AN EVOLUTIONARY ALGORITHM WITH A MULTILEVEL PAIRING STRATEGY FOR SINGLE AND MULTIOBJECTIVE OPTIMIZATION

Tapabrata RAY* and Kang TAI**

Abstract. This paper presents an evolutionary algorithm that incorporates a multilevel pairing strategy to solve single and multiobjective optimization problems. The algorithm is based on nondominance of solutions *separately* in the objective and the constraint space and uses cooperative mating strategies between solutions. Since the methodology is based on nondominance separately in the objective and the constraint space, scaling and aggregation affecting conventional penalty function methods for constraint handling does not arise. The proposed cooperative and intelligent pairing strategies result in mating between solutions that are good in objectives with those that are good in constraint satisfaction, thus helping to speed up convergence. The diversification mechanism in the algorithm is based on niching that results in a wide spread of solutions in the parametric space. Three constrained multiobjective design examples and a single objective optimization problem with continuous and mixed variables are used to illustrate the performance of the proposed algorithm.

1 Introduction

Optimization problems inherently involve optimizing multiple non-commensurable and conflicting objectives subject to various specifications and constraints. In a single

* Center for Advanced Numerical Engineering Simulations, School of Mechanical and Production Engineering, Nanyang Technological University, Singapore 639798.Email: mtray@ntu.edu.sg

** School of Mechanical and Production Engineering, Nanyang Technological University, Singapore 639798.Email: mktai@ntu.edu.sg

objective problem, the aim is to find the *best* solution that maximizes or minimizes the objective, while in a multiobjective optimization problem (in absence of inter-objective preference information) the goal is to arrive at a set of Pareto optimal designs. It is useful to make that set of Pareto optimal designs diverse so as to provide a wide choice for the decision-maker.

Classical optimization methods are in general not efficient for multiobjective problems as they often lead to a single solution instead of a set of optimal solutions. Multiple runs of the same method cannot guarantee a different point on the Pareto frontier each time and some methods cannot even handle problems with multiple optimal solutions. Evolutionary methods maintain a set of solutions as a population during its course of search and thus can result in a set of Pareto optimal solutions in a single run. A widely differing set of Pareto optimal solutions can be generated using a diversification strategy (e.g. niching) within the evolutionary algorithm.

In this paper, we propose an evolutionary algorithm that is capable of handling single and multiobjective design optimization problems. It incorporates a Pareto ranking scheme *separately* in the constraint and the objective function space, thereby eliminating the problems of objective or constraint aggregation and scaling. Since the algorithm does not involve any aggregation of objectives or constraints, it optimizes the true objective function. Incorporated within the algorithm are schemes for intelligent cooperative mating that use knowledge of individual constraint satisfaction and performance in the objective function domain to speed up convergence.

Section 2 provides a brief overview of multiobjective optimization and constraint handling concepts in the context of evolutionary computing. The proposed algorithm and its details are discussed in Section 3. The performance of the algorithm is illustrated by the design examples in Section 4.

2 Overview of Multiobjective Optimization and Constraint Handling Methods

Multiobjective problems form a major class of design optimization problems. Majority of these problems involve constraints arising out of physical laws, statutory requirements or resource limitations. Section 2.1 highlights the features of various evolutionary multiobjective formulations while Section 2.2 focuses on implementation issues of various constraint handling methods in the context of evolutionary computation.

2.1 Multiobjective Optimization

The goal in a multiobjective optimization problem (in absence of preference information among the objectives) is to arrive at a set of Pareto optimal solutions. As identified earlier, it is necessary to provide a wide variety in the set of solutions for the decision-maker to choose from. Fonseca and Flemming[10] present an excellent review of multiobjective optimization methods while a comprehensive recent survey has been reported by Coello[2].

Vector Evaluated Genetic Algorithm (VEGA) is the earliest example of an evolutionary algorithm designed for finding an approximation to the Pareto optimal solution set of a multiobjective problem. Shaffer[28] proposed VEGA as an extension of Grefenstette's[12] (GENESIS) program for simple genetic algorithm. It is based on the use of multiple populations and selective migration of individuals from one population to another. The methodology is simple to understand and implement but requires a large number of subpopulations with sufficient number of individuals in them for problems with many objectives. In addition to that, a fitness evaluation mechanism based on a linear combination of the objectives will fail to generate Pareto optimal solutions for non-convex search spaces regardless of weights used. Lis and Eiben[23] advocated the use of multiparent crossovers among individuals of different genders. One parent from each gender is selected to contribute to the generation of a child, and the child inherits the gender of parent that contributed to the maximum number of genes. The methodology requires a large population size incase of multiple objectives, as sufficient number of individuals of each gender should be present in the population apart from the fact that the panmictic crossover requires more number of parents to generate a child and is computationally expensive.

Weighted aggregation method was an obvious extension to the classical single objective optimization formulation to handle multiobjective problems. The principle of weighted aggregation of objectives is reflected in the works of Ishibuchi and Murata[17] and Hajela and Lin[13]. Ishibuchi and Murata[17] used a combination of a weighted-sum-based evolutionary algorithm [EA] and a local search algorithm while Hajela and Lin[13] employed a variable set of weights for objective aggregation along with sharing and mating restrictions. In both of these methods, multiple objectives are transformed to a single measure by the use of weights, and hence all the drawbacks of objective aggregation exist though it eliminates the requirement of having a number of subpopulations. Moreover, additional input parameters are required for sharing and mating restriction as in the method of Hajela and Lin[13], and these may not be easy to provide.

The use of Pareto ranks instead of aggregated fitness values is another approach to handle multiple objectives. Multiobjective Genetic Algorithm (MOGA) as proposed by Fonseca and Flemming[11] and the Nondominated Sorting Genetic Algorithm (NSGA) by Srinivas and Deb[31] belongs to this category. In MOGA, a solution is assigned a rank based on the number of solutions in that population that dominates it. Subsequently, a fitness value is assigned to the solution based on its rank. The process of Pareto ranking is computationally expensive and the use of blocked fitness assignment to a set of solutions with the same rank results in a large selection pressure and leads to premature convergence. To distribute the points evenly over the Pareto optimal region, MOGA employs a sharing mechanism in the objective function space. As sharing is performed in the objective function space, MOGA may not be able to find solutions to problems where different Pareto optimal points correspond to the same objective function value. NSGA on the other hand is based on multi-layered classification of nondominance and is computationally even more expensive when compared with MOGA. In addition, NSGA requires the sharing parameter as an input that may not be easy for the user to provide.

Apart from these three fundamental approaches for multiobjective modeling, there have also been interesting attempts to develop algorithms that share features of more than one of the above approaches. In order to carry good solutions across generations, the use of elitist strategy is quite common. Laumanns et al.[22] proposed the unified model for multiobjective optimization with elitism, while Deb et al.[4] introduced a fast, elligist NSGA. Horn et al.[16] incorporated the concepts of tournament selection and Pareto dominance in Niched Pareto Genetic Algorithm (NPGA). The performance of the NPGA is largely dependent on the size of the population against which a solution is compared to compute its nondominance. A small population size for comparison results in a few nondominated points in the population while a large one results in premature convergence. Valenzuela-Rendon and Uresti-Charre[33] used a Pareto selection and fitness sharing in their nongenerational GA model. The multiobjective problem is transformed to a bi-objective one by minimizing the domination count of an individual (weighted average of the number of individuals that have been dominated so far) and the minimization of its moving niche count (weighted average of the number of individuals that lie close to the above individual based on a sharing function). Additional inputs are required in the above method for weights and sharing function that may not be easy for the user to provide. Osyczka and Kundu[25,26] also proposed a multiobjective evolutionary algorithm that is based on the concepts of Pareto distance. This method does not require an explicit sharing function but the performance is largely dependent on the starting distance used to compare the quality of the solutions.

Some of the limitations that exist in modeling multiobjective optimization problems are be summarized as follows:

1. Weighted aggregation of objectives or constraints lead to problems of scaling and proper identification of weights may be difficult.
2. Additional parameters for sharing or niche count may not be easy to provide.
3. Large population size or a large number of subpopulations may be necessary for some of the approaches.
4. No specific guidelines are outlined in the multiobjective methods to incorporate constraints.

Comparison between various multiobjective optimization algorithms is a difficult task as identifying proper measures of performance is often non-trivial. Veldhuizen and Lamont[34] proposed the use of error ratio and generational distances to compare between Pareto fronts obtained by various multiobjective algorithms. However, both the measures require a known Pareto front. Zitzler and Thiele[35] introduced coverage and spread as two new measures that can be used to compare between fronts. In order to compute spread and coverage, both the solutions and their objective values are required as obtained by various algorithms. Knowles and Corne[20] proposed the use of statistical measures based on an attainment surface, while Fonseca and Flemming[8] applied the concepts of goal and priorities through relational operations to allow multiobjective decision making. There has also been significant work done by Hansen and Jaszkiewicz[14] on measures of comparison between various approximations to the nondominated set in presence of additional preference information through the use of outperformance, quantitative estimates or reference based methods. Jaszkiewicz[18] studied and reported the performance of the multiple objective genetic local search on the 0/1 knapsack problem and compared its performance with other existing multiobjective methods.

2.2 Constraint Handling Methods

After having discussed the various multiobjective methods and their limitations, it is necessary to focus our attention on the methods of constraint handling, as most design optimization problems involve constraints. The presence of constraints significantly affects the performance of any optimization algorithm including evolutionary optimization methods. A comprehensive review on constraint handling methods is presented by Michalewicz[24]. A wide range of constraint handling methods have evolved over the years ranging from rejection of infeasible solutions, use of penalty functions and their variants, repair methods, use of decoders, separate treatment of constraints and objectives and hybrid methods incorporating knowledge of constraint satisfaction. Each of the methods has its disadvantages; penalty functions using static, dynamic or adaptive concepts suffer from common problems of aggregation and scaling. Repair methods are based on additional function evaluations that may be expensive, while the decoders and special operators or constraint satisfaction methods are problem specific and cannot be used to model a generic constraint.

Separate treatment of constraints and objectives via Pareto ranking schemes is an interesting concept that eliminates the problem of constraint or objective scaling and aggregation. Constraint handling using a Pareto ranking scheme is a relatively new concept having its origin in multiobjective optimization. Jimenez and Verdegay[19] used a nondominated sorting ranking scheme as proposed by Srinivas and Deb[31] to deal with multiple objectives while a separate evaluation function was used for infeasible solutions. Surry et al.[32] applied a Pareto ranking scheme among constraints while fitness was used in the objective function space for the optimization of gas supply networks. Fonseca and Flemming[11] proposed a unified formulation to handle multiple constraints and objectives based on a Pareto ranking scheme. All the above attempts successfully eliminate the drawbacks of aggregation and scaling found in penalty function methods. An interesting attempt to incorporate the knowledge of constraint satisfaction during mating was proposed by Hinterding and Michalewicz[15]. In an attempt to match *the beauty with the brains*, constraint matching was employed during partner selection. A single measure (sum of squares of violation) was used to compute a solution's infeasibility. However, a single aggregate measure of infeasibility fails to incorporate the knowledge of individual constraint satisfaction/violation and in addition leads to scalability and aggregation problems. Moreover, the algorithm did not include any niching or diversification mechanism to ensure a uniform spread of points along the Pareto frontier that is required for multiobjective problems. There have also been attempts by Koziel and Michalewicz[21] to handle single objective constrained optimization problems through the use of homomorphous mapping.

It is clear from the above discussion that a generic evolutionary algorithm for constrained single and multiobjective optimization should avoid aggregation and scaling of objectives and constraints, provide a wide spread of points along the Pareto frontier for multiobjective problems and be computationally efficient as function evaluations are often expensive. An evolutionary algorithm is presented in this paper that is based on Pareto ranking separately in the constraint and objective space thereby eliminating problems of scaling and aggregation. Intelligent parent matching concept within the algorithm results in mating between solutions that are good in either constraints or objectives and its complementary partner with the hope of generating better solutions through collaborative pairing. Different mating strategies for single objective unconstrained and constrained problems have been discussed by Ray et al.[27]. Since a wide variety is required among the Pareto optimal points for multiobjective problems, a diversification strategy is implemented that relies on no additional inputs for sharing or niching.

3 Multiobjective/Single Objective Evolutionary Algorithm

A general multiobjective optimization problem (in minimization sense) is presented as:

Minimize $\mathbf{f} = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \ldots \quad f_k(\mathbf{x})]$

subject to $g_i(\mathbf{x}) \geq a_i \quad , i = 1,2,\ldots,q$

$h_j(\mathbf{x}) = b_j \ , \ j = 1,2,\ldots,r$

where $\mathbf{f}$ is a vector of k objectives and $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]$ is the vector of n design variables to be minimized subject to q inequality and r equality constraints.

It is common practice to transform each equality constraint into a pair of inequalities (with a tolerance δ) resulting in a unified formulation for all constraints:

$$h_j(\mathbf{x}) \leq b_j + \delta$$
$$h_j(\mathbf{x}) \geq b_j - \delta$$

Thus r equality constraints will give rise to $2r$ inequalities, and the total number of inequalities for the problem is denoted by s, where $s=q+2r$.

For each solution, $\mathbf{c}$ denotes the constraint satisfaction vector given by $\mathbf{c} = [c_1 \quad c_2 \quad \ldots \quad c_s]$ where

$$c_i = \begin{cases} 0 & \text{if } i^{\text{th}} \text{ constraint satisfied}, \ i = 1,2,\ldots,s \\ a_i - g_i(\mathbf{x}) & \text{if } i^{\text{th}} \text{ constraint violated}, \ i = 1,2,\ldots,q \\ b_i - \delta - h_i(\mathbf{x}) & \text{if } i^{\text{th}} \text{ constraint violated}, \ i = q+1, q+2,\ldots,q+r \\ -b_i - \delta + h_i(\mathbf{x}) & \text{if } i^{\text{th}} \text{ constraint violated}, \ i = q+r+1, q+r+2,\ldots,s \end{cases}$$

For the above c_i's, $c_i = 0$ indicates the i^{th} constraint is satisfied, whereas $c_i > 0$ indicates the violation of the constraint.

With the above understanding of the problem, the pseudo-code of the algorithm is presented below.

Algorithm

- Initialize M solutions to form a population

Do {

- Compute R_{obj}, the Pareto ranking based on objective vectors **f**.
- Compute R_{con}, the Pareto ranking based on constraint satisfaction vectors **c**.
- Compute R_{com}, the Pareto ranking based on combined objective and constraint satisfaction vectors.

Multiobjective Case

- Put Feasible and R_{com} =1 individuals into the population for the next generation

Single Objective Case

- Put Feasible individuals with $R_{obj} \leq$ average of R_{obj} into the population for the next generation

To fill up the remaining vacancies in the population for the next generation

Do {

- Select an individual A and its partner from the population of this generation
- Mate A with its partner
- Put the children and their parents into the population for the next generation

} while the population is not full.

- Remove duplicate points in parametric space and shrink population

} while the stopping condition is not attained.

3.1 Pareto Ranking, Fitness and Selection

From a population of M solutions, all nondominated solutions are assigned a rank of 1. The rank 1 individuals are removed from the population and the new set of nondominated solutions is assigned a rank of 2. The process is continued until every solution is assigned a rank. This ranking scheme is based on Nondominated Sorting Genetic Algorithm (NSGA) as proposed by Srinivas and Deb[31]. Every individual in the population has a rank of R_{obj} for an unconstrained problem and ranks of R_{obj}, R_{con} and R_{com} for a constrained problem. These ranks(R_{obj} , R_{con} or R_{com}) are converted to corresponding fitness values using the expression $\max(R) - R - 1$, where R is either R_{obj} , R_{con} or R_{com} and $\max(R)$ is the maximum rank in the population (i.e. the rank of the worst solution). A roulette wheel selection based on the above fitness is used to choose individuals for mating. The process allows the fitter solutions to have a higher probability of being selected for mating.

3.2 Multilevel Pairing Strategy for Mating

Mating is performed between a solution A and its partner (B or C). The process of partner selection is dependent on the type of the constrained problem. Problems are classified into the following:

1. Unconstrained problem (Objective-Objective Mating)
2. Constrained problem (Objective-Constraint Mating)

For an unconstrained problem, A, B and C are selected based on the fitness derived from R_{obj}. For a constrained problem, selection of A is based on the fitness derived from R_{obj} while the selection of B and C is based on fitness derived from R_{con}. Such a mating between solutions that are *good* in objective function with that of solutions that are *good* in constraint satisfaction is analogous to mating between the '*beauty and the brains*'. The process of partner selection for a constrained problem is described below.

<u>Case 1: B and C are both feasible</u>

If $R_{\mathrm{obj\_B}} < R_{\mathrm{obj\_C}}$, then partner is B

If $R_{\mathrm{obj\_B}} > R_{\mathrm{obj\_C}}$, then partner is C

If $R_{\mathrm{obj\_B}} = R_{\mathrm{obj\_C}}$, then choose the one with the minimum adaptive niche count (to be explained in Section 3.4).

where $R_{\mathrm{obj\_B}}$ denotes the rank of solution B based on R_{obj}.

<u>Case 2 : B and C are both infeasible</u>

If $R_{\mathrm{con\_B}} < R_{\mathrm{con\_C}}$, then partner is B

If $R_{\mathrm{con\_B}} > R_{\mathrm{con\_C}}$, then partner is C

If $R_{\mathrm{con\_B}} = R_{\mathrm{con\_C}}$, then choose the one with minimum overlapping constraint satisfaction with A (to be explained in Section 3.5).

<u>Case 3 : One is feasible and the other is not</u>

If B is feasible, then partner is B

If C is feasible, then partner is C

3.3 Mating

Every mating generates three additional solutions unlike the conventional process of crossover that generates two children. Out of the three solutions, the uniform crossover between A and its partner generate one solution while the other two are generated using the random mix and move. The process of random mix and move for any continuous variable x is presented as follows:

Considering, for example, that A and B are the two parents contributing the variable values x_A and x_B , respectively, and where $x_A < x_B$ (Figure 1). The child will have a value of x randomly generated from within either one of the intervals $x_{lower\ bound} \le x \le x_A$, $x_A \le x \le x_B$ or $x_B \le x \le x_{upper\ bound}$. The selection of one out of those three intervals is in turn also random and according to probabilities of 25%, 50% and 25% respectively. In this present work, random mix and move has been used. However, other operators that generate intermediate variable values like the blend crossover (BLX) as proposed by Eshelman and Shaffer[6], the simulated binary crossover (SBX) as proposed by Deb and Agrawal[3] can also be used. The proposed random mix and move operator is designed to cater to continuous variables in the domain of parametric design. The operator can easily be extended to handle discrete and integer value problems.

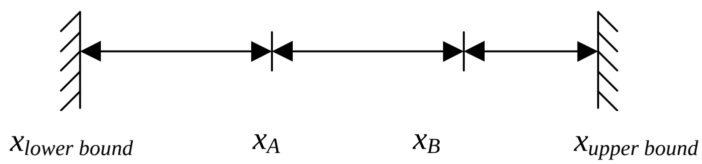

Figure 1. Schematic diagram for Random Mix and Move operator.

3.4 Adaptive Niche Count

A new parametric crowding measure referred as adaptive niche count is introduced in this work. Adaptive niche count of a solution is the number of other solutions in the population, which are within the average distance metric of that solution (average of the Euclidean distances between the solution and the rest of the *M-1* members in the population). A solution with a small niche count physically means that there are few other solutions in its parametric neighborhood. Such solutions are preferred over others and this is the diversification strategy used in the algorithm.

3.5 Non-overlapping Constraint Satisfaction

A partner selection strategy is proposed in this work that is based on a non-overlapping constraint satisfaction mechanism. An individual is allowed to mate with another if it complements the other towards constraint satisfaction. Such a mating between the *beauty* and the *brains* is incorporated with the hope of generating solutions with better constraint satisfaction. If the sets $\{S_A\}$, $\{S_B\}$ and $\{S_C\}$ denote the sets of constraints satisfied by solution A, B and C respectively, the selection of either B or C as the partner of A is based on the following conditions:

If $(\{S_A\} \cap \{S_B\}) > (\{S_A\} \cap \{S_C\})$, then partner is C.
If $(\{S_A\} \cap \{S_B\}) < (\{S_A\} \cap \{S_C\})$, then partner is B.
If $\{ \{S_A\} \cap \{S_B\}) = (\{S_A\} \cap \{S_C\})$, then partner is randomly chosen between B and C.

The operator $(\{S_A\} \cap \{S_B\})$ indicates the number of constraints that are both satisfied by A and B.

3.6 Population Shrinking

After each new population is full, a screening is done to remove identical points in the parametric (variable) space to give room for new and different solutions.

4 Results and Discussion

Three engineering design examples have been chosen to illustrate the performance of the algorithm. The first example is a continuous variable, constrained, multiobjective design of a welded beam with an aim to minimize cost and end deflection subject to constraints on shear, bending and buckling load. The second example is a multiobjective bulk carrier design with continuous variables and a large number of constraints. The third example is a mixed variable, multiobjective, constrained optimization problem dealing with a tanker fleet design.

4.1 Design of a Welded Beam

A welded beam is to be designed for minimum cost and minimum end deflection subject to constraints on shear stress, bending stress and buckling load. The four design variables *h, l, t* and *b* correspond to x_1, x_2, x_3 and x_4 and are shown in Figure 2.

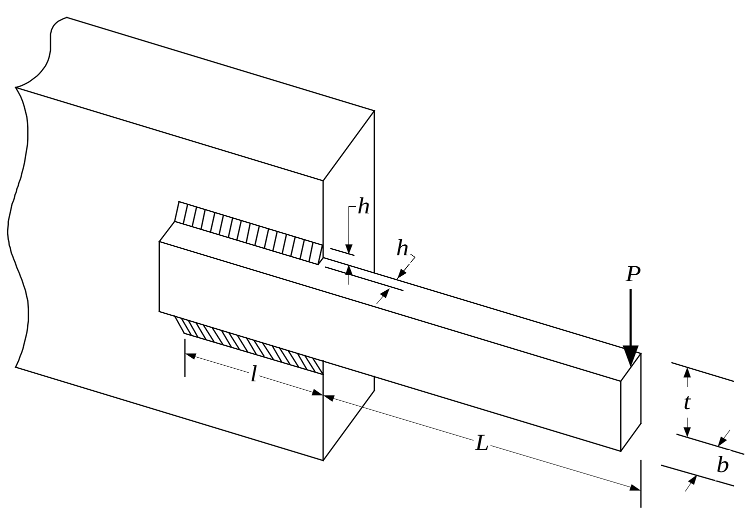

Figure 2. Welded beam design

Minimize $f_1(\mathbf{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$
Minimize $f_2(\mathbf{x}) = \delta(x)$

Subject to

$$\tau(\mathbf{x}) - \tau_{\max} \le 0$$
$$\sigma(\mathbf{x}) - \sigma_{\max} \le 0$$
$$x_1 - x_4 \le 0$$
$$0.125 - x_1 \le 0$$
$$P - P_C(x) \le 0$$

where $\tau(\mathbf{x}) = \sqrt{(\tau')^2 + \dfrac{2\tau'\tau'' x_2}{2R} + (\tau'')^2}$

$$\tau' = \frac{P}{\sqrt{2} x_1 x_2}$$
$$\tau'' = \frac{MR}{J}$$
$$M = P\left(L + \frac{x_2}{2}\right)$$
$$R = \sqrt{\frac{x_2{}^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$
$$J = 2\left\{\sqrt{2} x_1 x_2 \left[\frac{x_2{}^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\mathbf{x}) = \frac{6PL}{x_4 x_3{}^2}$$
$$\delta(\mathbf{x}) = \frac{4PL^3}{E x_4 x_3{}^3}$$

$$P_C(\mathbf{x}) = \frac{4.013E\sqrt{\dfrac{x_3{}^2 x_4{}^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

P= 6000 lb, L =14 in, δ_{max} =0.25 in, E =30 × 10^6 psi, G = 12 × 10^6 psi, τ_{max} = 13,600 psi, σ_{max} = 30,000 psi, 0.125 ≤ x_1 ≤ 5.0, 0.1 ≤ x_2 ≤ 10.0, 0.1 ≤ x_3 ≤ 10 and 0.125 ≤ x_4 ≤ 5.0.

A population size of 100 and a generation count of 300 have been used in the present study. Results of multiple runs are presented in Table 1. It can be observed from Table 1 that the number of Pareto points obtained and the number of function evaluations required are consistent across multiple runs. The Pareto optimal front as obtained from Run 1 is presented in Figure 3. The front consists of 99 points and has been obtained after 4481 function evaluations. This problem has also been attempted by Deb[5] using a real coded GA with simulated binary crossover(SBX). Deb[5] presented a similar front with the same limits and roughly the same visible spread of points along the Pareto curve using 50,000 function evaluations.

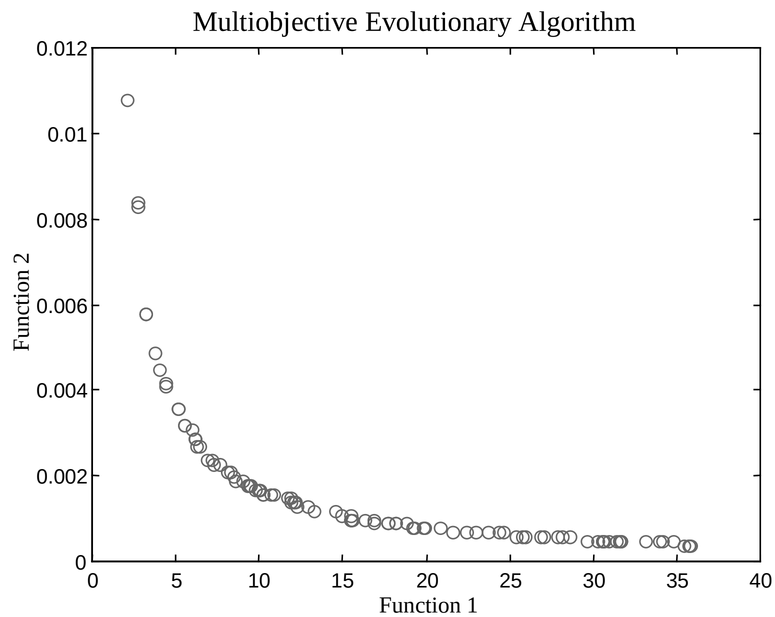

Figure 3. Pareto front for the welded beam design

Table 1: Results of Multiple Runs for the Welded Beam Design

	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Number of Pareto Points	99	91	96	98	96
Number of Function Evaluations	4481	4417	4472	4468	4470

4.2 Preliminary Design of a Bulk Carrier

Sen and Yang[30] originally introduced this bulk carrier design example. They considered the minimization of transportation cost, minimization of lightship mass and the maximization of annual cargo transport capacity as three objectives for the design. Since, the objective of lightship mass minimization is related to the maximization of the annual cargo transport capacity, the formulation presented in this paper considers the minimization of transportation cost and the maximization of annual cargo transport capacity as two objectives for the design. The modified optimization problem of the bulk carrier design is presented below along with all necessary relations for various estimates and bounds on the variables and their ratios. There are six continuous variables (L, T, D, C_B, B and V). The bounds for the variables used in the model are $0 \le L \le 500$, $0 \le T \le 50$, $0 \le D \le 50$, $0.63 \le C_B \le 0.75$, $0 \le B \le 100$ and $14 \le V \le 18$.

Minimize: Transportation Cost $= \dfrac{A\_COST}{A\_CARGO}$

Maximize: Annual Cargo Transport Capacity = $C\_DWT \times RTPA$

Subject to:

$$3{,}000 \le DW \le 500{,}000$$

$$\frac{V}{(gL)^{0.5}} \le 0.32$$

$$T \le 0.45 \times DW^{0.31}$$

$$T \le 0.7 \times D + 0.7$$

$$\frac{L}{B} \ge 6$$

$$\frac{L}{D} \le 15$$

$$\frac{L}{T} \le 19$$

$$GM \ge 0.07 \times B$$

$$A\_COST = CAP\_COST + RUN\_COST + VOY\_COST \times RTPA$$

$$CAP\_COST = 0.2 \times SHIP\_COST$$

$$RUN\_COST = 40000 \times DWT^{0.3}$$

$$VOY\_COST = F\_COST + P\_COST$$

$$F\_COST = 1.05 \times DC \times SD \times FP$$

$$DC = P \times 0.19 \times 24 / 1000 + 0.2$$

$$DWT = DISP - LSM$$

$$LSM = STEEL\_MASS + OUT\_MASS + MAC\_MASS$$

$$STEEL\_MASS = 0.034 \times L^{1.7} \times B^{0.7} \times D^{0.4} \times C_B^{\ 0.5}$$

$$OUT\_MASS = L^{0.8} \times B^{0.6} \times D^{0.3} \times C_B^{\ 0.1}$$

$$DISP = L \times B \times T \times C_B \times 1.025$$

$$C1 = -10847.2 \times CB^2 + 12817 \times CB - 6960.32$$

$$C2 = 4977.06 \times CB^2 - 8105.61 \times CB + 4465.51$$

$$P = DISP^{2/3} \times V^3 \times \frac{1}{\dfrac{C2 \times V}{(gL)^{0.5}} + C1}$$

$$MAC\_MASS = 0.17 \times P^{0.9}$$

$$SHIP\_COST = 1.3(2000 \times (STEEL\_MASS)^{0.85} + 3500 \times (OUT\_MASS) + 2400 \times P^{0.8})$$

$$SD = \frac{RTM}{24 \times V}$$

$$RTPA = \frac{350}{SD + PD}$$

$$PD = 2 \times \frac{C\_DWT}{C\_RATE} + 0.5$$

$$C\_DWT = DWT - FL - CSW$$

$$FL = DC \times (SD + 5)$$

$$CSW = 2 \times DWT^{0.5}$$

$$PC = 6.3 \times DWT^{0.8}$$

$$GM = 0.53 \times T + \frac{(0.085 \times C_B - 0.002)B^2}{T \times C_B} - 1.0 - 0.52 \times D$$

RTM = 5000 nautical miles
FP = 100 pounds / tonne
C_RATE = 8000 tonnes / day
g = 9.8065 m/s^2

A population size of 100 running for 50 generations has been used in this study. Results of multiple runs are presented in Table 2. It can be observed that the results presented in Table 2 are consistent across multiple runs. The design solutions obtained from Run 1 are presented in Table 3. The list of symbols used in the formulation is presented in Table 4.

The Pareto optimal front obtained from Run 1 is presented in Figure 4. The front consists of 20 points and has been obtained using 2377 function evaluations. The results are better as compared to Sen and Yang[30] as solutions with transportation cost of less than 9 pounds/tonne is available as can be seen from Figure 4.

Table 2: Results of Multiple Runs for the Bulk Carrier Design

	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Number of Pareto Points	20	20	20	22	19
Number of Function Evaluations	2377	2377	2377	2380	2379

Table 3: Pareto Optimal Solutions

Design	*L*	*T*	*D*	C_B	*B*	*V*
1	335.67580	20.42880	38.21770	0.74000	55.50070	16.33430
2	308.67380	17.11610	27.64110	0.69180	43.37090	16.94170
3	376.33460	18.38100	25.18870	0.73770	38.74600	16.82840
4	302.86420	18.27800	31.69210	0.65810	50.14410	17.07680
5	430.11660	24.82680	45.48020	0.68110	66.85150	17.27140
6	366.88500	22.31300	43.59060	0.70710	60.69840	17.82370
7	279.30360	17.30150	28.56920	0.71290	45.45560	15.66530
8	445.67970	24.48610	44.30420	0.72280	63.17070	17.96510
9	414.13200	23.64740	41.60850	0.73060	61.21980	17.04510
10	466.38480	25.52470	42.59910	0.67740	65.98000	17.48840
11	277.90920	17.02780	26.95820	0.72400	41.81020	16.22230
12	357.12570	21.81900	40.34000	0.71630	59.10760	17.82510
13	339.45320	19.13410	31.13030	0.71530	46.53580	17.86620
14	443.03240	25.30110	47.01590	0.70330	71.49320	17.51720
15	383.07270	22.80840	41.71110	0.71080	63.44940	16.33770
16	271.64910	14.37940	21.34030	0.67950	33.54440	16.27630
17	394.83540	23.07570	43.07370	0.67990	63.10800	17.96290
18	317.06880	18.62170	29.61330	0.72600	46.54240	15.98760
19	403.20650	24.67020	45.77710	0.72040	65.23120	16.82030
20	380.67390	20.76090	32.61870	0.71150	50.51520	17.99200

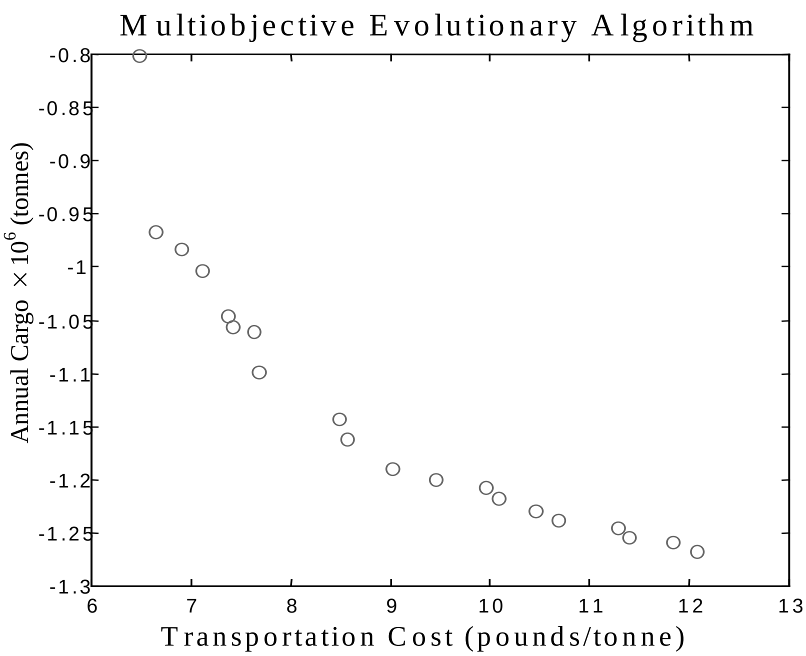

Figure 4. Pareto optimal front for Bulk Carrier Design

Table 4: List of Symbols

A_CARGO	Annual cargo carrying capacity (tonnes/year)
A_COST	Annual cost (pounds / year)
B	Breadth of the ship (m)
C_DWT	Cargo deadwight (tonnes)
C_RATE	Cargo handling rate (tonnes / day)
C1, *C2*	Coeffiecients
CAP_COST	Capital cost (pounds / year)
C_B	Block coefficient
CSW	Weight of crew, stores and water (tonnes)
D	Depth of the ship (m)
DC	Daily consumption of fuel (tonnes / day)
DISP	Displacement (tonnes)
DWT	Deadweight (tonnes)
F_COST	Fuel cost (pounds)
FL	Fuel carried (tonnes)
FP	Fuel price (pounds / tonne)
g	Acceleration due to gravity (m/s^2)
GM	Metacentric height (m)
L	Length of the ship (m)
LSM	Lightship mass (tonnes)
MAC_MASS	Machinery mass (tonnes)
OUT_MASS	Outfit mass(tonnes)
P	Shaft power (HP)
P_COST	Port cost (pounds)
RTM	Round trip (miles)
RTPA	Number of round trips per year
RUN_COST	Running cost (pounds / year)
SD	Number of sea days per year
SHIP_COST	Cost of ship (pounds)
STEEL_MASS	Steel mass (tonnes)
T	Draft of the ship (m)
V	Speed of the ship (knots)
VOY_COST	Voyage cost (pounds / voyage)

4.3 Design of a Tanker Fleet

Folkers[7] originally presented this tanker design example. A modified form has been attempted by Azarm and Narayanan[1] with new estimates of weight taken from Schneekluth[29]. The limits of the variables and their ratios and constraints from the original formulation of Folkers[7] have been modified substantially by Azarm and Narayanan[1], thereby making it difficult to compare results. The modified tanker design optimization problem is presented below that considers the minimization of cost and the maximization of cargo transportation capacity as two objectives. The cargo transport capacity in one direction should be greater than 10 million tonnes that translates to an annual cargo transport capacity of greater than 20 million tonnes as indicated in the original formulation by Folkers[7].

Minimize: Cost = $N \times (CHULL + CMAC + CF)$

Maximize: Cargo Capacity = $N \times U \times W \times (\frac{DWT \times V}{R} - \frac{FV^3 Z^{2/3}}{k_\alpha})$

Subject to

$$N \times U \times W \times (\frac{DWT \times V}{R} - \frac{FV^3 Z^{2/3}}{k_\alpha}) \geq 2 \times Q$$

$$WST + 0.02 \times (V^3 Z^{2/3})^{0.72} + DWT - Z \leq 0$$

$$U \leq \frac{\frac{R}{V}}{\frac{R}{V} + \frac{2 \times DWT}{O}} \leq 1$$

$$\frac{DWT}{L \times B \times D} - \frac{1}{3} \leq 0$$

$$1.5 + 0.45 \times D - B(\frac{0.08 \times B}{T \times C_M^{0.5}} + \frac{T \times (0.9 - 0.3 \times C_M - 0.1 \times C_B)}{B}) \leq 0$$

$$0.0019 \times L^{1.43} + T - D \leq 0$$

$$0.14 \leq \frac{V}{(gL)^{0.5}} \leq 0.32$$

$$0.60 \leq C_B \leq 0.72$$

$$5 \leq \frac{L}{B} \leq 7$$

$$10 \leq \frac{L}{D} \leq 14$$

$$2 \leq \frac{B}{T} \leq 4$$

$$0.61 \leq \frac{T}{D} \leq 0.87$$

The problem involves eight continuous variables and one integer variable. The variable bounds used in the model include $0 \leq B \leq 50$, $0 \leq D \leq 50$, $0 \leq DWT \leq 500{,}000$, $150 \leq L \leq 480$, $1 \leq N \leq 50$, $0 \leq T \leq 50$, $0 \leq U \leq 1$, $0 \leq V \leq 30$, $0 \leq Z \leq 600{,}000$.

$$CHULL = 0.25 \times K_{ST} \times Z \times (\alpha_L + 0.06\alpha_T \times (1.009 - 0.004 \times \frac{L}{B}) \times (28.7 - \frac{L}{D}))$$

$$CMAC = 2 \times (V^3 \times Z^{2/3})^{0.72}$$

$$CF = 0.8 \times U \times (V^3 \times Z^{2/3})^{0.72}$$

$$U = \frac{SHRS}{THRS}$$

$$WST = \frac{CHULL}{K_{ST}}$$

$$\alpha_L = (0.2771 + 0.02053\frac{L}{B}) \times (100 \times \frac{L}{D})^{-0.78}$$

$$\alpha_T = 0.029 + 0.00235 \times \frac{Z}{100000}$$

$$C_B = \frac{Z}{1.025 \times L \times B \times T}$$

$$k_1 = \frac{4}{L^{1/3}} + \frac{3}{L} + 0.2082$$

$$k_2 = \frac{3}{(2.58 + \frac{Z}{L \times B \times T})} - 0.07 \times (1 - \frac{Z}{0.65 \times L \times B \times T})$$

$$k_\alpha = 427.1$$

$$K_{ST} = k_0 \times k_1 \times k_2$$

$$C_M = 0.98$$

$$F = 0.00005 \text{ tonnes / SHP / hr}$$

$$g = 9.8065 \text{ m / s}^2$$

$$k_0 = 3{,}689.02$$

O = 2,500 tonnes / hr

W = 8,640 hrs / year

R =2,900 nautical miles

Table 5: Results of Multiple Runs for the Tanker Fleet Design

	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Number of Pareto Points	10	10	11	11	10
Number of Function Evaluations	5831	5831	5840	5839	5834

Table 6: Pareto Optimal Solutions

	B	D	DWT	L	N	T	U	V	Z
1	35.880	18.620	41450.000	244.000	24	12.910	0.825	15.170	71640.0
2	40.980	19.390	62310.000	262.200	47	12.090	0.805	12.810	93180.0
3	35.630	14.680	26480.000	205.200	34	9.068	0.814	12.050	44310.0
4	44.080	24.830	75660.000	256.500	49	18.430	0.684	14.310	130000.0
5	43.610	20.590	65970.000	236.300	18	14.070	0.762	14.680	91320.0
6	42.260	18.140	49400.000	237.000	36	12.600	0.701	14.040	78650.0
7	42.160	20.660	54180.000	269.500	35	12.680	0.782	16.450	95960.0
8	39.180	17.980	55690.000	243.300	16	11.810	0.618	12.540	70160.0
9	36.060	18.780	46740.000	207.300	34	12.940	0.756	13.710	64450.0
10	41.410	21.180	58140.000	248.700	13	14.970	0.620	14.740	100700.0

A population size of 200 running for 50 generations has been used in the study. Results of multiple runs are presented in Table 5. The Pareto optimal front in Figure 5 corresponds to Run 1 and consists of 10 points obtained after 5831 function evaluations. The design solutions corresponding to Run 1 are presented in Table 6.

Azarm and Narayanan[1] solved the above model without considering the constraint on minimum annual cargo transport capacity of 20 million tonnes. Pareto solutions reported by Azarm and Narayanan[1] are (18.00,199.99), (17.83,192.87), (17.71,181.91), (17.27,178.31) and (16.97,178.13), all with an annual cargo transport capacity of less than 20 million tonnes. The Pareto front obtained by our proposed algorithm without the annual transport capacity constraint is presented in Figure 6. The Pareto front presented in Figure 6 consists of 37 points and has been obtained after 4784 function evaluations with a population size of 200 running for 50 generations. Figure 7 shows the part of the Pareto curve with annual transport capacity less than 20 million tonnes. It can be clearly observed from Figure 7 that the solutions reported in this paper dominates the set of solutions presented by Azarm and Narayanan[1].

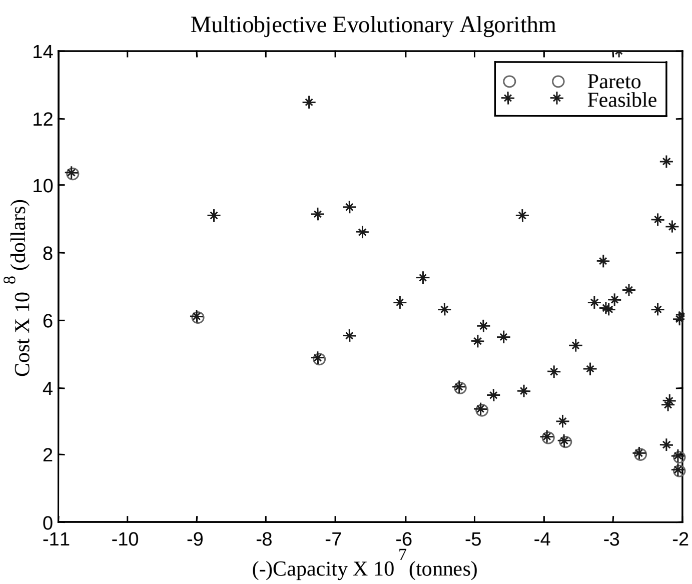

Figure 5. Pareto Optimal Front for Tanker Design

If the objective of cargo capacity maximization is dropped from initial formulation while maintaining the constraint on minimum annual cargo transport capacity and all other constraints, the problem turns out to be a single objective constrained mixed minimization problem. The best design obtained has a cost of 135,500,000 dollars and the optimal design is presented in Table 7. The symbols used in the formulation are listed in Table 8.

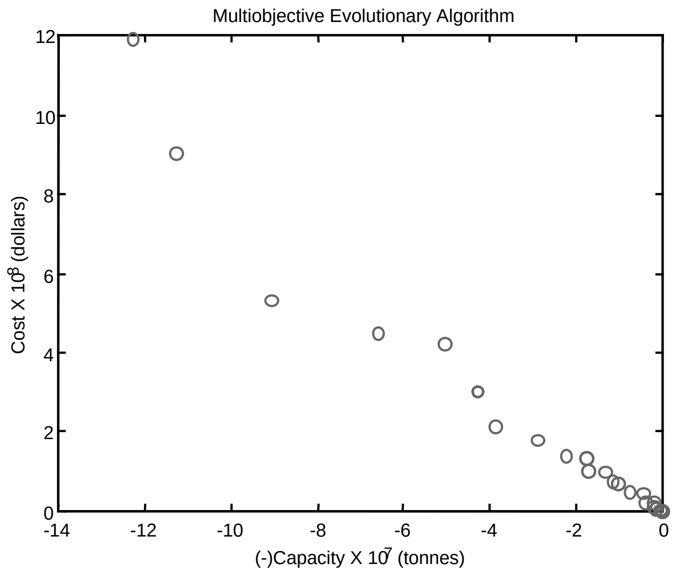

Figure 6. Pareto Optimal Front for Tanker Design without Transport Capacity Constraint

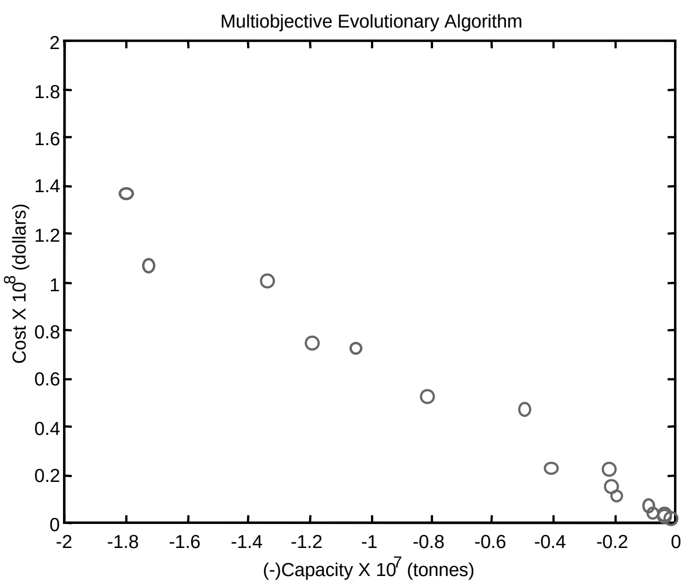

Figure 7. Pareto Optimal Front with Capacities less that 20 million tonnes

Table 7: Results of Single Objective Minimization

	B	D	DWT	L	N	T	U	V	Z
1	27.63	12.090	15200.000	165.200	44	7.4060	0.928	10.910	22660

Table 8: List of Symbols

B	Breadth of ship (m)
C_B	Block coefficient
CF	Cost of fuel (dollars)
$CHULL$	Cost of hull (dollars)
C_M	Midship area coefficient
$CMAC$	Cost of machinery (dollars)
D	Depth of ship (m)
DWT	Deadweight (tonnes)
F	Fuel consumption (tonnes / m^3)
G	Acceleration due to gravity (m / s^2)
α_L, α_T	Coefficients
K_0, k_1, k_2	Coefficients
K_{ST}	Coefficient
K_α	Admiralty Coefficient
L	Length of ship (m)
N	Number of ships
O	Loading rate (tonnes / hr)
Q	Cargo to be shipped per year (tonnes)
R	Range of operation (nautical miles)
$SHRS$	Hours at sea (hr)
T	Draft (m)
$THRS$	Total operating hours (hr)
U	Utilization rate = hrs at Sea/ Tot. Opr. Hrs.
V	Speed of ship (knots)
W	Number of operating hrs / year
WST	Weight of hull (tonnes)
Z	Displacement (tonnes)

5 Concluding Remarks

The evolutionary algorithm introduced in this paper is suitable for both single and multiobjective optimization problems from the domain of engineering design. The use of Pareto ranking concept in the objective and the constraint domain eliminates the problems of scaling and aggregation. The method is problem independent and can handle any computable constraint and in addition optimizes the true objective function without any additional user input. Though Pareto ranking is a computationally intensive operation, it eliminates the problem of finding adequate penalty function parameters or weights for objective aggregation. With an increase in the number of constraints or objectives, the process of Pareto ranking will be more and more computationally expensive. Intelligent multilevel mating strategies with niching i.e. mating the *beauty* with the *brains* result in an evenly spread Pareto optimal front with minimal function evaluations. Through niching and Pareto ranking are computationally expensive operations, it is meaningful to make use of all computed information to guide the search especially for problems with expensive objective function evaluations. The preliminary results presented in this paper are promising for both multiobjective and single objective optimization examples. Comprehensive tests are currently being conducted on suites of multiobjective and single objective examples to establish the efficiency of the proposed algorithm.

In order to compare results of the single objective optimization problem, the optimum value and the number of function evaluations for multiple runs have been reported in this paper. For the multiobjective problems, the visual Pareto front, the number of points in the front and the number of function evaluations required to attain the front has been reported in this paper. The Pareto optimal solutions for the bulk carrier design and the tanker fleet design example have been listed in this paper to allow future comparisons viable. Solutions to all the three examples including the welded beam design as discussed in this paper can be obtained from `http://www.cs.put.poznan.pl/fcds/Tapabrata_results`.

Acknowledgment

The authors would like to acknowledge the support for this work received from the Institute of High Performance Computing, Singapore. The authors would specially like to thank Professor Jaszkiewicz and the anonymous referees for their critical and constructive comments towards a better presentation of this work.

References

1. Azarm, S. and Narayanan, S.: A multiobjective interactive sequential hybrid optimization technique for design decision making, *Engineering Optimization*, 32(2000), 485-500.
2. Coello, C.A.C.: An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends, *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington(1999), 3-13.
3. Deb, K. and Agrawal, R.B.: Simulated binary crossover for continuous search space, *Complex Systems*, 9(1995), 115-148.
4. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature*, Paris(2000), 849-858.

5. Deb, K.: Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design, *Proceedings of Evolutionary Algorithms in Engineering and Computer Science*, EUROGEN-99(1999).
6. Eshelman, L.J. and Shaffer, J.D.: Real coded genetic algorithms and interval schemata, *Foundations of Genetic Algorithms II*, Morgan Kaufmann(1993), 187-202.
7. Folkers, J.S.: Ship operation and design. In: *Optimization and Design*, Avriel, M., Rijckaert, M.J., and Wilde, D.J. (eds.), (Prentice Hall)(1973), 221-226.
8. Fonseca, C. and Flemming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers, *Parallel Problem Solving from Nature IV*, (1995), 584-593.
9. Fonseca, C. and Flemming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, (Morgan Kaufmann)(1993), 416-423.
10. Fonseca, C.M. and Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, **3**(1)(1995), 1-16.
11. Fonseca, C.M. and Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms--Part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Systems and Humans, 28(1)(1998), 26-37.
12. Grefenstette, J.J.: GENESIS: A system for using genetic search procedure, *Proceedings of the 1984 Conference on Intelligent Systems and Machines*(1984), 161-165.
13. Hajela, P. and Lin, C.Y.: Genetic search strategies in multi-criterion optimal design. *Structural Optimization*, 4(1992), 99-107.
14. Hansen M.P., Jaszkiewicz A.: Evaluating the quality of approximations to the nondominated set. *Working paper*, Institute of Mathematical Modelling Technical University of Denmark, IMM-REP-1998-7, (1998).
15. Hinterding, R. and Michalewicz, Z.: Your brains and my beauty: parent matching for constrained optimisation, *Proceedings of the Fifth International Conference on Evolutionary Computation*, Anchorage, Alaska(1998), 810-815.
16. Horn, J., Nafpolitis, N. and Goldberg, D.E.: A niched Pareto genetic algorithm for multiobjective optimization, *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, Volume 1, (Piscataway, NJ, IEEE Service Center)(1984), 82-87.
17. Ishibuchi, H. and Murata, T.: Multiobjective genetic local search algorithm. *Proceedings of 1996 IEEE International Conference on Evolutionary Computation*(1996), 119-124.
18. Jaszkiewicz A. : On the performance of multiple objective genetic local search on the 0/1 knapsack problem. A comparative experiment, *Research Report*, Institute of Computing Science, Poznan University of Technology, RA-002/2000, (2000).
19. Jimenez, F. and Verdegay, J.L.: Constrained multiobjective optimization by evolutionary algorithms, *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems* (EIS'98), Tenerife, Spain(1998), 266-271.

20. Knowles, J.D. and Corne, D.W. : Approximating the nondominated front using the Pareto Archived Evolution Strategy, *Evolutionary Optimization*, 8(2), (2000),149-172.
21. Koziel, S. and Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, *Evolutionary Computation*, **7**(1)(1999), 19-44.
22. Laumanns, M. Zitzler, E. and Thiele,L. : A unified model for multiobjective evolutionary algorithm with elitism, *Proceedings of the Congress on Evolutionary Computation*, CEC 2000, IEEE Service Center, Piscataway(2000), 46-53.
23. Lis, J. and Eiben, A.E.: A multi-sexual genetic algorithm for multiobjective optimization, *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*(1997), 59-64.
24. Michalewicz, Z.: A survey of constraint handling techniques in evolutionary computation methods, *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, (MIT Press, Cambridge, MA)(1995), 135-155.
25. Osyczka, A. and Kundu, S.: A modified distance method for multicriteria optimization using genetic algorithms, *Computers in Industrial Engineering*, 30(4)(1996), 871-882.
26. Osyczka, A. and Kundu, S.: A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm, *Structural Optimization* 10, (Springer Verlag)(1995), 94-99.
27. Ray, T., Tai, K. and Seow, K.C.: An evolutionary algorithm for constrained optimization, *Proceedings of the Genetic and Evolutionary Computing Conference* (GECCO 2000), Las Vegas,(2000),771-777.
28. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms, *Proceedings of an International Conference on Genetic Algorithms and Their Applications*(1985), 93-100.
29. Schneekluth, H.: *Ship design for efficiency and economy*, Butterworths, 1987.
30. Sen, P., and Yang, J.B.: *Multiple criteria decision support in engineering design*, Springer-Verlag, London,1998.
31. Srinivas, N. and Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, **2**(3)(1994), 221-248.
32. Surry, P., Radcliffe, N.J. and Boyd, I.: A multiobjective approach to constrained optimization of gas supply networks, *Proceedings of the AISB-95 Workshop on Evolutionary* Computing, Vol. 993, (Springer Verlag)(1995), 166-180.
33. Valenzuela-Rendon, M. and Uresti-Charre, E.: A non-generational genetic algorithm for multiobjective optimization. *Proceedings of the Seventh International Conference on Genetic Algorithms*, San Fransisco, California(1997),658-665.
34. Veldhuizen, D.A.V. and Lamont,G.B.: On Measuring Multiobjective Evolutionary Algorithm Performance, *Congress on Evolutionary Computation*, Volume 1, IEEE Service Center, (2000), 204-211.

35. Zitzler E., Thiele L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, (1999), 257-271.