Two Multi-Objective Genetic Algorithms for Finding Optimum Design of an I-Beam

Ali Khazaee¹², Hossein Miar Naimi²

¹Department of Electrical Engineering, Islamic Azad University, Bojnourd Branch, Bojnourd, Iran
²Department of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran

E-mail: khazaeeali@yahoo.com, h_miare@yahoo.com

Received April 15, 2011; revised May 10, 2011; accepted June 1, 2011

Abstract

Many engineering design problems are characterized by presence of several conflicting objectives. This requires efficient search of the feasible design region for optimal solutions which simultaneously satisfy multiple design objectives. Genetic algorithm optimization (GAO) is a powerful search technique with faster convergence rates than traditional evolutionary algorithms. This paper applies two GAO-based approaches to multi-objective engineering design and finds design variables through the feasible space. To demonstrate the utility of the proposed methods, the multi-objective design of an I-beam will be presented.

Keywords: Genetic Algorithm, Multi-Objective, I-Beam, Optimization

1. Introduction

Many engineering design problems are characterized by presence of several often conflicting and incommensurable design objectives [1,2]. This raises the question how to effectively search the feasible design region for optimal solutions which simultaneously satisfy multiple design objectives.

Population-based optimization methods such as evolutionary algorithms (EA) have become increasingly popular for solving difficult optimization problems [3,4]. In addition to their insensitivity to assumptions of continuity and differentiability of the underlying objective functions, the success of EAs can be attributed to their ability to efficiently explore and exploit various portions of the pareto-front simultaneously [5,6].

A relatively recent arrival among optimizing algorithms is called genetic algorithm optimization (GAO). The basic element of a GA is the artificial individual. Similar to a natural individual, an artificial individual consists of a chromosome and a fitness value. The fitness of an individual describes how well an individual is adapted to the nature. It determines the individual’s likelihood for survival and mating. Every changing of the chromosome leads to a changing of the Individual’s fitness. GAO is a powerful stochastic search technique with faster convergence rates than the traditional evolutionary algorithms. A particular application area of interest for GAO-based methods is multi-objective optimization (MO) in engineering design environments [3].

Basically GAO and other evolutionary algorithms like PSO or Ant colony optimization work with single-objective problems (SOPs). In SOPs fitness evaluation of GAO is the goodness of individual for optimization problem so individuals with best goodness have upper fitness. But at multi-objective problems (MOPs), we face with multiple functions so each individual have multiple goodness at each functions. Because of this, we must define new fitness evaluation method. Two most popular methods that proposed for this are weighted aggregation and dominant Pareto fronts. We most convert multiple fitness of each individual for each objective into one fitness for each individual. Simplest way is aggregation of fitness of each objective. Weight of each objective’s fitness can pre-defined by user with respect to their importance. Fitness evaluation of non-inferior individuals in a population is accomplished by performing non-dominated sorting and full-factorial experiments on population members followed by the weighted aggregation technique. Fitter designs are selected to serve as archetypical design cases allowing other designs to move towards the fitter designs by emulating their behavior. But this is not a good method; another method is based on dominance that will propose at follows.

Thus far, we have discussed the motivation and a brief overview of the presented work. The remainder of the
paper is organized as follows. Sections 2 and 3 provide brief descriptions of multi-objective and genetic algorithm optimization methods, respectively. Section 4 introduces dominant-based GA methodologies devised specifically for multi-objective robust engineering design. Section 5 is the application of the mentioned two main methods to pareto-optimum design of an I-beam. And finally, Section 6 is the summary and conclusions.

2. Multi-Objective Optimization

Multi-objective optimization (MO) is a methodology for finding optimal solutions to multivariate problems with multiple, often conflicting, objectives [1].

The main goal of MO is to find the optimum input parameter vector which results in some desired combination of maximization, minimization, or nominalization of the involved product or process responses. Mathematically, MO attempts to optimize the $p$-dimensional vector function $F(X) = [f_1(X), \ldots, f_p(X)]^T$ where $X = [x_1, \ldots, x_n]^T$ is an $n$-dimensional vector of design variables and $p$ is the number of design objectives for a product. The problem can be stated formally as follows ($l$ is the inequality and $m$ is the equality constraints) [2]:

Minimize/Maximize:

$$f_i(X) \quad \text{for } i = 1, \ldots, p$$

Subject to:

$$g_j(X) \leq 0 \quad \text{for } j = 1, \ldots, l$$

$$h_k(X) = 0 \quad \text{for } k = 1, \ldots, m$$

Since MO problems often have conflicting objectives, it is virtually impossible to find any one ideal solution. Instead, MO produces Pareto-optimal solutions. A design vector is Pareto-optimal if there is no other design vector that optimizes one criterion without causing the simultaneous degradation of at least one other criterion.

Without loss of generality, it can be stated that the goal is to maximize $p$ objective responses considering the following key definitions [1, 2]:

**Definition 1.** A vector $u = (u_1, \ldots, u_p)$ is said to be inferior to (dominated by) vector $v = (v_1, \ldots, v_p)$ if

$$\forall i \in \{1, \ldots, p\}, v_i \geq u_i \land \exists i \in \{1, \ldots, p\} | v_i > u_i$$

**Definition 2.** Vectors $u$ and $v$ are said to be non-inferior to each other if neither $u$ is inferior to $v$ nor $v$ is inferior to $u$.

The main challenge of MO is to develop efficient algorithms that can quickly converge to well-spread Pareto-fronts in the feasible, non-dominated design regions. This and other pertinent issues relating to the specifics of MO can be found elsewhere [1,3,4].

3. Genetic Algorithm Optimization

Genetic algorithms [7,8] employ metaphor from biology and genetics to iteratively evolve a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved and is composed of a fixed number of genes. The number of possible values of each gene is called the cardinality of the gene. Figure 1 illustrates the operation of a general genetic algorithm. The operation starts from an initial population of randomly generated individuals. Then the population is evolved for a number of generations and the qualities of the individuals are gradually improved. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, i.e., selection, crossover, and mutation. First, a number of best-fit individuals are selected based on a user-defined fitness function. The remaining individuals are discarded. Next, a number of individuals are selected and paired with each other. Each individual pair produces one offspring by partially exchanging their genes around one or more randomly selected crossing points. At the end, a certain number of individuals are selected and the mutation operations are applied, i.e., a randomly selected gene of an individual abruptly changes its value [9].

4. Evolutionary Multi-Objective Optimization (Methodologies)

An evolutionary algorithm (EA) is a stochastic optimization algorithm that simulates the process of natural evo-
Thus, an EA operates on a set of candidate solutions, which are subsequently modified by simplified implementations of the two basic principles of evolution: selection and variation. Selection represents the competition for resources among living beings. Some are better than others and more likely to survive and transmit their genetic information. A stochastic selection process simulates natural selection. Each solution is given a chance to reproduce a certain number of times, dependent on their quality, which is assessed by assigning a fitness value to each individual. The other principle, variation, imitates the natural capability of creating new living beings by means of recombination and mutation (see Figure 2). Recombination and mutation are typically the variation operators. We use two methods for optimizing multi-objective problems:

4.1. Evaluation of Individuals in Aggregation Method

In a SOP (Figure 2(top)), Task2 obtains a scalar value (fitness) for each individual (candidate solution) by evaluating a single function. In MOPs (Figure 2(bottom)) Task2 has two subtasks: Task2a obtains a vector of scalar values (vector fitness) for each individual by evaluating the set of objective functions: the dimension of vector fitness components. Assessing the fitness of an individual into a scalar value (fitness) using some specific method of weighted aggregation [1] was used to compute the overall normalized fitness for the kth individual point as:

\[ t_k = \sum w_i f_i \]

Where, \( w_i \) is the weight of the i\(^{th} \) design objective and represents the importance of the associated design objective to the designer (\( \sum w_i = 1 \)). Thus, the overall fitness vector for a population represented by n individual points would be the vector \( T = [t_1, \cdots, t_n] \), and the non-dominated individual point with the highest \( t \) value would be considered the ‘best’ individual point.

4.2. Create a Set of Pareto Optima for a Multi-Objective Minimization (Pareto Optimality)

You see that for converting a vector of fitness to a scalar fitness we must specify their importance, i.e. there is a vector of objectives, \( F(x) = [F_1(x), F_2(x), \cdots F_m(x)] \), that must be traded off in some way. The relative importance of these objectives is not generally known until the system best capabilities are determined and tradeoffs between the objectives fully understood. As the number of objectives increases, tradeoffs are likely to become complex and less easily quantified. The designer must rely on his or her intuition and ability to express preferences throughout the optimization cycle. Thus, requirements for a multi-objective design strategy must enable a natural problem formulation to be expressed, and be able to solve the problem and enter preferences into a nu-
merically tractable and realistic design problem. So some other methods is presented to satisfy this requirements that we briefly express some of them at follows.

Multi-objective optimization is concerned with the minimization of a vector of objectives \( F(x) \) that can be the subject of a number of constraints or bounds:

\[
\min F(x), \ x \in \mathbb{R}^n \quad \text{subject to:}
\]

\[
G_i(x) = 0, i = 1, \cdots, k_i;
\]

\[
G_i(x) \leq 0, i = k_i + 1, \cdots, k; 1 \leq x \leq u
\]

Note that because \( F(x) \) is a vector, if any of the components of \( F(x) \) are competing, there is no unique solution to this problem. Instead, the concept of non-inferiority [10,11] (also called Pareto optimality [12,13]) must be used to characterize the objectives. A non-inferior solution is one in which an improvement in one objective requires a degradation of another. To define this concept more precisely, consider a feasible region \( \Omega \), in the parameter space. \( x \) is an element of the n-dimensional real numbers that satisfies all the constraints, i.e.,

\[
\Omega = \{ X \in \mathbb{R}^n \}, \ \text{subject to}
\]

\[
G_i(x) = 0, i = 1, \cdots, k_i;
\]

\[
G_i(x) \leq 0, i = k_i + 1, \cdots, k; 1 \leq x \leq u
\]

This allows definition of the corresponding feasible region for the objective function space \( \Lambda \):

\[
\Lambda = \{ y \in \mathbb{R}^m : y = F(x), x \in \Omega \}
\]

Evolutionary algorithms seem to be especially suited to multi-objective optimization because they are able to capture multiple Pareto-optimal solutions in a single run, and may exploit similarities of solutions by recombination. Indeed, some research suggests that multi-objective optimization might be an area where EAs perform better than other search strategies. The considerable amount of research related to MOEAs currently reported in the literature is evidence of present interest in this subject.

In MOPs, it is necessary to find not one but several solutions, in order to determine the entire Pareto front. Nevertheless, due to stochastic errors associated with the evolutionary operators, EAs can converge to a single solution [Goldberg89]. There exist literatures several methods, called niching techniques [Sareni98], to preserve diversity in the population, in order to converge to different solutions. These techniques can also be applied to MOP.

Pareto-based fitness assignment in a genetic algorithm (GA) was first proposed by Goldberg [Goldberg89]. The basic idea is to find a set of Pareto non-dominated individuals in the population. These individuals are then assigned the highest rank and eliminated from further competition. Then, another set of Pareto non-dominated individuals are determined from the remaining population and are assigned the next highest rank. This process continues until the whole population is suitably ranked. Goldberg also suggested the use of a niching technique to keep the GA from converging to a single point on the front.

The Non-dominated Sorting Genetic Algorithm (NSGA) [Srinivas93] uses several layers of ranked individuals. Before selection is performed, the population is ranked on the basis of non-domination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, those so classified are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population have been classified. Then a stochastic remainder proportionate selection is used, followed by the usual cross and mutation operators.

Fonseca and Fleming [Fonseca93] have proposed an algorithm called Multiple Objective Genetic Algorithm (MOGA) where the rank of each individual is obtained from the number of individuals in the current population that dominate it. Thus, if at generation \( t \), an individual \( x_i \) is dominated by \( p_i \) individuals, its current rank can be given by:

\[
\text{Rank (} x_i, \ t \text{)} = 1 + p_i \ (t)
\]

All non-dominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface. In this way, the fitness assignment is performed by the following steps:

1) Sort population according to the rank of the individuals.

2) Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank \( n \leq N \)), according to a function (not necessarily linear).

3) Average the fitness of individuals with the same rank, so that all of them will be sampled at the same rate. Sharing on the objective function values is carried out to distribute population over the Pareto-optimal region.

In their Niched Pareto Genetic Algorithm (NPGA), Horn and Nafpliotis [Horn93] proposed a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of other individuals (usually about 10) in the population are used to help determine dominance. Whether the competitors are dominated or non-dominated, the result is decided through fitness sharing.

In [Zitzler99], it was clearly shown that elitism helps
5. Optimum Design of an I-Beam

This section presents the MO problem of the design of an I-beam, which has previously been approached by several different types of optimization methods \([2,3,7]\).

Assuming that the I-beam in Figure 3 is subject to maximal bending forces of \(P = 600\) kN and \(Q = 50\) kN at the mid span, the objective of the design is to find the optimum dimensions of the beam, \(X^* = [x_1, x_2, x_3, x_4]^T\), such that the cross section area (\(f_1\) in cm\(^2\)) and static deflection of the beam (\(f_2\) in cm) are both minimized subject to the constraint that the beam’s bending stress (\(f_3\)) does not exceed 16 kN/cm\(^2\). The geometric side constraints are \(10 \leq x_1 \leq 80\), \(10 \leq x_2 \leq 50\), \(0.9 \leq x_3\) and \(x_4 \leq 5\) (all in cm).

The MO design of the I-beam problem can mathematically be stated as follows. Find \(X^*\) which minimizes \(F(X^*) = [f_1(X^*), f_2(X^*)]^T\) where:

\[
f_1(X) = 2x_2x_4 + x_1(x_1 - 2x_4) \tag{3}
\]

\[
f_2(X) = \frac{60000}{x_1 (x_1 - 2x_4)} + 2x_2x_4 \left[ 4x_2^2 + 3x_1(x_1 - 2x_4) \right] \tag{4}
\]

Subject to the bending stress constraint:

\[
f_3(X) = \frac{180000x_1}{x_1 (x_1 - 2x_4)} + 2x_2x_4 \left[ 4x_2^2 + 3x_1(x_1 - 2x_4) \right] + \frac{15000x_2}{(x_1 - 2x_4)x_3^3 + 2x_4x_2^2} \leq 16 \tag{5}
\]

Given the boundaries of the feasible design region, a linear search of the problem space can determine which extrema in the objectives space are simultaneously attainable. The computed ranges of responses for the two objective functions and the bending stress constraint reveal that \(f_1\) is in conflict with \(f_2\) and that the ideal solution \(F^* = [25.38, 0.0059]^T\), where the two objectives are simultaneously minimized, can never be attained.

At first we apply first method to problem and compare results with earlier other optimization methods. To account for the possibility of statistical fluctuations that can
potentially produce misleading results, the GAO algorithm was tested over 3 statistically independent runs, each over 6 iterations with a population of 20 individuals containing. The weights for the three design objectives described in Equation’s 3, 4, and 5 were set according to their importance to 0.05, 0.35, and 0.60, respectively. Also we use Deb method with tournament selection and crossover fraction of 0.8.

Closer analysis of design solutions generated by a single population throughout three independent runs of the algorithm shows the consistency of each design objective converging. This convergence behavior is illustrated in Figure 4.

Table 1 depicts solutions reported for the same I-beam problem by several MO studies. Each of the first three rows in the table depicts three Pareto-optimal solutions generated by various other MO methods [4] while the last row contains three hyper-cubic designs discovered by the first modified GAO algorithm described in this paper.

Secondly we find pareto optima for optimization problem by the other method. As illustrated in Figure 5 objective function 1 is at conflicting with objective function 2. Results in this Figure construct a Pareto front, i.e., they are all optimum for problem but at multiple directions. One whose want to have better optimization (minimization or maximization) at objective 1 than objective 2 selects his desired approach that objective 1 is optimized better than other approaches but it is clear that its objective 2 is inferior and so on.

Also we see that objective 1 is in conflicting with objective 2 but objective 2 and objective 3 (constraint) are like the other.

In many instances designers face situations where either due to manufacturing and operational constraints or economic factors a given design scenario can no longer remain optimal (e.g. a new motor may not provide the same range of cutting speeds). In these cases, the dominance-based GAO algorithm will be able to provide designers with myriad of optimal, tradeoff solutions. The

---

**Table 1. Comparison of design solutions produced by various methodologies.**

<table>
<thead>
<tr>
<th>METHOD</th>
<th>X = (x1, x2, x3, x4)</th>
<th>F(X) = (f1(X), f2(X), f3(X))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min Max</td>
<td>(79.99, 49.99, 0.90, 2.39)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.90, 2.08)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(79.99, 50.00, 0.90, 1.79)</td>
<td></td>
</tr>
<tr>
<td>Taguchi</td>
<td>(307.5, 0.0127, 4.318)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(276.5, 0.0143, 4.879)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(247.8, 0.0163, 5.579)</td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>(80.00, 50.00, 0.90, 2.30)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.90, 3.20)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.90, 4.10)</td>
<td></td>
</tr>
<tr>
<td>GAO(new)</td>
<td>(297.8, 0.0132, 4.467)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(386.2, 0.0099, 3.327)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(474.6, 0.0081, 2.671)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.903, 1.64)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.904, 1.71)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.902, 1.88)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(176.31, 0.025, 8.94)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(202.65, 0.021, 7.39)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(213.45, 0.026, 8.82)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 50.00, 0.900, 1.405)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(80.00, 49.937, 0.901, 1.324)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(79.999, 49.998, 0.916, 1.61)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(209.9704, 0.0199, 6.9065)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(201.8582, 0.0209, 7.2859)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(231.2765, 0.0177, 6.1213)</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 4. Cross section area (cm²), static deflection (cm) and bending stress (kN/cm²) values produced in three statistically independent runs.**
deterministic approaches of traditional MO approached, however, are rigid in that they do not easily allow incorporation of model uncertainty parameters into design.

6. Conclusions

This paper explained two general methodology for achieving multi-objective optimization and robust design. The one of important differences between SOPs and MOPs is that at first one there exist one fitness for each individual but at MOPs there exist a vector fitness per each individual, with the size of objective numbers. For converting vector fitness to a scalar fitness were addressed a modified aggregation method and dominant based pareto front method by using a modified GAO algorithm. Formulation of aggregation method, as mentioned, is a simple weighted collection with respect to objectives importance, as opposed to this traditional single point based representation, realistic reasonable methods proposed by researchers that are based upon dominant concept that construct a pareto front solutions. A historical view of improvements that has occurred and comparison of them by SOPs proposed.

Normalization and weighted aggregation of response objectives were employed to assign fitness values to individual design regions based on the criterion of non-inferiority at first way. The example of multi-objective design of an I-beam highlighted the advantages of using this over classical approaches. But applying second method gives pareto solutions and design selectivity are provided. Also conflicting between objectives are demonstrated.

7. References


