Proceedings of DETC99
1999 ASME Design Engineering Technical Conferences
September 12–15, 1999, Las Vegas, Nevada

DETC99/DAC-8575

CONFIGURATION DESIGN OPTIMIZATION METHOD

Pierre M. Grignon
Department of Research and Development,
Dassault Systemes,
Suresnes, France.
pierre_grignon@ds-fr.com

Dr. Georges M. Fadel
Mechanical Engineering Department
Clemson University
Clemson, SC 29634-0921
gfadel@ces.clemson.edu

KEYWORDS
Configuration, optimization, Pareto Genetic Algorithms, packing, multi-objective, design.

ABSTRACT
This work presents a method (CDOM) for optimizing multiple system level assembly characteristics of complex mechanical assemblies by placement of their components. It addresses any component shape (including non-convex, hollow, sharp edges) connected together by functional constraints. This method finds multiple solutions to the Engineering Configuration Design Problem (ECDP) and proposes a simple cooperation scheme with the engineer using a Genetic Algorithm working on population of sets instead of population of individual points. In this paper four areas are addressed:

- *defining the assembly components and their relations (ECDP),*
- *defining the Configuration Design Optimization Method (CDOM)*
- *propose two strategies to improve the CDOM performance*
- *show application of the CDOM to one academic and two engineering test cases*

INTRODUCTION
Packing problems have been among the most exciting human pastimes for centuries. Puzzles, Tangram, Pentominos and other Polyminos reflect this excitement through Human history. Nowadays, these problems find applications in industry. Loading ships, trucks and trains, designing plants, electronic boards, placing mechanical components, cutting stocks, scheduling, are some of the numerous problems that can be classified as packing optimization problems.

Among the geometrical packing problems, those involving spheres, squares and rectangles are the best known because they are the easiest to formulate and thus are the most studied. Unfortunately, even these simple cases are known to be difficult problems in the mathematical sense. So far, nobody has yet produced a global formal solution for them and even some mathematicians believe that this event will never happen (Lewis and Papadimitriou, 1978). Today's alternative is to accept a nearly optimal solution using computers and heuristics. These heuristics are case dependent and those guarantying good solutions are subject to combinatorial explosion. Moreover, engineering packing problems are more difficult to solve than basic packing problems since mechanical components are usually freeform and 3 dimensional. Common approaches involve different techniques such as dynamic programming, tree search or very specialized heuristics like finding optimal sub-configuration by assembling the objects in groups of two or three. In order to tackle the inherent complexity of simple packing problems, some non-deterministic methods are beginning to emerge. They involve the use of Simulated Annealing and Genetic Algorithms. Corcoran and Wainwright (1992) programmed a Genetic Algorithm to solve a bin packing problem using up to 500 packages and thus broke all the limits ever considered for such a problem. Wodziak (1994) and Grignon, Wodziak and Fadel (1996) used a mixed combinatorial and continuous formulation to address the truck-packing problem but their methods are dedicated to square like object shapes and thus are difficult to extend to engineering problems in which the objects are free form. Szykman and Cagan (1995, 1996) used a simulated annealing method to solve component layout problems. These non-deterministic approaches were enhanced by either local search (Hart, 1994), by specialized search heuristics (Yin and Cagan, 1998) or even combined with each other like in (Moscato, 1992). Eventually, Sachdev et al. (1998) proposed a framework of intelligent agents to integrate optimization techniques such as GA and SA to address layout problems.

Free form packing refers to the packing of objects and containers having a non-regular shape. This case was not studied until recently because of several reasons. First, the complexity of the problem is such that even the best algorithms combined with the best computers could not give a satisfactory answer in a reasonable amount of time. Second, the methods of description of free form objects were not flexible enough to be handled conveniently by programmers. Third, there was no theoretical interest in these cases. Kim and Gossard (1991), Szykman and Cagan (1995, 1996), Yin and Cagan (1998) and Lomangino (1994) used a CAD software to represent the complex shapes and coupled such a program to an optimization software to find better configurations. In addition, they developed their own methods to constrain objects to assume given positions due to their functionality. This case is the most general of the Configuration Design Problems (CDPs). These problems are usually constrained, involve multiple criteria and mixed discrete/continuous variables. The shapes of the objects involved in these CDPs are free and the characteristics of their objectives and constraints can be linear, quadratic, non-linear, multi modal, continuous or discrete.

Genetic Algorithms present both characteristics of being able to accept mixed discrete/continuous variables and being able to look for global optima of non-linear discrete/continuous non-linear objective functions. They have the additional advantage of working on populations of points, which eases the search for several solutions in the case of multi-objective optimization. A more complete literature review of the use of GA and Multi objective optimization can be found in Schaffer and Grefenstette (1985); Venugopal and Narendra (1992); Grignon and Fadel (1996, 1997).

Another aspect of engineering design optimization is the possibility of interaction between the search and the engineer. Sobol (1990) gives a method, especially suited for Computer Aided Design. The method consists of 3 steps and only uses optimization for refining the final decision of the engineer. The first step consists of computing the values of the objectives at some selected points belonging to an $Lp\tau$ sequence because these points constitute a much better sampling of a hypercube than the corners of the rectangular lattice (for a definition of these sequences of points see (Sobol, 1967, 1990). The computed values corresponding to feasible points are put into tables from the best to the worst. During step 2, a decision-maker (the engineer) sets some acceptable bounds for the objectives. Step 3 consists in checking if some points of the tables satisfy the bounds on the objectives. If the answer is positive then the search is stopped. Otherwise the engineer is asked to relax the bounds. The main difference between this method and the traditional approaches is the translation of the optimization problem into a constraint satisfaction one. The objectives are not optimized but reach at least a value driven interactively by the user. This seems to be a method well adapted to the engineering fields in which the magic black box, which gives the solution to the problem just by pushing the button, is not well accepted. However, when the number of variables increases, this method may be computationally expensive.

In many other methods, the means for incorporating knowledge in the solution process is either non-existent, relies on the choice of variables (case of the GA), or is incorporated in the heuristic (like in the heuristic used to solve scheduling or rectangular packing problems). There is no possibility to add knowledge during the run. Engineers usually prefer to be in control of the search process and be informed of the progress.

ENGINEERING CONFIGURATION DESIGN PROBLEM

Mathematical Formulation

In mechanics, as well as in CAD, there is a major difference between the object and the system. The object is an atomic solid that cannot be taken apart without being broken; whereas the system is an aggregation of objects that can move with respect to each other.

It is proposed to maintain this difference in the mathematical models involved in the CDPs description (as opposed to other approaches that tend to describe all the relations defining the geometry of the objects and the characteristics of the assembly at the same time). Thus, on one hand, a description of the objects as indivisible whole is provided, and on the other hand, a description for the system as a relative or global positioning of the objects is used. The characteristics of interest in this study are system level characteristics like volume, inertia, and maintainability.

Given :

- *A global Cartesian coordinate system.*
- *A set of N objects defined by their shape (S) material and positioned in space by 6 variables i.e. the 3 Cartesian coordinates of a particular point (x, y, z) and the 3 angles defining their orientation (a,b,c) with respect to a local Cartesian coordinate system.*
- *A set of equalities, usually called functional constraints* $H_k[\mathbf{x}] = 0$*, positioning the objects with respect to a reference coordinate system,*
- *A set of inequalities* ($G_j[\mathbf{x}] < 0$) *translating the notion of interior and exterior of the objects.*
- *M objective functions of the 6N variables.*

Find :

A set of values for **x** (a vector composed of all the positions of all the objects of the system) optimizing the vector objective function **F** and satisfying the equality and inequality constraints.

MOP definition:

Given a function $\mathbf{F}(\mathbf{x})$ where $\mathbf{x} = (x_1, x_2, .. , x_n)$ is a vector of a domain D of R^n and $\mathbf{F}(\mathbf{x})$ is also a vector $\mathbf{F}(\mathbf{x})=(f_1(\mathbf{x}), f_2(\mathbf{x}), .. , f_m(\mathbf{x}))$, find one $\mathbf{x}$ such that $\mathbf{x}$ is non-inferior.

A solution $\mathbf{x_1}$ is said to be a non-inferior solution of the MOP if there exists no other vector $\mathbf{x_2}$ such that $f_i[\mathbf{x_1}] <= f_i[\mathbf{x_2}]$ for all i with a strict inequality for at least one value of i.

This problem can be formulated as an unconstrained global optimization problem in which the objectives and the constraints are combined using a penalty function in order to build a single objective function.

Objects, Variables, Constraints and Objectives

Variables

All the variables involved in the CDPs presented so far were the positions of the components with respect to an absolute Cartesian coordinate system. In order to respect the functional constraints involved in many mechanical CDPs it was decided to replace the absolute positions variables (x, y, z, α,β,χ) by a translation vector $\mathbf{T}$(tx, ty, tz) and a rotation vector $\mathbf{R}$(rx, ry, rz) with respect to a coordinate system attached to another component. Both combined give a relative displacement matrix, $\{\mathbf{T},\mathbf{R}\}_{ref}$ with respect to a reference coordinate system (ref), easy to represent (Figure 1) and which allows to position exactly a component in space.

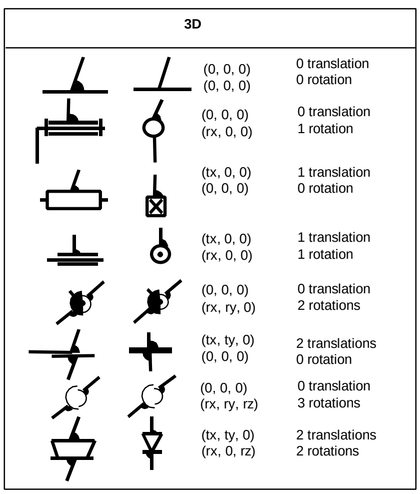


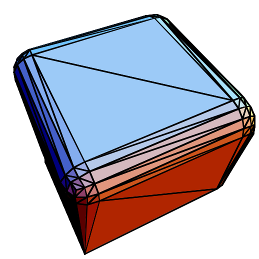
Figure 1. Correspondence between Mechanical Functional Links and Displacement Vectors (ISO 3956)(Chevalier, 1986).

This way of placing components offers the same flexibility as the absolute coordinates, it also gives the possibility to implicitly satisfy many mechanical functional links that, otherwise, would have been satisfied by equation solving. The choice of the movements attributed to a component with respect to another one is left to the engineer. Thus, he or she is in charge of defining displacements respecting the mechanical functional links between the components of the assembly.

Objects Shapes

In spite of the nearly 20 years of existence of CAD software, it is since very recently that the objects used in configuration optimization upgraded from simple shapes like cubes, blocks, spheres, circles, and squares to 'free-form objects'.

Lomangino (1994) and Szykman and Cagan (1994b) introduced the use of CAD to represent the more realistic objects necessary in engineering. Even if more cumbersome, this approach allows the engineers to take advantage of the piecewise polynomial or rational models implemented in CAD to build free-form geometric objects. CAD software is able to generate several representations of the objects in order to save, retrieve and exchange data with other software. Among the standard formats used to exchange information about the shape of an object, one of the more common and easy to use is the tessellated description of its surface. This type of description is used for rapid prototyping models, surface finite element models and for visualization models. Hence, in this work, each object is described by its skin paved with the triangles obtained by exporting an STL file from a CAD software (Figure 2).



```
facet normal 1.00 -0.00 -0.00
      outer loop
         vertex 23.62 39.37 -39.37
         vertex 23.62 55.11 -39.37
         vertex 23.62 55.11 -23.62
      endloop
endfacet
```

Figure 2. The STL file format and the tessellated object.

This format provides a good balance between the accuracy of a solid description and the computational speed of various functions involving objects, such as volume, interference, and clearance (O'Rourke 1993) calculation.

Configuration Optimization Problem Objectives Functions

A mechanical system's **compactness, balance** and **maintainability** were chosen as objective functions for testing the CDOM. Their mathematical representation relies either on analytical expressions or on a numerical integration over the meshed solids.

Compactness

In many cases, system volume was associated with the volume of a component's bounding box. This choice has several drawbacks, among which: first, the volume changes when the coordinate system changes, and second, it leads to results that are inaccurate. A second way to evaluate the volume of a set of objects is to measure the volume of its convex hull i.e. the volume of rubber bag one can wrap around the objects. The volume measured is invariant when the coordinate system changes however it is computationally expensive to evaluate. Moreover using the volume of the convex hull can lead to unexpected, however valid results, which have nothing to do with an admissible mechanical solution (Stewart, 1993). This points out that the notion of compactness in mechanics is different from that of minimal volume, and thus, an inertia matrix ([I]) norm shown below is a better measure.

$$[\mathbf{I}] = \begin{pmatrix} \int (y^2 + z^2)\,dm & -\int (xy)\,dm & -\int (xz)\,dm \\ -\int (xy)\,dm & \int (x^2 + z^2)\,dm & -\int (yz)\,dm \\ -\int (xz)\,dm & -\int (yz)\,dm & \int (x^2 + y^2)\,dm \end{pmatrix}$$

where the integration is done over a solid object.

The matrix is computed for each object and the whole system inertia matrix can be calculated by adding the individual matrices as long as they are expressed with respect to the same coordinate system. Once calculated, the system inertia matrix norm is used as an objective function.

$$\mathrm{Norm}[\,[\mathrm{I}]\,] = \textstyle\sum_{i=1..3} [\, \sum_{j=1..3} [\, I_{ij} \,]^2 \,].$$

Statical and Dynamical Balanced Loading

The static balance of a system is obtained by putting its center of gravity at a target location. The center of gravity of a system is usually easy to calculate once given the center of gravity of each of its components.

Statically balancing a system is sometimes not sufficient. A dynamic balance must be reached especially for objects that can move (cars, boats, planes...). The inertia matrix presents the interesting property of being a measure of the dynamic stability of a system. In order to be stable in dynamic situations as well as in static situations the principal axes of a system should respect some position. For example, the principal axes of the heaviest components of a car (under the hood components) must stay aligned with the longitudinal axis of the vehicle. Obtaining a diagonal matrix with respect to target axes is the goal for dynamically balanced loading.

Maintainability

The maintainability of a system is decomposed into two different tasks that must be performed and are represented by two objective functions.

Accessibility is the property of an object or part of this object to be seen and reached with the hand. This objective is implemented by a ray-tracing algorithm in which the ray is an object simulating the volume necessary to access the component (Figure 3).

The ease of removal of a component is defined by the amount of mechanical work needed to remove it from the system (Figure 4). This mechanical work is proportional to the weight of the component. If some other components must be removed to remove a given component, the additional work is taken into account. The system maintainability is the sum of the accessibility of all the relevant components and of their removability.

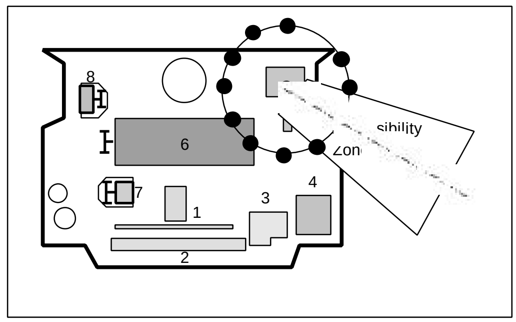


Figure 3. Accessibility.

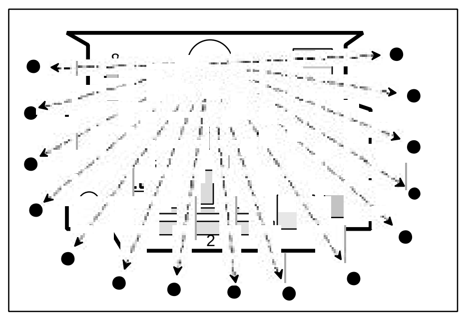


Figure 4. Removal path.

Constraints

Mechanical Functional Constraints

The mechanical functional constraints, usually translated by geometric constraints (angle, distance, coincidence, …) are taken into account by the displacement matrices, the only constraint remaining is the interference between assembly components.

Interference Constraints

Checking that there is no interference between the N objects of the mechanical system is the last constraint considered in the ECDP formulation. The satisfaction of this constraint has priority on the mechanical constraints since it guarantees the possibility of building the system.

THE CONFIGURATION DESIGN OPTIMIZATION METHOD

Populations of Sets and Genetic Algorithms

The Configuration Design Optimization Method (CDOM) was first designed to cope with the non-linearity of packing problems with two objective functions (center of gravity and volume, Grignon, Wodziak and Fadel, 1996).

When considering additional objectives, it was clear that the GA, which allows the gathering of several solutions in a single run, had another advantage with respect to other global optimization techniques. However, the different techniques proposed in the GA literature were not sufficient to cope with a continuous or discontinuous set of solutions like a Pareto Set.

Hence, instead of using a population of points, it was decided to remove the difficulty of forcing points to stay away from each others by using populations of sets of points (Grignon and Fadel, 1997). Such formulation brings different problems. By making the size of the genome bigger, the limitations of the GA were reached quicker and the rate of convergence was degraded. In order to overcome these new problems of convergence and to exploit the fact that the GA had a very good performance only during the first few hundreds generations and then levels off, the algorithm was made iterative (Grignon and Fadel, 1999)(Figure 5).

At first glance it seemed that this choice was only making the method slower for getting back a decent rate of convergence and acceptable results. At a second glance the iterations were opening the search process to an eventual interaction with the user. Thus, the experience of engineers, which is a level of knowledge that an automated method has few chances to reach, is now available for the CDOM. This was made possible by changing the bounds of each variable after each run of the GA in order to determine the new domain of interest for the next run. These new bounds are determined by the extreme points of the best approximation of the Pareto set discovered so far by the method and define the variable ranges for the next iteration. These ranges are increased by 20% (of the size of the Pareto set along each variable axis) in order to allow the CDOM to expand the Pareto set.

The cycles are repeated until the global objective function cannot be improved by more than 0.5% during two consecutive runs.

In order to fit the need of the GA for a single function value attributed to each point, a scalarization based on two components is proposed. First, a value is calculated in order to differentiate Pareto sets from other sets of points. This

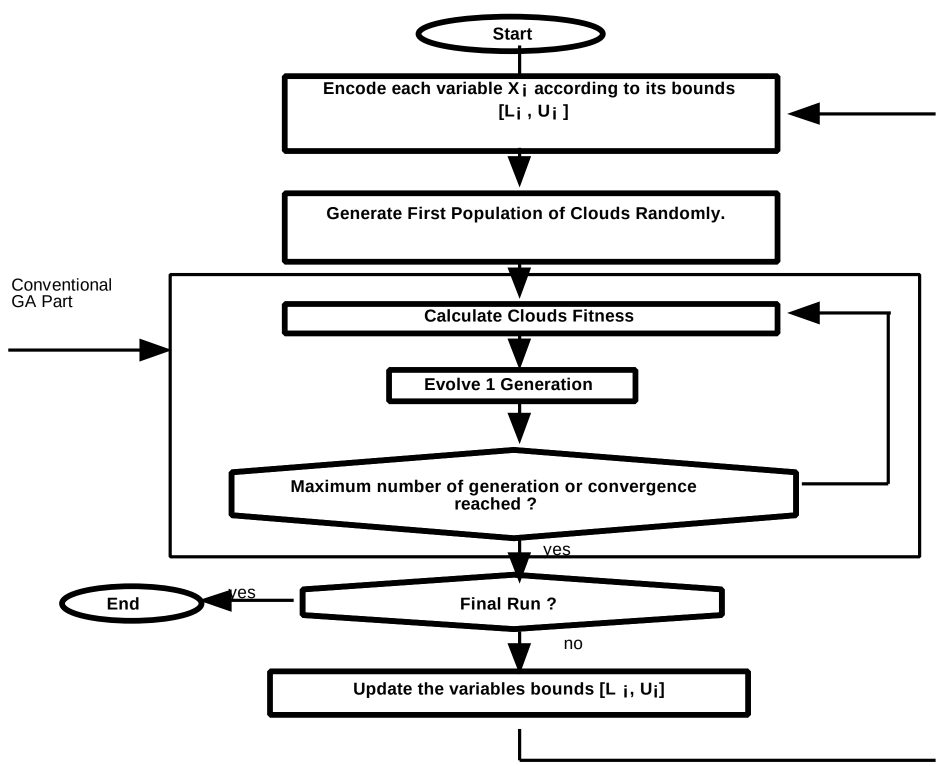


Figure 5. CDOM FlowChart

scalarization attributes better values to relative non-inferior points. Second, this scalar value is modified in order to take into account the spreading of the points.

The morphology is measured by giving a rank based on the notion of non-inferiority described in Grignon and Fadel (1999).

$$\text{Grade}_{cloud} = \left(\sum_{i=1..N} [\, \mathbf{rank}[P_i]\,] \right) / (N + M)^2 + \text{penalty}_{sp}.$$

Where **Rank**[P_i] is N*M minus the number of points dominating P_i. N is the number of points in the cloud being evaluated and M is the number of points in a reference Pareto set made of the best points discovered.

The spreading is introduced by a penalty that always gives priority to the morphology of the cloud since its value is chosen such that it remains between 0 and 1.

$$\text{penalty}_{sp} = \text{Atan}\,[\,\text{Max}[1/\text{Distance}[P_i, P_j]\,]\,/\,(\Pi/2)\ i=1..N,\ j=1..N.$$

This strategy guarantees a convergence toward the real extreme points (TEPts) which are unknown and which correspond to the solutions of the single objective CDPs involving each objective in turn.

Since overlapping components is a strong impediment to the building of any assembly the corresponding constraint value is removed from all the single objective functions.

$$Pi'[\,F1 + \text{penalty}_{int}, .., Fn + \text{penalty}_{int}]$$

Thus an unfeasible assembly cannot be a Pareto solution. The final objective function (Grade_{cloud}) is calculated using the new points Pi' as cloud members.

CDOM variations

Initial experiments using the CDOM showed that low performance is driven by the size of the feasible area (which is a direct consequence of the number of the objects and of their relative size with respect to the system).

Hence, the initial algorithm (R) (Figure 5) which consisted in restricting the bounds of each variable run after run, is transformed into 2 algorithms, whose performances are compared. First, simulating the behavior of a human being, the CDOM is allowed to expand the size of the search region by 33% in all directions in case of a first failure to improve the objective function. A second failure stops the CDOM. Second, a zero order local search method is used to quickly retrieve the feasible domain, if possible, when an unfeasible configuration is generated. However this approach can be used only if the calculation cost of the objective function is moderate since local numerical optimization methods require a high number of functions evaluations (especially if gradients or Hessian matrices are involved). A second impediment for the use of this method comes from the fact that the Engineering CDPs are multicriteria problems. Applying the local search to each objective in turn leads to three different points in the variable space, and applying the Local Search to the Ranking objective function (Grade_{cloud}) is impossible since it is discrete. Temporarily replacing the objective function by the penalty alone in the unfeasible region seems to solve all these drawbacks.

Hence in the following three strategies are compared:

- *Ranges Restriction (R), which is the algorithm used as reference and already tested on Multicriteria non-convex optimization problems by Grignon and Fadel (1997).*
- *Local Search (LS), relies on a zero order Local Search to retrieve feasible solutions from unfeasible configurations.*
- *Range Relaxation (RLX), which implements the idea of enlarging the search area whenever no solution is found within a region of the variable space.*

TEST CASES AND ANALYSIS OF THE RESULTS

Cubes Configuration Design Problems

Problem statement

N cubes must be placed into a container such that the 3 objectives functions i.e. compactness, position of center of gravity and maintainability are optimized (Figure 6). The variables are the cubes translations (tx, ty, tz).

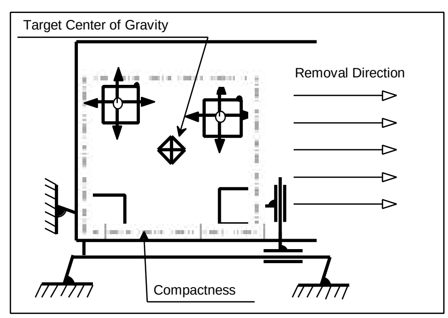


Figure 6. Objectives of the cube CDPs

In order to remove many configurations obtained by translation of the entire configuration as a block, one cube is fixed at the origin in order to get a non singular Pareto Set (otherwise the maximum compactness can be reached at the same time as the coincidence of the Center of gravity which makes the Pareto Set degenerate into a single point). All the cubes have the same weight (10) and volume (20x20x20).

The experiment is repeated with an increasing number of cubes and with 2D and 3D containers. The size of the containers is calculated such that the free-space remaining after all the objects are placed is the same (Table I).

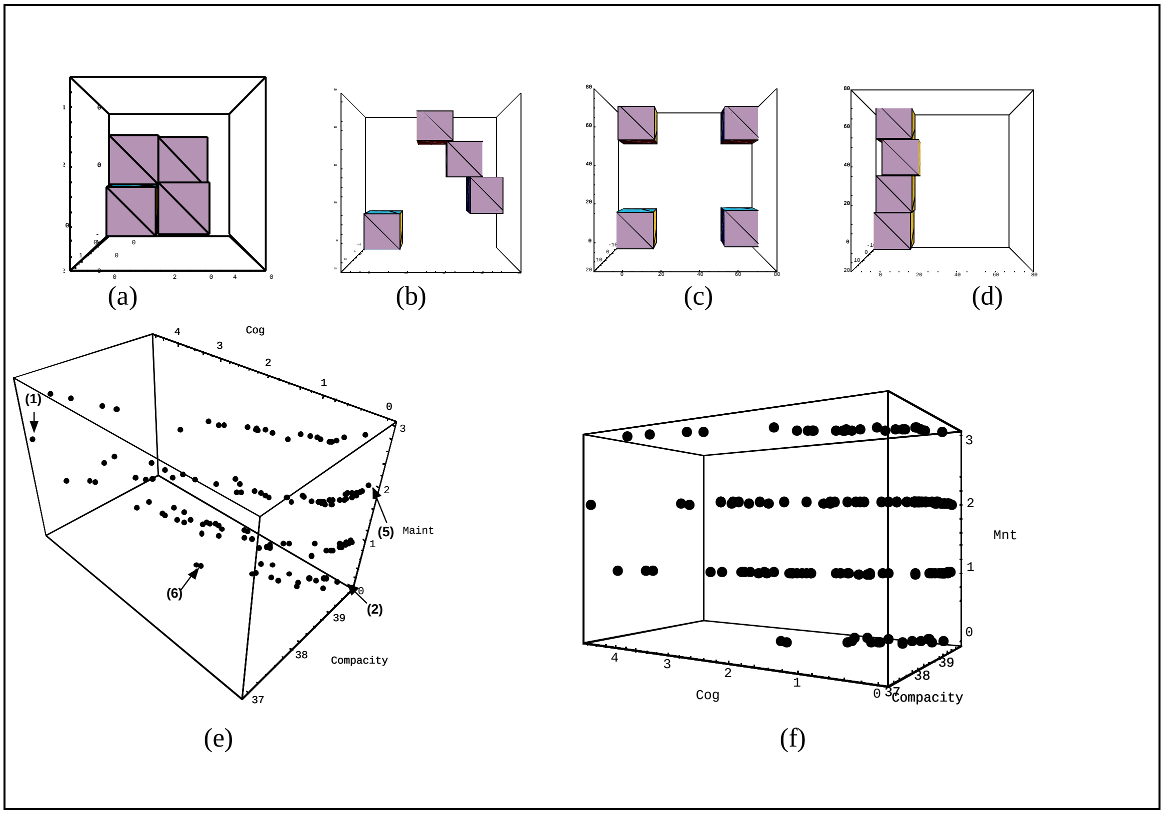


Figure 7. Example of final set distribution and configurations for a 4 cube 2D CDP. Each extreme point of the final set (e and f) in the objective space corresponds to an extreme configuration (a, b, c and d respectively, 1, 2, 5 and 6 on Figure e) in 3D space. Maintainability (Maint) is indicated on the vertical axis. Distance from the target center of gravity (Cog) and Compacity are indicated on the horizontal plane

Table I. Initial data for the 2D and 3D cube CDP. L2D and L3D are the respective sides of the square and cubic containers in which the cubes move.

# cubes	2	3	4	5	6	7	8	9
2D containers	49	60	69	77	85	92	98	104
3D containers	--	--	42	--	50	--	56	--

RESULTS PRESENTATION AND ANALYSIS

The final result that is of interest for the engineer is the Pareto set in the objective space and its corresponding values in the variable space each of which reflects a configuration in 3D space (Figure 7). Extreme points of this final set represent "extreme" configurations corresponding to the solutions of the Single Objective Optimization problems involving each objective alone.

Summary of the Cubes Test Cases Results

The criteria used to investigate the process of Configuration Optimization were classified into two main categories. First the criteria regarding the solution set and second the criteria related to the methods.

The Pareto set quality criteria are first, the distances from the 3 target extreme points (TEPs) corresponding to the solutions of the single objective optimization problems; second, the spreading of the points and third, their distribution (Grignon and Fadel 1999). The method criteria are speed measured in terms of function evaluations (objective and penalty combined) and repeatability measured as standard deviation of the previous criteria when the same test is repeated 20 times (Grignon 1999).

The study of the results, taking into account the TEPs criteria and the speed measures, allows to build a classification of the methods showing that the Iterative Genetic Algorithm combined with a local search (LS) is the best candidate for the CDOM (Table II). This algorithm is recommended whenever the additional computational cost of the LS routine is tolerable. For those who cannot afford this additional burden, the Iterative GA with Relaxation is a better choice.

Table II. Final Classification of the methods.

	R	RLX	LS
TEPs	19	9	9
SPEED	7	12	9*
Total	26	21	18*

Note: '*' indicates that the rank was calculated by taking into account the additional cost of the local search.

Engineering Applications

Two engineering configuration problems were submitted to the Iterative Genetic Algorithm with Range Relaxation (RLX). Both systems consist of one or several static components and multiple moving components that are placed in 3D space with respect to either the static objects or with respect to moving components. The Pareto sets of the 3 objective CDPs found by the method are presented as well as some corresponding configurations.

Aerospace Components

Problem Statement

A satellite made of a base frame, an external hull, and internal electronic and mechanical components (Figure 8), must be designed. All the components must fit inside the hull and the inertia of the whole system must be minimal for energy savings and stability considerations. In addition to this first objective, the center of gravity of the satellite must be put as close as possible to a target point located along the axis of the base frame, and the accessibility of the components must be maximized in order to ease construction and maintenance during an eventual repair intervention while on orbit. The components can be accessed by two opposite paths along the diameter of the satellite. The accessibility value is the number of components intersecting the access volume.

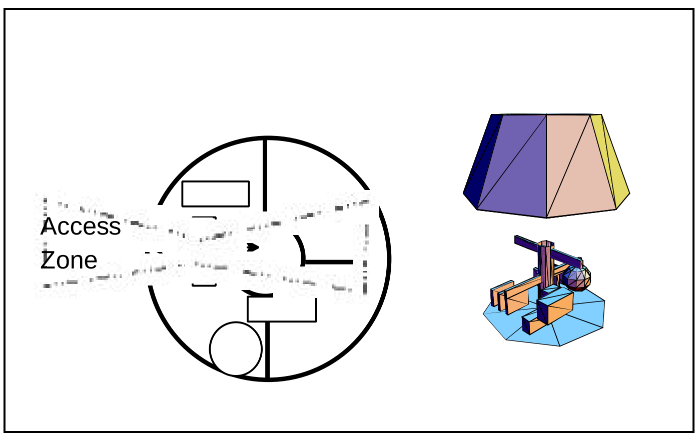


Figure 8. The satellite configuration design problem.

Objects and Variables

The relative degrees of freedom of the components with respect to each others are summarized in table III.

Table III. Satellite CDP components degrees of freedom.

Component	Degrees of Freedom	Reference Component
1 (hull)	{0, 0, 0}	Reference
2 (frame)	{0, 0, 0}	1
3 (tank)	{tx, ty, tz}	1
4 (electronic)	{tx, ty, 0}	1
5 (measuring eq.)	{tx, ty, tz}	1
6 (measuring eq.)	{0, ty, tz}	5
7 (electronic)	{0, ty, 0}	4

The best values of the Single Objective Optimization problems are reported in table IV.

Table IV. Results of the Single Objective Optimization Problems for the satellite CDP.

Objective	Solution	Speed	Penalty
Compactness	27	5950	4230
Distance to target Cog	0	2899	1432
Accessibility	0	2387	935

The multicriteria version of the satellite ECDP (Engineering Configuration Design Problem) is submitted to the RLX method. Its results are summarized in Table V. The number of points discovered by the CDOM is low (Figure 9) but corresponds to very different solutions (Figure 10). It was conjectured that the small feasible area is the cause of this problem. An additional clue corroborating this hypothesis is the high number of penalty function evaluations compared to the number of function evaluations.

Table V. Results of the multi objective optimization problem for the satellite CDP.

Objective	Extreme Solutions	Functions	Penalties
Compactness	28.1	23920	148945
Distance to target Cog	2.3 mm	Same	Same
Accessibility	0	Same	Same

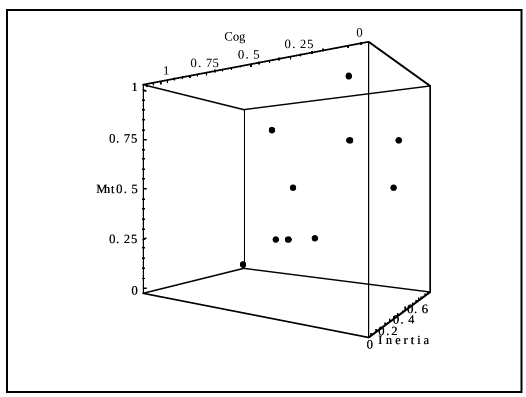


Figure 9. Three - objective Pareto set provided by the CDOM

However, the cloud of points seems relatively evenly distributed. This was already observed on small solution sets in the test cases. It seems that the cloud is also well spread along the compactness and accessibility axes, providing a range of compactness between 28.1 and 29.5 and an accessibility between 0 and 4. However, a wider range for the center of gravity was expected: it varies between 2.3 mm and 23 mm (knowing that the satellite is 1 meter in radius).

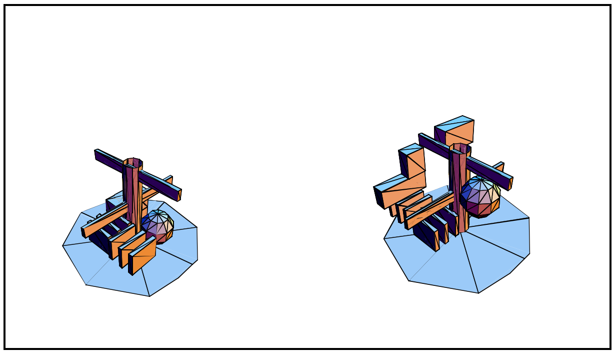


Figure 10. Two extreme configurations. At left, the maximum compactness configuration discovered by the CDOM. At right, the maximum maintainability configuration.

Car Engine

For mechanical systems such as a car engine, the center of gravity must be as low as possible while the components must be accessible as easily as possible. This later goal is contradictory with a minimum volume engine. Contrary to the satellite ECDP, the car engine problem benefits from a long history and many data that can be used for comparison. Although creating new design by patching old ones is not necessarily optimal, it was shown that this is a common way of designing, especially in the car industry (National Research Council 1991, Womack 1991). This is why, using the CDOM to generate new alternate designs is interesting.

Problem Statement

In the car engine example (Figures 11, 12), the problem consists of placing the main components of the car under the hood (motor, radiator, fan, fan engine, battery, power steering pump and tank, alternator, air-filter, and master cylinder) while trying to optimize the compactness, the position of the center of gravity and the maintainability of the system.

Is this case, accessibility is defined as it was for satellites, i.e. a static component simulates the volume necessary to access a given element. Then the value of the accessibility is calculated as the number of components intersecting this volume.

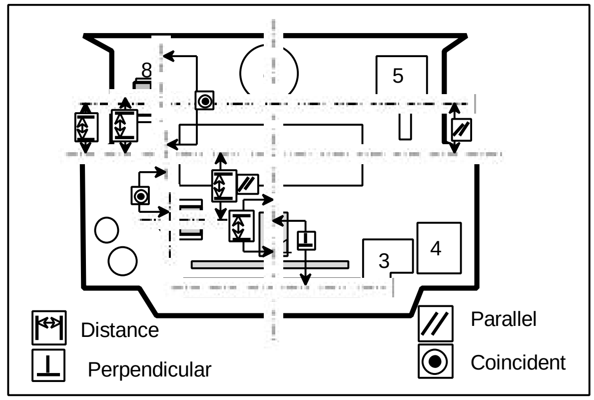


Figure 11. The components are placed using a global coordinate system and geometric constraints.

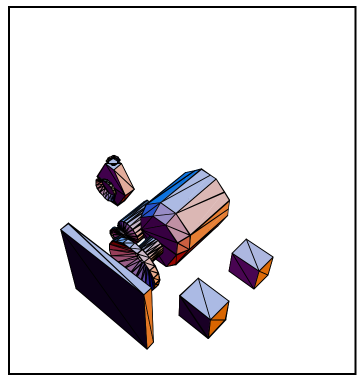


Figure 12. A simplified car engine submitted to the CDOM.

Objects and Variables

Seven components are selected for placement under the hood of a car. The motor block is used as reference for all the static components and for the axis of the car. It is purposefully put on the longitudinal axis (x-axis). The y-axis is horizontal and perpendicular to the longitudinal axis and the z-axis is vertical. The fan can move along the x and z axes with respect to the motor block. The ranges chosen for its translations constrain its position to be in front of the motor block. This constitutes the first example of mechanical constraint satisfaction using relative placement. If absolute placement had been chosen, an additional inequality constraint ($y_{fan} > y_{motor}$) would have been necessary. The radiator is placed with respect to the fan. Its only degree of freedom allows a transversal displacement. The alternator and the power steering pump are placed relatively to the motor block such that the pulley axes are always parallel to those of the engine. They also must remain in the same plane. These constraints are additional mechanical constraints that would have made the variable space more complex if absolute variables had been used. Finally, the battery and water tank can be put anywhere. The degrees of freedom of each component are summarized in Table VI.

Table VI. Car engine CDP components' degrees of freedom.

Component	Degrees of Freedom	Reference Component
1 (motor block)	{0, 0, 0}	Reference
2 (radiator)	{tx, 0, 0}	3
3 (fan)	{tx, 0, tz}	1
4 (radiator)	{tx, 0, tz}	1
5 (power steering)	{tx, 0, tz}	1
6 (battery)	{tx, ty, tz}	1
7 (water tank)	{tx, ty, tz}	1

Results and Discussion

The same procedure as the one used for solving the SOPs on the satellite CDP is used for this real case. The best values of each objective function are reported in Table VII.

Table VII. Results of the single objective optimization problems for the car engine CDP.

Objective	Best Solutions	Speed	Penalty
Compactness	41.15	6561	4716
Cog	17mm	5398	1022
Accessibility	3	3911	967

The configurations corresponding to the best compactness and best accessibility are displayed on Figures 13 and 14. In the best compactness configuration, the power steering pump and the alternator come slightly under the engine block in order to niche on both sides of the V shape. The battery and the water tank tend to have the same behavior but have too much height and thus cannot slide under the motor. The fan and the radiator are also slightly shifted on the right in order to counter-balance the effect of the alternator and battery on the left. In the maximum accessibility solution all the components are placed as far as possible from each others and as high as possible, which is the exact opposite of what is happening for the compactness objective.

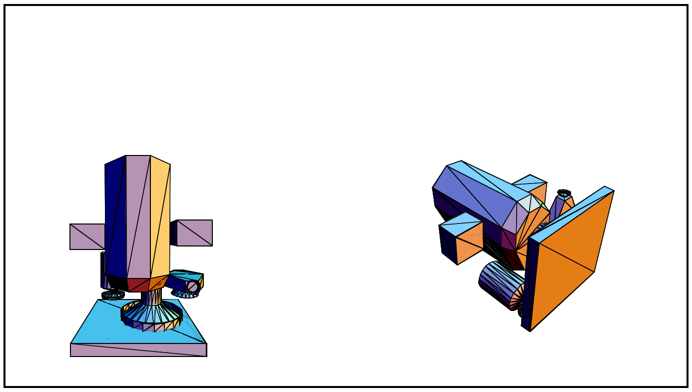


Figure 13. Example of maximum compactness.

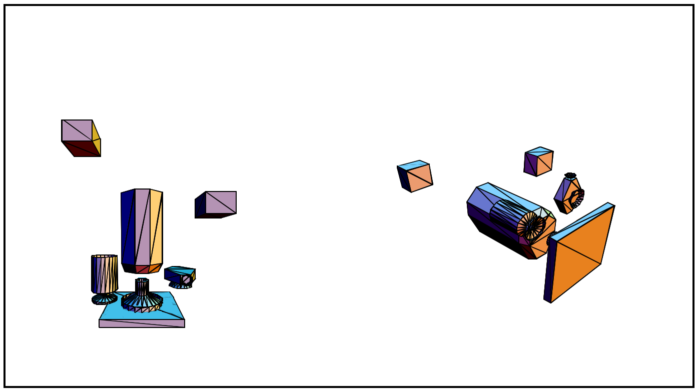


Figure 14. Example of maximum accessibility.

Many more Pareto design configurations are discovered in this case. The true reason for the discovery of many points is the size of the Pareto set. It was shown with the single optimization performed that the car engine presents extreme solutions that are, in variable space, very different and thus increase the chance of having a large Pareto set. Moreover, its feasible area is also larger than for that of the satellite CDP.

As was expected, the number of penalty function evaluation was low due to the large feasible area (Table VIII).

Table VIII. Results of the multi objective optimization problem for the car engine CDP.

Objective	Extreme Solutions	Functions	Penalties
Compactness	42.5	48850	9563
Cog	10 mm	same	Same
Accessibility	3	same	Same

The extreme compactness values oscillates between 42.5 and 43.5 (i.e. the solution gives the choice between solutions that can be more than twice less compact). The center of gravity is at a minimum distance of 10 mm and at a maximum of 17.3 mm away from the target. The accessibility exhibits the best range with a minimum of 3 and a maximum of 9. The minimum value cannot be less than 3 due to the fan, which must remain in between the radiator and the motor block.

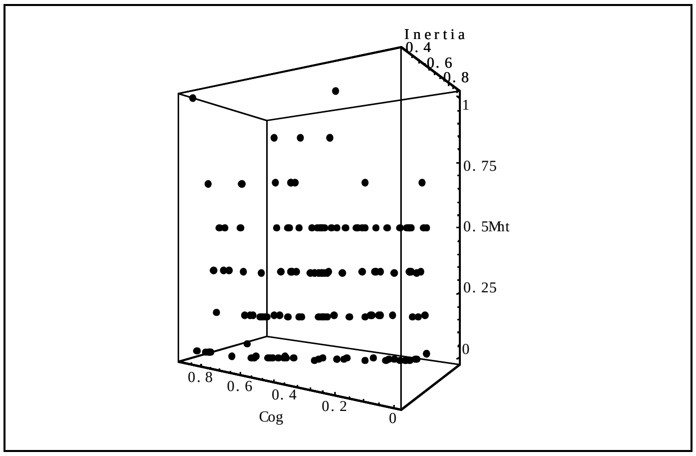


Figure 15. Three objective Pareto set of the car engine ECDP provided by the CDOM.

From the mechanical engineering point of view, the extreme solutions (Figures 13 and 14) can be compared to existing designs. First, the solutions presented do not align the fan and the longitudinal axis of the motor block. Since no fluid flow factors nor cooling were taken into account, the fan and the radiator are moved in order to optimize inertia and center of gravity objectives. In most cars the fan is aligned with the central longitudinal axis of the car. Putting the battery and water tank close to the motor might be a bad choice, first because the coolant must be put away from heat, second because the battery may take the place of another component necessary to the motor such as the air filter, and third because local areas of the motor block might be more specifically accessible. The addition of a heat transfer objective would address these issues.

CONCLUSION

This paper introduced a method based on a Genetic Algorithm using a population of clouds of points to discover Pareto alternate solutions to free-form configuration design problems.

The generality of the CDOM is assessed by its robustness in handling different shapes. The decrease in performance is, on average, linear, while the complexity increases exponentially. However, three limitations must be taken into account. First, the number of penalty functions increases exponentially with the number of objects. Second, the size of the genomes increases linearly as a function of the number of variables times the number of points in each cloud of the GA population. For a population composed of 15 point per cloud, the genome size of each individual is 15*(number of variables)*(number of bits per variable). Thus, there is a trade off between the number of components the CDOM is able to manage and the number of points per clouds used by the GA.

Two strategies have been presented to enhanced the CDOM performance: Local Search and Range Relaxation.

Local search, which is usually used to improve the accuracy of the solutions is used in CDOM to retrieve a feasible solution from an unfeasible one and thus improves the chances to discover new feasible configurations. Eventually, the analysis of the test cases pointed to two technical flaws in the LS and in the relative placement of objects. First, the LS used introduced a bias in the feasible configurations penalizing the maintainability objective function. Second, it is preferable to avoid relative placement when many components can be permuted (like in the cubes configuration problem).

In addition to the local minima introduced by the mechanical objective functions and the penalties, the distribution of the points along the Pareto set introduces many local optima in the landscape investigated by the GA. The RLX method, which consists of inflating the region investigated around the latest best solution, helps to escape these local minima and thus provides better points distribution.

From the mechanical engineering point of view, the output of the CDOM provides useful information to the designer. First, a set of alternate optimal solutions is given (Figures 9 and 15) guiding the positioning of the components. Second the Pareto set can be used as a basis to compare ‘hand generated’ solutions and either point out configurations that were never investigated or indicate if some previous designs were optimal. Third, mechanical characteristics of the system might emerge from the study of the solution set. For example, the fact that the set is small on the center of gravity axis might prove that moving components has little influence on this objective.

APPENDIX

Technical Descriptions

Galib-2.4 is a C++ library developed at the Massachusetts Institute of Technology by Matthew B. Wall (1996). The GA used in the CDOM is a Steady State GA i.e. a GA with overlapping populations. The overlap of each population is 25%. The selection scheme is a Roulette Wheel selector that selects individuals proportionally to the value of their fitness (Goldberg 1989). The scaling method consistently used along all this work is Sigma Truncation and the crossover is a single point crossover. The population size is of 15 members and the number of designs per child is 15. The genes length is 10 bits providing a precision of $L/2^{10}$ (roughly L/1000) on each variable, where L is the size of the variable interval of variation during the current run.

Due to the fact that each genome represents in fact a set of 15 different configurations (and each configuration being itself defined by N variables), the total length of each individual genome is equal to 10*(N)*15 bits. The stopping criterion is based on convergence. A failure to improve the best fitness of 0.001% stops the current GA run and possibly starts a new one.

Genetic Algorithms Technical Descriptions

Population size: 20.
Genome length: 10.
Termination Criterion upon convergence over 20 generations: 0.99999.
Probability of Mutation: 0.005.
Probability of Crossover: 0.6.
Scaling: Sigma Truncation.
Selection: Roulette Wheel.

The scaling method consistently used along all this work is Sigma Truncation. The selection scheme adopted is the Roulette Wheel Selector, which selects an individual with a probability proportional to the magnitude of its fitness score relatively to the rest of the population.

The second GA used in this work is a SteadyStateGA i.e. a GA with overlapping populations. The overlap of each population is 25%.

The population size was of 15 members and the number of designs per child is 15. The genome length is 10. This makes the length of each individual genome equal to 10*(number of variables)*15 bytes. The maximum genome length is thus equal to 2700 bytes (18 variable 3D CDPs).

The stopping criterion is based on convergence. A failure to improve the best fitness of 0.001% stops the GA.

ACKNOWLEDGMENTS

The authors wish to gratefully acknowledge the support of NASA under grants Number NGT10008 and NAG3-2046, and of the US Army Tank-Automotive and Armaments Command- National Automotive Center under Grant Number DAAE07-97-C-X130.

REFERENCES

Chevalier, A. (1986). Guide du dessinateur industriel, Hachette.

Corcoran, A. L., and Wainwright, R. L. (1992). "A genetic Algorithm for Packing in Three Dimensions". SAC'92 Proceedings of the 1992 ACM/SIGAPP Symposium, Kansas City, ACM Press NY.

National Research Council (1991). Improving Engineering Design. Designing for Competitive Advantage, National Academy Press.

Dassault Systemes, CATIA Solutions, 9 Quai Marcel Dassault, Suresnes, France.

Fonseca, C. M., and Fleming, J., (1993). "Genetic Algorithms for Multi Objective Optimization, Formulation, Discussion and Generalization". Proc. 5th Int. Conf. on Genetic Algorithms, Urbana Champaign, IL.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, Addison-Wesley.

Grignon P.M. and Fadel, G. M., (1999) " Multi-objective Optimization by Iterative Genetic Algorithm" ASME DETC99/DAC-8576, Las Vegas, NV

Grignon, P. M. (1999), "Configuration Design Optimization Method". Ph.D. Dissertation, Clemson University Mechanical Engineering Department, Clemson SC.

Grignon, P. M., Wodziak, J. R, and Fadel, G. M. (1996). "Bi-Objective Optimization of Components Packing Using a Genetic Algorithm." Multidisciplinary Analysis and Optimization Conference, AIAA/NASA, Seattle, WA,

Grignon, P. M., and Fadel, G. (1997). "Quality criteria measures for multi objective solutions Obtained with a Genetic Algorithm". AIAA SDM Conference, Orlando.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems., The University of Michigan Press. Ann Arbor, MI

Lewis, H. R., and Papadimitriou, C. H. (1978). "The Efficiency of Algorithms." Scientific American(January): 96-109.

Lomangino, P. F. (1994). "Grammar and Optimization-Based Mechanical Packaging", Ph.D. Thesis, Georgia Institute of Technology.

Moscato, P. (1989). "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms". Pasadena, CA, CalTech.

O'Rourke, J. (1993). Computational Geometry in C. Cambridge, Cambridge University Press.

Schaffer, J. D. (1984). "Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms".

Shaffer, J. D., and Grefenstette, J. J. (1985). "Multiobjective Learning via GA." Proc. 9th Int. J. Conf. Artif. Intel.: 593-595.

Sobol, I. M. (1967). "On the distribution of points in a cube and the approximate evaluation of integrals." USSR Comput. Math. math. Phys. 7: 86-112.

Sobol, I. M. (1990). "A Global search for multicriterial problems". 9th International conference on Multiple Criteria Decision Making, Washington.

Stewart, I. (1993). "Packing Problems in a Sport-gear Shipping Room." Scientific American: 142.

Szykman, S., and Cagan, J. (1995). "A Simulated Annealing Approach to Three-Dimensional Component Packing." ASME Journal of Mechanical Design, 117(2A):308-314.

Szykman, S., and Cagan, J. (1996). "Constrained Three Dimensional Component Layout Using Simulated Annealing." ASME Journal of Mechanical Design, 119(1):28-35.

Sachdev S., Paredis, C. J. J., Gupta S. K., and Talukdar S. N., "3D Spatial Layouts Using A-Teams," Proceedings of DETC98, Atlanta, GA.

Venugopal, V., and Narendran, T. T. (1992). "A Genetic Algorithm Approach to the Machine-Component Grouping Problem with Multiple Objectives." Computers and Industrial Engineering 22(4): 469-480.

Wodziak, J. R. (1994). "Optimal Packing Utilizing Genetic Algorithms". MS thesis, Mechanical Engineering. Clemson University.

Yin S. and Cagan, J. (1998). "A Pattern Search-Based Algorithm For Three-Dimensional Component Layout." Proceedings of DETC98, Atlanta, GA.