An Evolutionary Algorithm with Spatially Distributed Surrogates for Multiobjective Optimization

Amitay Isaacs, Tapabrata Ray, and Warren Smith

University of New South Wales, Australian Defence Force Academy, ACT, Australia
{a.isaacs,t.ray,w.smith}@adfa.edu.au

Abstract. In this paper, an evolutionary algorithm with spatially distributed surrogates (EASDS) for multiobjective optimization is presented. The algorithm performs actual analysis for the initial population and periodically every few generations. An external archive of the unique solutions evaluated using the actual analysis is maintained to train the surrogate models. The data points in the archive are split into multiple partitions using k-Means clustering. A Radial Basis Function (RBF) network surrogate model is built for each partition using a fraction of the points in that partition. The rest of the points in the partition are used as a validation data to decide the prediction accuracy of the surrogate model. Prediction of a new candidate solution is done by the surrogate model with the least prediction error in the neighborhood of that point. Five multiobjective test problems are presented in this study and a comparison with Nondominated Sorting Genetic Algorithm II (NSGA-II) is included to highlight the benefits offered by our approach. EASDS algorithm consistently reported better nondominated solutions for all the test cases for the same number of actual evaluations as compared to a single global surrogate model and NSGA-II.

1 Introduction

Evolutionary algorithms (EAs) are particularly attractive for multiobjective problems as they result in a set of nondominated solutions in a single run. Furthermore, EAs do not rely on functional and slope continuity and thus can be readily applied to optimization problems with mixed variables. However, EAs are essentially population based methods and require evaluation of numerous solutions before converging to the desired set of solutions. Such an approach turns out to be computationally prohibitive for realistic Multidisciplinary Design Optimization (MDO) problems and there is a growing interest in the use of surrogates to reduce the number of actual function evaluations.

A comprehensive review on the use of fitness approximation in the context of evolutionary computation has been reported by Jin [1]. The choice of surrogate models reported in literature range from neural network based models like multilayer perceptrons, radial basis function networks, quadratic response surfaces, Kriging and cokriging models. A vast majority of surrogate assisted

M. Randall, H.A. Abbass, and J. Wiles (Eds.): ACAL 2007, LNAI 4828, pp. 257–268, 2007.

optimization methods rely on the use of a single global surrogate model. The surrogate model is either created once and used subsequently throughout the course of search (one shot approach) or created periodically. Algorithms based on the one shot training of the approximation model(s) [2,3] are likely to face the problems when the initial set of solutions generated differ substantially from the final set as in the case of the test function SCH1 [4]. Periodic retraining is necessary as the search proceeds to localized areas. In order to capture local behavior, hierarchical surrogate models have been proposed by Zhou *et al* [5] and the use of artificial neural network models in the local search strategy have been used by Gasper-Cunha and Vieira [6].

To improve the prediction accuracy with limited samples, multiple surrogates can be used in place of the single surrogate model. Common use of multiple surrogates is in the form of surrogate ensembles where a collection of surrogate models with varying parameters usually trained simultaneously by techniques such as bagging [7], and boosting [8] are used. A survey of neural network ensemble has been reported by Zhao *et al* [9]. Jin and Sendhoff [10] reported the use of clustering and neural network ensembles to reduce the fitness evaluations. They use k-Means clustering to identify the candidate solutions which need to be evaluated using the actual analysis. Hamza and Saitou [11] have used polynomial regression surrogate ensembles with weighted average response and the most conservative response in the co-evolutionary genetic algorithm for vehicle crash-worthiness design.

Another approach based on multiple surrogates is to use different types of surrogate models simultaneously. Goel *et al* [3] and Zerpa *et al* [12] have used a weighted average model resulting from three surrogate types (polynomial response surface model, kriging, and radial basis function). The two approaches differ in the determination of the weights for averaging. Zhou *et al* [13] reported the use of multiple approximation models in the context of memetic algorithm to perform the local search. They even propose using a surrogate ensemble as one of the approximation models.

In the context of multiobjective optimization, Nain and Deb [14] proposed a multifidelity model (coarse to fine grain) for surrogate assisted multiobjective optimization where a multilayer perceptron was periodically retrained and used in alternation with actual computations to solve a B-spline curve fitting problem. A similar approach of alternating between actual analysis (K) and surrogate models (S) have been reported by Ray and Smith [15]. The study used a RBF model that was trained using the candidate solutions of the population after every K generations. Nain and Deb [16] reported the performance of successive surrogate models on two test functions viz. ZDT4 and TNK. Pareto Efficient Global Optimization (ParEGO) algorithm [17] relies on a kriging based surrogate and the sampling points are generated via design of experiments. However, the method requires knowledge about the limits of the objective function space and cannot guarantee a uniform distribution of the solutions along the nondominated front. Emmerich *et al* [18] use confidence interval predicted by the kriging model to screen candidates for actual evaluation, reducing the computational cost. A recent paper by Chafekar *et al* [19] reports the use of multiple GAs, each of which uses a reduced model of the objective function with a regular information exchange among GAs to obtain a well distributed nondominated set of solutions.

In this paper an evolutionary algorithm with spatially distributed surrogates (EASDS) is presented. This approach uses multiple surrogates that are spatially distributed in the design space. An archive of the solutions evaluated using the actual analysis is maintained and used to train the surrogate models. The solutions in the archive are split in multiple partitions using k-Means clustering. Using a fraction of the solutions in each partition a Radial Basis Function network surrogate model is trained. The unused points in each of the partition are used to assess the prediction accuracy of the surrogate model. The performance of the EASDS is compared with Nondominated Sorting Genetic Algorithm (NSGA-II) [20] using an equal number of actual function evaluations. The effect of the number of partitions is also studied and the performance is compared with a single global surrogate model.

2 An Evolutionary Algorithm with Spatially Distributed Surrogates

The pseudo code of the proposed Evolutionary Algorithm with Spatially Distributed Surrogates (EASDS) is outlined in Algorithm 1.

Algorithm 1. Evolutionary Algorithm with Spatially Distributed Surrogates

Require: $N_G > 1$ {Number of Generations}
Require: $M > 0$ {Population size}
Require: $K > 1$ {Number of partitions}
Require: $I_{TRAIN} > 0$ {Periodic Surrogate Training Interval}
$\mathcal{A} = \emptyset$ {Archive of the Solutions}
P_1 = Initialize()
Evaluate(P_1)
$\mathcal{A}$ = AddToArchive($\mathcal{A}, P_1$)
$\mathcal{S}$ = BuildSurrogates($\mathcal{A}, K$)
for $i = 2$ to N_G **do**
  if modulo(i, I_{TRAIN}) == 0 **then**
    Evaluate(P_{i-1}) {Evaluate parent population using the Actual Analysis}
    $\mathcal{A}$ = AddToArchive($\mathcal{A}, P_{i-1}$)
    $\mathcal{S}$ = BuildSurrogates($\mathcal{A}, K$)
  end if
  C_{i-1} = Evolve($P_{i-1}, \mathcal{S}$)
  EvaluateSurrogate($C_{i-1}, \mathcal{S}$)
  P_i = Reduce($P_{i-1} + C_{i-1}$)
end for

The basic evolutionary algorithm is on the same lines as that of NSGA-II by Deb *et al* [20]. The algorithm starts with a random initial population and evaluates the population using actual evaluations. Spatially Distributed Surrogate models (using Radial Basis Function network) are created for all the objectives and the constraints. An external archive of actual evaluations is maintained in EASDS and used to periodically train the surrogate models for all the objectives and the constraints. The components of EASDS are described below.

2.1 Initialization

All the solutions in the population are initialized randomly by selecting each variable value from the specified range for that variable.

2.2 Archive of the Actual Evaluations

All the unique candidate solutions that are evaluated using the actual analysis are maintained in an external archive. Every I_{TRAIN} generations, the parent population is evaluated using the actual analysis functions and then added to the archive. New solution is added to the archive only if the normalized distance (using the Euclidean norm) between the new solution and each of the solutions in the archive is more than user defined distance criterion. This condition avoids the numerical difficulties of building the surrogates if the solutions are too close.

2.3 Evolutionary Strategy

The evolutionary strategy of EASDS is the same as that of NSGA-II. Binary tournament is used for the selection the parents undergoing crossover. The simulated binary crossover (SBX) operator [21] and a polynomial mutation operator [22] are used to create an offspring population from the parent population.

2.4 Building Spatially Distributed Surrogate Models

Outlined in Algorithm 2 are the steps involved in building the RBF surrogate models for the objectives and the constraints. A collection of RBF surrogate models is created to approximate the objectives and the constraints. The archive is split into K partitions $(\mathcal{A}_1, \ldots, \mathcal{A}_K)$ using k-Means clustering algorithm where the design variables $x_1, \ldots, x_n$ are used as the clustering attributes.

The solutions in each of the partitions are used to build the RBF surrogate models for the objectives and the constraints. Only a fraction $(0 < \alpha < 1)$ of the solutions are used to train the surrogate model and the rest are used as the validation data set. EASDS uses 80% of the solutions in each partition as the training data and the remaining 20% are used to validate the surrogate models.

If there are very few solutions in a partition (insufficient to build the RBF surrogate model), no surrogate models are built using that partition. If the prediction error on the validation data set in the partition is more than the user defined threshold, the surrogate model on that partition is deemed invalid.

Algorithm 2. Building Spatially Distributed Surrogate Models

Require: $\mathcal{A}$ {Archive of actual evaluations}
Require: K {Number of partitions}
Require: $m \geq 2$ {Number of objectives}
Require: $p \geq 0$ {Number of constraints}
$\mathcal{A}_1, \ldots, \mathcal{A}_K = \text{KMeans}(\mathcal{A}, K)$
for $i = 1$ to K **do**
  for $j = 1$ to m **do**
    $\mathcal{S}_{i,f_j} = \text{RBF Train}(\mathcal{A}_i, f_j)$
  end for
  for $j = 1$ to p **do**
    $\mathcal{S}_{i,g_j} = \text{RBF Train}(\mathcal{A}_i, g_j)$
  end for
end for

k-Means Clustering Algorithm. A k-Means clustering algorithm [23] is used to split given data points into k clusters or partitions. The main idea of k-Means clustering is to define k centroids, one for each cluster, and then assign each data point to one of the k clusters so as to minimize a measure of dispersion within the clusters. A very common measure is the sum of squared Euclidean distances from the centroid of each cluster.

Radial Basis Function Network Surrogate. Radial Basis function networks belong to the class of Artificial Neural Networks (ANNs) and are a popular choice for approximating nonlinear functions. A radial basis function (RBF) ϕ has its output symmetric around an associated centre μ.

$$\phi(\mathbf{x}) = \phi(\|\mathbf{x} - \mu\|)$$

where the argument of ϕ is a vector norm. A common RBF is the Gaussian function with the Euclidean norm.

$$\phi(r) = e^{-r^2/\sigma^2}$$

where σ is the scale or width parameter. A set of RBFs can serve as a basis for representing a wide class of functions that are expressible as linear combinations of the chosen RBFs as shown in Eq. 1.

$$y(\mathbf{x}) = \sum_{i=1}^{k} w_i \, \phi(\|\mathbf{x} - \mu_i\|) \tag{1}$$

Here, k is typically smaller than the number of data points. The coefficients w_i are the unknown parameters that are to be “learned.” The training is usually achieved via the least squares solution:

$$\mathbf{w} = \mathbf{A}^{+} \, \mathbf{y} \tag{2}$$

where $\mathbf{A}^{+}$ is the pseudo-inverse and $\mathbf{y}$ is the target output vector.

2.5 Evaluation Using Spatially Distributed Surrogate Models

For accurate prediction of the objectives and the constraints for a new candidate solution, a surrogate model with the least prediction error is chosen from spatially distributed surrogate models. If the new candidate solution is far (using the Euclidean distance measure) from all the solutions in the archive, it is evaluated using the actual analysis.

From the archive of solutions, S solutions closest (using the Euclidean norm) to a new candidate are selected. The values of the objectives and the constraints of these S points are predicted using each of the surrogate models in the collection. For each of the surrogate models, prediction error (RMSE) is computed using the actual and the predicted values of the objectives and the constraints. Surrogate model with the least prediction error is then used to predict the value at the new candidate solution.

2.6 Reduction

The reduction procedure retains the best individuals from the parent and the offspring population (elitism). Combined solutions from the parent population and the offspring population are ranked using the non-dominated sorting and the crowding distance criterion [24]. M elite solutions (better fitness) are retained for the next generation from a set of $2M$ solutions (parent and offspring population). If there are less than M feasible solutions, then infeasible solutions with smaller values of maximum constraint violation are retained.

3 Numerical Examples

3.1 Test Problems

The first two constrained test problems are SRN and OSY [25]. The ZDT test problems [25] are two objective unconstrained problems framed by Zitzler *et al* and they are of the form as shown in Eq. 3.

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}), \\ & f_2(\mathbf{x}) = g(\mathbf{x})\, h(f_1(\mathbf{x}), g(\mathbf{x})). \end{aligned} \tag{3}$$

The description of all the test problems is given in Table 1.

3.2 Experimental Setup

A population size of 100 is used for all the test problems and the algorithm is run for 101 generations. All the test problems are evaluated using EASDS and NSGA-II. For EASDS, the surrogate models were retrained every 5 generations ($I_{TRAIN} = 5$). The probability of crossover is set to 0.9 and the probability of mutation is set to 0.1. The Distribution index for crossover is 10 and the distribution index for mutation is 20. A new solution is added to the archive if the normalized distance between the new solution and closest solution in the

Table 1. Test Problems

Problem	Dim	Objectives & Constraints	Bounds
SRN	2	$f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2$ $f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2$ $x_1^2 + x_2^2 \leq 225,\ x_1 - 3x_2 + 10 \leq 0$	$\mathbf{x} \in [-20, 20]^2$
OSY	6	$f_1(\mathbf{x}) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2$ $+(x_4 - 4)^2 + (x_5 - 1)^2]$ $f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$ $x_1 + x_2 - 2 \geq 0,\ 6 - x_1 - x_2 \geq 0,$ $2 - x_2 + x_1 \geq 0,\ 4 - (x_3 - 3)^2 - x_4 \geq 0,$ $2 - x_1 + 3x_2 \geq 0,\ (x_5 - 3)^2 + x_6 - 4 \geq 0$	$x_1, x_2, x_6 \in [0, 10]$ $x_3, x_5 \in [1, 5]$ $x_4 \in [0, 6]$
ZDT1	10	$f_1(\mathbf{x}) = x_1$ $g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g}$	$\mathbf{x} \in [0, 1]^{10}$
ZDT2	10	$f_1(\mathbf{x}) = x_1$ $g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - (f_1/g)^2$	$\mathbf{x} \in [0, 1]^{10}$
ZDT3	10	$f_1(\mathbf{x}) = x_1$ $g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1)$	$\mathbf{x} \in [0, 1]^{10}$

archive is more than 0.01. If the prediction error (RMSE) of a surrogate over validation data is less than 20%, then it is considered valid.

To compare the effects of the number of the surrogate models (corresponding to the number of partitions of the archive), each of the problems was run with 3, 5 and 8 partitions. All the test problems are also run with single surrogate model.

The same random seed and hence the same initial population is used for both, EASDS and NSGA-II. Since the number of actual function evaluations in EASDS are much less than 10100 (100×101), NSGA-II is run for fewer generations (with similar number of function evaluations) for performance comparison.

4 Results

Shown in Table 2 are the function evaluations used by EASDS for different number of partitions (K). Traditional evolutionary algorithms with population size of 100 evolved over 101 generations will result in $10,100$ function evaluations. In EASDS, the population is evaluated using the actual evaluations every $I_{TRAIN} = 5$ generations, hence the minimum number of actual evaluations is 2100.

As seen from Table 2, the number of function evaluations for problem OSY decreases as the number of partitions is increased. This shows that the prediction

Table 2. Function Evaluations used by EASDS

	Function Evaluations		
Problem	$K = 3$	$K = 5$	$K = 8$
OSY	7139	4111	3620
SRN	2100	2100	2100
ZDT1	2100	2600	2600
ZDT2	2100	2600	2600
ZDT3	2100	2600	2600

accuracy of the surrogate models with $K = 3$ partitions is poor as compared to the surrogate models with $K = 8$ partitions. If all the surrogates models are invalid (prediction accuracy over the validation data set is more than the user defined threshold), actual evaluations are used.

The non-dominated solutions for problem OSY obtained by EASDS using 3, 5, and 8 partitions are shows in Fig. 1. It is observed that the non-dominated solutions of the EASDS with 8 partitions follow the Pareto front much more accurately than the EASDS with 3 and 5 partitions.

For test problem SRN, the number of function evaluations used are 2100, the minimum possible. This indicates that EASDS with 3, 5, or 8 partitions is able to correctly capture the behavior of the function SRN. As seen in Fig. 2, the non-dominated solutions of EASDS with 3, 5, and 8 partitions are overlapping.

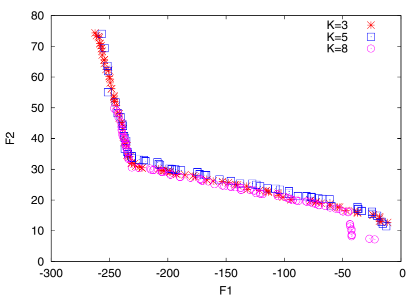

Fig. 1. Effect of number of partitions on problem OSY

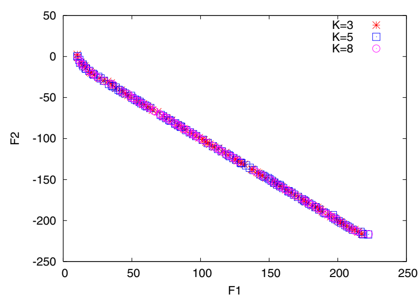

Fig. 2. Effect of number of partitions on problem SRN

Test problems ZDT1, ZDT2, and ZDT3 show a different trend in function evaluations as compared to OSY. The number of function evaluations used by EASDS increase for 5 and 8 partitions as compared to 3 partitions. This can be explained by the fact that the surrogate models in the initial few generations are not very accurate and the actual evaluations are used to evaluate the entire population. In the earlier generations, there are fewer solutions in the archive and those solutions are distributed spatially and split in to multiple partitions to build the surrogate models. Thus each partition might have insufficient number of points to capture the correct behavior of the function and the prediction accuracy is low. As the number of solutions accumulate in the archive, the accuracies of the surrogate models also increase.

Shown in Fig. 3 are the non-dominated solutions for problem ZDT1 obtained by EASDS using 3, 5, and 8 partitions and they are overlapping. It shows that the function ZDT1 is approximated accurately using the surrogate models with 3, 5, and 8 partitions. For the test function ZDT2, the non-dominated solutions are shown in Fig. 4. The surrogate models with 8 partitions are able to achieve better spread of the non-dominated solutions on the Pareto front indicating that surrogate models with 8 partitions have better prediction accuracy than surrogate models with 3 and 5 partitions.

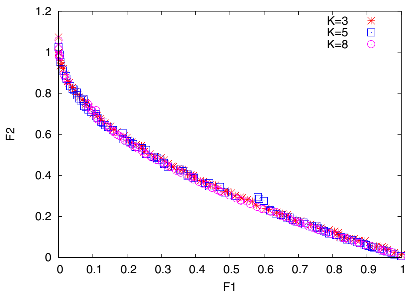

Fig. 3. Effect of number of partitions on problem ZDT1

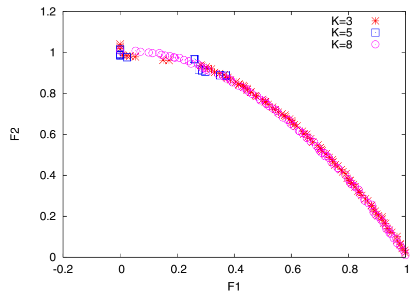

Fig. 4. Effect of number of partitions on problem ZDT2

Shown in Fig. 5 are the non-dominated solutions for problem ZDT3 obtained by EASDS using 3, 5, and 8 partitions. It is seen that none of the surrogate models are able to completely capture the disjoint Pareto front. Surrogate models with 8 partitions seem to have a better spread than the ones with 5 partitions which are better than the ones with 3 partitions.

Shown in Fig. 6 are the non-dominated solutions for problem OSY obtained by EASDS using 8 partitions (EASDS), single global surrogate model (SGS) and NSGA-II with the same number of function evaluations. The performance of EASDS is better at capturing the Pareto front.

The non-dominated solutions obtained for the problem SRN by EASDS, SGS, and NSGA-II are shown in Fig. 7. Even a single global surrogate is able to capture the behavior of the function adequately and the non-dominated solutions overlap.

The benefit of the spatially distributed surrogate models can be seen from the results of ZDT1, ZDT2, and ZDT3 which are 10-D functions. It can be seen from Figs. 8, 9, and 10 that EASDS captures the Pareto front better than NSGA-II and single global surrogate model (SGS).

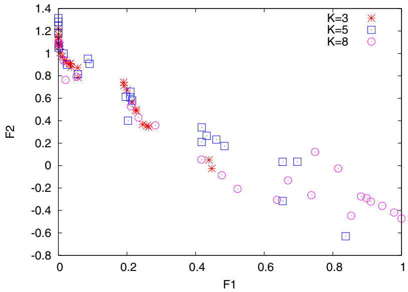

Fig. 5. Effect of number of partitions on problem ZDT3

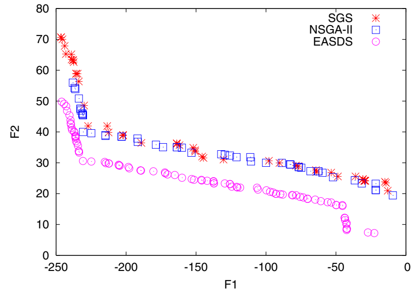

Fig. 6. Non-dominated solutions for problem OSY

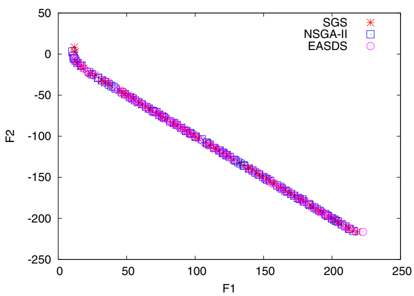

Fig. 7. Non-dominated solutions for problem SRN

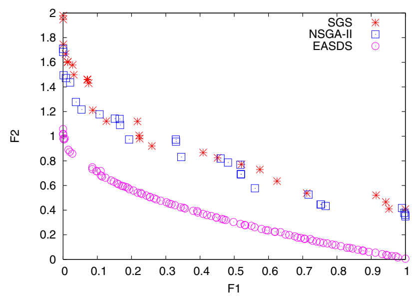

Fig. 8. Non-dominated solutions for problem ZDT1

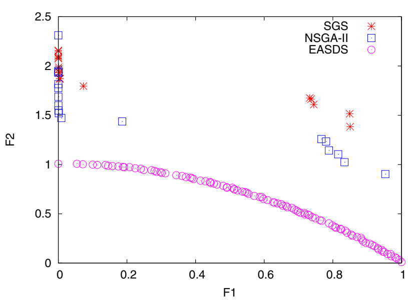

Fig. 9. Non-dominated solutions for problem ZDT2

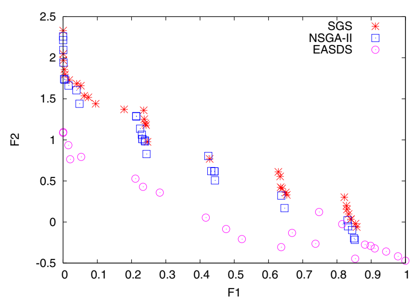

Fig. 10. Non-dominated solutions for problem ZDT3

5 Summary and Conclusions

In this paper an evolutionary algorithm with spatially distributed surrogates (EASDS) for multiobjective optimization is presented. This approach is an alternative to recent surrogate ensemble proposals, where either multiple types of global surrogates are used or multiple number of same type of surrogates are used for better approximation. EASDS is compared with the non-dominated sorting algorithm (NSGA-II) and single global surrogate model on a set of test functions. Different number of partitions (3, 5, and 8) are used to build the surrogate models and corresponding performance is compared.

It is seen from Figs. 1 - 5 that the surrogate models with more partitions perform better. With more partitions the function behavior is captured better by splitting the design space in multiple regions and approximating each region locally. But as the number of partitions is increased, more number of evaluations are required to populate each partition sufficiently (to be able to correctly capture the behavior of the function locally in the partition). With the computational budget of 1200 evaluations EASDS is able to capture the behavior of 10-D optimization problem with up to 8 partitions. For a smaller computational budget or higher dimensional problem, one may need to use more conservative number of partitions.

Compared to the single global surrogate model and NSGA-II, EASDS performs much better indicating the benefits of the local surrogates built over smaller regions. Effectiveness of EASDS at capturing the Pareto front and the spread of solutions along the Pareto front is clearly seen from Figs. 6, 8, and 9.

References

1. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing - A Fusion of Foundations, Methodologies and Applications 9, 3–12 (2005)
2. Wilson, B., Cappelleri, D., Simpson, T.W., Frecker, M.: Efficient pareto frontier exploration using surrogate approximations. Optimization and Engineering 2, 31–50 (2001)
3. Goel, T., Haftka, R.T., Shyy, W., Queipo, N.V.: Ensemble of surrogates. Structural and Multidisciplinary Optimization 33, 199–216 (2007)
4. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Proceedings of the 1st International Conference on Genetic Algorithms, pp. 93–100. Lawrence Erlbaum Associates, Mahwah, NJ (1985)
5. Zhou, Z.Z, Ong, Y.S, Nair, P.B, Keane, A.J, Lum, K.Y: Combining global and local surrogate models to accelerate evolutionary optimization. IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews 37, 66–76 (2007)
6. Gaspar-Cunha, A., Vieira, A.S.: A hybrid multi-objective evolutionary algorithm using an inverse neural network. In: Hybrid Metaheuristics (HM 2004) Workshop at ECAI 2004, Valencia, Spain (2004)
7. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
8. Abney, S., Schapire, R.E., Singer, Y.: Boosting applied to tagging and PP attachment. In: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999)

9. Zhao, Y., Gao, J., Yang, X.: A survey of neural network ensembles. In: International Conference on Neural Networks and Brain (ICNN&B 2005), vol. 1, pp. 438–442 (2005)
10. Jin, Y., Sendhoff, B.: Reducing fitness evaluations using clustering techniques and neural networks ensembles. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 688–699. Springer, Heidelberg (2004)
11. Hamza, K., Saitou, K.: Vehicle crashworthiness design via a surrogate model ensemble and a coevolutionary genetic algorithm. In: Proceedings of IDETC/CIE 2005 ASME 2005 International Design Engineering Technical Conference, California, USA (September 2005)
12. Zerpa, L.E., Queipo, N.V., Pintos, S., Salager, J.L.: An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates. Journal of Petroleum Science and Engineering 47, 197–208 (2005)
13. Zhou, Z., Ong, Y.S., Lim, M.H., Lee, B.S.: Memetic algorithm using multi-surrogates for computationally expensive optimization problems. Soft Computing - A Fusion of Foundations, Methodologies and Applications 11, 957–971 (2007)
14. Nain, P., Deb, K.: A computationally effective multi-objective search and optimization techniques using coarse-to-fine grain modeling. In: 2002 PPSN Workshop on Evolutionary Multiobjective Optimization (2002)
15. Ray, T., Smith, W.: Surrogate assisted evolutionary algorithm for multiobjective optimization. In: Proceedings of 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, pp. 1–8 (2006)
16. Nain, P.K.S., Deb, K.: A multi-objective optimization procedure with successive approximate models. Technical Report 2005002, KanGAL, IIT Kanpur (2005)
17. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 10, 50–66 (2006)
18. Emmerich, M.T.M., Giannakoglou, K.C., Naujoks, B.: Single and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Transactions on Evolutionary Computation 10, 421–439 (2006)
19. Chafekar, D., Shi, L., Rasheed, K., Xuan, J.: Multiobjective ga optimization using reduced models. Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 35, 261–265 (2005)
20. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
21. Deb, K., Agrawal, S.: Simulated binary crossover for continuous search space. Complex Systems 9, 115–148 (1995)
22. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. Computer Science and Informatics 26, 30–45 (1996)
23. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys 31, 265–323 (1999)
24. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the Parallel Problem Solving from Nature VI, pp. 849–858 (2000)
25. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley and Sons, Chichester (2001)