

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

Ali Khazaei^{1,2}, 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: khazaeali@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 of objective responses

$F(X) = [f_1(X), \dots, f_p(X)]^T$ where $X = [x_1, \dots, x_n]^T$ 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:

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

Subject to:

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

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

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 $u = (u_1, \dots, u_p)$ is said to be inferior to (dominated by) vector $v = (v_1, \dots, v_p)$ if

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

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 re-

gions. 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-

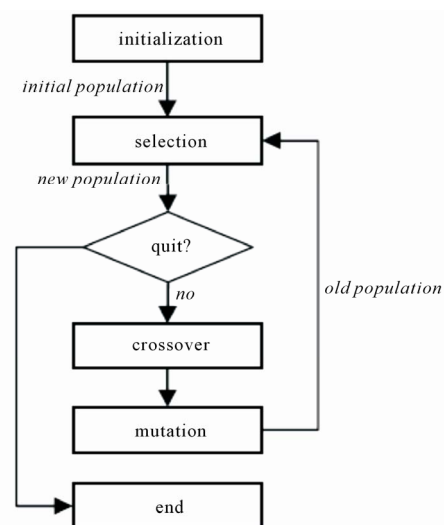


Figure 1. The operation of a generic GA.

lution [Bäck97]. 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 is equal to the size of the set of objective functions. *Task2b* converts the vector fitness for each individual into a scalar value (fitness) using some specific technique (different for each MOEA). *Task2b* usually incorporates a niching technique.

In this method for converting vector fitness into a scalar fitness we use a weighted aggregation of all of vector fitness components. Assessing the fitness of a individual

involves evaluation of the individual points

$X = [x_1, \dots, x_n]^T$ within that objectives using the response vector $F(X)$. Since design objective functions may have different scales and requirements (minimization or maximization), the algorithm normalizes the computed raw fitness f_i for the i th objective to the value J_i which ranges in $[0, 1]$; the closer J_i is to 1 the better the fitness is and the closer J_i is to 0 the worse the fitness. The normalized fitness for the i th design objective is calculated using that design objective's global reference maximum ($f_{i \max}$) and minimum ($f_{i \min}$) values, which are specified by the problem's geometric constraints. So, if the i th design objective is to be maximized, J_i is calculated by $(f_i - f_{i \min}) / (f_{i \max} - f_{i \min})$, and if minimization of the objective is the goal, J_i is calculated by $(f_{i \max} - f_i) / (f_{i \max} - f_{i \min})$.

Even with a normalized fitness for each design objective of an individual point, determining which individual point is the "best" one within a population is still problematic primarily because of the nature of multi-objective problems, that is, multiple individual points may be Pareto-optimal (see Section 2). To this end, the MO method of weighted aggregation [1] was used to compute the overall normalized fitness for the k th individual point as:

$$t_k = \sum w_i \phi_i \quad (1)$$

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 $T = [t_1, \dots, t_n]^T$, 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, $F(x) = [F1(x), F2(x), \dots, Fm(x)]$, 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-

<p><i>Task1: Initiate population</i> Repeat $t = 1, 2, \dots$ <i>Task2: Evaluate solutions in the population:</i> a) for each individual obtain a scalar fitness <i>Task3: Perform competitive selection in population</i> <i>Task4: Apply variation operators to the population</i> Until convergence criterion is satisfied</p>
<p><i>Task 1: Initiate population</i> Repeat $t = 1, 2, \dots$ <i>Task 2: Evaluate solutions in the population:</i> a) for each individual obtain a vector fitness b) for each individual convert vector fitness into a scalar fitness <i>Task3: Perform competitive selection in the population</i> <i>Task4: Apply variation operators to the population</i> Until convergence criterion is satisfied</p>

Figure 2. Pseudo-code of an EA for single-objective optimization (top) and multi-objective optimization (bottom).

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:

$$\min F(x), x \in R^n \text{ subject to:}$$

$$G_i(x) = 0, i = 1, \dots, k_l;$$

$$G_i(x) \leq 0, i = k_l + 1, \dots, k; l \leq x \leq u$$

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 Ω , in the parameter space. x is an element of the n -dimensional real numbers that satisfies all the constraints, *i.e.*,

$$\Omega = \{X \in R^n\}, \text{ subject to}$$

$$G_i(x) = 0, i = 1, \dots, k_l;$$

$$G_i(x) \leq 0, i = k_l + 1, \dots, k; l \leq x \leq u$$

This allows definition of the corresponding feasible region for the objective function space Λ :

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

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 $\pi_i(t)$ individuals; its current rank can be given by:

$$\text{Rank}(x_i, t) = 1 + \pi_i(t) \quad (2)$$

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

to achieve better convergence in MOEAs. Zitzler and Thiele [Zitzler98] suggested an elitist multi-criterion EA with the concept of non-domination in their *Strength Pareto Evolutionary Algorithm (SPEA)*. They suggested maintaining an external population at every generation storing all non-dominated solutions discovered so far beginning from the initial population. This external population participates in genetic operations. At each generation, a combined population with the external and the current population is first constructed. All nondominated solutions in the combined population are assigned a fitness based on the number of solutions they dominate, and dominated solutions are assigned fitness worse than the worst fitness of any non-dominated solution. This fitness assignment assures that the search is directed towards the non-dominated solutions. A deterministic clustering technique is also used to maintain diversity among non-dominated solutions. Although the implementation suggested is $O(mN^3)$, with appropriate book-keeping the complexity of SPEA can be reduced to $O(mN^2)$. An improved version of SPEA known as SPEA2 [Zitzler01] has recently been proposed. This incorporates additionally fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method.

Knowles and Corne [Corne00] suggested a simple MOEA using an evolutionary strategy (ES). In their *Pareto-Archived ES (PAES)*, a parent and a child are compared. If the child dominates the parent, the child is accepted as the next parent and the iteration continues.

On the other hand, if the parent dominates the child, the child is discarded and a new mutated solution (a new child) is found. However, if the child and the parent do not dominate each other, the choice between the child and the parent considers the second objective of maintaining diversity among obtained solutions. To achieve this diversity, an archive of non-dominated solutions is created. The child is compared with the archive to determine whether it dominates any member of the archive. If so, the child is accepted as the new parent and the dominated solution is eliminated from the archive. If the child does not dominate any member of the archive, both parent and child are examined for their proximity to the solutions of the archive. If the child resides in an un-crowded region in the parameter space among the members of the archive, it is accepted as a parent and a copy is added to the archive. Knowles and Corne later, suggested a *multi-parent PAES* with similar principles to the above. The authors have calculated the worst case complexity of PAES for N evaluations as $O(amN)$ where a is the archive length. Since the archive size is usually chosen proportional to the population size N , the overall complexity of the algorithm is $O(mN^2)$ [14]. Finally,

Deb [Deb00] has also proposed an improved parameter-less version of NSGA, called NSGA-II.

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:

$$f_1(X) = 2x_2x_4 + x_3(x_1 - 2x_4) \quad (3)$$

$$f_2(X) = \frac{60000}{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]} \quad (4)$$

Subject to the bending stress constraint:

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

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

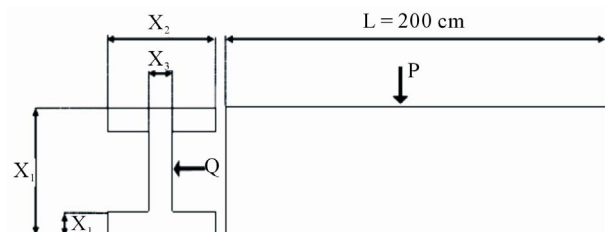


Figure 3. The frontal and side views of the I-beam.

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

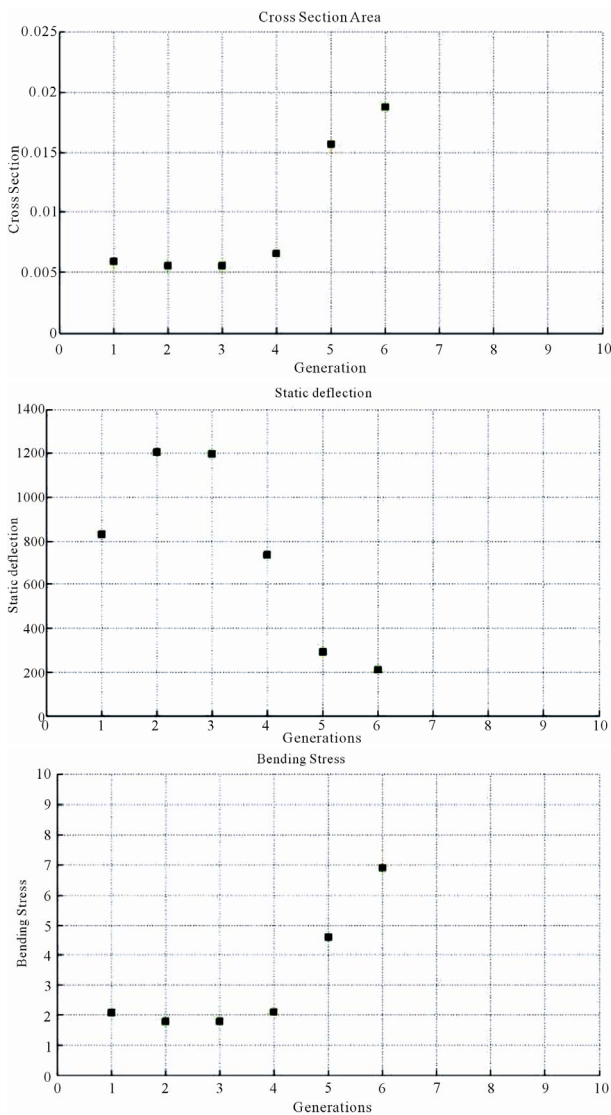


Figure 4. Cross section area (cm³), static deflection (cm) and bending stress (kN/cm²) values produced in three statistically independent runs.

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

METHOD	$X = (x1, x2, x3, x4)$ $F(X) = (f_1(X), f_2(X), f_3(X))$
Min Max	1. (79.99, 49.99, 0.90, 2.39) 2. (80.00, 50.00, 0.90, 2.08) 3. (79.99, 50.00, 0.90, 1.79)
	1. (307.5, 0.0127, 4.318) 2. (276.5, 0.0143, 4.879) 3. (247.8, 0.0163, 5.579)
	1. (80.00, 50.00, 0.90, 2.30) 2. (80.00, 50.00, 0.90, 3.20) 3. (80.00, 50.00, 0.90, 4.10)
Taguchi	1. (297.8, 0.0132, 4.467) 2. (386.2, 0.0099, 3.327) 3. (474.6, 0.0081, 2.671)
	1. (80.00, 50.00, 0.903, 1.64) 2. (80.00, 50.00, 0.904, 1.71) 3. (80.00, 50.00, 0.902, 1.88)
	1. (176.31, 0.025, 8.94) 2. (202.65, 0.021, 7.39) 3. (213.45, 0.020, 6.82)
PSO	1. (80.00, 50.00, 0.900, 1.405) 2. (80.00, 49.937, 0.901, 1.324) 3. (79.999, 49.998, 0.916, 1.61)
	1. (209.9704, 0.0199, 6.9065) 2. (201.8582, 0.0209, 7.2859) 3. (231.2765, 0.0177, 6.1213)
	1. (209.9704, 0.0199, 6.9065) 2. (201.8582, 0.0209, 7.2859) 3. (231.2765, 0.0177, 6.1213)

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 function1 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

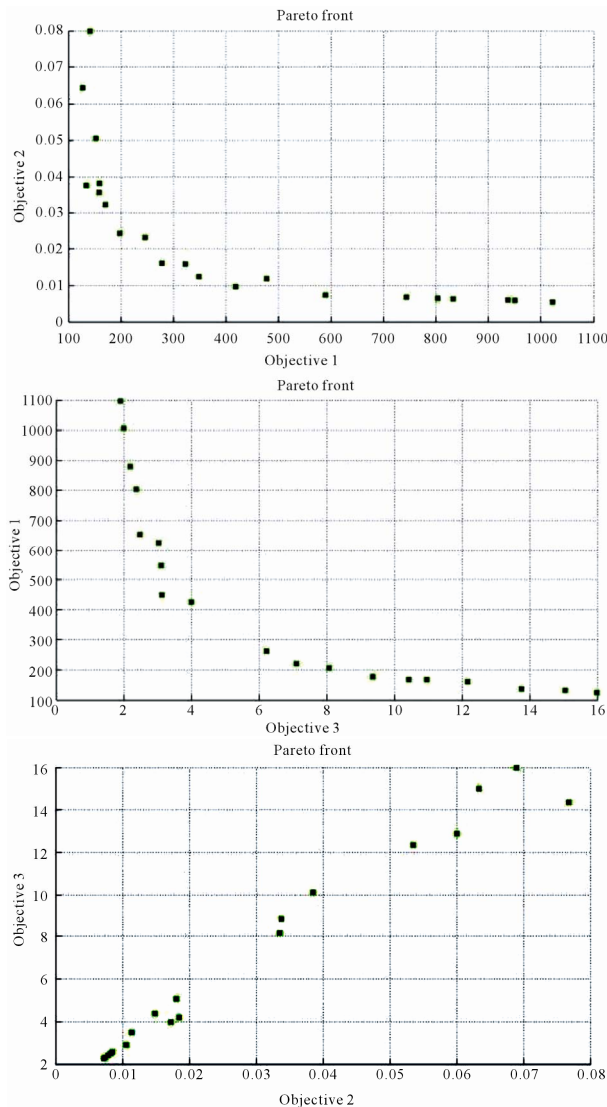


Figure 5. Cross section area (cm^3), static deflection (cm) and bending stress (kN/cm^2) values produced in three statistically independent runs.

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

- [1] J. Andersson, "A Survey of Multi-objective Optimization in Engineering Design," Technical Report No. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, Linköping, 2000.
- [2] A. Osyczka, "Multicriteria Optimization for Engineering Design," In: J. Gero, Ed., *Design Optimization*, 1985, pp. 193-227.
- [3] J. Arkat, L. Hosseini and M. H. Farahani, "Minimization of Exceptional Elements and Voids in the Cell Formation Problem Using a Multi-Objective Genetic Algorithm," *Expert Systems with Applications*, Vol. 38, 2011, pp. 9597-9602. [doi:10.1016/j.eswa.2011.01.161](https://doi.org/10.1016/j.eswa.2011.01.161)
- [4] B. Fourouraghi, "A Genetic Algorithm for Multi-Objective Robust Design," *Journal of Applied Intelligence*, Vol. 12, No. 3, 2000, pp. 151-161. [doi:10.1023/A:1008356321921](https://doi.org/10.1023/A:1008356321921)
- [5] X. Li, "Better Spread and Convergence: Particle Swarm Multi-Objective Optimization Using the Maximum Fitness Function," *Lecture Notes in Computer Science*, Vol. 3102, 2004, pp. 117-128. [doi:10.1007/978-3-540-24854-5_11](https://doi.org/10.1007/978-3-540-24854-5_11)
- [6] P. R. McMullen, "An Ant Colony Optimization Approach to Addressing a JIT Sequencing Problem with Multiple Objectives," *Artificial Intelligence in Engineering*, Vol. 15, No. 3, 2001, pp. 309-317. [doi:10.1016/S0954-1810\(01\)00004-8](https://doi.org/10.1016/S0954-1810(01)00004-8)
- [7] H. Pohlheim, "Genetic and Evolutionary Algorithms: Principles, Methods and Algorithms," 2005. <http://www.geatbx.com/docu/index.html>
- [8] P. G. Espejo, S. Ventura and F. Herrera, "A Survey on the Application of Genetic Programming to Classification," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, Vol. 40, No. 2, 2010, pp. 121-144. [doi:10.1109/TSMCC.2009.2033566](https://doi.org/10.1109/TSMCC.2009.2033566)
- [9] H. Gong, R. Zulkernine and P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection," *Proceedings of*

- the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05)*, Canada K7L 3N6, 2005.
- [10] E. Ochlak and B. Forouraghi, "A Particle Swarm Algorithm for Multi-Objective Design Optimization," *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, Philadelphia, 20-23 August 2006.
- [11] L. A. Zadeh, "Optimality and Nonscalar-Valued Performance Criteria," *IEEE Transactions on Automatic Control*, Vol. AC-8, 1963, p. 1.
- [12] A. Guigue, M. Ahmadi, R. Langlois and J. Hayes, "Pareto Optimality and Multiobjective Trajectory Planning for a 7-DOF Redundant Manipulator," *IEEE Transactions on Robotics*, Vol. 26, No. 6, 2010, pp. 491-500.
[doi:10.1109/TRO.2010.2068650](https://doi.org/10.1109/TRO.2010.2068650)
- [13] N. O. Da Cunha and E. Polak, "Constrained Minimization under Vector-Valued Criteria in Finite Dimensional Spaces," *Journal of Mathematical Analysis and Applications*, Vol. 19, 1967, pp. 103-124.
[doi:10.1016/0022-247X\(67\)90025-X](https://doi.org/10.1016/0022-247X(67)90025-X)
- [14] F. de Toro, J. Ortega, J. Fernández and A. Díaz, "PSFGA: A Parallel Genetic Algorithm for Multi-Objective Optimization," *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-Based Processing (EUROMICRO-PDP.02)*, Canary Islands, 9-11 January 2002.