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

DETC99/DAC-8576

MULTIOBJECTIVE OPTIMIZATION BY ITERATIVE GENETIC ALGORITHM

Dr. 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

Design, multi-objective, Pareto, optimization, Genetic Algorithms.

ABSTRACT

This paper presents a method to simultaneously produce multiple solutions to unconstrained multi-objective optimization problems. The proposed methodology uses populations of sets instead of populations of individuals and iterative calls to a Genetic Algorithm (IGA) to obtain a set of solutions spread across the Pareto set in the objective space. The superiority of such an approach to single run, conventional population Pareto GAs is shown. The various difficulties of the algorithm and the methods used to overcome them are detailed. Finally, the paper expands upon how this method can be used with or without user inputs, and shows an analysis of its performance by applying it to a succession of increasingly difficult problems, identifying its range of application.

INTRODUCTION

This research is one step in our search for a methodology to deal with multi-objectives problems (MOPs) encountered in complex configuration type designs. Research in the field of design methodology addresses relatively poorly the task of assembly of components subject to the optimization of several criteria. In the course of our research in Optimal Configuration Design, we were confronted many times by non-linear non-polynomial functions with too many parameters to rely on other techniques than numerical calculus. In addition to this first difficulty, the objectives commonly used did not have any algebraic expression but were derived from the integration of differential equations over freeform domains. These characteristics directed us toward non-deterministic strategies. The presence of both discrete and continuous variables, as well as disconnected regions in both parameter and objective spaces favored the genetic algorithm (GA) as the tool to search for a better solution. In addition to the versatility of the GAs to deal with various types of variables and functions, the topology of the final set was expected to be more complex than the initial sets in both parameter and objective spaces, and this confirmed the choice of the GA.

After using with success a method based on a Genetic Algorithm to produce Pareto solutions to a packing problem involving center of gravity and volume (Grignon et al., 1996),the method was improved in order to produce multiple solutions to MOPs.

BACKGROUND

Overview of Genetic Algorithms

Genetic Algorithms (GAs) (Goldberg 1989; Holland 1975) are non-deterministic search algorithms based upon the evolution of successive populations of points called generations. Under some external pressure, and using a heuristic, a new population is deduced from the previous one. From generation to generation, the populations, pushed by an external pressure, migrate through the variable space under investigation. This migration stops when either convergence is reached, i.e. when the fitness of the best element does not improve sufficiently during a specified number of generations, or when the specified maximum number of generations is reached.

In the field of function optimization, the external pressure is the function to be optimized and the individuals of the population are some of the points of the variable space to be investigated. The GA then pushes the points toward the optimum of the function using a procedure first described in 1975 by John Holland of the University of Michigan (Holland, 1975) and based on the mechanisms of selection, recombination, reproduction and mutation.

Each of these operators is the subject of research in the field of Genetic Algorithms but it is not the purpose of this paper to detail them. The GA can be considered as a function that deduces a new generation $\{g\}_{i+1}$ of individuals from a previous generation $\{g\}_i$: $\{g\}_{i+1} = GA[\{g\}_i]$

Multicriterion Optimization

Problem Statement

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 $\mathbf{F}(\mathbf{x})=(f_1(\mathbf{x}), f_2(\mathbf{x}), .. , f_m(\mathbf{x}))$, the MOP can be stated as follows:

$$\mathbf{F} : \mathbf{x} \rightarrow \mathbf{F}[\mathbf{x}]$$
$$\mathbf{R^n} \rightarrow \mathbf{R^m}$$
Find one $\mathbf{x}$ such that $\mathbf{x}$ is non-inferior. (1)

Non-inferiority definition: 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.

Different Techniques

Multi-objective optimization really took-off after the publication, in 1951, of the necessary and sufficient condition for non-inferiority by Kuhn and Tucker (1951). Since then, the mathematical methods to find a solution to an unconstrained multi-objective problem can be put into one of the following categories (Statkinov and Matusov, 1995):

- Substitution of all the criteria by a single one (scalarization) either by choosing the most important, or by building a combination of all the criteria into a single one. In this last category one can find the weighting method (Zadeh, 1963) in which all the criteria are replaced by their weighted sum and the k^{th}-objective Lagrangian problem.
- Consecutive optimization of all the criteria.

In the first category, the weighted sum method finds a single solution in the Pareto set for each set of weights. The restrictions on the differentiability and continuity of the objective and constraint functions depend on the optimization technique chosen to solve the new problem. Independently of this first restriction, the weighted sum method is unable to find Pareto points inside a duality gap (non-convex portion of the Pareto set in objective space, Figure 1). Many classical methods present the same drawbacks. A description of these methods can be found in (Chankong, 1983). Athan and Papalambros (1996) also looked at using non-linear weights to better capture the non-convex Pareto set.

Several other approaches to solve the multi-objective optimization problem can be found in the literature, among them is Goal Programming, first studied in depth in the work of Levary (1984), and Compromise Programming.

Recent efforts were made to find methods applicable to non-convex Pareto sets. Kostreva and Wiecek (1993) proposed a method for non-convex Pareto sets generated by linear complementary problems. TenHuisen and Wiecek (1996) use a nonlinear scalarization based on a quadratic Lagrangian function for producing locally efficient solutions.

The major impediment for using many of these techniques comes from the strong assumptions made on the continuity and differentiability of the objective and constraints functions. These assumptions are difficult to guarantee in Engineering Configuration Design. A second drawback comes from the fact that they offer a single solution to the Multi-objective optimization problem. These two drawbacks fostered our proposition to use the Genetic Algorithm for solving MOPs.

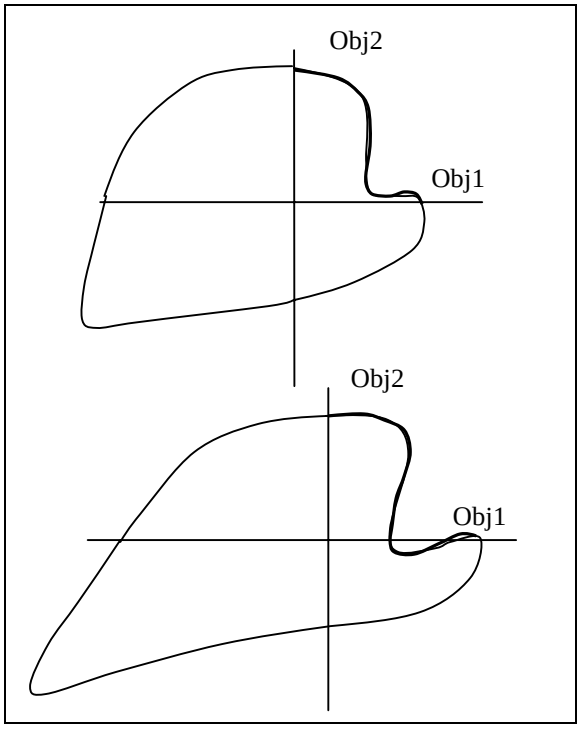

Figure 1. Duality gaps refer to the non-convex part of the boundary of the curves.

GA and Multi-objective Optimization

The major problem encountered by researchers trying to use GAs to obtain multiple solutions at the same time is to balance the trend of the GA to bring points together. For example, Schaffer in (Schaffer, 1984) proposed to select sub-populations of points proportionally to each objective in turn. For n objectives, n sub-populations are generated with an equal number of individuals. The final population is obtained by shuffling all the sub-populations. Unfortunately this algorithm tends to converge to few extreme members of the population after a large number of iterations. Schaffer then tried to prevent this convergence from happening by mating far-away individuals. This mechanism was not selective enough to remove poorly fitted individuals, which mated with good individuals. Finally, he concluded that random mating was more

efficient.

Baker (1985) used a single criterion based on a ranking of the individuals. The points are first sampled and all non-dominated points are removed from the population. Goldberg (1989) points out that 'this should be used in conjunction with niche and speciation for stabilizing the multiple sub-populations arising along the Pareto-optimal front and thereby preventing excessive competition among distant population members'. Richardson et al. (1989) argued that this is equivalent to linearly combining the objectives like in the weighting method with different weights at each generation. Similarly to Schaffer's method, this leads to speciation.

However, Horn and Nafpliotis (1994) used a niched Pareto GA through a combination of two methods: Multi-attribute Utility Analysis (MAUA) and GA to deal with multicriteria optimization problems. Two approaches were chosen: either using MAUA in order to combine the criteria and then optimize with the GA, or use the GA to find the Pareto set and eventually use MAUA to select among the Pareto points. In order to find the Pareto set they relied on the Niched Pareto Genetic Algorithm which uses a new kind of tournament selection: Pareto domination tournaments with sharing since, Oei et al. (1991 #86) showed that niched GA with conventional tournament selection does not give steady results.

In order to avoid the previous drawbacks, Fonseca et al. (1993) proposed a selection scheme based on the definition of the Pareto set which ends up with a ranking of the points based on how many points dominate the current point. However, this latter method does not differentiate individuals that are closer to the Pareto set. They also used a sharing in the objective space instead of in the variable space in order to get a uniformly distributed population in the objective space. They calculated the sharing parameter σ_{share} using an over estimate of the size of the Pareto set in the objective space. Then, they noted that mating restrictions (a parameter similar to σ_{share} called σ_{mating}) which helps to avoid excessive competition between the distant members of the population may not be efficient due to the respective low number of points in the population with respect to the area of the Pareto set. In order to prevent this drawback, they used higher-level decisions (a decision-maker) in order to focus on some areas of the Pareto set. Finally, Osyczka and Kundu (1996) presented a method based on a fitness calculated using the Euclidean distance of the current point in the objective space from a Pareto set updated during the run of the GA. A in depth analysis of these methods (Bentley and Wakefield 1997) shows that the Non-Dominated Sorting described in Goldberg (1989) and used in the IGA, gives good points distribution along the Pareto set.

Two issues relevant to the use of the GA for MOPs have been attracting the efforts of the researchers:

- First, find a fitness function that translates the multiple criteria into a single one, since the GA is only able to handle one objective function.
- Second, prevent the GA population from collapsing to one or few points.

These two issues directed the current research, especially the replacement of population of individuals by populations of sets of individuals which, in the view of the authors, brings back the use of the GA to its initial design-i.e. clustering solutions together using the evolutionary principles.

TERMINOLOGY AND PROBLEM REFORMULATION

Thus, the initial multi-objective optimization problem (MOP) statement (1) was modified as follows:

Given a function $\mathbf{F}[\mathbf{x}]$ where $\mathbf{x} = (x_1, x_2, .. , x_n)$ is a vector of a domain D in R^n and $\mathbf{F}[\mathbf{x}]$ is a vector such that $\mathbf{F}[\mathbf{x}]=(f_1(\mathbf{x}), f_2(\mathbf{x}), .. , f_m(\mathbf{x}))$ in which the f_i are functions of R^n -> R

$\mathbf{F} : \mathbf{x} \rightarrow \mathbf{F}[\mathbf{x}]$

$\mathbf{R^n} \rightarrow \mathbf{R^m}$

Find a set of vectors $\mathbf{x}$, denoted { PS_X } = $\{\mathbf{x_1}, \mathbf{x_2}, , \mathbf{x_{ps}}\}$ such that

$\mathbf{F[x_k]}$ is non-inferior for all $\mathbf{x_k}$ in { PS_X }

And

Min[Dist[$\mathbf{F[x_k]}$, $\mathbf{F[x_l]}$]] is maximized for any two $\mathbf{x_k}$ & $\mathbf{x_l}$ in {PS_X} with k different from l.

With all the components of the vectors $\mathbf{x}$ bounded such that

$xl_i <= x_i <= xu_i$ for i= 1..n (2)

DEVELOPMENT OF THE ALGORITHM

In our previous attempts to identify multiple solutions to multicriteria problems, we developed a method based on the search for a cloud of points rather than a single point used in conventional linear and non-linear optimization processes (Grignon and Fadel, 1997). In this previous algorithm, the GA was used to identify a cloud of points that covered as much as possible, and as evenly as possible, the Pareto set in the objective space in a single run. This approach gave very good results on small scale problems with and without dual gaps, but converged to sub-optimal solutions when the Pareto set became more diluted in the total objective space (for more details on dual gaps see Chankong and Haimes, 1983, p. 136). This phenomenon was accentuated by the numerous local minima introduced by the even distribution requirement.

Neither our attempts to increase the population size and number of generations, nor to delay the convergence by imposing a mutation rate dictated by a heuristic (like the cooling schedule of the Simulated Annealing (Kirkpatrick 1983) method which makes the search first very random but gradually prevents wide variations) gave satisfying results. The first method was still converging to non-optimal solutions and the second needed to be adjusted as soon as the problem was changing.

In order to overcome these convergence problems, and to

exploit the fact that the GA had a very good performance only during the first few hundred generations and then leveled off, we decided to base our algorithm on iterative runs. This multiple run method gave the additional advantage to allow the user to intervene during the optimization process in order to focus the search on some areas of the parameter space, which was not easy with the single run method. 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. A single run of the GA consists of:

```
   Generate randomly the first
population {G}0 of 'c' clouds of 's'
genomes
   {g}1 = { {g1, g2, .., gs}1, {g1, g2,
.., gs}2, .., {g1, g2, .., gs}c }0
  attributed to c clouds of vectors x:
  {x}0 = { {x1, x2, .., xs}1, {x1, x2, ..,
  xs}2, .., {x1, x2, .., xs}1 }0
  each variable being bracketed by two
  bounds xil and xiu

   Begin
    i = 1
    While stopping criterion not met
       Update the reference Pareto set
       {PS}ref
       Evaluate the fitness of each
       cloud {g1, g2, .., gs}j of {g}i.
       Do
            Selection
            Crossover
            Mutation
       EndDo   (this gives {g}i+1)
       i = i + 1
    EndWhile
   End                                  (3)
```

The fitness function calculation is based on the best approximation of the Pareto set in the objective space $\{PS\}_{ref}$ discovered by the GA at the moment of evaluation of an individual. This fitness function is designed to privilege the cloud with a shape close to a Pareto set morphology and presenting the largest spread. The cloud morphology is measured using a rank based definition of the notion of non-inferiority (similar to the fitness used by Baker) coupled with a measure of the distribution.

In order to calculate the objective value of a cloud, each solution point inside this cloud is considered in the objective space and has a rank calculated by comparing its position with the position of its cloud mates and with the points of a reference Pareto set $\{PS\}_{ref}$. The reference Pareto set is made of all the best points (non-inferior) found so far (Figure 2). Since the number of points in this reference set is bounded (because of the limited memory size) but not fixed, the grade of the cloud is normalized with respect to the total number of points (4). Thus, for a cloud of 'c' points $\mathbf{F[x_i]}$ and a reference set $\{PS\}_{ref}$ of 'rs' points, the grade of the cloud is

$$Grade_{cloud} = \sum_{i=1..c} [\ \mathbf{Rank}[\ \mathbf{F[x_i]}\ ]\ ] / (\ c + rs\ )^2 \qquad (4)$$

with

Rank[$\mathbf{F[x_i]}$] = c*rs minus the number of non-inferior points with respect to $\mathbf{F[x_i]}$ in $\{PS\}_{ref}$. (5)

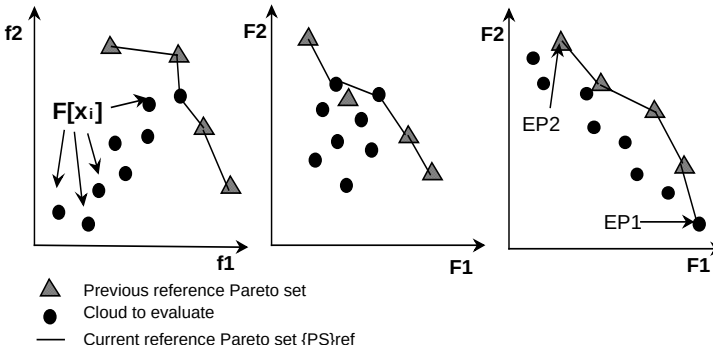

Figure 2. The reference Pareto set in the case of a maximization MOP.

The grade of a cloud is maximum when its points are all non-inferior with respect to its cloud mates and with respect to the points of the reference Pareto set. This grade is then updated in order to take into account the spread and the distribution of the points inside the cloud. They are both 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.

$$penalty_{sp} = Atan\ [\ Max[1/Distance[\mathbf{F[x_i]}, \mathbf{F[x_j]}]\ ]\ ] / (\Pi/\ 2)$$
$$i= 1..c,\ j=1..c. \qquad (6)$$

The cloud grade then becomes equal to

$$Grade_{cloud} = (\ \sum_{i=1..c} [\ \mathbf{Rank}[\ \mathbf{F[x_i]}\ ]\ ]\ ) / (\ c + rs\ )^2 + penalty_{sp}. \qquad (7)$$

Special care must be given to the recording of the extreme points of the reference set (EP1 and EP2, Figure 2). In order to get solutions as close as possible to the bounds of the real Pareto set, the EPs are only replaced by better extreme points such that the spreading of the reference Pareto set is always the best spreading found so far. This strategy guarantees a convergence of the reference set extreme points 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.

The stopping criterion is based on a maximum number of objective function calculations (which is dictated by the objective function computation time). From this maximum number, one can make a tradeoff between the population size, the cloud size, the number of generations per iteration and the

total number of iterations.

After each iteration, the bounds $\mathbf{x_l}$ and $\mathbf{x_u}$ are adjusted in function of the reference Pareto set. The components of $\mathbf{x_l}$ and $\mathbf{x_u}$ respectively are the minima and the maxima of all the corresponding components of the reference Pareto.

QUALITY CRITERIA OF THE SOLUTION

Several criteria were used to measure the quality of the solutions. First, the spreading of the final cloud, by measuring the length of the best front and by the enclosing two ellipses. Second, the accuracy of its bounds (EPs) with respect to the bounds of the real Pareto set (TEPts). Finally, the distribution of the points in the final front (Figure 3).

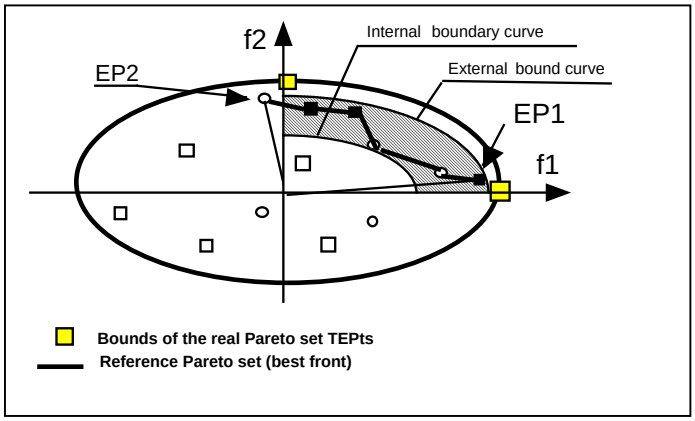

Figure 3. Quality criteria.

EXAMPLES

The evolution of the performance of this algorithm was studied as the problem grew in complexity. The complexity can increase in many ways; we selected two: keeping the same shape and topologies in the objective space, the size of the Pareto set in the parameter space decreases with respect to the total variable space size. This means that the Pareto set becomes more difficult to find. This is not always achieved by increasing the number of variables, as is shown in the examples. The complexity of the problem also increases when the number of objectives increases and when they become more non-linear. In multi-objective optimization this can be translated by the presence of large dual gaps which split the Pareto set into several regions.

In order to study these two phenomena, we built three 2-objective examples and three 3-objective examples ranging from the minimum number of variables, i.e. 1 variable for 2-objectives and 2 for 3-objectives, to 8 variables in each case. The goal of the process is to maximize the objectives and to obtain a set of points spread along the Pareto set. The evolution of the quality criteria is studied when the algorithm is applied to these examples (Table 1).

Table 1. Summary of the examples used.

Variables	2-Objective Examples			3-Objective Examples		
1	1	2	3			
2	1	2	3	4	5	6
3	1			4	5	6
4	1			4		
5	1			4	5	6
6	1	2	3	4		
7	1			4		
8	1			4		

The genetic algorithm used for performing the test cases is a steady-state GA with a population of 20 individuals (each corresponding to a cloud of 15 points). The mutation rate is 0.005. The crossover probability is 0.6 and the population replacement is 0.25. No elitism is used. The selection is a roulette wheel algorithm that favors the selection of the population members with the highest fitness.

Example 1: 2-Objective MOP with no dual gap

This MOP is based on 2-objective functions of one real variable x_1 (Equations 8) whose range is]-Π, Π]. The representation of the image of the interval]-Π, Π] in objective space is an ellipse depicted on Figure 4.

$$f_1[x_1] = 3\,\mathrm{Cos}[x_1] \qquad (8)$$
$$f_2[x_1] = 2\,\mathrm{Sin}[x_1]$$

The n variable version of this problem consists in introducing a common multiplier in front of each of these functions. The range of x_1 remains]-Π, Π] and the range of all the other variables is [0, 1].

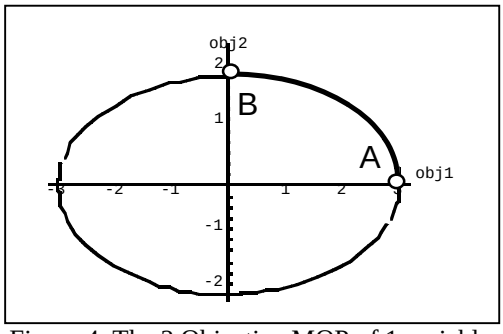

Figure 4. The 2 Objective MOP of 1 variable.

The Pareto set in the objective space covers $1/4^{th}$ of the total objective and variable space size.

$$f_1[x_1, .. , x_n] = 3\,\mathrm{Cos}[x_1] \prod_{i=2..n}(x_i) \qquad (9)$$
$$f_2[x_1, .. , x_n] = 2\,\mathrm{Sin}[x_1] \prod_{i=2..n}(x_i)$$

For each additional variable, the Pareto set size in objective and variable spaces must be divided by 65536 (due to the variable encoding, see Results section for more explanation) since only $x_i = 1$ (i=2..n) corresponds to the real Pareto set.

Example 2: 2-Objective MOP with shallow Dual Gap

The only difference between examples 2 and 1 is in the definitions of functions f_1 and f_2 (Equations 10). The representation of the image of]-Π, Π] in the objective space is an ellipse with a shallow dual gap depicted on Figure 5.

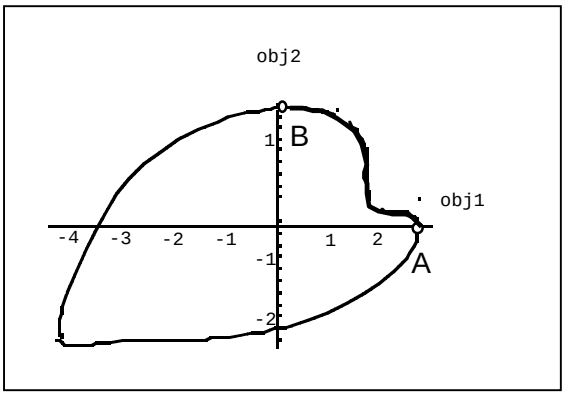

Figure 5. Deformed ellipse exhibiting a shallow dual gap

$$f_1[x_1] = \cos(a)(\cos(x_1)+2\sin(x_1))- \sin(a)(3\cos(x_1)+1/(b+(2\sin(x_1)-1)\ (2\sin(x_1)+1)$$
$$f_2[x_1] = \sin(a)(\cos(x_1)+2\sin(x_1))- \cos(a)(3\cos(x_1)+1/(b+(2\sin(x_1)-1)\ (2\sin(x_1)+1)$$
with a=130 degrees and b=1.65. (10)

The Pareto set in the objective space extends from points A to B (Figure 5) and from 0 to Π/2 in variable space. Its relative size is $1/4^{th}$ of the total variable space size. This set has exactly the same dimension as the one of example 1.

The n variable version of this problem is produced in the same way as in the previous case. The range of x_1 remains:]-Π, Π] and the range of all the other variables is [0, 1].

$$f_1[x_1, .. , x_n] = f_1[x_1] \prod_{i=2..n}(x_i)$$
$$f_2 [x_1, .. , x_n] = f_2[x_1] \prod_{i=2..n}(x_i) \quad (11)$$

Example 3: 2-Objective MOP with Deep Dual Gap

The formulation of this example is obtained with equations (10) and (11) except that the value of parameters a and b are adjusted in order to produce a deeper dual gap in objective space (Figure 6).

The Pareto set is split into two regions in both objective and variable spaces. In the objective space, it extends from A to B and from C to D. In the variable space the Pareto set extends from 2.2 to 2.78 and from 3.56 to 3.82. Its length is 0.84. Since the range of the variable is 2Π, the relative size of the Pareto set in the variable space is $0.84/(2\Pi) > 0.13$.

The n variables version of this problem is produced in the same way as in the previous case (Equation 12). The range of x_1 remains]-Π, Π] and that of all the others is [0, 1].

$$f_1[x_1, .. , x_n] = f_1[x_1] \prod_{i=2..n}(x_i)$$
$$f_2 [x_1, .. , x_n] = f_2[x_1] \prod_{i=2..n}(x_i) \quad (12)$$

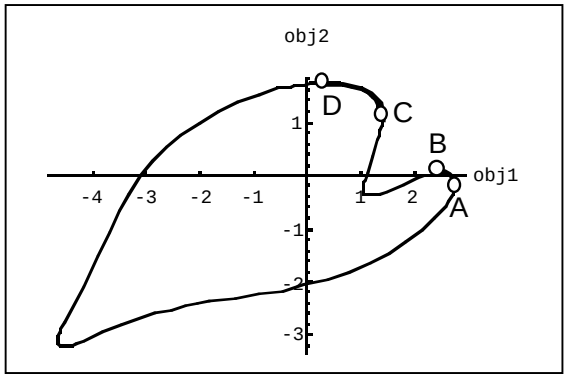

Figure 6. Deformed ellipse exhibiting a deep dual gap obtained with a= 135 degrees and b=1.4.

Example 4: 3-Objective MOP with no dual gap

This MOP is based on 3-objective functions of two real variables x_1 and x_2 (Equation 13). The representation of this MOP in objective space is an ellipsoid depicted on Figure 7.

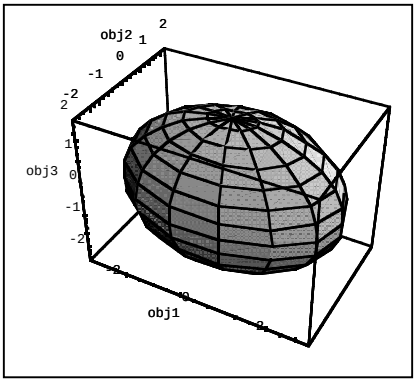

Figure 7. The 3-Objective MOP of 2 variables.

$$f_1[x_1, x_2] = 3\ \mathrm{Cos}[x_1]\ \mathrm{Cos}[x_2] \quad (13)$$
$$f_2[x_1, x_2] = 2\ \mathrm{Cos}[x_1]\ \mathrm{Sin}[x_2]$$
$$f_3[x_1, x_2] = 2\ \mathrm{Sin}[x_1]$$
with x_1 in [-Π/2, Π /2] and x_2 in]- Π, Π]

In this case, the Pareto set in the objective space covers $1/8^{th}$ of the total space size.

The n variables version of this problem (n > 2) is given by:

$$f_1[x_1, .. , x_n] = f_1[x_1, x_2] \prod_{i=3..n}(x_i) \quad (14)$$
$$f_2[x_1, .. , x_n] = f_2[x_1, x_2] \prod_{i=3..n}(x_i)$$
$$f_3[x_1, .. , x_n] = f_3[x_1, x_2] \prod_{i=3..n}(x_i)$$

The Pareto set size generated by these examples is equal to the size of the previous set divided by 65536 for each additional variable beyond 2.

Example 5: 3-Objective MOP with shallow dual gap

In this example the equations allow a dual gap to appear along the surface described in objective space. The depth of this dual gap can be adjusted in function of the value of parameter b. The orientation of the gap is driven by a rotation matrix **[R]** involving three rotation angles a_1, a_2, a_3 applied to the vector $\{f_1, f_2, f_3\}$.

$$f_1[x_1, .. , x_n] = 3 \text{ Cos}[x_1] \text{ Cos}[x_2] + 1 /(((b + (-1 + 2 \text{ Sin}[x_1]))(1 + 2 \text{ Sin}[x_1]))(b + (-1 + 2 \text{ Sin}[x_2]) (1 + 2 \text{ Sin}[x_2])))$$
$$f_2[x_1, .. , x_n] = 2 \text{ Cos}[x_1] \text{ Sin}[x_2]$$
$$f_3[x_1, .. , x_n] = 2 \text{ Sin}[x_1] \quad (15)$$
$$\{f1, f2, f3\} [\mathbf{R}]$$

The parameters values were chosen such that the dual gap is not deep enough to change the topology of the real Pareto set in objective space. Hence, the size of this set remains the same as in the previous example.

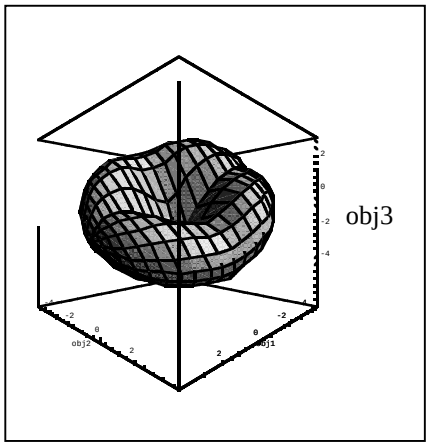

Figure 8. Deformed ellipsoid exhibiting a shallow dual gap. This shape was generated using equation 15 and parameters $a_1 = 40$, $a_2 = -55$, $a_3 = 115$, $b = 1.75$.

The n variable version of the 3 objective MOP consists in introducing the remaining variables $\{x_2,.., x_n\}$ as multipliers in front of each of these functions (Equation 16). The range of x_1 remains]-Π, Π] and that of all the other variables is [0, 1].

$$f_1[x_1, .. , x_n] = f_1[x_1, x_2] \prod_{i=3..n}(x_i) \quad (16)$$
$$f_2[x_1, .. , x_n] = f_2[x_1, x_2] \prod_{i=3..n}(x_i)$$
$$f_3[x_1, .. , x_n] = f_3[x_1, x_2] \prod_{i=3..n}(x_i)$$

with

x_1 in [-Π/2, Π/2]

x_2 in]-Π, Π]

x_i [0, 1] for i= 3 .. n.

Example 6: 3-Objective MOP with deep dual gap

A deeper dual gap is produced by using the same equations and by changing the parameter values.

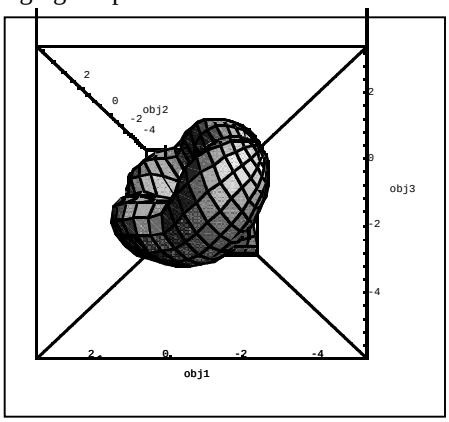

Figure 9. $a_1 = 40$ deg., $a_2 = -65$ deg., $a_3 = 115$ deg., $b = 1.4$.

Results

In our GA, each continuous variable was encoded with a 16-bit binary string. Hence, each continuous variable was seen by the GA as a discrete variable able to take 65536 different values. With this condition, an n-variable problem meant a total variable space size of $(65536)^n$ points.

Knowing this, and for our examples, the increase of the number of variables implied a decrease of the Pareto set size with respect to the total variable space. Adding variables to the problem produced then the worst possible effect-i.e. decreasing the chances to find the Pareto set (Table 2).

Table 2. Evolution of the real Pareto set size.

	2-Objs. Examples			3-Objs. Examples		
Vars	**1**	**2**	**3**	**4**	**5**	**6**
1	0.25	0.25	1.30^{-01}	-	-	-
2	3.81^{-06}	3.81^{-06}	1.98^{-06}	0.25	0.25	1.30^{-01}
3	5.82^{-11}	5.82^{-11}	3.03^{-11}	3.81^{-06}	3.81^{-06}	1.98^{-06}
4	8.88^{-16}	8.88^{-16}	4.62^{-16}	5.82^{-11}	5.82^{-11}	3.03^{-11}
5	1.35^{-20}	1.35^{-20}	7.05^{-21}	8.88^{-16}	8.88^{-16}	4.62^{-16}
6	2.06^{-25}	2.06^{-25}	1.08^{-25}	1.35^{-20}	1.35^{-20}	7.05^{-21}
7	3.15^{-30}	3.15^{-30}	1.64^{-30}	2.06^{-25}	2.06^{-25}	1.08^{-25}
8	4.81^{-35}	4.81^{-35}	2.50^{-35}	3.15^{-30}	3.15^{-30}	1.64^{-30}

All the results presented are based on statistics made on 20 tests for each case. Using a single GA run does not provide satisfactory answers as soon as the number of variables reaches 2. The distance between the internal and external boundary curves is large, showing that the shape of the best solution remains far from looking like a curve or a surface (Figure 11 right, 1 run raw). The minimum distance between the solution and the target Pareto set becomes large as soon as the second variable is introduced (Figure 11 top, 1st run row) indicating that the solution does not converge toward the target Pareto curve. The distances between points present large variations (Stdev = 0.8, Figure 12) indicating that the distribution is not even and the extreme points (EP1 and EP2) fail to converge

toward their targets (Figure 10). Beyond 3 variables, using a single run does not provide any useful results.

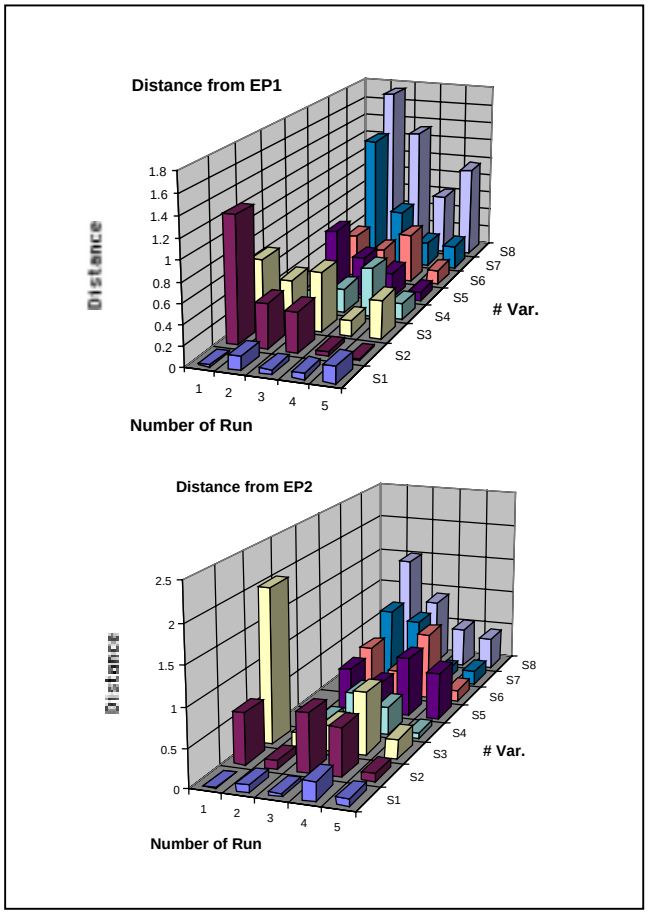

Figure 10. Distance between the best Extreme Points (Eps) and the Target Extreme Points TEPs

Introducing a second run brings an immediate gain with respect to the single GA run approach. The extreme points are closer to their target values. The degradation of the performance is slower and appears in the distances from the bounds (comparison based on row one and two of Figure 10) and in the convergence toward the boundary of the domain (Figure 11). The Pareto set keeps its curve like shape until 6 variables since the distance between the internal and external boundary curves remains below 20% (Figure 11). The distance between the internal boundary curve and the target curve increases slowly until 6 variables due to offline points. Beyond 6 variables the results cannot be exploited due to the lack of insight they provide concerning the location of the real Pareto set (Figure 11). The points distribution remains better than the distribution obtained with a single run but there is no clear trend of this evolution in subsequent trials (Figure 12).

The gain in the result quality obtained through additional iterations is not as large as the gap between single run and two run tests. The results still exhibit improvements especially in the shape of the final solution (Figure 11). The distribution of the points is the factor which is the less improved by the iterations (Figure 12).

Eventually, the size and shape of the duality gaps did not affect the ability of the method to correctly spread the points continuously even if the Pareto set is non convex and to split the spreading into several sets when the Pareto set presents several regions (Figure 13).

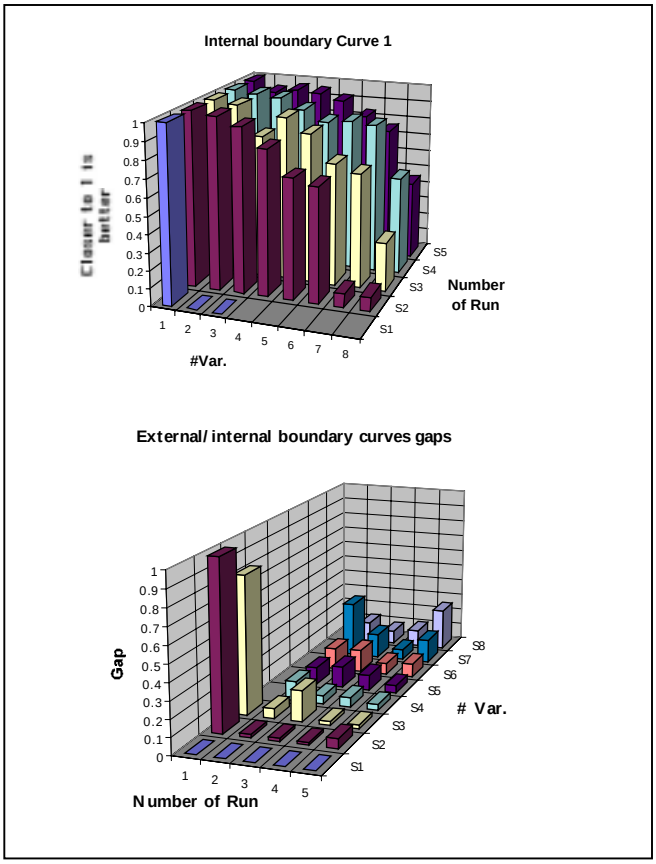

Figure 11. At left the maximum distance between the real Pareto set curve and the best solution found. At right the distance between the internal and external boundary curves of the best solution.

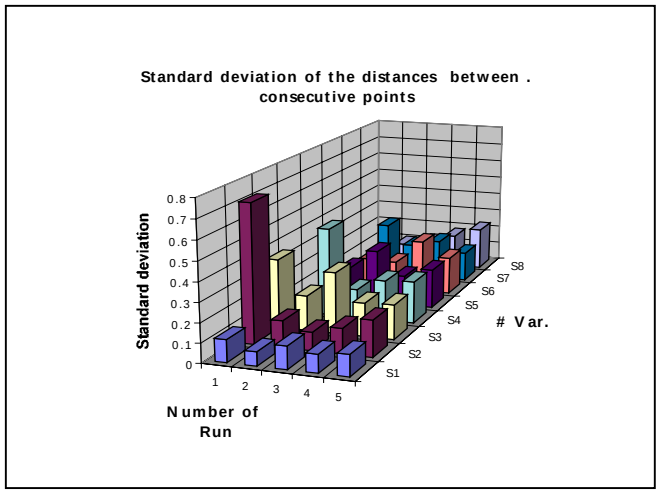

Figure 12. The distribution of the points along the solution curve is measured by the standard deviation of the distance separating neighbors.

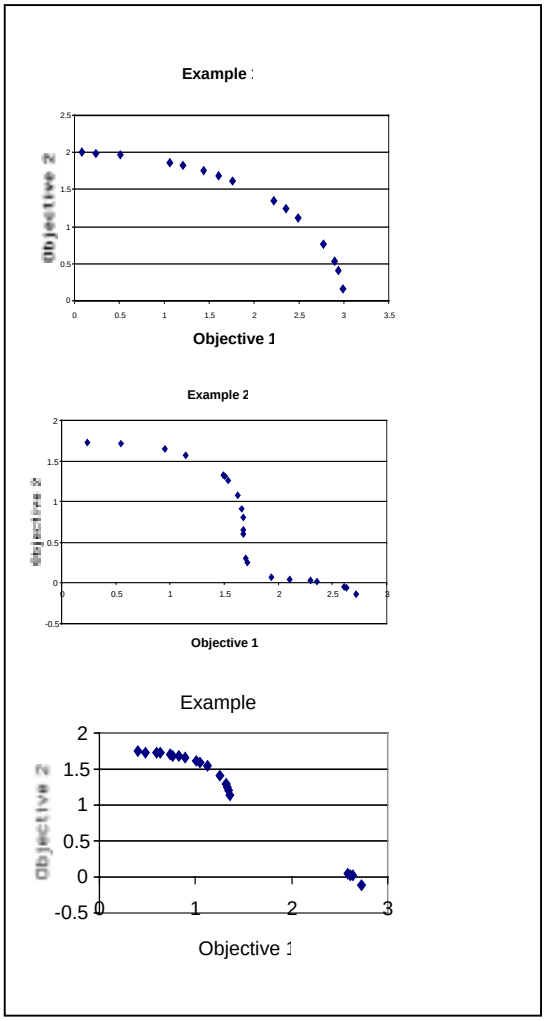

Figure 13. Examples of best solution clouds.

Observations and Discussion

The premature convergence exhibited by the GA when the number of variables increases might be due to the small size of the population. This effect is the first explanation that we could give for the small number of variables handled by this method. A second explanation could come from the absence of a relaxation strategy in the bounds of the variables after each iteration. Indeed, these bounds were set at the exact value provided by the Extreme Points (EPs) coordinates in variables space, even if these points were clearly too restrictive with respect to the real targets (TEPs). The second side effect of the bound restriction strategy might explain the fact that performance cannot increase indefinitely and can even drop if too many iterations are used (comparison of S4 and S5 rows on Figure 11 top). When the Extreme Points are close to their target, the benefit of the iteration is lost since the GA explores a domain of the same size after each iteration. The only amelioration could only come from a better distribution of the points. Since no information concerning the shape of the best solution is passed from one run to the next one, which is discussed thereafter, there is no hope for improvement from this factor either.

We were conscious that by using this approach the number of variables would be reduced due to the large size of each genome. For example, if we consider a population of clouds each owning 15 points, each of these points being a 10 variable mathematical vector, each component of these vectors being themselves coded with 20 bits, in these conditions, each genome size is 15*10*20 = 3000 bits for each population member. However, it was not clear if the addressable size was less than 10 parameters, between 10 and 100 or more than 100. Once again it seemed that it would be better to reduce the gene size allocated to each variable and increase the population size in order to obtain better convergence.

Concerning the objective functions, beyond the convenience of visualizing the results of the tests, we purposefully restricted the number of objectives to 3, because it seemed reasonable to assume that understanding the behavior of more factors would be difficult for a human being. Our final goal being the building of a tool for applied problems. The method did not seem to be sensitive to the shape of the duality gap. However, the example Pareto sets were not badly non-convex. The number of split regions was small, leaving open the question concerning the behavior of such a method if the number of these regions increases.

Using recursively a Genetic Algorithm had another purpose. It allows user-guided search after each iteration by specification of new bounds. This is useful in two cases. Either when the GA has a problem to identify good regions for the parameters, or when the user wants to focus the search in a given region after viewing the results of previous iterations. In both cases, iterations present the advantage of combining the knowledge of the user with the search power of the GA. The fact that engineers are reluctant to use black box programs to

find solutions to a problem also supports such an approach.

From the technical standpoint, each iteration consisted of 40 generations. This number allowed retaining the part of the run in which the GA is the most effective for the examples. However, choosing a criterion based on convergence would have ensured that the GA was rerun at the correct moment. Hence, the number of objective function evaluation was fixed for each iteration (15*15 + 0.25*40*15*15 = 2475).

Concerning the cumulating of knowledge from one iteration to the next one, no information is passed, the new bounds excepted. The gains due to passing more than these bounds are not granted. Seeding the initial population of the next run, thus ensuring that some of the best schemas of the previous run are immediately available, is a way to pass cloud shape information in addition to variables bounds. However, population seeding has the disadvantage of impairing the search for new solutions and is not advisable with highly non-linear functions. Knowing that, in the examples presented, the penalty functions (for evenly distributing the points) introduced n*(n-1)/2 local minima (n being the number of points in a cloud) in the initial ranking objective function, it is not clear whether population seeding is advisable.

Among the quality criteria, measuring the even distribution was a concern. In 2D, measuring if some points are well distributed along a curve only requires the knowledge of their order and the choice of a distance. In 3D the task is much more difficult and requires the building of a mesh of triangles based on these points and then the comparison of the size of each triangle. In both cases the points are expected to be evenly distributed if the Pareto set in the objective space is continuous. As soon as the Pareto set is split into several regions (Figures 6 and 9), the distances separating the different domains bias the even distribution measures. Thus, these measures can only be applied to compare two solutions of the same problem.

CONCLUSION

A method for finding multiple solutions to multi-objective optimization problems was presented. This method relies on the iterative use of a Genetic Algorithm working on populations of clouds of points rather than on populations of single points. After each iteration, the range of investigation is reduced to the most promising region. This choice eliminates the need for sharing but reduces the size of the problems that can be addressed. Finding alternate solutions to the MOP brings several advantages especially in the domain of mechanical engineering design. Combining the GA search power with engineering knowledge seems a good balance between black box optimization method and rule of thumb approaches. Several quality criteria are proposed for rating how close the best solutions are from the target Pareto set. The evolution of these same criteria is studied to compare the proposed method with a single GA run.

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 : 15.
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.

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

Athan, T. W. and Papalambros, P. Y., (1996), "A note on weighted criteria methods for compromise solutions in multi-objective optimization", *Eng. Opt*. 1996, 27, 155-176

Baker (1985). “Adaptive Selection Methods for Genetic Algorithms.” *First International Conference on Genetic Algorithms and Their Applications*, Pittsburg, 101-111.

P. J. Bentley and J. P. Wakefield, (1997). "Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms." In P. K.

Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, Part 5, 231-240, London, June 1997. Springer Verlag

Chankong, V., and Haimes, Y. Y. (1983). "Multiobjective Decision Making: Theory and Methodology."

Fonseca, C. M., and Fleming, J. (1993). "Genetic Algorithms for Multi Objective Optimization, Formulation, Discussion and Generalization." *Proceedings of the Fifth International Conference on Genetic Algorithms*, Urbana Champaign, IL, 416-423.

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

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

Grignon, P. M., and Fadel, G. (1997) "Quality Criteria Measures for Multiobjective 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 Harbor.

Horn, J., Nafpliotis, N. and Goldberg D. (1994) "Multiobjective Optimization Using The Niched Pareto Genetic Algorithm." *First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Piscataway, NJ, 82-87.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M.P. (1983). "Optimization by Simulated Annealing." *Science*, 220(4598), 671-680.

Kostreva, M. M., and Wiecek, M. M. (1993). "Linear Complementarity Problems and Multiple Objective Programming." *Mathematical Programming*, 60, 349-359.

Kuhn, H. W., and Tucker, A. W. "Nonlinear Programming." Proceedings of the Second Berkeley Symposium on Mathematics, Statistics, and Probability, Berkeley, 481-492.

Oei, C. K., Goldberg, D.E., and Chang, S. J. (1991). "Tournament Selection, Niching, and the Preservation of Diversity." Report 91011, University of Illinois at Urbana Champaign, Urbana Champaign, Illinois.

Osyczka, A., and Kundu, S. (1996). "A Modified Distance Method for Multicriteria Optimization, Using Genetic Algoritm." *Computers in Industrial Engineering*, 30(4), 871-882.

Richardson, J. T., Palmer, M. R., Liepins G., and Hillliard, M. "Some Guidelines for Genetic Algorithm with Penalty Functions." *Proceeding of the Third International Conference on Genetic Algorithms*, George Mason University, 191-197.

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

Statkinov, and Matusov. (1995). *Multicriteria Optimization and Engineering*, Capman and Hall.

TenHuisen, M. L., and Wiecek, M. M. (1996). "An Augmented Lagrangian Scalarization for Multiple Objective Programming." Advances in Multiple Objective and Goal Programming, Springer, 151-159.

Zadeh, L. A. (1963). "Optimality and non-scalar-valued performance criteria." *IEEE Transactions on Automatic Control*(AC-8), 59- 60.