

# Méthode générique pour l'optimisation d'agencement géométrique et fonctionnel

Guillaume JACQUENOT

Directeurs de thèse: Fouad BENNIS & Philippe WENGER  
Encadrant industriel: Jean-Jacques MAISONNEUVE

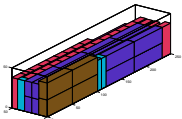
Le 18 janvier 2010, Amphithéâtre S, École Centrale de Nantes

Jury:

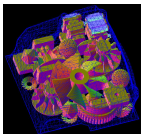
Président :	Georges FADEL	<i>Professeur, Clemson University, États-Unis d'Amérique</i>
Rapporteurs :	Georges FADEL Jean-Pierre NADEAU	<i>Professeur, Clemson University, États-Unis d'Amérique</i> <i>Professeur, TRÈFLE, ENSAM de Bordeaux</i>
Examineurs :	Carlo POLONI Fouad BENNIS Philippe WENGER Jean-Jacques MAISONNEUVE	<i>Professeur, Università degli Studi di Trieste, Italie</i> <i>Professeur, IRCCyN, École Centrale de Nantes, Nantes</i> <i>Directeur de recherche CNRS, IRCCyN, Nantes</i> <i>Docteur, SIREHNA, Nantes</i>
Invité :	Vitali TELERMAN	<i>Docteur, Dassault Systèmes, Vélizy-Villacoublay</i>

# Constat

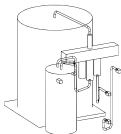
- ▶ Le **placement** des composants joue un **rôle déterminant** sur les performances d'un système à concevoir.
- ▶ Les problèmes de **placement** ont fait l'objet de **nombreux travaux** dans la littérature.
- ▶ À ce jour, de **nombreuses méthodes** ont été proposées pour des **problèmes spécifiques**.
- ▶ Objectif de la thèse : Proposer une méthode générique de placement



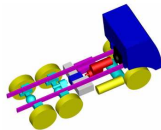
Chargement de containers  
[Pisi 02]



Conditionnement 3D  
[Tiwa 08]



Pompe à chaleur  
[Caga 96]



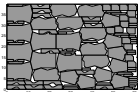
Agencement de composants  
[Miao 08]

# Plan de la présentation

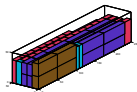
- 1 Présentation générale
- 2 Agencement de locaux
- 3 Modélisation générique proposée
- 4 Cas d'application
- 5 Conclusion

# Définition du problème de placement

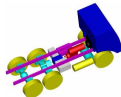
- ▶ Étant donné un **ensemble de composants** et un **contenant**, un problème de placement consiste à trouver l'ensemble des **variables de positionnement** des composants afin de minimiser un ensemble d'**objectifs**, tout en respectant certaines **contraintes**.
- ▶ Les **contraintes de placement** incluent entre autres :
  - ▶ les contraintes de **non-chevauchement** entre composants
  - ▶ les contraintes d'**appartenance** des composants au contenant



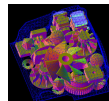
Problème découpe  
[Benn 07]



Chargement de containers  
[Pisi 02]



Agencement de composants  
[Miao 08]

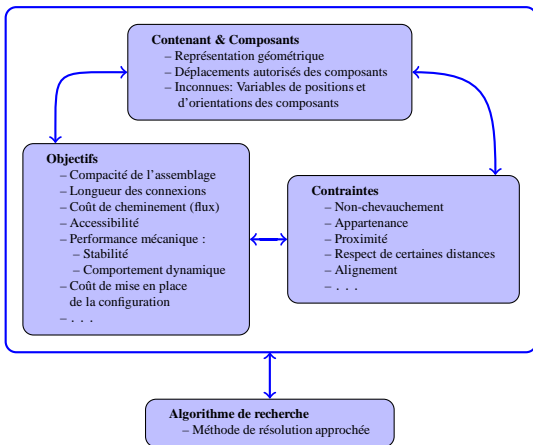


Empaquetage 3D  
[Tiwa 08]

# Problèmes de placement

Problèmes de **placement**:

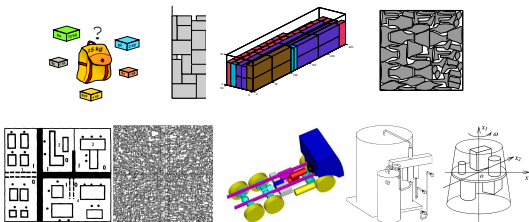
- ▶ Problème de **découpe et conditionnement**
- ▶ Problèmes d'**agencement**



Les **différences** entre ces problèmes déterminent les **méthodes de modélisation et résolution** utilisées.

# Exemples de problèmes de placement

- ▶ Problèmes de **découpe et de conditionnement**
- ▶ Problèmes d'**agencement**
  - ▶ Problème d'agencement de locaux (*FLP*)
  - ▶ Problème d'intégration à grande échelle (*VLSI*) [Egeb 03]
  - ▶ Problème d'agencement 3D dans l'ingénierie
    - ▶ Répartition des différents composants d'un camion [Yin 00, Miao 08]
    - ▶ Agencement des différents éléments d'une pompe à chaleur [Caga 96]
    - ▶ Placement des composants d'un satellite [Teng 01, Zhan 08]



# Méthodes de placement

Deux méthodes possibles :

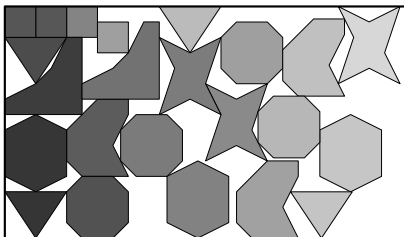
- ▶ Méthode de **placement "légal" (strict)**
  - ▶ Garantit le **respect des contraintes** de placement
  - ▶ Bien adaptée pour les problèmes avec des **objectifs de compacité**
  - ▶ Utilisation de schéma d'**encodage-décodage**
  - ▶ **Difficulté** de prise en compte des **contraintes spécifiques**
- ▶ Méthode de **placement relaxé**
  - ▶ Autorise le **non-respect des contraintes de placement** lors de la résolution
  - ▶ Variables du problème  $\Leftrightarrow$  Variables de position des composants
  - ▶ Méthode **générique** capable de modéliser tout problème

# Exemple de méthode de placement légal : Heuristique *Left-Bottom*

- ▶ Méthode de **placement séquentiel**
- ▶ Principe : Placement *le plus à gauche, le plus en bas*
- ▶ Variable : **Permutation**
- ▶ Espace de recherche **fini** (Cardinalité espace =  $n!$ )
- ▶ Couplé à un **algorithme stochastique** pour la résolution
- ▶ Plusieurs **représentations géométriques** possibles

## Exemple

Instance *Blaz* 1 – 28 composants



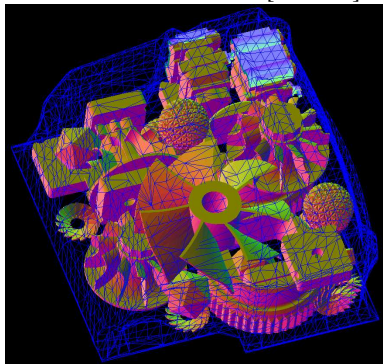
▶ [Planer animation](#)



# Exemple de méthode de placement légal : Heuristique *Bottom-Left*– Voxel

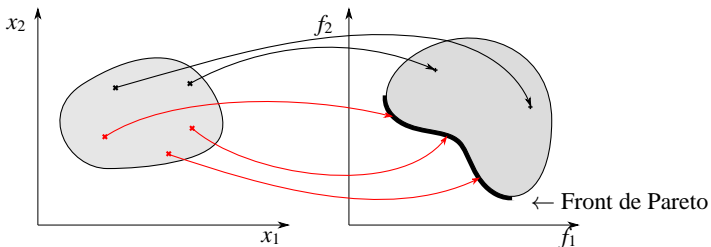
- ▶ **Discrétisation** des objets et du contenant
- ▶ **Parcours des emplacements** jusqu'à trouver une place
- ▶ Facilement adaptable au cas 3D

Travaux de Tiwari *et al.* [Tiwa 08]



# Principes de l'optimisation multi-objectif

$$(\mathcal{P}) \begin{cases} \min_{\mathbf{x}} f = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})) \\ \text{s.c.} \begin{cases} g_j(\mathbf{x}) \leq 0 & \forall j \in \{1, \dots, G\} \\ h_k(\mathbf{x}) = 0 & \forall k \in \{1, \dots, H\} \\ x_i^l \leq x_i \leq x_i^u & \forall i \in \{1, \dots, n\} \end{cases} \end{cases}$$

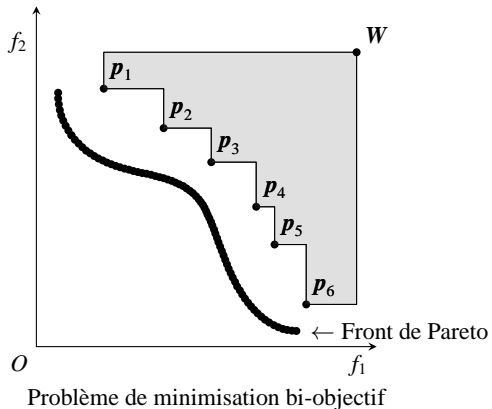


Espaces des variables et d'arrivée pour un problème de minimisation bi-objectif.

# Indicateurs multi-objectifs unaires

Comparaison numérique d'une **surface de compromis** par rapport au **front de Pareto**

- ▶ Hypervolume
- ▶ Diversité phénotypique



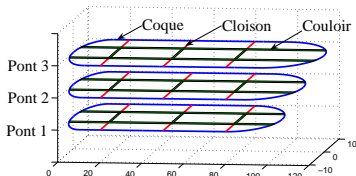
# Les problèmes de placement, un défi intéressant ...

Trois points clés pour la résolution des problèmes de placement :

- ▶ Formulation du problème
- ▶ Aspects géométriques
- ▶ Méthodes de résolution

# Objectifs du problème

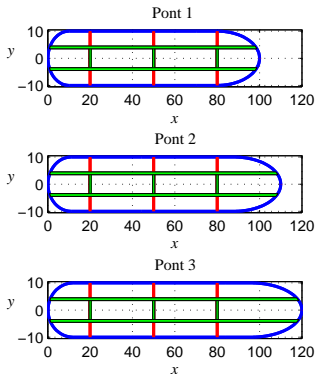
- ▶ Placer l'ensemble des compartiments à l'intérieur du navire
- ▶ Compartiments définis par leur surface au sol
- ▶ Objectifs :
  - ▶ Minimisation du flux entre compartiments
  - ▶ Minimisation de la fonction d'adjacence



# Choix de la méthode de placement

## Méthode de **placement séquentiel**

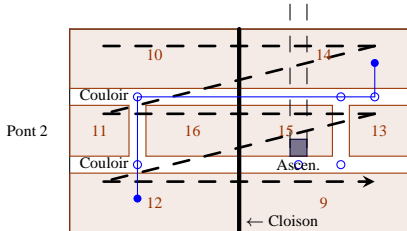
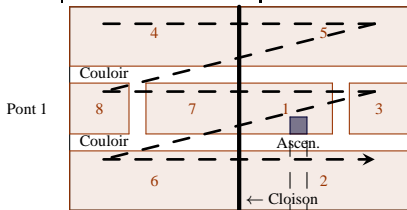
- ▶ Respect **automatique** des contraintes de placement
- ▶ Prise en compte de la **géométrie du navire**
- ▶ Problème **combinatoire**
- ▶ Nombre de solutions fini
- ▶  $78! \approx 10^{115}$  solutions
- ▶ Algorithme d'optimisation stochastique : **NSGA-II**



# Agencement de compartiments d'un navire

Encodage / Décodage des données

	Séquence des compartiments	Aire associée à chaque compartiment
Pont 1	4, 5, 8, 7, 1, 3, 6, 2	35, 35, 9, 15, 15, 9, 35, 35
Pont 2	10, 14, 11, 16, 15, 13, 12, 9	35, 35, 9, 15, 15, 9, 35, 35

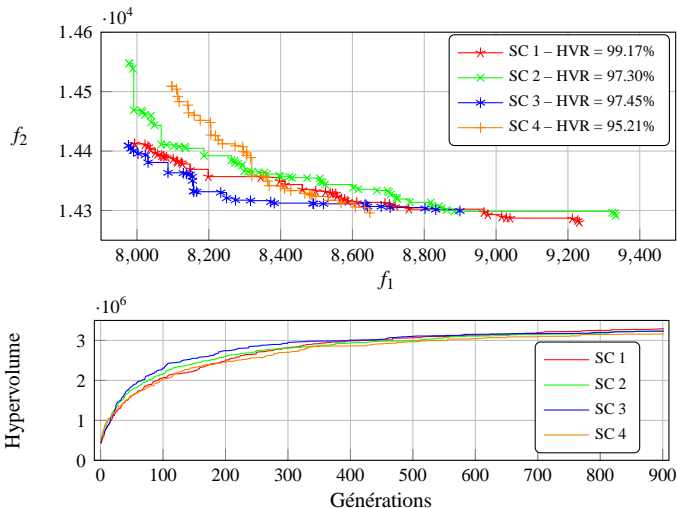


- Placement **un à un** des compartiments
- Calcul des distances avec l'algorithme de **Dijkstra** [Dijk 71]

# Évolution des surfaces de compromis pour 4 optimisations

4 optimisations avec 4 populations initiales différentes – 128 individus

Génération 900

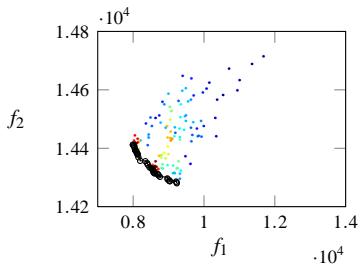




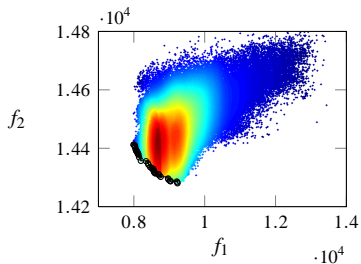
# Convergence algorithme génétique

Analyse de la **densité de répartition** des points dans l'espace des objectifs

900<sup>ème</sup> génération



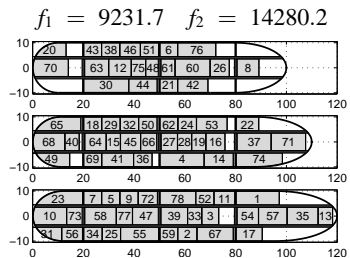
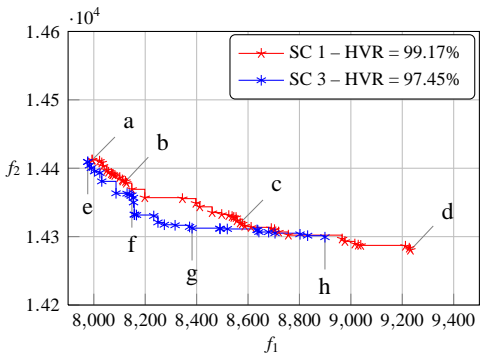
Génération courante



Ensemble des points

## Solutions du front de Pareto

- ▶  $\min f_1$ : Minimisation du flux entre compartiments
- ▶  $\min f_2$ : Minimisation de la fonction d'adjacence



# Conclusion

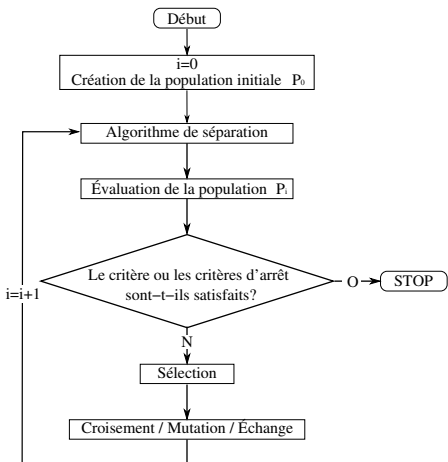
- ▶ **Méthode simple** pour la résolution **multi-objectif** de problèmes de placement avec des composants définis par une surface au sol et non une géométrie
- ▶ **Grande variété des solutions** permet au concepteur d'effectuer son **choix a posteriori**
- ▶ Modélisation applicable à **différents problèmes** comme
  - ▶ les problèmes de découpe et de conditionnement
  - ▶ les problèmes d'intégration à grande échelle (*VLSI*)
- ▶ **Toutefois**, la méthode **ne permet pas** de résoudre tous les problèmes de placement . . .

# Objectif et choix de la modélisation

- ▶ Objectif :
  - ▶ Proposer une **méthode de placement** capable de traiter des composants de **géométries complexes** pour la résolution de problèmes de placement **multi-objectifs**.
- ▶ Modélisation :
  - ▶ Choix d'une **méthode de placement relaxée**, pour sa souplesse d'adaptation à de nombreux problèmes.
  - ▶ Problème : Peu de chance que la ou les solutions proposées vérifie les contraintes de placement.
  - ▶ Idée : Utilisation d'une **méthode de séparation** pour minimiser le non-respect des contraintes de placement.
- ▶ Méthode proposée :
  - ▶ **Hybridation** d'un algorithme d'optimisation globale et d'un algorithme de séparation (optimisation locale).

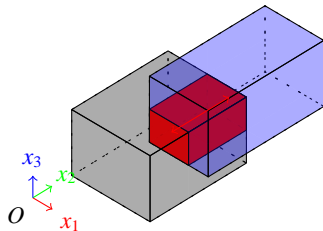
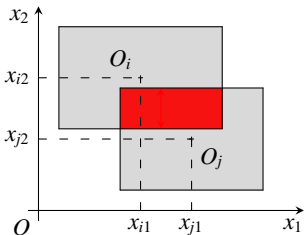
# Algorithme hybride proposé

- ▶ Algorithme génétique  
**OmniOptimizer** [Deb 08]
- ▶ Intégration d'un **algorithme de séparation**
- ▶ Modification des opérateurs génétiques
- ▶ Codage **réel** pour la position des composants
- ▶ Codage **réel** ou **discret** pour leur orientation.



# Algorithme de séparation

- ▶ L'algorithme de séparation est chargé de faire **respecter au mieux** les contraintes de placement en déplaçant **localement** les composants. [Imam 09]
  - ▶ Minimisation d'une **fonction continue** caractérisant le non-respect des contraintes
  - ▶ Calcul **analytique** du gradient de cette fonction
  - ▶ Optimisation à l'aide de la méthode **quasi-Newton BFGS**
- ▶ Fonction à minimiser **dépend des géométries** et des *ddl* des composants
- ▶ Minimisations basées sur **deux grandeurs**
  - ▶ Aire/Volume d'intersection  $\Gamma_{ij}$
  - ▶ Profondeur de pénétration  $\Delta_{ij}$



# Algorithme de séparation – Rectangles

Influence du choix de la fonction à minimiser pour le problème de séparation 2D

## Génération de différentes fonctions test

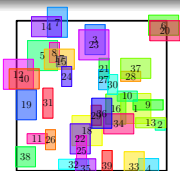
$$\begin{aligned}
 F_1(\mathbf{x}) &= \sum_{1 < i < j < m} \Gamma_{ij}^2(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Gamma}_i^2(\mathbf{x}) \\
 F_2(\mathbf{x}) &= \sum_{1 < i < j < m} \Delta_{ij}^2(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Delta}_i^2(\mathbf{x}) \\
 F_3(\mathbf{x}) &= \sum_{1 < i < j < m} \Gamma_{ij}(\mathbf{x}) \Delta_{ij}(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Gamma}_i(\mathbf{x}) \bar{\Delta}_i(\mathbf{x}) \\
 F_{\text{comp}}(\mathbf{x}) &= \sum_{1 < i < j < m} \Gamma_{ij}(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Gamma}_i(\mathbf{x})
 \end{aligned}$$

## Données exemple

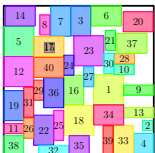
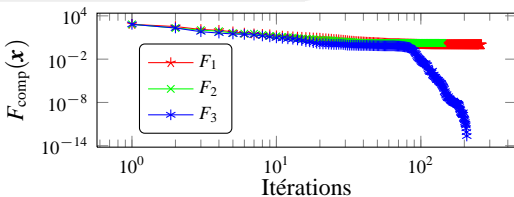
40 rectangles

80 dll

Compacité = 80.4%

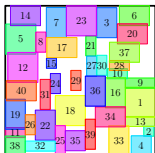


Solution originale



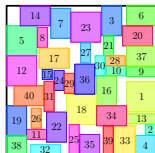
Solution optimale pour  $F_1$

$$F_{\text{comp}}(\mathbf{x}^*) = 1.0$$



Solution optimale pour  $F_2$

$$F_{\text{comp}}(\mathbf{x}^*) = 1.7$$



Solution optimale pour  $F_3$

$$F_{\text{comp}}(\mathbf{x}^*) = 0$$

# Algorithme de séparation – Parallélépipèdes

Influence du choix de la fonction à minimiser pour le problème de séparation 3D

## Génération de différentes fonctions test

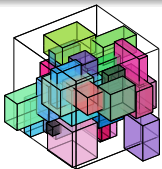
$$\begin{aligned}
 F_1(\mathbf{x}) &= \sum_{1 < i < j < m} \Gamma_{ij}^2(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Gamma}_i^2(\mathbf{x}) \\
 F_2(\mathbf{x}) &= \sum_{1 < i < j < m} \Delta_{ij}^2(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Delta}_i^2(\mathbf{x}) \\
 F_3(\mathbf{x}) &= \sum_{1 < i < j < m} \Gamma_{ij}(\mathbf{x}) \Delta_{ij}(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Gamma}_i(\mathbf{x}) \bar{\Delta}_i(\mathbf{x}) \\
 F_{\text{comp}}(\mathbf{x}) &= \sum_{1 < i < j < m} \Gamma_{ij}(\mathbf{x}) & + \sum_{1 < i < m} \bar{\Gamma}_i(\mathbf{x})
 \end{aligned}$$

## Données exemple

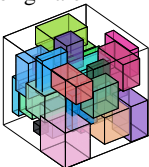
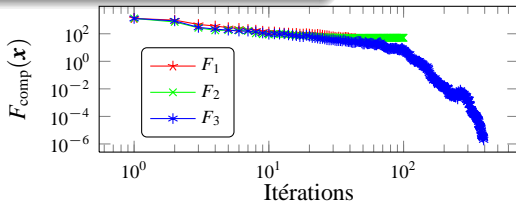
30 parallélépipèdes

90 dll

Compacité = 53.0%

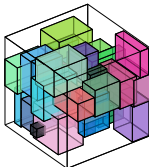


Solution originale



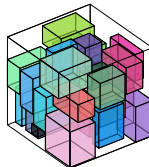
Solution optimale pour  $F_1$

$$F_{\text{comp}}(\mathbf{x}^*) = 15.9$$



Solution optimale pour  $F_2$

$$F_{\text{comp}}(\mathbf{x}^*) = 51.3$$



Solution optimale pour  $F_3$

$$F_{\text{comp}}(\mathbf{x}^*) = 1.9 \times 10^{-6}$$

Méthode générique pour l'optimisation d'agencement géométrique et fonctionnel



# Algorithmes de séparation – Polygones

## Séparation en translation

- ▶ Minimisation basée sur les **profondeurs de pénétration** entre polygones [Imam 09]
- ▶ Polygones de non-recouvrement (**NFP**) et polygones d'appartenance (**IFP**)

### Calcul des profondeurs de pénétration

Détection des **intersections**

$$\begin{aligned} NFP(P_i, P_j) &= \text{int}(P_i) \oplus (-\text{int}(P_j)) \\ &= \{v - w \mid v \in \text{int}(P_i), w \in \text{int}(P_j)\} \end{aligned}$$

$$\begin{aligned} \delta(P_i \oplus x_i, P_j \oplus x_j) &= \min \left\{ \|z\| \right. \\ &\quad \left. \mid \text{int}(P_i \oplus x_i) \cap (P_j \oplus x_j \oplus z) = \emptyset \right. \\ &\quad \left. z \in \mathbb{R}^2 \right\} \end{aligned}$$

Calcul de la **pénalité**

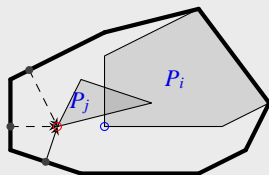
$$f_{ij}(x) = \delta(P_i \oplus x_i, P_j \oplus x_j)^2, 1 \leq i < j \leq m$$

Calcul **analytique** des gradients

$$\nabla_i f_{ij}(x) = -\nabla_j f_{ij}(x) = 2z$$

$$\nabla_k f_{ij}(x) = \mathbf{0}, k \in \{1, \dots, m\} \setminus \{i, j\}$$

### Illustration



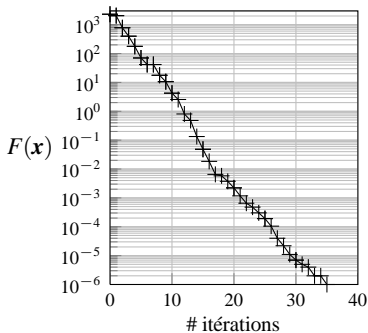
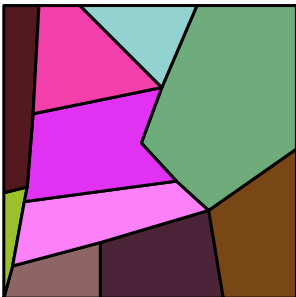
# Algorithme de séparation – Polygones

Séparation en translation

Minimisation des **profondeurs de pénétration** à l'aide des **NFP & IFP**

Instance *Dighe 2 – 20 ddl*

Itération : 35 –  $F(\mathbf{x}) = 1 \times 10^{-6}$



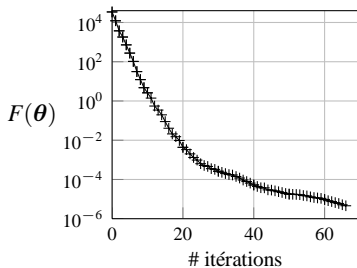
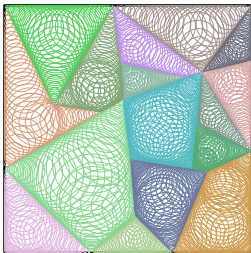
# Algorithme de séparation – Polygones

Séparation en rotation

Minimisation des **profondeurs de pénétration** entre chaque paire de cercles

Instance *Dighe 1 – 16 ddl*

Itération : 66 –  $F(\mathbf{x}) = 4 \times 10^{-6}$



# Algorithme de séparation – Adaptation de la méthode en 3D

La méthode présentée ne peut pas être directement utilisée pour les problèmes 3D

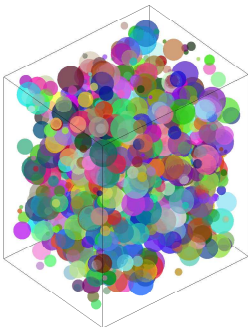
- ▶ Calcul des équivalents 3D des NFP et IFP difficile
- ▶ Représentation géométrique adaptée à chaque problème
  - ▶ Contenant et composants parallélépipédiques
  - ▶ Contenant parallélépipédique et composants quelconques
  - ▶ Contenant et composants quelconques
- ▶ Dans le cas quelconque, transformation des composants en assemblages de sphères

# Algorithme de séparation 3D – Composants sphériques

Séparation en translation

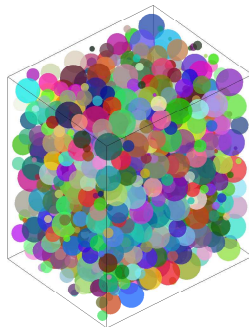
Minimisation des **profondeurs de pénétration** entre chaque paire de sphères

1000 sphères – 3000 variables



Solution originale

$$F(\mathbf{x}_0) = 5.63$$



Solution finale

$$F(\mathbf{x}_{120}) = 10^{-15}$$

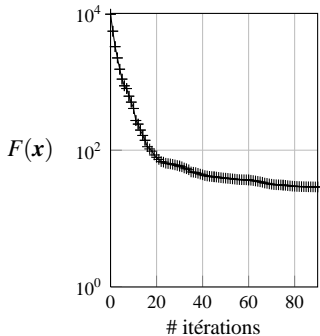
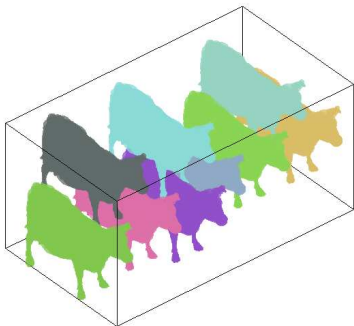
# Algorithme de séparation 3D

Contenant parallélépipédique et composants quelconques – Séparation en translation

Minimisation des **profondeurs de pénétration** entre chaque paire de sphères

9 composants – 27 variables – 512 sphères par composant

Itération 105 –  $F(\mathbf{x}) = 28.7879$



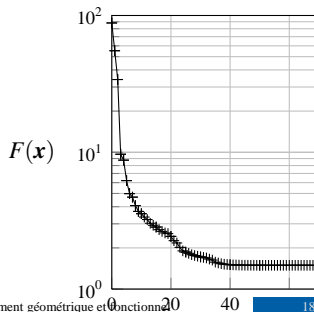
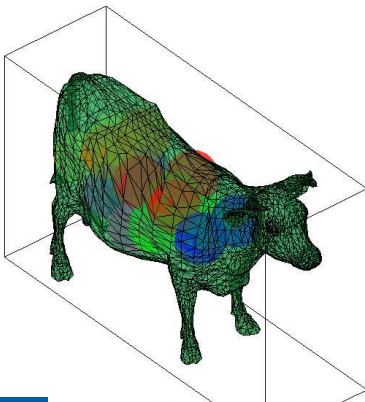
# Algorithme de séparation 3D

Contenant et composants quelconques – Séparation en translation

- ▶ Minimisation des **profondeurs de pénétration** entre chaque paire de sphères
- ▶ Minimisation des **profondeurs de pénétration approchées** entre les sphères et le contenant

7 composants – 21 variables – Contenant : 2500 sommets, 5000 triangles

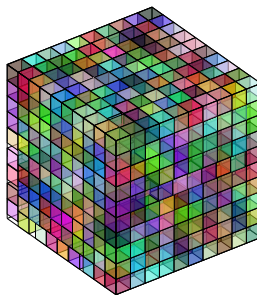
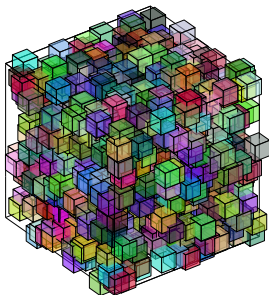
Itération 60 –  $F(x) = 1.49185$



# Algorithme de séparation

Résumé des différentes variantes de l'algorithme de séparation

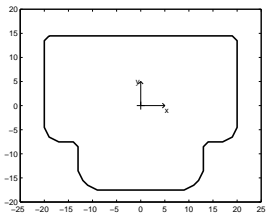
	Composants	Contenant	Type de séparation
2D	Rectangles	Rectangle	Min. du produit aire intersection profondeur de pénétration
	Cercles	Rectangle / Cercle	Min. pénétration
	Polygones à orient. discrètes	Polygone	Min. pénétration avec NFP/IFP
	Polygones à orient. continues	Polygone	Min. pénétration avec ass. cercles
3D	Parallélépipèdes	Parallélépipède	Min. du produit volume intersection profondeur de pénétration
	Sphères	Parallélépipède / Sphère	Min. pénétration
	Polyèdres	Parallélépipèdes	Min. pénétration avec ass. sphères
	Polyèdres	Polyèdre	Min. pénétration avec ass. sphères et nuage de points



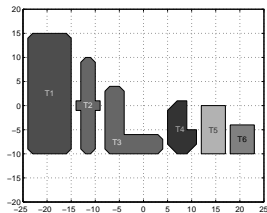


# Exemple 1 – Problème d'agencement bi-objectif

Problème de placement de **10 composants** dans un coffre :



(a)



(b)

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

Niveau de gris proportionnel à la densité des composants.

Compacité = 59.7%

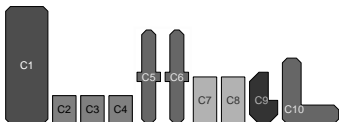
(c)

- ▶ **Méthode de placement** : Placement relaxé
- ▶ **Variables** : 20 variables réelles et 10 variables discrètes

# Exemple 1 – Objectifs

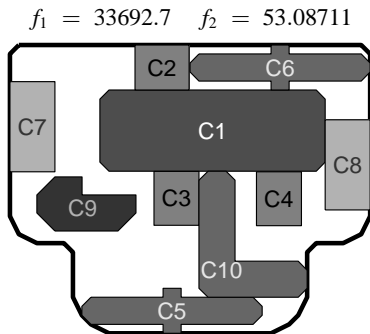
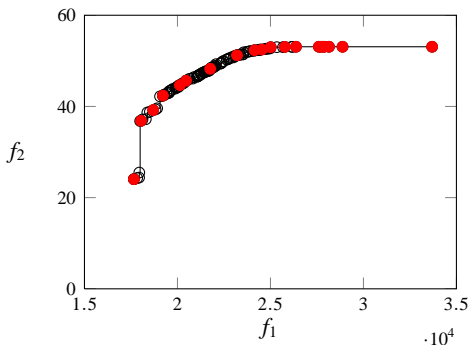
- ▶ Minimisation de l'inertie transversale  $I_{yy}$
- ▶ Maximisation des distances minimales entre les composants  $C_5$ ,  $C_6$  ainsi que  $C_7$  et  $C_8$

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



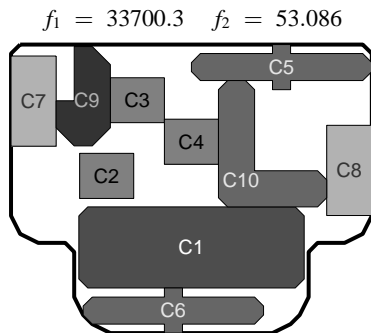
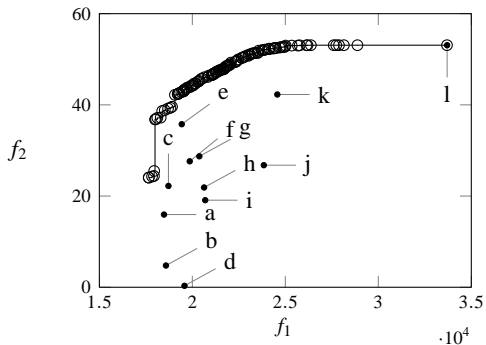
# Exemple 1 – Solutions du front de Pareto

- ▶  $\min f_1$ : Minimisation de l'inertie transversale  $I_{yy}$
- ▶  $\max f_2$ : Maximisation des distances minimales entre les composants  $C_5$ ,  $C_6$  ainsi que  $C_7$  et  $C_8$



## Exemple 1 – Solutions sous-optimales

- ▶  $\min f_1$ : Minimisation de l'inertie transversale  $I_{yy}$
- ▶  $\max f_2$ : Maximisation des distances minimales entre les composants  $C_5$ ,  $C_6$  ainsi que  $C_7$  et  $C_8$

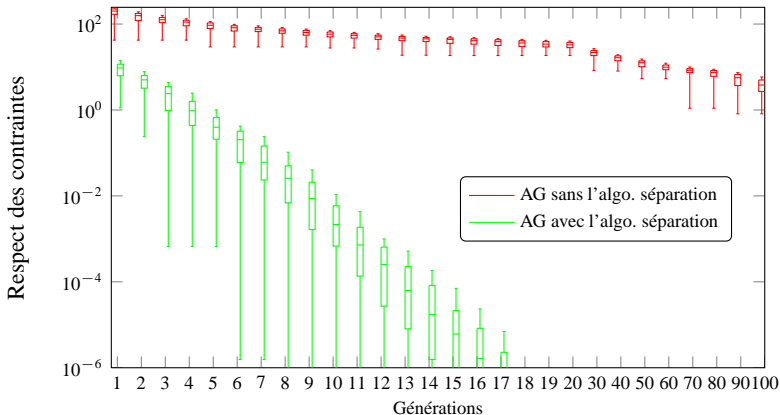


# Exemple 1 – Validation méthode

Efficacité de l'algorithme de séparation

Évolution du respect de la contrainte de placement pour

- ▶ l'algorithme génétique sans séparation
- ▶ l'algorithme génétique avec séparation

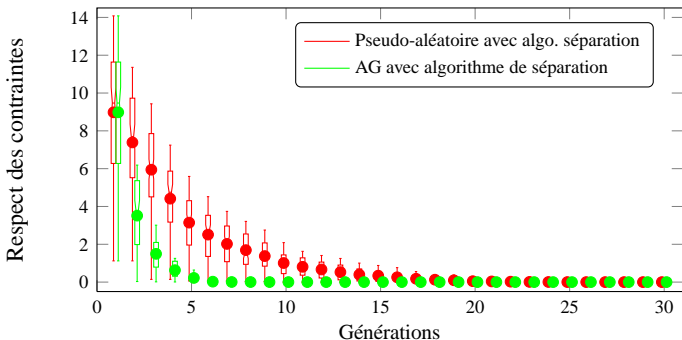


# Exemple 1 – Validation méthode

Efficacité des opérateurs génétiques

Évolution du respect de la contrainte de placement pour

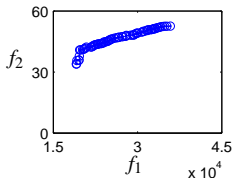
- ▶ un algorithme pseudo-aléatoire élitiste générationnel (perturbations aléatoires à chaque génération)
- ▶ l'algorithme génétique proposé (Croisement – Mutation – Échange)



# Exemple 1 – Influence de la population initiale

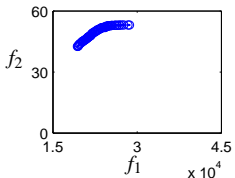
Génération : 200

$\theta_1$  Initialisation aléatoire



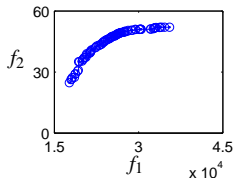
HVR = 91.94%

$\theta_2$  Initialisation aléatoire

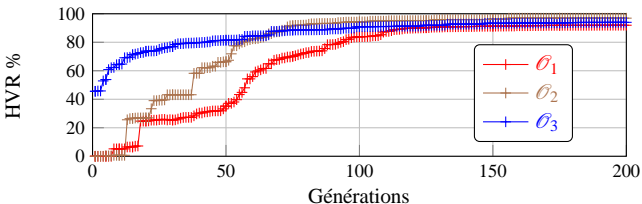


HVR = 96.90%

$\theta_3$  Initialisation *Bottom-Left*



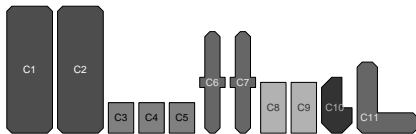
HVR = 93.98%



## Exemple 2 – Problème d'agencement bi-objectif

- ▶ Ajout d'un composant de type 1. Compacité : 59.7% → 79.5%.
- ▶ Objectifs :
  - ▶ Minimisation de la distance entre le CdG de l'assemblage et le CdG du coffre.
  - ▶ Minimisation de l'inertie  $I_{yy}$  au CdG géométrique du coffre.

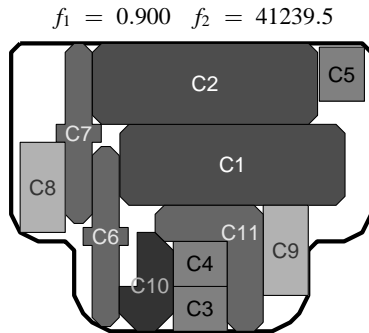
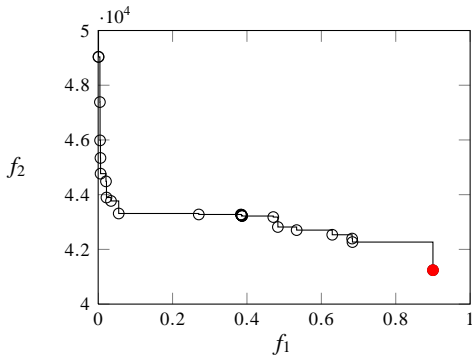
$$\mathcal{P} \left\{ \begin{array}{l} \min f_1(\mathbf{v}) = \sqrt{d_x^2 + d_y^2} \\ \min f_2(\mathbf{v}) = \sum_{i=1}^n \rho_i I_{i/yy} \end{array} \right\} \left\| \begin{array}{l} d_x = \frac{1}{\sum_{i=1}^n \rho_i A_i} \sum_{i=1}^n \rho_i A_i (x_i - x_G) \\ d_y = \frac{1}{\sum_{i=1}^n \rho_i A_i} \sum_{i=1}^n \rho_i A_i (y_i - y_G) \end{array} \right. \\ \text{s.c. les contraintes de placement soient satisfaites}$$





## Exemple 2 – Solutions du front de Pareto

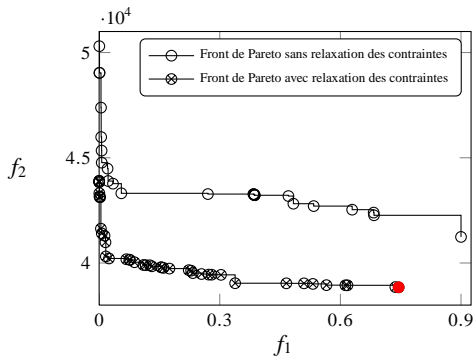
- ▶  $\min f_1$ : Minimisation de la distance entre le CdG de l'assemblage et le CdG du coffre.
- ▶  $\min f_2$ : Minimisation de l'inertie  $I_{yy}$  au CdG géométrique du coffre.



## Exemple 2 – Solutions du front de Pareto

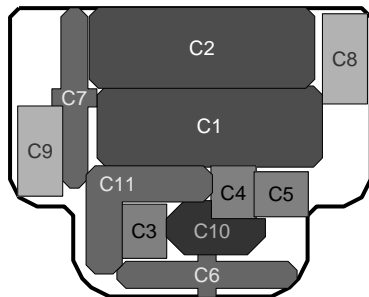
Relaxation de l'indice de violation de contraintes

- ▶  $\min f_1$ : Minimisation de la distance entre le CdG de l'assemblage et le CdG du coffre.
- ▶  $\min f_2$ : Minimisation de l'inertie  $I_{yy}$  au CdG géométrique du coffre.



$$f_1 = 0.74488 \quad f_2 = 38845.4$$

$$CV = 0.4796$$

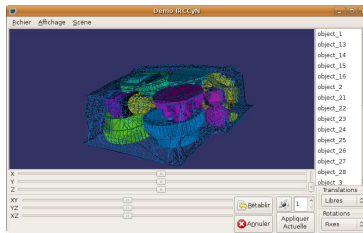
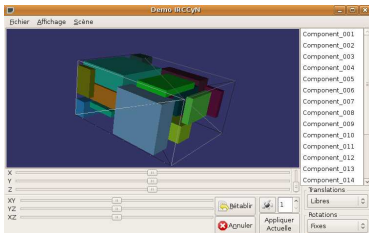


# Conclusion

- ▶ **Catégorisation** des problèmes de placement, mise en avant des *difficultés* de résolution.
- ▶ Présentation d'une méthode de placement s'adaptant à une **grande variété** de problèmes.
  - ▶ Possibilité de traiter réellement les problèmes **multi-objectifs**.
  - ▶ Méthode 2D donne de **bons résultats** même pour les problèmes à forte compacité.
  - ▶ Éléments pour la méthode 3D en cours de développement.
  - ▶ Permet au concepteur d'**interagir** avec les solutions proposées.
  - ▶ **Cadre de travail ouvert** permettant de nombreux développements futurs.

# Démonstrateur développé

- ▶ **Objectif** : Tester les différentes méthodes de résolution proposées.
  - ▶ Fonctionnalités graphiques
  - ▶ Fonctionnalités d'interaction : Déplacement des composants
  - ▶ Fonctionnalités de résolution : Appel algorithme génétique et de séparation



# Perspectives

- ▶ À court terme :
  - ▶ Utiliser un **placement relatif** des composants pour modéliser de nouvelles contraintes ;
  - ▶ **Intégrer de l'algorithme de séparation 3D** dans l'optimiseur global pour obtenir un solveur 3D ;
  - ▶ Développer la **version hiérarchique** de l'algorithme de séparation 3D.
- ▶ À moyen terme :
  - ▶ Développer l'**interaction avec le concepteur** ;
  - ▶ Intégrer l'optimisation d'agencement dans les **processus de conception** (Optimisation multidisciplinaire).

Merci de votre attention !

# Publications



Guillaume JACQUENOT, Fouad BENNIS, Jean-Jacques MAISONNEUVE et Philippe WENGER :  
2D Multi-Objective Placement Algorithm for Free-Form Components.

*In Proceedings of ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, numéro DETC2009-87603, San Diego, CA, USA, septembre 2009.



Guillaume JACQUENOT, Fouad BENNIS, Jean-Jacques MAISONNEUVE et Philippe WENGER :  
Méthode générique pour la résolution des problèmes de placement bidimensionnel.

*In AIP Primeca*, mars 2009.



Guillaume JACQUENOT, Fouad BENNIS, Jean-Jacques MAISONNEUVE et Philippe WENGER :  
Optimisation de placement dans les problèmes de conception.

*In Congrès Français de Mécanique 2009*, 2009.



Guillaume JACQUENOT, Fouad BENNIS, Jean-Jacques MAISONNEUVE et Philippe WENGER :  
Présentation d'une méthode de résolution pour les problèmes de placement 2D et 3D.

*In ROADEF*, février 2009.

# Bibliographie I



J. A. Bennell and X. Song.

“A Beam Search Implementation for the Irregular Shape Packing Problem”.  
*Journal of Heuristics*, 2007.



J. Cagan, R. Clark, P. Dastidar, S. Szykman, and P. Weisser.

“HVAC CAD Layout Tools : A Case Study Of University/Industry Collaboration”.  
1996.



K. Deb and S. Tiwari.

“Omni-Optimizer : A generic Evolutionary Algorithm for single and multi-objective optimization”.  
*European Journal of Operational Research*, Vol. 185, pp. 1062–1087, March 2008.



E. W. Dijkstra.

“A short introduction to the art of programming”.  
Tech. Rep. EWD-316, Technisch Hogeschool Eindhoven, 1971.



J. Egeblad.

*Placement Techniques for VLSI Layout Using Sequence-Pair Legalization*.  
Master’s thesis, Department of Computer Science, University of Copenhagen, July 2003.



T. Imamichi, M. Yagiura, and H. Nagamochi.

“An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem”.  
*Discrete Optimization*, Vol. 6, No. 4, pp. 345–361, Nov. 2009.



Y. Miao, G. M. Fadel, and V. B. Gantovnik.

“Vehicle configuration design with a packing genetic algorithm”.  
*International Journal of Heavy Vehicle Systems*, Vol. 15, pp. 433–448, Dec. 2008.

# Bibliographie II



D. Pisinger.

“Heuristics for the container loading problem”.  
*European Journal of Operational Research*, Vol. 141, pp. 382–392, 2002.



H.-F. Teng, S.-L. Sun, D. quan Liu, and Y.-Z. Li.

“Layout optimization for the objects located within a rotating vessel – a three-dimensional packing problem with behavioral constraints”.  
*Computers & Operations Research*, Vol. 28, pp. 521–535, May 2001.



S. Tiwari, G. M. Fadel, and P. Fenyés.

“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.



G. Wäscher, H. Haußner, and H. Schumann.

“An improved typology of cutting and packing problems”.  
*European Journal of Operational Research*, Vol. 183, pp. 1109–1130, Dec. 2007.



S. Yin.

*A Computational Framework for Automated Product Layout Synthesis Based on Extended Pattern Search Algorithm*.  
PhD thesis, Carnegie Mellon University, Pittsburgh, 2000.



B. Zhang, H.-F. Teng, and Y.-J. Shi.

“Layout optimization of satellite module using soft computing techniques”.  
*Applied Soft Computing*, Vol. 8, No. 1, pp. 507–521, 2008.