HYBRID METHODS FOR MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Tushar Goel

John F. Welch Technology Centre
Export Promotion Industrial Park
Hoodi Village, Whitefield Road
Bangalore - 560 066
Tushar.Goel@geind.ge.com

Kalyanmoy Deb

Kanpur Genetic Algorithms Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
deb@iitk.ac.in

ABSTRACT

Hybrid methods of using evolutionary algorithms with a local search method are often used in the context of single-objective real-world optimization. In this paper, we discuss a couple of hybrid methods for multi-objective real-world optimization. In the *posteriori* approach, the obtained non-dominated solutions of a multi-objective evolutionary algorithm (MOEA) run are modified using a local search method. In the *online* approach, a local search method is applied to each solution obtained by genetic operations in a MOEA run. Both these approaches are compared on two engineering shape optimization problems for a fixed number of trials. Simulation results suggest important insights about the extent of local search and the extent of an MOEA needed to achieve an overall efficient hybrid approach.

1. INTRODUCTION

In an multi-objective optimization problem there exist more than one objectives. If the objectives are of conflicting in nature, one single solution cannot be the optimal solution. Instead, a set of solutions (known as Pareto-optimal set) is optimal. Since evolutionary methods work with a population of solutions, they have been found suitable to find multiple and well-diverse set of Pareto-optimal solutions in one single simulation run [1, 2, 6, 7]. In solving real-world problems, a straightforward application of an existing MOEA may not be efficient in finding the true Pareto optimal set. The argument is similar to that in single-objective evolutionary algorithms, in which an evolutionary algorithm (EA) is expected to run for some iterations. Thereafter, a local search method is started from the best solution obtained by the EA. This process is believed to be better for two reasons: (i) most optimum is unimodal (best solvable by a local search approach) in the neighborhood of the optimum and (ii) a hybrid method may use the efforts of both EAs and a local search approach in a way better than either approach alone.

In the proposed hybrid multi-objective optimization, we make use of a local search method and an MOEA. However, a local search method can only optimize a single objective. Thus, a careful hybridization of the two approaches is necessary to take advantage of better convergence and computational effort. In this paper, we discuss two implementations and compare their performances on a couple of engineering shape design problems. The simulation results provide interesting working behaviors of each hybrid approach.

2. HYBRID TECHNIQUES

There are at least two reasons why a hybrid method would be useful in real-world optimization problems, including multi-objective optimization: (i) it ensure better convergence to the global Pareto-optimal front and (ii) it demands smaller computational effort than each individual method applied alone.

In the case of real-world problems the knowledge about the Pareto-optimal front is not usually available. Though evolutionary algorithms have shown the potential of reaching close to the global Pareto-optimal front in many problems, it is wise to make use of a specialized method (local search method) to increase the probability of convergence to the global Pareto-optimal front. Since the local search methods have good convergence properties to a local optimal solution, and an EA has overall global perspective, a hybrid method of combining the two approaches is a natural choice. In one approach, an MOEA can be used to find good initial solutions for the local search method, which then can make an attempt to find the solutions even closer to the true Pareto-optimal front. Because of the same reason, the combined use of an MOEA and a local search method may result in a saving of computational effort, if used properly.

In this paper, we discuss two extreme hybrid approaches (i) posteriori approach and (ii) online approach.

2.1. Posteriori Approach

In this approach, we allow the multi-objective evolutionary algorithm to run for a fixed number of generations. This would produce a number of non-dominated solutions as the outcome. Then, a local search method is started from each of these solutions independently, as depicted in Figure 1. Since the local search requires a single objective, an aggregated objective function can be formed for each non-dominated solution. The authors suggested one such aggregate objective function in an earlier study [3]. We use the identical objective function here, although the concept can be used for other more generic aggregate functions, such as the Tchebycheff function etc. [1]. As the figure illustrates, the *location* of each solution in the objective space is used to form a weighted objective function. First, from the extreme non-dominated solutions, a pseudo-weight vector is derived for each intermediate solution. Thereafter, a weighted function is formed using the pseudo-weight vector, providing a direction of search in the objective space. Each solution is thus directed in a different direction and the best solution is found by a local search method.

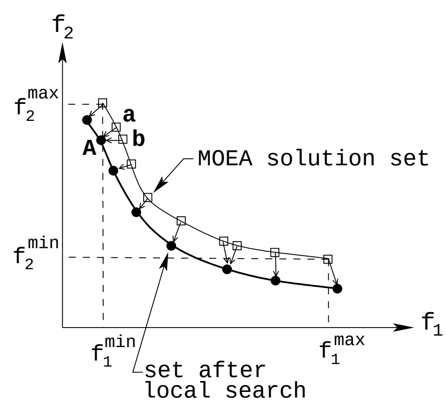

Figure 1: Posteriori approach is illustrated.

2.2. Online Approach

Other extreme of a hybrid approach is the use of a local search procedure within an MOEA. Every solution created by the genetic operators, is modified by the local search procedure before being accepted. Figure 2 illustrates the online approach with randomly-created search directions (motivated by other implementations (Ishibuchi and Murata [5])) for the local search approach. Once again a combined objective is used in such a case. Here, the local search approach is embedded in the MOEA. The overall computational effort (and function evaluations) will depend on the extent of local search method applied to each generated solution.

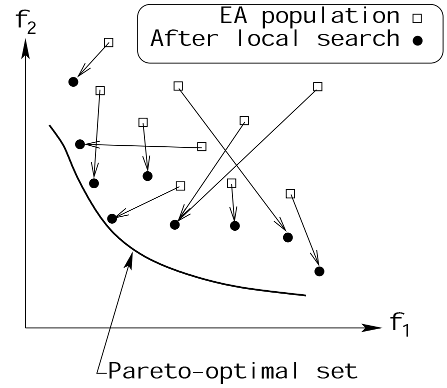

Figure 2: Online approach is illustrated.

The two methods discussed above are the two extreme cases. However, there may exist some other ways of combining a local search and an evolutionary method. The need is to find a balance between the extent of local search and the evolutionary method. The optimal setting of their extent will reduce the computational effort needed in solving a problem.

2.3. Relative Importance

In real-world problem solving, the computational time required for evaluating a solution (we call a function evaluation here) is the main concern for an optimum procedure. For the comparison of the above two approaches, it is, therefore, necessary to allocate a fixed overall number of function evaluations to each approach and then compare the quality of obtained solutions. With a fixed number of trials, we estimate the MOEA iterations to be allocated to each approach.

2.3.1. Posteriori Approach

Let the allowed number of overall function evaluations be F. We allocate F_C number of functions evaluations to each individual during each local search approach. Then, the number of function evaluations that can be allocated to the MOEA will be $F_{\rm GA}$. Then, we have

$$F_{\rm GA} = F - F_C \times N. \qquad (1)$$

In each generation, an MOEA takes N functions evaluations. Thus, the maximum number of generations ($t_{\rm post}$) that can be allowed to the MOEA are given by

$$t_{\rm post} = \frac{F_{\rm GA}}{N} = \frac{F - N \times F_c}{N} = \frac{F}{N} - F_C. \qquad (2)$$

If the extent of local search (or F_C) is more, MOEA is not allocated many iterations (or $t_{\rm post}$ is less).

2.3.2. *Online Approach*

Here, in each generation of an evolutionary method, we apply the local search on each individual. Then the total number of function evaluations in each generation F_{gen} is

$$F_{\text{gen}} = N \times F_C. \tag{3}$$

Thus, an MOEA is allocated

$$t_{\text{online}} = \frac{F}{N \times F_C} \tag{4}$$

number of iterations. If F_C is large, t_{online} will be small. In such a case, this method may not allow the evolutionary search to play a major role in the search for optimal solutions.

The simple comparison of equation 2 and equation 4 is given in the following:

$$\frac{t_{\text{post}}}{t_{\text{online}}} = \frac{F_C \times F_{\text{GA}}}{F}. \tag{5}$$

For $F_C \times F_{GA} > F$, we have $t_{\text{post}} > t_{\text{online}}$. Since an optimization method (whether a local search method or an EA) requires a substantial number of function evaluations to choose and compare solutions in finding a near-optimal solution, the product of F_C and F_{GA} is likely to be more than the allocated number of function evaluations, particularly in solving real-world problems. In such situations, the posteriori approach allows more number of MOEA generations than what would be allowed in the online approach.

3. OPTIMAL ENGINEERING SHAPE DESIGN

We have chosen a number of shape optimization problems to compare the performance of the above two hybrid approaches. We represent a shape by the presence or absence of small material elements covering a two-dimensional rectangular plate [4] (refer to Figure 3). Since the shape is represented by the presence or absence of the elements, which also act as a binary decision variable. The binary string corresponding to the shape shown in the Figure 4 is given as following :

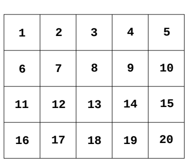

Figure 3: Rectangular plate divided into small elements.

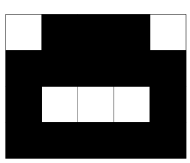

Figure 4: The skeleton of a shape.

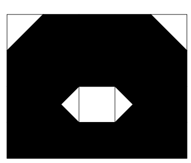

Figure 5: Final smoothen-ed shape.

```
01110 11111 10001 11111
```

Here the presence of the element is denoted by a 1 and the absence is shown by a 0. A left-to-right coding procedure as shown in Figure 3 is followed here. Since the strings are generated at random, it is possible to have some disconnected regions in the rectangular plate. For this case, we use the biggest cluster of the given shape of connected elements (where two elements are defined to be connected if they have at least one common corner). The string is repaired by assigning a 0 at all elements which are not part of the biggest cluster (Lamarckian approach). The shape obtained after finding the connected region are smoothened using triangular elements. For details, please refer [3].

The shapes represented by the binary strings are evaluated by the finite element analysis. For this the shape is further divided into small triangular elements, as we take constant strain triangle as the element for finite element analysis. All the interior elements are divided into two triangles and all the boundary elements (including the elements around a hole) are divided into four small triangles. The boundary triangles are also divided into smaller triangles. The linear shape functions in natural co-ordinates and isoparametric elements are used. The maximum stresses and the maximum displacements developed in the body under the action of loads are calculated. In all the applications here: weight and deflection, are taken as two minimization-type objectives. These are two conflicting objectives because a shape with a very small weight produces a large deflection and a shape with densely packed elements (large weight) tends to produce a very small deflection. The maximum stress and deflection developed in the body are restricted to lie within the specified limits decided by the user, by using them as the constraints.

4. SIMULATION RESULTS

We have used NSGA-II [2] as the MOEA in both hybrid approaches. Since the binary-coded strings are used, we use a bit-wise hill-climbing strategy as the local search approach. We start from the left of the string and flipped every bit, one at a time, to see if it improves the design. If it does, the change is accepted otherwise the original bit is restored. For the local search approach, we have considered following three levels:

LS 1 Here, we perform the local search till we reach the end of the binary string.

LS 2 After reaching the end of string we again begin flipping the first bit of the string and move unless we reach the end.

LS 3 Here, the hill-climbing is repeated three times on the complete string.

In all the simulation runs, we have used the following GA parameters.

Population Size	: 30,	Crossover Prob.	: 0.95,
Mutation Probability	: 1/String Length,		
Function Evaluations	: $F = 10,000$.		

For all the problems we have used following material properties:

Plate thickness	: 50 mm,	Yield strength	: 150 MPa,
Young's modulus	: 200 GPa,	Poisson's ratio	: 0.25.

4.1. Cantilever Plate

The first problem taken is the design of the cantilever plate, where an end load of $P = 10$ kN is applied. The rectangular plate of the size 60×100 mm^2 is divided into small rectangular elements. Hence, a 60-bit string is taken to represent the shape of the cantilever plate. The problem is solved by both the approaches. The results obtained with the posteriori approach are presented in the Figure 6. The figure

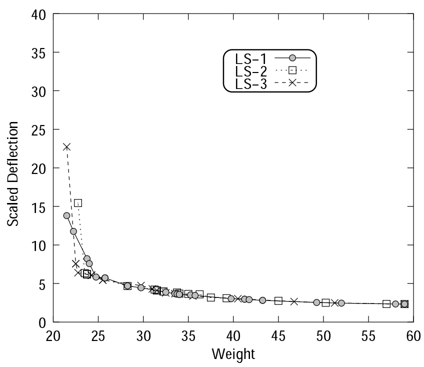

Figure 6: Solutions obtained by using different local search degrees in the posteriori approach for the cantilever plate design problem.

shows that increasing the importance of local search does not have too much effect on the overall performance, except near the minimum weight solutions. With more function evaluations allocated to the local search, better solutions are obtained near the minimum weight solutions. A gradual addition or deletion of crucial elements becomes necessary to find a near minimum weight solution. The steep slope of the non-dominated solutions near minimum weight solutions indicates that removal of one element from the structure can be allowed with a large sacrifice on the deflection, but the removal of the right element is important. A local search is ideal to obtain such a gradual search procedure.

Figure 7 shows the non-dominated fronts for the online approach. As the level of local search is increased, the performance of the online approach deteriorates. In the

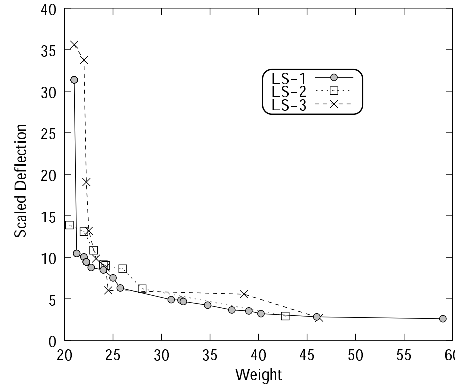

Figure 7: Solution set obtained by using different local search degrees in the online approach for cantilever plate design problem.

first level of local search, the emphasis on the evolutionary method is more than that in the third level of local search. The non-dominated front obtained by the third level of local search has worse diversity and convergence than that in the first level of local search. With a smaller extent of local search, an EA effectively gets more iterations. In this problem, the combination of one round of bit-flipping allows an adequate number of generations for the EA to find a better non-dominated front.

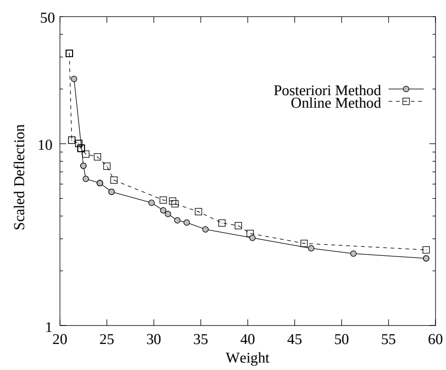

Figure 8: Non-dominated fronts obtained by the best of the posteriori (LS-3) and the online approach (LS-1) on the cantilever beam design problem.

Figure 8 shows the comparison of results obtained by the best of both methods. The posteriori method with the third level of local search is compared with the online approach having the first level of local search. It is clear that the posteriori method has a better diversity and the convergence. However, for small values of weights the convergence of the online approach based search is better.

4.2. Simply-Supported Plate

Next problem is the design of a simply-supported plate. The initial size of the plate is same as that taken in previous problem. A vertical load of $P = 10$ kN is applied on the top middle node of the plate. In the posteriori approach, the

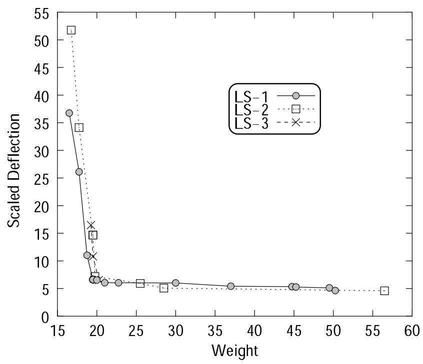

Figure 9: Solution set obtained for simply-supported plate design problem by using different local search levels in the online approach.

effect of different levels of local search is not found to be important, as also observed in the previous problem. The results obtained by the online approach (Figure 9) also follow a similar trend as in the first problem. The Pareto-optimal front obtained by the first level of local search and the front obtained by the second level of local search dominates each other in some region and get dominated by other in some other region. However, the performance of the third level of local search method is worst in terms of achieving both diversity and convergence. With the online approach, a small level of local search is better. Figure 10 shows the performance of the best of both approaches. It is clear that the posteriori approach have a better convergence and diversity in a wide range, but for very small weight cases the online approach is able to find better solutions.

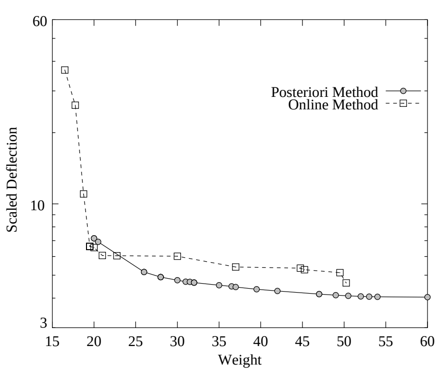

Figure 10: Non-dominated fronts obtained by the best of the posteriori (LS-3) and the online approach (LS-1) on the simply-supported beam design problem.

5. CONCLUSION

Simulation results on two engineering design problems show that the posteriori approach of hybridization is better than the online approach, as the former can obtain better convergence as well as better diversity. The main reason of deterioration of performance of online approach as the emphasis on the local search is increased is the little emphasis allocated to EA and more emphasis allocated to the local search method. It is also clear from the results that the optimum balance between the local search and the evolutionary search is essential to achieve the best results – good diversity and convergence to the global Pareto-optimal front. However, it is also evident that the hybrid methods are candidates of being a good and robust algorithm for solving the real-world multi-objective optimization problems.

6. REFERENCES

[1] Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, Chichester: Wiley.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[3] Deb, K. and Goel, T. (2000). A hybrid multi-objective evolutionary approach to engineering shape design. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001)*, pp. 385–399.

[4] Duda, J. W. and Jakiela, M. J. (1997). Generation and classification of structural topologies with genetic algorithm speciation. *ASME Journal of Mechanical Design, 119*, 127–131.

[5] Isibuchi, M. and Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and reviews, 28*(3), 392-403.

[6] Knowles, J. and Corne, D. (1999) The Pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway: New Jersey: IEEE Service Center, 98–105.

[7] Zitzler, E. and Thiele, L. (1998). An evolutionary algorithm for multi-objective optimization: The strength Pareto approach. *Technical Report No. 43 (May 1998)*. Zürich: Computer Engineering and Networks Laboratory, Switzerland.