Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

PARETO FRONT APPROXIMATION WITH ADAPTIVE WEIGHTED SUM METHOD IN MULTIOBJECTIVE SIMULATION OPTIMIZATION

Jong-hyun Ryu

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA

Sujin Kim

Department of Industrial & Systems Engineering
National University of Singapore
117576, Singapore

Hong Wan

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA

ABSTRACT

This work proposes a new method for approximating the Pareto front of a multi-objective simulation optimization problem (MOP) where the explicit forms of the objective functions are not available. The method iteratively approximates each objective function using a metamodeling scheme and employs a weighted sum method to convert the MOP into a set of single objective optimization problems. The weight on each single objective function is adaptively determined by accessing newly introduced points at the current iteration and the non-dominated points so far. A trust region algorithm is applied to the single objective problems to search for the points on the Pareto front. The numerical results show that the proposed algorithm efficiently generates evenly distributed points for various types of Pareto fronts.

1 INTRODUCTION

Many real-world problems involve decisions based on multiple and conflicting criteria. The optimal decisions need to be taken by identifying the best trade-offs among these criteria, which is the goal of multiobjective optimization. Multiobjective optimization problems (MOP) can be found in many situations, for example, in product design where several criteria must be simultaneously satisfied (Wilson et al. 2001; Tappeta et al. 2002; Shan and Wang 2005), or in financial portfolio management where it is necessary to find trade-offs between the return of the portfolio and its volatility risk (Markowitz 1991).

Given a subset X of $\mathbb{R}^n$ and p functions $f_j : X \to \mathbb{R}$ for $j = 1,2,...,p$, we define MOP as follows:

$$\text{(MOP)} \quad \min_{x\in\mathbf{X}} F(x) = (f_1(x), f_2(x), ..., f_p(x)),$$

where $F : X \to \mathbb{R}^p$ is the objective function vector. We assume that X is of the form $X = \{x = (x_1, x_2, ..., x_n) \in \mathbb{R}^n : a_i \le x_i \le b_i, i = 1,2.,...,n\}$, where a_i and b_i are the lower and upper bound of the ith component of variable x, respectively. When the objective functions conflict with each other, no single solution can simultaneously minimize all scalar objective functions $f_j(x), j = 1,...,p$. Therefore, it is necessary to introduce a new notion of optimality in multiobjective framework. A most commonly used one is that of Pareto optimality or Pareto efficiency, which is an important criterion for evaluating economic and engineering systems. The definition of Pareto optimality can be provided using Pareto dominance relation as defined below (Abraham et al. 2005):

Definition 1 *Let x_u, $x_v \in \mathbf{X}$ be two decision vectors. $F(x_u)$ is said to dominate $F(x_v)$ (denoted $F(x_u) \prec F(x_v)$) if and only if $f_i(x_u) \le f_i(x_v)$, $^\forall i \in \{1,2,...,p\}$ and $f_j(x_u) < f_j(x_v)$, $^\exists j \in \{1,2,...,p\}$.*

Definition 2 *A point $x^* \in \mathbf{X}$ is said to be globally Pareto optimal if and only if there is no $x \in \mathbf{X}$ such that $F(x) \prec F(x^*)$. Then, $F(x^*)$ is called globally efficient. The image of the set of globally efficient points is called the Pareto front.*

In general, computational methods cannot guarantee global Pareto optimality (Nocedal and Wright 1999), but at best local Pareto optimality which can be defined below.

Definition 3 *A point $x^* \in \mathbf{X}$ is said to be locally Pareto optimal if and only if there exists an open neighborhood of x^*, $\mathbf{B}(x^*)$, such that there is no $x \in \mathbf{B}(x^*) \cap \mathbf{X}$ satisfying $F(x) \prec F(x^*)$. $F(x^*)$ is then called locally efficient. The image of the set of locally efficient points is called the local Pareto front.*

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

Pareto optimality will henceforth refer to local Pareto optimality unless noted otherwise. The set of Pareto optimal points is denoted here by $\mathscr{X}^*$ and the set of efficient points, $F(x_i)$ for $\forall x_i \in \mathscr{X}^*$, is denoted by $\mathscr{P}^*$. The goal of MOP is to identify $\mathscr{P}^*$ which is able to represent the Pareto front. Usual optimization methods for scalar objective functions cannot be employed to MOP. In general, identifying the set of all Pareto optimality points is not a tractable problem, particularly when the knowledge on the structure of the problem is very minimal or not available. A common approach for solving MOP is to find many evenly distributed efficient points, and use those points to approximate the Pareto front. In this paper, we consider the blackbox simulation optimization problems where the structures of the objective functions are not available or very complex, and evaluating the values of objective functions is computationally expensive. Further, the current framework assumes that the simulation output is deterministic. In the future work, we plan to extend it to the case where the simulation output is stochastic.

There are a number of computational methods for solving MOP. The most widely used one in practice is to employ evolutionary algorithms (Abraham et al. 2005, Deb 1999) to generate a large number of Pareto points for decision makers. However, they are usually computationally expensive because a massive number of non-Pareto set points have to be evaluated (Shan and Wang 2005). For a black-box simulation optimization problem, in particular, evolutionary algorithms are not suitable to approximate Pareto points.

Another commonly used approach is weighted sum (WS) method (Cohon 1978, Shan and Wang 2005). The idea of the WS method is to convert the MOP into a single objective optimization problem by using a convex combination of objectives. More formally, the WS method solves the following scalar optimization problem:

$$\begin{array}{ll}(WS(w)) & \min\limits_{x \in \mathbf{X}} \sum\limits_{k=1}^{p} w_k f_k(x) \\ & \text{s.t. } \sum\limits_{i=k}^{p} w_k = 1 \text{ where } w_k \geq 0.\end{array} \tag{1}$$

Under some conditions, the solution to $(WS(w))$ is a Pareto optimal point, and by appropriately changing the weight vector $w = (w_1, w_2, ..., w_p)$, one can approximate the Pareto front. For well-structured problem, the WS method performs well and has nice mathematical properties such as convergence. However, the WS method cannot generate any point in the nonconvex part of the Pareto front as shown in Figure 1. Also, WS method may duplicate solutions with different weight combinations.

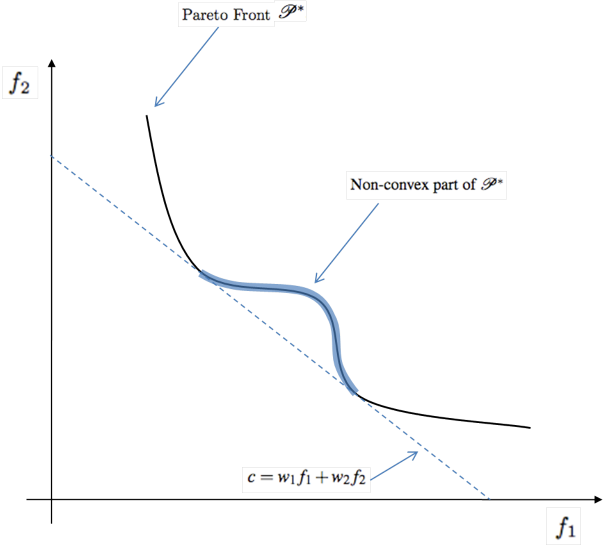

Pareto Front $\mathscr{P}^*$
f_2
Non-convex part of $\mathscr{P}^*$
$c = w_1 f_1 + w_2 f_2$
f_1

Figure 1: WS is unable to generate the non-convex part of the Pareto front

Normal Boundary Intersection (NBI) method also uses a series of tractile single objective optimization problems to approximate Pareto front (Das and Dennis 1998). Starting from a point on the utopia plane which passes through individual function minimizers, a single objective optimization problem in NBI is to maximize the distance from the starting point to a point located on the normal line of the utopia plane. With different starting points on the utopia plane, NBI produces well-distributed solutions. However, this method may be inapplicable to the blackbox simulation optimization problems because the utopia plane and a starting point are not easily obtained without the explicit forms of objective functions.

In the blackbox simulation optimization context, few work has been devoted to MOP. Shan and Wang (2005) developed a method called PSP (Pareto set pursuing) using sampling guidance functions. PSP approximates the objective functions with samples and conduct an iterative process with more samples toward non-dominated points found so far. Audet et al. (2008) proposed a direct search method called BiMads (biobjective mesh adapative direct search). BiMads uses single-objective formulations of BOP (biobjective optimization problem) and then solves single objective problems using Mads (Audet and Dennis 2007). BiMads is used as a benchmark method in our numerical examples.

In this paper, we propose a new method, called Paws (Pareto front Approximation with an adaptive Weighted Sum method), for a blackbox simulation multiobjective optimization problem. Paws is an iterative algorithm which improves a set of non-dominated points toward the Pareto front with the help of a metamodeling scheme and a trust region type method. At each iteration, the metamodel of each single objective function is constructed. The idea of the trust region method (Conn et al. 2000) is adopted to determine the sampling region for the metamodels. Then, the weighted sum method for the MOP is employed to search for Pareto optimal points, where the weight combination is adaptively determined by taking into account all current non-dominated points. Our numerical results show that the method has the potential to achieve a uniform coverage of the Pareto front, even where the Pareto front is nonconvex.

This paper is organized as follows. The main steps of Paws is detailed in Section 2. In Section 3, we test Paws on two problems, each being designed to have a convex and a nonconvex Pareto front, respectively. Section 4 presents some concluding remarks.

2 PARETO FRONT APPROXIMATION WITH ADAPTIVE WEIGHTED SUM METHOD (Paws): PROCEDURE

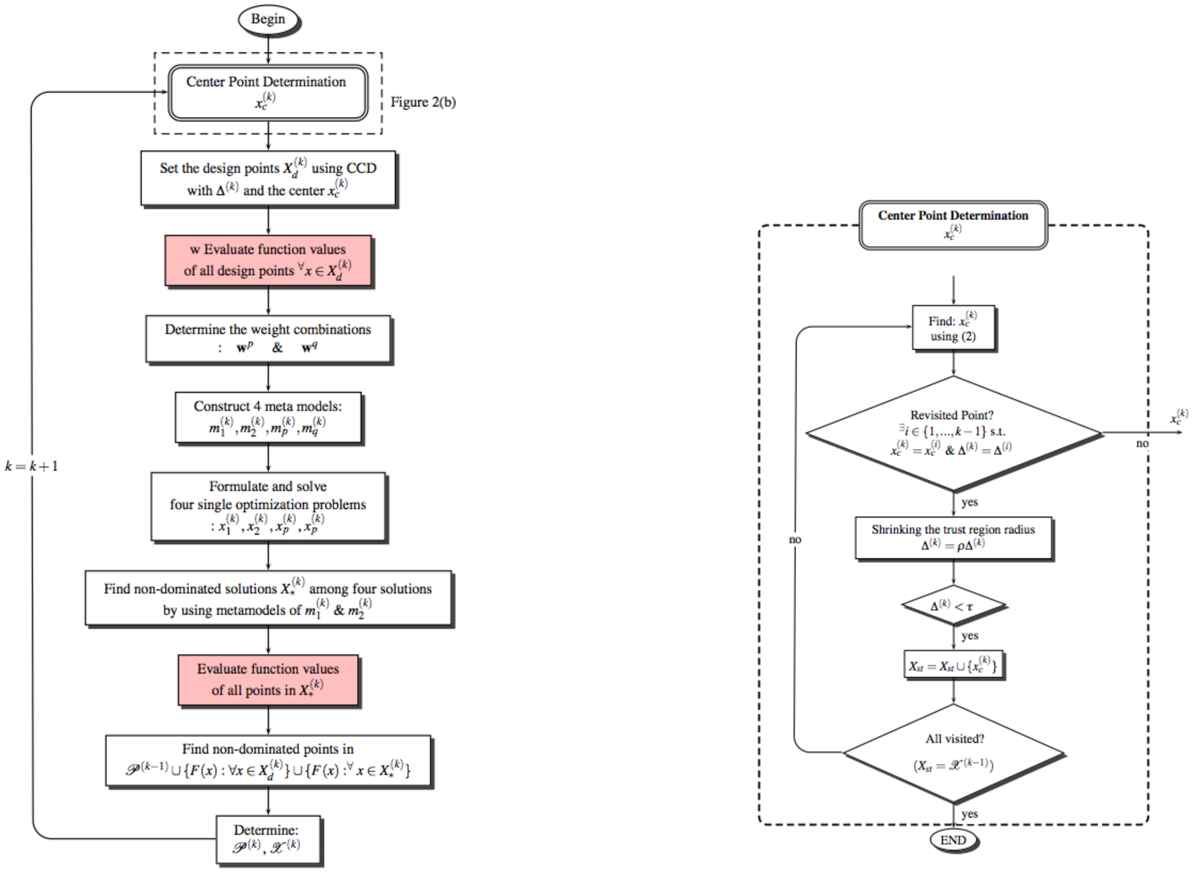
In this paper, we focus on a biobjective optimization problem (BOP), i.e., MOP with two objective functions, $f_1(x)$ and $f_2(x)$. We assume that the considered Pareto front is smooth and continuous. At the initial stage, the user specifies three constants, Δ, ρ, and τ, and a starting point $x_c^{(0)} \in \mathbb{R}^n$. Here, Δ represents the initial trust region radius, ρ is the contraction parameter for TRM, and τ is the tolerance rate for the trust region radius. Initialize $\Delta^{(k)} = \Delta$ for each iteration, $k = 0, 1, 2, ...$, and initialize the set of satisfied points as $X_{st} = \emptyset$. In the conventional trust-region algorithm, the trust region radius will be expanded when there is good agreement between the metamodel and the real function. In Paws, the weight on each function will be determined at each iteration, so the resulted weighted sum function, which is the objective function of Paws, will not be fixed for every iteration. Since the objective function of each iteration tends to be changed, it cannot be guaranteed that a good metamodel in an iteration fits the objective function in the next iteration as well. Thus, the expanding procedure in the conventional trust-region algorithm is not adopted in Paws. Each iteration of Paws mainly consists of three steps: determining a center point, constructing metamodels, and solving optimization problems. The main flow chart of Paws is shown in Figure 2(a).

2.1 Center point Determination (Figure 2(b))

At each iteration k, Paws implements a central composite design (CCD) with a center point $x_c^{(k)}$ and the trust region $\{x : \|x_c^{(k)} - x\| \leq \Delta^{(k)}\}$ (Myers and Montgomery 2002). The center point $x_c^{(k)}$ is determined among the elements of a set resulted from the previous iteration. In the previous iteration $k-1$, a set of non-dominated points, $\mathscr{P}^{(k-1)}$, and a set of decision vectors, $\mathscr{X}^{(k-1)} = \{^{\forall}x^{(k-1)} : F(x^{(k-1)}) \in \mathscr{P}^{(k-1)}\}$, are returned. The cardinality of $\mathscr{P}^{(k-1)}$ (or $\mathscr{X}^{(k-1)}$) is denoted by $J^{(k-1)}$. Both $\mathscr{P}^{(k-1)} = \{F(x_1^{(k-1)}), F(x_2^{(k-1)}), ..., F(x_{J^{(k-1)}}^{(k-1)})\}$ and $\mathscr{X}^{(k-1)} = \{x_1^{(k-1)}, x_2^{(k-1)}, ..., x_{J^{(k-1)}}^{(k-1)}\}$ are sorted in increasing order of f_1 value such that $f_1(x_1^{(k-1)}) \leq f_1(x_2^{(k-1)}) \leq ... f_1(x_{J^{(k-1)}}^{(k-1)})$. Then, in order to measure the gaps between non-dominated points in the objective space, $d_j^{(k-1)}$ for $j = 2, \ldots, J^{(k-1)}$ is computed as

$$d_j^{(k-1)} = \|F(x_{j-1}^{(k-1)}) - F(x_j^{(k-1)})\| + \|F(x_j^{(k-1)}) - F(x_{j+1}^{(k-1)})\|,$$

where $\|\cdot\|$ is the Euclidian distance. The center point $x_c^{(k)}$ is determined as follows.


Begin
Center Point Determination $x_c^{(k)}$
Figure 2(b)
Set the design points $X_d^{(k)}$ using CCD with $\Delta^{(k)}$ and the center $x_c^{(k)}$
w Evaluate function values of all design points $\forall x \in X_d^{(k)}$
Determine the weight combinations : $\mathbf{w}^p$ & $\mathbf{w}^q$
Construct 4 meta models: $m_1^{(k)}, m_2^{(k)}, m_p^{(k)}, m_q^{(k)}$
Formulate and solve four single optimization problems : $x_1^{(k)}, x_2^{(k)}, x_p^{(k)}, x_p^{(k)}$
Find non-dominated solutions $X_*^{(k)}$ among four solutions by using metamodels of $m_1^{(k)}$ & $m_2^{(k)}$
Evaluate function values of all points in $X_*^{(k)}$
Find non-dominated points in $\mathscr{P}^{(k-1)} \cup \{F(x): \forall x \in X_d^{(k)}\} \cup \{F(x): \forall x \in X_*^{(k)}\}$
Determine: $\mathscr{P}^{(k)}, \mathscr{X}^{(k)}$
$k = k+1$
Center Point Determination $x_c^{(k)}$
Find: $x_c^{(k)}$ using (2)
Revisited Point? $\exists i \in \{1,\ldots,k-1\}$ s.t. $x_c^{(k)} = x_c^{(i)}$ & $\Delta^{(k)} = \Delta^{(i)}$
no
$x_c^{(k)}$
yes
Shrinking the trust region radius $\Delta^{(k)} = \rho\Delta^{(k)}$
$\Delta^{(k)} < \tau$
yes
$X_{st} = X_{st} \cup \{x_c^{(k)}\}$
All visited? $(X_{st} = \mathscr{X}^{(k-1)})$
no
yes
END

(a) Main flow chart of PAWS.

(b) Central Point Determination.

Figure 2: Flow charts of PAWS. The dashed box in Figure 2(a) is detailed in Figure 2(b). The shaded boxes in Figure 2(a) indicate the need of function evaluations by simulation.

$$\begin{array}{lll} \text{if } J^{(k-1)} > 2, & x_c^{(k)} = x_{\hat{\imath}}^{(k-1)} & \text{where } \hat{\imath} = \operatorname{argmax}_{j=1,\ldots,J^{(k-1)}} \left[d_j^{(k-1)} \text{ s.t. } x_j^{(k-1)} \notin X_{st} \right], \\ \text{if } J^{(k-1)} = 2, & x_c^{(k)} = x_1^{(k-1)} \text{ or } x_2^{(k-1)} & \text{(randomly select)}, \\ \text{if } J^{(k-1)} = 1, & x_c^{(k)} = x_1^{(k-1)}. & \end{array} \tag{2}$$

When $J^{(k-1)} > 2$, neither $x_1^{(k-1)}$ nor $x_{J^{(k-1)}}^{(k-1)}$ will be selected as $x_c^{(k)}$. If $x_1^{(k-1)}$ and $x_{J^{(k-1)}}^{(k-1)}$ are not extreme points, i.e., the Pareto optimal points having the minimum value of f_1 and f_2, $x_1^{(l)}$ and $x_{J^{(l)}}^{(l)}$ with $l > k$ will be replaced with other points as the iteration goes on. The reason is that this algorithm also searches the minimized solutions for f_1 and f_2 individually (see (7)).

In this algorithm, the elements of X_{st} are considered as Pareto optimal points which is not further selected as the center point. If $x_c^{(k)}$ turns out to be a revisited point, i.e., $\exists i \in \{1,2,\ldots k-1\}$ such that $x_c^{(k)} = x_c^{(i)}$, and $\Delta^{(k)} = \Delta^{(i)}$, then $\Delta^{(k)}$ is replaced by $\rho\Delta^{(i)}$ to shrink the trust region. It is indicated that PAWS at the i^{th} iteration did not generate any point dominating the center point. If $\Delta^{(k)} < \tau$, $x_c^{(k)}$ is considered a satisfied point which are added to the set X_{st}, i.e., $X_{st} = X_{st} \cup \{x_c^{(k)}\}$. Then, $x_c^{(k)}$ is redetermined by (2). If $|X_{st}| = |\mathscr{X}^{(k-1)}|$, it is indicated that PAWS is not able to provide any more non-dominated points, so it is terminated.

2.2 Metamodeling

In this section, we will construct several metamodels, each of which is used as the objective function of a single objective optimization problems formulated in the next section. After determining center point $x_c^{(k)}$, we implement a CCD with $x_c(k)$ as the center point and $\Delta^{(k)}$ as the experimental, and then obtain the function values of the design points through simulations. Note that the simulation results are assumed to be deterministic. The set of design points is denoted by $X_d^{(k)}$. If any point in $X_d^{(k)}$ is outside the feasible region X, then $\Delta^{(k)}$ decreases to ensure that all design points are inside the trust region and at least one of them is located on the boundary. Even if a Pareto optimal point x^* is on the boundary, this procedure enables us to find a point x such that $\|x - x^*\| \leq \tau$ with enough iterations. Using the response surface model technique (Myers and Montgomery 2002), the quadratic meta models for f_1 and f_2 around $x_c^{(k)}$, denoted by $m_1^{(k)}$ and $m_2^{(k)}$ are constructed as follows.

$$
\begin{aligned}
m_1^{(k)}(x) &= f_1(x_c^{(k)}) + \widehat{g_1^{(k)}}(x_c^{(k)})^T (x - x_c^{(k)}) + \tfrac{1}{2}(x - x_c^{(k)})^T \widehat{H_1^{(k)}}(x_c^{(k)})(x - x_c^{(k)}), \\
m_2^{(k)}(x) &= f_2(x_c^{(k)}) + \widehat{g_2^{(k)}}(x_c^{(k)})^T (x - x_c^{(k)}) + \tfrac{1}{2}(x - x_c^{(k)})^T \widehat{H_2^{(k)}}(x_c^{(k)})(x - x_c^{(k)}),
\end{aligned} \tag{3}
$$

where $\widehat{g_\cdot^{(k)}}(x_c^{(k)})$ and $\widehat{H_\cdot^{(k)}}(x_c^{(k)})$ are estimates of gradient, and Hessian matrix at $x_c^{(k)}$, respectively. Additionally, we will construct one or two more metamodels based on the weighted sum method. Note that any additional function evaluations are not required for these metamodels. The two metamodels, $m_p^{(k)}$ and $m_q^{(k)}$, are constructed as the approximations of $\mathbf{w^p} \cdot F^T (= w_1^p f_1 + w_2^p f_2)$, and $\mathbf{w}^q \cdot F^T (= w_1^q f_1 + w_2^q f_2)$, respectively. $m_p^{(k)}$ and $m_q^{(k)}$ are designed for WS in (1) in order to find Pareto optimal points within the trust region around $x_c^{(k)}$. The weight vector for each iteration, $\mathbf{w}^p$, $\mathbf{w}^q$, are determined considering $x_{\hat{\imath}-1}^{(k-1)}$, $x_{\hat{\imath}}^{(k-1)}$ and $x_{\hat{\imath}+1}^{(k-1)}$ in (2) as follows.

$$
\begin{aligned}
&\text{if } J^{(k-1)} > 2, \\
&\qquad \mathbf{w}^p = [w_1^p, w_2^p] \quad = c^p \left[(-f_2(x_{\hat{\imath}}^{(k-1)}) + f_2(x_{\hat{\imath}-1}^{(k-1)})) \quad (f_1(x_{\hat{\imath}}^{(k-1)}) - f_1(x_{\hat{\imath}-1}^{(k-1)})) \right], \\
&\qquad \mathbf{w}^q = [w_1^q, w_2^q] \quad = c^q \left[(-f_2(x_{\hat{\imath}+1}^{(k-1)}) + f_2(x_{\hat{\imath}1}^{(k-1)})) \quad (f_1(x_{\hat{\imath}+1}^{(k-1)}) - f_1(x_{\hat{\imath}}^{(k-1)})) \right], \\
&\text{if } J^{(k-1)} = 2, \\
&\qquad \mathbf{w}^p = [w_1^p, w_2^p] \quad = c^p \left[(-f_2(x_2^{(k-1)}) + f_2(x_1^{(k-1)})) \quad (f_1(x_2^{(k-1)} - f_1(x_1^{(k-1)})) \right], \\
&\text{if } J^{(k-1)} = 1, \\
&\qquad \mathbf{w}^p = [w_1^p, w_2^p] \quad = [0.5 \quad 0.5],
\end{aligned} \tag{4}
$$

where c^p and c^q are constants leading to $w_1^p + w_2^p = w_1^q + w_2^q = 1$. Note that $\mathbf{w}^q$ is not available when $J^{(k-1)} \leq 2$. Based on (3) and (4), the additional quadratic metamodels are constructed as follows.

$$
\begin{aligned}
&\text{if } J^{(k-1)} > 2, \\
&\qquad m_p^{(k)}(x) = w_1^p \cdot m_1^{(k)} + w_2^p \cdot m_2^{(k)}, \\
&\qquad m_q^{(k)}(x) = w_1^q \cdot m_1^{(k)} + w_2^q \cdot m_2^{(k)}, \\
&\text{if } J^{(k-1)} \leq 2, \\
&\qquad m_p^{(k)}(x) = w_1^p \cdot m_1^{(k)} + w_2^p \cdot m_2^{(k)}.
\end{aligned} \tag{5}
$$

2.3 Single objective optimization problems

Using each of two metamodels in (3) as the objective function, two single objective optimization problems can be formulated with the trust region constraint:

$$
\begin{array}{llcll}
\min\limits_{x\in\mathbf{X}} & m_1^{(k)}(x) & \qquad & \min\limits_{x\in\mathbf{X}} & m_2^{(k)}(x) \\
\text{s.t.} & \|x-x_c^{(k)}\| \le \Delta^{(k)}, & & \text{s.t.} & \|x-x_c^{(k)}\| \le \Delta^{(k)}.
\end{array}
\tag{6}
$$

The solutions in (6) will help to find the two extreme points of the Pareto front, so that PAWS is able to approximate the full local Pareto front.

To find Pareto optimal points within the trust region, two single objective optimization problems based on the weighted sum methods in (5) can be formulated as follows.

$$
\begin{array}{llcll}
 & & \text{if } J^{(k-1)} > 2 & & \\
\min\limits_{x\in\mathbf{X}} & m_p^{(k)}(x) & & \min\limits_{x\in\mathbf{X}} & m_q^{(k)}(x) \\
\text{s.t.} & \|x-x_c^{(k)}\| \le \Delta^{(k)}, & & \text{s.t.} & \|x-x_c^{(k)}\| \le \Delta^{(k)}.
\end{array}
\tag{7}
$$

Note that the optimization problems in (6) and (7) are deterministic quadratic optimization problems, so that each solution can be obtained by the any popular solver such as Cauchy point calculation and Sequential Quadratic Programming (Conn et al. 2000). Once the solutions of (6) and (7), denoted by $x_1^{(k)}$, $x_2^{(k)}$, $x_p^{(k)}$ and $x_q^{(k)}$, respectively, are obtained, non-dominated points among these solutions are selected according to the metamodels in (3), and then evaluated through simulations. This can save the expensive evaluation cost by avoiding evaluations of potential dominated points. It is possible that either $x_1^{(k)}$ or $x_2^{(k)}$ can be a dominated point because it is obtained without considering the Pareto optimality. Let $X_*^{(k)}$ be a set of these non-dominated points. Thus, at each iteration, the evaluations are made for design points and all or some of these optimal points. Comparing all elements in a union set of $\mathscr{P}^{(k-1)} \cup \{F(x) :^{\forall} x \in X_d^{(k)}\} \cup \{F(x) :^{\forall} x \in X_*^{(k)}\}$, an updated set of non-domintaed points are determined as denoted by $\mathscr{P}^{(k)}$. Figure 3 shows an example to illustrate how two single objective optimizations in (7) give solutions, $x_p^{(k)}$ and $x_q^{(k)}$.

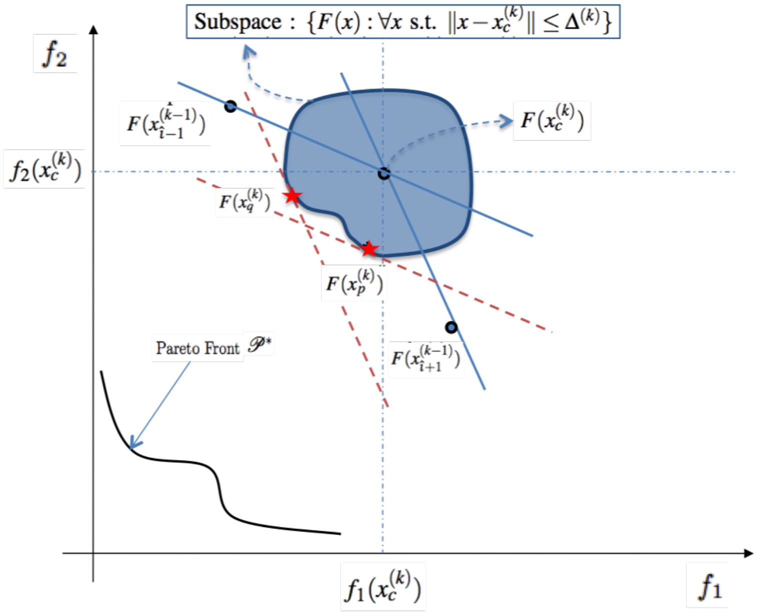
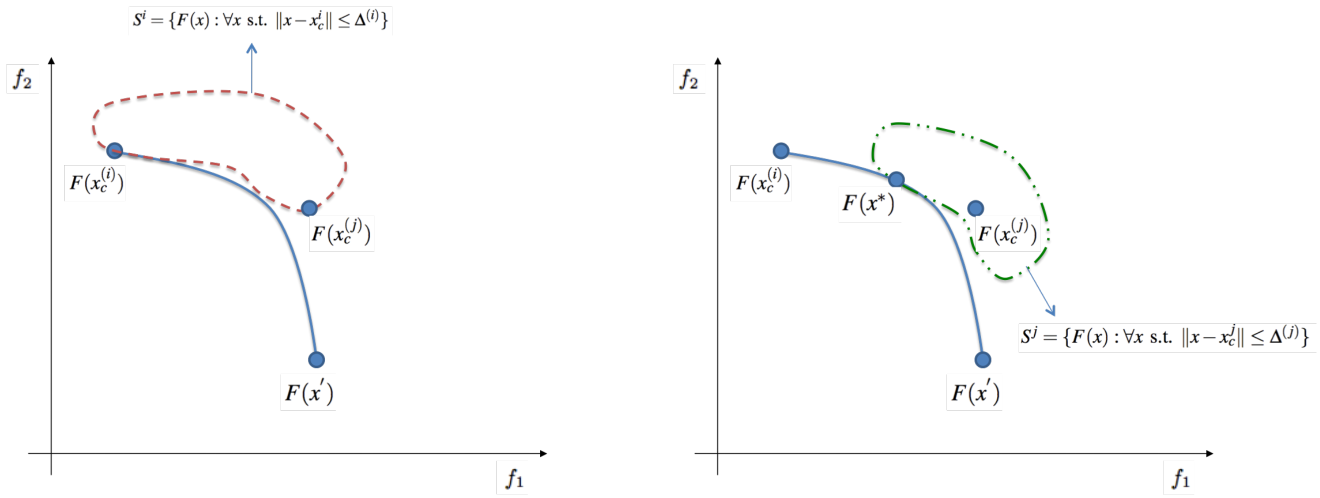


Figure 3: $x_p^{(k)}$ and $x_p^{(k)}$ are obtained in order to obtain points closer to the Pareto front


$S^i = \{F(x) : \forall x \text{ s.t. } \|x - x_c^i\| \le \Delta^{(i)}\}$
f_2
$F(x_c^{(i)})$
$F(x_c^{(j)})$
$F(x^{'})$
f_1
f_2
$F(x_c^{(i)})$
$F(x^*)$
$F(x_c^{(j)})$
$S^j = \{F(x) : \forall x \text{ s.t. } \|x - x_c^j\| \le \Delta^{(j)}\}$
$F(x^{'})$
f_1

(a) The subspace $S^i = \{F(x) : \forall x \text{ s.t. } \|x - x_c^i\| \le \Delta^{(i)}\}$.

(b) The subspace $S^j = \{F(x) : \forall x \text{ s.t. } \|x - x_c^j\| \le \Delta^{(j)}\}$.

Figure 4: $F(x_c^{(j)})$ could be the solution of the second problem in (3), and $F(x^*)$ could be the solution of (5)

2.4 Paws with a nonconvex Pareto front

Paws is a method combining two well-known methods: weighted sum method (WS) and trust region method (TRM). We use WS to approximate the Pareto front by changing the weight combination. If the structure of each objective function in MOP is known, a solution resulted from WS in (1) will converge to a Pareto optimal point (Zadeh 1963). For complicated objective functions in which the evaluation cost is computationally expensive, we use TRM which sequentially constructs a simple quadratic approximation model to generate an approximated solution within a defined trust region. The solution will converge to the stationary point (Conn et al. 2000, Nocedal and Wright 1999). Hence, Paws has a potential to converge to the Pareto optimal point.

As mentioned previously, WS has a difficulty if the Pareto front is nonconvex. In Paws, the trust region around a center point could be small enough to find some points in the nonconvex part of Pareto front as illustrated in Figure 4. All Pareto optimal points between $F(x_c^{(i)})$ and $F(x^{'})$ are located on the nonconvex part of Pareto front. With $x_c^{(i)}$ as a center point at i^{th} iteration, $x_c^{(j)}$ can be possibly found by solving the right-hand-side problem of (6). For $j > i$, $x_c^{(j)}$ will be determined as a center point at j^{th} iteration. Then, one of problems in (7) could produce the solution x^*, which is a Pareto optimal point located on the nonconvex part of the Pareto front. This example could be too simple or ideal to explain how Paws handles the nonconvex part of Pareto front. In the next section, we will test Paws with a biobjective optimization problem whose Pareto front is concave.

3 NUMERICAL EVALUATIONS

In this section, we use two test problems to compare the performance of Paws with that of BiMads. Each test problem has a continuous Pareto front, which can be approximated by evaluating all points on a very dense mesh covering all feasible region. We evaluate $250,000$ uniformly-spaced points $\in \mathbf{X}$, and then find a set $\mathbf{H}$ of 500 uniformly-spaced solutions from the Pareto front in the objective space, which are used to measure convergency of each method.

3.1 Comparison with BiMads

Audet et al. (2008) has developed an algorithm called BiMads (Biobjective Mesh Adaptive Direct Search) as designed to approximate the Pareto front of a biobjective optimization problem. BiMads constructs and solves a series of bound-constrained single objective formulations BOP using Mads (Audet and Dennis 2007) with increasingly stringent stopping criteria. BiMads first solves each objective function individually by Mads, and then formulates and solves a series of single objective optimization problems using a set of nondominated points found so far. For comparison, we set the parameters of BiMads as follows.

- Starting point $x_0 = (0.5, 0.5)$
- Initial mesh size $= 0.01$
- Mesh expansion rate $= 2$
- Mesh contraction rate $= 0.5$
- Stopping criteria
  - Poll size termination: mesh size ≤ 0.0001
  - Truth evaluations termination: at most 50 evaluations of the reformulated objective function unless indicated explicitly.

In this numerical experiment, BiMads generates 10 single-objective formulations, so it is called BiMads10. As a result, the maximum number of function evaluations will be 500. Then, the total number of function evaluations will be the number of the initial evaluations by two Mads for two individual objective functions plus the number of evaluations in BiMads10. As for Paws, the initial parameter setting is that $x_c^{(0)} = (0.5, 0.5)$, $\Delta = 0.2$, $\rho = 0.5$, and $\tau = 0.001$. Paws will be terminated after 30 iterations in this numerical test, so it is called Paws30.

3.2 Performance Measures

In the multi-objective optimization problems, there are usually two considerations to evaluate an algorithm :

- Convergence to the Pareto optimal points,
- Diversity in solutions of Pareto-optimal set.

In this paper, we use two performance measures to evaluate each of the above two goals in a solution set obtained by each algorithm. The first one is GD (Generational Distance) introduced by Veldhuizen and Lamont (1998) as a way of estimating how far the found solutions are from the Pareto front. Thus, GD=0 indicates that all solutions are placed on the Pareto front. We first compute the minimum Euclidean distance $\delta_i, i = 1, 2, ..., n_p$ of each solution from the elements in $\mathbf{H}$, where n_p is the number of solutions found (Deb et al. 2002), and then GD is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n_p} \delta_i^2}}{n_p}.$$

The second one is SP (Spacing) (Abraham et al. 2005) to measure the distance variance of neighboring solutions. In a biobective optimization problems, the found solutions are sorted by values of f_1 and the distances of consecutive solutions are computed as $s_i, i = i, ..., (n_p - 1)$, and then SP is defined as:

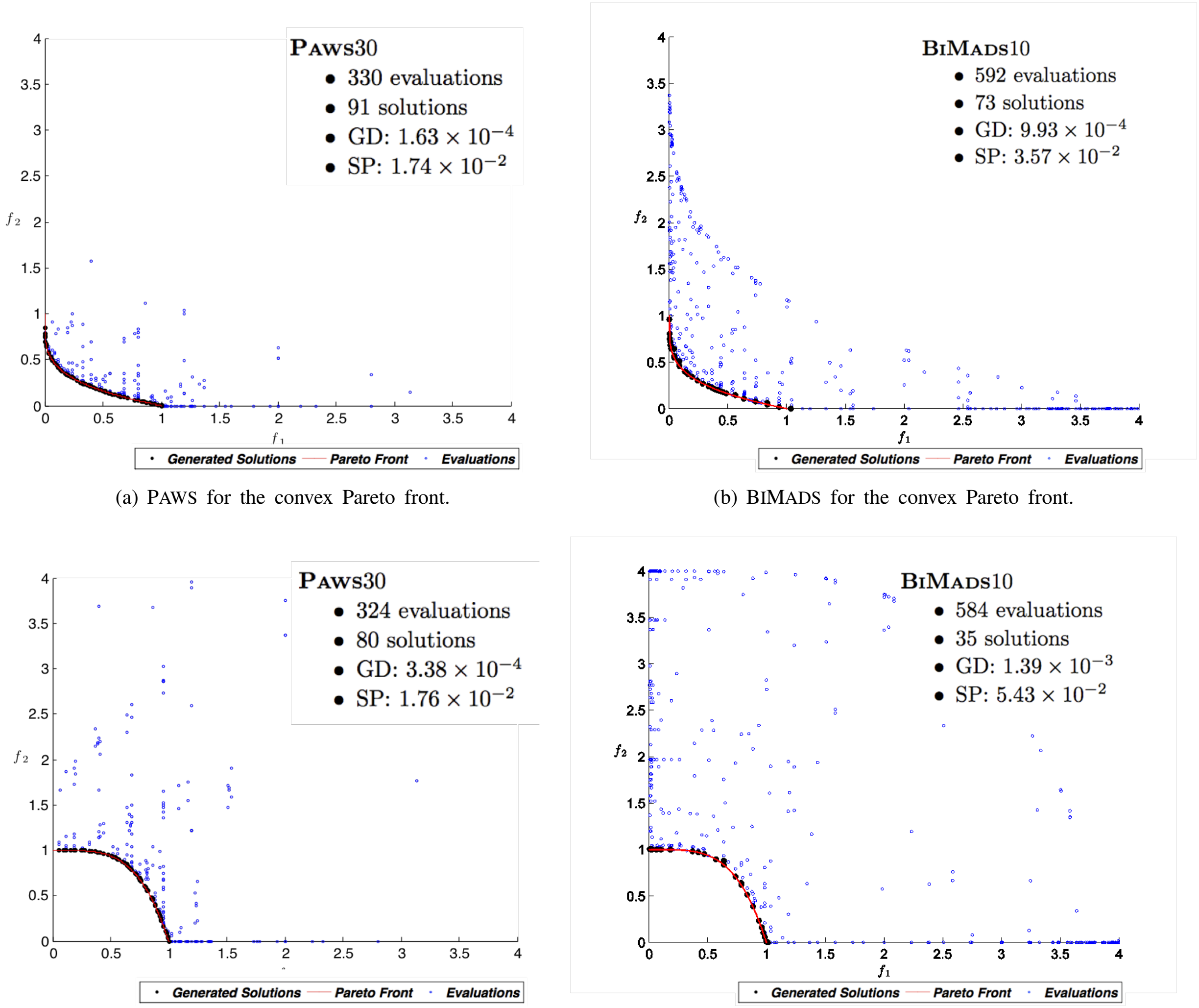
$$SP = \sqrt{\frac{1}{n_p - 2} \sum_{i=1}^{n_p - 1} (\bar{s} - s_i)^2},$$

where $\bar{s}$ is the mean of all s_i. The smaller SP indicates that the solutions are more uniformly spread out.

Additionally, the number of evaluations made are important in simulation optimization problems to see the effectiveness with convergency. Thus, we use these three measures (GD, SP, and the number of evaluations) to compare Paws with BiMads.

3.3 Test Problems

In this paper, two problems in the literature are tested. Both problems are biobjective bound constrained problems from Audet et al. (2008) and Deb (1999). The first problem has a convex Pareto front, and the second problem has a concave Pareto front, so each problem is intended to show how Paws can handle various types of Pareto front. The two objective functions are constructed as follows.



(a) PAWS for the convex Pareto front.

(b) BIMADS for the convex Pareto front.

(c) PAWS for the nonconvex Pareto front.

(d) BIMADS for the nonconvex Pareto front.

Figure 5: Comparisons of PAWS and BIMADS for a convex and a nonconvex Pareto front (starting point $x_0 = (0.5, 0.5)$)

- $f_1(x_1, x_2) = 4x_1$ where $x_1, x_2 \in [0, 1]$
- $f_2(x_1, x_2) = g \cdot h(f_1, g)$

$$\begin{aligned} h(f_1, g) &= \begin{cases} 1 - (\frac{f_1}{g})^{\alpha} & \text{if } f_1 \leq g \\ 0 & \text{otherwise} \end{cases} \\ g(x_2) &= 4 - 3\exp(-\left\{\frac{x_2 - 0.2}{0.02}\right\}^2) \end{aligned} \tag{8}$$

- **Convex Pareto front**
  A biobjective problem with the convex Pareto front can be obtained by setting α to 0.25 in (8). Figure 5(a) and 5(b) show the numerical results of PAWS and BIMADS10, respectively.
- **Nonconvex Pareto front**
  By setting α to 4 in (8), a biobjective problem with the concave Pareto front can be constructed. Figure 5(c) and 5(d) show the numerical results of PAWS and BIMADS10, respectively.

3.4 Numerical Results

As shown in Figure 5, all measures including the number of evaluations, the number of generated solutions, GD, and SP are given in the right-upper part of each plot. We can see that PAW generates much less function evaluations with better convergency (smaller GD) and better diversity of solutions (smaller SP) than BiMads. Since BiMads needs to implement a random search method, it generates a different set of solutions at each run. Therefore, we tested it with at least 30 runs, and then selected the best results among them. The selected results are as good as the results presented in Audet et al. (2008). However, Paws will always generate the same results if the same set of parameters and the same number of iterations are used.

4 DISCUSSION

We developed a new algorithm Paws for blackbox simulation multiobjective optimization problems. When objective functions are evaluated only by a long series of simulations, evaluations of decision variables will be computational expensive and in practice there could be a limit of the number of evaluations under a given budget. Thus, a faster convergency to Pareto optimal with a smaller number of evaluations is usually preferred. We compared our algorithm Paws with BiMads assuming this environment with two test problems from literature. Simulation results for the test problems showed that Paws outperforms BiMads in terms of speed, convergency, and uniformity.

It should be noticed that Paws can only achieve local optimum. When several local Pareto fronts or disconnected Pareto fronts are present, Paws may approximate only a part of the local Pareto fronts. While most of computational methods have this limitation, a recently developed direct search method for biobjective optimization, BiMads, is able to approximate the global Pareto front with a large enough number of runs of a random algorithm over a wide feasible region. However, it is unlikely to achieve the global Pareto front in practice due to the lack of sufficient computational budget. We are currently looking into ways to prove the convergency and the uniformity of solutions generated by Paws for biobjective optimization problems. In future work, we plan to extend this method to problems with more than two objective functions or with stochastic noise.

REFERENCES

Abraham, A., L. C. Jain, and R. Goldberg. 2005. *Evolutionary multiobjective optimization: theoretical advances and applications*. London: Springer-Verlag.

Audet, C., and J. Dennis. 2007, Jan. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization* 17:188–217.

Audet, C., G. Savard, and W. Zghal. 2008, Jan. Multiobjective optimization through a series of single-objective formulations. *SIAM Journal on Optimization* 19 (1): 188–210.

Cohon, J. L. 1978. *Multiobjective programming and planning*. New York: Academic Press.

Conn, A. R., N. I. M. Gould, and P. L. Toint. 2000, Jan. *Trust-region methods*. Philadelphia: Society for Industrial and Applied Mathmatics and Mathematical Programming Society.

Das, I., and J. Dennis. 1998. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8:631–657.

Deb, K. 1999, Jan. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation* 7:205–230.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation* 6 (2): 182–197.

Markowitz, H. 1991. *Portfolio selection: efficient diversification of investments*. Malden, Massachusetts: Blackwell.

Myers, R. H., and D. C. Montgomery. 2002. *Response surface methodology: Process and product optimization using designed experiments*. the second ed. New Jersey: Wiley.

Nocedal, J., and S. J. Wright. 1999. *Numerical optimization*. New York: Springer.

Shan, S., and G. G. Wang. 2005, Jan. An efficient pareto set identification approach for multiobjective optimization on black-box functions. *Journal of Mechanical Design* 127 (5): 866.

Tappeta, R., J. Renaud, and J. Rodríguez. 2002, Jan. An interactive multiobjective optimization design strategy for decision based multidisciplinary design. *Engineering Optimization* 34 (5): 523–544.

Veldhuizen, D. V., and G. Lamont. 1998. Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Writhe-Patterson AFB, OH.

Wilson, B., D. Cappelleri, T. Simpson, and M. Frecker. 2001, Jan. Efficient pareto frontier exploration using surrogate approximations. *Optimization and Engineering* 2:31–50.

Zadeh, L. 1963. Optimality and non-scalar-valued performance criteria. *IEEE Transaction on Automatic Control* 8:59–60.

AUTHOR BIOGRAPHIES

JONG-HYUN RYU is a Ph.D. student in the School of Industrial Engineering at Purdue University. His research interests include simulation optimization, stochastic process and modeling, and quality control. His email address for these proceedings is <ryuj@purdue.edu>.

SUJIN KIM is an assistant professor in the Department of Industrial and Systems Engineering at the National University of Singapore. She received her Ph.D. degree in Operations Research from Cornell University in 2006. Before she joined the National University of Singapore, she was a visiting assistant professor in the Department of Industrial Engineering at Purdue University. Her research concerns simulation methodology and stochastic simulation-based optimization, with applications in electric power and health service systems. Her e-mail address is <iseks@nus.edu.sg>.

HONG WAN is an Assistant Professor in the School of Industrial Engineering at Purdue University. She is currently visiting the department of Operations Research at Naval Postgraduate School. Her research interests include design and analysis of simulation experiments, simulation optimization, applied statistics, quality management, and healthcare engineering. She has taught a variety of courses and is a member of INFORMS and ASA. Her email address is <hwan@purdue.edu > and her web page is <http://web.ics.purdue.edu/hwan>.