11th World Congress on Structural and Multidisciplinary Optimisation
07th -12th, June 2015, Sydney Australia

Multi-objective Optimization Using Adaptive Explicit Non-Dominated Region Sampling

Anirban Basudhar

Livermore Software Technology Corporation, Livermore, CA, USA, abasudhar@lstc.com

1. Abstract
A new method to efficiently perform multi-objective optimization (MOO), referred to as Adaptive Explicit Multi-objective Optimization (AEMOO), is presented. Unlike existing methods, it uses binary classification to explicitly define the decision boundary between dominated and non-dominated (ND) regions in the design space. An adaptively refined support vector machine (SVM) is used to define the boundary. AEMOO has several advantages that stem from the availability of the estimated explicit boundary bounding the ND design space, which represents Pareto-optimal (PO) designs at convergence. It allows for an effective adaptive sampling strategy that samples "important" regions in the design space. Additionally, explicit knowledge of the PO design space facilitates efficient real time Pareto-optimality decisions. AEMOO uses a hybrid approach that considers the distribution of samples in both design and objective spaces. Two variants of AEMOO are presented - one based purely on classification and the other based on both classification and metamodel approximation. The results are compared to the widely used NSGAII method and Pareto Domain Reduction (PDR) using test problems up to 30 variables. AEMOO shows significantly better efficiency and robustness compared to these existing methods.
2. Keywords: support vector machine, hybrid adaptive sampling, real time optimality decision, binary response.

3. Introduction
Multi-objective optimization (MOO) often involves locating a set of Pareto optimal points that are ND over the entire design space (x-space). The multiple objectives typically represent quantities with different units that are not comparable. Additionally, finding a solution set that can represent the complete Pareto front is more challenging than finding a single solution. Therefore, efficient and accurate solution of MOO problems is overall much more challenging compared to single-objective optimization (SOO), and is still an evolving research area.
Classical MOO approaches involve scalarization of the objectives to convert MOO into several SOO problems [1], but these approaches are not well suited to find a set of points that can represent the complete PO front. Evolutionary algorithms, e.g. SPEA and NSGAII, are extensively used to solve MOO problems in a true multi-objective sense [2,3]. These methods are applied in a direct optimization framework or in conjunction with metamodel approximations to alleviate the cost of potentially expensive evaluations (e.g. crashworthiness) [4]. The metamodel-based method is further classified based on sampling schemes. In some of these, sampling is based on direct optimization (e.g. genetic operators) [5]. The metamodel's accuracy estimate is used to determine which samples need to be evaluated using the expensive evaluator, and rest of the samples are evaluated using a metamodel. In other methods, sampling is directed to obtain accurate metamodels and a method like NSGAII is then applied on the metamodels to find PO points. A basic approach is to use sequential space-filling samples for metamodeling [4]. While this approach is global, filling the space is prohibitively expensive in higher dimensions. Another metamodel-based MOO algorithm is ParEGO [6], which is an extension of its SOO counterpart. ParEGO involves scalarization of objectives. Additionally, it was tested only up to eight variables in [6].
One major limitation of commonly used MOO methods is the lack of explicit definition of PO regions of design space. To determine whether a given sample is ND, one needs to evaluate the objective functions and constraints at not only that point but over the entire design space. ND points are determined using pairwise comparison in objective (f) space before mapping them back to the design space. For large sample sizes, these calculations can add significant overhead. Additionally, in the absence of an explicit definition of ND regions of the design space, it is difficult to adaptively add samples in those regions of particular interest. A sequential metamodel-based method known as Pareto Domain Reduction (PDR) was recently developed that focuses on sampling based on design space sparsity in the vicinity of predicted front [7]. Hyperboxes constructed around previous ND points were sampled. However, performance of PDR depended on the number and size of such hyperboxes.
This paper presents a novel classification-based adaptive explicit MOO approach (AEMOO) that mitigates aforementioned limitations of existing methods by providing an efficient sampling scheme. This is made possible by constructing an explicit definition of the boundary separating dominated and ND samples in the design space using a support vector machine (SVM) [8] and using a hybrid sampling scheme that accounts for sparsity in both spaces (x and f space). Treatment of MOO as a binary classification problem involves a paradigm shift in its solution approach. The goal in MOO is to determine whether a design is PO or not, and to find a set of such designs. Therefore, it is naturally suited for treatment as binary classification. Using the proposed approach, a

single evaluation of the trained classifier can determine if a design is ND. This makes it especially attractive for real time Pareto-optimality decisions. Explicit definition of ND regions also facilitates implementation of a proposed adaptive sampling scheme that selects samples in these regions. Restricting the sampling region using the decision boundary improves the PO front search efficiency. AEMOO can be implemented either as a classifier assisted direct method or in conjunction with metamodel approximations. It should be noted that in the latter approach, constraints can be handled using metamodel approximations or by approximating the zero level using a classifier. Classifier-based handling allows the handling of discontinuous and binary constraint functions [9].

4. Adaptive Explicit Multi-objective Optimization (AEMOO)

Two variants of the proposed AEMOO method are presented in this section. In Section 4.1 the basic idea of classification-based MOO is presented. In Section 4.2 a classifier-assisted direct optimization method is presented. A second method that utilizes classification as well metamodel approximation is presented in Section 4.3.

4.1. Classification Approach for MOO - Dynamic Classifier

Typically, solution of MOO is a set of PO points. Part of the design space is PO while other regions are dominated. Such a problem is ideal for applying classification methods, the two classes being dominated (-1 class) and ND (+1 class) (Fig. 1). Once a classifier separating dominated and ND samples is trained, Pareto-optimality of any design is determined using a single classifier evaluation, in contrast with existing methods.
SVM is used to construct the decision boundary in this work (Eq.(1)). It constructs the optimal boundary that maximizes the margin between two sample classes (± 1) in a feature space. A kernel K maps the design and feature spaces. In this work, a Gaussian kernel is used to construct SVM boundaries ($s(\mathbf{x}) = 0$). Spread of the kernel is assigned the largest value without any training misclassification. The SVM is defined as:

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) \tag{1}$$

Here, $y_i = \pm 1$ is class label, α_i is the Lagrange multiplier for i^{th} sample and b is bias. Each sample's class needs to be assigned prior to the construction of SVM boundary. An issue in using classification for MOO is that although it is known that dominated samples (class -1) cannot be PO, the opposite is not true; being ND among current samples isn't sufficient to be PO. However, +1 samples represent PO designs if the data is sufficient. A decision boundary obtained by assigning +1 class to the current ND samples represents an estimate of the PO front, and is refined adaptively. As points are added, samples may switch from +1 to -1 as ND samples may be dominated by newer samples until convergence. As class definition of existing samples can change during the course of AEMOO, the classifier is referred to as *dynamic*. The classification based AEMOO method has several advantages.

- Explicit definition of the ND region facilitates implementation of an efficient sampling scheme
- It facilitates efficient real time Pareto optimality decisions
- It uses information in both design and objective spaces to enhance its efficiency and robustness
- The classification approach allows the handling of binary and discontinuous constraint functions
- As ND region is explicitly defined in the design space, AEMOO is unaffected by number of objectives.

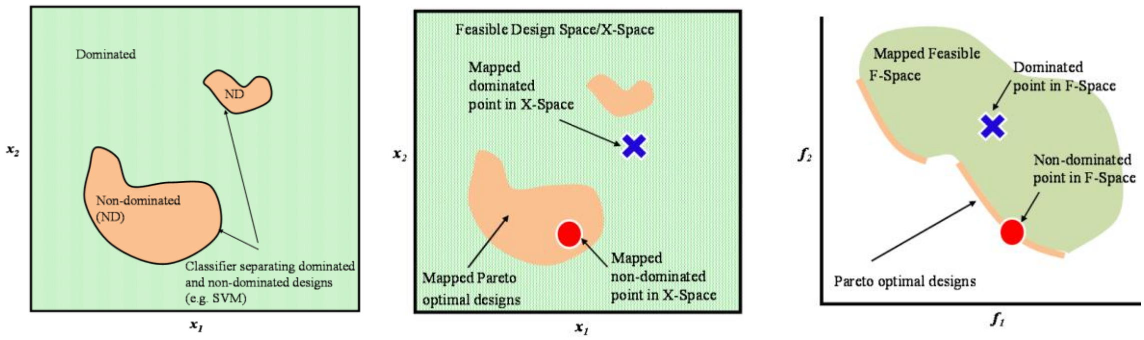

Figure 1: Boundary defining ND regions in design space (left). Design-Objective space Mapping (right).

4.2. Direct AEMOO Method

Direct AEMOO is based on SVM classification only and does not approximate function values. It evaluates more samples in the explicitly defined SVM-based ND regions to avoid waste of samples and increase efficiency. The sign of SVM value s($\mathbf{x}$) determines whether a sample is ND or not, and is straightforward to determine.
A fraction of samples per iteration are selected within the s($\mathbf{x}$) > 0 ND region of design space. Details are shown in

Fig. 2. Sampling important regions of the design space allows a faster increase in the SVM accuracy, as sampling based only on the objective space can lead to clustering of samples in design space, where the SVM is constructed. Maximizing the value of SVM, one of the sampling criteria, is equivalent to maximizing the probability of locating a ND sample [8,10]. Additionally, one generation of NSGAII is also used to select samples, first within the s(**x**) > 0 regions and then using an unconstrained formulation. Using NSGAII-based samples, the algorithm ensures that the effects of sparsity in the objective function space and genetic operator-based evolution are also accounted for. In order to ensure a global search, a small fraction of samples is added based on maximum minimum distance in the entire unconstrained design space. Such samples are not expected to provide an efficient sampling scheme, and are therefore optional, but guarantee the location of the complete Pareto front when run for a sufficient time.

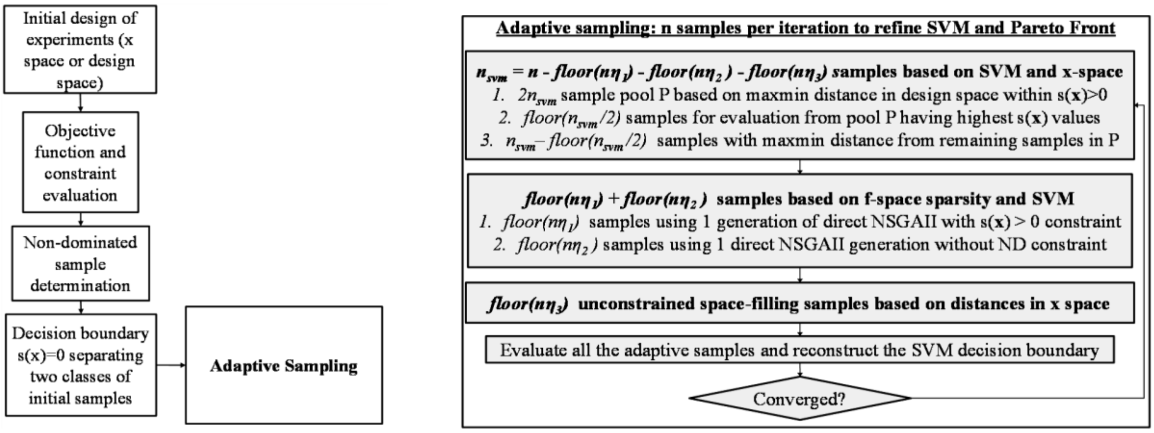

Figure 2: Summary of direct AEMOO method (left) and sampling scheme (right)

4.3. Metamodel-assisted AEMOO Method

In this approach, metamodel approximation and SVM are used together. Basic idea is same as direct AEMOO - to consider ND ranking along with sample sparsity in both x and f spaces. However, the single generation of direct NSGA-II samples is replaced with converged predicted PO samples obtained using metamodel-based NSGAII. Metamodel approximation and the SVM-based classification serve as complementary approaches that help in enhancing accuracy by accounting for distribution of samples in both spaces. The methodology is shown in Fig. 3.

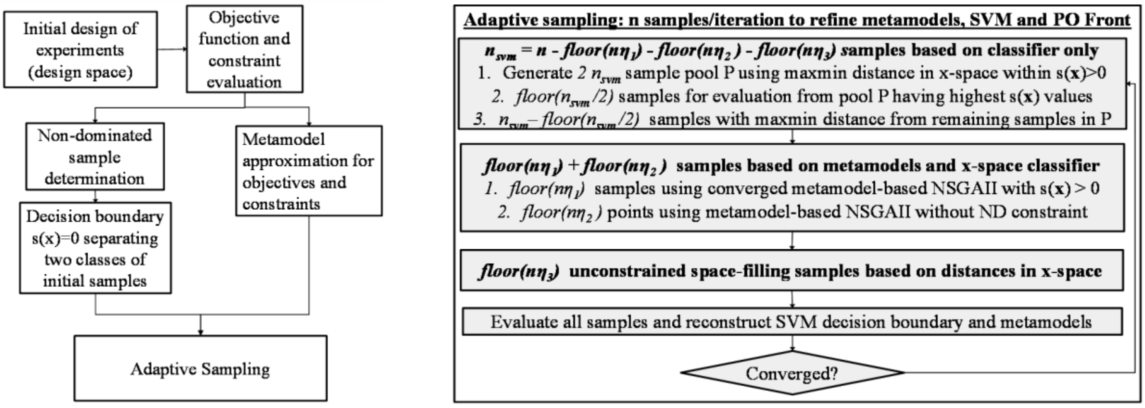

Figure 3: Summary of metamodel-assisted AEMOO method (left) and sampling scheme (right)

5. Results

Several examples are presented to validate the efficacy and efficiency of AEMOO. A 10 variable analytical example using direct AEMOO is presented, followed by three 30 variable examples using metamodel-assisted AEMOO. Efficiency is compared to existing methods PDR and NSGAII. Finally, AEMOO is used for tolerance based MOO of a Chevrolet truck [11]. The sample type fractions are $\eta_1 = \eta_2 = 0.2$ and $\eta_3 = 0.1$ (Fig. 2 and 3). Unless mentioned, *floor(1.5*(m+1))+1* samples are used per iteration, *m* being the number of variables. AEMOO showed comparatively higher efficiency also when larger sample sizes were used, but those results haven't been shown. Radial basis function metamodels have been used for function approximations, but others can also be used. For examples 1-4, one of the objectives is $f_1 = x_1$. The second objective f_2 is provided with the individual examples.

5.1. Example 1. Analytical example with ten variables and two objectives - ZDT3 (Direct AEMOO)

This example has 10 variables (m = 10) and 2 objectives. The second objective f_2 is:

$$f_2 = r(\mathbf{x})h(f_1(\mathbf{x})r(\mathbf{x})), \text{ where } \; r(\mathbf{x}) = 1 + \frac{9}{m-1}\sum_{i=2}^{m} x_i \; , \; h(\mathbf{x}) = 1 - \sqrt{\frac{f_1(\mathbf{x})}{r(\mathbf{x})}} - \left(\frac{f_1(\mathbf{x})}{r(\mathbf{x})}\right)\sin(10\pi f_1(\mathbf{x})) \quad (2)$$

The Pareto fronts at successive iterations are plotted in Fig. 4. The front at iteration 125 (2250 points) is quite close to the actual one, and shows that AEMOO can locate disjoint PO fronts even when only classification is used. NSGAII found four of the disjoint regions on the front accurately, but completely missed one region. This can be attributed to sampling based on the f-space only without considering design space sparsity, unlike in AEMOO.

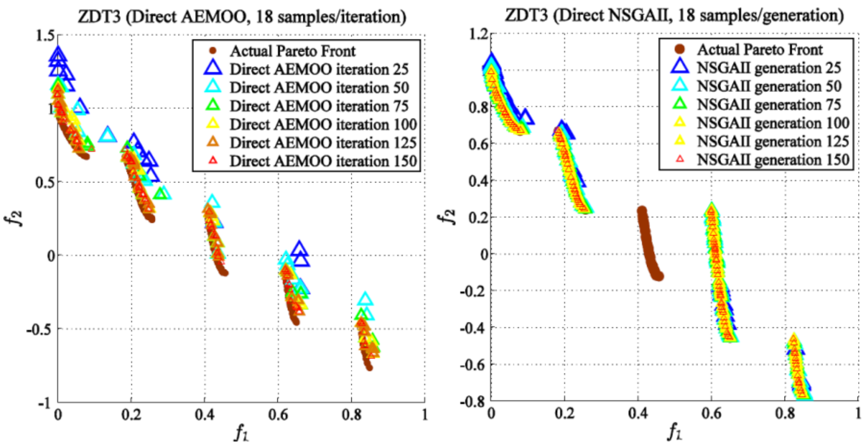

Figure 4: Results for Example 1. Direct AEMOO (left) and direct NSGAII (right).

5.2. Example 2. Analytical example with 30 variables and 2 objectives - ZDT1 (Metamodel-assisted AEMOO)
This problem consists of two objectives f_1 and f_2. The second objective is:

$$f_2 = r(\mathbf{x})h(f_1(\mathbf{x})r(\mathbf{x})), \text{ where } \; r(\mathbf{x}) = 1 + \frac{9}{m-1}\sum_{i=2}^{m} x_i \; , \; h(\mathbf{x}) = 1 - \sqrt{\frac{f_1(\mathbf{x})}{r(\mathbf{x})}} \quad (3)$$

Optimization results are shown in Fig. 5 using trade off plots at different iterations at intervals of 10. The results shown are the evaluated Pareto optimal points. The proposed AEMOO method is able to locate the entire spread of Pareto front at the 10th iteration itself (470 samples), after which it adds diversity. The samples on the front are very well diversified by the 20th iteration. In comparison, performance of direct NSGAII is much slower and it takes 40-50 generations (1920-2400 samples) to obtain a diversified front. Even at 50th generation, the Pareto front using NSGAII is not as accurate as the 10th iteration of AEMOO. PDR performs more efficiently than direct NSGAII, but still takes 20-30 iterations (940-1410 samples) to obtain a well diversified accurate front.

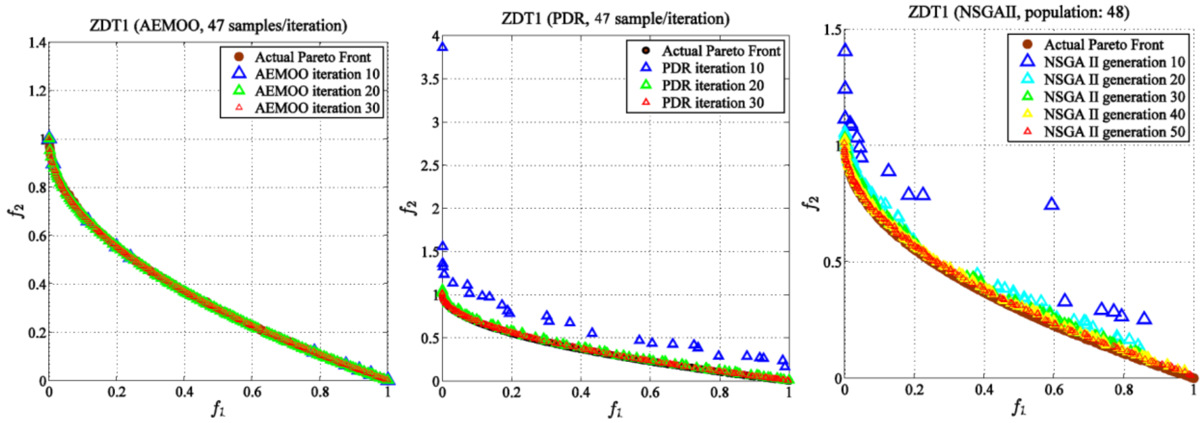

Figure 5: Example 2. Metamodel-assisted AEMOO (left), PDR (center) and direct NSGAII (right).

5.3. Example 3. Analytical example with 30 variables and 2 objectives - ZDT2 (Metamodel-assisted AEMOO)
This problem consists of 30 variables (m = 30) and two objective functions. The second objective is:

$$f_2 = r(\mathbf{x})h(f_1(\mathbf{x})r(\mathbf{x})), \text{ where } \; r(\mathbf{x}) = 1 + \frac{9}{m-1}\sum_{i=2}^{m} x_i \; , \; h(\mathbf{x}) = 1 - \left(\frac{f_1(\mathbf{x})}{r(\mathbf{x})}\right)^2 \quad (4)$$

Optimization results are shown using computed trade off plots in Fig. 6. AEMOO is able to find a very well diversified and accurate front before the 10th iteration itself (470 samples). On the contrary, with a comparable population size of 48, direct NSGAII failed to obtain the Pareto front even after 50 generations. The population size for NSGAII had to be increased to find the actual front. PDR was able to locate the Pareto front with a sample size of 47 per iteration, but was slower than AEMOO, as it took 30 iterations (1410 samples) to obtain a front of comparable (but not quite as good) accuracy and diversity.

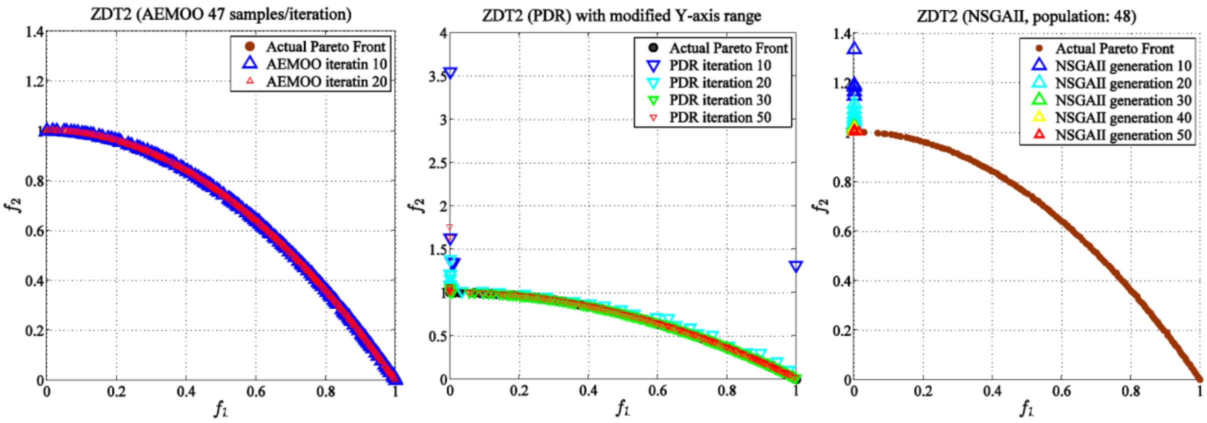

Figure 6: Example 3. Metamodel-assisted AEMOO (left), PDR (center) and direct NSGAII (right).

5.4. Example 4. Analytical example with 30 variables and 2 objectives - ZDT3 (Metamodel-assisted AEMOO)
This optimization problem is similar to Example 1 (Eq.(2)), except that it has 30 variables instead of 10. PO fronts using the three methods are shown in Fig. 7, along with the actual front. AEMOO located all five disjoint regions on the front within first 10 iterations, following which it further added samples on the front to increase diversity. Both NSGAII and PDR were significantly slower and had lower accuracy. Using population size of 48, direct NSGAII completely missed 2 out of 5 regions. PDR was able to sample 4 of the regions satisfactorily at 40-50 iterations (1880-2350 samples). It found one ND sample close to the fifth region, but not on it.

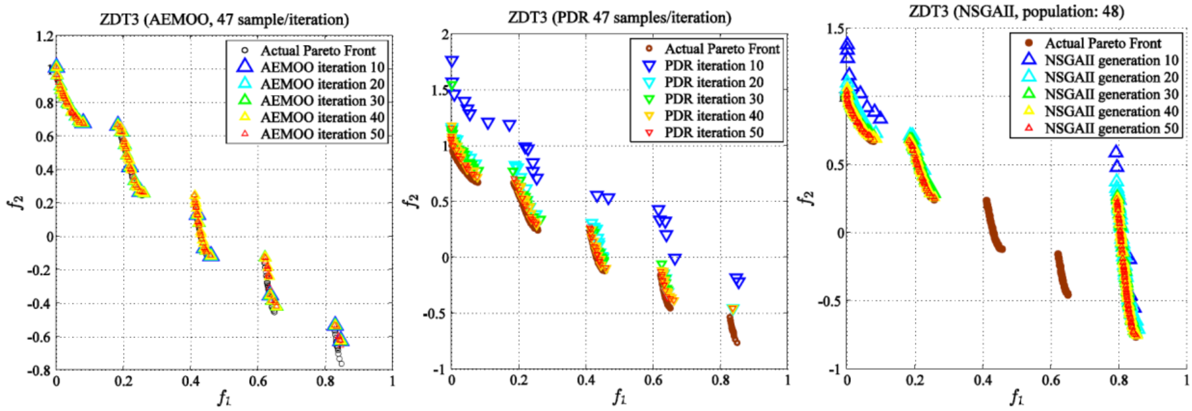

Figure 7: Example 4. Metamodel-assisted AEMOO (left), PDR (center) and direct NSGAII (right).

5.5. Example 5. 7 variable tolerance optimization of Chevrolet C2500 truck (Metamodel-assisted AEMOO)
AEMOO is used for multi-objective tolerance optimization of a Chevrolet truck. LS-OPT is used to build global metamodels, based on the truck's responses at 1500 samples (using LS-DYNA). MOO is run on the metamodels.. Mass is minimized while tolerance is maximized (Eq. (5)). Relative tolerance and 6 thicknesses x are the variables.

$$\begin{aligned} &\min_{tolerance,\bar{\mathbf{x}}} \quad \{-tolerance, scaled\_mass(\bar{\mathbf{x}})\} \\ &s.t. \quad P(scaled\_mass(\mathbf{x}) > 0.9) \le P_{target} \\ &\qquad P(scaled\_stage1\_pulse(\mathbf{x}) > 1) \le P_{target} \\ &\qquad P(scaled\_stage2\_pulse(\mathbf{x}) > 1) \le P_{target} \\ &\qquad P(scaled\_disp(\mathbf{x}) > 1) \le P_{target} \qquad \text{where } P_{target} = 0 \text{ in this work} \end{aligned} \tag{5}$$

In figure 3, the vehicle parts to be optimized are shown along with the optimization results. The Pareto front obtained using AEMOO and NSGAII are shown. 100 samples per iteration are used to solve this example to ensure at least one sample of each class in the initial sampling. Other approaches to avoid this restriction are possible, but are outside the scope of this paper. AEMOO results are provided for 30 iterations that were completed at the time of writing this paper and compared to NSGAII is run up to 50 generations. The PO front using AEMOO has a better spread, and diversity compared to the NSGAII front even at the 50^{th} generation, which shows its superior performance. At the same stage (30^{th} iteration), the PO front using AEMOO is clearly better. The PO front consists of a knee at 6% tolerance suggesting it to be a good design, as there is rapid mass increase beyond it.

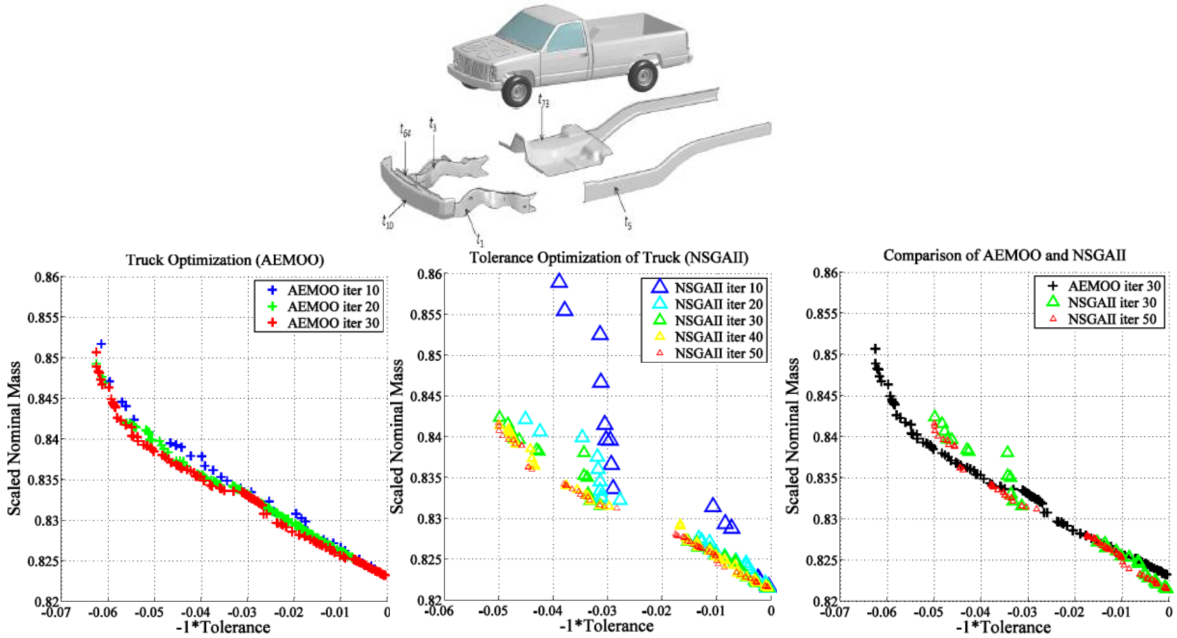

Figure 8: Truck to be optimized (top), AEMOO (left), PDR (center), NSGAII (center) and overlaid fronts (right).

5. Concluding Remarks

A new classification-based adaptive sampling approach to solve MOO problems is presented. The proposed AEMOO method has several advantages compared to existing methods due to this radically different approach. Two variations of the method are presented - direct and metamodel assisted. The method's efficacy is validated using standard examples of up to 30 variables. It has been shown to clearly outperform existing methods like NSGAII and PDR in terms of efficiency. Additionally, ability to locate disjoint Pareto fronts has been shown. Ability to solve constrained MOO has also been shown using a tolerance-based crashworthiness optimization example. As AEMOO explicitly defines the ND region boundaries in the design space, it also naturally facilitates real-time Pareto optimality decisions. This work is expected to open new avenues for research in the field of MOO. As the sampling scheme is partly based on design space classification, which discards significant regions of the space as dominated, it is expected to be affected by objective space dimensionality to a lesser extent. Future work is needed to validate this hypothesis. Additionally there is scope for further improvement in constraint handling.

6. Acknowledgements

The author is thankful to Dr. Nielen Stander and Mr. Imtiaz Gandikota of LSTC for their valuable feedback and for allowing the use of their machines for running some of the comparative studies.

7. References

[1] Deb, K. (2001). Multi-objective optimization using evolutionary algorithms (Vol. 16). John Wiley & Sons.
[2] Laumanns, M. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm.
[3] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation*, *IEEE Transactions on*, *6*(2), 182-197.
[4] Stander, N., Roux, W.J., Basudhar, A., Eggleston, T., Goel, T., Craig, K.J. LS-OPT User's Manual Version 5.0, April 2013.
[5] Li, M., Li, G., & Azarm, S. (2008). A kriging metamodel assisted multi-objective genetic algorithm for design optimization. *Journal of Mechanical Design*, *130*(3), 031401.
[6] Knowles, J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. Technical report TR-COMPSYSBIO-2004-01, September 2004.
[7] Stander, N. An Adaptive Surrogate-Assisted Strategy for Multi-Objective Optimization. *10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, 2013
[8] Vapnik, V.N., and Vapnik, V.. *Statistical learning theory*. Vol. 1. New York: Wiley, 1998.
[9] Basudhar, A., and Missoum, S. Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. *Computers & Structures* 86.19 (2008): 1904-1917.
[10] Basudhar, A., Dribusch, C., Lacaze, S. and Missoum, S. Constrained efficient global optimization with support vector machines. *Structural and Multidisciplinary Optimization* 46, no. 2 (2012): 201-221.
[11] National Crash Analysis Center. Finite element model archive, www.ncac.gwu.edu/vml/models.html. 2008.