



**HAL**  
open science

## Optimisation de placement dans les problèmes de conception

Guillaume Jacquenot, Fouad Bennis, Jean-Jacques Maisonneuve, Philippe Wenger

► **To cite this version:**

Guillaume Jacquenot, Fouad Bennis, Jean-Jacques Maisonneuve, Philippe Wenger. Optimisation de placement dans les problèmes de conception. CFM 2009 - 19ème Congrès Français de Mécanique, Aug 2009, Marseille, France. pp.6. hal-03391298

**HAL Id: hal-03391298**

**<https://hal.science/hal-03391298v1>**

Submitted on 21 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Optimisation de placement dans les problèmes de conception

GUILLAUME JACQUENOT <sup>a,b</sup>, FOUAD BENNIS <sup>a</sup>, JEAN-JACQUES MAISONNEUVE <sup>b</sup>, PHILIPPE WENGER <sup>b</sup>

*a. Ecole Centrale de Nantes, IRCCyN, UMR CNRS 6597, Nantes, France*

*b. SIREHNA, Nantes, France*

## Résumé :

*Dans les problèmes de conception, l'optimisation du placement des composants peut jouer un rôle prédéterminant sur les performances du système à concevoir. Le papier présenté propose une méthode de placement des composants, basée sur un algorithme évolutionnaire couplé à une méthode de séparation des composants. La méthode permet de traiter les problèmes multi-objectifs contraints, et permet au concepteur d'interagir avec les solutions proposées.*

## Abstract :

*In a design problem, component layout plays a major role in the performance of the global system. This paper presents a placement method, based on an evolutionary algorithm and a separation algorithm method. The method allows to treat constrained multi-objective problems, and allows the designer to interact with the proposed solutions.*

**Mots clefs :** Problèmes de placement, d'agencement, Optimisation multi-objectif, Optimisation continue, Algorithme génétique, Algorithme séparation

## 1 Introduction

Les problèmes de placement consistent à positionner un ensemble de composants dans l'espace, de manière à minimiser un ensemble de fonctions objectifs sous certaines contraintes. Tous ces problèmes sont caractérisés par des contraintes de non-chevauchement entre composants ainsi que des contraintes d'appartenance au contenant. Ces contraintes définissent une partie des contraintes de placement du problème.

Il existe principalement deux types de problèmes de placement : les problèmes de découpe et d'emballage (*Cutting & Packing problem*) et les problèmes d'agencement (*Layout problem*). Pour les premiers, les composants sont uniquement géométriquement reliés entre eux. Pour les seconds, les composants sont géométriquement et fonctionnellement reliés entre eux.

Cagan *et al.* [1] ont rédigé un état de l'art sur les différentes manières d'aborder les problèmes d'agencement. Aladahalli *et al.* [2, 3] ont développé une méthode de recherche par motifs (*Pattern Search*) pour la résolution des problèmes d'agencement, où objectifs et contraintes sont agrégés dans une seule et même fonction. Yi *et al.* [4] ont utilisé un algorithme génétique multi-objectifs (NSGA-II) pour positionner les différents composants d'un châssis de camion. Un modèle CAO est utilisé pour modéliser le problème et effectuer les calculs géométriques. Les contraintes de non-pénétration entre composants sont prises en compte en pénalisant le rang de chaque solution. Tiwari *et al.* [5] ont proposé une heuristique de placement de type *bottom-left* associée à un algorithme génétique pour résoudre un problème de chargement 3D. La représentation des composants sous forme de voxels leur permet de résoudre des problèmes avec des géométries complexes. D'autres approches basées sur la programmation par contraintes ont permis de résoudre des problèmes de placement formulés comme des problèmes de satisfaction de contraintes [6].

Le papier s'articule en deux points : le premier présente la méthode proposée, les différents outils mis en œuvre pour la résolution des problèmes de placement et le second montre l'application de la méthode sur trois exemples, tous bi-dimensionnels.

## 2 Méthode proposée

La modélisation des problèmes de placement que nous avons choisie est basée sur une méthode de placement relaxé, qui autorise le non-respect des contraintes de placement lors de la résolution du problème. Ce choix présente plusieurs avantages : la modélisation peut ainsi s'adapter à une grande variété de problèmes, les variables d'optimisation sont directement les variables de positionnement des composants, ce qui permet au concepteur d'interagir avec les solutions proposées. La difficulté résidera dans la satisfaction des contraintes de placement. Une solution proposée qui satisfait ces contraintes sera dite admissible ou réalisable.

La méthode d'optimisation proposée consiste en l'hybridation d'un algorithme d'optimisation globale et d'un algorithme de séparation. L'algorithme d'optimisation globale est chargé d'explorer efficacement l'espace de conception et l'algorithme de séparation a pour mission de rendre admissibles autant que possible les solutions proposées par l'optimiseur global. L'utilisation d'algorithmes d'optimisation globale est quelque chose de classique dans la résolution des problèmes de placement. De nombreuses méthodes utilisant des algorithmes évolutionnaires ont été proposées [4, 5]. En revanche, le couplage d'un algorithme évolutionnaire et d'une méthode de séparation est quelque chose de nouveau, et permet de trouver des solutions réalisables rapidement. L'algorithme génétique *OmniOptimizer* proposé par Deb *et al.* [7] est utilisé comme optimiseur global. Version évoluée de l'algorithme génétique NSGA-2, il est adapté à la résolution de problèmes mono- et multi-objectifs. Les opérateurs de croisement SBX et de mutation polynomiale sont utilisés pour générer de nouvelles variables réelles. Les variables discrètes utilisent l'opérateur de croisement à deux points, ainsi que l'échange de bit pour la mutation. Un opérateur d'échange a été ajouté aux opérateurs classiques de croisement et de mutation. Il permet d'échanger la position de deux composants et s'avère très utile dans les problèmes à forte compacité. L'algorithme de séparation est chargé, à partir d'une solution initiale ne respectant pas les contraintes de placement, de déplacer localement les composants afin de minimiser une fonction traduisant le non-respect de ces contraintes. Il existe plusieurs variantes de cet algorithme suivant la représentation géométrique choisie, mais l'idée centrale est identique : il s'agit de concevoir un programme d'optimisation continue sans contrainte qui minimise une fonction  $F$  exprimant le non-respect des contraintes de placement, à savoir les contraintes de non-pénétration et les contraintes d'appartenance. Concrètement, cet algorithme intervient juste avant l'évaluation des objectifs et des contraintes d'une solution. Si les contraintes de placement ne sont pas satisfaites, l'algorithme de séparation modifie les variables.

Dans notre cas, où l'on considère des polygones de formes quelconques, l'algorithme de séparation cherche à minimiser la somme des profondeurs de pénétrations entre composants. La profondeur de pénétration entre deux composants est définie comme la distance de translation minimale pour que ces deux composants ne s'intersectent plus. L'algorithme de séparation translate uniquement les composants de manière à ce que les solutions proposées par l'algorithme évolutionnaire satisfassent au mieux les contraintes de placement. Dans le cas où les composants sont des rectangles ou des parallélépipèdes, l'algorithme de séparation cherche à minimiser la somme des volumes d'intersection entre chaque paire de composants. Enfin dans la situation où les composants sont libres de toute rotation ou dans le cas 3D, les composants sont convertis en assemblages de cercles/sphères pour minimiser la somme des profondeurs de pénétration des primitives élémentaires.

Pour plus de détails sur l'implémentation de cet algorithme et le calcul analytique du gradient de la fonction à minimiser, le lecteur peut se référer à Imamichi *et al.* [8, 9]. La méthode quasi-Newton L-BFGS proposée par Liu *et al.* [10] est utilisée pour la résolution de ce problème d'optimisation locale.

### 3 Exemples

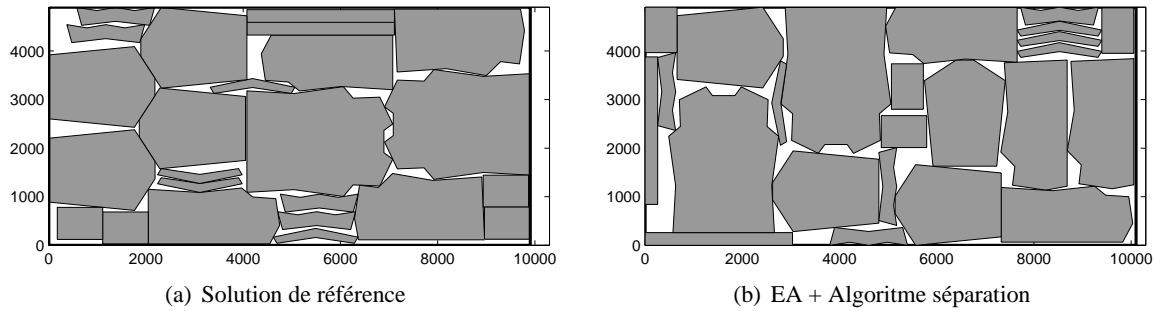
#### 3.1 Test sur un problème de la littérature

Dans ce premier exemple, on compare les résultats obtenus avec notre méthode et les solutions de référence pour un problème de découpe de forme irrégulières. L'objectif d'un tel problème est de découper un ensemble de composants dans un contenant de taille minimum. On compare nos résultats sur l'instance ALBANO [11]<sup>1</sup>. Chacun des 24 composants peut avoir quatre orientations discrètes réparties tous les 90°. La solution de référence a été proposée par Bennell *et al.* [12] à l'aide de leur méthode de recherche par faisceaux (*Beam search*). La figure 1 présente les résultats graphiques et numériques. La solution trouvée est à 1.4% de la solution de référence, ce qui est correct pour une méthode non dédiée à ce type de problème.

#### 3.2 2<sup>ème</sup> exemple

Dans ce second exemple, on considère un problème bi-objectif avec 10 composants ayant les mêmes degrés de liberté que les composants du premier exemple. La figure 2 présente la géométrie du contenant, de même que la géométrie des différents types de composants. Le tableau 1 donne les caractéristiques des différents composants. Les composants de même type auront par la suite les mêmes propriétés. La densité des composants sera représentée par le niveau de gris des composants : Plus un composant sera foncé, plus sa densité sera importante. Avec 10 composants, le problème possède 20 variables réelles qui codent la position des composants et 10 variables discrètes représentant l'orientation discrètes des composants. Avec un contenant d'une aire de 1123, la compacité du problème est de 59.7%. Les paramètres de l'algorithme sont donnés table 2. Les objectifs de ce problème consistent à minimiser l'inertie  $I_{yy}$  autour de l'axe de symétrie du contenant, et maximiser les distances euclidiennes minimales respectivement entre les deux composants de type  $T1$ , ainsi que ceux de type  $T5$ . Dans cet exemple, les deux composants de type  $T2$  sont notés  $C_5$  et  $C_6$ , et ceux de type  $T5$  sont notés  $C_7$  et  $C_8$ . (Voir figure 4)

<sup>1</sup>Les données de cette instance sont disponibles sur le site Internet de l'ESICUP (*EURO Special Interest Group on Cutting and Packing*) <http://www.fe.up.pt/esicup>



	Méthode	Compacité	Longueur
1	Solution de référence [12]	87.88%	9906.44
2	EA + Algorithme séparation	86.62%	10050

(c) Comparaison des résultats numériques

FIG. 1 – Comparaison de solutions obtenues sur un problème de découpe de formes irrégulières. (Instance ALBANO)

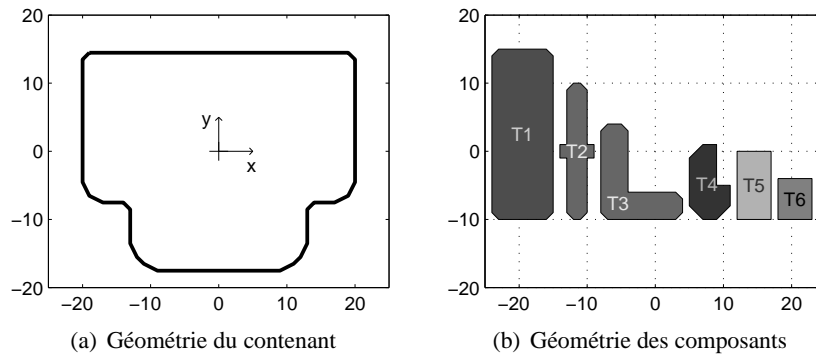


FIG. 2 – Géométrie du contenant et des différents composants.

Type de composants		T1	T2	T3	T4	T5	T6
Nombre de composants	$n_i$	1	2	1	1	2	3
Aire des composants	$A_i$	223	62	85.5	48	50	30
Densité des composants	$\rho_i$	0.7	0.6	0.6	0.8	0.3	0.5

TAB. 1 – Propriétés des composants

Les objectifs du problème s'écrivent mathématiquement :

$$\mathcal{P} \begin{cases} \min f_1(\mathbf{v}) = \sum_{i=1}^n \rho_i I_{i/yy} \\ \max f_2(\mathbf{v}) = d_{\min}(C_5, C_6) + d_{\min}(C_7, C_8) \\ \text{s.c. les contraintes de placement soient satisfaites} \end{cases} \quad (1)$$

où  $n$  désigne le nombre de composants du problème,  $I_{i/yy}$  caractérise l'inertie de surface du composant  $i$  autour de l'axe  $y$ .  $d_{\min}(C_i, C_j)$  correspond à la distance euclidienne minimale entre les composants  $C_i$  et  $C_j$ . La figure 4 présente quelques-unes des solutions du front de Pareto, notamment les deux solutions extrémales. L'exécution de plusieurs simulations a permis de montrer que la convergence du problème était peu ou pas sensible à la population initiale ainsi qu'aux différents paramètres de l'algorithme d'optimisation. Les différents indicateurs multi-objectifs de convergence (Diversité, Hypervolume, Représentation) donnent des faibles variations (3%) ce qui n'est pas forcément le cas, notamment lorsque la compacité du problème augmente comme dans l'exemple 3.

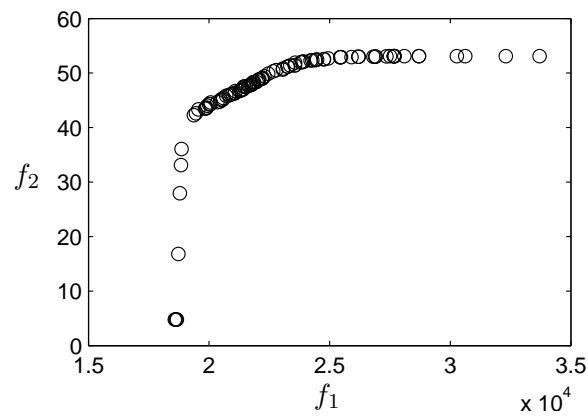


FIG. 3 – Front de Pareto obtenu pour le second exemple.

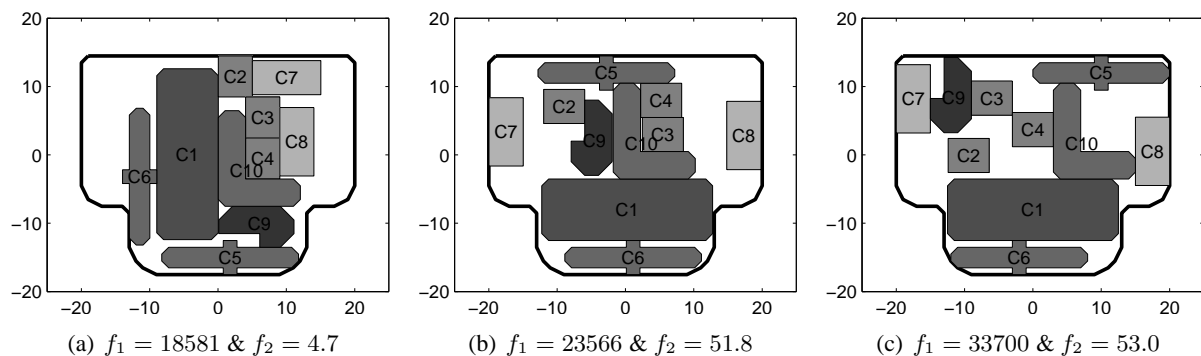


FIG. 4 – Solutions extraites des différents fronts de Pareto de la figure 3

Nombre de générations	100
Nombre d'individus	100
Probabilité de croisement des variables réelles	0.05
Probabilité de mutation des variables réelles	0.4
Probabilité de croisement des variables discrètes/binaires	0
Probabilité de mutation des variables discrètes/binaires	0.3
Probabilité d'échange de deux composants d'une même solution	0.05
Dispersion pour l'opérateur de croisement continu	5
Dispersion pour l'opérateur de mutation continu	5
Coefficient relatif pour l'utilisation de l' $\varepsilon$ -dominance	0.001
Utilisation du calcul des distances phénotypiques et génotypiques	Oui
Utilisation de l'opérateur de sélection restreinte	Oui

TAB. 2 – Configuration standard des paramètres de l'algorithme *OmniOptimizer*

### 3.3 3<sup>ème</sup> exemple

Dans ce troisième exemple, on considère toujours un problème d'optimisation bi-objectif avec les composants du problème 2, auxquels on ajoute un composant de type *T1* supplémentaire. La compacité passe alors de 59.7% à 79.5%. Le nombre de variables réelles passent à 22 et 11 variables discrètes codent à présent les orientations des composants. Le premier objectif consiste à minimiser la distance entre le centre de gravité de l'assemblage et le centre géométrique du contenant. Le second objectif est de minimiser l'inertie de l'assemblage autour de l'axe *y*, comme dans le deuxième exemple. Ces deux objectifs caractérisent un comportement statique et dynamique de l'assemblage.

$$\mathcal{P} \begin{cases} \min f_1(\mathbf{v}) = \sqrt{d_x^2 + d_y^2} \\ \min f_2(\mathbf{v}) = \sum_{i=1}^n \rho_i I_{i/yy} \\ \text{t.q. les contraintes de placement soient satisfaites} \end{cases} \quad \text{où} \quad \begin{cases} d_x = \frac{1}{\sum_{i=1}^n \rho_i A_i} \sum_{i=1}^n \rho_i A_i (x_i - x_G) \\ d_y = \frac{1}{\sum_{i=1}^n \rho_i A_i} \sum_{i=1}^n \rho_i A_i (y_i - y_G) \end{cases}$$

Des résultats préliminaires de ce problème ont été présentés dans [13]. Dans ce papier, nous avons montré que pour les problèmes à forte compacité, il devient difficile de trouver une solution réalisable. Par conséquent, la ou les solutions respectant au mieux les contraintes de placement exerce une pression de sélection importante qui entraîne une perte de la diversité de la population. Il arrive donc que l'optimiseur propose un ensemble de solutions efficaces très proches les unes des autres. Pour obtenir de meilleurs résultats, on peut étudier la sensibilité des paramètres de l'algorithme d'optimisation. On se propose ici d'analyser l'influence d'un des paramètres de l'algorithme génétique, à savoir la probabilité de mutation sur les variables réelles  $p_m$ . Pour chaque variation du paramètre  $p_m$ , on effectue 50 simulations et on extrait le front de Pareto de ces simulations. Les autres paramètres de l'algorithme sont donnés dans le tableau 2. La figure 5 illustre les différents fronts de Pareto obtenus pour des variations  $p_m$  allant de 0% à 100%. Chaque point de cette figure représente une solution réalisable, *c-à-d* une solution vérifiant les contraintes de placement. On observe sur cette figure une amélioration des résultats lors de la progression du paramètre de probabilité  $p_m$  allant de 0.0 jusqu'à 0.2, puis une dégradation de ces mêmes résultats pour des valeurs supérieures. Le problème apparaît donc très sensible aux réglages du paramètre  $p_m$ . La figure 6 présente quelques solutions extraites des fronts de Pareto de la figure 5. Même pour un problème à forte compacité, on peut observer la diversité des solutions qui existe.

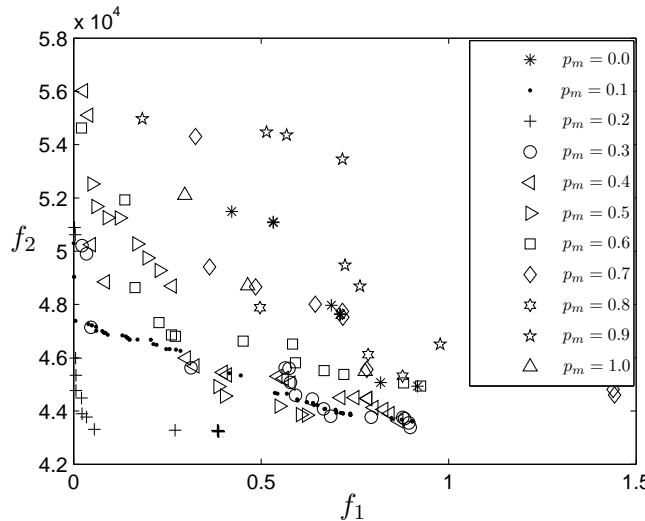


FIG. 5 – Influence du paramètre de probabilité de mutation sur les variables réelles

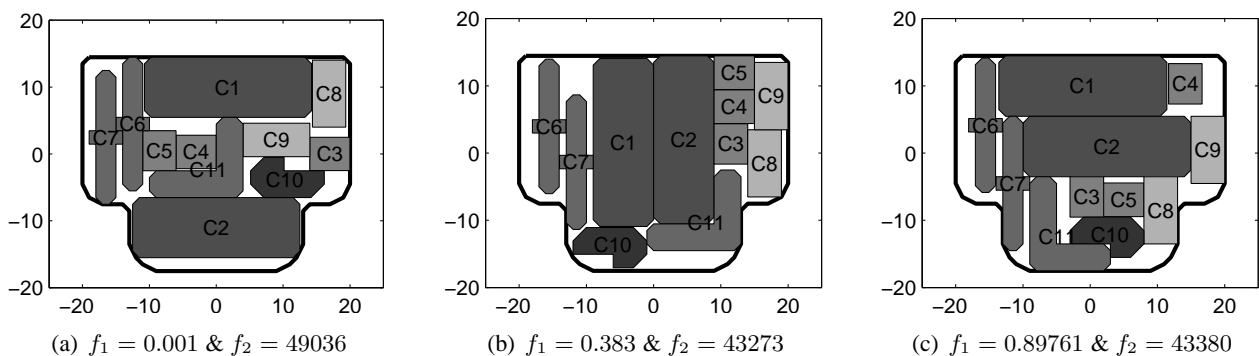


FIG. 6 – Solutions extraites des différents front de Pareto de la figure 5

Ces résultats montrent que l'influence du paramètre de probabilité de mutation joue un rôle important dans la convergence des problèmes. Une autre analyse non présentée ici montre que l'influence de la probabilité du paramètre d'échange est tout aussi importante. Pour les problèmes à forte compacité, les réglages de l'algorithme évolutionnaire ne sont pas triviaux. De plus grandes études statistiques sont nécessaires pour comprendre l'influence des paramètres de l'algorithme d'optimisation.

## 4 Conclusion et développements futurs

Ce papier a présenté une méthode de placement 2D pour des problèmes mono- et multi-objectifs. Hybridation d'un algorithme évolutionnaire multi-objectifs et d'un algorithme de séparation, notre méthode permet de résoudre une large palette de problèmes, et de s'approcher des résultats de référence de la littérature. Les exemples ont montré que les résultats dépendaient des paramètres de réglages de l'algorithme génétique, comme pour toutes les métaheuristiques. Un travail supplémentaire d'analyse des paramètres de réglages en fonction des données du problème est en cours de réalisation. Enfin, le choix d'une méthode de placement relaxé permet de nombreuses extensions comme la prise en compte de nouvelles contraintes entre composants, (contraintes topologiques, contraintes d'alignement entre composants, . . .) ou encore l'interaction entre le concepteur et l'optimiseur. L'intervention du concepteur dans la boucle d'optimisation lui permettra d'évaluer des solutions pour des problèmes pour lesquels certains objectifs ne peuvent pas être exprimés de manière mathématique. Elle lui permettra aussi de proposer à l'optimiseur des solutions qui lui semblent prometteuses, même si celles-ci ne satisfont pas entièrement les contraintes de placement.

## Références

- [1] Cagan J., Shimada K., et Yin S. A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design*, 34(8), 597–611, 2002.
- [2] Aladahalli C., Cagan J., et Shimada K. Objective function effect based pattern search - theoretical framework inspired by 3D component layout. *Journal of Mechanical Design*, 129(3), 243–254, Mars 2007.
- [3] Aladahalli C., Cagan J., et Shimada K. Objective function effect based pattern search - an implementation for 3D component layout. *Journal of Mechanical Design*, 129(3), 255–265, Mars 2007.
- [4] Miao Y., Fadel G. M., et Gantovnik V. B. Vehicle configuration design with a packing genetic algorithm. *International Journal of Heavy Vehicle Systems*, 15, 433–448(16), 12 2008.
- [5] Tiwari S., Fadel G., et Fenyes P. A fast and efficient three dimensional compact packing algorithm for free-form objects. In *Proceedings of ASME International Design Engineering & Computers and Information in Engineering Conference (ASME DETC/CIE 2008)*, New York, NY, 2008. DETC2008-50097. (Invited for publication in *ASME Journal of Computing and Information Science in Engineering*).
- [6] Beldiceanu N., Carlsson M., et Poder E. New filtering for the cumulative constraint in the context of non-overlapping rectangles. In Perron L. et Trick M. A., editors, CPAIOR, volume 5015 of *Lecture Notes in Computer Science*, pages 21–35. Springer, 2008.
- [7] Deb K. et Tiwari S. Omni-Optimizer : A generic Evolutionary Algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185, 1062–1087, Mars 2008.
- [8] Imamichi T., Yagiura M., et Nagamochi H. An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. In *Proceedings of the Third International Symposium on Scheduling*, Tokyo, Japan, number 9, pages 132–137, Juillet 2007.
- [9] Imamichi T. et Nagamochi H. Designing algorithms with multi-sphere scheme. In *ICKS '08 : Proceedings of the International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)*, pages 125–130, Washington, DC, USA, 2008. IEEE Computer Society.
- [10] Liu D. C. et Nocedal J. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(3), 503–528, 1989.
- [11] Albano A. et Sappupo G. Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *IEEE Transactions on Systems, Man and Cybernetics*, 5, 242–248, 1980.
- [12] Bennell J. A. et Song X. A beam search implementation for the irregular shape packing problem. *Journal of Heuristics*, 2007.
- [13] Jacquenot G., Bennis F., Maisonneuve J.-J., et Wenger P. Méthode générique pour la résolution des problèmes de placement bidimensionnel. In *AIP Primeca*, Mars 2009.