A Hybrid Multi-objective Evolutionary Approach to Engineering Shape Design

Kalyanmoy Deb and Tushar Goel

Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
{deb,tusharg}@iitk.ac.in
http://www.iitk.ac.in/kangal

Abstract. Evolutionary optimization algorithms work with a population of solutions, instead of a single solution. Since multi-objective optimization problems give rise to a set of Pareto-optimal solutions, evolutionary optimization algorithms are ideal for handling multi-objective optimization problems. Over many years of research and application studies have produced a number of efficient multi-objective evolutionary algorithms (MOEAs), which are ready to be applied to real-world problems. In this paper, we propose a practical approach, which will enable an user to move closer to the true Pareto-optimal front and simultaneously reduce the size of the obtained non-dominated solution set. The efficacy of the proposed approach is demonstrated in solving a number of mechanical shape optimization problems, including a simply-supported plate design, a cantilever plate design, a hoister design, and a bicycle frame design. The results are interesting and suggest immediate application of the proposed technique in more complex engineering design problems.

1 Introduction

For last decade or so, a number of multi-objective optimization techniques using evolutionary algorithms are suggested [3,6,10,14,16,17]. The outcome of these studies is that different multi-objective optimization problems are possible to solve for the purpose of finding multiple Pareto-optimal solutions in one *single* simulation run. Classical means of finding one solution at a time with a weight vector or with a similar approach requires a priori knowledge of weight vector and need to be run many times, hopefully finding a different Pareto-optimal solution each time. In addition to converging close or on the true Pareto-optimal set, multi-objective evolutionary algorithms (MOEAs) are capable to finding a widely distributed set of solutions.

In this paper, we suggest a hybrid technique to take evolutionary multi-objective optimization procedures one step closer to practice. Specifically, in a real-world problem, we would like to ensure a better convergence to the true Pareto-optimal front and would also like to reduce the size of obtained non-dominated solutions to a reasonable number. The solutions obtained by an MOEA are modified using a local search method, in which a weighted objective function is minimized. The use of a local search method from the MOEA solutions will allow a better convergence to the true Pareto-optimal front.

E. Zitzler et al. (Eds.): EMO 2001, LNCS 1993, pp. 385–399, 2001.

A clustering method is suggested in general to reduce the size of the obtained set of solutions. For finite search space problems, the local search approach may itself reduce the size the the obtained set.

A specific MOEA—elitist non-dominated sorting GA or NSGA-II—and a hill-climbing local search method are used together to solve a number of engineering shape optimization problems for two objectives. Minimizing the weight of a structure and minimizing the maximum deflection of the structure have conflicting solutions. When these two objectives are considered together in a design, a number of Pareto-optimal solutions result. By representing presence and absence of small constituting elements in a binary string [1,2,5], NSGA-II uses an innovative crossover operator which seems to help in combining good partial solutions together to form bigger partial solutions. The finite element method is used to evaluate a string representing a shape. The paper shows how the proposed hybrid technique can find a number of solutions with different trade-offs between weight and deflection. On a cantilever plate design, a simply-supported plate design, a hoister plate design, and a bicycle frame design problem, the proposed technique finds interesting and well-engineered solutions. These results indicate that the proposed hybrid technique is ready to be applied to more complex engineering shape design problems.

2 Hybrid Approach

It has been established elsewhere that NSGA-II is an efficient procedure of finding a wide-spread as well as well-converged set of solutions in a multi-objective optimization problem [3,4]. NSGA-II uses (i) a faster non-dominated sorting approach, (ii) an elitist strategy, and (iii) no niching parameter. It has been shown elsewhere [3] that the above procedure has $O(MN^2)$ computational complexity. Here, we take NSGA-II a step closer to practice by

1. ensuring convergence closer to the true Pareto-optimal front, and
2. reducing the size of the obtained non-dominated set.

We illustrate both the above issues in the following subsections.

2.1 Converging Better

In a real-world problem, the knowledge of the Pareto-optimal front is usually not known. Although NSGA-II has demonstrated good convergence properties in test problems, we enhance the probability of its true convergence by using a hybrid approach. A local search strategy is suggested from each obtained solution of NSGA-II to find a better solution. Since a local search strategy requires a single objective function, a weighted objective or a Tchebyscheff metric or any other metric which will convert multiple objectives into a single objective can be used. In this study, we use a weighted objective:

$$F(\mathbf{x}) = \sum_{j=1}^{M} \bar{w}_j^{\mathbf{x}} f_j(\mathbf{x}), \tag{1}$$

where weights are calculated from the obtained set of solutions in a special way. First, the minimum $f_j^{\min}$ and maximum $f_j^{\max}$ values of each objective function f_j are noted. Thereafter, for any solution $\mathbf{x}$ in the obtained set, the weight for each objective function is calculated as follows:

$$\bar{w}_j^{\mathbf{x}} = \frac{(f_j^{\max} - f_j(\mathbf{x}))/(f_j^{\max} - f_j^{\min})}{\sum_{k=1}^{M}(f_k^{\max} - f_k(\mathbf{x}))/(f_k^{\max} - f_k^{\min})}. \tag{2}$$

In the above calculation, minimization of objective functions is assumed. When a solution $\mathbf{x}$ is close to the individual minimum of the function f_j, the numerator becomes one, causing a large value of the weight for this function. For an objective which has to be maximized, the term $(f_j^{\max} - f_j(\mathbf{x}))$ needs to be replaced with $(f_j^{\mathbf{x}} - f_j^{\min})$. The division of the numerator with the denominator ensures that the calculated weights are normalized or $\sum_{j=1}^{M} \bar{w}_i^{\mathbf{x}} = 1$. Once the pseudo-weights are calculated, the local search procedure is simple. Begin the search from each solution $\mathbf{x}$ independently with the purpose of optimizing $F(\mathbf{x})$. Figure 1 illustrates this procedure. Since, the pseudo-weight

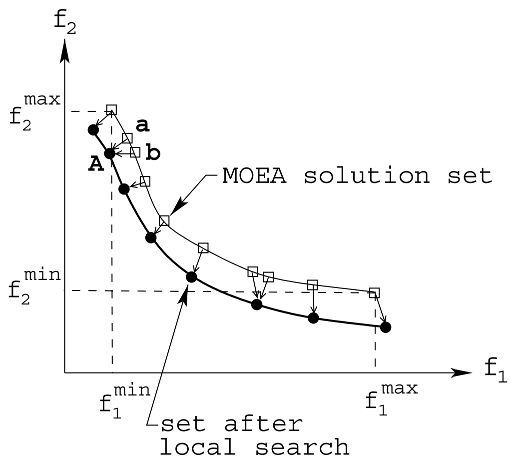

Fig. 1. The local search technique may find better solutions.

vector $\bar{\mathbf{w}}$ dictates roughly the priority of different objective functions at that solution, optimizing $F(\mathbf{x})$ will produce a Pareto-optimal or a near Pareto-optimal solution. This is true for convex Pareto-optimal regions. However, for non-convex Pareto-optimal regions, there exists no weight vector corresponding to Pareto-optimal solutions in certain regions. Thus, a different metric, such as Tchebysheff metric can be used in those cases. Nevertheless, the overall idea is that once NSGA-II finds a set of solutions close to the true Pareto-optimal region, we use a local search technique from each of these solutions with a differing emphasis of objective functions in the hope of better converging to the true Pareto-optimal front. Since independent local search methods are tried from each solution obtained using an MOEA, all optimized solutions obtained by the local search method need not be non-dominated to each other. Thus, we find the non-dominated set of solutions from the obtained set of solutions before proceeding further. Other studies as [11] use the local search method during a GA run. Each solution is modified with a local search method before including it in the population. The proposed approach is likely to have a lesser computational cost, however this will be a matter of future research to find a comparison between the two studies.

The complete procedure of the proposed hybrid strategy is shown in Figure 2. Starting from the MOEA results, we first apply a local search technique, followed by a non-domination check. After non-dominated solutions are found, a clustering technique is used to reduce the size of the optimal set, as discussed in the next subsection.

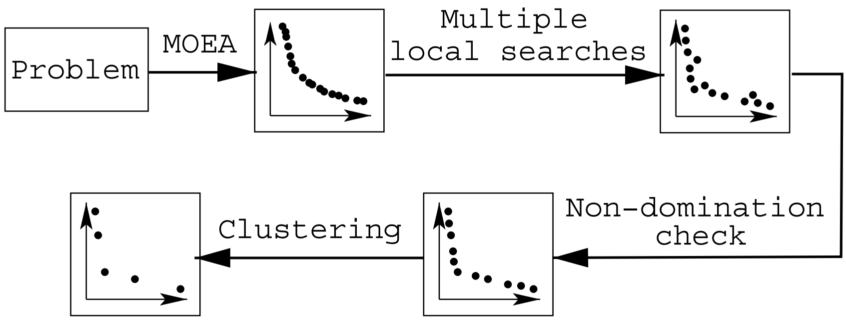

Fig. 2. The proposed hybrid procedure of using a local search technique, a non-domination check, and a clustering technique is illustrated.

2.2 Reducing the Size of Non-dominated Set

In an ideal scenario, an user is interested in finding a good spread of non-dominated solutions closer to the true Pareto-optimal front. From a practical standpoint, the user would be interested in a handful of solutions (in most cases, 5 to 10 solutions are probably enough). Interestingly, most MOEA studies use a population of size 100 or more, thereby finding about 100 different non-dominated solutions. The interesting question to ask is ‘Why are MOEAs set to find many more solutions than desired?’

The answer is fundamental to the working of an EA. The population size required in an EA depends on a number of factors related to the number of decision variables, the complexity of the problem, and others [7,9]. The population cannot be sized according to the desired number of non-dominated solutions in a problem. Since in most interesting problems, the number of decision variables are large and are complex, the population sizes used in solving those problems can be in hundreds. Such a population size is mandatory for the successful use of an EA. The irony is that when an MOEA works well with such a population size N, eventually it finds N different non-dominated solutions, particularly if the niching mechanism used in the MOEA is good. Thus, we need to devise a separate procedure of identifying a handful of solutions from the large obtained set of non-dominated solutions.

One approach would be to use a clustering technique similar to that used in [17] for reducing the size of the obtained non-dominated set of solutions. In this technique, each of N solutions is assumed to belong to a separate cluster. Thereafter, the distance d_c between all pairs of clusters is calculated by first finding the centroid of each cluster and then calculating the Euclidean distance between the centroids. Two clusters having the minimum distance are merged together into a bigger cluster. This procedure is continued till the desired number of clusters are identified. Finally, with the remaining clusters, the solution closest to the centroid of the cluster is retained and all other solutions from each cluster are deleted. This is how the clusters can be merged and the cardinality of the solution set can be reduced. Figure 3 shows the MOEA solution set in open boxes and the reduced set in solid boxes. Care may be taken to choose the extreme solutions in the extreme clusters.

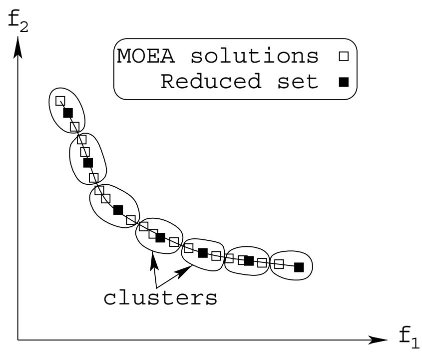

Fig. 3. The clustering method of reducing the set of non-dominated solutions is illustrated.

However, in many problems the local search strategy itself can reduce the cardinality of the obtained set of non-dominated solutions. This will particularly happen in problems with a discrete search space. For two closely located solutions, the pseudo-weight vectors may not be very different. Thus, when a local search procedure is started from each of these solutions (which are close to each other) with a $F(\mathbf{x})$ which is also similar, the resulting optimum solutions may be identical in a discrete search space problem. The solutions a and b in Figure 1 are close and after the local search procedure they may converge to the same solution A. Thus, for many solutions obtained using NSGA-II, the resulting optimum obtained using the local search method may be the same. Thus, the local search procedure itself may reduce the size of the obtained non-dominated solutions in problems with a finite search space. Figure 2 shows that clustering is the final operation of the proposed hybrid strategy.

3 Engineering Shape Design

With the advent of evolutionary algorithm as an alternate optimization method, there exist a number of applications of optimal shape design, where shapes are evolved by deciding presence or absence of a number of small elements [1,2,8,12,15]. A predefined area (or volume) is divided into a number of small regular elements. The task of an evolutionary optimization procedure is to find which elements should be kept and which should be thrown away so that the resulting shape is optimal with respect to an objective function. This procedure has a number of advantages:

1. The use of numerical finite element method (or boundary element method) is an usual method of analyzing an engineering component. Since finite element method procedure requires the component to be divided into a number of small elements, this approach reduces one computation step and is complimentary to the usual finite element method.
2. Since no a priori knowledge about the shape is required, this method does not have any bias from the user.
3. By simply using three-dimensional elements, the approach can be extended to three-dimensional shape design problems.
4. The number and shape of holes in a component can evolve naturally without explicitly fixing them by the user.

Most studies of this method, including the studies with evolutionary algorithms, have concentrated on optimizing a single objective. In this study, we apply this evolutionary procedure for multiple conflicting objectives.

3.1 Representation

In this study, we consider two-dimensional shape design problems only. However, the procedure can be easily extended to three-dimensional shape design problems as well. We begin with a rectangular plate, describing the maximum overall region, where the shapes will be confined. Thereafter, we divide the rectangular plate into a finite number of small elements (refer to Figure 4). We consider here square elements, although any other shape including triangular or rectangular elements can also be considered. Since the presence or absence of every element is a decision variable, we use a binary coding describing a shape. For the shape shown in Figure 5, the corresponding binary coding is as follows:

```
01110 11111 10001 11111
```

The presence is denoted by a `1` and the absence is shown by a `0`. A left-to-right coding procedure as shown in Figure 4 is adopted here. In order to smoothen the stair-case like shape denoted by the basic skeleton representation, we add triangular elements (shown shaded) for different cases in Figure 7. The resulting skeleton shape shown in Figure 5 represents the true shape shown in Figure 6.

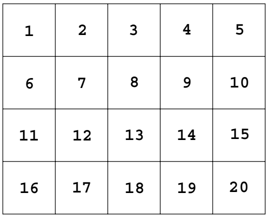

Fig. 4. Rectangular plate divided into small elements.

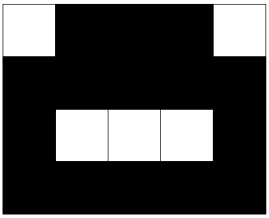

Fig. 5. The skeleton of a shape.

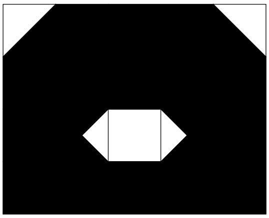

Fig. 6. Final smoothened shape.

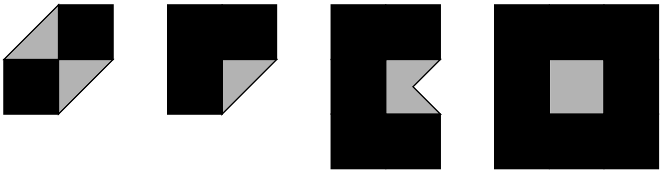

Fig. 7. Different cases of smoothening through triangular elements.

3.2 Evaluation

When the shape is smoothened, the shape is further divided into smaller elements. All interior rectangular elements are divided into two triangles and all boundary elements (including elements around a hole) are divided into four small triangles. Even the boundary triangles used for smoothening is divided into smaller triangles. The shape is evaluated by finding the maximum stress and deflection developed at any point in the component by the application of the specified loads. Since no connectivity check is made while creating a new string or while creating the initial random population, a string may represent a number of disconnected regions in the rectangle. In this case, we proceed with the biggest cluster 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.

In all applications here, two conflicting objectives are chosen: weight and deflection. These two objectives are conflicting because a minimum weight design is usually not stiff and produces a large deflection, whereas a minimum deflection design has densely packed elements, thereby causing a large weight of the overall component. The maximum stress and deflection values are restricted to lie within specified limits of the design by using them as constraints.

4 Simulation Results

To show the efficacy of the proposed hybrid multi-objective optimization procedure in solving optimal shape design problems, we use a number of mechanical component design problems. Since binary-coded strings are used to represent a shape, we use a bit-wise hill-climbing strategy as the local search operator. The procedure is simple. Starting from the left of the string, every bit is flipped to see if it improves the design. If it does, the flipped bit is retained, else the bit is unchanged. This procedure is continued until no bit-flipping over the the entire string length has resulted an improvement.

Since the shapes are represented in a two-dimensional grid, we introduce a new crossover operator which respects the rows or columns of two parents. Whether to swap rows or columns are decided with a probability 0.5. Each row or column is swapped with a probability $0.95/d$, where d is the number of rows or columns, as the case may be. This way on an average all most one row or column will get swapped between the parents. A bit-wise mutation with a probability of 1/string-length are used. NSGA-II is continued till 150 generations. It is important to highlight that NSGA-II does not require any extra parameter setting. In all problems, a population of size 30 is used.

For all problems, we use the 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 Design

First, we consider a cantilever plate design problem, where an end load $P = 10$ kN is applied as shown in Figure 8. The rectangular plate of size 60×100 mm^2 is divided into 60 small rectangular elements. Thus, 60 bits are used construct a binary string representing a shape.

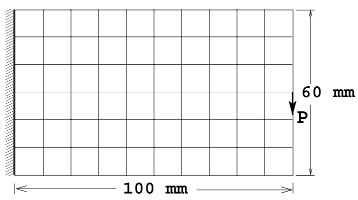

Fig. 8. The loading and support of the cantilever plate are shown.

Figure 9 shows the four steps of the proposed hybrid method in designing the cantilever plate. First plot shows the non-dominated solutions obtained using NSGA-II.

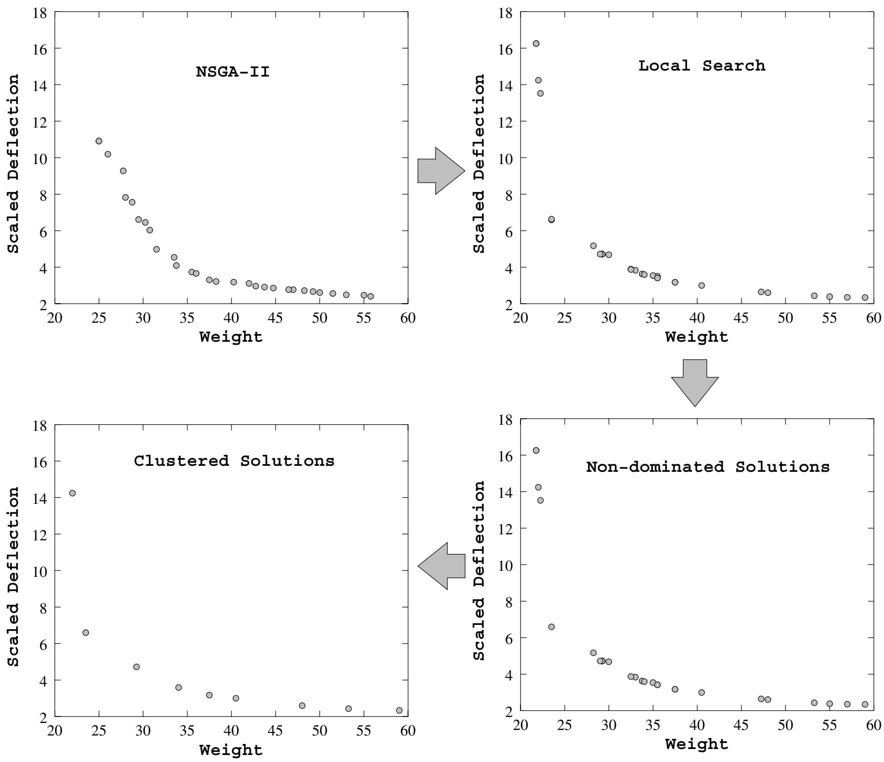

Fig. 9. Hybrid procedure to find nine trade-off solutions for the cantilever plate design problem.

Since the population size is 30, NSGA-II is able to find 30 different non-dominated solutions. Thereafter, the local search method is applied from each non-dominated solution and new and improved set of solutions are obtained. The third plot is the result of the non-dominated check of the solutions obtained after the local search method. Three dominated solutions are eliminated by this process. The final plot is obtained after the clustering operation with a choice of nine solutions. The plot shows how nine well distributed set of solutions are found from the third plot of 27 solutions. If fewer than nine solutions are desired, the clustering mechanism can be set accordingly.

In order to visualize the obtained set of nine solutions having a wide range of trade-offs in the weight and scaled deflection values, we show the shapes in Figure 10. It is clear that starting from a low-weight solution (with large deflection), how large-weight (with small deflection) shapes are found by the hybrid method. It is interesting to note that the minimum weight solution eliminated one complete row (the bottom-most row) in order to reduce the overall weight. The second solution (the element (1,2) in the above 3×3 matrix) corresponds to the second-best weight solution. It is well known that for an end load cantilever plate, a parabolic shape is optimal. Both shapes (elements (1,1) and

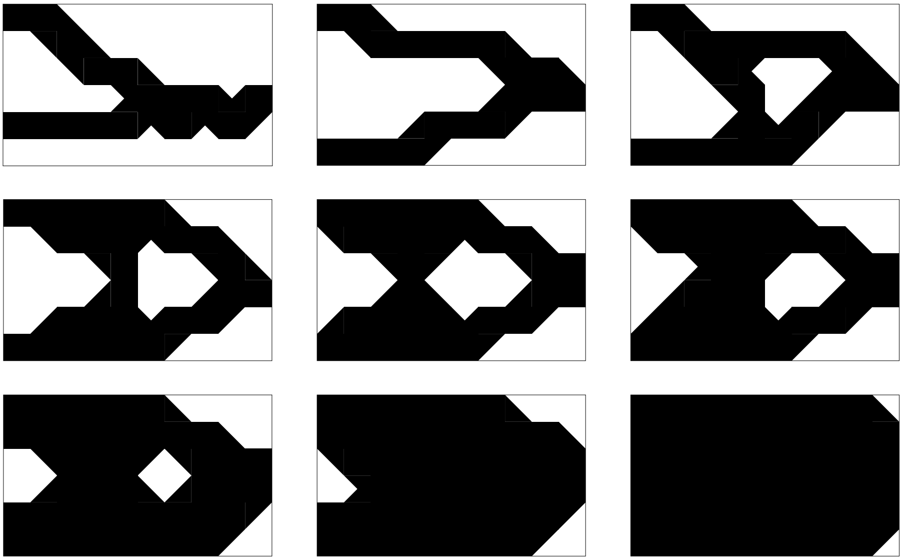

Fig. 10. Nine trade-off shapes for the cantilever plate design.

(1,2)) exhibits a similar shape. As the importance of deflection increases, the shapes tend to retain more and more elements, thereby making the plate rigid enough to have smaller deflection. In the middle, the development of vertical stiffener is interesting. This is a compromise between the minimum weight solution and a minimum deflection solution. By adding a stiffener the weight of the structure does not increase much, whereas the stiffness of plate increases (hence the deflection reduces). Finally, the complete plate with right top and bottom ends chopped off is the minimum deflection solution.

We would like to reiterate here that the above nine solutions are not results of multiple runs of a multi-objective optimization algorithm. All nine solutions (and if needed, more can also be obtained) with interesting trade-offs between weight and deflection are obtained using in one simulation run of the hybrid method.

4.2 Simply-Supported Plate Design

Next, we consider a simply-supported plate design, starting from a rectangular plate of identical dimension as in the previous design. The plate is supported on two supports as shown in Figure 11 and a vertical load $P = 10$ kN is acted on the top-middle node of the plate.

Figure 12 shows the obtained non-dominated solutions using NSGA-II. After local search method, the obtained non-dominated solutions have a wider distribution. The number of solutions have been reduced from 30 solutions to 22 solutions by the non-dominated checking. Finally, the clustering algorithm finds nine widely separated solutions from 22 non-dominated solutions. The shape of these nine solutions are shown in Figure 13. The minimum weight solution tends to use one row (the top-most row) less,

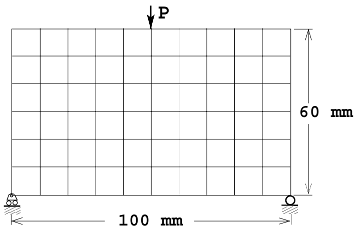

Fig. 11. The loading and support of the simply-supported plate are shown.

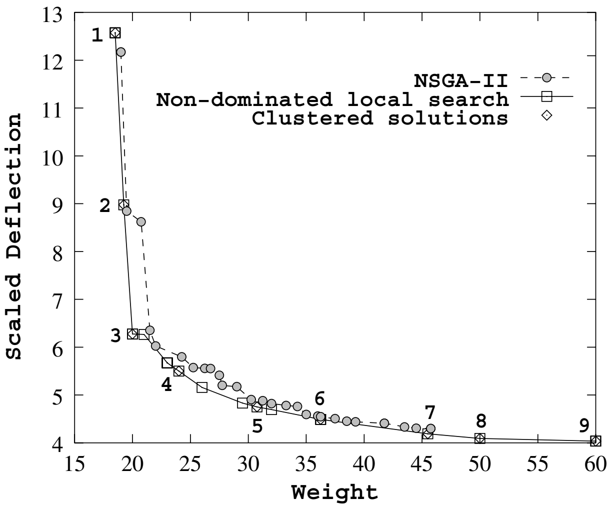

Fig. 12. Hybrid procedure finds nine trade-off solutions for the simply-supported plate design problem.

but since the load is acting on the top of the plate, one element is added to have the load transferred to the plate. The third solution (shown in the (1,3)-th position in the matrix) is interesting. A careful look at Figure 12 reveals that this solution is a 'knee' solution. To achieve a small advantage in weight-loss, a large sacrifice in the deflection-gain is evident. Similarly, to achieve a small advantage in deflection-loss, a large sacrifice in weight is needed. Shapes in position (1,2) and (2,1) can be compared with respect to the shape in position (1,3). Shape in position (3,1) or solution 7 is also interesting. In order to have further reduction in deflection stiffening of the two slanted arms is needed. Finally, the absolute minimum deflection shape is the complete rectangle with maximum possible weight.

Starting with the minimum weight design having two slim slanted legs down to thickening the legs to make them stiff, followed by joining the legs with a stiffener,

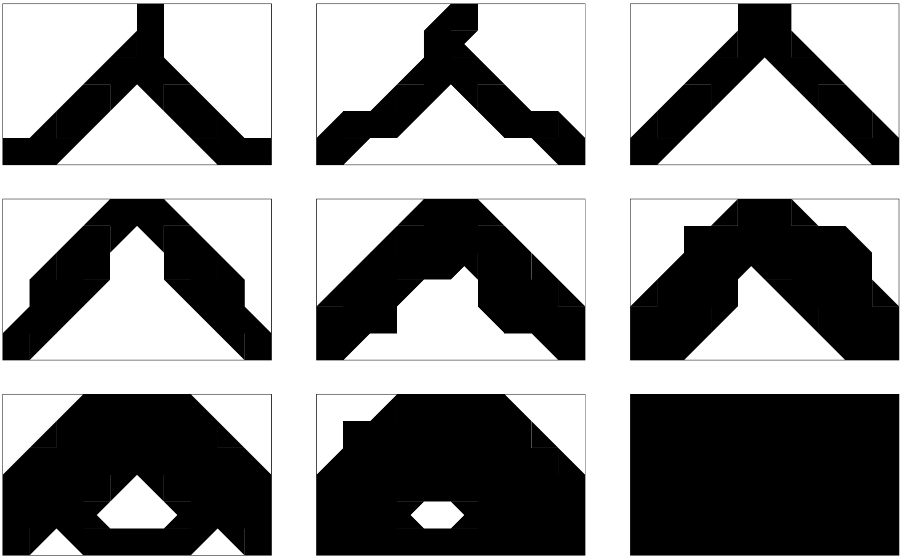

Fig. 13. Nine trade-off shapes for the simply-supported plate design.

and finally finding the complete rectangular plate having minimum deflection are all intuitive trade-off solutions. In the absence of any such knowledge, it is interesting how the hybrid procedure with NSGA-II is able to find the whole family of different trade-off solutions.

4.3 Bicycle Frame Design

Finally, we attempt to design a bicycle frame for a vertical load of 10 kN applied at A in Figure 14. The specifications are similar to that used elsewhere [13]. The plate is 20 mm thick and is restricted to be designed within in the area shown in Figure 14. The frame is supported at two places B and C. The point B marks the position of the axle of the rear wheel and the point C is the location of the handle support. The filled element is the location of the pedal assembly and is always present. The material yield stress is 140 MPa, Young's modulus is 80 GPa and Poisson's ratio is 0.25. The maximum allowed displacement is 5 mm.

Figure 15 shows the NSGA-II solutions and corresponding solutions obtained by the hybrid approach. Here, we are interested in finding four different trade-off solutions.

These four solutions obtained by NSGA-II are shown mounted on a sketch of a bicycle in Figure 16. The top-left solution is the minimum weight design. The second solution joins the two vertical legs to make the structure more stiff. The other two solutions make the legs more thick in order to increase the stiffness of the frame. The interior hole and absence of top-left elements are all intuitive. The proposed hybrid approach can evolve such solutions without these knowledge and mainly by finding and

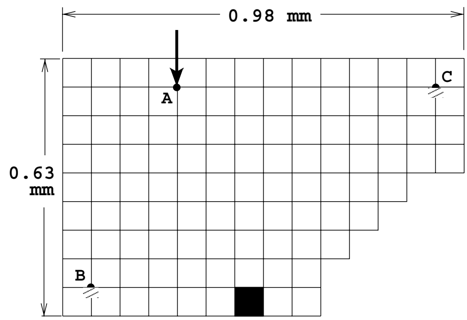

Fig. 14. The hybrid procedure is illustrated for the bicycle frame design.

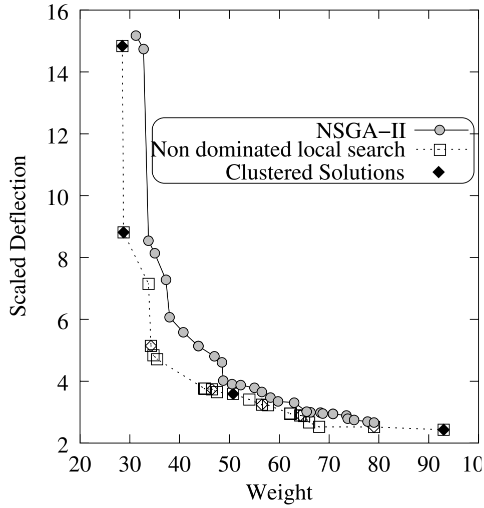

Fig. 15. The hybrid procedure is illustrated for the bicycle frame design.

maintaining trade-off solutions among weight and deflection. The presence of many such solutions with different trade-offs between weight and stiffness provides a plethora of information about various types of design.

5 Conclusion

The hybrid multi-objective optimization technique proposed in this paper uses a combination of an multi-objective evolutionary algorithm (MOEA) and a local search operator. The proposed technique ensures a better convergence of MOEAs to the true

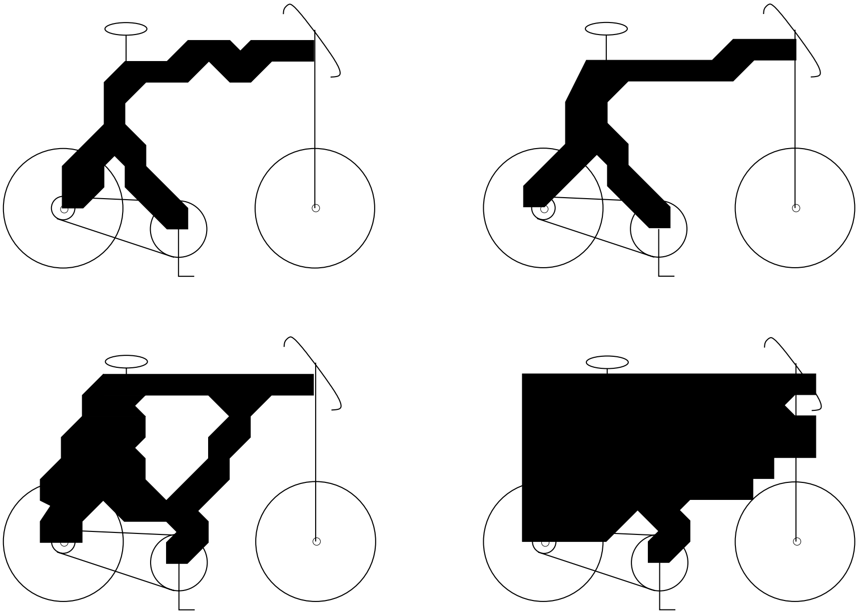

Fig. 16. Four trade-off shapes for the bicycle frame design.

Pareto-optimal region and helps in finding a small set of diverse solutions for practical reasons.

The efficacy of the proposed technique is demonstrated by solving a number of engineering shape design problems for two conflicting objectives—weight of the structure and maximum deflection of the structure. In all cases, the proposed technique has been shown to find a set of four to nine diverse solutions better converged than an MOEA alone. The results are encouraging and takes the evolutionary multi-objective optimization approach much closer to practice.

Acknowledgements. Authors acknowledge the support provided by Ministry of Human Resource Development (MHRD) for this study.

References

1. Chapman, C. D., Jakiela, M. J. (1996). Genetic algorithms based structural topology design with compliance and topology simplification considerations. *ASME Journal of Mechanical Design, 118*, 89–98.
2. Chapman, C. D., Saitou, K., Jakiela, M. J. (1994). Genetic algorithms as an approach to configuration and topology design. *ASME Journal of Mechanical Design, 116*, 1005–1012.
3. Deb, K., Pratap, A., Agrawal, S. and Meyarivan, T. (2000). A fast and elitist multi-objective genetic algorithm: NSGA-II. Technical Report No. 2000001. Kanpur: Indian Institute of Technology Kanpur, India.

4. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849–858.
5. 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.
6. Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion, and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*. 416–423.
7. Goldberg, D. E., Deb, K., and Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems, 6*, 333–362.
8. Hamada, H. and Schoenauer, M. (2000). Adaptive techniques for evolutionary optimum design. *Proceedings of the Evolutionary Design and Manufacture.*, pp. 123–136.
9. Harik, G., Cantu-paz, E. Goldberg, D. E., and Miller, B. L. (1999). The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation, 7*(3), 231–254.
10. Horn, J. and Nafploitis, N., and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multi-objective optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation*. 82–87.
11. 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.
12. Jakiela, M. J., Chapman, C., Duda, J., Adewuya, A., abd Saitou, K. (2000). Continuum structural topology design with genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*, 339–356.
13. Kim, H., Querin, O. M., and Steven, G. P. (2000). Post-processing of the two-dimensional evolutionary structural optimization topologies. In I. Parmee (Ed.) *Evolutionary Design and Manufacture*, London: Springer. pp. 33–44.
14. 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.
15. Sandgren, E., Jensen, E, and Welton, J. (1990). Topological design of structural components using genetic optimization methods. *Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers*, pp. 31–43.
16. Srinivas, N. and Deb, K. (1995). Multi-Objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*(2), 221–248.
17. 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.