des inconvénients constatés est le sceau, car il s'agit de la partie la plus faible d'un contenant; et al, en 2015 [4], par le comportement élastique du blister et de améliorer les performances du matériau. Nous avons examiné la modélisation de l'emballage de Li, en 2017 [5] pour les caractéristiques des récipients qui sont perméables à l'humidité. La proposition a été analysée pour résoudre un problème d'optimisation avec des restrictions associées à un mécanisme de Vega-Alvarado, et al, 2014 [6], ils ont utilisé un algorithme d'évolution différentielle modifié, en appliquant une sélection de type de tournoi basée sur les règles de faisabilité de Deb nécessite des ressources informatiques étendues et un temps de réponse favorable.

III. PROBLÈME D'OPTIMISATION MONO-OBJECTIF

Laissez le problème d'optimisation numérique mono-objectif décrit par 1 en haut 7, pour obtenir la solution au problème de conception afin de minimiser les ressources nécessaires la production de récipients de 1); ceci passe par une définition adéquate de la représentation d'un individu, de sorte que des

individus valides soient toujours générés. La représentation de chacun des termes de la fonction aptitude est visible dans les figures 2 et 3.

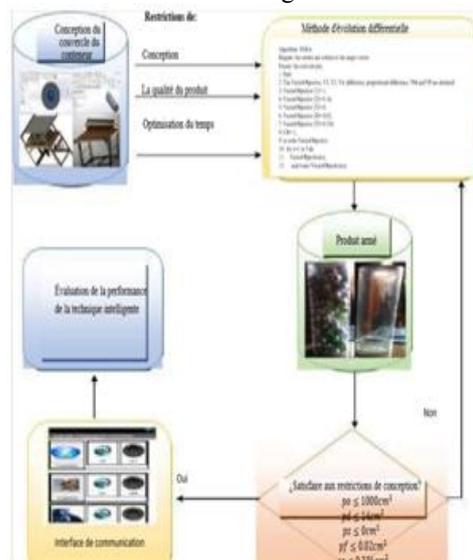


Fig.1. Processus pour la conception optimale du contenant. Élaboration propre.

Fonction proposée pour le calcul optimal du blister:

$$\text{Minf}(p) = (-pd + \alpha po - ps - \beta pf) + sc, p \in R^5(1)$$

Défini:

- f: Fitness ou fitness
- pd: zone de déchets
- pf: aire en dehors du rectangle pénalisé
- po: zone occupée
- ps: zone de chevauchement
- sc: surface latérale du cylindre
- β : 2000

Soit alors le vecteur de variables de conception pour le mécanisme donné par:

$$p = [p1, p2, p3, p4, p5]^T(2)$$

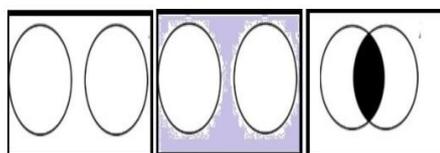


Fig.2. a) Zone occupée b) zone de déchets c) zone superpose

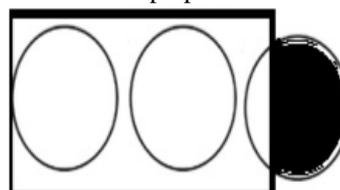


Fig.3. d) Zone en dehors du rectangle pénalisé
 Les restrictions du problème au cours de l'évolution sont soumises à:

$$\begin{aligned} p_o &\leq 1000\text{cm}^2(3) \\ p_d &\leq 14\text{cm}^2 \quad (4) \\ p_s &\leq 0\text{cm}^2(5) \\ p_f &\leq 0.02\text{cm}^2(6) \\ s_c &\leq 0.336\text{cm}^2(7) \end{aligned}$$

IV. MÉTHODOLOGIE

Les algorithmes de recherche stochastiques évolutifs sont basés sur le principe de la survie du plus apte de Charles Darwin, composé des composants suivants:

- Structures qui se répliquent (individus)
- Une fonction fitness
- Un mécanisme de sélection (probabiliste)
- Opérateurs agissant sur des individus (croisements ou mutations)

Les algorithmes évolutifs partent d'un ensemble de solutions générées aléatoirement (populations), pour chaque solution, nous calculons une valeur correspondant généralement à la fonction objectif que nous souhaitons optimiser. Cette valeur s'appelle aptitude. Par la suite, un ensemble de parents est sélectionné auquel est appliqué au moins un opérateur génétique (croisement ou mutation), avec une certaine probabilité, à savoir que l'opérateur ne s'applique pas à tous les parents. Les nouveaux individus s'appellent des enfants et plus tard ils mutent. Ces nouveaux individus constituent une nouvelle population qui remplacera partiellement ou totalement la précédente.

V. LES RÉSULTATS

Le modèle commence par les vecteurs de paramètres p population de l'algorithme d'évolution différentielle, avec vecteurs de paramètre NP de dimension D, où chaque paramètre peut être limité à une plage de valeurs définie par la restriction 3 à la 7. Le tableau 1 montre la population des individus qui la composent.

Table 1. Population d'individus

Population	Ind.1	Ind. 2	Ind. 3	Ind. 4	Ind. 5
P1	1	0.440813	0.991445	0.120522	0.015856
P2	0.14	0.035339	0.142988	0.383743	0.217033
P3	0	0.550456	0.973029	0.667508	0.285156
P4	0.02	0.861895	0.934426	0.948631	0.171407
P5	0.336	0.931739	0.009910	0.778922	1.221688
F(p)	1.4962	2.820245	3.051800	2.899327	0.911142

Au moyen de la mutation des différences, on ajoute la différence proportionnelle de deux individus V2 et V3, choisis au hasard dans la population (V1, individu, cible), également choisis au hasard. Le

nouvel individu M_i s'appelle individu mute ou vecteur muté.

$$M_i = V1 + \mu(V2 - V3)(8)$$

Le tableau 2 montre le calcul du vecteur de la différence pondérée entre les individus 2 et 5, ainsi que la différence proportionnelle résultant de la multiplication de μ par 0.80.

Table 2. Calcul du vecteur différence pondérée

Popu	Ind. 2	Ind. 5	Difference	Difference proportionnelle
P1	0.440813	0.015856	0.4249568	0.339965
P2	0.035339	0.217033	-0.181694	-0.145355
P3	0.550456	0.285156	0.265300	0.212240
P4	0.861895	0.171407	0.690488	0.552390
P5	0.931739	1.221688	0.710051	0.568041
F(p)	2.820245	0.911142	1.909102	1.527282

La constante de mutation $\mu > 0$ établit la plage de différenciation entre les individus V2 et V3 dans le but d'éviter une stagnation du processus de recherche. Après la mutation, une opération de recombinaison a été effectuée sur chaque individu V1 (cible) pour générer un individu ou un vecteur test F_i , l'individu à tester est F contrôlé par la fonction de condition établie dans 9:

$$F_i(j) = \begin{cases} M_i(j), & \text{if } \text{rand} \leq CR \\ V_i(j) \end{cases} \quad (9)$$

L'opérateur de sélection décide en fonction de la meilleure forme physique, si le test individuel F_i est accepté et remplace le test individuel F est rejeté et l'individu cible est retenu dans la génération suivante.

M_i le vecteur muté et le vecteur cible V_i sélectionnent un nombre compris entre 0 et 1, si la condition de la formule 9 n'est pas remplie, la première position passe la deuxième place et passe à la première place du vecteur test. Le tableau 3 montre la population des individus qui la composent. Dans le tableau 3. Nous trouvons le calcul effectué pour obtenir le vecteur mute.

Table 3. Obtention du vecteur mute

Population	Difference proportionnelle	Individual Mutated Vector	M_i
P1	0.339965	0.120522	0.460487
P2	-0.145355	0.383743	0.238388
P3	0.212240	+ 0.667508	0.879748
P4	0.552390	0.948631	1.501021
P5	0.568041	0.778922	1.346963
F(p)	1.527282	2.899327	4.426610

L'index nous donne la position de chaque individu, génère des valeurs à vérifier par

rapport à CR et, en fonction de la condition, la valeur doit faire partie du vecteur de test, puis un nombre aléatoire est pris entre 0 et 1 et attribué à CR. Ensuite, un nombre aléatoire est obtenu et cette donnée est comparée à CR si elle est plus petite, nous passons la valeur du vecteur muté et sans que la condition de l'équation 9 ne soit remplie.

Table 4. Calcul du vecteur de test

Population	Vecteur cible (V_i)	Vecteur Test muté (M_i)	de Vecteur (F_i)
P1	1	0.460487	1
P2	0.14	0.238388	0.14
P3	0	CR = 0.879748	0.3349
P4	0.02	1.501021	0.02
P5	0.336	1.346963	0.336
F(p)	1.496	4.426610	1.496

Lors de la comparaison du résultat du vecteur test avec le vecteur objectif, nous avons constaté qu'il n'y avait pas de changement dans le vecteur test; mais lorsque $Cr = 1$, le vecteur de test est calculé et la première génération est obtenue car le résultat du vecteur de test est inférieur au vecteur cible, comme le montre le tableau 5 donnant lieu à la première génération de la présente étude.

Table 5. Calcul du vecteur de test pour la première génération quand $F = 1.054876$ $Cr = 1$.

Population	Vecteur cible (V_i)	Vecteur muté (M_i)	Test de Vecteur (F_i)
P1	1	0.460487	0.460487
P2	0.14	0.238388	0.238388
P3	0	0.879748	0
		CR = 1	
P4	0.02	1.501021	0.02
P5	0.336	1.346963	0.336
F(p)	1.496	4.426610	1.054876

La conception des couvercles supérieur et inférieur du conteneur du blister a été développée dans SolidWorks 2015 pour son élaboration ultérieure du moule en aluminium pour le thermoformage du matériau en PVC.

Algorithm 1 NOEA

Require: the entries are written to the target vector.

Ensure: the exist are put.

1: Start

```

2: The VectorObjective, V2, V3, V4, difference,
   proportional difference, VM and VP are declared
3: VectorObjective [1]= 1;
4: VectorObjective [2]= 0.14;
5: VectorObjective[3]= 0;
6: VectorObjective [4]= 0.02;
7: VectorObjective [5]= 0.336;
8: CR= 1;
9: to write VectorObjective
10:for x=1 to 5do
11:VectorObjective(x);
12: sum=sum+VectorObjective(x);
13: end for
14: Write "Sum of Objective Vector:", sum
15: Write "Vector Values V2"
16:for x=1 to 5do
17: V2(x)=rand();
18: Write V2(x);
19: end for
20: Write "Vector Values V3"
21:forx=1 to 5do
22: V3(x)=rand();
23: Write V3(x);
24: end for
25: Write "Vector Values V4"
26:for x=1 to 5do
27: V4(x)=rand();
28: Write V4(x);
29: end for
30: Write "The Vector Difference data are:"
31:forx=1 to 5do
32:difference(x)=V2(x)-V3;
33:Write difference(x);
34: end for
35: Write "Values of the proportional difference:"
36: for x=1 to 5 do
37:ProportionalDifference(x)=difference(x)*0.80;
38: Write difference(x);
39: end for
40:Write "Values of the Muted Vector:"
41: for x=1 to 5 do
42: VM(x)=ProportionalDifference(x)+V4(x);
43: Write VM(x);
44: end for
45:NCR=rand();
46:Write NCR;
47:if NCR<CR then
48: VP(1)=VM(1);
49:else
50: VP(1)=VectorObejtive(1);
51:end if
52:NCR=rand();
53:Write NCR;
54:if NCR<CR then
55: VP(2)=VM(2);
56:else
57: VP(2)=VectorObejtive(2);
58:end if
    
```

```
59:NCR=rand();
60:Write NCR;
61:if NCR<CR then
62:  VP(3)=VM(3);
63:Else
64:  VP(3)=VectorObejtive(3);
65:end if
66:NCR=rand();
67:Write NCR;
68:if NCR<CR then
69:  VP(4)=VM(4);
70:Else
71:  VP(4)=VectorObejtive(4);
72:end if
73:NCR=rand();
74:Write NCR;
75:if NCR<CR then
76:  VP(5)=VM(5);
77:Else
78:  VP(5)=VectorObejtive(5);
79:end if
80:  VP(1)=VectorObejtive(1);
81:end if
82:for x=1 to 5do
83:sumVP=sumVP+VP(x);
84:end for
85:Write "Algorithm result"
86:if sumVP<sum then
87:Write "The new generation is given by the test
  vector VP";
88:for x=1 to 5do
89:Escribir VP(x);
90:end for
91:Write "Sum of the test vector is:", sumVP;
92:Else
93:"the values of the target vector of the previous
  generation are left:";
94:Write VectorObjetive(x);
95:end if
```

Le logiciel Overleaf a été utilisé pour le développement de l'algorithme NOEA. Vous pouvez consulter l'algorithme occupé, il est programmé dans Octave, son temps d'exécution varie entre 9.20 et 9.80 secondes.

Zapata, et al. 2017 [7] a adapté un algorithme L-SHADE automatiquement aux paramètres avec mutation (F) et au croisement (CR) à travers via un mémoire de paramètres, contrôlant la population (NP) à l'aide d'une fonction linéaire, il s'incorpore une stratégie de gestion de restriction à L-SHADE, afin d'optimiser trois études de cas du mécanisme à quatre barres appelé CLSHADE, comparant leurs performances à celles de DE / rand / 1 / bin en utilisant des paramètres aléatoires pour résoudre des problèmes d'optimisation mécanique, a effectué une expérience. Ils ont comparé les résultats obtenus avec les deux algorithmes et un autre analysant le

comportement de convergence de chacun: pour valider les données obtenues, ils ont utilisé le test de Wilcoxon rank sum avec un niveau de signification de 95%. C-LSHADE est arrivé à un meilleur résultat de DE / rand / 1 / bin, a conclu que C-LSHADE est capable de trouver de meilleures solutions ou égal à DE, le pourcentage d'évaluations de la fonction objectif est inférieur à DE ajuster les paramètres F, CR et NP. La modification principale consistait à ajouter aux critères de faisabilité proposés par DEB à l'étape correspondante du meilleur individu, l'algorithme était capable de trouver de meilleures solutions plus rapidement et permettait à l'utilisateur de procéder à des ajustements manuels des paramètres, sur la base d'essais et d'erreurs. Brest et al. 2016 [8 et 9] qui a proposé une version étendue de L-SHADE appelée iL-SHADE pour résoudre les problèmes d'optimisation des paramètres réels d'un objectif unique. Ils ont effectué les expériences dans 30 fonctions de référence et quadrimensionnelles, se révélant très compétitifs par rapport à la version originale de L-SHADE.

L'algorithme iL-SHADE différait de L-SHADE en utilisant les mécanismes suivants: Le mécanisme de mise à jour stockait les valeurs de mémoire historique de la génération précédente et les utilisait pour calculer les valeurs de la mémoire historique de la génération suivante entrées, et l'un d'eux contient des valeurs qui sont fixes. Cette entrée n'est pas mise à jour mais les valeurs sont utilisées pour générer les paramètres de contrôle CR_i et F_i . Toutes les valeurs de mémoire historique dans M_{CR} sont initialisées à 0.8 la valeur terminale, alors M_{CR} est réinitialisée à 0.0.

VI. CONCLUSIONS

Lors de la vérification de la valeur du vecteur de test avec CR, on peut voir dans le tableau 5 que la somme du vecteur de test est 1.054876, donné dont la valeur est inférieure à la somme du vecteur cible, donnant lieu à la première génération lorsque le facteur de croisement est égal à 1, une amélioration de la conception des couvercles de blister a donc été obtenue à la cinquième itération. L'algorithme NOEA est simple à mettre en œuvre. Il se caractérise par des recherches globales, il est basé sur les règles de faisabilité de Deb, il ne nécessite pas de ressources informatiques excessives et son temps de réponse est raisonnable.

REFERENCES

- [1]. García - de Loera, F.: Création de sphères à Chignahuapan. <http://wikipebla.poblanerias.com/galeria-elaboracion-de-esferas-en-chignahuapan/>, (Récupéré le 31 juillet 2017)
- [2]. Follows: Abattre et sceller les conteneurs. Elsevier. 26. pp. 782-803 (2009)

- [3]. Follows: Abattage et fermeture des conteneurs. Elsevier. 25. pp. 1045-1076 (2017)
- [4]. Mohammad S. I. and Liyong T.: Effets du rayon initial de la cloque et du diamètre de l'arbre sur le taux de libération d'énergie du revêtement composite métal-polymère, *International Journal of Adhesion and Adhesives*, (2015), pp 1-35. <http://dx.doi.org/10.1016/j.ijadhadh.2015.07.007>.
- [5]. Li, Y. and Chen, Y.: Modélisation dans les emballages pharmaceutiques, Abbvie Inc., North Chicago, IL, États-Unis, Novast Laboratories, Ltd., Nantong, Chine. 12. pp. 317-341 (2017)
- [6]. Vega-Alvarado, E., Santiago-Valentín, E., Sánchez-Márquez, A., Portilla-Flores EA., Flores-Pulido, L.: Synthèse optimale d'un mécanisme plat pour le suivi de trajectoire utilisant l'évolution différentielle. *En science informatique*, pp. 85-98 (2014)
- [7]. Zapata-Zapata, M-F., Mezura-Montes, E., Portilla-Flores E-A.: Evolution différentielle avec mémoire de paramètres pour l'optimisation de mécanismes à quatre barres. *En sciences informatiques*, pp. 9-22 (2017)
- [8]. Brest, J., Sepesy, M.M., Bosković, B.: iL-SHADE: Algorithme L-SHADE amélioré pour l'optimisation des paramètres réels à un seul objectif. Dans: IEEE, F., rédacteur en chef, S. (éd.) *Congres Sur Le Calcul De L'Evolution 2016*, pp. 1188-1195. Vancouver, BC, Canada, <https://doi.org/10.1109/CEC.2016.7743922>
- [9]. Brest, J., Sepesy, M.M., Bosković, B.: iL-SHADE: Algorithme L-SHADE amélioré pour l'optimisation des paramètres réels à un seul objectif. Dans: IEEE, F., rédacteur en chef, S. (éd.) *Congres Sur Le Calcul De L'Évolutionnaire 2016*, pp. 1188-1195. Vancouver, BC, Canada, <https://doi.org/10.1109/CEC.2016.7743922>

Noemí González-León "Optimisation des conteneurs pour les sphères de Noël, par évolution différentielle" *International Journal of Engineering Research and Applications (IJERA)*, vol. 9, no. 10, 2019, pp 47-52