An Adaptive Hyperplane Approach for Multiple Objective Optimization Problems with Complex Constraints

Runwei Cheng

Department of Industrial and Information Systems Engineering
Ashikaga Institute of Technology
Room 103, Hakusan 4-34-10
Bunkyo-ku, Tokyo 112-0001, Japan
runweicheng@hotmail.com
Phone/Fax: +81(3)3947-8288

Mitsuo Gen

Department of Industrial and Information Systems Engineering
Ashikaga Institute of Technology
Omae-chao 268,
Ashikaga 326-8558, Japan
gen@ahsitech.ac.jp
Phone: +81(284)62-0605
Fax: +81(284)64-1071

Shmuel S. Oren

Department of Industrial Engineering and Operations Research
University of California
Berkeley, CA 94720 USA
oren@ieor.berkeley.edu
Phone: +1(510)642-4989
Fax: +1(510)642-1403

Abstract

Many real-world decision problems involve multiple and conflicting objectives, which need to be optimized simultaneously while respecting various complex constraints. In this paper, we investigated how to solve these kinds of problems by using genetic algorithms. A new fitness assignment method for multiple objective optimization problems - the adaptive hyperplane-based fitness assignment method - was proposed in order to give a genetic algorithm the search pressure toward the positive ideal point in the objective space. An adaptive penalty function was used together with the adaptive hyperplane method in order to let genetic search explore the optima through both feasible and infeasible areas in the solution space. A Pareto solution reserving method was incorporated into normal genetic algorithm loop in order to maintain a set of Pareto solutions during the evolutionary process.

1 INTRODUCTION

Optimization deals with the problems of seeking solutions over a set of possible choices to optimize certain criteria. If there is only one criterion to consider, it becomes to a single objective optimization problem, a type studied extensively for the past 50 years. If there is more than one criterion and they must be treated simultaneously, we have multiple objective optimization problems (Steuer, 1986). Multiple objective problems arise in the design, modeling, and planning of many complex real-world problems. Almost every important real-world decision problem involves multiple and conflicting objectives that need to be tackled while respecting various complex constraints, leading to overwhelming problem complexity. Multiple objective optimization problems have been receiving growing interest from researchers with various backgrounds since early 1960. During the last two decades, genetic algorithms have received considerable attention as a novel approach to multiple objective optimization problems, known as genetic multiobjective optimizations or evolutionary multiobjective optimization (Fonseca and Flemming, 1995). The growing researches on applying genetic algorithms to multiple objective optimization problems present a formidable theoretical and practical challenge to the mathematical community.

In this paper, we investigated how to solve these kinds of problems by using genetic algorithms. A new fitness assignment method for multiple objective optimization problems - the adaptive hyperplane-based fitness assignment method - was proposed in order to give a genetic algorithm the search pressure toward the positive ideal point in the objective space. An adaptive penalty function was used together with the adaptive hyperplane method in order to let genetic search explore the optima through both feasible and infeasible areas in the solution space. A Pareto solution reserving method was incorporated into normal genetic algorithm loop in order to maintain a set of Pareto solutions during the evolutionary process.

We have applied the hyperplane approach to several real world problems. The experimental results were very encouraging and showed that it can be easily applied to most real world multiobjective optimization problems.

2 MULTIOBJECTIVE OPTIMIZATION PROBLEM

Without loss of generality, a multiple objective optimization problem can be represented formally as follows:

$$\max \quad \{z_1 = f_1(\boldsymbol{x}), z_2 = f_2(\boldsymbol{x}), ..., z_q = f_q(\boldsymbol{x})\}$$
$$\text{s.t.} \quad g(\boldsymbol{x}) \le 0, \quad i = 1,2,..., m$$

where $\boldsymbol{x} \in R^n$ is a vector of n decision variables, $f_i(\boldsymbol{x})$ the ith objective function, and $g_i(\boldsymbol{x})$ the ith inequality constraint. The m functions form an area of feasible solutions denoted by the set S as follows:

$$S = \{\boldsymbol{x} \in R^n \mid g_i(\boldsymbol{x}) \le 0, i = 1,2,..., m, \boldsymbol{x} \ge 0\}$$

The set of images of all points in S forms the feasible region in criterion space, denoted by the set Z as follows:

$$Z = \{\boldsymbol{z} \in R^q \mid z_i = f_i(\boldsymbol{x}), i = 1,2,..., q, \boldsymbol{x} \in S\}$$

We sometime graph multiple objective optimizations in both decision space and criterion space.

There is a special point in the criterion space called an *ideal point* or a *positive ideal solution*, denoted by $\boldsymbol{z}^* = (z_1^*, z_2^*, ..., z_q^*)$, where $z_k^* = \sup\{f_k(\boldsymbol{x}) \mid \boldsymbol{x} \in S\}$. The point is called an ideal point because usually it is not attainable. Note that, individually, z_k^* may be attainable, but to find a point that can maximize each objective function $f_k(\boldsymbol{x})$ simultaneously is usually very difficult. In contradiction to the positive ideal point, a *negative ideal solution* is also defined to represent the pessimistic status in the criterion space, denoted by $\boldsymbol{z}^- = (z_1^-, z_2^-, ..., z_q^-)$, where $z_k^- = \sup\{f_k(\boldsymbol{x}) \mid \boldsymbol{x} \in S\}$. Similar to the positive ideal point, the negative ideal point is also usually not attainable. Individually, z_k^- may be attainable.

In principle, multiple objective optimization problems are very different from single objective optimization problems. For the single objective case, one attempts to obtain the best solution, which is absolutely superior to all other alternatives. In the case of multiple objectives, there does not necessarily exist such a solution that is the best with respect to all objectives because of incommensurability and conflict among objectives. A solution may be best in one objective but worst in other objectives. Therefore, there usually exists a set of solutions for the multiple objective cases, which cannot be simply compared with each other. For such solutions, called *nondominated* solutions or *Pareto optimal* solutions, no improvement in any objective function is possible without sacrificing at least one of the other objective functions. For a given nondominated point in the criterion space Z, its image point in the decision space S is called *efficient* or *noninferior*. A point in S is efficient if and only if its image in Z is nondominated.

In real decision making cases, we are usually asked to select one of those *nondominated* solutions as a final solution to a given problem. It is most likely, however, that we will be unable to settle for one of those solutions without providing additional preferences regarding various objectives. Therefore, how to make a final choice from those alternative solutions essentially depends on one's subjective preferences. Conceptually, the *preference* intends to give an order to the incomparable solutions within the efficient set by using ones' value judgments on objectives, The preference reflects either ones' tradeoffs among objectives or emphasis for some particular objectives according some prior knowledge to the problem. With a given preference, we can order the alternative solutions in the nondominated set, and then we can obtain a final solution, which is the usual outcome of a decision making process. Such a final solution is called *best-compromised* solution.

The general expectation for a decision making process can be either to obtain a compromised or preferred solution or to identify all nondominated solutions. Therefore, there are basically two kinds of techniques for constituting a solution method to multiple objective optimization problems: (1) *generating approaches* and (2) *preference-based approaches*. The generating approaches have been developed to identify an entire set of Pareto solutions or an approximation. Preference-based approaches attempt to obtain a comprised or preferred solution. If we have no prior knowledge for preference structure over objectives, we have to adopt the generating approach to examine all nondominated alternatives. If we have some ideas of the relative importance of objectives, we can quantify the preference. With the preference information, a compromised or preferred solution can be identified.

From the viewpoint of solution techniques, most traditional methods reduce multiple objectives into a single objective, and then solve the problem with mathematical programming tools. To utilize mathematical programming tools to solve a multiple objective problem, we first need to express our preference in terms of numbers, so that the larger the number, the stronger the preference. By using scalarization techniques, multiple objective optimization problems are usually transformed into a single objective or a sequence of single objective optimization problems; then traditional techniques can be adapted to solve the altered problems (Morris and Oren, 1980, Arbel and Oren, 1999). Famous methods among them are utility function approach, weighted-sums approach, and compromise approach.

3 FEATURES OF GENETIC SEARCH

The inherent characteristics of genetic algorithms demonstrate why genetic search may be well suited to multiple objective optimization problems. The basic feature of genetic algorithms is multiple directional and global searches by maintaining a population of potential solutions from generation to generation. The population-to-population approach is useful when exploring Pareto solutions.

Genetic algorithms do not have much mathematical requirements and can handle all types of objective functions and constraints. Because of their evolutionary nature, genetic algorithms can be used to search for solutions without regard to the specific inner workings of problems. Therefore, it is hoped that we can solve many more complex problems by using genetic algorithms.

Because genetic algorithms provide us a great flexibility to hybridize with conventional methods into their main framework, we can exploit the advantages of both genetic algorithms and conventional methods to establish much

more efficient implementations to multiple objective optimization problems.

4 FITNESS ASSIGNMENT MECHANISM

Genetic algorithms are essentially a type of metastrategy of solutions. When applying genetic algorithms to solve a given problem, it is necessary to refine each of their major components, such as encoding methods, recombination operators, fitness assignment, selections, constraints handling, and so on, to obtain an effective implementation to the given problem. Because multiple objective optimization problems are natural extensions of constrained and combinatorial optimization problems, many useful methods developed for constrained and combinatorial optimization problems during the past two decades are readily applicable. Therefore, when considering how to adapt genetic algorithms to multiple objective optimization problems, we just need to examine some special issues concerning the problems.

4.1 FITNESS ASSIGNMENT METHODS

One of special issues arising in solving multiple objective optimization problems by use of genetic algorithms is how to determine the fitness value of individuals according to multiple objectives. The fitness assignment mechanism has been studied extensively during the past decade and several methods have been suggested and tested. Roughly, these methods can be classified as follows: (1) vector evaluation method, (2) Pareto-based method, (3) weighted-sum method, (4) compromise method, and (5) goal programming method.

Perhaps, the first notable work to extend simple genetic algorithms to solve the multiple objective optimization problems is the vector evaluation method proposed by Schaffer (1985). Instead of using a scalar fitness measure to evaluate each chromosome, it uses a vector fitness measure to create the next population.

There are two kinds of Pareto-based methods: *Pareto ranking and Pareto tournament*. The Pareto ranking-based fitness assignment method was first suggested by Goldberg as a means of achieving equal reproductive potential for all Pareto individuals (Goldberg, 1989). It includes two major steps: (1) Sort the population based on Pareto ranking. (2) Assign selection probabilities to individuals according to the ranking. The Pareto tournament method was proposed by Horn, Nafpliotis, and Goldberg (1994). Instead of nondominated sorting and ranking selection method, a *niched Pareto* concept was used in tournament, where a Pareto solution with least number of individuals in its neighbor wins the competition.

The weighted-sum method takes its basic ideas from conventional multiobjective optimizations. It assigns weights to each objective function and combines the weighted objectives into a single objective function. Conceptually, it is simple to understand and easy to compute; but to make it work requires only a proper weighting vector. Therein, however, lies the difficulty. Once embedded in genetic algorithms, its weakness can be compensated for by the powers of population-based and evolutionary search. Several weight-adjust methods have been proposed in order to fully utilize the power of genetic search: (1) fixed weight approach, (2) random weight approach, and (3) adaptive weights (Gen and Cheng, 2000). Recently, this approach was combined with a spanning tree-based genetic algorithm applied to a multi-objective transportation problem (Gen and Li, 1999).

Cheng and Gen proposed the compromise method as a means to obtain a compromised solution instead of generating all Pareto solutions (Cheng and Gen, 1998). Its basic idea and techniques are borrowed from conventional multiple objective optimizations. The compromise approach identifies solutions that are closest to the ideal solution as determined by some measure of distance.

Goal programming is one of the powerful techniques for solving the multiobjective optimization problems. Gen, Liu and Ida investigated the application of genetic algorithms to solve nonlinear goal programming problems (Gen, Liu and Ida, 1996). Because lexicographic ordering among objectives is preferred in goal programming, individuals are sorted on the value of objectives in a lexicographic manner in their genetic algorithms. Individual fitness values are then assigned by interpolating from the best to the worst according to an exponential function.

According to how much preference information is incorporated into the fitness function, these approaches range from complete preference information given, as when combining objective functions directly or prioritizing them, to no preference information given, as with Pareto-based approaches. In addition, a notable feature is that given a rough preference, progressive refinement of the preference can be carried out by evolutionary search. The progressive refinement of preferences is like an interactive procedure often used in multiple objective optimization, where preferences are modified at each iteration by decision makers. What makes it unique is the refinement mechanism: The preference is refined gradually through the evolutionary search by some adaptive refinement mechanism, not by the intervention of decision makers at each iteration. Of course, an interactive procedure can also be embedded in genetic search to guide preference refining.

4.2 TWO BASIC APPROACHES

From the viewpoint of methodology, there are two basic approaches to multiple objective optimizations: generating approach and preference-based approach. Generating approaches are used to identify an entire set of Pareto solutions or an approximation, whereas preference-based approaches attempt to obtain a comprised or preferred solution. Conceptually, the vector evaluation approach, the Pareto ranking-based approach, and the random weighting approach are designed as the generating methods; the compromise approach, the adaptive weighting

approach, and the goal programming approach are designed as the preference-based approaches.

In multiobjective optimizations, generating and preference-based methods both exhibit their strengths and weaknesses. Generating techniques require decision makers to make a judgment by selecting from among entire Pareto solutions. For problems with more than three criteria, making a choice becomes very complicated, increasing in difficulty approximately exponentially with the number of criteria. Computational costs also increase rapidly with the number of criteria. In genetic multiobjective optimizations, the situation essentially does not change. In contrast, preference-based techniques seem not to put as great a burden on decision makers as used in multiobjective optimization. Because preferences can be refined gradually along with the evolutionary process, a rough preference can be made to work by evolutionary search.

A solution method can be designed as either a generating method for obtaining an entire set of Pareto solutions or a preference-based method for obtaining a preferred or compromised solution. In multiobjective optimizations, we cannot have a solution method which implements the two distinct ideas into one solution procedure, but in genetic multiobjective optimization, we can.

4.3 THE CONCEPT OF PARETO SOLUTION

In a strict sense, the term of Pareto solution used in genetic algorithms has a different meaning as used in a conventional way. In the original definition, a point is said to be a Pareto solution if and only if it is a nondominated point with respect to all points in the criterion space for a given problem. In genetic algorithms, Pareto solutions are identified at each generation. Because a population at each generation contains only partial solutions of the original problem, a Pareto solution has its meaning only with respect to all solutions currently examined. A nondominated solution in one generation may become dominated by a new solution emerged in a later generation. Therefore, for a given generation of genetic algorithms, a Pareto solution obtained in that generation may be a true Pareto solution to the problem, or it may not be. There is no guarantee that a genetic algorithm certainly produces Pareto solutions to a given problem. But a genetic algorithm will provide a better approximation of Pareto solutions.

How to maintain a set of nondominated individuals during the evolutionary process is a special issue for multiple objective optimization problems. Basically, there are two different ways to handling Pareto solutions, which lead to two different overall structures of genetic algorithms implementations: (1) preserving Pareto solutions separately from population pool and (2) without preserving mechanisms.

In most existing methods, Pareto solutions are identified at each generation and used only to calculate fitness values or ranks for each chromosome. No mechanism is provided to guarantee that Pareto solutions generated during the evolutionary process enter the next generation. In other words, some Pareto solutions may get lost during the evolutionary process. To avoid such sampling errors, a preserving mechanism for Pareto solutions has been suggested by many researchers (Gen and Cheng, 2000). A special pool for preserving Pareto solutions is added onto the basic structure of genetic algorithms. At each generation, the set of Pareto solutions is updated by deleting all dominated solutions and adding all newly generated Pareto solutions. The overall structure with Pareto preserving is given as follows:

```
Procedure: Pareto genetic algorithms
begin
   t = 0
   Initialize P(t);
   Objectives P(t);
   Pareto E(t);
   Fitness P(t);
   while (not termination condition) do
   begin
      Crossover P(t);
      Mutation P(t);
      Objective P(t);
      Update Pareto E(t);
      Fitness P(t);
      Selection P(t+1) from P(t);
      t = t+1;
   end
end
```

Without a preserving mechanism, Pareto solutions can be gathered only from the last generation. If the method used has a tendency of speciation as mentioned by Schaffer (1985), the entire population will converge toward the individual optimum regions after a large number of generations. The preserving mechanism is, to a certain extent, helpful to minimize speciation through a Pareto preserving procedure at each generation.

5 ADAPTIVE HYPERPLANE APPROACH

Adaptive hyperplane approach belongs to the type of adaptive weight method. It constructs a hyperplane by some special points in objective space in each generation and fitness values of individuals are then calculated based on the hyperplane. The hyperplane is adjusted adaptively based on the current generation to obtain a search pressure toward the positive ideal point in the objective space.

5.1 ADAPTIVE WEIGHT

The basic idea of assigning weights to each objective function and combining them into a single-objective function was firstly proposed by Zadeh (Zadeh, 1963). The weighted-sums method can be represented as follows:

$$z(\boldsymbol{x}) = \sum_{k=1}^{q} w_k \, f_k(\boldsymbol{x})$$

The weight w_k can be interpreted as the relative emphasis or worth of that objective when compared to the other

objectives. In the other words, the weight can be interpreted as representing our preference over objectives. Therefore, an optimal solution to a given problem relates to a particular preference structure. Moreover, the optimal solution to the problem is a nondominated solution provided that all the weights are positive. Because of the numerical ordering by the weighted-sum function, there is no ambiguity in preference comparison. For any two points, either one is better, worse or equivalent to another. Exactly one of the three cases must happen. There are no such things as indefinite sets in the preference structure.

The adaptive weights approach proposed in this paper adjusts weights adaptively according to the current generation in order to obtain a search pressure toward to the positive ideal point. This approach is designed for genetic algorithms in order to fully utilize the power of genetic search. It works only due to the nature of population-based evolutionary search of genetic algorithms.

Consider the maximization problem with q objectives described in Section 2.

For the solutions examined in each generation, we define two extreme points: the maximum extreme point z^+ and the minimum extreme point z^- in criteria space as follows:

$$z^+ = \{z_1^{\max}, z_2^{\max}, ..., z_q^{\max}\}$$

$$z^- = \{z_1^{\min}, z_2^{\min}, ..., z_q^{\min}\}$$

where $z_k^{\max}$ and $z_k^{\min}$ are the maximal value and minimal value for objective k in current population. Let P denote the set of current population. For a given individual $\boldsymbol{x}$, the maximal value and minimal value for each objective are defined as the follows:

$$z_k^{\max} = \max\{f_k(\boldsymbol{x}) \mid \boldsymbol{x} \in S\}, \quad k = 1,2,...,q$$

$$z_k^{\min} = \min\{f_k(\boldsymbol{x}) \mid \boldsymbol{x} \in S\}, \quad k = 1,2,...,q$$

The hyper-parallelogram defined by the two extreme points is a minimal hyper-parallelogram containing all current solutions. The two extreme points are renewed at each generation. The maximum extreme point will gradually approximate to the positive ideal point. The adaptive weight for objective k is calculated by the following equation:

$$w_k = \frac{1}{z_k^{\max} - z_k^{\min}} \qquad k = 1,2,...,q$$

For a given individual $\boldsymbol{x}$, the weighted-sum objective function is given by the following equation:

$$z(\boldsymbol{x}) = \sum_{k=1}^{q} w_k (z_k - z_k^{\min}) = \sum_{k=1}^{q} \frac{f_k(\boldsymbol{x}) - z_k^{\min}}{z_k^{\max} - z_k^{\min}}$$

For the cases that all objective functions takes only positive value, the equation can be simplified as follows:

$$z(\boldsymbol{x}) = \sum_{k=1}^{q} w_k z_k = \sum_{k=1}^{q} \frac{f_k(\boldsymbol{x})}{z_k^{\max} - z_k^{\min}}$$

As the extreme points are renewed at each generation, the weights are renewed accordingly. Equation above is a hyperplane defined by the following extreme point in current solutions:

$$\{z_1^{\max}, z_2^{\min}, ..., z_k^{\min}, ..., z_q^{\min}\}$$

$$\{z_1^{\min}, z_2^{\max}, ..., z_k^{\min}, ..., z_q^{\min}\}$$

.....

$$\{z_1^{\min}, z_2^{\min}, ..., z_k^{\max}, ..., z_q^{\min}\}$$

.....

$$\{z_1^{\min}, z_2^{\min}, ..., z_k^{\min}, ..., z_q^{\max}\}$$

The hyperplane divides the criteria space Z into two half-spaces: one half space contains the positive ideal point, denoted as Z^+, and the other half space contains the negative ideal point, denoted as Z^-. All Pareto solutions examined lie in the space Z^+, and all points lying in the Z^+ have larger fitness values than those in the space Z^-. As the maximum extreme point approximates to its possible largest value along with the evolutionary progress, the hyperplane will gradually approach to the positive ideal point. Therefore, the adaptive weight method can readjust its weights according to the current population in order to obtain a search pressure toward to the positive ideal point.

Let us see an example of bicriteria maximization problem given below:

$$\max \quad \{z_1 = f_1(\boldsymbol{x}),\ z_2 = f_2(\boldsymbol{x})\}$$

$$\text{s.t.} \quad g(\boldsymbol{x}) \le 0, \quad i = 1,2,..., m$$

For a given generation, two extreme points are identified as:

$$z_1^{\max} = \max\{z_1(\boldsymbol{x}^j), j = 1,2,.....,pop\_size\}$$

$$z_2^{\max} = \max\{z_2(\boldsymbol{x}^j), j = 1,2,.....,pop\_size\}$$

$$z_1^{\min} = \min\{z_1(\boldsymbol{x}^j), j = 1,2,.....,pop\_size\}$$

$$z_2^{\min} = \min\{z_2(\boldsymbol{x}^j), j = 1,2,.....,pop\_size\}$$

and the adaptive weights are calculated as follows:

$$w_1 = \frac{1}{z_1^{\max} - z_1^{\min}} \quad \text{and} \quad w_2 = \frac{1}{z_2^{\max} - z_2^{\min}}$$

The weighted-sum objective function is then given by the following equation:

$$z(\boldsymbol{x}) = w_1 z_1 + w_2 z_2 = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x})$$

It is an adaptive moving line defined by the extreme points $(z_1^{\max}, z_2^{\min})$ and $(z_1^{\min}, z_2^{\max})$ as shown in Figure 1. The rectangle defined by the extreme points $(z_1^{\max}, z_2^{\max})$ and $(z_1^{\min}, z_2^{\min})$ is the minimal rectangle containing all current solutions.

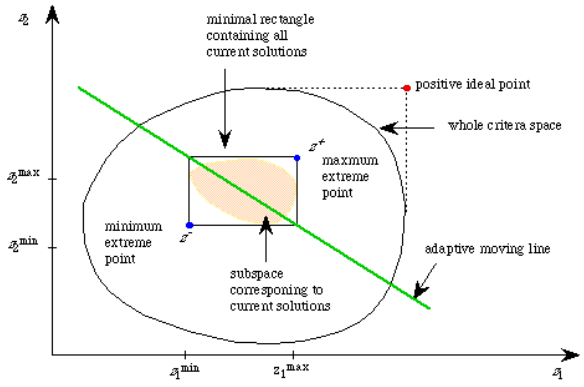

Figure 1: Adaptive weights and adaptive hyperplane for a bicriteria case.

Conceptually, the weighted-sum approach can be viewed as an extension of methods used in multiobjective optimizaiton to genetic algorithms. It assigns weights to each objective function and combines the weighted objectives into a single objective function. In fact, the weighted-sum approaches used in genetic algorithms are very different in nature from that in conventional multiobjective optimizations. In multiobjective optimization, the weighted-sum approach is used to obtain a compromise solution. To make the method work, all that is needed is a good weighting vector. It is usually very difficult to determine a set of appropriate weights for a given problem. In genetic algorithms, the weighted-sum approach is primarily used to adjust genetic search toward to the Pareto frontier. Weights are readjusted adaptively along with the evolutionary process. Therefore, a good weighting vector is not a mandatory precondition to making genetic algorithms work. In addition, the drawbacks exhibited in the multiobjective optimization can be compensated by the powers of population-based search and evolutionary search.

5.2 ADAPTIVE PENALTY FUNCTION

Penalty technique is perhaps the most common technique used to handle infeasible solutions in genetic algorithms for constrained optimization problems. In essence, this technique transforms the constrained problem into an unconstrained problem by penalizing infeasible solutions, in which a penalty term is added to the objective function for any violation of the constraints (Gen and Cheng, 1997).

The basic idea of the penalty technique is borrowed from conventional optimization. In conventional optimization, the penalty technique is used to generate a sequence of infeasible points whose limit is the optimal solution to the original problem. The major concern is how to choose a proper value of penalty so as to hasten convergence and avoid premature termination. In genetic algorithms, the penalty technique is used to keep a certain amount of infeasible solutions in each generation so as to enforce genetic search toward an optimal solution from both sides of feasible and infeasible regions. The major concern is how to determine the penalty term so as to strike a balance between the information preservation (keeping some infeasible solutions) and the selective pressure (rejecting some infeasible solutions), and both under-penalty and over-penalty (Gen and Cheng, 1996a).

Given an individual $\boldsymbol{x}$ in current population $P(t)$, the adaptive penalty function is constructed as follows:

$$p(\boldsymbol{x}) = 1 - \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\Delta b_i(\boldsymbol{x})}{\Delta b_i^{\max}}\right)^{a}$$

where

$$\Delta b_i(\boldsymbol{x}) = \max\{0, g_i(\boldsymbol{x}) - b_i\}$$
$$\Delta b_i^{\max} = \max\{e, \Delta b_i(\boldsymbol{x}) \mid \boldsymbol{x} \in P(t)\}$$

where $\Delta b_i(\boldsymbol{x})$ is the value of violation for constraint i for the ith chromosome, $\Delta b_i^{\max}$ is the maximum violation for constraint i among current population, and e is a small positive number used to avoid zero-division. For highly constrained optimization problems, infeasible solutions take a relatively big portion among the population at each generation. The penalty approach adjusts the ratio of penalties adaptively at each generation in order to make a balance between the preservation of information and the pressure for infeasibility to avoid over-penalty. With the penalty function, the fitness function then takes the following form;

$$\text{eval}(\boldsymbol{x}) = z(\boldsymbol{x})p(\boldsymbol{x})$$

6 APPLICATION TO SOME PROBLEMS

6.1 INTERVAL PROGRAMMING PROBLEM

Let us consider the following example with an interval objective function:

$$\begin{aligned}
\max\ & z(\boldsymbol{x}) = [15,17]x_1 + [15,20]x_2 + [10,30]x_3 \\
\text{s.t.}\ & g_1(\boldsymbol{x}) = x_1 + x_2 + x_3 \le 30 \\
& g_2(\boldsymbol{x}) = x_1 + 2x_2 + x_3 \le 40 \\
& g_3(\boldsymbol{x}) = x_1 + 4x_3 \le 60 \\
& x_j \ge 0,\ j = 1,2,3.\ \text{integer}
\end{aligned}$$

The problem can be transformed into the following bicriteria programming problem:

$$\begin{aligned}
\max\ & z^L(\boldsymbol{x}) = 15x_1 + 15x_2 + 10x_3 \\
\max\ & z^C(\boldsymbol{x}) = 16x_1 + 17.5x_2 + 20x_3 \\
\text{s.t.}\ & g_1(\boldsymbol{x}) = x_1 + x_2 + x_3 \le 30 \\
& g_2(\boldsymbol{x}) = x_1 + 2x_2 + x_3 \le 40 \\
& g_3(\boldsymbol{x}) = x_1 + 4x_3 \le 60 \\
& x_j \ge 0,\ j = 1,2,3.\ \text{integer}
\end{aligned}$$

The integer vector was used as the chromosome representation. Uniform crossover and random perturbation mutation were used as genetic operators. The fitness value of each individual was calculated by the hyperplane method. The Pareto solutions found by the proposed method are shown in Figure 2 (Gen and Cheng, 1996b).

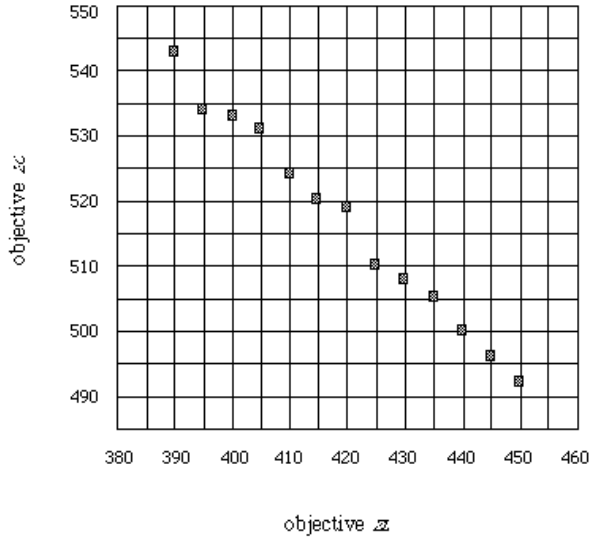

Figure 2: Pareto Solutions to the Interval Programming Problem

6.2 BICRITERIA LINEAR TRANSPORTATION PROBLEM

Consider following bicriteria linear transportation problem given by Aneja and Nair (1978):

$$\max \; z_1 = x_{11} + 2x_{12} + 7x_{13} + 7x_{14} + x_{21} + 9x_{22} + 3x_{23} + 4x_{24} + 8x_{31} + 9x_{32} + 4x_{33} + 6x_{34}$$

$$\max \; z_2 = 4x_{11} + 4x_{12} + 3x_{13} + 4x_{14} + 5x_{21} + 8x_{22} + 9x_{23} + 10x_{24} + 6x_{31} + 2x_{32} + 5x_{33} + x_{34}$$

$$\text{s.t.} \; \sum_{j=1}^{4} x_{1j} = 8, \; \sum_{j=1}^{4} x_{2j} = 19, \; \sum_{j=1}^{4} x_{3j} = 17$$

$$\sum_{i=1}^{3} x_{i1} = 11, \; \sum_{i=1}^{3} x_{i2} = 3, \; \sum_{i=1}^{3} x_{i3} = 14, \; \sum_{i=1}^{3} x_{i4} = 16$$

$$x_{ij} \geq 0, \quad \text{for all } i \text{ and } j$$

The allocation matrix was used as the chromosome representation. The special crossover operation and mutation operation proposed by Vignaux and Michalewicz (1991) were adopted. The fitness value for each individual was determined by the adaptive hyperplane method. The Pareto solutions found by the proposed method are depicted in Figure 3 (Yang and Gen, 1994). From it we can know that the proposed method is much more effective than the method of Aneja and Nair.

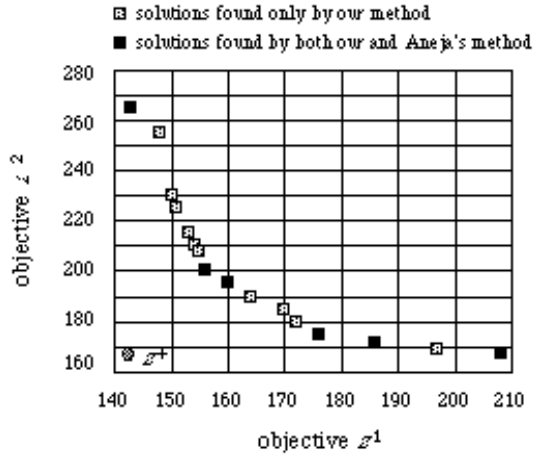

Figure 3: Pareto solutions obtained by our method and Aneja's method

6.3 BICRITERIA MINIMUM SPANNING TREE PROBLEM

Consider the minimum spanning tree problem, where each edge has two associated positive real numbers. Then it can be formulated as the following bicriteria optimization problem (Zhou and Gen, 1999).

$$\max \left\{ z_1(\boldsymbol{x}) = \sum_{i=1}^{m} w_{1i} x_i, \;\; z_2(\boldsymbol{x}) = \sum_{i=1}^{m} w_{2i} x_i \right\}$$

$$\text{s.t.} \quad \boldsymbol{x} \in \mathrm{T}$$

where $\boldsymbol{x}$ is a binary decision variable defined as:

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

and T denotes the set of all spanning trees corresponding to a given problem.

The Prüfer number was adopted as the tree encoding. It is capable for representing all possible spanning trees. Uniform crossover and perturbation mutation were used as genetic operations, and the adaptive hyperplane method was used to determine fitness values for each tree. The Pareto solutions found by the proposed method are depicted in Figure 4 (Zhou and Gen, 1999).

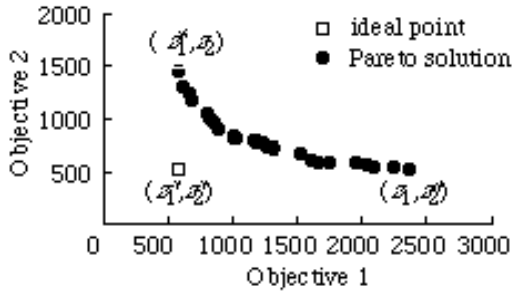

Figure 4: Pareto solutions found by the proposed method for a 50-vertex instance.

7 CONCLUSIONS

In general, multiple objective optimization problems are too complex to be solved easily. There are two kinds of difficulties associated with the problem solving that need to be distinguished: (1) difficulties inherent in problems and (2) difficulties related to solution techniques. The most profound drawback of many conventional methods is that they are very sensitive toward to the value of weights, or the prescribed order of objectives, or the shape of utility functions. In essence, such kinds of difficulties are caused by solution techniques but not the problem itself. However, in genetic multiobjective optimizations, the drawbacks exhibited in conventional methods can be compensated by the powers of population-based search and evolutionary search.

In this paper, we proposed a new fitness assignment method for multiple objective optimization problems: the adaptive hyperplane method. It is designed for genetic algorithms to fully utilize the power of genetic search. In this method, weights are adjusted adaptively from generation to generation. It provides a search pressure toward to the Pareto frontier. An adaptive penalty function was used together with the adaptive hyperplane method in order to let genetic search explore the optima through both feasible and infeasible areas in the solution space. A Pareto solution reserving method was incorporated into a normal genetic algorithm loop in order to maintain a set of Pareto solutions during the evolutionary process. The proposed approach has been applied to several real world decision-making problems with multiple conflict objectives and complex constraints. The experimental results are very encouraging and show that the method can be easily applied to multiple objective optimization problems.

Acknowledgments

This research work was supported by the International Scientific Research Program, the Grant-in-Aid for Scientific Research (No. 10044173: 1998.4—2001.3) by the Ministry of Education, Science and Culture, the Japanese Government.

References

A. Arbel and S. Oren (1999). Using approximate gradients in developing interactive interior primal-dual multiobjective linear programming algorithm, *European Journal of Operational Research:* **89**:202-211.

Y. Aneja and K. Nair (1978). Bicriteria transportation problem. *Management Sciences*: 25: 73-78.

R. Cheng and M. Gen (1998). Compromise approach-based genetic algorithms for bicriterion shortest path problems. Technical report, *Ashikaga Institute of Technology*, Japan.

C. Fonseca and P. Fleming (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* **3**(1):1-16.

M. Gen and R. Cheng (1996a). A survey of penalty techniques in genetic algorithms. *Proceedings of 1996 IEEE International Conference on Evolutionary Computation*: 804-809.

M. Gen and R. Cheng (1996b). Interval programming using genetic algorithms. *Intelligent Automation and Control*: 4: 243-248.

M. Gen and R. Chang (1997). *Genetic Algorithms & Engineering Design*. New York: John Wiley & Sons.

M. Gen and R. Chang (2000). *Genetic Algorithms & Engineering Optimization*. New York: John Wiley & Sons.

M. Gen and Y. Li (1999). Spanning tree-based genetic algorithms for bicriteria fixed charge transportation problem. *Proceeding of the Congress on Evolutionary Computation*: 2265-2271.

M. Gen, B. Liu and K. Ida (1996). Evolution program for deterministic and stochastic optimizations. *European Journal of Operational Research*: **94**(3):618-625.

D. Goldberg (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Reading: Addison-Wesley.

J. Horn, N. Nafpliotis and D. Goldberg (1994). A niched Pareto genetic algorithm for multiobjective optimization. *Proceedings of First IEEE Conference on Evolutionary Computation*: 82-87.

P. Morris and S. Oren (1980). Multiattribute Decision making by sequential resource allocation, *Operations Research*, **28**(1): 233-252.

J. Schaffer (1985). Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms* 93-100.

R. E. Steuer (1986). *Multiple criteria optimization: theory, computation, and application.* New York, John Wiley & Sons.

G. Vignaux and Z. Michalewicz (1991). A Genetic Algorithm for the Linear Transportation Problem. *IEEE Transactions on Systems, Man, and Cybernetics*: **21**:445-452.

X. Yang and M. Gen (1994). Evolution program for bicriteria transportation problem. *Proceedings of the 16th International Conference on Computer and Industrial Engineering*: 451-454.

L. Zadeh (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*: **8**(59).

D. Zheng, M. Gen and R. Cheng (1999). Multiobjective optimization using genetic algorithms. *Engineering Valuation and Cost Analysis*: **3**:303-310.

G. Zhou and M. Gen (1999). Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*: **114**:141-152.