10th World Congress on Structural and Multidisciplinary Optimization
May 19 -24, 2013, Orlando, Florida, USA

A Predictive Pareto Dominance Based Algorithm for Many-Objective Problems

Edgar Galvan[1], Erin M. Bose[2] Richard J. Malak Jr.[3]

[1] Texas A&M University, College Station, Texas, USA, e_galvan@tamu.edu
[2] Texas A&M University, College Station, Texas, USA, emboese@tamu.edu
[3] Texas A&M University, College Station, Texas, USA, rmalak@tamu.edu

1. Abstract
Multiobjective genetic algorithms (MOGAs) have successfully been used on a wide range of real world problems. However, it is generally accepted that the performance of most state-of-the-art multiobjective genetic algorithms tend to perform poorly for problems with more than four objectives, termed many-objective problems. The contribution of this paper is a new approach for identifying members close to the true Pareto frontier. The proposed criterion, termed predicted Pareto dominance, is extendable to many-objective problems and is relatively fast to compute. Predicted Pareto dominance machine learning to generate a model of the technology frontier. The results show that, on two test problems, predicted Pareto dominance can result in improved performance of a MOGA for many-objective problems; in one of the test problems, increased performance was seen in problems with three or four objectives. The computational study also reveals at least one class of problems for which a naïve implementation of predicted Pareto dominance results in poor performance.
2. Keywords: Multi-objective Programming, Genetic, High Dimensions

3. Introduction
Most real-world engineering optimization problems require that the designer simultaneously consider multiple objectives. Often, these engineering problems are cast as a multi-objective optimization problems (MOPs), where the objectives typically conflict with one another. Evolutionary computation techniques have been very successful in solving MOPs with up to three objectives [1, 2]. However, the performance of most multi-objective optimization algorithms (MOAs) deteriorates severely with increased conflicting objectives [3]. As a result, there is increasing interest on multi-objective optimization for a large number of objectives, known as *many-objective optimization*. The principle cause of the performance degradation in many-objective problems is related to the loss of the ability of Pareto dominance to adequately identify members that are likely close to the true Pareto frontier [4]. In many-objective problems, almost all solutions in the population are nondominated, weakening the selection bias towards the true nondominated Pareto frontier.
Several approaches for dealing with many-objective optimization problems have been proposed in the literature. One approach is to reduce the number of objectives used for decision making by identifying redundant objectives [5, 6]. Dimensionality reduction, however, is not suited for problems with independent and conflicting objectives. Another approach is to increase the selection pressure towards the true Pareto frontier by modifying the Pareto dominance relation and rank definition, e.g., ε *-dominance,* α *-dominance,* etc [7, 8]. Other approaches use aggregation and ranking schemes dependent on weight or target vectors [9, 10]. Although these approaches have been shown to outperform Pareto dominance based approaches in many-objective problems, the quality of the solution depends on several operating parameters specified by the designer. Finally, hypervolume indicator based search algorithms have been reported to outperform established multi-objective algorithms such as NSGA-II and SPEA2 for many-objective problems [11]. However, the computational expense of computing the hypervolume indicator increases exponentially with the number of objectives. As a result, a hypervolume indicator approximation algorithm based on Monte Carlo sampling has been developed [12]. Still, the hypervolume indicator may be too restricted so that some interesting information about the current population may not be fully explored.
The aim of this paper is to introduce a new approach for identifying members close to the true Pareto frontier—termed predicted Pareto dominance—that may resolve some of the drawbacks associated alternative approaches in the literature. Using predicted Pareto dominance, dominance analysis is performed using design sites that are *predicted* to be feasible rather than current members of the population. Those members that are not dominated by those that are predicted to be feasible lie on the Pareto frontier of the predicted feasible set. These nondominated members are typically a small fraction of the total population, even in many-objective problems. We use a machine learning technique to predict the feasible domain in the performance space by using the current population as training data. The machine learning technique is incremental allowing for fast online updating as

new populations are added to the training set. The purpose of this paper is to perform an investigate into some of the benefits and limitations of the proposed approach, as well as to identify questions and challenges that should be addresses to make the approach practical.

The paper is organized as follows. Section 4 is a brief introduction to multiobjective optimization and an overview of several approaches that have been proposed to address the performance degradation of Pareto dominance in many-objective problems. In Section 4, we introduce the proposed approach, predicted Pareto dominance. Section 6 is the results and analysis of the numerical experiments performed to investigate the performance of predicted Pareto dominance. Section 7 is a summary of the paper.

4. Multi-objective Genetic Optimization and Dominance Criteria

Without loss of generality, the multi-objective optimization problem can be stated as follows. Suppose designer preferences are monotonically decreasing in each performance attribute, $\mathbf{y} = \left[F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_J(\mathbf{x})\right]^T$, where $\mathbf{x}$ is the vector of optimization or decision variables. The search problem for the Pareto frontier is as follows:

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{Minimize}} & F_j(\mathbf{x}), & j = 1,2,\dots,J, \\ \text{subject to} & g_k(\mathbf{x}) \le 0, & k = 1,2,\dots,K, \\ & h_m(\mathbf{x}) = 0, & m = 1,2,\dots,M, \\ & x_i^{(\mathrm{L})} \le x_i \le x_i^{(\mathrm{U})}, & i = 1,2,\dots,n. \end{array} \tag{1}$$

A solution $\mathbf{x}$ is a vector of n decision variables, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$. The last set of constraints are variable bounds restricting each decision variable x_i within the lower and upper bounds $x_i^{(\mathrm{L})}$ and $x_i^{(\mathrm{U})}$, respectively [1]. The optimization problem has multiple objectives, $F_j(\mathbf{x})$ for $j = 1,2,\dots,J$, that must be minimized simultaneously. The problem has an infinite number of solutions that define the Pareto frontier.

The goal of a multi-objective genetic algorithm (MOGA) is to find a set of discrete solutions that are representative of the Pareto frontier. The goals of a MOGA are to: **(1)** Find a set of solutions close to the Pareto frontier and **(2)** maintain diversity in the solution set.

4.1. Pareto Dominance

Most state-of the-art MOGAs such as NAGAII[13] and SPEA[14] rely on the concept of dominance to assign higher fitness values to members closer to the true Pareto frontier. In these algorithms, two members are compared on the basis of whether one dominates the other or not. The nondominated member is assigned a higher fitness value. The second goal is typically achieved through the use of diversity preservation operator. The nondominated members are assigned higher fitness values in an attempt to bias the population towards the true Pareto frontier. Pareto dominance is defined as follows. Suppose designer preferences are monotonically decreasing in each performance attribute y_i for $i = 1, \dots, M$. If $\mathbf{y} = [y_1, y_2, \dots, y_M]$ denotes a vector of performance attributes and $\mathbf{Y} \subseteq \mathbb{R}^M$ is the set of all feasible performance attributes, then classical Pareto dominance is defined as follows [15]:

Definition 1. *An alternative having attributes $\boldsymbol{y}" \in \boldsymbol{Y}$ is Pareto dominated by one with attributes $\boldsymbol{y}' \in \boldsymbol{Y}$ if and only if, $y_i' \le y_i"$ $\forall i = 1, \dots, M$ and $y_i' < y_i"$ $\exists i = 1, \dots M$.*

The set of all feasible solutions that are Pareto optimal is the Pareto frontier. The Pareto dominance criterion is simple and easy to program for an arbitrary number of objectives, does not require the adjustment of operating parameters, and relatively low computational complexity. However, it has been shown that the fraction of the population that is nondominated increases with the dimensionality of the objective space [16]. In order to better understand the ability of Pareto dominance to order solutions in terms of objective function values, we randomly generate 200 designs in an M-dimensional unit hypercube for $M = 2,6,\dots,30$. The average percentage of nondominated designs over 10 runs is presented in Figure 1. The percentage of nondominated solutions increases with the number of objectives. For dimensions greater than 10, nearly all solutions are nondominated. The high number of nondominated solutions weakens the selection bias towards the true Pareto frontier. In the following subsection is a brief review of alternative dominance criteria.

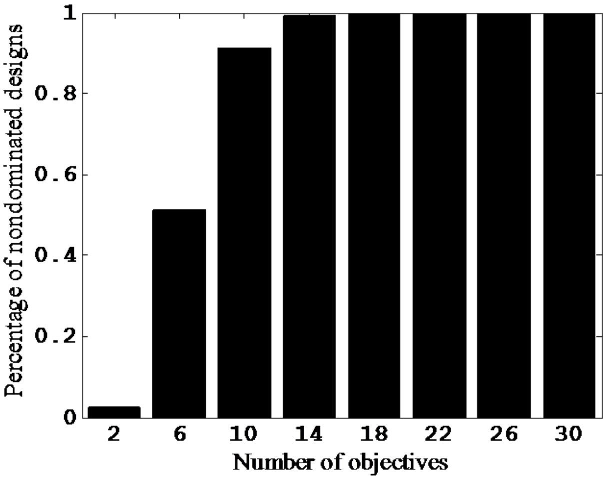

Figure 1. Mean percentage of Pareto nondominated designs from a set of 200 uniformly distributed designs in an M-dimensional unit hypercube.

4.2. Alternative Dominance Criteria

In order to address the limitations of Pareto dominance, several alternatives dominance criteria have been proposed in the literature, e.g., $\boldsymbol{\alpha}$ *-dominance,* $\boldsymbol{\varepsilon}$ *-dominance, favour,* etc. [7, 8, 17]. Under α-dominance lower and upper bounds are set for trade-off rates between two objectives such that solutions characterized by only small improvements in some objectives are rejected. This approach requires that the designer provide n^2 operating parameters that define the trade-off rate between the objectives, where n is the number of objectives. Another alternative, $\boldsymbol{\varepsilon}$ *-dominance,* is a relaxation of strict dominance. There are several different versions of $\boldsymbol{\varepsilon}$ -dominance. Under additive $\boldsymbol{\varepsilon}$ *-dominance,* an alternative $\boldsymbol{y}' \in \boldsymbol{Y}$ is $\boldsymbol{\varepsilon}$ -dominated by $\boldsymbol{y}'' \in \boldsymbol{Y}$ if and only if, $y_i' - \epsilon \le y_i''$.for all $i = 1, \dots M$. As a result of their superior ordering ability, these dominance criteria have been shown to be more appropriate than Pareto dominance for many-objective problems [18]. However, the performance of the dominance criteria depends on values of several operating parameters specified by the designer.

The relation *favour* is based on the number of objectives for which one solution is better than another. An alternative $\boldsymbol{y}' \in \boldsymbol{Y}$ is dominated by $\boldsymbol{y}'' \in \boldsymbol{Y}$ if and only if the following relation holds $|\{j: \boldsymbol{y}'' < \boldsymbol{y}'\}, 1 \le j \le M| < |\{i: \boldsymbol{y}' < \boldsymbol{y}'\}, 1 \le i \le M|$. This dominance criteria introduces different ranks to the non-dominated solutions leading to increased selection bias towards the true Pareto frontier. However, it also results in a decrease in solution diversity, sometimes converging to very few solutions [19]. Many proposed alternative dominance criteria involve some modification of Pareto dominance. In order to increase selection bias towards the Pareto frontier, Pareto dominance is modified as shown in Figure 2. Using (a)Pareto dominance all the alternatives are nondominated; using (b)modified Pareto dominance alternative c is dominated. Similar to the $favour$ relation, this approach increases solution bias towards the true frontier but at the cost of losing diversity in the solution set [20].

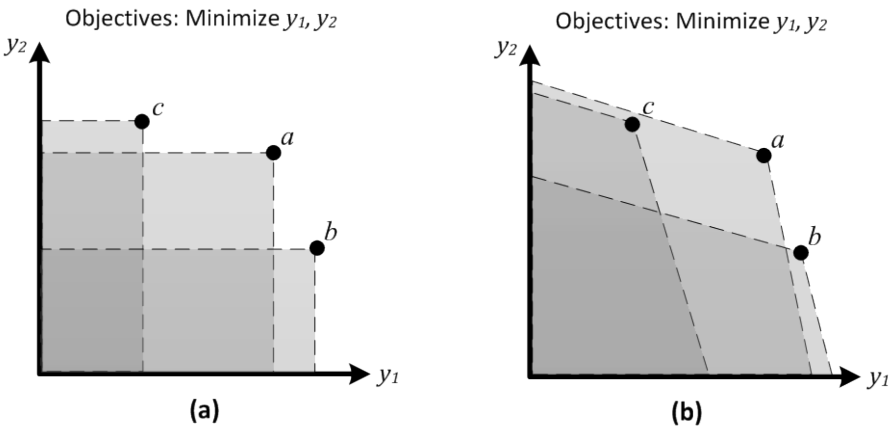

Figure 2. Illustration of (a)Pareto dominance, and (b) modified Pareto dominance. Alternative c is nondominated

using Pareto dominance criterion but dominated using a modified Pareto dominance criterion.

4.3. Quality Measure Directed Approaches

There exist in the literature a number of quality measures for the set of nondominated solutions [21]. Several approaches form many-objective problems proposed in the literature directly optimize the quality measure. The hypervolume indicator is a quality measure that is often used to assign fitness values to the members of the population. The hypervolume indicator is hypervolume of the space dominated by a solution set bounded by a reference point. Optimizing the hypervolume indicator is an attractive approach since: **(1)** it is a comprehensive search metric that captures convergence to the Pareto frontier and diversity in the solution set, and **(2)** it is the only known strictly monotonic quality measure with respect to the Pareto frontier. The hypervolume indicator of solution set S is larger than that of solution set S' if and only if S is better than S' [22]. Several hypervolume indicator based MOGAs have been proposed [23-25]. The principal drawback of hypervolume indicator based approaches is the high computational expense for the hypervolume calculation; most algorithms are not feasible for problems with greater than 6 objectives. The HypE algorithm developed by Bader and Zitzler addresses this limitation by approximating the hypervolume using Monte Carlo sampling [12]. By approximating the hypervolume rather than computing it directly, the many-objective problems become feasible with hypervolume indicator based search algorithms. However, the hypervolume indicator may be too restricted so that some interesting information about the current population may not be fully explored. Furthermore, there is some debate as to whether maximizing the hypervolume indicator is equivalent to the overall objective of finding a 'good' approximation of the Pareto front in high dimensional spaces [26].

In the following section, we introduce a new approach for determining the dominated members. Under the proposed approach, a prediction of the feasible set is generated using a machine learning technique. Then Pareto dominance analysis is performed using the set of solutions that are *predicted* to be feasible rather than the current members of the population. The approach—termed *predicted Pareto dominance*—alleviates the deterioration Pareto dominance for many-objective problems. The approach is potentially fast to evaluate and only introduces single new user specified operating parameter. The purpose of this paper is to investigate the performance of the approach and to identify challenges that must be addresses in order to make the approach practical.

5. Predicted Pareto Dominance

Under the proposed approach, dominance analysis be performed using design sites that are *predicted* to be feasible rather than current members of the population. In the proposed algorithm, we use a machine learning technique to predict the feasible domain in the performance space using the current population as training data. A nondominated member is one that is not Pareto dominated by any point in the predicted feasible domain.

5.1. Determining the Predicted Feasible Set

In order to determine the predicted feasible set, we implement kernel based support vector domain description (SVDD) proposed by Tax and Duin [27]. The SVDD method is a machine learning technique for modeling the boundary of a set of data in a Euclidian space. For a detailed description of the procedure for generating domain descriptions using this technique see [27, 28]. The SVDD method determines whether a new point is in the data domain by using a hypersphere that envelops the training data. One tries to find the minimum-radius hypersphere that contains a set of N data points, $\{\mathbf{x}_i|, i = 1, \dots, N\}$. The domain can then be represented by a hypersphere center at $\mathbf{a}$ and radius R. Thus the most rudimentary constraint is

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \le R^2 \ \ \forall i \, . \tag{2}$$

However, because a hypersphere is typically a poor representation of the domain, a kernel function is used to nonlinearly remap the training data into a higher-dimensional feature space where a hypersphere is a good model. There are several valid kernel functions common in the literature [29]. The proposed algorithm uses the Gaussian kernel function

$$K_G(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) = e^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2}, \tag{3}$$

where $\mathbf{\Phi}(\cdot)$ is the nonlinear mapping from the data space to the feature space. The q parameter determines how "tightly" or "loosely" the domain description is fit around the training data. The constraint in Eq. (2) becomes

$$\|\Phi(\mathrm{x_i}) - \mathbf{b}\|^2 \le \mathrm{R}^2 + \xi_\mathrm{i} \ \ \forall \mathrm{i} \, , \tag{4}$$

where $\mathbf{b}$ is the centroid of the feature space hypersphere and $\boldsymbol{\xi_i}$ are slack variables introduced that allow the exclusion of outliers. Rewriting in terms of the kernel function, the Wolfe dual problem can be developed from Eq. (4) as

$$\max_{\beta_i} W = \sum_i \beta_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{5}$$

subject to the constraints $0 \le \beta_i \le C \; \forall \; i$ and $\sum_i \beta_i = 1$. For a detailed description of the method for formulating the Wolfe dual problem see [30]. For each data point, $\mathbf{x}_i$ for $i = 1, \ldots, N$, there are three possible classifications:

- It is inside the hypersphere, which is indicated by $\beta_i = 0$,
- It is on the boundary of the hypersphere, which is indicated by $0 < \beta_i < C$,
- It is an outlier outside of the hypersphere, which is indicated by $\beta_i = C$.

Data on the boundary of the hypersphere are called *support vectors* and are essential to the domain description representation. The squared distance of the feature space image of a point, $\mathbf{z}$, to the centroid of the hypersphere is

$$R^2(\mathbf{z}) = K(\mathbf{z}, \mathbf{z}) - 2\sum_i \beta_i K(\mathbf{x}_i, \mathbf{z}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{6}$$

A new test point is inside the domain description if the distance from the feature space image of test point to the hypersphere centroid and is less than the radius of the hypersphere. The expression for classification, Eq. (6), is a simple algebraic expression that is fast to evaluate.

Outliers are data points for which $\beta_i = C$. The outliers are not part of the domain description. Choosing $C \ge 1$ yields no outliers. There must be at least one support vector, (i.e., $0 < \beta_i < C$), which means $\beta_i = C$ cannot occur for $C \ge 1$. In the proposed algorithm we set $C = 1$ implying that the training data does not contain outliers.

An important benefit of the SVDD method is that it can be constructed incrementally and decrementally [31]. This allows a relatively inexpensive update procedure to be used when members are added or removed from the population (domain description). For experimental analysis on the performance of the incremental SVDD procedure and the effects of the q parameter see [32].

5.2. Predicted Pareto Dominance Analysis

Using the concept of Pareto dominance, a nondominated member is one that is not Pareto dominated by any member that is predicted to be feasible. In this subsection, we will refer to *predicted Pareto nondominated* members simply as *nondominated* members and the *nondominated frontier of the predicted feasible members* as the *nondominated frontier*.

In the proposed algorithm, any point inside of the domain description is considered a predicted feasible member. Therefore, any population member in the interior of the domain description must be dominated. If every member in the population is part of the domain description, ($C \ge 1$) only members on the boundary of the domain description can be nondominated, i.e., support vectors. Therefore, the non-support vectors can be ignored when performing dominance analysis. Recall from Section 5.1 that the support vectors are classified when solving the Wolf dual problem, Eq. (5). This is significant since for most data sets only a small portion of the data will lie on the hypersphere boundary and be classified as support vectors.

Using the SVDD approach, the nondominated members of the population are the support vectors that lie on the nondominated frontier. A simple but computationally expensive way to test whether the support vector lies on the nondominated frontier is to sample the space that dominates the support vector and test for *predicted* feasibility. The space to be tested is bounded by the minimum bounding hyperrectangle or envelope that contains the training data. If any member in that space is feasible, we classify the member as dominated. We should note that testing for predicated feasibility, Eq. (7), is a simple algebraic expression that is fast to evaluate. Figure 3 is an illustration of this procedure for two members in the population that are support vectors. As illustrated in Figure 3, the sampling procedure for a support vector can be terminated as soon as we find a feasible member. Under this approach, if there are n support vectors, M objectives, and v samples along each dominator attribute, determining the nondominated members requires at most nv^M evaluations of Eq. (6). In practice the number of evaluations required is much lower than the upper limit since many of the support vectors are classified as dominated after the first sample, see the dominated member in Figure 3. Furthermore, the number of samples along each dimension, v, is typically low since the nondominated members are likely near the edges of the bounding hyperrectangle, see the nondominated member in Figure 3. Still, due to the exponential growth with dimensionality, this approach may not be feasible for problems with greater than six objectives.

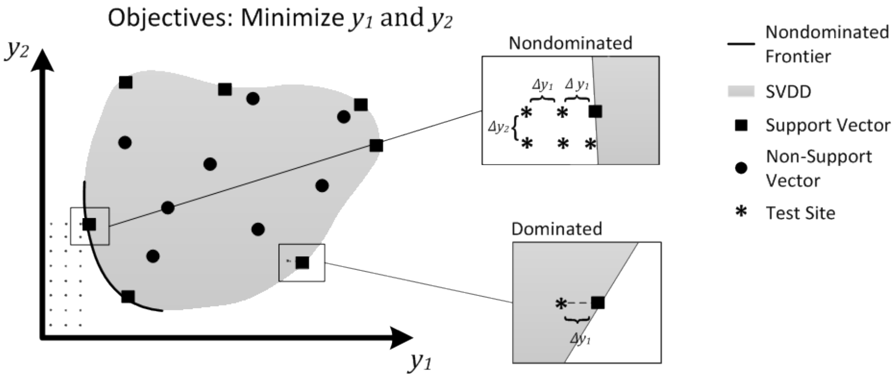

Figure 3. An illustration of predictive parameterized Pareto dominance.

A more sophisticated and potentially less computationally expensive approach would be to use the gradient of the distance to the feature space hypersphere centroid to search for predicted feasible solutions that dominate the test member. However, since the focus of this paper to determine whether the concept of predicted dominance merits further investigation, we leave finding a sampling or search procedure that reduces the computational growth pattern of this step for future work. In the computational studies to follow, only the dominating neighborhood of a test point is sampled and checked for predicted feasibility. The predicted nondominated members are the support vectors whose dominating neighbors are predicted to be infeasible. If the domain description is convex, sampling the dominating epsilon-neighborhood of a test point for feasibility is sufficient for classification over the entire predicted feasible set (domain description). In the following section, we investigate the performance of predicted Pareto dominance using test problems from the literature.

6. Numerical Experiments and Analysis

In this section, we investigate some of the performance characteristics of predicted Pareto dominance. In order to better understand the ability of predicted Pareto dominance to order solutions in terms of objective function values, we perform the same experiment reported in Figure 1 using predicted Pareto dominance. The results are illustrated in Figure 4. The results show that using predicted Pareto dominance, only a small fraction of solutions are nondominated, even in very high dimensions. However, this result does not indicate whether the nondominated solutions identified are useful in many-objective optimization problems. In order to investigate the usefulness of predicted Pareto dominance in many-objective optimization, we investigate the performance of predicted dominance on scalable test problems.

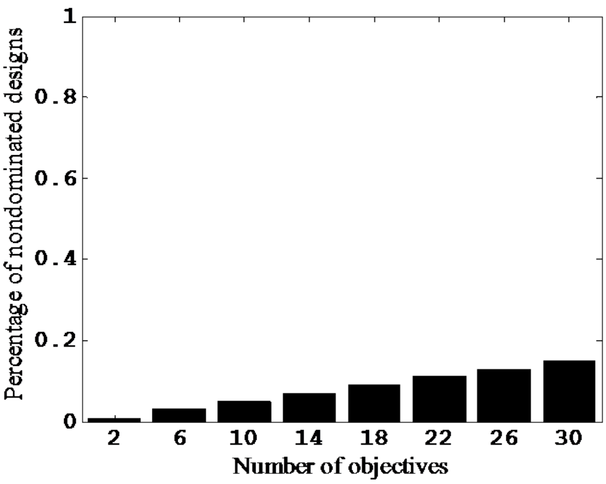

Figure 4. Mean percentage of predictive Pareto nondominated designs from a set of 200 uniformly distributed designs in an M-dimensional unit hypercube.

Several scalable test problems have been proposed in the literature [33]. The scalable test problems used in this case study are DTLZ1, DTLZ2, and DTLZ4. For each test problem there are $n = M + k - 1$ decision variables where M is the number of objective function and k specifies the distance to the Pareto front. As specified by Deb *et al.*, $k = 5$ for DTLZ1 and $k = 10$ for DTLZ2 and DTLZ4. The domain of the n decision varialbes is $[0,1]^M$.

For performance assessment, the hypervolume indicator is used. The hypervolume indicator is hypervolume of the space dominated by a solution set bounded by a reference point r. The hypervolume indicator is a useful performance metric since it measures not only convergence to the Pareto frontier but also diversity in the solution set. The reference point used to compute the hypervolume are $r = 0.7\hat{}M$ for DTLZ1 and $r = 1.1^M$ for DTLZ2 and DTLZ4. The hypervolume is normalized with respect to the hypervolume of the analytical optimal solution.
In order to investigate the usefulness of the predicted Pareto dominance concept, we substitute Pareto dominance used in the NSGAII strategy for predicted Pareto dominance. In order to distinguish between the predictive Pareto dominance based algorithm and NSGAII, we will refer to the former as Predicted Pareto Genetic Algorithm (PPGA).
A total of 20,000 function evaluations and a population size of $N = 100$ is used for all test cases. For each test problem, 10 runs are performed. Each test problem is performed at objective values of $M = 3,4,5,6,$ and 7. The HypE hypervolume estimation algorithm by Bader and Zitzler algorithm is used to approximate the hypervolume of the solution sets. A total of 1,000,000 samples are used per Pareto set approximation is used leading to a very small uncertainty, below 10^{-3} in relation to the hypervolume indicator. The results are reported in Table 1.
The results for problems DTLZ2 and DTLZ4 show that mean hypervolume of the PPGA generated solutions outperformed NSGAII for all dimension tested, including three and four (which are typically not considered many-objective problems). On test problem DTLZ4, PPGA outperformed NSGAII only on many-objective problems.

Table 1: Relative Hypervolume for the Test problems

Obj.	Algorithm	DTLZ2		DTLZ4	
		Mean	*Std.dev*	*Mean*	*Std.dev*
3	NSGAII	0.97562	0.0070795	0.85408	0.27688
	PPGA	0.9858	0.008013	0.79530	0.18356
4	NSGAII	0.90672	0.018814	0.94189	0.006415
	PPGA	0.93351	0.016809	0.83986	0.073458
5	NSGAII	0.55291	0.11036	0.72266	0.05142
	PPGA	0.88385	0.020746	0.85803	0.081193
6	NSGAII	0.20155	.05314	0.63394	0.093642
	PPGA	0.940185	0.012909	0.88019.	0.051374
7	NSGAII	0.10588	0.029444	0.38505	0.13833
	PPGA	0.92077	0.013323	0.88040	0.03371

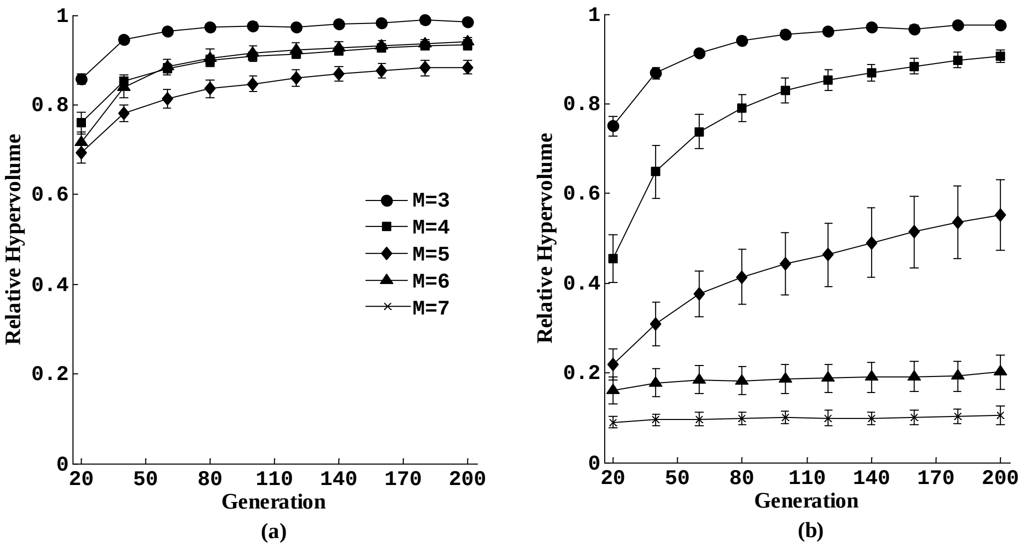

Figure 5. Relative hypervolume of the nondominated solution set fort test problem DTLZ2 at $M = 2,3,4,5,6,7$ dimensions for **(a)** PPGA and **(b)** NSGAII. The error bars indicate 95% confidence interval.

Figure 5 depicts the relative hypervolume of the nondominated solution set fort test problem DTLZ2 at $M = 2,3,4,5,6,7$ dimensions for (a) PPGA and (b) NSGAII. The figure shows the convergence rate of both approaches on the DTLZ2 test problem. On test problem DTLZ2, the convergence rate of PPGA is similar for all dimensions tested.
The results for test problem DTLZ1 are not reported in Table 1. This is because of the poor performance of the PPGA algorithm on that test problem (hardly any solutions were generated within the reference piont). The performance of NSGAII on test problem DTLZ1 is reported by Wagner *et al.* [11]. For test problem DTLZ1, predictive Pareto dominance failed to bias selection towards the true Pareto frontier. The feature that we suspect caused difficult for predictive Pareto dominance is the large volume of the initial predicted feasible set in the objective space relative to the nondominated frontier. In test problem DTLZ1, the initial randomly generated solutions are widely spaced in the objective space. When the domain description is generated around the initial population, the boundary members (support vectors) are even more widely spaced in the objective space. As a result, predictive Pareto dominance fails to rank the solutions, since almost all solutions are classified as dominated. Figure 6 is an illustration of this issue. This drawback could potentially be addressed by introduction an approach for eliminating population members that are far from the nondominated frontier from the domain description. This is left to future work.

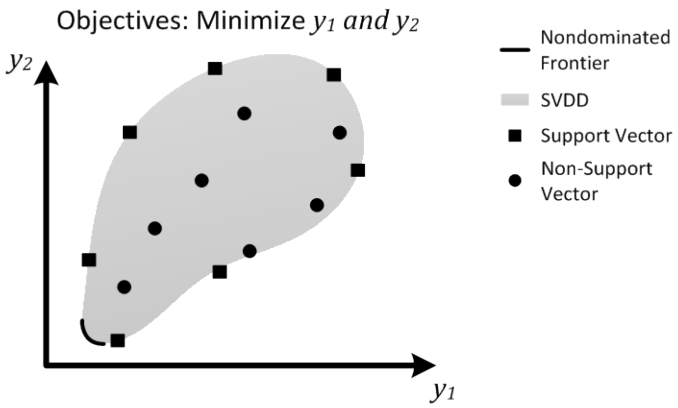

Figure 6. Illustration of a case where predicted Dominance fails to identify nondominated members.

7. Summary
The contribution of this paper is a new approach for identifying members close to the true Pareto frontier—termed predicted Pareto dominance—that may resolve some of the drawbacks associated alternative approaches in the literature. Predicted Pareto dominance is extendable to many-objective problems and is relatively fast to compute. Using predicted Pareto dominance, dominance analysis is performed using design sites that are predicted to be feasible rather than current members of the population. A machine learning technique is used to predict the feasible domain in the performance space by taking the current population as training data. An algorithm using predicted Pareto dominance, PPGA, was compared to NSGAII on several scalable test problems, i.e., DTLZ1, DTLZ2, and DTLZ4. The results show that on test problem DTLZ2, predicted Pareto dominance results in improved performance of a MOGA not only in many-objective problems but also in three and four objectives. The results from test problem DTLZ4 show that predictive Pareto dominance has lower degradation in performance compared to Pareto dominance. The results from test problem DTLZ1, revealed that a naïve implementation of predicted Pareto dominance results in poor performance for problems where the initial predicted feasible set, based on the initial random population, is relatively large compared to the predicted nondominated frontier. If this occurs, the predictive dominance approach is likely to classify nearly all population members as dominated, weakening the selection pressure towards the Pareto frontier and decreasing diversity in the solution set. This drawback could be addressed by introduction an approach for eliminating population members that are far from the nondominated frontier from the domain description.

8. Acknowledgements
This work is supported by the National Science Foundation under Grant No. EFRI-1240483.

9. References
[1] Deb, K., 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley & Sons, Chichester.
[2] Coello, C. a. C., Lamont, G. B., and Veldhuizen, D. a. V., 2006, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer-Verlag New York, Inc.
[3] Ishibuchi, H., Tsukamoto, N., and Nojima, Y., 2008, "Evolutionary Many-Objective Optimization: A Short

Review," Proc. Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, pp. 2419-2426.
[4] Hughes, E. J., 2005, "Evolutionary Many-Objective Optimisation: Many Once or One Many?," Proc. Evolutionary Computation, 2005. The 2005 IEEE Congress on, 1, pp. 222-227 Vol.1.
[5] Brockhoff, D., and Zitzler, E., 2007, *Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem*, Springer Berlin Heidelberg, Chap. 68.
[6] López Jaimes, A., and Coello Coello, C. A., 2009, "Study of Preference Relations in Many-Objective Optimization," Proc. Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pp. 611-618.
[7] Ikeda, K., Kita, H., and Kobayashi, S., 2001, "Failure of Pareto-Based Moeas: Does Non-Dominated Really Mean near to Optimal?," Proc. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2, pp. 957-962 vol. 2.
[8] Laumanns, M., Thiele, L., Deb, K., and Zitzler, E., 2002, "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization," Evolutionary Computation, **10**(3), pp. 263-282.
[9] Hughes, E. J., 2007, "Msops-Ii: A General-Purpose Many-Objective Optimiser," Proc. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 3944-3951.
[10] Hughes, E. J., 2003, "Multiple Single Objective Pareto Sampling," Proc. Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, 4, pp. 2678-2684.
[11] Wagner, T., Beume, N., and Naujoks, B., 2007, "Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization," Proc. Evolutionary Multi-Criterion Optimization, pp. 742-756.
[12] Bader, J., and Zitzler, E., 2011, "Hype: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization," Evolutionary Computation, **19**(1), pp. 45-76.
[13] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-Ii," IEEE Transactions on Evolutionary Computation, **6**(2), pp. 182-197.
[14] Zitzler, E., 1999, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, Shaker Ithaca.
[15] Malak, R. J., and Paredis, C. J. J., 2010, "Using Parameterized Pareto Sets to Model Design Concepts," Journal of Mechanical Design, **132**(4).
[16] Kukkonen, S., and Lampinen, J., 2007, "Ranking-Dominance and Many-Objective Optimization," Proc. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 3983-3990.
[17] Drechsler, N., Drechsler, R., and Becker, B., 2001, "Multi-Objective Optimisation Based on Relation Favour," Proc. Evolutionary Multi-Criterion Optimization, pp. 154-166.
[18] Batista, L. S., Campelo, F., Guimaraes, F. G., and Ramirez, J. A., 2011, "A Comparison of Dominance Criteria in Many-Objective Optimization Problems," Proc. Evolutionary Computation (CEC), 2011 IEEE Congress on, pp. 2359-2366.
[19] Corne, D. W., and Knowles, J. D., 2007, "Techniques for Highly Multiobjective Optimisation: Some Nondominated Points Are Better Than Others," Proc. Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 773-780.
[20] Ishibuchi, H., Tsukamoto, N., and Nojima, Y., 2008, "Evolutionary Many-Objective Optimization: A Short Review," Proc. Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, pp. 2419-2426.
[21] Knowles, J. D., Thiele, L., and Zitzler, E., 2006, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.
[22] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G., 2003, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," Evolutionary Computation, IEEE Transactions on, **7**(2), pp. 117-132.
[23] Emmerich, M., Beume, N., and Naujoks, B., 2005, "An Emo Algorithm Using the Hypervolume Measure as Selection Criterion," Proc. Evolutionary Multi-Criterion Optimization, pp. 62-76.
[24] Igel, C., Hansen, N., and Roth, S., 2007, "Covariance Matrix Adaptation for Multi-Objective Optimization," Evolutionary Computation, **15**(1), pp. 1-28.
[25] Brockhoff, D., and Zitzler, E., 2007, "Improving Hypervolume-Based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods," Proc. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 2086-2093.
[26] Bringmann, K., and Friedrich, T., 2013, "Approximation Quality of the Hypervolume Indicator," Artificial Intelligence, **195**(0), pp. 265-290.
[27] Tax, D. M. J., and Duin, R. P. W., 1999, "Support Vector Domain Description," Pattern Recognition Letters, **20**(pp. 1191-1199.
[28] Malak, J. R. J., and Paredis, C. J. J., 2010, "Using Support Vector Machines to Formalize the Valid Input

Domain of Predictive Models in Systems Design Problems," Journal of Mechanical Design, **132**(10), pp. 101001-14.
[29] Scholkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J., 2000, "Support Vector Method for Novelty Detection," Proc. Advances in neural information processing systems, pp. 582-588.
[30] Wolfe, P., 1961, "A Duality Theorem for Nonlinear Programming," Quarterly of applied mathematics, **19**(3), p. 239.
[31] Cauwenberghs, G., and Poggio, T., 2000, "Incremental and Decremental Support Vector Machine Learning," Proc. Neural Information Processing Systems, pp. 409-415.
[32] Roach, E., Parker, R. R., and Malak, R. J., 2011, "An Improved Support Vector Domain Description Method for Modeling Valid Search Domains in Engineering Design Problems," Proc. ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, EC.
[33] Deb, K., Thiele, L., Laumanns, M., Zitzler, E., Abraham, A., Jain, L., and Goldberg, R., 2005, *Scalable Test Problems for Evolutionary Multiobjective Optimization Evolutionary Multiobjective Optimization*, Springer Berlin Heidelberg.