A Multi-Objective Search and Optimization Procedure with Successive Approximate Models

Pawan K. S. Nain and Kalyanmoy Deb

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

KanGAL Report Number 2004012

Abstract. This paper explores the possibility of using approximate model for fitness landscape in multi-objective optimization. A multi-objective genetic algorithm based optimizer, namely, NSGA-II is integrated with artificial neural network (ANN). This presented technique makes use of successive fitness landscape modelling for reducing the precise function evaluation calls while retaining the basic robust search capability of genetic algorithms (GA). Such a technique finds application is computationally expensive real world optimization problems. To test this procedure, problems which cover various practical aspects of optimization problems are used. The procedure is tried on some of the standard test problems available in literature on multi-objective optimization. A shape optimization problem for mechanical component design is also taken. The simulation results show a considerable savings in precise function evaluations and a good diversity in the obtained front. finally, the issue of replacing ANN by some other approximation techniques is briefly discussed.

1 Introduction

Often the real world optimization problems require high computational loads. This computational load increases further for multi-objective optimization problems. Most real world optimization problems need robust optimizer with proven search capabilities. Evolutionary optimization methods are often used for such problems. Evolutionary optimization methods are inherently capable of dealing with mixed kind of variables (real, binary or integer), which makes them more useful tool for practical optimization. Generally, all evolutionary optimizers require large number of function evaluation in order to reach the optima or near optima solutions. Sometimes, the time required for single precise function evaluation of the problem can take few hours to several days. Computational fluid dynamics problem and finite element analysis based problems are two such examples. Under such conditions, a strong need to reduce the number of precise function evaluations during the course of optimization is felt. This problem can be handled by using an approximate fitness landscape model of the original problem [1, 3]. However, the use of approximation model raises few questions. The first question is that whether it will be able to reach the same optima as that of original problem. This question is basically related to the capability of the approximation technique which is used for generating the model of the original problem. The second question is related to multi-objective optimization. In multi-objective optimization, the final outcome of the optimizer is a set of solutions which is called the efficient set or Pareto-Optimal set. The second question is about the diversity in the final set of optimal solutions. If the diversity among the efficient set achieved by the original problem and that of solved with approximate model is comparable then the use of approximate model can be said to be successful. Lastly, the parameters which arise due to integration of approximation technique and the multi-objective optimizer can also play important roles.

In remainder of this paper, we first briefly discuss the previous work on the present topic. Next, the approximation based optimization procedure is introduced, which is tested on some multi-objective test problems and a finite element based shape optimization problem. Different simulations show encouraging results for the practical use of this method. In the next section, some issues regarding integrating other approximation techniques are briefly discussed. Lastly, some conclusions and recommendations based on the simulation results are made.

2 Past Studies

Various groups have reported their work on use of approximation models in evolutionary algorithms. A complete survey on the use of fitness approximation in evolutionary algorithms is reported by Jin and Sendhoff [1]. In this paper, authors have broadly classified the approximation methods, which are used currently, in three categories, namely response surface methodology, kriging models and artificial neural networks. The other issues of concern are the efficient use of original fitness function, termed as model management, and the quality of model, which should improve with iterations. The latter refers to training methods, error measures and active data sampling which are employed in generating a model for fitness surface.

Jin et al. [10], while working with approximate models, have also demonstrated the controlled evolution in evolution strategy. Here, the word control refers to the original fitness function evaluations. So if the entire generation is evaluated using the original fitness function, it is called as controlled generation. Similarly, if only few population members are evaluated using original fitness function, it is referred as controlled individuals. A framework which guarantee the correct convergence while reducing the computational cost is established.

The idea of taking less function evaluations in order to reach the optima using controlled individuals with the help of clustering and neural network is explored by Jin and Sendhoff [11]. The individuals near the center of the cluster is evaluated using expensive fitness function evaluation to create neural network ensemble which is used for fitness values of remaining individuals. The structure and parameters of the neural network ensemble are also optimized using a standard evolution strategy.

Branke and Schmidt [12] have used two estimation methods, namely, regression and interpolation, to achieve faster convergence to the optima. In there technique, at every generation, a fixed percentage of the population is evaluated with exact objective function. The fitness of the rest of the population is estimated. The individuals which are evaluated accurately are determined based on their estimated fitness and uncertainty. Savings in accurate function evaluations up to fifty percent are reported.

Farina has also used radial basis neural network for objective function approximation. The algorithm described elsewhere [13] has been successfully tested on test problem ZDT3 which has a typical discrete Pareto-optimal front.

The attempt to reduce number of function evaluations using fitness inheritance [14, 15] is also reported. Sastry et al. [14] have used inheritance combined with population sizing models and have reported a saving of 20% in function evaluations. In case of fixed population size, authors have reported a saving of 70% by employing a simple inheritance technique. Chen et al. [15] have extended the fitness inheritance concept for multi-objective optimization. The authors have reported a 40% saving in terms of function evaluations for the case of fitness inheritance without fitness sharing, while in the case of fitness inheritance with fitness sharing, a saving of 25% is claimed.

Rasheed and Hirsh [16] have used informed operators for speeding up the genetic algorithms. They have modified the genetic operators such as mutation and crossover and have made them more informed using reduced models. In genetic algorithm procedure, where a random choice is made, they generate a number of candidates and rank them using inexpensive reduced models, and then take the best of the result. Naturally, it gives a speed-up in the genetic algorithm procedure. They have successfully tested their method on a complex engineering design problem. In other work by Rasheed et al. [17], a comparison between two methods for using reduced models to speed up the search in genetic algorithm based engineering design optimization is presented. They have reported that the informed operators approach is better than the genetic engineering approach. It is also found that least square approximation with any of the above two speed-up approach produces better results than neural network approximations.

Applications using surrogate models which can be substituted for exact and costly evaluation tools is also discussed by Giannakoglu [18]. In this work, radial basis function neural networks are used for generating surrogate model. The prediction capability of the surrogate model is dependent on the shape of the real response surface, availability of sufficient training data, number of hidden units and the activation function used. Author has also discussed [18, 19] a genetic algorithm based on inexact pre-evaluations (GA-IPE). This method starts with all exact evaluations during first generation. Later, when surrogate model is constructed, the entire population is evaluated approximately using the model. Only the best individuals identified as per inexact evaluation are exactly evaluated which reduces the computational cost. Hence GA-IPE takes more generations but has a low evaluation cost. It is successfully used to solve the turbomachinery problem of airfoil design.

El-Beltagy et al. [20] have suggested the use of metamodels to reduce the computational burden on the evolutionary algorithms. They have used a metamodelling technique, namely the kriging approach. This paper points toward the use of local metamodels instead of global metamodels, as building a global metamodel will be a computationally expensive process. In other work Nair et al. [21] have combined approximation concept with genetic algorithms for structural optimization applications. Investigators have tried to reduce the number of exact function evaluations while ensuring to converge to the optima of the original problem. They have employed a dynamic optimization technique, wherein the fitness function changes over successive generations. They have controlled the generational delay before the approximation model is updated alongwith an adaptive selection operator. This technique is applied to solve a 10-bar truss design problem. They have used a strategy by which the fitness function is changed during the run but granularity of the optimization model is not changed.

A reconstruction algorithm is proposed by Ratle [22] which uses a kriging metamodel for fitness landscape approximation. The algorithm shows a overall reduction in the number of fitness calls. The algorithm is tested successfully for a two-dimensional problem and for a difficult twenty dimensional multi-modal problem.

Emmerich et al. [23] have proposed a metamodel assisted evolution strategy for reducing the computational cost for computationally expensive optimization problems. Local kriging metamodel is built which uses fixed number of nearest neighbors. The method is tested on Kean's function and airfoil shape optimization problem. The results are quite encouraging for single objective optimization. In other work by Emmerich and Naujokes [24], the kriging metamodels are used in multi-objective optimization. The local metamodels are used to decide the potential of a new population member, i.e. to decide whether it should be evaluated precisely or rejected. Three different rejection principles are tested over three different optimizers.

3 Solution Strategy

As suggested by authors elsewhere [3], here we have combined genetic algorithm with an approximation technique. However, a brief discussion on the basic approach is as follows. Genetic algorithms take a sufficiently large number of function evaluations to reach the optima of the problem. So to reduce the number of precise function evaluations, an approximate model of the problem is used.

The basic idea can simply be illustrated with the help of Fig. 1. The figure shows a hypothetical one-dimensional function for minimization. As concluded elsewhere [5], if GA has to start with random initial population such a function will require large population size. Initially, the optimizer is run with precise

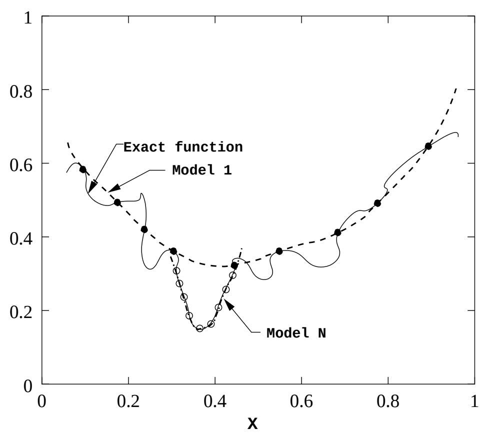

Fig. 1. Successive approximate modeling

function evaluations. During the initial generations, the population of a GA is usually scattered all over the search space. If the number of such data points is small then approximate model built will only capture the general trend of the function and it will miss the finer details. Hence, if an approximation model with such a database is built, then it can only capture the general trend of the fitness landscape. So the basic job of this initial approximate model is to give initial search direction to the optimizer. However, as generations of GA progress, the population tend to converge towards the better regions of the approximated fitness landscape. Now, if at this stage, the approximation model is rebuilt using new data with precise function evaluations, then this new approximate model will be able to capture the finer details of the fitness landscape as it will have more data points concentrated over smaller search region. The new approximate model will be a the refined model of the fitness landscape. This refined model will again help GA identify the better regions, and this process can be repeated till the termination criteria of the optimizer is met. Hence this optimization technique basically employs successive approximate models for reaching the optima.

Figure 2 shows a schematic line diagram of the successive approximate model based optimization procedure. The combined procedure begins with a set of randomly created N solutions, where N is the population size. Since an adequate size of solutions are required to arrive at an approximated problem, we execute a GA with exact function evaluations for n generations, thereby collecting a total of $N' = nN$ solutions for approximation. At the end of n generations, the approximation technique is invoked with N' solutions and the first approximated problem is created. The GA is then performed for the next $(Q - n)$ generations with this approximated problem. Thereafter, the GA is performed with the exact function for the next n generations and a new approximated problem is created. This procedure is continued till the termination criterion is met. Thus, this

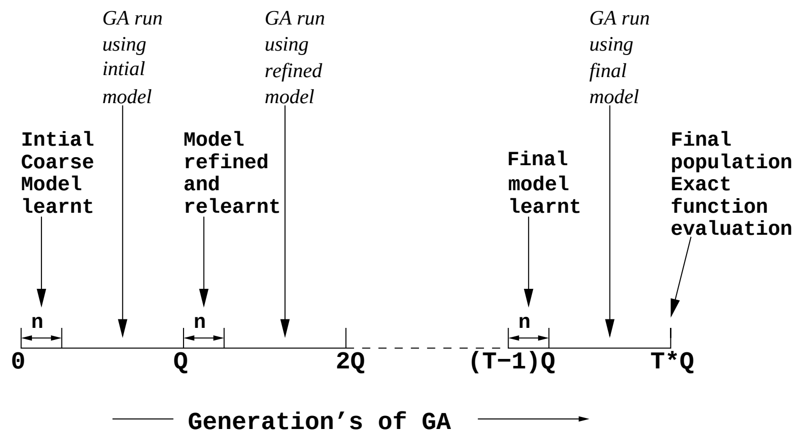

Fig. 2. A line diagram of the solution procedure

procedure requires a fraction n/Q of total evaluations in evaluating the problem exactly. An efficient approach will be to decrease N' solutions towards the later generations of a GA run, as during this phase approximate model will not change significantly.

4 Principle Results

In this section, some results of the multi-objective optimization based on successive approximations are presented. The multi-objective optimizer is non-dominating sorting genetic algorithm (NSGA-II) [4]. Artificial neural networks based on standard error back-propagation algorithm with sigmoidal activation function [25] is used for generating approximate model of the problem. For the ANN, input neurons represent problem variables and output neurons represent different objective functions and constraint functions. The combined procedure will be referred as NSGA-II-ANN simulation. The ANN is trained using two different models, namely, batch training and incremental training. If a simulation is performed with batch training in which after every Q generation the approximation model is refined and the training database is collected over n generations, It is called as the B-Q-n model. If the training method is incremental, then it is called as the I-Q-n model.

4.1 Case Study: Some Test Problems

A number of mathematical test problems are available in literature. Here, some of the test problems, namely TNK, ZDT4, DTLZ2 [2, 6] are selected for testing the procedure. These problems are selected as they test different aspects of the optimizer, which will be highlighted in the subsequent sections.

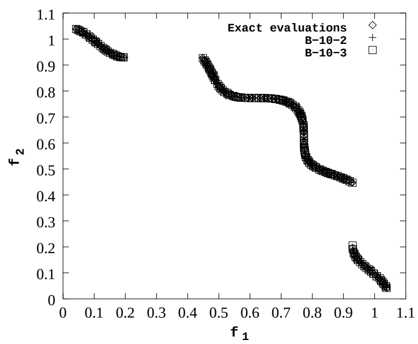

Fig. 3. Batch model simulation results for the TNK test problem

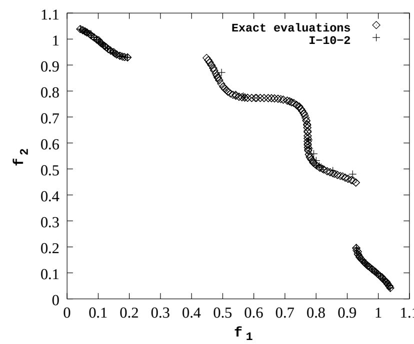

Fig. 4. Incremental model simulation results for the TNK test problem

TNK: It is a two variable constrained test problem. The problem variables are real valued. The Pareto-optimal front is disconnected and have three parts as is visible in Figs. 3 and 4. The number of precise function evaluations taken by simulations of various models in order to reach the Pareto front are given in Table 1. Fig. 3 shows that B-10-2 and B-10-3 NSGA-II-ANN simulations reach the Pareto front with a saving of about 50% in precise function evaluations. Figs. 4 shows that I-10-2 NSGA-II-ANN simulations reach to Pareto-optimal front without any saving in precise function evaluations.

Table 1. Spread metric results of NSGA-II-ANN simulations on the TNK test problem

Model Name	Part-I	Part-II	Part-III	Precise Function Evaluations
NSGA-II	0.644	0.872	0.704	5,00,400
B-10-2	0.597	0.810	0.613	2,50,183
B-10-3	0.588	0.833	0.724	2,50,256
I-10-2	0.540	1.047	0.507	4,99,933

The spread of the above three simulations and that of NSGA-II alone is compared in the Table 1. The spread metric calculation method is as suggested in Deb et al. [2]. In this metric calculation, the Euclidean distance of two extreme ends of simulation from corresponding ends of Pareto-optimal front as well as the uniformity of distribution for intermediate solutions is considered. A smaller value of the spread metric means a better spread. Batch model simulations show better spread results as well as savings in precise function evaluation. If we concentrate on the spread metric value, B-10-2 simulation is the best and is closely followed by the B-10-3 simulation. However, incremental mode simulations have difficulty in capturing the middle portion of the Pareto-optimal front which is also evident from the high spread metric value obtained for the I-10-2 simulation.

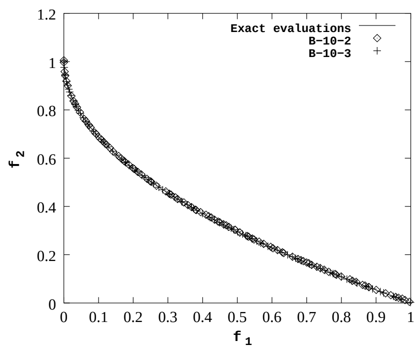

Fig. 5. Batch model simulation results for the ZDT4 test problem

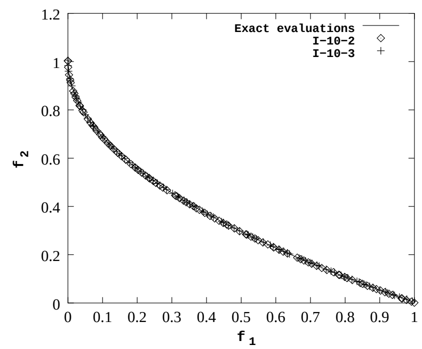

Fig. 6. Incremental model simulation results for the ZDT4 test problem

ZDT4: It is a ten-variable problem. The problem variables are real valued. It has a convex global Pareto-optimal front. This is a difficult unconstrained test problem as it has 100 distinct Pareto-optimal fronts, out of which only one is global. Here, all NSGA-II-ANN simulations have reached the global Pareto-optimal front. But due to presence of local sub-optimal Pareto fronts, the savings in precise function evaluations are slightly lower. Figs. 5 and 6 show the NSGA-II-ANN simulations in batch and incremental mode, respectively. Both figures show that simulations are capturing the global Pareto-optimal front. The number of precise function evaluations taken by simulations of various models in order to reach the Pareto front are given in Table 2. The saving in both batch models namely, B-10-2 and B-10-3 models are approximately 25%.

Table 2. Spread metric results of NSGA-II-ANN simulations on the ZDT4 test problem

Model Name	Spread metric value	Precise Function Evaluations
NSGA-II	0.386	30,200
B-10-2	0.422	22,675
B-10-3	0.332	22,730
I-10-2	0.344	22,675
I-10-3	0.388	21,781

The spread metric value are given in Table 2. Here, the best spread metric value is achieved by NSGA-II-ANN simulation in batch mode of training, namely, for B-10-3 model. The incremental models, namely, I-10-2 and I-10-3 also show better spread metric results. Only one simulation, i.e. B-10-2 gives a poor spread when compared to NSGA-II simulation running only with precise function evaluations.

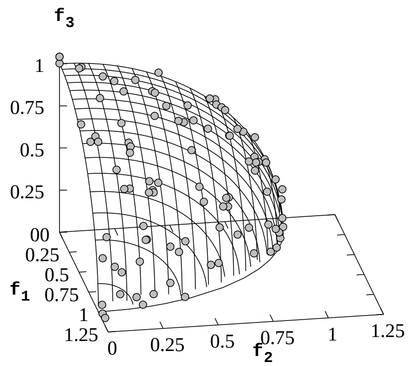

Fig. 7. Batch model simulation results for the DTLZ2 test problem

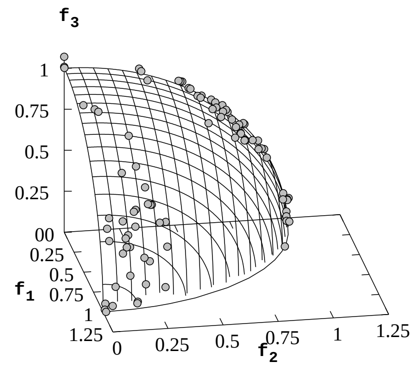

Fig. 8. Incremental model simulation results for the DTLZ2 test problem

DTLZ2: This test problem is selected from the work of Deb et al. [6]. It is a three objective test problem. The Pareto-optimal surface is the first octant surface of unit sphere. It has 12 real variables. It is a problem where most of the multi-objective optimizers find difficulty in reaching the Pareto surface. The uniform distribution of the solutions on the Pareto surface is also difficult to achieve. This problem has a non-convex Pareto-optimal surface. Figs. 7 and 8 are the results of two simulations of NSGA-II-ANN. The number of precise function evaluations taken by simulations of various models in order to reach the Pareto front are given in Table 3. The batch model B-10-3 shows a saving of 62% in precise function evaluations. The incremental model, I-10-3 gives a saving of 44% in precise function evaluations.

Table 3. Sparsity measure results of NSGA-II-ANN simulations on the DTLZ2 test problem

Model Name	Sparsity measure value	Precise Function Evaluations
NSGA-II	0.951	30,200
B-10-3	0.964	11,496
I-10-3	0.908	17,120

The distribution of the solutions on the Pareto-optimal surface is evaluated by using a sparsity measure. The method to evaluate the sparsity measure is given elsewhere [7]. It is basically a method to quantify the distribution of solutions similar to the entropy measure or the grid diversity measure. A higher value of the sparsity measure means a better diversity among the solutions. Table 3 shows the sparsity measure for NSGA-II-ANN simulation. The B-10-3 batch model shows the best diversity among Pareto-optimal solutions followed by NSGA-II run with only precise function evaluations. I-10-3 incremental model shows the poorest distribution of solutions.

4.2 Case Study: Cantilever Plate Design

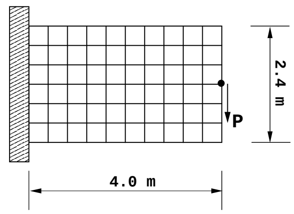

Fig. 9. A cantilever plate

This is a constrained multi-objective shape optimization problem for a mechanical component design [8]. The two conflicting objectives are minimization of the weight of the plate and minimization of the maximum deflection. The maximum stress should be less than the yield strength of the material. The mesh grid size of 12×20 is considered here. This is a 240 binary variable problem. A binary variable represents presence or absence of the material at any particular grid element. A load of 100 kN is applied at the free end of the cantilever. The plate material is taken to be steel. In addition to the material constraint stated above, there is an additional geometry constraint in this problem. Any feasible solution shall have material at the fixed end of the cantilever and a continuous material connection to the point of application of the load. However, this constraint is automatically handled by the coding of the problem. In this problem, a finite element procedure is employed for precise function evaluations.

The previous experiments with NSGA-II-ANN simulations on various test problems and on the curve fitting problem in computer aided design [3] favor batch model of training with ANN. Hence in this problem, only the batch model is tried. Figs. 10 to 13 show simulation results of different batch models. In all simulations, 5,400 precise function evaluations are used. The NSGA-II simulation with only precise function evaluations is shown by a thin line. However to add to clarity, the one best simulation each from Figs. 10 to 13 is selected. Fig. 14 shows such a plot. The B-20-3 is the best NSGA-II-ANN simulation. In order to check the savings, NSGA-II is permitted to run for more than 5,400 precise function evaluations (100 generations) fixed earlier. It is found that B-20-3 NSGA-II-ANN simulation fully dominates the NSGA-II simulation till 10,800 precise function evaluations i.e. up to 200 generations. It is shown in Fig. 15. Hence B-20-3 model saves 50% function evaluations.

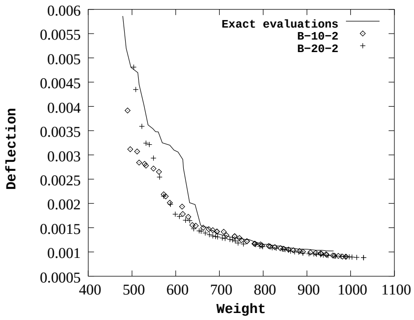

Fig. 10. Batch model results with two-generation ($2N$) database

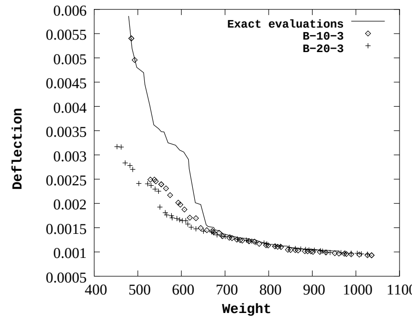

Fig. 11. Batch model results with three-generation ($3N$) database

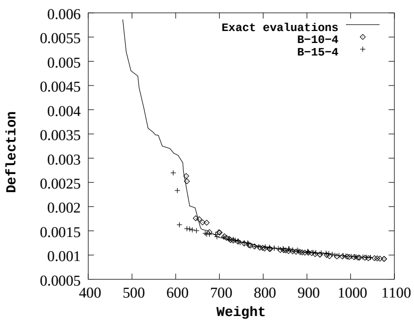

Fig. 12. Batch model results with four-generation ($4N$) database

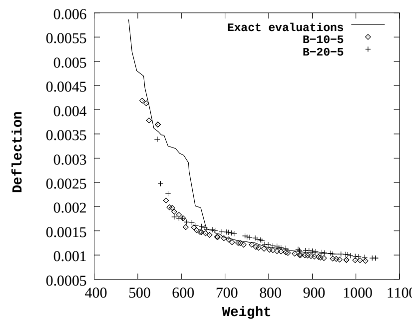

Fig. 13. Batch model results with five-generation ($5N$) database

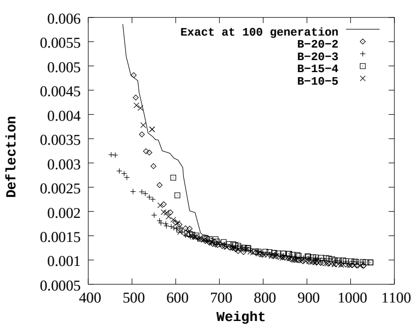

Fig. 14. Best of different batch model simulations

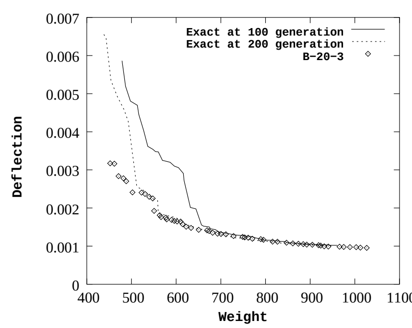

Fig. 15. Comparison of best batch simulation with exact NSGA-II runs

Convergence and Spread Metric Values: To quantify the level of convergence achieved in each simulation, we compute a convergence metric suggested elsewhere [9]. The metric computes the average distance of each obtained solution from a reference set of points. Naturally, a smaller metric value is judged to be better. As the current problem is a real world problem, the true Pareto-optimal front is not known. As suggested in [9], we choose a reference set P^* containing 94 data points obtained from a combined pool of 15 simulations (shown in Fig. 16) with a total of 810 points. For calculating the convergence metric value, first the non-dominated set F of the final generation of each simulation is identified. Then for each point in F, smallest normalized Euclidean distance to P^* is calculated. Next the convergence metric value is calculated by averaging the normalized distance of all points in the F. Lastly, in order to keep the convergence metric within $[0, 1]$, we divide the convergence metric value by the maximum value found among all simulations. Fig. 16 shows the normalized convergence metric value calculated for various simulations by the NSGA-II-ANN procedure and the NSGA-II working with precise function evaluations (labeled as exact). This figure shows that convergence metric value for all the NSGA-II-ANN simulations is either near or better than NSGA-II run with precise function evaluations for 200 generations. The convergence metric value of the best simulation, namely B-20-3, is somewhere in between 250 generation and 300 generation. Hence, a saving higher than 50% is achieved.

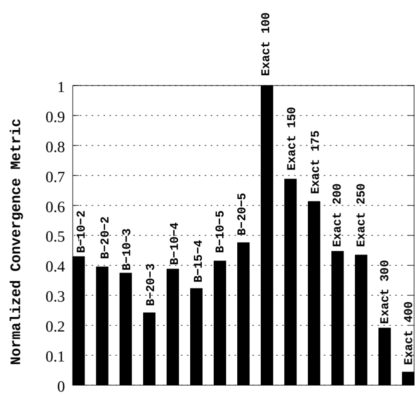

Fig. 16. Convergence metric results of cantilever plate design problem

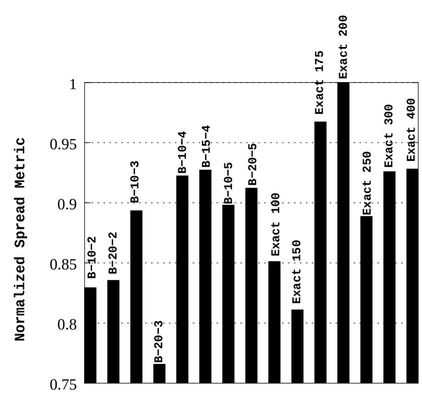

Fig. 17. Spread metric results of cantilever plate design problem

Now, a similar procedure is adopted for calculating the spread metric value. The two extremes of the a reference set P^* containing 94 data points obtained from a combined pool of 15 simulations (shown in Fig. 17) with a total of 810 points are identified. As stated earlier the rest of the calculation procedure is same as followed for the various test problems. Hence a smaller spread metric value is better. Fig. 17 shows that B-20-3 model has best spread, i.e. it is having a uniform distribution as well as it is also close to two extremes of the reference set.

5 Other Approximation Technique

Authors are currently trying to integrate other approximation techniques in their procedure. The issue of primary concern in choosing an approximation technique with evolutionary algorithms is that it should be sufficiently tested for generating approximate models. The literature [20, 22–24] supports the use of kriging metamodels as it is used extensively in various kind of applications. However, while integrating any approximation technique with evolutionary algorithms, a frequent assumption, that the time taken to develop the approximate model is negligible in comparison of the precise function evaluation is made. Hence, researchers focus on limiting the number of precise function evaluations while developing any such procedure. The fact we wish to highlight here is related to the assumption stated above. While generating an approximate model in kriging, we need to minimize a unconstrained problem for estimating an important parameter which is generally denoted with θ in the literature. This process of solving the unconstrained minimization problem requires repetitive inversion of a matrix, the size (n) of which depends on number of data used for developing the model. The CPU cost for matrix inversion is quite significant as it has $O(n^3)$ complexity. This problem is further aggravated if a global model of the fitness landscape is constructed. It may be said, the use of local metamodel can reduce the problem as number of data points used for developing the metamodel will reduce. But, then it will call for additional trouble of generating a large number of metamodels for the same problem. So this tradeoff is critical. Another disadvantage of the kriging metamodel is that a different approximate model is required for each of the objectives and constraints. Hence, in case of constrained multi-objective optimization, it will lead to high computational load. Authors are trying to implement it on a CAD curve fitting problem described elsewhere [3]. The primary results show that while a complete simulation of NSGA-II-ANN procedure took around 3 minutes to complete on a single processor machine, NSGA-II-KRIGING procedure took around three days on a 16 processor, P-III 1 GHz cluster. Hence, NSGA-ANN procedure is a better overall approach.

6 Conclusions

The real world application of optimization finds many computationally expensive problems. This computational load is still higher for multi-objective optimization. Hence there is a need for a generic multi-objective optimization procedure which can work reliably with approximate models. However, in such cases, the reliability of both the optimizer and approximate modelling technique are significant. So in this paper, a multi-objective optimizer, namely NSGA-II is used in conjunction of ANN. The NSGA-II-ANN procedure performance is tested on some of the test problems. Later it is used to solve an engineering shape optimization problem involving finite element method for solution evaluation. The case study involve different aspects of optimization problems. The aspects tested here are:

- real variable problem and binary variable problem,
- unconstrained problem and constrained problem,
- discontinuous Pareto front problem and continuous Pareto front problem,
- non-convex Pareto front problem and convex Pareto front problem,
- Pareto front problem and Pareto surface problem.

The simulation results are analyzed for two aspects, namely, convergence to Pareto-optimal front and the spread of obtained front. It is concluded that NSGA-II-ANN simulations generally show a saving in precise function evaluations of about 25% to 62%. Simulations presented here show a good convergence and spread. An important finding of this exclusive study is that the batch model B-10-3 shows consistently better performance in all problems and it is recommended. Lastly, the issue of integrating some other approximate modelling in place of ANN is discussed. Some problems related to commonly used kriging metamodel generating schemes is briefly highlighted. However it remains an open question which needs further detailed explorations.

References

1. Jin, Y., and Sendhoff, B.: Fitness approximation in evolutionary computation - A survey. In *Proceedings, Genetic and Evolutionary Computation Conference, 2002.* Morgan Kaufmann, 2002, pp. 1105-1112.
2. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms.* First Edition, Chichester, Uk: Wiley, 2001.
3. Nain, P.K.S., and Deb, K.: Computationally effective search and optimization procedure using coarse to fine approximation. In *Proceedings, Congress on Evolutionary Computation, 2003.* IEEE Computer Society Press, 2003, pp. 2081-2088.
4. Deb, K. , Agarwal, S. , Pratap, A. , and Meyarivan, T.: A fast elitist non-dominating sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings, 6th International Conference on Parallel Problem Solving from Nature - PPSN VI.* Lecture Notes in Computer Science 1917 Springer, 2000, pp. 849-868.
5. Goldberg, D. E., Deb, K., and Clark, J. H.: Genetic algorithms, noise, and the sizing of populations. *Complex System*, 6, pp. 333-362, 1992.
6. Deb, K., Thiele, L., Laumanns, M., and Zitzler, E.: Scalable multi-objective test problems. In *Proceedings, Congress on Evolutionary Computation, 2002.* pp. 825-830, 2002.
7. Deb, K., Mohan, M., and Mishra, S.: Towards a quick computation of well-spread pareto-optimal solutions. In *Proceedings, Evolutionary Multi-Criterion Optimization, 2003.* Springer, 2003, pp. 226-236.
8. Deb, K., Chaudhauri, S., Jain, P., Gupta, N., Maji, H., and Reddy, A.R.: Discovering innovative design principles using multi-conflicting objectives. In *Proceedings, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial problems, EUROGEN 2003*, pp. 118-119, 2003.

9. Deb, K., and Jain, S.: Running performance Metrics for evolutionary multi-objective optimization. In *Proceedings, Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02).* (Singapore),2002,pp. 13-20.
10. Jin, Y., Olhofer, M., and Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. In *IEEE Transactions on Evolutionary Computation,* 6(5), pp. 481-494, 2002.
11. Jin, Y., and Sendhoff, B.: Reducing fitness evaluations using clustering techniques and neural network ensembles. In *Proceedings, Genetic and Evolutionary Computation Conference, 2004.* Springer, 2004, pp. 688-699.
12. Branke, J., and Schmidt, C.: Faster convergence by means of fitness estimation. In *Soft Computing Journal.* (in press).
13. Farina, M.: A neural network based generalized response surface multi-objective evolutionary algorithms. In *Congress on Evolutionary Computation, 2002.* IEEE Press, 2002, pp. 956-961.
14. Sastry, K., Goldberg, D. E., and Pelikan, M.: Don't evaluate, inherit. In *Proceedings, Genetic and Evolutionary computation Conference, 2001.* Morgan Kaufmann, 2001, pp. 551-558.
15. Chen, J., Goldberg, D. E., Ho, S., and Sastry, K.: Fitness inheritance in multi-objective optimization. In *Proceedings, Genetic and Evolutionary computation Conference, 2002.* Morgan Kaufmann, 2002, pp. 319-326.
16. Rasheed, K., and Hirsh, H.: Informed operators: speeding up genetic-algorithm-based design optimization using reduced models. In *Proceedings, Genetic and Evolutionary Computation Conference, 2000.* Morgan Kaufmann, 2000, pp. 628-635.
17. Rasheed, K., Vattam, S., and Ni, X.: Comparison of methods for using reduced models to speed up design optimization. In *Proceedings, Genetic and Evolutionary Computation Conference, 2002.* Morgan Kaufmann, 2002, pp. 1180-1187.
18. Giannakoglou, K.C.: Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. In *Progress in Aerospace Science,* Vol. 38, pp. 43-76, 2002.
19. Giotis, A. P., and Giannakoglou, K.C.: Chapter 23: low cost GAs assisted by ANNs - applications in turbomachinery. edited by *Periaux, J. et al.*, Jhon Wiley & Sons, (to appear).
20. El-Beltagy, M. A., Nair, P. B., and Keane, A. J.: Metamodelling techniques for evolutionary optimization of computationally expensive problems: promises and limitations. In *Proceedings of the Genetic and Evolutionary Computation Conference, 1999.* Morgan Kaufman, 1999, pp. 196-203.
21. Nair, P. B., Keane, A. J., and Shimpi, R. P.: Combining approximation concepts with genetic algorithm based structural optimization procedures. In *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 1998, pp. 1741-1751
22. Ratle, A.: Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In *Proceedings, Parallel Problem Solving from Nature, 1998.* volume V, 1998, pp. 87–96.
23. Emmerich, M., Giotis, A., Ozdenir, M., Back, T., and Giannakoglou, K.: Metamodel-assisted evolution strategies. In *Proceedings, Parallel Problem Solving from Nature, 2002.* Springer, 2002, pp. 371-380.
24. Emmerich, M., and Naujoks, B.: Metamodel assisted multi-objective optimization strategies and their application in airfoil design. In *Proceedings, Adaptive Computing in Design and Manufacture VI, 2004.* Springer, 2004, pp. 249–260.
25. Haykin, S.: *Neural networks a comprehensive foundation.* second edition, Singapore: Addison Wesley, 2001. pp. 208.