Multiobjective Optimization Using an Adaptive Weighting Scheme

Shubhangi Deshpande

Department of Computer Science, Virginia Polytechnic Institute & State University,

Blacksburg, VA, USA, shubhgd@cs.vt.edu

Layne T. Watson

Department of Computer Science, Mathematics, and Aerospace & Ocean Engineering,

Virginia Polytechnic Institute & State University, Blacksburg, VA, USA, ltw@cs.vt.edu

Robert A. Canfield

Department of Aerospace & Ocean Engineering, Virginia Polytechnic Institute & State University,

Blacksburg, VA, USA, bob.canfield@vt.edu

Abstract A new Pareto front approximation method is proposed for multiobjective optimization problems with bound constraints. The method employs a hybrid optimization approach using two derivative free direct search techniques, and intends to solve blackbox simulation based multiobjective optimization problems where the analytical form of the objectives is not known and/or the evaluation of the objective function(s) is very expensive. A new adaptive weighting scheme is proposed to convert a multiobjective optimization problem to a single objective optimization problem. Another contribution of this paper is the generalization of the star discrepancy based performance measure for problems with more than two objectives. The method is evaluated using five test problems from the literature. Results show that the method achieves an arbitrarily close approximation to the Pareto front with a good collection of well-distributed nondominated points for all five test problems.

Keywords: multiobjective optimization; Pareto optimality; direct search method; DIRECT; MADS; surrogates; triangulation.

1. Introduction

Multiobjective optimization problems (MOPs) arise in practically every area of science and engineering where optimal decisions need to be made in the presence of two or more conflicting objectives. A decision maker has to take into account different criteria that need to be satisfied simultaneously.

Let $\mathbb{R}^n$ denote real n-dimensional Euclidean space. For vectors $x^{(1)}, x^{(2)} \in \mathbb{R}^n$ write $x^{(1)} < x^{(2)}$ ($x^{(1)} \leq x^{(2)}$) if $\forall i,\ x_i^{(1)} < x_i^{(2)}$ $\left(x_i^{(1)} \leq x_i^{(2)}\right)$, $i = 1, \ldots, n$. Let $l,\ u \in \mathbb{R}^n$ with $l < u$ and $B = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$. Given p conflicting objective functions $f_i : B \to \mathbb{R}$, $i = 1, \ldots, p$, the multiobjective optimization problem is to simultaneously minimize all the f_i over B.

For a nontrivial multiobjective optimization problem with p conflicting objectives no single solution can optimize all p objectives simultaneously. Hence, a new notion of optimality, known as Pareto optimality (the set of optimal solutions is known as the Pareto front), is generally used in the context of multiobjective optimization problems. Pareto optimality is generally defined using a dominance relation as follows: Given two decision vectors $x^{(1)}$ and $x^{(2)}$, $x^{(1)}$ is said to dominate $x^{(2)}$, denoted as $x^{(1)} \prec x^{(2)}$, if and only if $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives, and $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective. A point $x^* \in B$ is a globally Pareto optimal solution if and only if there does not exist any x in the design space B such that $x \prec x^*$. A point $x^* \in B$ is a locally Pareto optimal solution if and only if for some $\epsilon > 0$ there does not exist any x in the neighborhood $\left\{x \mid \|x - x^*\| < \epsilon\right\} \cap B$ of x^* that dominates x^*.

The goal of a MOP is to find all the Pareto optimal solutions for all the conflicting objectives. In general, finding infinitely many solutions on a Pareto front is impossible, and finding even a large discrete subset is hard when the structure and/or the analytical form of the underlying objective functions is not known or is very complex (e.g., blackbox simulation optimization). An approximation to the true Pareto front is obtained as an alternative in such situations. The most common approach is to approximate the Pareto front by a discrete set of points. The other difficulty is the efficient generation of well distributed Pareto optimal solutions. In real design problems a decision maker might be able to consider only a subset of the available solution set. In this context, having a good representation of the entire Pareto front is a crucial requirement in order to make an informed decision. Efficiency (in terms of the total number of true function evaluations) of a multiobjective optimization algorithm is a key factor in practical multidisciplinary design optimization problems where a design cycle involves time consuming and expensive computations for each discipline. For example, in a conceptual aircraft design process several different disciplines influence each other resulting in several interdisciplinary iterations to reach a feasible optimal design. The task becomes even more complicated when the objective functions are evaluated using expensive blackbox simulations where analytical form of the objective functions is not available or is very complex.

The aim of this paper is to propose a new multiobjective optimization method that provides a well distributed representation of the Pareto front for multiobjective blackbox optimization with a minimal number of true function evaluations. The general idea is to systematically move towards the Pareto front at the end of every iteration by adaptively filling the gaps between the nondominated points obtained so far. The biobjective optimization method proposed in (Deshpande et al., 2013) and (Deshpande et al., 2013a) by the authors was based on the scalarization scheme of Ryu et al. (2009). However, the same adaptive weighting scheme does not work for multiobjective

problems with $p > 2$ objectives. The algorithm presented in this paper is an alternative approach for generalizing the adaptive weighting scheme of Ryu et al. (2009) to problems with more than two objectives. The algorithm employs a hybrid approach using two derivative free direct search techniques — a deterministic global search algorithm DIRECT (Jones et al., 1993) for global exploration of the design space and a local direct search method MADS (mesh adaptive direct search) (Audet and Dennis 2006) to exploit the design space by fine tuning the potentially optimal regions returned by DIRECT and to speed up the convergence. Inexpensive surrogates for the objective functions are used in order to minimize the number of expensive simulation runs. The proposed method uses the global search algorithm DIRECT as a sampling method instead of using traditional random sampling (e.g., Latin hypercube sampling) or factorial designs that are expensive in higher dimensional design spaces. Results show that the proposed method provides well distributed Pareto optimal points on the Pareto front for diverse types of problems. Another contribution is the use of a precise mathematical measure, known as star discrepancy, for the distribution of the Pareto optimal solutions. A biobjective version presented in (Deshpande et al., 2013) has been generalized in the present work for problems with more than two objectives. A detailed description is provided in Section 4.1.

The organization of this paper is as follows. Section 2 gives an overview of the existing multiobjective optimization approaches in general and the previous work by the authors on biobjective optimization problems in particular, and also provides background on surrogates and hybrid optimization approaches in the context of multiobjective optimization. Section 3 describes the proposed alternative scalarization scheme and presents the multiobjective optimization algorithm. Numerical evaluation on a set of test problems from the literature is presented, and results are discussed in Section 4. Section 5 offers concluding remarks.

2. Background

Many different methods have been suggested in the literature for solving multiobjective optimization problems. The solution approaches can be divided into two broad categories — evolutionary approaches and scalarization or aggregation approaches. Applying a Pareto based ranking scheme using genetic algorithms has become very common and popular in most of the evolutionary approaches (e.g., (Deb et al., 2002)), although some schemes are based on particle swarm intelligence and simulated annealing. An evolutionary approach yields a set of nondominated points at the end of each iteration, requires a very large number of true function evaluations, and does not guarantee optimality in general. Aggregate or scalarization approaches (Eichfelder, 2008) are considered to

be classical methods for solving multiobjective optimization problems. The general idea is to combine all objective functions to convert a multiobjective optimization problem into a single objective optimization problem. Thus each iteration yields a single solution by solving a single objective optimization problem. The most commonly used scalarization approach is the convex combination method where the multiobjective optimization problem is converted to a single optimization problem using a predefined weight vector. A major drawback of the convex combination method is that if the Pareto front has a nonconvex (or nonconcave) region then the method fails to produce any points in that region. Also uniformly distributed weights may not produce uniformly distributed optimal points on the front. To alleviate this problem an adaptive weighting scheme was suggested by Ryu et al. (2009) in their biobjective optimization algorithm PAWS. PAWS has an efficient weighting scheme, however, the central composite design based sampling strategy used in PAWS is impractical in higher dimensional search domains. Deshpande et al. (2013) and (2013a) proposed an algorithm for biobjective optimization that employs the adaptive weighting scheme of PAWS, but tries to overcome the limitation of PAWS with a novel sampling strategy and a global search method. The ordering property exploited in these algorithms ((Deshpande et al., 2013) and (Ryu et al., 2009)) for the scalarization of the objectives does not work in higher dimensional objective spaces (i.e., problems with more than two objectives). Hence, an alternative strategy is proposed in this paper that works for problems with any number of objectives.

Other methods like normal boundary intersection (NBI) (Das and Dennis, 1998) and its variations solve multiobjective optimization problems by constructing several aggregate objective functions. A scalarization method called BiMADS (Audet et al., 2010) solves a biobjective optimization problem through a series of single objective formulations using the direct search method MADS, and attempts to obtain a uniform coverage of the Pareto front, even in the case when the Pareto front is nonconvex or disjoint. However, results produced using BiMADS are dependent on the random sampling used in the algorithm (e.g., Latin hypercube sampling used for the very first run in NOMAD 3.5.1 may not produce consistent results across different computing systems), and BiMADS also exploits the ordering property available in the two–dimensional case, and hence is not applicable for problems with more than two objectives. Audet et al. (2010) suggested an alternate formulation in their algorithm MultiMADS to generalize the scalarization scheme of BiMADS to problems with more than two objectives. MultiMADS was proposed with an intention to overcome some of the issues in the method NBI, and could be considered as a state-of-the-art in the classical deterministic approaches for solving MOPs. MultiMADS (and BiMADS for the biobjective case) is used as a benchmark for the numerical evaluation of the proposed method.

Several real world scientific and engineering applications require analysis of spatially distributed data from expensive experiments or complex computer simulations that can take hours to run even on a fast workstation. This makes it difficult to explore and produce well distributed design combinations in high dimensional design spaces. Identifying the regions that contain good designs in such data-scarce domains by keeping the number of simulation runs to a minimum is a nontrivial problem. The proposed algorithm employs a systematic way to enrich an incomplete database by adaptively filling the gaps between the nondominated points obtained from available data. This paper intends to solve multiobjective optimization problems for such black box simulations where the structure of the objective functions is not known or is very complex. To tackle the problem of expensive simulation runs and to reduce the number of true function evaluations, computationally cheap(er) approximations (known as surrogates) of the underlying complex functions are used. Statistical sampling is a crucial part of the surrogate building process which becomes nontrivial in higher dimensional search domains. The novel idea proposed in (Deshpande et al., 2013) of using the optimization algorithm DIRECT as a sampling strategy is employed in the current work as well.

A popular approach in the optimization community, hybrid optimization, is where several different optimization techniques are combined to improve robustness and blend distinct strengths of different approaches (Gray and Fowler, 2011). This same notion has been extended to multiobjective optimization problems in (Wang et al., 2008). The Pareto front approximation method of this paper is a hybrid approach using two direct search methods, DIRECT (to explore the design space globally) and MADS (to fine tune the potentially optimal regions returned by DIRECT), to balance between global and local searches and to improve the convergence rate.

3. The Multiobjective Optimization Algorithm

The proposed new method for multiobjective optimization applies to possibly nonsmooth functions. The ordering property $\big($for biobjective pairs$(f_1(x), f_2(x))\big)$ available in two dimensions is exploited in the biobjective case ((Deshpande et al., 2013), (Deshpande et al., 2013a)) to find the most isolated point from a set of nondominated points at each iteration. However, due to the lack of ordering in higher dimensions, the same strategy cannot be implemented in the multiobjective case. Hence, an alternative using the concept of triangulation, from simplical topology and computational geometry, is proposed in this paper. Section 3.1 describes this alternate approach, and the general scheme of the multiobjective optimization algorithm is presented in Section 3.2.

3.1. An alternative using Delaunay triangulation

In the biobjective optimization case, a point with the largest Euclidean distance from its neighbors is selected as the most isolated point provided that the nondominated data points are ordered in either of the two objectives. However, this notion of ordering does not work in higher dimensions. Hence, an alternative is proposed for identifying the most isolated point in a high dimensional ($p > 2$) objective space, using the concept of triangulation from simplical topology and computational geometry. Triangulation of a discrete set of points is a subdivision of the convex hull of the points into simplices (e.g., a set of triangles in two dimensions or tetrahedra in three dimensions). A frequently used and studied point set triangulation (in two dimensions) is the Delaunay triangulation, which has the property that no point in the point set falls in the interior of the circumcircle of any triangle in the triangulation. Once the triangulation is constructed for the current nondominated point set, the average distance of each vertex from its neighbors (given two vertices $v^{(1)}$ and $v^{(2)}$, $v^{(1)}$ is a neighbor of $v^{(2)}$ [and $v^{(2)}$ a neighbor of $v^{(1)}$] if there is an edge between $v^{(1)}$ and $v^{(2)}$ in the given triangulation) is computed, and the vertex with the largest average distance from its neighbors is then considered as the most isolated point for that iteration. This triangulation concept generalizes to p-simplices in p dimensions (a triangle is a 2-simplex, a tetrahedron is a 3-simplex, etc.). The precise mathematical formulation of this process is presented in the next subsection.

3.2. The optimization algorithm

As a preprocessing step, a global search is performed using DIRECT to explore the design space in unsampled regions of a database (if the database is empty, the global exploration step becomes the data generation step). Since DIRECT is an optimization algorithm, the design space exploration is conducted in an intelligent manner by focusing the global search in potentially optimal regions only. For this initialization step, $p + 1$ single objective formulations using $p + 1$ weight vectors w are obtained — p of which correspond to the individual optima of the p objective functions and one more with equal preference given to all the objectives ($w_i = 1/p$, $i = 1, \ldots, p$, $\sum_{i=1}^{p} w_i = 1$). A peculiarity of DIRECT is that it never samples the boundary points of a design space and takes a large number of iterations to reach reasonably close to the boundary. On account of this behavior of DIRECT, the potentially optimal regions returned by DIRECT are fine tuned using MADS. For each of these $p+1$ single objective formulations using the data collected in DIRECT's global search an interpolating surrogate is built over the entire design space and optimized using MADS. These $p + 1$ candidate solutions are evaluated and stored in the database.

The preprocessing step is a crucial part of the proposed algorithm. It helps in eliminating a subset of local Pareto optimal points and accelerates the process of moving to the Pareto front.

However, it may not eliminate all the local Pareto optimal points (as the preprocessing is done using a fixed set of weights), and hence the algorithm may not always find the global Pareto front.

A set $X^{(0)}$ of nondominated points (with respect to all database points) is obtained at the end of the preprocessing step that serves as input for the iterative procedure of the algorithm that follows. Each iteration of the algorithm consists of three steps: (1) finding the most isolated point from the current nondominated point set, (2) constructing surrogates for the objective functions with the isolated point determined in Step 1 as the center, and (3) solving several single objective optimization problems to produce candidate Pareto solutions. At the beginning of the iterative process, a trust region radius Δ_0, the trust region contraction parameter ρ, and the tolerance value τ for the trust region radius are initialized.

It is possible to have two nondominated points $x \neq y$ with $F(x) = F(y)$. In this case only one point is used in the algorithm, and the other point(s) are stored as long as x remains nondominated.

Step 1: Isolated point determination

Let $X^{(k)} = \{x^{(k,1)},\ldots,x^{(k,J_k)}\}$ be the set of nondominated points at the start of the iteration k and $P^{(k)} = \{F(x^{(k,1)}), F(x^{(k,2)}), \ldots, F(x^{(k,J_k)})\}$ be the corresponding objective space set, where $F(x) = (f_1(x), \ldots, f_p(x))$ for p objectives and J_k is the cardinality of $X^{(k)}$ (and $P^{(k)}$). Note that for the very first iteration ($k = 0$) the nondominated set $X^{(0)}$ is the result of the preprocessing step. For a typical p-objective optimization problem the Pareto front is a $(p-1)$-dimensional manifold. Hence to construct a $(p-1)$-dimensional triangulation from a set of p-dimensional points, each point $(f_1, f_2, \ldots, f_p)$ is transformed to homogeneous coordinates $(f_1/f_p, f_2/f_p, \ldots, f_{p-1}/f_p, 1)$ assuming that the objectives are appropriately scaled. (Precisely shift $f_p(x)$ by an amount s so that $f_p(x) + s > 0$ and $f_i(x)/(f_p(x) + s) = O(1)$ for all i.) Essentially the triangulation is done in the $(p-1)$-dimensional real projective space $\mathbb{P}^{p-1}$. A Delaunay triangulation D_{Tr} is constructed using this $(p-1)$-dimensional transformed data set. The gap $\delta_j^{(k)}$ between the nondominated points in the objective space set $P^{(k)}$ is computed for each point $F(x^{(k,j)})$, $j = 1, \ldots, J_k$, in $P^{(k)}$ by,

$$\delta_j^{(k)} = \frac{\sum_{i \in N_j} \left\| F\left(x^{(k,j)}\right) - F\left(x^{(k,i)}\right) \right\|_2}{|N_j|},$$

where $N_j = \{i \mid F(x^{(k,i)})$ is a neighbor of $F(x^{(k,j)})$ in $D_{Tr}\}$. The point $x^{(k,j)}$ (not already accepted as Pareto optimal) with the largest $\delta_j^{(k)}$ value is selected as the most isolated point, and is used as the center point $\tilde{x}^{(k)}$ for a local model with a trust region radius Δ_k. In case of ties, choose the smallest index j. If $N_j = \phi$ ($J_k = 1$), then $\tilde{x}^{(k)}$ is the single point in $X^{(k)}$.

All the most isolated nondominated points $\tilde{x}^{(k)}$ found, together with an associated trust region radius Δ_k, are stored in a database. If $\tilde{x}^{(k)}$ happens to be a previously visited point for $k > 1$, i.e., if

for some $i < k$, $\tilde{x}^{(k)} = \tilde{x}^{(i)}$, then the trust region radius is modified as $\Delta_i := \rho\Delta_i$, and if the new Δ_i value is greater than τ, $\tilde{x}^{(k)}$ is the center point for the local model for the next step; otherwise $\tilde{x}^{(k)}$ is accepted as a Pareto optimal point (and excluded from further consideration as an isolated point) and the above procedure is repeated until a new isolated point is found. If $\tilde{x}^{(k)}$ is not a previously visited point, then $\tilde{x}^{(k)}$ and the trust region radius $\Delta_k := \Delta_0$ for the iteration k are added to the database. If the point $\tilde{x}^{(k)}$ is some previously visited point then the algorithm failed to produce any better points (i.e., a point that dominates the center point) at the ith iteration. This behavior could be seen in either of two situations: the surrogate for the local trust region for the ith iteration was not accurate enough or $\tilde{x}^{(k)}$ is a locally Pareto optimal point. Reducing the trust region radius (and generating a new experimental design) would help in constructing a more accurate surrogate to check if any better points around $\tilde{x}^{(k)}$ exist.

In classic trust region algorithms a trust region is expanded when the surrogate is in reasonably good agreement with the true objective function. In the proposed algorithm the center point, and hence the trust region, at each iteration changes as a result of the process of identifying the most isolated point, and weights are redetermined at each iteration that changes the resulting objective function. As a result, the objective function changes at each iteration, and hence is not plausible that a good surrogate for one iteration is likely good for the next iteration as well. Hence, expansion of the trust region is omitted in the proposed algorithm.

Step 2: Surrogates

DIRECT sampling.

The algorithm uses a linear Shepard method (LSHEP from (Thacker et al., 2010)) to approximate a response surface within a trust region. The isolated point $\tilde{x}^{(k)}$ determined in the previous step serves as the center point for the local trust region with radius Δ_k taken as the intersection of the local trust region box $\left\{x \mid \|x - \tilde{x}^{(k)}\|_\infty \leq \Delta_k\right\}$ with the box $[l, u]$ defined by the variable bounds $l \leq x \leq u$. A tuning parameter ϵ for DIRECT controls the exploration mode for DIRECT (local versus global). The number of points to be sampled can be specified as an input to DIRECT, as for a Latin hypercube, the difference being deterministic adaptive rather than random point placement. The points to be sampled using DIRECT are based on the value of the aggregate objective constructed as a convex combination of the p objective functions. $p+1$ different weight vectors, the same as those used for preprocessing are used here by DIRECT to sample the design space. After these $p+1$ runs of DIRECT, a data set $D^{(k)} = \left\{\left(x, f_1(x), f_2(x), \ldots, f_p(x)\right) \mid x \in X_d^{(k)}\right\}$ of design samples $x \in X_d^{(k)}$ and function values $f_1(x)$, $f_2(x)$ …, $f_p(x)$ is generated.

Surrogate construction.

p LSHEP (Thacker et al., 2010) surrogates $\tilde{f}_1$, $\tilde{f}_2$, ..., $\tilde{f}_p$ are built for the p objective functions using the database $D^{(k)}$. At every iteration, several single objective optimization problems are formulated using convex combinations of these surrogates. p of the weight combinations used have fixed values (e.g., (1, 0, ... ,0)) for each iteration to compute the individual optima of the p objective functions. Additional weights are derived using an adaptive weighting scheme. Based on the value of the objective functions at the center point $\tilde{x}^{(k)}$ for an iteration and its neighbors, $\max\{1, |N_k|\}$ additional weight vectors w indexed by $N_k = \left\{ i \mid F\left(x^{(k,i)}\right) \text{ is a neighbor of } F\left(\tilde{x}^{(k)}\right) \right\}$ are generated as follows.

If $J_k = 1$, $w^{(p+1)} = (w_1^{(p+1)}, w_2^{(p+1)}, \ldots, w_p^{(p+1)}) = (1/p, 1/p, \ldots, 1/p)$. Now consider the case $J_k \geq 2$ ($N_k \neq \phi$). For each $l_i \in N_k = \{l_1, l_2, \ldots, l_{|N_k|}\}$, define a weight vector $w^{(p+i)}$ by

$$w_j^{(p+i)} = \begin{cases} 0, & \text{if } |f_j\left(\tilde{x}^{(k)}\right) - f_j\left(x^{(k,l_i)}\right)| = 0; \\ \dfrac{c_i}{\left|f_j\left(\tilde{x}^{(k)}\right) - f_j\left(x^{(k,l_i)}\right)\right|}, & \text{otherwise,} \end{cases}$$

where $c_i > 0$ is a normalization constant such that $\sum_{j=1}^{p} w_j^{(p+i)} = 1$. Thus, there are $p + \max\{1, |N_k|\}$ weight vectors $w^{(i)}$ used to construct aggregate objectives for MADS in the next step. All these weight vectors $w^{(i)}$ then define surrogates

$$\sum_{j=1}^{p} w_j^{(i)} \tilde{f}_j(x) \approx \sum_{j=1}^{p} w_j^{(i)} f_j(x)$$

for use in Step 3.

Step 3: Solving optimization problems

Each of the surrogates constructed in Step 2 is optimized using MADS to get a candidate solution. Thus at the end of iteration k, $p + \max\{1, |N_k|\}$ candidate Pareto solutions are obtained, evaluated to get their true function values, and then nondominated points among those are selected. Let $X_*^{(k)}$ be the set of these nondominated points. An updated set $X^{(k+1)}$ of nondominated points is obtained at the end of iteration k by comparing all points from the union $X^{(k)} \cup X_d^{(k)} \cup X_*^{(k)}$. Redundant nondominated points ($x \neq y$ for which $F(x) = F(y)$) are removed from $X^{(k+1)}$ and stored separately. Redundant dominated points are deleted.

These three steps are iterated by the algorithm until a stopping condition is satisfied. In practice, termination is either determined by a fixed number of iterations, or when the value of the star discrepancy (discussed in detail in the next section) drops below a predefined threshold.

3.3. Algorithm pseudocode

Algorithm

Input: p (number of objective functions), v (number of design variables), l, u (bounds on objective functions), Δ (trust region radius), ρ (trust region contraction parameter), τ (tolerance for the trust region radius), N_F (total number of true function evaluations), $\bar{N}_F$ (budget for N_F), D_T (database of all true function evaluations), $D^{(k)}$ (database of true function evaluations within kth local trust region), $w^{(i)}$ (weight vectors), $N_k = \{i \mid F(x^{(k,i)}) \text{ is a neighbor of } F(\tilde{x}^{(k)})\}$ in a triangulation D_{Tr}

Output: sets $X^{(k)}$ of nondominated points and $P^{(k)}$ of corresponding objective function values.

for $i := 1$ **step** 1 **until** p **do**

begin

$w^{(i)} :=$ ith standard basis vector;

$w_i^{(p+1)} := 1/p$;

end

for $i := 1$ **step** 1 **until** $p + 1$ **do**

begin

minimize $\sum_{j=1}^{p} w_j^{(i)} f_j(x)$ using DIRECT;

store all points x sampled by DIRECT and

corresponding objective function values $F(x) = \{f_1(x), \ldots, f_p(x)\}$ in D_T;

end

Using LSHEP and D_T, build p surrogates $\tilde{f}_i(x)$, $i = 1, \ldots, p$ for the p objective functions;

for $i := 1$ **step** 1 **until** $p + 1$ **do**

begin

minimize $\sum_{j=1}^{p} w_j^{(i)} \tilde{f}_j(x)$ using MADS;

Evaluate and store the candidate solutions in D_T;

end

$N_F = |D_T|$;

$k := 0$;

Obtain the sets $X^{(k)}$ and $P^{(k)}$ of nondominated points and

corresponding objective function values with respect to D_T;

$J_k := |X^{(k)}|$;

$X^{(k)} = \{x^{(k,1)}, \ldots, x^{(k,J_k)}\}$ and $P^{(k)} = \{F(x^{(k,1)}), \ldots, F(x^{(k,J_k)})\}$;

Remove redundant nondominated points ($x \neq y$ for which $F(x) = F(y)$)

and corresponding objective function values, and

store them separately as $Y^{(k)}$ and $Q^{(k)}$, respectively;

while $N_F < \bar{N}_F$ **do**

begin

Transform each point in $P^{(k)}$ to homogenous coordinates (f_1/f_p, ..., f_{p-1}/f_p, 1);

Construct Delaunay triangulation D_{Tr} for the transformed dataset;

for $j := 1$ **step** 1 **until** J_k **do**

Compute $\delta_j^{(k)}$ using the definition of Step 1;

$L := \{1, \ldots, J_k\}$;

$accept := false$;

while $accept = false$ and $L \neq \phi$ **do**

begin

Find smallest m such that $\delta_m^{(k)} = \max_{j \in L} \delta_j^{(k)}$;

$\tilde{x}^{(k)} := x^{(k,m)}$;

if $k > 0$ **then**

if $\tilde{x}^{(k)} = \tilde{x}^{(i)}$ for some $i < k$ **then**

begin

$\Delta_i := \rho \Delta_i$;

if $\Delta_i > \tau$ **then**

$accept := true$;

else

$L := L \setminus \{m\}$;

end

else

begin

$\Delta_k := \Delta_0$;

Store $\tilde{x}^{(k)}$, $F(\tilde{x}^{(k)})$, and Δ_k;

$accept := true$

end

else

begin

Store $\tilde{x}^{(0)}$, $F(\tilde{x}^{(0)})$, and Δ_0;

$accept := true$

end

end $\tilde{x}^{(k)}$ acceptance loop

if $L = \phi$ **then**

begin

$X^{(k)} := X^{(k)} \cup Y^{(k)}$;

$P^{(k)} := P^{(k)} \cup Q^{(k)}$;

Return $X^{(k)}$ and $P^{(k)}$;

end

Build a local trust region $LTR = \{x \mid \|x - \tilde{x}^{(k)}\|_\infty \leq \Delta_k\}$;

for $i := 1$ **step** 1 **until** p **do**

begin

minimize $\sum_{j=1}^{p} w_j^{(i)} f_j(x)$ using DIRECT;

Store the point set $X_d^{(k)}$ sampled within LTR by DIRECT

and corresponding objective function evaluations in D_T;

Let $D^{(k)} = \{(x, f_1(x), \ldots, f_p(x)) | x \in X_d^{(k)}\}$;

end

Using $D^{(k)}$ and LSHEP, build p surrogates $\tilde{f}_j(x)$ for p objectives;

Formulate p single objectives $s_i(x) := \sum_{j=1}^{p} w_j^{(i)} \tilde{f}_j(x)$ for $i = 1, \ldots, p$;

if $J_k = 1$ **then**

$s_{p+1}(x) := \sum_{j=1}^{p} w_j^{(p+1)} \tilde{f}_j(x)$;

else

for $i := 1$ **step** 1 **until** $|N_k|$ **do**

begin

Compute $w_j^{(p+i)}$ using the definition in Step 2;

$s_{p+i}(x) := \sum_{j=1}^{p} w_j^{(p+i)} \tilde{f}_j(x)$;

end

for $i := 1$ **step** 1 **until** $p + \max\{1, |N_k|\}$ **do**

begin

Minimize s_i within the LTR using MADS;

Evaluate and store the candidate solutions in D_T;

end

$X_*^{(k)}$:= set of nondominated points among the $p + \max\{1, |N_k|\}$ solutions;

$X^{(k+1)}$:= nondominated points in $X^{(k)} \cup X_d^{(k)} \cup X_*^{(k)}$;

Remove redundant nondominated points ($x \neq y$ for which $F(x) = F(y)$) and

corresponding objective function values from

$X^{(k+1)}$ and $P^{(k+1)}$ and add them to $Y^{(k)}$ and $Q^{(k)}$ giving $Y^{(k+1)}$ and $Q^{(k+1)}$, respectively;

Remove dominated points from $Y^{(k+1)}$ and $Q^{(k+1)}$;

$N_F := |D_T|$;

$k := k + 1$;

$J_k := |X^{(k)}|$;

end $\bar{N}_F$ while loop

Return ($X^{(k)}$, $P^{(k)}$, $Y^{(k)}$, and $Q^{(k)}$);

4. Numerical Evaluation

The performance of the proposed method is evaluated using test problems from (Deb et al., 2001), which are formulated in order to evaluate the capability of a multiobjective optimization algorithm to handle some of the inherent difficulties in MOPs. The main features of the test suite from (Deb et al., 2001) are the simplicity of problem construction, scalability to any number of design variables and objective functions, a priori knowledge of the Pareto front, and variety of complexities in the Pareto fronts. The test suite tests the ability of an algorithm to accommodate complexity and variety in Pareto front landscapes, converge to the Pareto front, and maintain a good spread of the Pareto optimal solutions. The selection of test problems in this paper is intended to illustrate the capability of the proposed method to handle diverse landscapes of Pareto fronts, discontinuity in the objective functions, and scalability to higher dimensional design and objective spaces.

4.1. Performance Measures

Obtaining good distribution of the nondominated set is a nontrivial aspect of a multiobjective optimization problem in general, and hence dispersion of the nondominated solution set is a good performance measure for comparing different algorithms. There is no known measure in the literature that better assesses the spread of the nondominated set. Most approaches in the literature are based on some statistical measure such as Euclidean distance, standard deviation, or skewness, which may not be sufficient for assessing the distribution of a point set in higher dimensions. Hence, a precise mathematical measure, star discrepancy, was proposed by the authors in (Deshpande et al., 2013) for a biobjective case. A new technique is proposed in this paper for extending the star discrepancy based measure to problems with more than two objectives.

General discrepancy (Kugele et al., 2007) of a sequence $S_N = \{s_i^N\}$, where $i = 1, \ldots, N$ and $s_i \in [0,1]^d$ is defined as

$$D(\beta; S_N) = \sup_{B \in \beta} \left| \frac{A(B; S_N)}{N} - \lambda_d(B) \right|,$$

where β denotes a family of Lebesgue measurable subsets of $[0,1]^d$, $A(B;S_N)$ is the number of points from S_N that are also elements of B, and $\lambda_d(B)$ is the d-dimensional Lebesgue measure of B. For practical purposes, a variation of general discrepancy, star discrepancy (Caflisch, 1998), (Kugele et al., 2007), is often used. Let J^* denote the family of Lebesgue measurable subsets of $[0,1]^d$ of the form $\prod_{i=1}^{d}[0, v_i)$. Then the star discrepancy of the sequence S_N is defined as

$$D_N^*(S_N) = D(J^*; S_N).$$

A family of sequences $\{S_N\}_{N=1}^{\infty}$ is said to be uniformly distributed if and only if

$$\lim_{N \to \infty} D_N^*(S_N) = 0.$$

Typically, Pareto fronts in p-objective optimization problems are $(p-1)$-dimensional manifolds, e.g., for a typical triobjective optimization problem the Pareto front would be a two-dimensional surface. The volume of a Pareto front is then approximated by adding up volumes of all the simplices in the corresponding triangulation (e.g., the volume of a triobjective optimization Pareto front is its area, which is approximated by adding the areas of all the triangles in the corresponding triangulation), and then normalized to unit volume. The star discrepancy is computed by approximating J^* with unions of adjacent simplices in the triangulation of the Pareto front.

Another performance measure used to evaluate the proposed algorithm is representativeness (number of nondominated points obtained) N_{ND}. Efficiency in terms of the total number of true function evaluations N_{FE} is another good measure for assessing the performance of a multiobjective optimization method. However, the total number of true function evaluations is set as a budget (and hence a stopping criterion) for all the test problems presented in this paper, and hence is set to a predefined number. The percentage of nondominated points with respect to the total number of true function evaluations N_{ND}/N_{FE} indicates the ability of a method to eliminate dominated points and focus its effort on obtaining more nondominated points.

The Pareto optimal fronts for the five test problems along with all three performance measures (D_N^*, N_{ND}, and N_{ND}/N_{FE}) along with the number N_{FE} of true function evaluations are presented in Section 4.3.

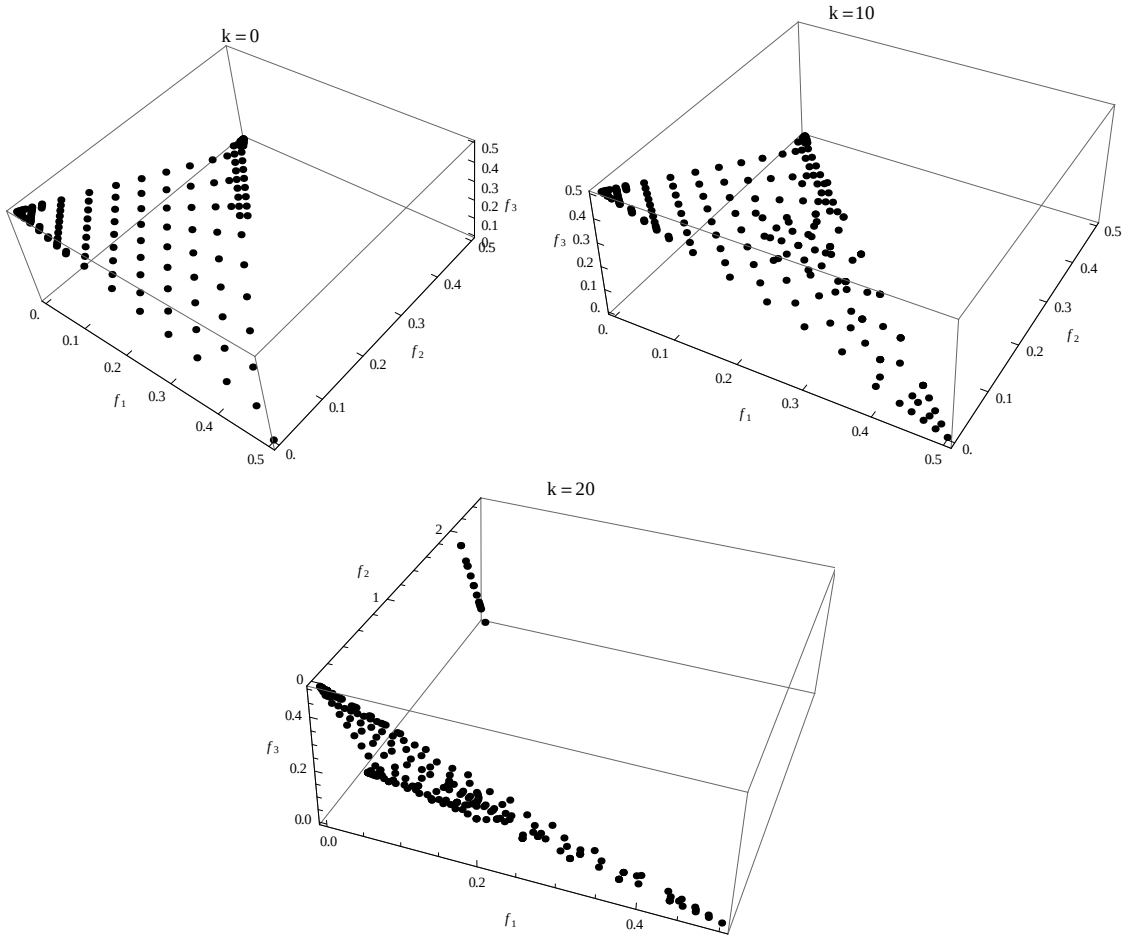

Figure 1. Problem 4.3.1 with a planar Pareto front: Pareto fronts after preprocessing, 10 iterations, and 20 iterations of the algorithm (clockwise).

4.2. Parameter Setup

All the test problems presented in Section 4.3 have v design variables x_i with bound constraints $0 \leq l_i \leq x_i \leq u_i \leq 1$, and p conflicting objectives f_1, f_2, ..., f_p to minimize. The numerical evaluation uses a Fortran 95 implementation VTDIRECT95 (He et al., 2002) of the algorithm DIRECT and a C++ implementation NOMAD (version 3.5.1) of the algorithm MADS (Audet and Dennis 2006).

The initial parameter setup for VTDIRECT95 for the preprocessing step is (using VTDIRECT95 variable names) design variables lower bound, $L := [l_1, \ldots, l_v]$, design variables upper bound, $U := [u_1, \ldots, u_v]$, upper limit on the number of true function evaluations, MAX_EVAL := 2000, and $EPS := 0.001$, where EPS is a tuning parameter that controls the exploration mode of DIRECT (local versus global). VTDIRECT95 is run $p + 1$ times with this setup for $p + 1$ single objective optimization problems of the form $\sum_{i=1}^{p} w_i f_i(x)$, where the vector w is set to optimize individual objectives for the first p runs and to $(1/p, \ldots, 1/p)$ for the last run. If a point already

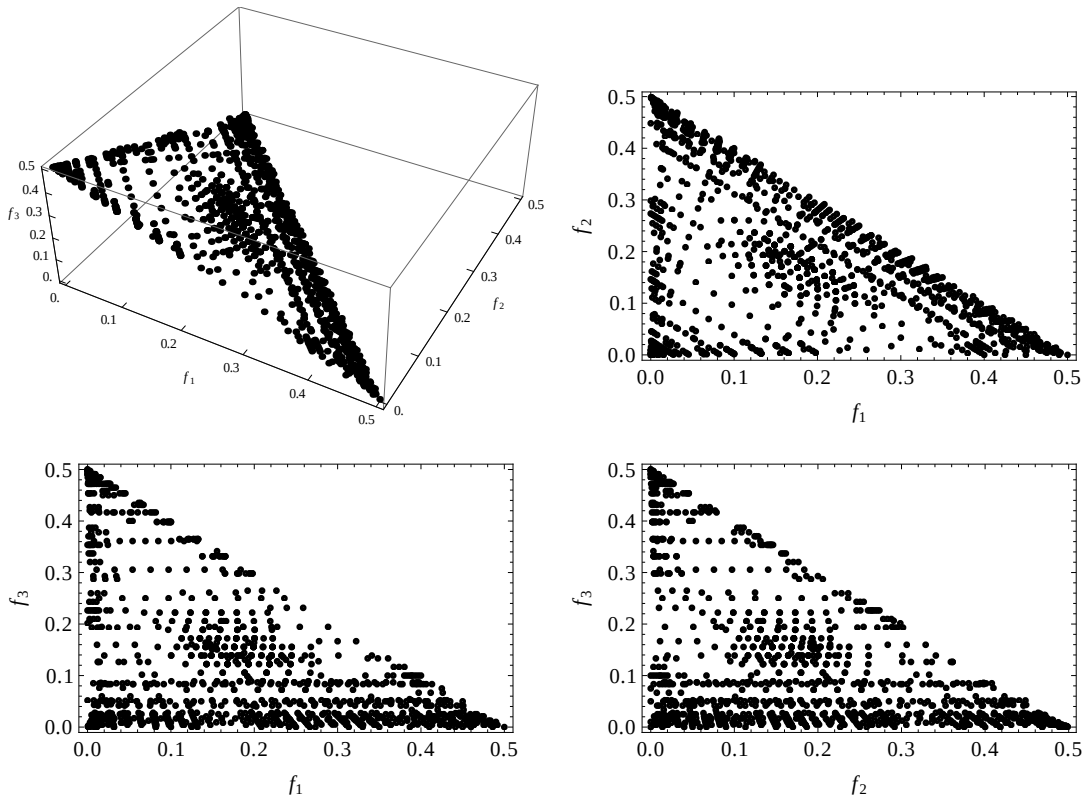

Figure 2. Problem 4.3.1: Planar Pareto front using proposed method with 30000 function evaluations.

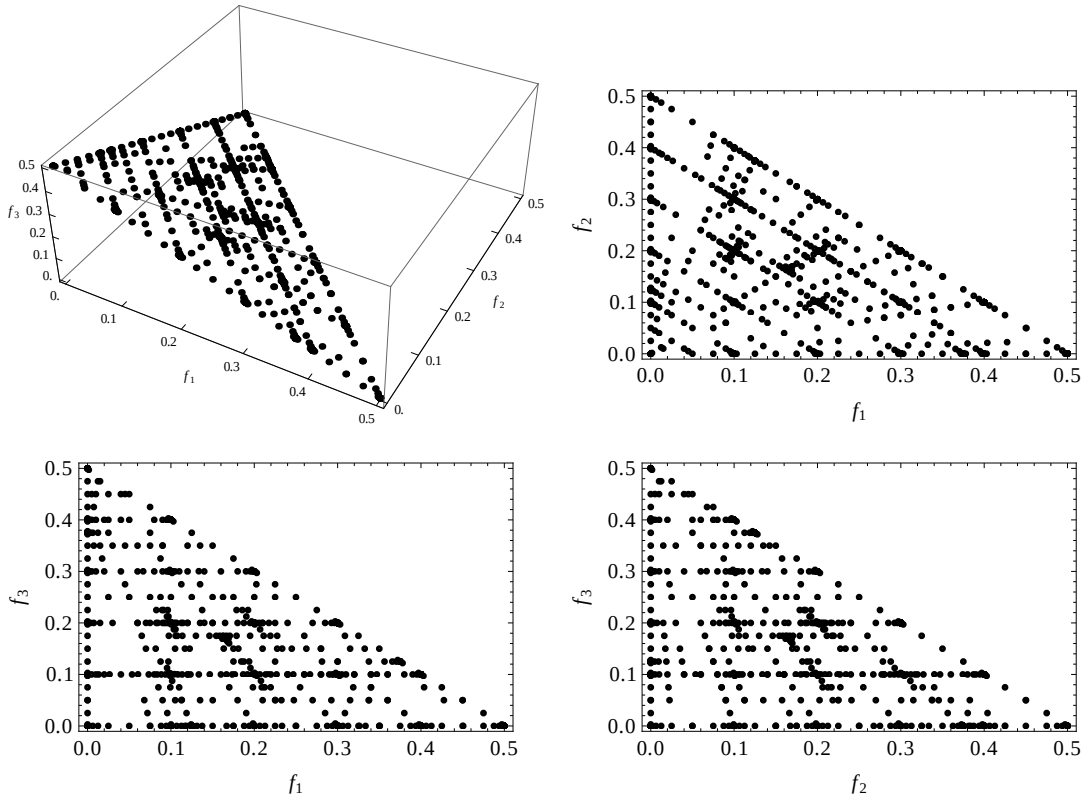

Figure 3. Problem 4.3.1: Planar Pareto front using MultiMADS with 30000 function evaluations.

exists in the database within a tolerable distance (1E−13) from the current sampled point, then the objective function values in the database are used instead of repeating the system analysis at the sampled point. For the preprocessing step, the starting point $x^{(0)}$ for MADS is set to the center point of the entire design space, bound constraints are set to the design space boundaries, and the maximum number of surrogate evaluations is set to 500.

The parameter settings for MADS and VTDIRECT95 for the main iterative procedure are: MADS: the starting point $x^{(0)}$ is set to the center point determined for that iteration, the bound constraints for the MADS run are set to the local trust region boundaries, and the maximum number of surrogate evaluations to 50. VTDIRECT95: L and U are set to the local trust region boundaries for that iteration, MAX_EVAL := 100, and $EPS := 0.1$. The single objective formulation is the same as the one used in the preprocessing step. The local trust region parameters are set as $\Delta := 0.2$, $\rho := 0.5$, and $\tau := 0.02$.

4.3. Test Problems and Results

The proposed method is evaluated by five bound constrained problems from (Deb et al., 2001). Diversity in the test problems lies in the landscape (a hyperplane for the first problem and a sphere for the second and fourth problems) of the Pareto front, scalability (the fourth test problem is a replica of the second problem with thrice the number of design variables and the last problem has $p = 6$) to higher dimensional design and objective spaces, and discontinuity in the objective functions (the third test problem has a Pareto front with four connected components). For all the test problems, let $x = \big(x_1, \ldots, x_p, \ldots, x_v\big)$. All the test problems except for the last one have three objective functions ($p = 3$); the last problem has six objective functions ($p = 6$). The number v of design variables is set to 12 for the problems 4.3.1, 4.3.2, and 4.3.5, to 20 for the problem 4.3.3, and to 36 for the problem 4.3.4, , all described in subsequent subsections. A budget of thirty thousand true function evaluations (this was suggested by Deb et al. (2001) to solve the test problems using evolutionary algorithms) is set as a stopping criterion for all the experiments.

For the first four test problems a figure with four plots with four different views in (f_1, f_2, f_3), (f_1, f_2), (f_1, f_3), and (f_2, f_3) spaces is presented. In addition, for the test problem 4.3.1 a figure with three additional plots is presented for illustrating progress of the proposed algorithm at different points in time (preprocessing, after ten iterations of the algorithm, and after twenty iterations of the algorithm). No plots are presented for the problem 4.3.5 owing to the difficulty of visualizing higher dimensional objective space. The three performance measures along with the number N_{FE} of true function evaluations for all the test problems are presented in Table 1.

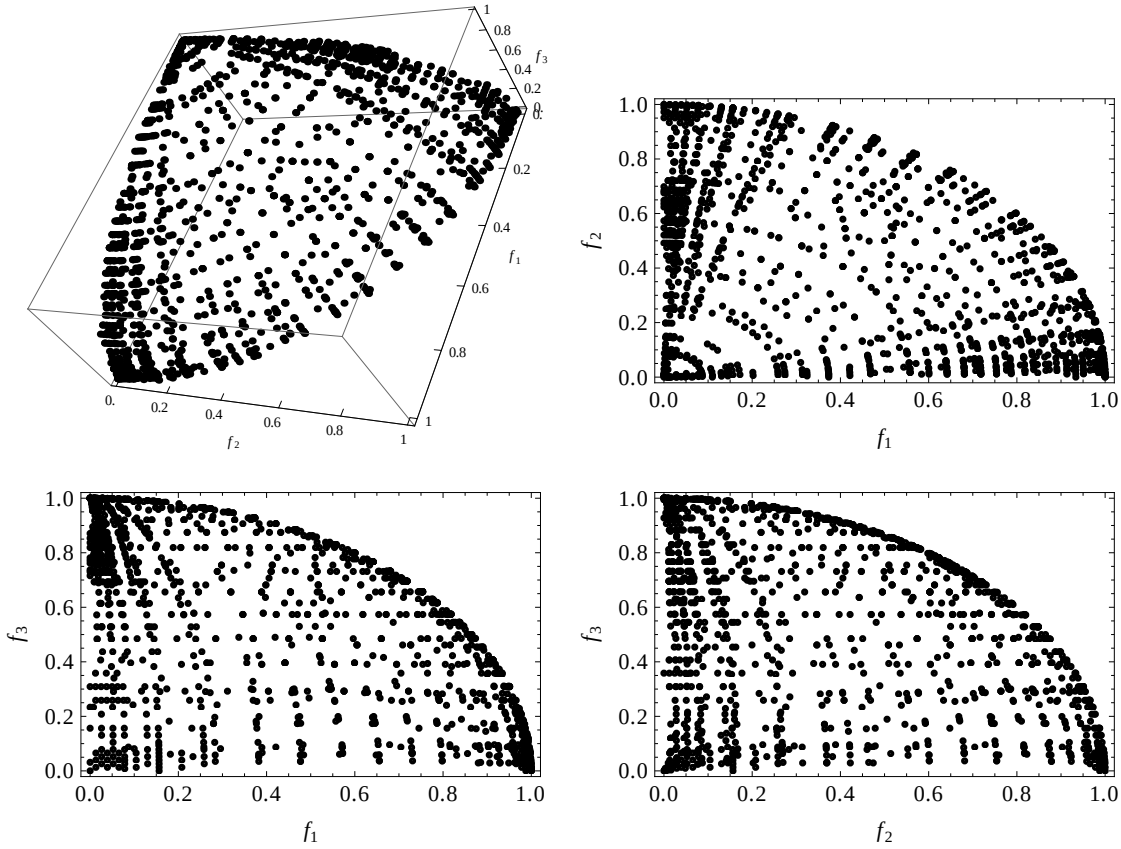

Figure 4. Problem 4.3.2: Spherical Pareto front using proposed method with 30000 function evaluations.

4.3.1. A problem with a planar Pareto front

The problem with the planar Pareto front ((Deb et al., 2001), Problem $DTLZ1$) is

$$\text{minimize} \begin{cases} f_1(x) = 1/2 x_1 x_2 \dots x_{p-1}\big(1 + g(x_p, \dots, x_v)\big), \\ f_2(x) = 1/2 x_1 x_2 \dots (1 - x_{p-1})\big(1 + g(x_p, \dots, x_v)\big), \\ \vdots \\ f_{p-1}(x) = 1/2 x_1 (1 - x_2)\big(1 + g(x_p, \dots, x_v)\big), \\ f_p(x) = 1/2 (1 - x_1)\big(1 + g(x_p, \dots, x_v)\big), \end{cases}$$

$$g(x_p, \dots, x_v) = 100[(v - p + 1) + \sum_{i=p}^{v} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))],$$

subject to $0 \le x_i \le 1$, for $i = 1, \dots, 12$. The set of Pareto optimal solutions is $\{x \in \mathbb{R}^{12} : 0 \le x_1 \le 1, 0 \le x_2 \le 1, x_i = 0.5 \ \forall \ i = 3, 4, \dots, 12\}$. The Pareto front is the simplex $\{f_i \ge 0 : \sum_{i=1}^{3} f_i = 0.5\}$. The difficulty in this problem, introduced by the multimodal function g, is to converge to the Pareto front. The search space of the problem contains $11^{10} - 1$ local Pareto fronts. The results are presented in Figures 1 and 2, and Table 1.

Results show that the proposed method successfully produces well distributed points near the global Pareto front. The plots in the objective space (Figure 1) and the corresponding performance

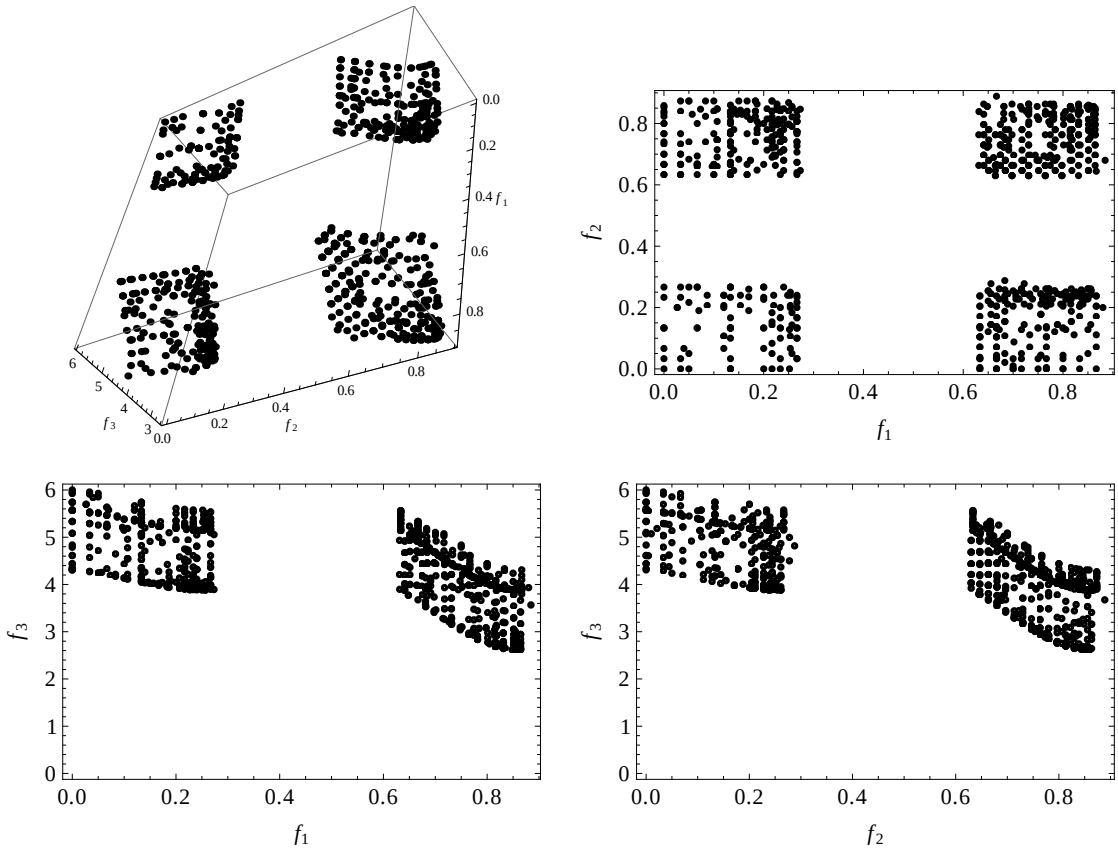

Figure 5. Problem 4.3.3: Discontinuous Pareto front using proposed method with 30000 function evaluations.

measures (first three rows in Table 1) at different times (after preprocessing, ten iterations, and twenty iterations of the algorithm) during the course of the algorithm show that the proposed method gradually moves towards the global front by eliminating local Pareto optimal solutions (see the two local Pareto fronts of the third plot in Figure 1). The value for the star discrepancy keeps dropping during the course of the algorithm, which is an indication that the algorithm fills in the gaps around the most isolated point at each iteration as intended. A sudden rise in the star discrepancy value for the third plot (and the third row of Table 1) in Figure 1 is because of the huge gap in the objective space created due to two local fronts. Figure 2 and the fourth row in Table 1 present the progress of the algorithm after exhausting the budget of thirty thousand true function evaluations.

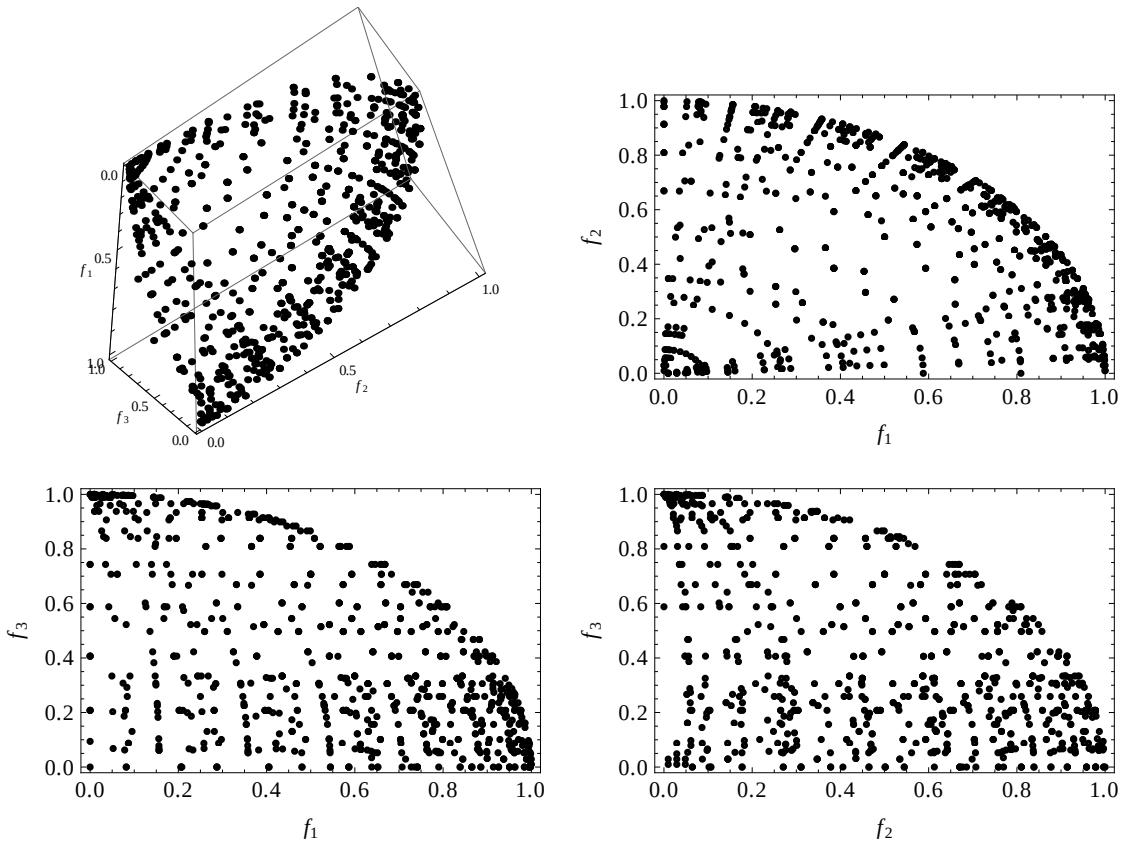

Figure 6. Problem 4.3.4: Pareto front in higher dimensional design space using proposed method with 30000 function evaluations.

4.3.2. A problem with a Pareto front on a sphere

The problem with the spherical Pareto front ((Deb et al., 2001), Problem $DTLZ2$) is

$$\text{minimize} \begin{cases} f_1(x) = \cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{p-2}\pi/2)\cos(x_{p-1}\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \\ f_2(x) = \cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{p-2}\pi/2)\sin(x_{p-1}\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \\ \vdots \\ f_{p-1}(x) = \cos(x_1\pi/2)\sin(x_2\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \\ f_p(x) = \sin(x_1\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \end{cases}$$

$$\text{where } g(x_p,\ldots,x_v) = 100[(v-p+1) + \sum_{i=p}^{v}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))],$$

subject to $0 \le x_i \le 1$, for $i = 1, \ldots, 12$. The set of Pareto optimal solutions is $\{x \in \mathbb{R}^{12} : 0 \le x_1 \le 1$, $0 \le x_2 \le 1$, $x_i = 0.5 \ \forall \ i = 3\ ,4,\ \ldots,\ 12\}$. The Pareto front is the first octant of the unit sphere $\sum_{i=1}^{3} f_i^2 = 1$. The difficulty in this problem, introduced by the multimodal function g, is again to converge to the Pareto. The search space of the problem contains $3^{10}-1$ local Pareto fronts. Results are presented in Figure 4 and Table 1 (row 5), and show that the algorithm achieves a very good approximation to the global Pareto front maintaining good spread of the nondominated points.

4.3.3. A problem with a discontinuous Pareto front

The problem with a discontinuous Pareto front ((Deb et al., 2001), Problem $DTLZ7$) is

$$\text{minimize} \begin{cases} f_1(x) = x_1, \\ f_2(x) = x_2, \\ \vdots \\ f_{p-1}(x) = x_{p-1}, \\ f_p(x) = \big(1 + g(x_p, \ldots, x_v)\big) h(f_1, f_2, \ldots, f_{p-1}, g(x_p, \ldots, x_v)), \end{cases}$$

$$g(x_p, \ldots, x_v) = 1 + \frac{9}{v - p + 1} \sum_{x_i \in X_P} x_i,$$

$$h(f_1, f_2, \ldots, f_{p-1}, g(x_p, \ldots, x_v)) = p - \sum_{i=1}^{p-1} \Big[\frac{f_i}{1 + g(x_p, \ldots, x_v)} (1 + \sin(3\pi f_i)) \Big],$$

where $X_P = \{x_p, \ldots, x_v\}$, $0 \leq x_i \leq 1$, for $i = 1, \ldots, 20$ and $p = 3$. This test problem has $2^{p-1} = 4$ connected components. The set of Pareto optimal solutions is $\forall x_i \in X_P$, $x_i = 0$. The difficulty in this test problem is to identify the connected regions in the objective space. Results obtained by applying the proposed method are illustrated in Figure 5 with four different plot views of the objective space, and Table 1 (row 6) presents the performance measures for this test problem after thirty thousand true function evaluations. Results show that the method successfully identifies the four connected components by maintaining a good spread of nondominated points in each region. Computing the star discrepancy for a discontinuous Pareto front is rather difficult and has not been considered further here.

4.3.4. A problem with a high dimensional search space

This problem is a replica of the test problem 4.3.2 with the only difference being the number v of design variables set to 36. The intention was to test the ability of the proposed method to handle a higher dimensional design space. Results (Figure 6 and Table 1 (row 7)) show that the method successfully handles a higher dimensional design space and the results are comparable to those obtained with a lower dimensional design space for test problem 4.3.2.

4.3.5. A problem with a 6-dimensional objective space

This problem ((Deb et al., 2001), Problem $DTLZ3$) uses the objective function definitions of the problem 4.3.2 except for a different function g and the number of objective functions $p = 6$. Here the function g is

$$g(x_p, \ldots, x_v) = \sum_{x_i \in X_P} (x_i - 0.5)^2,$$

where $X_P = \{x_p, \ldots, x_v\}$, subject to $0 \leq x_i \leq 1$, for $i = 1, \ldots, 12$. The performance measures are presented in Table 1 (last row). Results show that the method successfully produces a good collection (N_{ND}) of evenly spread (D^*) Pareto optimal solutions for a higher dimensional objective space.

4.4. Comparison with Other Algorithms

As mentioned in the introduction, the proposed method falls under the classical scalarization approaches for solving multiobjective optimization problems. Hence it is logical to compare the proposed method with other deterministic multiobjective optimization methods rather than with the evolutionary approaches. The method is compared to state-of-the-art techniques in the class of deterministic approaches for solving MOPs, BiMADS ((Audet et al.,, 2008)) for the biobjective case and MultiMADS ((Audet et al., 2010)) for the multiobjective (more than two objectives) case. Results for three test problems (a biobjective problem from 4.4.1 using BiMADS, the triobjective problem from 4.1.1 and the six-objective problem from 4.1.5 using MultiMADS) are presented and compared to those obtained using the proposed method.

4.4.1. Comparison with BiMADS on a biobjective optimization problem

The biobjective optimization problem, an engineering design problem from (Aurora, 2004), with two objectives f_1 (fundamental vibration frequency) and f_2 (weight), and two design variables x_1 (height) and x_2 (base) is

$$f_1(x) = x_1 x_2,$$

$$f_2(x) = -\left(\frac{1/\left(\frac{1}{K_s} + \frac{L^3}{3Ex_1x_2^3/12}\right)}{W/g} \right)^{1/2},$$

where $x = (x_1, x_2)$, $0.5 \leq x_1 \leq 1$, $0.2 \leq x_2 \leq 2$, and $K_s = 10$, $L = 12$, $E = 3.00E + 07$, $W = 50$, and $g = 386.4$ are the model constants. The C++ implementation from the package NOMAD 3.5.1 (Digabel and Tribes, 2012) for BiMADS is used with the default settings for all the parameters, and the optimization algorithm presented in (Deshpande et al., 2013) is used for handling the biobjective case of the proposed method with the same settings of the tuning parameters that were used in (Deshpande et al., 2013). A budget of two hundred true function evaluations is set for both the methods. Figure 7 shows two plots in the objective space (f_1, f_2) for the two methods, the proposed method and BiMADS, respectively. The performance measures for the two methods presented in Tables 1 and 2 (last rows) show that the nondominated point set distribution for the proposed method compares favorably to that for BiMADS, although the cardinality of the nondominated

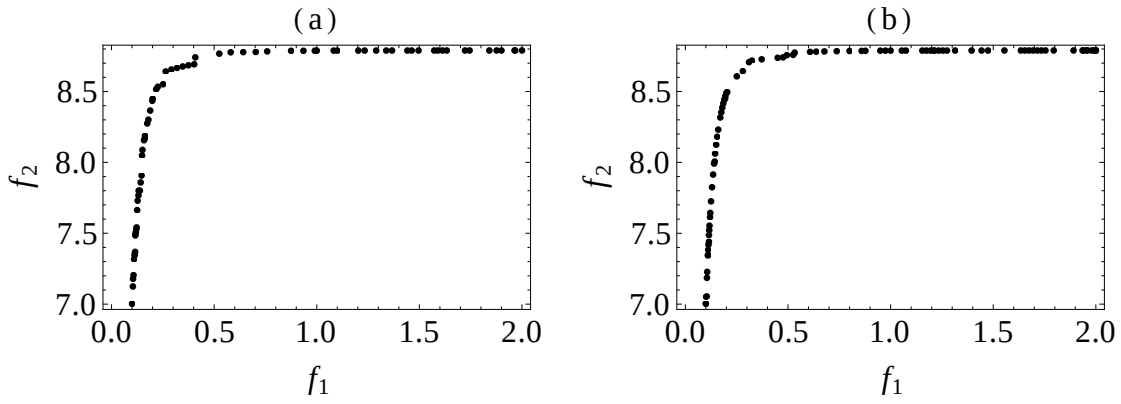

Figure 7. (a) the proposed method and (b) BiMADS with a budget of 200 (maximum) evaluations each for the biobjective optimization problem in 4.4.1.

set produced by BiMADS is better than that using the proposed method. Readers are referred to (Deshpande et al., 2013) and (Deshpande et al., 2013a) for a detailed comparison of the biobjective version of the algorithm with BiMADS on diverse test problems.

4.4.2. Comparison with MultiMADS on multiobjective optimization problems

Results for two of the five test problems, 4.3.1 and 4.3.5, presented in the previous section are used for the comparison with the results obtained using the multiobjective optimization algorithm MultiMADS (Audet et al., 2010). All the default settings in the C++ implementation NOMAD 3.5.1 of the algorithm MADS (Audet and Dennis 2006) are used. As discussed in (Audet et al., 2010) a fifty point Latin hypercube experimental design is used in the initialization of the algorithm MultiMADS to diversify the starting point selection. A budget of thirty thousand true function evaluations is set for both problems with a budget of one thousand evaluations per MADS run except for the construction of the tangent hull where the budget for a MADS run is set to two thousand evaluations. For the triobjective optimization problem in 4.3.1, MultiMADS calls MADS 29 times, including four calls during the initialization step: three times to compute the shadow minimum and once to construct the tangent hull, and 25 single objective optimizations in the main iterative procedure; for the six objective optimization problem in 4.3.5, MADS is called 29 times again, including seven calls during the initialization step: six times to compute the shadow minimum and once to construct the tangent hull, and 22 calls to solve 22 different single objective formulations. Both problems are solved using the Euclidean norm based distance formulation for the single objective optimization. Visualizations for the triobjective case are presented in Figure 3. Table 2 shows the performance measures for both problems.

Results are again very favorable for the proposed method in the star discrepancy measure. MultiMADS (and BiMADS for the biobjective case) in general produced a very good collection of nondominated points but the distribution of the point set compared to that using the proposed method is not so good. This is an indication of either larger gaps or clusters of nondominated points in the objective space.

4.5. Discussion

Figures 1–2 and 4–6 and Table 1 show that the algorithm successfully finds a fairly good approximation to the global Pareto front with well distributed (D_N^* values in Table 1) Pareto optimal points for diverse types of multiobjective optimization problems. The new adaptive weighting scheme and the guided sampling using DIRECT effectively approximate the Pareto front by gradually moving towards the global front by filling the gaps in the incumbent nondominated point set. In that sense the method gradually learns the shape of the true Pareto front and concentrates its computational effort in potentially optimal regions. Both the quality and the quantity of the obtained nondominated point set improves with the number of iterations of the algorithm (for example, the plots in Figure 1 and first three rows in Table 1). The division of the global (preprocessing) and the local search in the course of the algorithm plays an important role in directing the results towards global Pareto fronts. Although all the local Pareto optimal solutions were not eliminated in the preprocessing step for the problem 4.3.1 (see the third plot of Figure 1), those were eliminated eventually during the course of the algorithm (see Figure 2). However, this might not always be the case, and the algorithm might find only a local Pareto front. The star discrepancy based measure for assessing the distribution of the obtained nondominated set seems to be working reasonably well, and can be used as a stopping criterion.

The method produced reasonably good results for a multiobjective optimization problem (4.3.5) with six objective functions. Delaunay triangulation used in this work has a well known (Amenta et al., 2007) worst case bound of $O(n^{\lceil d/2 \rceil})$ for a set of n randomly distributed discrete points in d-dimensional space. Testing the proposed method on problems with a large number of objective functions might divulge useful information about the ability of the algorithm to handle such problems. The comparative study from Section 4.4 reveals that the adaptive weighting scheme based on the triangulation technique is very effective in producing well-distributed fronts as compared to other state-of-the-art methods for solving MOPs.

All the test problems presented in this paper have known analytical forms for the objective functions, which makes it easy to assess the results obtained using the proposed method, and

Table 1

Performance measures: proposed method

	D_N^*	N_{ND}	N_{FE}	N_{ND}/N_{FE}
Problem 1 (preprocessing)	0.3202	165	4218	0.0391
Problem 1 (10 iterations)	0.1814	225	5751	0.0391
Problem 1 (20 iterations)	0.5508	258	7261	0.0355
Problem 1 (all)	0.0690	1229	30000	0.041
Problem 2	0.1050	1541	30000	0.0514
Problem 3	—	800	30000	0.0266
Problem 4	0.0403	920	30000	0.0307
Problem 5	0.1198	2409	30000	0.0803
BiObjective	0.0410	71	200	0.35

Table 2

Performance measures: BiMADS and MultiMADS

	D_N^*	N_{ND}	N_{FE}	N_{ND}/N_{FE}
Problem 1 (all)	0.1594	1223	30000	0.0408
Problem 5	0.2335	4938	30000	0.1646
BiObjective	0.1187	89	200	0.46

hence provides a way of evaluating the method. This is not the only class of problems that the proposed method targets; as mentioned in the introduction, the method intends to solve blackbox multiobjective optimization problems with possibly nonsmooth functions where the analytical form of the objectives is not known. The authors intend to test the proposed method on real engineering applications where such blackbox simulation based multiobjective optimization is prevalent.

5. Conclusion

A new method is proposed for multiobjective optimization of possibly nonsmooth functions targeting the class of blackbox simulation based optimization problems. The algorithm proposes a new adaptive weighting scheme using the concept of triangulation from simplical topology and computational geometry. Cheap(er) surrogates are employed for the expensive objective functions, and an optimization based guided sampling is used as an experimental design for each local trust region optimization. The method combines two direct search algorithms DIRECT (to explore the design space globally) and MADS (to fine tune the potentially optimal regions returned by DIRECT), to balance between global and local searches and to improve the convergence rate. The method is evaluated using five test problems from the literature with different Pareto front landscapes. Results show that the proposed method achieves a reasonably good approximation to the global Pareto front with a good spread of the nondominated points for all the test problems. Star discrepancy is introduced as a new quantitative measure of point distribution. The method compares very favorably to other deterministic multiobjective optimization methods in those quantitative measures.

In the future, the authors plan to apply the proposed method to real engineering problems where the analytical form of the objectives is not known and the objective function evaluation is very expensive. Another extension of the method would be to consider problems with constraints.

References

[1] Arora J (2004), *Introduction to optimum design*, Academic Press, 429

[2] Audet C, Dennis JE (2006), "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, 17(1):188–217

[3] Amenta N, Attali D, Devillers O (2007), "Complexity of Delaunay triangulation for points on lower-dimensional polyhedra," in *Proc. 18TH annual ACM-SIAM SYMPOS. on Discrete Algorithms*, 1106–1113

[4] Audet C, Savard G, Zghal W (2010), "A mesh adaptive direct search algorithm for multiobjective optimization," *European Journal of Operational Research*, 204:545–556

[5] Audet C, Savard G, Zghal W (2008), "Multiobjective optimization through a series of single objective formulations," *SIAM Journal of Optimization*, 19:188–210

[6] Caflisch RE (1998), "Monte Carlo and quasi-Monte Carlo methods," *Acta Numerica*, 7:1–49

[7] Deb K (1999), "Multiobjective genetic algorithms: problem difficulties and construction of test problems," *Evol. Comput.*, 7:205–230

[8] Das I, Dennis JE (1998), "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," *SIAM Journal on Optimization*, 8:631–657

[9] Deb K, Pratap A, Agarwal S, Meyarivan T (2002), "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, 6(2):181–197

[10] Digabel SL, Tribes C (2012), *NOMAD User Guide*, version 3.5.1

[11] Deb K, Thiele L, Laumanns, Zitzler E (2001), "Scalable Test Problems for Evolutionary Multiobjective Optimization," TIK–Technical Report No. 112 Institut fur Technische Informatik und Kommunikationsnetze, ETH Zurich, Switzerland

[12] Deshpande SG, Watson LT, Canfield RA (2013), "Biobjective Optimization using Direct Search Techniques," in *IEEE SoutheastCon*, Jacksonville, FL, Apr 4-7, CD-ROM, 6p

[13] Deshpande SG, Watson LT, Canfield RA (2013a), "Pareto Front Approximation using a Hybrid Approach," in *Procedia Computer Science*, 18:521–530

[14] Eichfelder G (2008), *Adaptive Scalarization Methods in Multiobjective Optimization (Vector Optimization)*, Springer

[15] Gray GA, Fowler KR (2011), "Traditional and hybrid derivative-free optimization approaches for black box functions," *Computational Optimization, Methods and Algorithms*, 356:125–151

[16] He J, Watson LT, Ramakrishnan N, Shaffer CA, Verstak A, Jiang J, Bae K, Tranter WH (2002), "Dynamic data structures for a direct search algorithm," *Computational Optimization and Applications*, 23:5–25

[17] Jones DR, Perttunen CD, Stuckman BE (1993), "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Application*, 79(1):157–181

[18] Kugele SC, Watson LT, Trosset MW (2007), "Multidimensional numerical integration for robust design optimization," in *Proc. ACM Southeast Regional Conference*, 385–390

[19] Ryu J, Kim S, Wan H (2009), "Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization," in *Proceedings of the 2009 Winter Simulation Conference, Austin, TX*, 623–633

[20] Thacker WI, Zhang J, Watson LT, Birch JB, Iyer MA, Barry MW (2010), "Algorithm 905: SHEPPACK: modified Shepard algorithm for interpolation of scattered multivariate data," *ACM Trans. Math. Software*, 37(34):1–20

[21] Veldhuizen DA, Lamont GB (1998), *Evolutionary Computation and Convergence to a Pareto Front*, Stanford University Bookstore

[22] Wang L, Ishida H, Hiroyashu T, Miki M (2008), "Examination of multiobjective optimization method for global search using DIRECT and GA," in *IEEE World Congress on Computational Intelligence*, 2446–2451