A Swarm Metaphor for Multiobjective Design Optimization

Tapabrata Ray and **K M Liew** *

Center for Advanced Numerical Engineering Simulations
School of Mechanical and Production Engineering,
Nanyang Technological University, Nanyang Avenue,
Singapore 639798.

Email: mtray@ntu.edu.sg ; mkmliew@ntu.edu.sg
Tel: 0065-790-4076; Fax: 0065-793-6763

* Corresponding Author

A Swarm Metaphor for Multiobjective Design Optimization

Abstract: In this paper we present a new optimization algorithm to solve multiobjective design optimization problems based on behavioral concepts similar to that of a real swarm. The individuals of a swarm update their flying direction through communication with their neighboring leaders with an aim to collectively attain a common goal. The success of the swarm is attributed to three fundamental processes: identification of a set of leaders, selection of a leader for information acquisition and finally a meaningful information transfer scheme. The proposed algorithm mimics the above behavioral processes of a real swarm. The algorithm employs a multilevel sieve to generate a set of leaders, a probabilistic crowding radius-based strategy for leader selection and a simple generational operator for information transfer. Two test problems, one with a discontinuous Pareto front and the other with a multi-modal Pareto front is solved to illustrate the capabilities of the algorithm in handling mathematically complex problems. Three well-studied engineering design optimization problems (unconstrained and constrained problems with continuous and discrete variables) are solved to illustrate the efficiency and applicability of the algorithm for multiobjective design optimization. The results clearly indicate that the swarm algorithm is capable of generating an extended Pareto front, consisting of well spread Pareto points with significantly less number of function evaluations when compared to the nondominated sorting genetic algorithm (NSGA).

Keywords: Multiobjective, Pareto Solutions, Swarm, and Constrained Optimization.

1. Introduction

Swarm strategies are fairly new methods that have received considerable attention over recent years as optimization methods for complex functions. Kennedy and Eberhart (1995) initially proposed the swarm strategy for optimization. Unlike conventional evolutionary methods, the swarm strategy is based on simulation of social behavior where each individual in a swarm adjusts its *flying* according to its own *flying* experience and companions' *flying* experience. The key to the success of such a strategy in solving an optimization problem lies on the mechanism of information sharing across individuals to collectively attain a common goal. After the initial concept was proposed, there have been comparative studies between swarm and other evolutionary strategies by Eberhart and Shi (1998) and Angeline (1998).

In any multiobjective optimization problem (in absence of preference information among the objectives) the goal is to arrive at a set of Pareto optimal solutions. It is preferable to have such solutions well spread along the Pareto front while maintaining diversity in the parametric space as it would to allow the designer to choose from a set of competing solutions. A comprehensive review on various multiobjective optimization methods appears in Fonseca and Flemming (1995). A more recent paper by Coello (1999) outlines various multiobjective optimization methods and lists their advantages and disadvantages. One may refer to the above references for the necessary background on multiobjective optimization, as it is not repeated here.

It is well known that the presence of constraints significantly affects the performance of all optimization algorithms including evolutionary search methods. There has been a number of approaches to handle constraints in the domain of evolutionary computing including rejection of infeasible individuals, penalty functions and their variants, repair methods, use of decoders, separate treatment of constraints and objectives and hybrid methods incorporating knowledge of constraint satisfaction. Michalewicz and Schoenauer (1996) provide a comprehensive review on constraint handling methods. All the methods have limited success as they are problem dependent and require a number of additional inputs. Penalty functions using static, dynamic or adaptive concepts have been developed over the years. All of them suffer from common problems of aggregation and scaling. Repair methods are based on additional function evaluations, while the decoders and special operators or constraint satisfaction methods are problem specific and cannot be used to model a generic constraint.

The use of Pareto ranks based on constraint violations is quite an innovative approach to handle constraints. Surry et al. (1995) applied the Pareto ranking scheme among constraints while fitness was used in the objective function space for the optimization of gas supply networks. Fonseca and Flemming (1998) proposed a unified formulation to handle multiple constraints and objectives based on Pareto ranking scheme. Ray et al. (2000) introduced a multiobjective evolutionary algorithm using Pareto ranking in both the objective and constraint domain along with partner matching strategies to efficiently handle a wide variety of single and multiobjective unconstrained and constrained optimization problems. The use of Pareto ranking to handle constraints on one hand eliminates the need for additional inputs while on the other adds on to the computational cost as Pareto ranking is computationally expensive and more so in presence of multiple constraints.

Having provided a brief discussion and relevant references on multiobjective and constraint modeling methods, the concept of the swarm is introduced. A swarm in the present context of an optimization problem is considered as a collection of individuals having a goal to reach the best value (minimum or maximum) of a function. In a multiobjective optimization problem, in absence of any preference among the objectives, the goal as already mentioned earlier is to arrive at a set of Pareto optimal solutions. Ideally, the set of Pareto solutions should maintain a uniform spread along the Pareto front and maintain diversity in the parametric space. A set of diverse solutions in the parametric space allows the designer to choose from a set of alternative designs. For a swarm strategy to be efficient for constrained problems there is additional information about the constraint satisfaction that needs to be meaningfully shared among the individuals. In an unconstrained problem, the nondominated rank of the solutions based on the objective values are used to generate the set of leaders (SOL) while a multilevel sieve is implemented to generate the SOL for constrained problems. From the SOL, a leader is probabilistically selected based on the crowding radius of the leaders (in the objective space). Leaders with less number of individuals around them have a higher probability of being selected for information sharing thus allowing the strategy to explore new areas and maintain a spread along the Pareto front. The information acquisition strategy is based on a simple generational operator that ensures all the individuals in the swarm are unique (based on the variable space) as in a real swarm, where at a given time instant two individuals cannot share the same location. The proposed algorithm is described in detail in the next section separately for unconstrained and constrained multiobjective problems.

2. Mathematical Formulation

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

Minimize $\quad f = [f_1(\boldsymbol{x}) \quad f_2(\boldsymbol{x}) \quad \cdots \quad f_m(\boldsymbol{x})]$

Subject to
$$g_i(\boldsymbol{x}) \geq a_I \ , \ i = \mathbf{1, 2, \ldots} \ q \ .$$
$$h_j(\boldsymbol{x}) = b_j \ , \ \mathrm{j} = \mathbf{1, 2, \ldots} \ r \ .$$

where there are q inequality and r equality constraints and $\boldsymbol{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]$ is the vector of n design variables.

It is a common practice to transform the equality constraints (with a tolerance δ) to a set of inequalities and use a unified formulation for all constraints:

$h_j(\boldsymbol{x}) \leq b_j + \delta$ which is same as $-h_j(\boldsymbol{x}) \geq -b_j - \delta$ and $h_j(\boldsymbol{x}) \geq b_j - \delta$.

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 individual, **c** denotes the constraint satisfaction vector given by **c** $=[c_1 \quad c_2 \quad \ldots \quad c_s]$ where

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

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. The CONSTRAINT matrix for a swarm of M individuals assumes the form

$$CONSTRAINT = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1s} \\ c_{21} & c_{22} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & & c_{Ms} \end{bmatrix}$$

The objective matrix assumes the form

$$OBJECTIVE = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1k} \\ f_{21} & f_{22} & \cdots & f_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & & f_{Mk} \end{bmatrix}$$

In a swarm of M individuals, all nondominated individuals are assigned a rank of 1. The rank 1 individuals are removed from the swarm and the new set of nondominated individuals is assigned a rank of 2. The process is continued until every individual in the swarm is assigned a rank. Rank=1 in the objective or the constraint matrix indicates that the individual is nondominated. It can be observed from the constraint matrix that when all the individuals in the swarm are infeasible, the Rank=1 solutions are the best in terms of minimal constraint violation. Whenever there is one or more feasible individual in the swarm, the feasible solutions assume the rank of 1.

The initial swarm consists of a collection of random individuals. Over time, the individuals communicate with the better performers and derive information from them. The pseudo code of the algorithm is presented below.

2.1 Algorithm

```
Initialize M unique individuals in the Swarm;
Unconstrained Problem Strategy
Do {
    Compute Objective Values of each Individual in the Swarm;
    Compute Ranks based on the Objective Matrix;
```

```
        Compute the Average Rank based on the Objective Matrix;
        Count the Number of Rank 1 Individuals;
        If (Number of Rank 1 Individuals  ≤  M/2)
              Assign Individuals to SOL if their Rank ≤ Average
              Rank;
        If (Number of Rank 1 Individuals  > M/2)
              Assign Rank 1 Individuals to SOL;
        For (Each Individual not in SOL) {
              Do {
                    Select a Leader from SOL to derive information;
                    Acquire information from the Leader and move to a
                    new point in the search space;
              } while (all individuals are not unique)
        }
} while (termination condition = False)
```

Constrained Problem Strategy

```
Do {
        Compute Objective values for each Individual in the Swarm;
        Compute Constraint values for each Individual in the Swarm;
        Use Non Dominated Sorting to Rank Individuals based on
        Objective Matrix;
        Use Non Dominated Sorting to Rank Individuals based on
        Constraint Matrix;
        Compute the Average Rank based on the Objective Matrix;
        Count the Number of Rank 1 Individuals based on the
        Objective Matrix;
        Compute the Average Rank based on the Constraint Matrix;
        Count the Number of Rank 1 Individuals based on the
        Constraint Matrix;
```

```
        Count the Number of Feasible Individuals;
        If Number of Individuals with Objective Rank=1 > M/2:
        Average Objective Rank=1;
        If Number of Individuals with Constraint Rank=1 > M/2:
        Average Constraint Rank=1;
        If Number of Feasible Individuals = 0 {
              Assign Individuals to SOL with Constraint Rank ≤
              Average Constraint Rank;
              Shrink the above SOL, to contain Individuals with
              Objective Rank ≤  Average Objective Rank;
        }
        If Number of Feasible Individuals > 0 {
              Assign Feasible Individuals to SOL;
              Shrink the above SOL, to contain Individuals with
              Objective Rank ≤ Average Objective Rank;
        }
        For (Each Individual not in SOL) {
              Do {
                    Select a Leader from the SOL to derive
                    information;
                    Acquire information from the Leader and move to a
                    new point in the search space;
              } while (all individuals are not unique)
        }
  } while (termination condition = False)
```

For a constrained problem, individuals with a constraint rank = 1 (nondominated based on constraint matrix) are the best performers based on constraint satisfaction. When there are no feasible individuals in the swarm, the SOL consists of all nondominated individuals based on the constraint matrix. When there is one or more feasible individuals, the SOL will consist of all feasible individuals as they will have a constraint rank = 1. The size of SOL is expected to grow, as more and more individuals become feasible. To maintain a selective pressure in such a feasible SOL, individuals are allowed to be members of SOL only if they are better than average performers based on objective ranks. This process ensures that there is a pressure maintained among the feasible SOL members to improve their objective performance to remain as SOL members.

2.2 Selection of a Leader from SOL

The selection of a leader from the SOL is an important element of this algorithm that results in the spread of points along the Pareto frontier and extended limits of the Pareto curve. For every member of SOL, a crowding radius is computed that is the average of the distance between its left and right neighbor (based on the objective space) in the swarm. SOL members with a smaller crowding radius indicate that there are more individuals in the objective space around it. Deb et al. (2000) implemented a similar crowding radius concept in their fast elitist NSGA implementation. Selection of a leader from the SOL is based on a roulette wheel scheme that ensures SOL members with a larger crowding radius have a higher probability of being selected as a leader. The process in turn results in a spread along the Pareto frontier.

2.3 Acquiring Information through the Generational Operator

A simple generational operator is used in this study to acquire information from a leader. The operator can result in a variable value even if it does not exist in either the individual or its leader, which is useful to avoid premature convergence. The probability of a variable value generated between an individual and its leader is 50%. The probability of a variable value generated between the lowerbound of the variable and the minimum among the individual and its leader or between the upperbound of the variable and the maximum among the individual and its leader is 25% each. In addition to the above process, a solution is regenerated if it is non-unique in the variable space. The above generational operator has been used in an evolutionary algorithm to solve single objective constrained optimization problems by Ray et al (2000) and for multiobjective engineering design optimization problems by Ray et al (2001). Though the present examples have been solved using the above operator, other generational operators like the simulated binary crossover (SBX) proposed by Deb and Kumar (1995) may well be used instead.

3. Examples

Two test functions one with a discontinuous Pareto front and the other with a multimodal Pareto front is taken up to illustrate the capabilities of the proposed algorithm to handle mathematically difficult problems. Three engineering design optimization problems are then solved to illustrate the efficiency and applicability of the algorithm for multiobjective design optimization problems.

3.1 Discontinuous Pareto Optimal Front

Deb (1999) introduced this multiobjective optimization problem that has a discontinuous Pareto optimal front.

Minimize $f_1(x) = x_1$

Minimize $f_2(x) = (1+10x_2)\left(1 - \left(\frac{x_1}{1+10x_2}\right)^2 - \frac{x_1 \sin(8\pi x_1)}{1+10x_2}\right)$

Subject to $0 \le x \le 1$.

The Pareto optimal front obtained by the swarm algorithm is presented in Figure 1. A discontinuous Pareto front is considered difficult to obtain by methods using function space niching and such fronts can only be generated by an effective diversification mechanism (Deb, 1999). The Pareto front consists of 40 points and was obtained after 29,572 function evaluations that compares well with the Pareto front obtained by Deb (1999) using 60,000 function evaluations.

Insert Figure 1

3.2 Multimodal Multiobjective Problem

This bimodal, two objective problem has also been introduced by Deb (1999). The problem is considered difficult as a local Pareto front exists at x_2=0.6, while the global front exists at x_2=0.2 (which is a sharp inverted spike). Deb (1999) obtained the global front in 41 instances out of 100 trials with a population size of 60 running for 100 generations. The proposed algorithm reached the global Pareto front consisting of 100 solutions using 2323 function evaluations in all 10 out of 10 trials when run with a swarm size of 100 flying for 1000 time steps.

Minimize $f_1(x) = x_1$

Minimize $f_2(x) = 2.0 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\}$

Subject to $0 < x_1 \le 1$ and $0 \le x_2 \le 1$.

Insert Figure 2

3.3 Design of a Welded Beam

This example deals with a welded beam that 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 3. This problem has been solved by Deb (1999) using a real coded GA with simulated binary crossover (SBX) with a population size of 100 running for 500 generations. The mathematical formulation of the problem is presented below.

Insert Figure 3

Insert Figure 4

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

Minimize $f_2(\boldsymbol{x}) = \delta(x)$

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

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}$$

$$\tau'' = \frac{MR}{J}$$

$$M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{{x_2}^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{{x_2}^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\boldsymbol{x}) = \frac{6PL}{x_4{x_3}^2}$$

$$\delta(\boldsymbol{x}) = \frac{4PL^3}{Ex_4{x_3}^3}$$

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

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.

Insert Figure 5

The Pareto front obtained by the swarm algorithm is presented in Figure 4. A swarm size of 100 flying for 500 time steps resulted the front. It consists of 42 points and was obtained after 18,389 evaluations. It can be observed from Figure 4 that the Pareto front is similar to the one presented by Deb (1999) and both of them have the same limits of the Pareto curve. The progress of the swarm is presented in Figure 5. It can be observed from Figure 5 that the swarm initially has about 30% feasible individuals while towards the end has about 70% to 85% feasible solutions indicating that the process of communication between the individuals and their leaders have helped to improve feasibility of the individuals in the swarm. The number of individuals in the Pareto front also increased from a mere 5% to approximately 40% of the swarm over the time period.

3.4 Design of a Disc Brake

This example deals with the design of a multiple disc brake and has been discussed by Osyczka and Kundu (1995). The objectives of the design are to minimize the mass of the brake and to minimize the stopping time. The variables are the inner radius of the discs, outer radius of the discs, the engaging force and the number of friction surfaces and are represented as x_1, x_2, x_3 and x_4 respectively. The constraints for the design include minimum distance between the radii, maximum length of the brake, pressure, temperature and torque limitations. The problem is a mixed, constrained, multiobjective problem. The mathematical details of the optimization problem formulation are presented below.

$$\text{Minimize } f_1(\boldsymbol{x}) = 4.9 \times 10^{-5}(x_2^2 - x_1^2)(x_4 - 1)$$

$$\text{Minimize } f_2(\boldsymbol{x}) = \frac{9.82 \times 10^6 (x_2^2 - x_1^2)}{x_3 x_4 (x_2^3 - x_1^3)}$$

Subject to

$$(x_2 - x_1) - 20 \geq 0$$

$$30 - 2.5(x_4 + 1) \geq 0$$

$$0.4 - \frac{x_3}{3.14(x_2^2 - x_1^2)} \geq 0$$

$$1 - \frac{2.22 \times 10^{-3} x_3 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)^2} \geq 0$$

$$\frac{2.66 \times 10^{-2} x_3 x_4 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)} - 900 \geq 0$$

where $55 \leq x_1 \leq 80$, $75 \leq x_2 \leq 110$, $1000 \leq x_3 \leq 3000$ and $2 \leq x_4 \leq 20$.

Insert Figure 6

The Pareto front is presented in Figure 6 as obtained with a swarm size of 500 flying for 20 time steps. It consists of 52 points and was obtained after 6,385 evaluations. The proposed algorithm results in an extended Pareto curve between (0.2, 32) and (2.7, 2) whereas Osyczka and Kundu (1995) reported 30 Pareto solutions between (1.7, 2.9) and (3.4, 2.1) with 10,000 function evaluations. It is interesting to note that the proposed algorithm generated the extended Pareto front with more number of evenly spread Pareto points using significantly less number of function evaluations.

3.5 Design of a Four Bar Truss

This problem has been introduced by Stadler and Dauer (1992). Cheng and Li (1999) solved the problem and obtained a single point in the Pareto front based on a generalized center method. Figure 7 illustrates the problem where the structural volume (f_1) and the displacement (f_2) at joint (2) are to be minimized subject to the stress constraints on the members. The cross sectional areas of the members 1, 2, 3 and 4 are the design variables represented as x_1, x_2, x_3 and x_4 respectively. Figure 8 presents the Pareto optimal front for the problem.

$$\text{Minimize } f_1(\boldsymbol{x}) = L(2x_1 + \sqrt{2}x_2 + \sqrt{2}x_3 + x_4)$$

$$\text{Minimize } f_2(\boldsymbol{x}) = \frac{FL}{E}\left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right)$$

$$\text{Subject to:} \quad F/\sigma \le x_1 \le 3F/\sigma$$

$$\sqrt{2}F/\sigma \le x_2 \le 3F/\sigma$$

$$\sqrt{2}F/\sigma \le x_3 \le 3F/\sigma$$

$$F/\sigma \le x_4 \le 3F/\sigma$$

where F = 10 kN, E = 2.00E05 kN/cm^2, L = 200 cm and σ = 10 kN/cm^2

Insert Figure 7

Insert Figure 8

Cheng and Li (1999) obtained a single solution on the Pareto optimal front while the present algorithm resulted in a set of solutions along the Pareto optimal curve between (3000, 0.0035) and (1400, 0.035) as shown in Figure 8. The algorithm with a swarm size of 100 flying for 100 time steps resulted in 91 Pareto optimal points after 2,525 function evaluations.

4. Summary and Conclusions

A new algorithm has been introduced in this paper that mimics the social behavior of a real swarm. The algorithm is capable of handling unconstrained and constrained multiobjective problems with continuous, discrete or mixed variables without having any restriction on the number of variables, constraints or objectives. Moreover the algorithm does not require any additional inputs or need further assumptions on functional form or continuity. The success of the swarm is attributed the identification of a set of leaders, selection of a leader for information acquisition and finally the effective information sharing between the leaders and the rest of the individuals. The above information sharing process results in the holistic improvement of swarm rather than a greedy search improving a few individuals of the swarm. Such a feature is effective for problems with multiple sub-optimal Pareto fronts.

The use of Pareto ranks to handle constraints eliminates the problem of scaling and aggregation at the expense of nondominated sorting. Nondominated sorting is a computationally expensive process. However, in solving problems in engineering design, it is meaningful to make use of all computed information to better guide the search, as objective function evaluations are equally (or a few times more) expensive.

Unlike a real swarm, where a leader is a neighbor in the variable space, a new leader selection process is introduced in this algorithm that is based on the objective space to help maintain a spread along the Pareto front. The probabilistic selection of a leader from the set of leaders is based on the crowding radius. The roulette wheel selection scheme ensures that the solutions having less number of individuals around them (in the objective space) have a greater chance of being selected as leaders. Subsequently, a simple generational operator is used to derive information from its leader. The process is effective in finding extended limits of the Pareto curve while maintaining a reasonable spread along the front.

The results of the mathematical test problems clearly illustrate that the swarm algorithm is capable of solving problems with multimodal Pareto fronts and even problems with discontinuous Pareto fronts. For the three engineering design optimization problems, the swarm algorithm consistently resulted in more number of Pareto points with fewer function evaluations. It also identified extended limits of the Pareto front while maintaining a good spread along the front.

5 Future Work

The preliminary results of the swarm algorithm are promising. New leader selection and individual-to-leader matching strategies are currently being investigated that can reduce the number of function evaluations while maintaining the same spread. The algorithm is also currently being tested on a suite of mathematical test functions for constrained and unconstrained multiobjective optimization to conclusively comment on its performance. Such results will be reported once the studies are complete.

References

1. Angeline, P. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and Performance Differences, *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, San Diego, pp. 601-610.
2. Cheng, F.Y. and Li, X.S. (1999). Generalized method for multiobjective engineering optimization, *Engineering Optimization*, Vol. 31, pp. 641-661.

3. Coello, C.A.C. (1999). An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends, *Proceedings of the 1999 Congress on Evolutionary Computation. Washington,* pp. 3-13.
4. Deb, K. (1999). Evolutionary algorithms for multi-criterion optimization in engineering design. *In: Evolutionary Algorithms in Engineering and Computer Science*. Wiley, pp. 135-161.
5. Deb, K. and Kumar, A. (1995). Real coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. *Complex Systems*. **9**(6), pp. 431-454.
6. Deb, K., Agrawal, S, Pratap, A., and Meyarivan, T. (2000). A fast ellitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA-II, *Proceedings of Parallel Problem Solving from Nature VI*, Springer, pp. 849-858.
7. Eberhart, R. and Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization, *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, Springer Verlag, pp. 611-618.
8. Fonseca, C.M. and Flemming, P.J. (1995). An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, **3**(1), pp. 1-16.
9. Fonseca, C.M. and Flemming, P.J. (1998). 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), pp. 26-37.

10. Kennedy, J., and Eberhart, R. C. (1995). Particle swarm optimization, *Proceedings of the 1995 IEEE International Conference on Neural Networks*, IEEE Service Center, Piscataway, NJ, pp. 1942-1948.

11. Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation*, 4(1): pp.1-32.

12. Osyczka, A. and Kundu, S. (1995). A Genetic Algorithm-Based Multicriteria Optimization Method, *Proceedings of First World Congress of Structural and Multidisciplinary Optimization*, Elsevier Science, pp. 909-914.

13. Ray, T., Tai, K. and Seow, K.C. (2000). An evolutionary algorithm for constrained optimization, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2000*, Las Vegas, Morgan Kaufmann Publishers, pp. 771-777.

14. Ray, T., Tai, K. and Seow, K.C. (2001). Multiobjective design optimization by an evolutionary algorithm, *Engineering Optimization*, 33(3), pp.399-424.

15. Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, **2**(3), pp. 221-248.

16. Stadler, W. and Dauer, J. (1992). Multicriteria optimization in engineering: A tutorial and survey. Structural optimization: status and future, *AIAA Journal*, pp. 209-249.

17. Surry, P., Radcliffe, N.J. and Boyd, I. (1995). A multiobjective approach to constrained optimization of gas supply networks, *Proceedings of the AISB-95 Workshop on Evolutionary Computing*, Springer Verlag, Vol. 993, pp. 166-180.

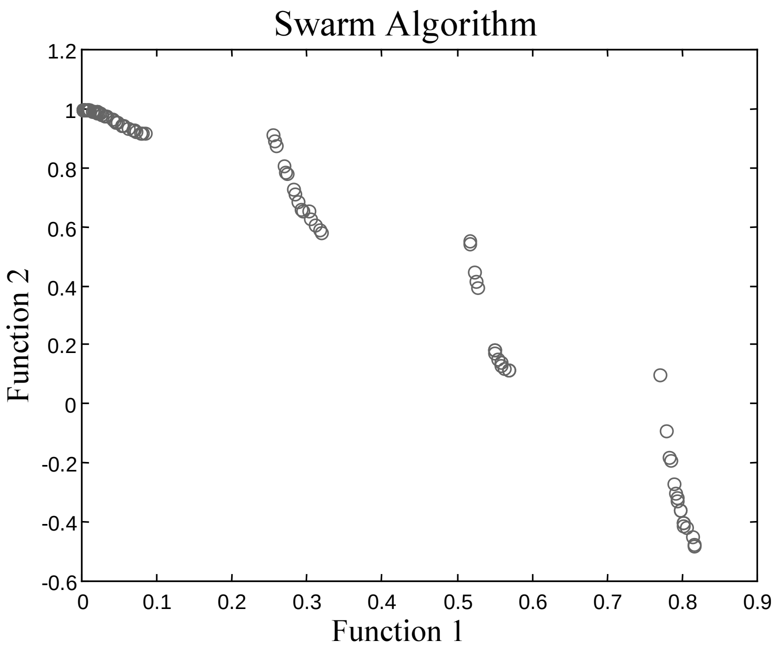

Figure 1 Discontinuons Pareto Front

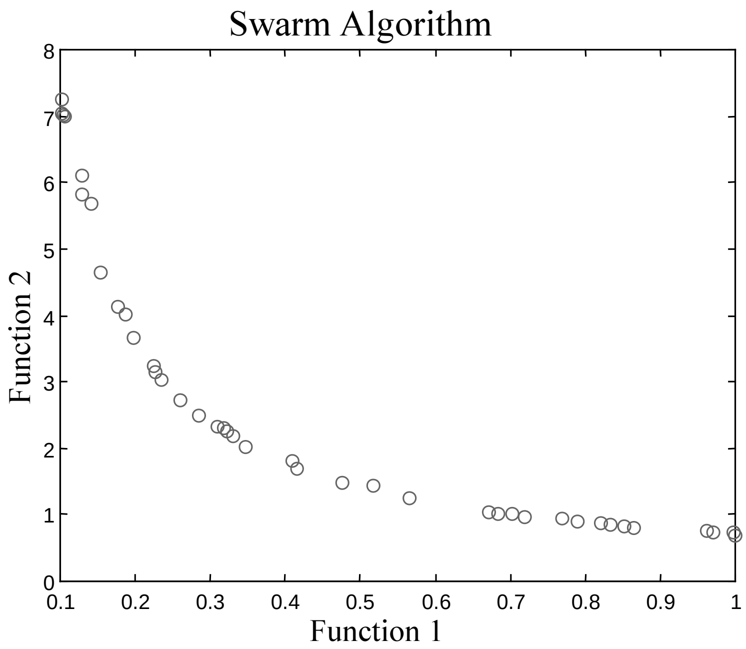

Figure 2 Global Pareto Front

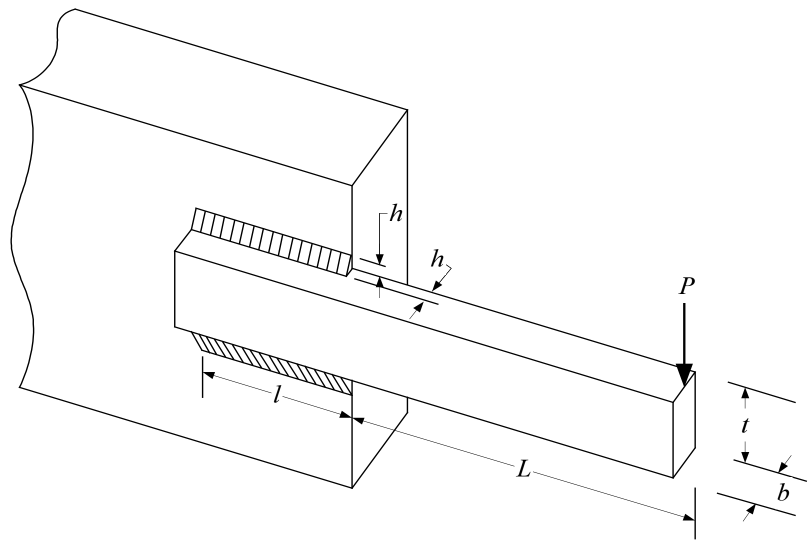

Figure 3 Welded Beam

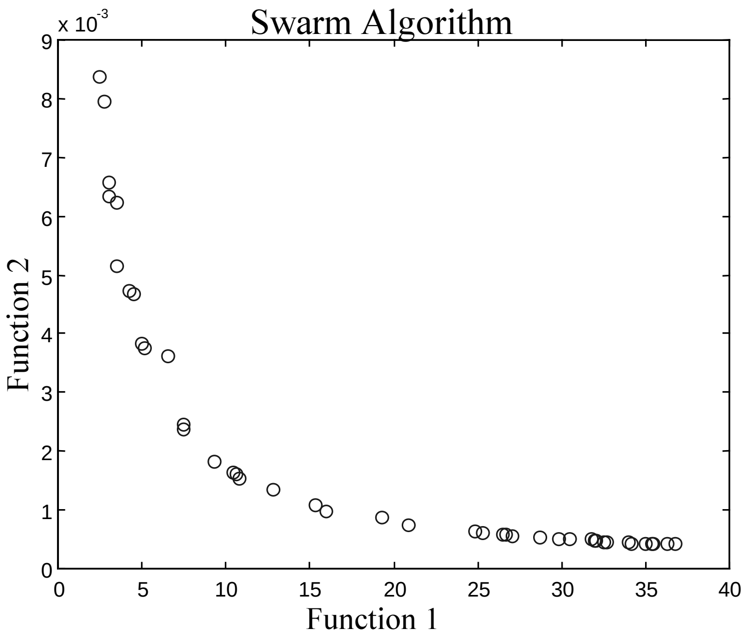

Figure 4 Pareto Front for the Welded Beam Design

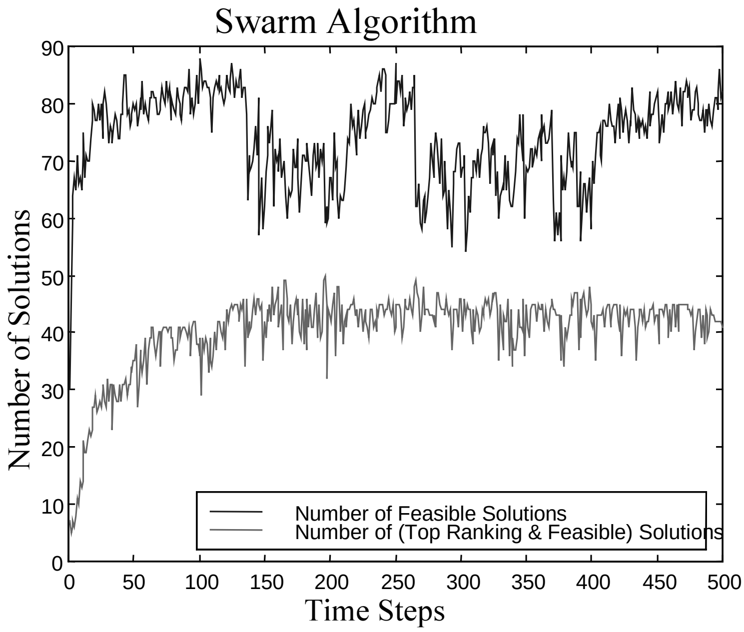

Figure 5 Progress of the Swarm with time

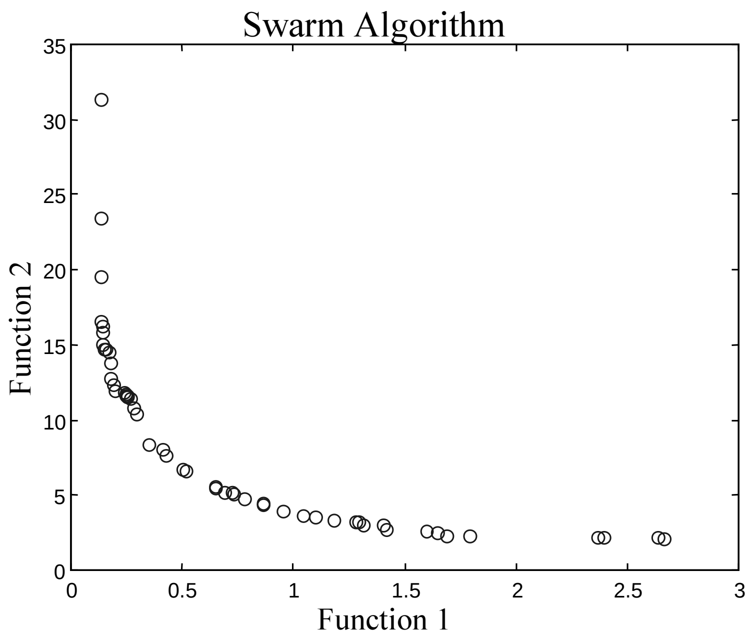

Figure 6 Pareto Front for the Disc Brake Design

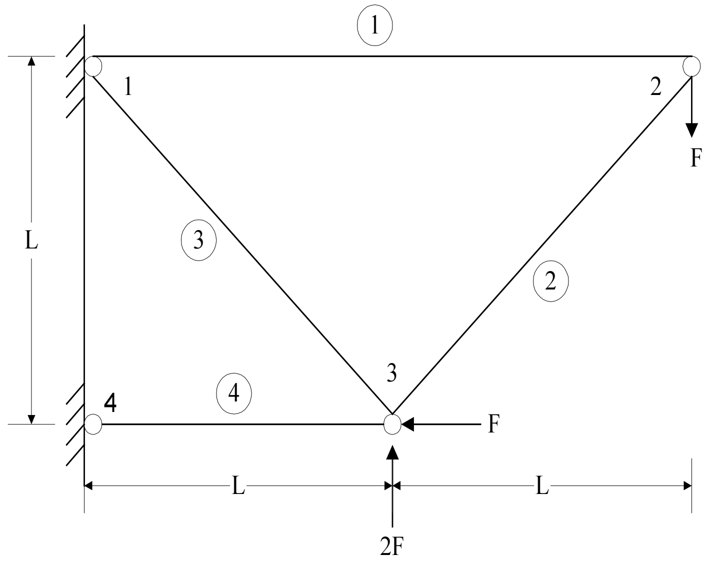

Figure 7 Four Bar Truss

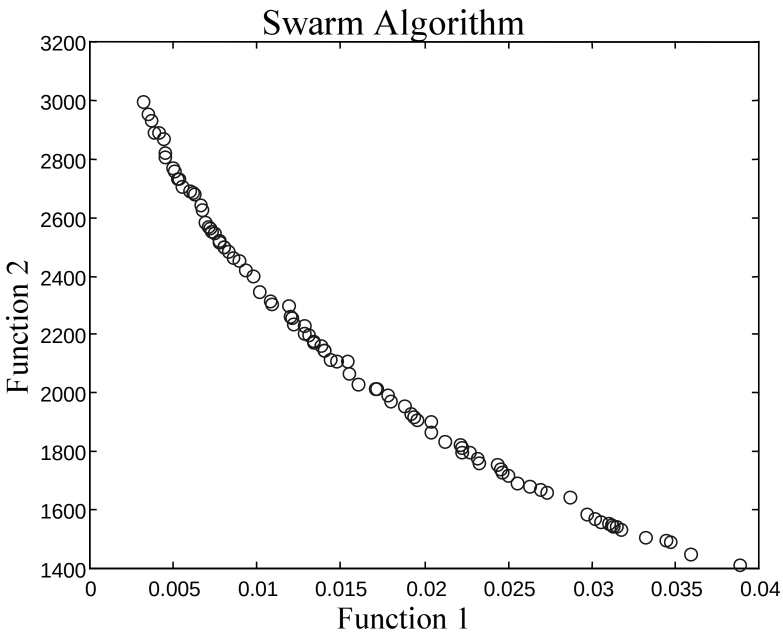

Figure 8 Pareto Front for the Four Bar Truss Design