AN INTRODUCTION TO MULTI-OBJECTIVE SIMULATION OPTIMIZATION

Susan R. Hunter\textsuperscript{1}, Eric A. Applegate\textsuperscript{1}, Viplove Arora\textsuperscript{1}, Bryan Chong\textsuperscript{1}, Kyle Cooper\textsuperscript{1,2}, Oscar Rincón-Guevara\textsuperscript{1}, and Carolina Vivas-Valencia\textsuperscript{3}

\textsuperscript{1}School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA
\textsuperscript{2}Tata Consultancy Services
\textsuperscript{3}Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN 47907, USA

Abstract

We provide an introduction to the multi-objective simulation optimization (MOSO) problem aimed at researchers and practitioners who wish to begin working in this nascent and underdeveloped area. The MOSO problem is a nonlinear multi-objective optimization (MOO) problem in which multiple simultaneous and conflicting objective functions can only be observed with stochastic error. The solution to this problem is called the Pareto set, that is, the set of decision points for which no other feasible point is better on all objectives. We focus on the so-called \textit{a posteriori} or \textit{vector optimization} methods that characterize the entire Pareto set as input to the decision-making process. Since these methods enable the decision-maker to articulate solution preferences after the optimization is conducted, their popularity in the MOO literature has increased with the increase in available computing power. Likewise, recent MOSO application papers demonstrate an interest in returning several or all points in the Pareto set to the decision-maker. The recent availability of mature and efficient single-objective simulation optimization algorithms, coupled with ubiquitously available parallel computing power, makes identifying the Pareto set as the solution to a MOSO problem seem like an increasingly realistic goal. However, the methodological and theoretical development of specialized \textit{a posteriori} MOSO algorithms remains in its infancy. To assist researchers beginning to work in this area, we provide an introduction to the current MOSO literature, review existing MOSO methods, and discuss some of the key open questions that remain in this emerging field.

\textbf{Key words:} multi-objective simulation optimization; stochastic multi-objective optimization; stochastic vector optimization

1 Introduction

The simulation optimization (SO) problem is a nonlinear optimization problem in which the objective and constraint functions can only be observed with stochastic error. SO problems arise frequently when computer models are used to design stochastic systems, such as when the objective and constraint functions are defined implicitly through a Monte Carlo simulation model. Since almost any level of system complexity can be embedded in such a model, the SO problem formulation arises in a variety of application contexts including transportation and logistics (Osorio and Bierlaire, 2013), energy systems (Subramanyan et al., 2011), building design and optimization (Nguyen et al., 2014), and healthcare (Huang et al., 2016; Rashwan et al., 2016). Because of its generality, the SO problem has received significant attention in the literature. SO methodology has been an active research area for approximately twenty-five years, focused primarily on solving single-objective simulation optimization (SOSO) problems. Many literature overviews and tutorials provide entry
HUNTER ET AL.

points into the SO literature, including Fu (2015), Jian and Henderson (2015), Pasupathy and Ghosh (2013), and Chen and Lee (2010).

Recently, the SO community has expressed interest in developing methods to solve multi-objective simulation optimization (MOSO) problems (Fu et al., 2014). MOSO problems are SO problems with multiple simultaneous and conflicting objectives. The “solution” to a MOSO problem is often characterized as the set of decision points for which no other feasible point is better on all objectives. This set is called the Pareto set. As an example, consider a MOSO problem with two objectives, in which a hospital decision-maker wishes to select an emergency room staffing schedule that simultaneously minimizes the expected staffing cost and minimizes the expected patient wait time. If no single staffing schedule minimizes both objectives, the decision-maker must instead select from among several staffing schedules for which no schedule is better on both objectives. Among the schedules in the Pareto set, trade-offs exist between the expected staffing cost and expected patient wait time. Despite the existence of such MOSO problems in real-world applications, there are currently few MOSO methods available in the literature.

In contrast, deterministic multi-objective optimization (MOO) has been an active research topic for over one hundred years, yielding a variety of solution methods (see, e.g., Chankong and Haimes, 1983; Miettinen, 1999; Marler and Arora, 2004; Ehrgott, 2005). The primary solution method, however, is scalarization (Miettinen, 1999). Scalarization transforms the MOO problem into a parameterized single-objective problem. Varying the scalarization parameters and solving each resulting optimization problem usually yields multiple points in the Pareto set. Thus MOO methods that attempt to characterize the entire Pareto set, called a posteriori or vector optimization methods (Miettinen, 1999, p. 61–64), are arguably more computationally intensive than their single-objective counterparts. Further, they frequently require strong single-objective optimization “engines” (Marler and Arora, 2004, p. 390). The recent availability of increased computing power has fueled interest in a posteriori methods (Eichfelder, 2008). These methods allow decision-makers to articulate their preferences after seeing a characterization of the entire Pareto set.

In the MOSO context, the recent availability of mature and efficient SOSO algorithms, coupled with ubiquitously available parallel computing power, makes identifying the Pareto set as the solution to a MOSO problem seem like an increasingly realistic goal. While early MOSO papers such as Butler et al. (2001) use utility functions to convert the MOSO problem into a single SOSO problem, more recent papers solve the MOSO problem by locating several “good” solutions in the Pareto set, or by characterizing the entire Pareto set. Examples include the flight scheduling application in Lee et al. (2007), the aircraft maintenance scheduling application in Mattila and Virtanen (2014), the supply chain application in Li et al. (2016), and the plant breeding application in Hunter and McClosky (2016). However, the methodological and theoretical development of specialized MOSO algorithms for identifying the entire Pareto set remains in its infancy.

To facilitate future research on MOSO theory, methods, and algorithms, we introduce SO researchers and practitioners to MOSO problem statements, existing MOSO methods, and open MOSO questions, with a focus on a posteriori methods. Since our target audience is SO researchers and practitioners, we assume the reader has knowledge of the SOSO literature. For readers who have not studied the deterministic MOO problem, we provide an introduction to MOO concepts and a brief survey of common MOO methods in §2. Since the MOO literature is vast, we focus our survey on MOO methods that are helpful to understanding the existing MOSO literature. Then, we review the existing MOSO literature in §3–§5. As with any literature review, we are forced to be selective regarding which papers and results we cover in detail. We tend to emphasize results and methods
that represent, in our opinion, those that are the most fundamental, well-developed, or well-known. We have, however, made an earnest attempt to provide a compendium of relevant references for the interested reader. To emphasize the impact of developing MOSO methods, throughout the paper, we also discuss existing MOSO applications. As part of the concluding remarks (§6), we discuss several open questions that remain in this emerging area.

1.1 Problem Setting

To facilitate our discussion of SO problems, we formulate the general SO problem as

Problem $S$: \[
\begin{align*}
\text{minimize} & \quad (E[G_1(x, \xi)], \ldots, E[G_d(x, \xi)]), \\
\text{s.t.} & \quad E[H_i(x, \xi)] \leq 0, \text{ for all } i = 1, 2, \ldots, c, \\
& \quad x \in D,
\end{align*}
\]

where $(E[G_1(x, \xi)], \ldots, E[G_d(x, \xi)])$ is a vector of $d \geq 1$ unknown objective functions, $E[H_i(x, \xi)]$ are unknown constraint functions for all $i = 1, 2, \ldots, c$ ($c = 0$ implies no stochastic constraints), the feasible set $D \subseteq \mathbb{R}^q$ is nonempty and known, and $\xi$ is a random vector with support $\Xi$, such that $\mathbb{P}\{\xi \in \Xi\} = 1$.

Then we define the MOSO problem, called Problem $M$, as Problem $S$ with $d \geq 2$ objectives and no stochastic constraints ($c = 0$),

Problem $M$: \[
\begin{align*}
\min_{x \in D} & \quad \{g(x) = (g_1(x), \ldots, g_d(x)) : (E[G_1(x, \xi)], \ldots, E[G_d(x, \xi)])\},
\end{align*}
\]

where $g : D \rightarrow \mathbb{R}^d$ is a vector-valued function. The solution to Problem $M$ is the Pareto optimal set $\mathcal{P}$. To define the Pareto set formally, we first define what it means for one point in the decision space to dominate another. Then we define what it means for a decision vector to be considered Pareto optimal.

Definition 1. Let $x_1, x_2 \in D$. We say that a decision point $x_1$ dominates $x_2$, and we write that $g(x_1) \preceq g(x_2)$, if and only if $g_k(x_1) \leq g_k(x_2)$ for all $k = 1, \ldots, d$ and $g_{k^*}(x_1) < g_{k^*}(x_2)$ for at least one $k^* \in \{1, \ldots, d\}$.

Definition 2 (see Miettinen, 1999, p. 11). A decision vector $x^* \in D$ is Pareto optimal if there does not exist another decision vector $x \in D$ such that $g(x) \preceq g(x^*)$.

The globally Pareto optimal set is the set of non-dominated points $\mathcal{P}$, and the image of $\mathcal{P}$ is the efficient set $\mathcal{E}$. That is,

$$\mathcal{P} := \{x^* \in D : \forall x \in D \ni g(x) \preceq g(x^*)\} \quad \text{and} \quad \mathcal{E} := \{g(x^*) : x^* \in \mathcal{P}\}.$$ 

To avoid the trivial case, we assume $|\mathcal{P}| > 1$. This assumption implies that there is no single point that is the best on all objectives. Note that a feasible point $x \in D$ may be locally Pareto optimal but not globally Pareto optimal, in the sense that no other point in an appropriately-defined neighborhood dominates $x$, even though some other point in $D$ dominates $x$.

Definition 3. A vector $x^* \in D$ is locally Pareto optimal if, in an appropriately-defined neighborhood $B(x^*)$ of $x^*$, there does not exist $x \in D \cap B(x^*)$ such that $g(x) \preceq g(x^*)$. 

We further discuss the difference between locally and globally Pareto optimal sets in §2.1, which also contains a figure depicting a local Pareto set and a global Pareto set.

Since the objective functions in Problem $M$ are unknown, we assume the existence of a simulation oracle capable of producing consistent estimates of all unknown functions for each $x \in D$. That is, we assume that for some simulation budget $n$, we can obtain estimators of the function values to construct the sample-path problem

$$\text{Problem } \hat{M} :$$

$$\min_{x \in D} \left\{ \bar{G}(x, n) = (\bar{G}_1(x, n), \ldots, \bar{G}_d(x, n)) := (\frac{1}{n} \sum_{i=1}^{n} G_1(x, \xi_i), \ldots, \frac{1}{n} \sum_{i=1}^{n} G_d(x, \xi_i)) \right\},$$

where $E[\bar{G}_k(x, n)] = g_k(x)$, implying the estimators are unbiased for all $k = 1, \ldots, d$. We further assume that the Law of Large Numbers (LLN) holds pointwise, so that for each $x \in D$ and $k = 1, \ldots, d$, $G_k(x, n) \to g_k(x)$ with probability one (w.p.1) as $n \to \infty$. (When we require the uniform LLN, which implies for each $k = 1, \ldots, d$, $\bar{G}_k(x, n) \to g_k(x)$ uniformly in $x$ w.p.1 as $n \to \infty$, we discuss it explicitly.) We denote the solution to Problem $M$ as the estimated (globally) Pareto optimal set $\hat{P}$, and we denote its estimated image as the estimated efficient set $\hat{E}$,

$$\hat{P} := \{ X^* \in D : \forall x \in D \ni \bar{G}(x, n) \preceq \bar{G}(X^*, n) \} \quad \text{and} \quad \hat{E} := \{ \bar{G}(X^*, n) : X^* \in \hat{P} \},$$

where $X^*$ denotes a random vector. Thus the broad problem statement we consider in this paper is,

**Given:** an oracle capable of producing the $d$-dimensional consistent estimate $G(x, n)$ of $g(x)$ for each $x \in D$,

**Find:** the solution to Problem $M$, which is the Pareto set $P$, using only $\bar{G}(\cdot, n)$.

### 1.2 Scope

As mentioned previously, the literature on MOO methods has been a topic of research spanning multiple fields for over one hundred years. In this time, many MOO methods have been proposed, resulting in some disagreement: “[c]learly, . . . a rational person would never deliberately select a dominated point. This is probably the only important statement in multiobjective optimization that can be made without the possibility of generating some disagreement,” (Rosenthal, 1985, p. 138). Likewise, there exist many potential approaches for solving the MOSO problem. In this section, we state the scope of our paper, which focuses on a subset of these potential approaches.

First, we consider only a posteriori methods that have the goal of characterizing the entire Pareto set, or some well-defined subset of the Pareto set that has cardinality larger than one. Second, we adopt the philosophy of Schmeiser (2008), that solution methods for SO problems should require the decision-maker to provide only the problem context and the desired solution quality. In the context of MOSO, providing the problem context means providing the simulation oracle and determining the objectives. Specifying the desired solution quality means specifying some measure of the quality of the estimated Pareto set, or a total simulation budget that limits the number of simulation oracle calls. As such, we do not consider interactive MOO or MOSO methods that repeatedly solicit information from the decision-maker. (Some references to interactive methods for MOSO are provided in Tekin and Sabuncuoglu, 2004.)

We cover existing methods to solve MOSO problems that appear as part of the broader literature on SO. Table 1 categorizes example papers in the existing SO literature by the nature of the variables
that comprise the feasible set, the desired solution type, and the number of objectives and constraints considered in Problem $S$. As in Pasupathy and Henderson (2006), the variables comprising the feasible set $\mathcal{D}$ may be categorical, integer-ordered, continuous, or mixed, and the solution type may be local or global. If the variables comprising $\mathcal{D}$ are categorical, integer-ordered, or continuous, then we refer to $\mathcal{D}$ as a finite, integer-ordered, or continuous set, respectively. Since methods designed specifically for mixed variables are uncommon, we omit this category in the table. However some methods, such as Huang and Zabinsky (2014), can accommodate mixed variables. When categorizing methods by the number of objectives and constraints accommodated, we place each method in the most general category of problem types for which it may be used.

In the sections that follow, we review the existing MOSO methods that appear in the two right-hand columns of Table 1. We group the methods into sections by the nature of the feasible set $\mathcal{D}$: methods for categorical variables or finite sets appear in §3, methods for continuous MOSO appear in §4, and methods for integer-ordered MOSO appear in §5. While those familiar with SOSO literature likely understand what we mean by convergence to a local or global solution in the SOSO context, the notion of convergence to a local or global Pareto set in the MOSO context may be less clear. We address convergence separately in each section, based on the nature of the feasible set $\mathcal{D}$. Note that as in the SOSO context, for MOSO on finite sets the returned estimated Pareto set is always the estimated globally Pareto optimal set.

Metaheuristics have broadly been omitted from Table 1, since they tend not to contain the probabilistic guarantees standard in SO literature (Ólafsson, 2006). However, Multi-Objective Evolutionary Algorithms (MOEAs) are currently popular methods for solving MOSO problems [Zhou et al., 2011 and Lee et al., 2007]. For completeness, we provide a brief overview of MOEAs in Appendix A. Beyond Appendix A, we do not consider MOEAs or other metaheuristics, such as the Nelder-Mead approach of Kuriger and Grant (2011), in this paper. We also do not cover methods to solve the multi-objective multi-armed bandit problem (Yahyaa et al., 2014a,b,c,d, 2015; Drugan and Manderick, 2015).

1.3 Terminology and Notation

There is some flexibility in the literature regarding how to name the set of Pareto optimal points in the decision vector space and the image of this set in the objective vector space. For example, Ehrgott and Tenfelde-Podehl (2003, p. 120) and Audet et al. (2008, p. 190) state that Pareto optimal points are in the decision space, and efficient points are in the objective vector space. Using an opposite convention, Wieck et al. (2016, p. 742) specify that efficient solutions are in the decision space and Pareto points are the images of the efficient solutions. Because conventions vary, Ehrgott (2005, p. 24, 60) reminds the reader to check the definitions each author adopts.

For broad consistency with the literature on MOSO so far, as defined in §1.1, we keep the notation of $\mathcal{P}$ for the set of Pareto optimal points in the decision space (see, e.g., Li et al., 2015a, Feldman and Hunter, 2016), and we use the notation $\mathcal{E}$ for the image of $\mathcal{P}$ (see, e.g., Fliege and Xu, 2011). We call the points in $\mathcal{E}$ “efficient.” Since almost everyone seems to agree that the term frontier refers to the objective vector space, we frequently refer to the set $\mathcal{E}$ as the efficient frontier.

We use the following notation in the remainder of the paper. We use $\mathbb{N}$ to denote the set of natural numbers (positive integers) and $\mathbb{Z}^d$ to denote the set of all $d$-dimensional integer-valued vectors, $\mathbb{Z}^d \subset \mathbb{R}^d$. Where it is reasonable to do so, capital letters denote random variables ($X$), script capital letters denote sets ($\mathcal{P}$), vectors appear in bold ($x$), and random vectors appear in capital bold ($X$). For two vectors $a \in \mathbb{R}^d$ and $b \in \mathbb{R}^d$, we say $a \leq b$ if $a_i \leq b_i$ for all $i = 1, \ldots, d$. 
Table 1: Example papers representing the state of the art in solving SO problems are categorized by the nature of the feasible set $D$, the type of solution desired, and the number of objectives and constraints in Problem $S$.

<table>
<thead>
<tr>
<th>$D$, Global (G) or Local (L)</th>
<th>Solution</th>
<th>SOSO ($d = 1, c = 0$)</th>
<th>SOSO with Stochastic Constraints ($d = 1, c \geq 1$)</th>
<th>MOSO with Two Objectives ($d = 2, c = 0$)</th>
<th>MOSO ($d \geq 2, c = 0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite (G); also called R&amp;S</td>
<td>See Kim and Nelson (2006), Parallel alg.: Luo et al. (2000), Luo et al. (2015), Ni et al. (2015)</td>
<td>Andradóttir and Kim (2010), Luo and Lim (2013), SCORE (Pasupathy et al., 2014)</td>
<td>Hunter and McClusky (2016), SCORE (Feldman and Hunter, 2016), M-MOBA (Branke and Zhang, 2015; Branke et al., 2016)</td>
<td>MOCBA (Lee et al., 2010)</td>
<td></td>
</tr>
<tr>
<td>Integer-Ordered, L</td>
<td>COMPASS (Xu et al., 2010), Lim (2012), R-SPLINE (Wang et al., 2013)</td>
<td>Luo and Lim (2013)</td>
<td>MO-COMPASS (Li et al., 2015a)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous, G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Parts of this table were adapted from Wang et al. (2013).
Acronyms: Multi-Objective Optimal Computing Budget Allocation (MOCBA); Multi-Objective Probabilistic Branch and Bound (MOPBnB); Ranking and Selection (R&S); Stochastic Approximation (SA); Sample Average Approximation (SAA); Sampling Criteria for Optimization using Rate Estimators (SCORE). COMPASS, R-SPLINE, and variants are algorithm names.

The $d$-dimensional vector of zeros is denoted $0_d$. We define the sum of two sets $A$ and $B$ as the Minkowski sum $A + B := \{a + b : a \in A, b \in B\}$. If $g : D \subseteq \mathbb{R}^q \rightarrow \mathbb{R}^d$ is a vector-valued function, then for some set $S \subseteq D$, we define the set $g(S)$ as the image of the set $S$, $g(S) := \{g(x) : x \in S\}$. We also require the following notions of distance. Let $A \subseteq \mathbb{R}^q$ and $B \subseteq \mathbb{R}^q$ be two nonempty, bounded sets. Then (1) $d(x, x') = ||x - x'||$ is the Euclidean distance between two points $x, x' \in \mathbb{R}^q$; (2) $d(x, B) = \inf_{x' \in B} ||x - x'||$ is the distance from the point $x \in \mathbb{R}^q$ to the set $B$; (3) $\mathbb{D}(A, B) = \sup_{x \in A} d(x, B)$ is the distance from set $A$ to set $B$; and (4) $\mathbb{H}(A, B) := \max\{\mathbb{D}(A, B), \mathbb{D}(B, A)\}$ is the Hausdorff distance between sets $A$ and $B$.

2 Overview of Deterministic Multi-Objective Optimization

In this section, we consider the deterministic version of Problem $M$, that is, the version of Problem $M$ having no noise. Our goal is to provide the reader with background material on MOO concepts, theory, and methods that facilitate the discussion of existing MOSO methods. We first cover preliminary MOO concepts, including the basic structure of MOO problems, ideal and nadir points, global and local Pareto sets, and weakly Pareto optimal points. Then we discuss a variety of scalarization methods for MOO and briefly discuss non-scalarization methods. Finally, we discuss available MOO problem testbeds. Since the MOO literature is vast, we refer the reader to Miettinen (1999), Ehrigott (2005), Eichfelder (2008), Wieczek et al. (2016), Ehrigott et al. (2016), and references therein for additional theoretical results, and to Stadler (1979), White (1990), Marler and Arora (2004), and Chinchuluun and Pardalos (2007) for additional MOO overviews, background
information, and interesting MOO applications. Unless stated otherwise, in this section, we assume continuous decision variables in Problem $M$.

### 2.1 Preliminary MOO Concepts

We begin by introducing basic structural aspects of MOO problems. Unlike in the single-objective setting, we do not necessarily have one optimal objective function value in Problem $M$. Instead, we may have many $d$-dimensional “optimal” objective vectors that comprise the efficient frontier, $E$. An example Pareto set $P$ and efficient frontier $E$ are shown in Figure 1 for the case of two objectives.

We also define two important points in the objective space: the ideal and nadir vectors. The ideal vector is constructed by minimizing each objective function over the feasible set $D$. The resulting optimal objective function values are then stored in a vector $g^{\text{ideal}}$, defined as

$$
g^{\text{ideal}} := \left( \min_{x \in D} g_1(x), \min_{x \in D} g_2(x),..., \min_{x \in D} g_d(x) \right) = \left( \min_{x \in P} g_1(x), \min_{x \in P} g_2(x),..., \min_{x \in P} g_d(x) \right).
$$

Since the ideal vector can be obtained by solving for the global minimum of each objective, it is considered relatively “easy” to obtain. Notice that the ideal vector can only be the solution to Problem $M$ if none of the objectives conflict. The nadir vector is defined as

$$
g^{\text{nadir}} := \left( \max_{x \in P} g_1(x), \max_{x \in P} g_2(x),..., \max_{x \in P} g_d(x) \right).
$$

In two dimensions, one can obtain the nadir point from values used to create the ideal point (Ehrgott and Tenfelde-Podehl, 2003). In more than two dimensions, the value of the nadir objective vector may be more difficult to determine. Ehrgott and Tenfelde-Podehl (2003) and Kirlik and Sayin (2013) consider methods to obtain or approximate $g^{\text{nadir}}$ when $D$ is continuous and discrete, respectively. Figure 1 shows an illustration of a bi-objective efficient frontier $E$ and its corresponding ideal and nadir objective vectors. Together, these two points can be used to construct upper and lower bounds on the possible efficient objective function values that can be obtained by solving the MOO problem (Ehrgott and Tenfelde-Podehl, 2003).

![Diagram](image.png)
In Figure 1, the pre-image of the efficient frontier $\mathcal{E}$ is the globally Pareto optimal set $\mathcal{P}$. It is also possible to have a set of locally Pareto optimal points, whose image we have chosen to depict as a “local efficient frontier.” Recall that locally Pareto optimal points are Pareto optimal in some appropriately-defined neighborhood which is a subset of the decision vector space $\mathcal{D}$ (see Definition 3). Unless otherwise noted, we use the term “Pareto set” and the notation $\mathcal{P}$ to refer to the global Pareto set.

Figure 1 also contains what we call “weakly efficient points,” which are the images of weakly Pareto optimal points, defined as follows.

**Definition 4** (Miettinen, 1999, p. 19). A decision vector $x^* \in \mathcal{D}$ is weakly Pareto optimal if there does not exist another decision vector $x \in \mathcal{D}$ such that $g_k(x) < g_k(x^*)$ for all $k = 1, \ldots, d$.

Thus weakly Pareto points are points for which no other point is “strictly better” on all objectives. We denote the set of (globally) weakly Pareto points as

$$\mathcal{P}^w := \{ x \in \mathcal{D} : \forall x' \in \mathcal{D} \exists g_k(x') < g_k(x) \text{ for all } k = 1, \ldots, d \},$$

which is not depicted in Figure 1. Notice that $\mathcal{P} \subseteq \mathcal{P}^w$.

As in the single-objective context, MOO problems in which all of the objective functions $g_k(x), k = 1, \ldots, d$ are convex and the feasible region $\mathcal{D}$ is convex have special structure. Specifically, if the objective functions and feasible region are convex in Problem $M$, then we say Problem $M$ is convex (Miettinen, 1999, p. 7), and all locally Pareto optimal points are also globally Pareto optimal (Miettinen, 1999, p. 12, Theorem 2.2.3). Further, the set $g(\mathcal{D}) + \{ y \in \mathbb{R}^d : 0_d \leq y \}$ is a convex set (Wieckel et al., 2016, p. 747).

### 2.2 Scalarization Methods for Solving MOO Problems

Since scalarization is among the most prominent methods for solving MOO problems, we now discuss several common scalarization functions. Recall that scalarization methods use scalarization functions to convert the multi-objective problem into a parameterized single-objective problem that can be solved using well-established nonlinear optimization algorithms. Under appropriate regularity conditions, solving the single-objective scalarized problem results in a Pareto optimal point. By varying the parameters of the scalarized problem, we may retrieve a set of Pareto optimal points and a characterization of the efficient frontier in the objective function space.

The scalarization methods we discuss in this section include linear weighted sum, $L_p$-norm, Chebyshev, $\varepsilon$-constraint, and the single-objective product formulation. Many other popular scalarization methods exist that, due to space constraints, we do not discuss. These include the Normal Boundary Intersection (NBI) method, developed in Das and Dennis (1998), and the Normal Constraint (NC) method, discussed in Messac et al. (2003). For intuition, we present each scalarization separately, although some are special cases of the Pascoletti-Serafini scalarization (Eichfelder, 2008; Pascoletti and Serafini, 1984). In §2.2.5, we briefly discuss methods for choosing a set of parameterizations that result in an “even” representation of points in the efficient frontier $\mathcal{E}$.

#### 2.2.1 Linear Weighted Sum

In the linear weighted sum method, we minimize the sum of the weighted objectives. Given a weight vector $w = (w_1, \ldots, w_d) \in \mathbb{R}^d$ such that $\sum_{k=1}^d w_k = 1$ and $w_k \geq 0$ for all $k = 1, \ldots, d$, define

Problem $S_{\text{linear}}(w)$: minimize$_{x \in \mathcal{D}} \sum_{k=1}^d w_k g_k(x)$. 

If the weights $w_k$ are strictly positive for all $k = 1, \ldots, d$, then the solution to Problem $S_{\text{linear}}(w)$ is Pareto optimal (see Miettinen, 1999, p. 78, Theorem 3.1.2). If the MOO problem is convex, then for each Pareto optimal point $x^* \in \mathcal{P}$, there exists a weight vector $w$ such that $x^*$ is a solution to Problem $S_{\text{linear}}(w)$ (see Miettinen, 1999, p. 79, Theorem 3.1.4). If the MOO problem is non-convex, there may be points in the Pareto set that no weight vector can retrieve. Marler and Arora (2004, p. 389) consider the linear weighted sum a “benchmark approach” to solving MOO problems, since it is among the most common and easiest-to-implement approaches.

2.2.2 $L_p$-norm and Chebyshev

Given weights $w$ defined as in the linear weighted sum method, we generalize the linear weighted sum into the $L_p$-norm formulation

$$\text{Problem } S_p(w): \quad \min_{x \in \mathcal{D}} \left[ \sum_{k=1}^{d} w_k |g_k(x) - z^*_k|^p \right]^{1/p},$$

where $1 \leq p < \infty$ and $z^* = (z^*_1, \ldots, z^*_d)$ is a reference point such as the ideal vector. If $p = \infty$, we have the weighted Chebyshev ($L_\infty$-norm) scalarization

$$\text{Problem } S_\infty(w): \quad \min_{x \in \mathcal{D}} \max_{k \in \{1, \ldots, d\}} w_k |g_k(x) - z^*_k|.$$

Figure 2 shows contours of the $L_p$-norm scalarization for varying values of $p$. Like the linear weighted sum method, for positive weights, the $L_p$-norm scalarization with $1 \leq p < \infty$ returns Pareto optimal points, although the computational complexity increases as $p$ increases. If the MOO problem is non-convex, there may be points that no weight vector can retrieve. Chebyshev scalarization, however, can retrieve all Pareto points, even for non-convex problems. For Chebyshev scalarization, positive weighting coefficients return weakly Pareto optimal solutions. (See Miettinen, 1999, p. 97–99.)

If we wish to solve Problem $M$ using Chebyshev scalarization, the resulting Problem $S_\infty(w)$ is nonsmooth due to the maximum operation. Several reformulations of Chebyshev scalarization exist to make the optimization problem smooth. Fliege and Xu (2011) introduce a smoothing Chebyshev scalarization defined as

$$\text{Problem } S_\infty^{\text{smooth}}(w, t): \quad \min_{x \in \mathcal{D}} p((w_1(g_1(x) - z_1^*), \ldots, w_d(g_d(x) - z_d^*)), t),$$

where the function $p(y, t) := t \log \sum_{k=1}^{d} e^{y_k/t}$ for parameter $t > 0$, and $p(y, 0) := 0$. Fliege and Xu (2011) investigate the properties of this scalarization, including the deviation between the efficient sets resulting from solving the smoothed and non-smoothed problems as a function of $t$. Fliege and Xu (2011) also investigate the use of the smoothing Chebyshev scalarization in a MOSO context, which we revisit in §4. Other modifications to the Chebyshev scalarization include the augmented version by Steuer and Choo (1983). For further details, see Miettinen (1999, p. 67–71, 97–107).
2.2.3 \(\varepsilon\)-Constraint

The \(\varepsilon\)-constraint method is a scalarization method that reformulates the MOO problem into a constrained single-objective optimization problem. That is, for a chosen objective \(k^*\) and vector of constraints \(\varepsilon = (\varepsilon_1, \ldots, \varepsilon_{k^*-1}, \varepsilon_{k^*+1}, \ldots, \varepsilon_d)\), we formulate the \(\varepsilon\)-constraint problem as

\[
\text{Problem } S(\varepsilon, k^*): \quad \text{minimize}_{x \in D} \ g_{k^*}(x) \quad \text{s.t.} \quad g_k(x) \leq \varepsilon_k \text{ for all } k = 1, \ldots, d; k \neq k^*.
\]

If Problem \(S(\varepsilon, k^*)\) is feasible, then its solution is weakly Pareto optimal (Miettinen, 1999, p. 85, Theorem 3.2.1). Further, a point \(x^*\) is Pareto optimal if and only if, for all objectives \(k = 1, \ldots, d\), it is the solution to Problem \(S(\varepsilon, k^*)\) with \(\varepsilon_k = g_k(x^*)\) for all \(k = 1, \ldots, d, k \neq k^*\) (Miettinen, 1999, p. 85, Theorem 3.2.2). Thus we must solve \(d\) \(\varepsilon\)-constraint problems to certify that a solution is Pareto optimal. Like the Chebyshev scalarization method, the \(\varepsilon\)-constraint method can retrieve all Pareto optimal points in non-convex problems by appropriately varying the \(\varepsilon\) vector.

2.2.4 Single-Objective Product Formulation

The single-objective product formulation is a scalarization introduced by Audet et al. (2008) that results in an optimization problem with desirable properties. For example, under mild regularity conditions, the scalarized objective function is locally Lipschitz continuous and differentiable whenever those properties are held by the functions comprising the vector \(g\). The scalarization is defined as

\[
\text{Problem } S_{\text{prod}}(r): \quad \text{minimize}_{x \in D} \quad \phi(x; r) := -\prod_{k=1}^d \left(\max\{r_k - g_k(x), 0\}\right)^2,
\]

where \(r = (r_1, \ldots, r_d)\) is a reference point. We indicate the reference point as \(r\) instead of \(z^*\) since it is usually a dominated feasible point and hence not the ideal. Audet et al. (2008) show that if \(x^*\) is a solution to Problem \(S_{\text{prod}}(r)\) with \(\phi(x, r) < 0\), then \(x^*\) is a Pareto optimal point. Multiple points in the Pareto set can be found by varying the reference point \(r\). Kim and Ryu (2011a) use this scalarization to solve continuous MOSO problems, which we cover in §4.

2.2.5 Obtaining an “Even” Representation of the Efficient Frontier through Scalarization

For each of the scalarizations we present, a characterization of the Pareto set can be obtained by varying the scalarization parameters and solving many single-objective optimization problems. Ideally, the points returned by the scalarized problems should be sufficiently spread out in the efficient frontier \(E\) and not clustered together in one region. Thus researchers are interested in identifying scalarization parameters that result in a set of Pareto optimal points that is, in some rigorously-defined sense, “evenly spaced” across the efficient frontier. (See Eichfelder, 2008, p. 101–107 for one such definition of evenly-spaced points.) Unfortunately, as explained in Das and Dennis (1997) in the context of the linear weighted sum method, choosing evenly-spaced scalarization parameters does not necessarily produce evenly-spaced points in the efficient frontier. Methods to produce an even representation of the efficient frontier include the adaptive scalarization methods of Eichfelder (2008), the derivative pursuit method of Singhee and Castalino (2010), and the adaptive weighting scheme of Deshpande et al. (2016). The optimal rate for adding vertices to polyhedral approximations of convex bodies is discussed in Efremov and Kamenev (2011).

2.3 Non-Scalarization Methods for Solving MOO Problems

Unlike scalarization methods, non-scalarization methods do not convert the multi-objective problem into a set of parameterized single-objective problems. Instead, they use gradient information from
each of the objectives to find a point that satisfies first-order necessary conditions in the multi-objective context (see, e.g., Miettinen, 1999, p. 37–57). Since these methods seek a point in the Pareto set instead of a characterization of the entire Pareto set, we do not cover them in detail. However, we note that multiple points in the Pareto set can be found using random restarts, and some non-scalarization methods may adapt well to the stochastic context if gradient information is available. Non-scalarization methods include the steepest descent method of Fliege and Svaiter (2000), the Newton-like method of Fliege et al. (2009), and the multiple gradient descent algorithm (MGDA) of Désidéri (2012).

2.4 MOO Problem Testbeds

Deb et al. (2001) and Deb et al. (2002) present a testbed of MOO problems for assessing MOO algorithm performance. According to Deb et al. (2002), a good test problem should (1) be easy to construct, (2) be scalable in both the number of decision variables and number of objective functions, (3) have an efficient frontier whose location and shape are known exactly, and (4) have structural properties that make convergence to the efficient frontier and the even distribution of efficient points nontrivial. The test problems in Deb et al. (2001) include problems with non-convex efficient frontiers, multiple local Pareto optimal sets, and disconnected efficient frontiers. Such problems can be adapted for use in the MOSO context by adding noise to the objective functions.

3 Multi-Objective Simulation Optimization on Finite Sets

We now consider methods to solve the stochastic version of Problem $M$ where the cardinality of the feasible region, $|\mathcal{D}|$, is finite, the decision variables may be categorical, and the design points are known as systems. In particular, we require the number of systems to be “small enough” that we are able to sample from every system. Such problems are broadly known as ranking and selection (R&S) problems (Kim and Nelson, 2006); multi-objective R&S problems are also called MORS problems (Lee et al., 2010). MORS problems are unique among MOSO problems in the sense that they exist only in the stochastic setting. In the corresponding deterministic setting, we observe the objective vector for every system and return the true Pareto set. In the stochastic setting, we can return only an estimated Pareto set after expending some finite simulation budget $n$ to obtain estimates of the objective vectors. Since the Pareto set can only be estimated, it is possible that one or more of the systems will be incorrectly classified as Pareto or non-Pareto when a MORS solution method, or procedure, terminates.

As in Pasupathy and Ghosh (2013) for single-objective R&S, we sort the existing MORS solution methods into one of two categories: (1) those with the primary objective of providing a probabilistic guarantee on the returned solution $\hat{\mathcal{P}}$, and (2) those with the primary objective of guaranteeing sampling efficiency. To the best of our knowledge, the only method that arguably could be placed in the first category is Chen and Lee (2009). However, we do not cover Chen and Lee (2009) in detail, since it is based on the Enhanced Two-Stage Selection (ETSS) procedures of Chen and Kelton (2005) that do not provide a provable probabilistic guarantee. Instead, we discuss two MORS problem statements for future research on probabilistic guarantee methods and literature from the multivariate statistics community that may be relevant in creating future probabilistic guarantee procedures. The remainder of existing MORS algorithms fall exclusively into the second category of efficiency guarantee methods. We provide an overview of these methods, which include the popular Multiobjective Optimal Computing Budget Allocation (MOCBA) framework (Lee et al., 2010), the bi-objective Sampling Criteria for Optimization using Rate Estimators (SCORE) framework (Feld-
man and Hunter, 2016), and the Bayesian Myopic Multi-Objective Budget Allocation (M-MOBA) procedure (Branke and Zhang, 2015; Branke et al., 2016). We do not discuss the MO-MO²-TOS algorithm by Li et al. (2015b), since it is a multi-objective optimization algorithm for the context of multi-fidelity simulation models with biased, and not noisy, objective vector estimators (see §1.1).

Given that we consider a finite set of systems, in this section, we let the total number of systems be \( r \) and reserve the indices \( 1, \ldots, p \) for the Pareto systems. We refer to the true performance of system \( i \) as \( g(i) \), and the true performance of system \( i \) on objective \( k \) as \( g_k(i) \). We refer to the estimated performance of system \( i \) on objective \( k \) as \( \hat{G}_k(i, n) := (1/n) \sum_{\ell=1}^{n} G_{k\ell}(i) \) for random variables \( G_{k1}(i), \ldots, G_{kn}(i) \). Whenever it is reasonable to do so, we use the letter \( j \) to index the non-Pareto systems. Since we sample from every system, in this section, \( P \) and \( \hat{P} \) always refer to the true and estimated globally Pareto optimal sets, respectively.

### 3.1 Probabilistic Guarantee Methods

To begin our discussion of probabilistic guarantee methods for MORS, first, let us recall the statements of the probability of correct selection (PCS) and probability of good selection (PGS) guarantees from single-objective R&S. Let the true best system be system \( 1 \), the estimated best system after expending a total simulation budget \( n \) be \( \hat{1} \), and the objective of interest be objective \( 1 \). Then, given that system \( 1 \) is at least \( \delta > 0 \) better than the other systems, the PCS guarantee states that the probability the estimated best system \( \hat{1} \) is equal to the true best system is larger than a user-specified amount \( 1 - \alpha \) for \( \alpha \in (0, 1 - 1/r) \) (Kim and Nelson, 2006). In other words, a procedure with a PCS guarantee ensures \( \Pr\{\hat{1} = 1 \mid g_1(j) - g_1(1) > \delta \text{ for all } j = 2, \ldots, r\} \geq 1 - \alpha \). The requirement that system \( 1 \) is at least \( \delta \) better than the rest is called the PCS assumption. Ideally, a procedure would not require the PCS assumption and would simply guarantee \( \Pr\{\hat{1} = 1\} \geq 1 - \alpha \), as long as the best system is unique. Historically, procedures have not provided such a guarantee; only recently have Fan et al. (2016) proposed such a procedure. While PCS procedures put a guarantee on returning the best system, PGS procedures put a guarantee on returning a system within \( \delta > 0 \) of the best. That is, the PGS guarantee ensures \( \Pr\{g_1(\hat{1}) - g_1(1) \leq \delta\} \geq 1 - \alpha \). As noted in Ni et al. (2015), PGS procedures may be especially desirable when the number of systems is large and the performances of the best systems are close to each other. The parameter \( \delta \) is often called the indifference zone.

In the MORS context, we would like to develop procedures that provide some analogous form of PCS or PGS guarantee. Ideally, a procedure providing a PCS guarantee would ensure that the returned estimated Pareto set, \( \hat{P} \), would be equal to the true Pareto set \( P \) with probability greater than \( 1 - \alpha \), that is, \( \Pr\{\hat{P} = P\} \geq 1 - \alpha \). However, it is possible that MORS procedures would require an analogous PCS assumption, which would come from the proof of the statistical validity of the procedure. To formulate a PGS guarantee in the MORS context, we could take several routes, depending on the desired measure of difference between the true Pareto set and the estimated Pareto set. Using Hausdorff distance, one could attempt to create a procedure that guarantees, with probability greater than \( 1 - \alpha \), the Hausdorff distance between the set of true objective values of the systems in the estimated Pareto set, \( g(\hat{P}) \), and the true efficient set, \( E \), is less than \( \delta \), for some user-specified \( \delta > 0 \). That is, we could formulate the PGS guarantee as \( \Pr\{D(\hat{P}, E) < \delta\} \geq 1 - \alpha \). This formulation could be relaxed in various ways, such as by considering only the distance \( D(g(\hat{P}), E) \) between the sets \( g(\hat{P}) \) and \( E \) instead of the Hausdorff distance. The relaxed formulation yields no penalty for returning only a subset of the Pareto set, \( \hat{P} \subseteq P \), and may be useful if the decision-maker is indifferent to solutions within the Pareto set. Other metrics could also be used in place of Hausdorff distance, e.g., one could use the hypervolume difference measure.
of Branke et al. (2016), which we discuss in §3.2. Given the potential computational burden of correctly estimating a Pareto set, if the number of systems is large, procedures guaranteeing some form of PGS may be more practical than those attempting to provide a PCS guarantee.

In related multivariate statistics literature, Dudewicz and Taneja (1978) and Dudewicz and Taneja (1981) discuss two-stage procedures for selecting the best among several d-variate normal populations with unknown means and unknown, possibly unequal, covariance matrices. The best population is defined through a user-defined preference function \( h : \mathbb{R}^{dr} \rightarrow \{1, \ldots, r\} \). That is, letting \( g = (g(1), \ldots, g(r)) \) be a point in \( \mathbb{R}^{dr} \) containing the true mean values on each of \( d \) objectives for all \( r \) systems, then \( h(g) = i \) for some \( i \in \{1, \ldots, r\} \) if and only if the user prefers system \( i \) among all systems. The two-stage procedure guarantees correct selection of the user-preferred system with probability at least \( 1 - \alpha \) for any function \( h \) under the following PCS assumption. For each \( i = 1, \ldots, r \), let \( \mathcal{G}_i = \{ g : h(g) = i \} \) be a preference set containing configurations of the means in which system \( i \) is the preferred system. The preference sets \( \mathcal{G}_i, i \in \{1, \ldots, r\} \) partition \( \mathbb{R}^{kd} \). Let the distance from \( g \) to the boundary of \( \mathcal{G}_{h(g)} \) be \( d_B(g) = \inf \{ d(g, b) : b \notin \mathcal{G}_{h(g)} \} \). Then the required PCS assumption is \( d_B(g) \geq \delta \) for \( \delta > 0 \). To summarize, the two-stage procedure guarantees that the probability the true user-preferred system is selected as the best is greater than \( 1 - \alpha \) whenever the mean configuration \( g \) is at least Euclidean distance \( \delta \) from mean configurations where other systems are best. Subsequent work on the multivariate Behrens-Fisher problem and the Heteroscedastic Method (see, e.g., Dudewicz and Taneja, 1987, Dudewicz et al., 1991, and Anderson, 2003) may provide a foundation for designing procedures that provide PCS or PGS guarantees on the returned estimated Pareto set.

### §3.2 Efficiency Guarantee Methods

Efficiency guarantee methods do not provide a frequentist PCS or PGS guarantee on the returned solution \( \hat{\mathcal{P}} \). Instead, they usually offer some form of guarantee on the efficiency of the procedure. Frequently, this efficiency guarantee takes the form of optimizing some type of loss function that penalizes the misclassification (MC) of a system, although other loss functions exist. Define the MC event as \( \hat{\mathcal{P}} \neq \mathcal{P} \), which is the complement of the correct selection event, \( \mathcal{P} = \mathcal{P} \). Adopting the terminology of Hunter and McClosky (2016), there are two types of MC events: misclassification by inclusion (MCI) and misclassification by exclusion (MCE). MCI occurs if there exists a non-Pareto system \( j \in \mathcal{P}^c \) that is falsely estimated as Pareto, \( j \in \hat{\mathcal{P}} \). Thus MCI occurs whenever a non-Pareto system is falsely estimated as “better” than every other system on at least one objective. MCE occurs if there exists a Pareto system \( i \in \mathcal{P} \) that is estimated as non-Pareto, \( i \in \hat{\mathcal{P}}^c \). Thus MCE occurs if there exists a Pareto system that is falsely estimated as “worse” than at least one other system on every objective. As in Feldman et al. (2015) and Li (2012), the MCI and MCE events can be written as

\[
\text{MCI} = \bigcup_{j \in \mathcal{P}^c} \bigcap_{\ell=1}^{r} \bigcup_{k=1}^{d} \bar{G}_k(\ell, n\alpha_j) \leq \bar{G}_k(\ell, n\alpha_j), \quad \text{MCE} = \bigcup_{i \in \mathcal{P}} \bigcup_{\ell=1}^{r} \bigcap_{k=1}^{d} \bar{G}_k(\ell, n\alpha_i) \leq \bar{G}_k(\ell, n\alpha_i),
\]

respectively, where \( \alpha := (\alpha_1, \ldots, \alpha_r) \) is the proportion of the sampling budget allocated to each system, \( \sum_{i=1}^{r} \alpha_i = 1 \), \( \alpha_i > 0 \) for all \( i = 1, \ldots, r \). Notice that when considering the overall MC event, \( \text{MC} = \text{MCI} \cup \text{MCE} \), there is some overlap between the MCI and MCE events. Feldman...
(2017) eliminates this redundancy by writing the MC event as

\[ \text{MC} = \bigcup_{j \in P} \bigcap_{\ell \in P} G_k(j, n_{\alpha j}) \leq \bar{G}_k(\ell, n_{\alpha \ell}) \cup \bigcup_{i \in P} \bigcap_{\ell \in P} G_k(\ell, n_{\alpha \ell}) \leq \bar{G}_k(i, n_{\alpha i}). \]

same as MCI: a non-Pareto system falsely "beats" each Pareto system on at least one objective

one Pareto system falsely dominates, and thus excludes, another

In the following sections, we discuss the efficiency guarantee methods MOCBA, bi-objective SCORE, and M-MOBA. Since efficiency guarantee methods frequently derive a simulation budget allocation rule \( R \) by first assuming all required parameters are known, we present the rules for each method in terms of the true unknown parameters. However, these methods cannot be implemented as written, since the true unknown parameters of the rule \( R \) can only be estimated. The allocation rules we present can be implemented in a sequential sampling framework that broadly consists of the following steps: (1) sample \( n_0 > 2 \) initially from each system; (2) estimate the parameters of the sampling rule \( R \) and select the system(s) from which to take the next \( \Delta \geq 1 \) samples according to the rule, and (3) sample the required amount from the selected systems and, unless some termination criterion is met, go to Step (2). Each round of obtaining the next \( \Delta \) samples is frequently called a "stage" of sampling. The termination criterion is often that a predetermined total simulation budget \( n \) has been expended. We refer the reader to Lee et al. (2010), Feldman and Hunter (2016), and Branche and Zhang (2015), Braneke et al. (2016) for the implementation details of the sequential algorithms used in MOCBA, SCORE, and M-MOBA, respectively, and to Pasupathy and Ghosh (2013) for a generic implementation framework for efficiency guarantee methods in the case of single-objective R&S that can be extended to the MORS case.

3.2.1 MOCBA

We now discuss the popular MOCBA framework, which is a multi-objective version of the Optimal Computing Budget Allocation (OCBA) framework (Chen et al., 2000). MOCBA provides a heuristic simulation budget allocation rule that is derived by attempting to minimize the probability of MC. Because the probability of MC is difficult to minimize directly, under the assumptions that the random variables for each objective are independent and the sample means for each system are normally distributed, Lee et al. (2010) derive upper bounds on the MC event probabilities,

\[ \mathbb{P}\{\text{MCI}\} \leq ae_1 := \sum_{j \in P} \min_{\ell \neq j} \left[ 1 - \prod_{k=1}^d \mathbb{P}\{\bar{G}_k(\ell, n_{\alpha j}) \leq \bar{G}_k(j, n_{\alpha j})\} \right], \]

and

\[ \mathbb{P}\{\text{MCE}\} \leq ae_2 := \sum_{i \in P} \left( 1 - \prod_{\ell \neq i} \mathbb{P}\{\bar{G}_k(\ell, n_{\alpha \ell}) \leq \bar{G}_k(i, n_{\alpha i})\} \right), \]

where \( n_i := n_{\alpha i} \) for each \( i = 1, \ldots, r \). To avoid the difficulty of minimizing both probability bounds simultaneously, Lee et al. (2010) then formulate an optimization problem that minimizes the largest of the two bounds,

\[ \text{Find: } \arg\min_{(n_1, n_2, \ldots, n_r)} \max\{ae_1, ae_2\} \text{ s.t. } n_1 + n_2 + \ldots + n_r \leq n. \]

Two allocation rules result, depending on which bound is larger. We call these allocation rules the \( ae_1 \) rule and the \( ae_2 \) rule.

Before we present the \( ae_1 \) and \( ae_2 \) rules, we require additional concepts and notation. In the MOCBA framework, systems play one of two roles: dominating or dominated. In the \( ae_1 \) and \( ae_2 \) rules, the allocation to each system is determined by its role. Loosely speaking, a system \( j \) in a
dominated role receives a sampling allocation that is inversely proportional to its scaled squared “distance” from the system most likely to dominate it, where “distance” is the difference in objective values on the objective least likely to allow that system to dominate system \( j \). To write the allocations precisely, for any system \( j \), let \( \ell_j^* \) be the system most likely to dominate system \( j \). For any two systems \( i \) and \( j \), let \( k_i^j \) be the objective on which system \( i \) is least likely to dominate system \( j \). Then intuitively, if system \( \ell_j^* \) is the system most likely to dominate \( j \), then objective \( k_i^j \) is the objective that is most likely to prohibit system \( \ell_j^* \) from dominating system \( j \). An example configuration of systems is shown in Figure 3. The allocation to a system \( j' \) in a dominated role, relative to another system \( j \) in a dominated role, is

\[
\frac{\alpha_{j'}}{\alpha_j} = \left( \frac{(g_{k_{j'}^i}^j(j) - g_{k_{j'}^i}^j(\ell_j^*))^2}{\sigma_{k_{j'}^i}^2(j) + \sigma_{k_{j'}^i}^2(\ell_j^*)/\gamma_j} \right) / \left( \frac{(g_{k_{j'}^i}^{j'}(j') - g_{k_{j'}^i}^{j'}(\ell_j^*))^2}{\sigma_{k_{j'}^i}^2(j') + \sigma_{k_{j'}^i}^2(\ell_j^*)/\gamma_{j'}} \right),
\]

where \( \gamma_j = \alpha_{j'} / \alpha_j \) and \( \gamma_{j'} = \alpha_{j'} / \alpha_j \). Assuming that \( \ell_j^* \) and \( \ell_j^* \), are Pareto systems and \( j \) and \( j' \) are non-Pareto systems with \( \alpha_{\ell_j^*} \gg \alpha_j \) and \( \alpha_{\ell_j^*} \gg \alpha_{j'} \), then \( \gamma_j \) and \( \gamma_{j'} \) are large, and our previously stated intuition regarding these relative allocations holds approximately. We have written the relative allocations in equation (1) in this form to facilitate their comparison with the bi-objective SCORE allocations in the next section. To write the allocation to systems \( i \) in a dominating role, we require additional notation. First, let us denote the allocation to any system \( \ell \) as \( \alpha_{\ell} = \beta_{\ell} / \sum_{\ell' = 1}^{\ell} \beta_{\ell'} \), where for all systems \( j \) in a dominated role, the value of \( \beta_j \) can be obtained from equation (1). For systems \( i \) in a dominating role, we define

\[
\beta_i := \sqrt{\sum_{j \in \Omega_i} \left[ \frac{\sigma_{k_{j'}^i}^2(i)}{\sigma_{k_{j'}^i}^2(j)} \right] \beta_j^2},
\]

where the definition of \( \Omega_i \) is the set of all systems \( j \) playing a dominated role in which system \( i \) is the system most likely to dominate system \( j \). If \( \Omega_i \) is empty, then the allocation to system \( i \) is \( \alpha_i = 0 \). We are now ready to present the \( ae_1 \) and \( ae_2 \) rules.

**The \( ae_1 \) rule, to minimize the bound on \( P\{MCI\} \).** If \( ae_1 > ae_2 \), then we are most concerned that a non-Pareto system will be estimated as Pareto and falsely included in the estimated Pareto set. Therefore non-Pareto systems play the dominated role, and Pareto systems play the dominating role. In this context, from (2), we define \( \Omega_i := \{ j \in P : \ell_j^* = i \} \) as the set of all non-Pareto systems for which Pareto system \( i \) is the system most likely to dominate system \( j \).

**The \( ae_2 \) rule, to minimize the bound on \( P\{MCE\} \).** If \( ae_1 \leq ae_2 \), then we are most concerned that a Pareto system will be estimated as non-Pareto and falsely excluded from the estimated Pareto set.
Thus all non-Pareto systems play the dominating role, while some Pareto systems play the dominating role and others play the dominated role. The set of Pareto systems that play the dominated role, \( S_A \subseteq P \), is defined as

\[
S_A := \left\{ i \in P : \frac{(g_{k_i^2}(i) - g_{k_i^2}(\ell_i'))^2}{\sigma_{k_i^2}(i)/\alpha_i + \sigma_{k_i^2}(\ell_i')/\alpha_{\ell_i'}} < \min_{i' \in \Theta_i} \begin{cases} \frac{(g_{k_i}(i) - g_{k_i}(i'))^2}{\sigma_{k_i}(i)/\alpha_i + \sigma_{k_i}(i')/\alpha_{i'}} & \text{a simplified measure of likelihood that system } \ell_i' \text{ dominates Pareto } i \\ \frac{(g_{k_i'}(i) - g_{k_i'}(i'))^2}{\sigma_{k_i'}(i)/\alpha_i + \sigma_{k_i'}(i')/\alpha_{i'}} & \text{a simplified measure of likelihood that Pareto } i \text{ dominates Pareto } i' \end{cases} \right\},
\]

where \( \Theta_i := \{ i' \in P : \ell_i' = i \} \) is the set of all Pareto systems \( i' \) for which Pareto system \( i \) is the system most likely to dominate \( i' \). Then loosely speaking, set \( S_A \) is the set of all Pareto systems that are more likely to be dominated than to dominate another Pareto. Notice that \( S_A \) can be empty. The set of systems that play dominating roles, \( S_B \), is defined as \( S_B := (P \setminus S_A) \cup P^c \). Then in this context, from (2), \( \Omega_i := \Theta_i \cap S_A \) is the set of all Pareto systems \( i' \) in a dominated role for which Pareto system \( i \) is the system most likely to dominate system \( i' \).

**The MOCBA framework.** Lee et al. (2010) show that as the total simulation budget \( n \) tends to infinity, the \( ae_1 \) and \( ae_2 \) rules asymptotically minimize the \( ae_1 \) and \( ae_2 \) bounds, respectively. As mentioned previously, of course we do not know the values of the \( ae_1 \) and \( ae_2 \) bounds or of the parameters that comprise the \( ae_1 \) and \( ae_2 \) rules, such as the means and variances in equations (1) and (2). Thus the MOCBA framework consists of estimating the \( ae_1 \) and \( ae_2 \) bounds, selecting a sampling rule based on the largest estimated bound, and implementing an estimated version of the selected sampling rule. As mentioned previously, this process is usually embedded in a sequential sampling framework. We note that Teng et al. (2010) provide a modification to the MOCBA framework that allows the user to specify an indifference zone, which is the smallest difference in objective values the user cares to detect. This indifference zone approach prevents the algorithm from expending significant effort distinguishing between two systems whose performance difference is smaller than the indifference zone.

### 3.2.2 Bi-objective SCORE

While MOCBA is popular and can accommodate many objectives, when there are only two objectives, bi-objective SCORE provides a competing allocation framework that can simultaneously control the probability of MCI and MCE and accounts for correlation between the objectives.

To write the bi-objective SCORE allocations, we first require the concept of phantom Pareto systems, introduced by Hunter and McClosky (2016) and illustrated in Figure 4. Phantom Pareto systems are constructed from, and exist in relation to, the efficient set \( \mathcal{E} \). To define these systems, first label the Pareto systems from smallest to largest on objective 1, so that \( g_1(1) < g_1(2) < \ldots < g_1(p) \). Since we have only two objectives, this ordering implies that \( g_2(1) > g_2(2) > \ldots > g_2(p) \). Then the \( p + 1 \) phantom Pareto systems are defined as having objective values \((g_1(\ell + 1), g_2(\ell))\) for all Pareto systems \( \ell = 1, \ldots, p - 1 \), in addition to two phantom Pareto system placed at \((g_1(1), \infty)\) and \((\infty, g_2(p))\). Henceforth in this section, we index the phantom Pareto systems using the letter \( \ell \in \{0, \ldots, p\} \), and define \( g_1(p + 1) := \infty, g_2(0) := \infty \).

Using the phantom Pareto systems, the MCI event can be written as the event that a non-Pareto system is falsely estimated as dominating a phantom Pareto system, which enables the analysis of the probability of an MC event as the union of two MCE-like events. Based on the rate of decay of
the probability of misclassification, under a bivariate normality assumption and a certain asymptotic limit, Feldman and Hunter (2016) write the allocations to the non-Pareto systems as being inversely proportional to an intuitive measure called the score. For a non-Pareto system \( j \in P^c \), the score \( S_j \) is the minimum squared standardized “distance” from the non-Pareto system to the phantom Pareto systems, in the objective function space. That is,

\[
S_j := \min_{\ell \in \{0, \ldots, p\}} \left\{ \inf_{x \leq g_1(\ell+1), y \leq g_2(\ell)} \frac{1}{2(1-\rho_j^2)} \left\{ \frac{(x-g_1(j))^2}{\sigma_1^2(j)} - 2\rho_j(x-g_1(j))(y-g_2(j))\frac{\sigma_1^2(j)}{\sigma_2^2(j)} + \frac{(y-g_2(j))^2}{\sigma_2^2(j)} \right\} \right\},
\]

where \(-1 < \rho_j < 1\) is the correlation between the objectives for system \( j \). Then in the SCORE framework, the relative allocations between two non-Pareto systems \( j, j' \in P^c, j \neq j' \) is \( \alpha_{j'}/\alpha_j = S_j/S_{j'} \). This allocation can be compared with the relative allocations between systems in a dominated role in MOCBA, which appear in equation (1).

As in the original SCORE framework (Pasupathy et al., 2014), the bi-objective SCORE allocations are characterized by the relative allocations to the non-Pareto systems, and the allocations to the Pareto systems can be determined in a variety of ways. Feldman and Hunter (2016) suggest a method for solving for the allocation to the Pareto systems that controls the probabilities of both MCI and MCE events and requires solving a convex optimization problem whose complexity depends on the number of Pareto systems only. Therefore solving for the SCORE allocations is fast. We note that Feldman et al. (2015) provide preliminary work toward a multi-objective SCORE framework in more than two objectives; this research is ongoing.

3.2.3 M-MOBA

In this section, we discuss two bi-objective extensions of the small-sample Expected Value of Information procedures of Chick et al. (2010), M-MOBA and M-MOBA-HV. In the context of a sequential sampling framework and under a normality assumption, both M-MOBA and M-MOBA-HV adopt a Bayesian perspective to calculate the “best” system from which to obtain an additional \( \Delta \geq 1 \) samples in the next stage. In the case of M-MOBA, the best system is the one that has the highest probability of changing the observed Pareto set. In the case of M-MOBA-HV, the best system is the one that leads to the largest change in the observed expected hypervolume (HV), a measure that Branke et al. (2016) argue is more likely to be relevant for decision-makers and which we discuss below. Like MOCBA, M-MOBA and M-MOBA-HV do not account for correlation between the objectives in the allocation model.

Since the goal of MOCBA, bi-objective SCORE and M-MOBA is to efficiently allocate samples in such a way that the probability of a correct selection event is high, these algorithms are likely to expend a significant amount of samples attempting to distinguish between systems “near” the efficient frontier. The goal of M-MOBA-HV, however, is to allocate samples in such a way that the estimated expected HV is “close” to the true expected HV. Therefore M-MOBA-HV may expend more effort estimating the objective values of clearly Pareto systems than distinguishing the minor differences between Pareto and nearly-Pareto systems. If the decision-maker is likely to implement a clearly Pareto-optimal system, M-MOBA-HV may be preferable to M-MOBA.
Since we have already discussed the correct selection event in some detail, for brevity, we only describe the HV measure and M-MOBA-HV. To understand the HV measure, for a system $i \in \{1, \ldots, r\}$, first define the hyperrectangle $G(i, z_{ref}) := \{ g' \in \mathbb{R}^d : g(i) \preceq g' \preceq z_{ref} \}$, where $z_{ref}$ is a user-selected reference point. For ease of presentation, we assume the reference point equals the nadir, which is easily estimated in bi-objective R&S problems. Thus we let $G(i) := G(i, g_{nadir})$.

Then letting $\Lambda(\cdot)$ denote the Lebesgue measure, define the HV of a set of systems $A \subseteq \{1, \ldots, r\}$ as $HV(A) = \Lambda(\bigcup_{i \in A} G(i))$, which can be thought of as a measure of the “volume” of the set $g(A)$. Notice that $HV(\{1, \ldots, r\}) = HV(P)$, so that dominated systems do not contribute to the hypervolume calculation. To obtain a measure of change in the HV after expending the next $\Delta$ samples, Branke et al. (2016) propose the hypervolume difference (HVD). For sets $A \subseteq \{1, \ldots, r\}$ and $B \subseteq \{1, \ldots, r\}$, we define the HVD as

$$HVD(A, B) = HV(A) + HV(B) - 2\Lambda(\bigcup_{i \in A} G(i)) \cap \bigcup_{i \in B} G(i))$$

(Our definition of HVD is inferred from Branke et al., 2016, p. 861, Figure 2.) In M-MOBA-HV, for each system $i$, the expected HVD is calculated between the current observed Pareto set and the Pareto set that would result from obtaining the next $\Delta$ samples from system $i$. All $\Delta$ samples in the next stage are allocated to the system resulting in the largest expected HVD. Branke et al. (2016) show that M-MOBA and M-MOBA-HV may result in different allocation rules.

3.3 Applications of MOSO on Finite Sets

As in the single-objective R&S context, the generality of the MORS problem formulation allows it to be applied in a variety of contexts. First, MORS algorithms can be applied directly to problems with categorical variables. Second, MORS algorithms can be used for efficient allocation of samples within other search algorithms.

In the first category, Hunter and McClosky (2016) derive and implement an optimal allocation framework to solve a bi-objective R&S problem in the context of a plant breeding application. The two objectives of this problem are the mean and variance of a normal distributional family, which implies that the random variables corresponding to the first objective are normally distributed, while the random variables corresponding to the second objective belong to the chi-squared family. Hunter and McClosky (2016) provide an optimal allocation framework for mean-variance criteria in which the objectives are independent, and show that the framework performs well numerically in the context of plant breeding. Since the optimal allocation framework of Hunter and McClosky (2016) is similar to that of bi-objective SCORE, we do not cover its methodology in detail, but instead note its relevance to the plant breeding application.

In the second category of MORS methods used within other search algorithms, MOCBA is frequently used for identifying non-dominated solutions within meta-heuristic algorithms. For example, Lee et al. (2008) use MOCBA in an evolutionary algorithm context to allocate aircraft spare parts. Chew et al. (2009) also use nested partitions [see, e.g., Ólafsson, 2006] together with MOCBA to solve an inventory problem. We also briefly discuss MOCBA’s use inside MOEAs in §A.

4 Continuous Multi-Objective Simulation Optimization

In this section, we consider continuous MOSO problems, that is, MOSO problems with continuous decision variables. Since the Pareto set and the efficient set may be uncountable in this case, methods to solve Problem $M$ (see §1.1) implemented with a finite simulation budget $n$ usually identify only a
countable number of estimated Pareto points that, ideally, provide an accurate representation of the efficient set in the objective function space. However as the simulation budget \( n \) tends to infinity, we would like MOSO methods to provide some form of probabilistic guarantee on convergence of the returned solution to a local or global Pareto set.

As noted in the literature review by Rosen et al. (2008), at the time, no provably convergent algorithms existed for finding the set of Pareto optimal solutions, \( \mathcal{P} \), as the solution to a continuous MOSO problem. Indeed, the work on theory, methods, and algorithms for continuous MOSO problems has been recent. The existing methods fall into one of two broad categories: sample average approximation (SAA) methods and probabilistic branch and bound (PBNB) methods. Popular in the SOSO context, SAA methods provide a way of estimating the Pareto and efficient sets by solving the sample-path problem for a fixed sampling budget \( n \) at each point. We discuss multi-objective SAA methods in §4.1. PBNB methods provide a general framework for optimization by sampling uniformly in sub-regions of the bounded feasible space, and pruning undesirable regions. Huang and Zabinsky (2014) develop a multi-objective PBNB algorithm called MOPBNB that works on continuous, integer-ordered, or mixed spaces that are bounded. We discuss MOPBNB in §4.2.

We acknowledge two papers that we do not cover in detail, but that the reader nevertheless may find interesting: Fliege and Werner (2014), which provides a robust formulation of the MOSO problem, and Engau (2016), which provides results that may be useful for the development of future MOSO theory and methods.

4.1 Sample Average Approximation for Continuous MOSO

SAA can be used in the multi-objective context by finding the sample-path solution to the multi-objective sample-path problem (see §1.1)

\[
\text{Problem } \hat{M} : \min_{x \in D} \{ \hat{G}(x, n) = (\hat{G}_1(x, n), \ldots, \hat{G}(x, n)) := (\frac{1}{n} \sum_{i=1}^{n} G_1(x, \xi_i), \ldots, \frac{1}{n} \sum_{i=1}^{n} G_d(x, \xi_i)) \}.
\]

As noted in Fliege and Xu (2011) and in the literature review by Gutjahr and Pichler (2016), an estimated solution to the MOSO problem can be found by realizing the randomness in Problem \( \hat{M} \) and solving the resulting deterministic MOO problem. Since prominent MOO methods use scalarization, in addition to generic consistency results based on the formulation of Problem \( \hat{M} \), it is also desirable to obtain consistency results on scalarized versions of the sample-path problem. In this section, we summarize several important convergence results from Fliege and Xu (2011), Kim and Ryu (2011a), and Bonnel and Collonge (2014) for the general MOSO formulation and relevant scalarizations. We also briefly discuss algorithms for solving MOSO problems using SAA. We refer the reader to the original papers for details and proofs, and to Shapiro (2003) and Shapiro et al. (2009) for analogous results in the SOSO context.

Since the simulation budget expended to estimate the objective vector at each decision point is \( n \), in this section, we use \( \hat{\mathcal{P}}_n \) to denote the estimated global solution and \( \hat{\mathcal{E}}_n \) to denote the estimated image of \( \hat{\mathcal{P}}_n \). We use \( \hat{\mathcal{P}}_{nw} \) to denote the estimated global weakly Pareto optimal set (see Definition 4 for the definition of the weakly Pareto optimal points).

4.1.1 General Convergence Results

Under appropriate regularity conditions, as the simulation budget \( n \) increases to infinity, ideally, some metric of difference between the sample-path solution to Problem \( \hat{M} \) and the true solution to
Problem $M$ converges to zero with probability one. Bonnel and Collonge (2014) provide regularity conditions and convergence results regarding the Hausdorff distance between the estimated weakly Pareto set and the true weakly Pareto set, as well as the Hausdorff distance between the estimated efficient set and the true efficient set. In this section, we present the key results of Bonnel and Collonge (2014). To begin, we present the required assumptions; some subset of these assumptions will be needed for each result we present.

**Assumption 1** (Bonnel and Collonge, 2014, p. 411). We assume one or more of the following:

1. $\mathcal{D}$ is a nonempty and compact subset of $\mathbb{R}^d$;
2. the random variables $\xi_i, i = 1, 2, \ldots$ are i.i.d. with $P\{\xi_i \in \Xi\} = 1$ for all $i = 1, 2, \ldots$;
3. for each objective $k = 1, \ldots, d$, the function $G_k(x, \xi)$ is finite-valued and continuous on $\mathcal{D}$ for a.e. $\xi \in \Xi$.
4. for each objective $k = 1, \ldots, d$, the function $G_k(x, \xi)$ is dominated by an integrable function $r_k$, that is, $E_{\xi}[r_k(\cdot)] < \infty$ and $|G_k(x, \xi)| \leq r_k(\xi)$ for all $x \in \mathcal{D}$ and a.e. $\xi \in \Xi$;
5. $\mathcal{D}$ is convex.
6. for each objective $k = 1, \ldots, d$, $G_k(x, \xi)$ is strictly convex on $\mathcal{D}$ for a.e. $\xi \in \Xi$.

Recall that under Assumptions 1(1)–1(4), for each objective $k \in \{1, \ldots, d\}$, $g_k(x)$ is finite-valued and continuous on $\mathcal{D}$, and the uniform LLN holds (Shapiro, 2003, p. 363, Proposition 7). Recall that the uniform LLN states that a sequence of functions $\tilde{G}(x, n)$ converges to $g(x)$ w.p.1. as $n \to \infty$, uniformly on $x \in \mathcal{D}$ if and only if $P\{\lim_{n\to\infty} \sup_{x \in \mathcal{D}} |\tilde{G}(x, n) - g(x)| = 0\} = 1$.

By the uniform LLN, the following result regarding the maximum Euclidean distance between the estimator $\tilde{G}(x, n)$ and the objective vector $g(x)$ holds (see Bonnel and Collonge, 2014, p. 412 and Fliege and Xu, 2011, p. 151).

**Proposition 1.** Under Assumptions I(1)–I(4),

$$P\{\lim_{n \to \infty} \sup_{x \in \mathcal{D}} ||\tilde{G}(x, n) - g(x)|| = 0\} = 1.$$ 

Thus, under mild conditions, a version of the uniform LLN applies to the vector-valued estimated objective function.

Under equally mild conditions, Bonnel and Collonge (2014) show that, as $n$ goes to infinity, the Hausdorff distance between the estimated efficient frontier $\hat{\mathcal{E}}_n$ and the true efficient frontier $\mathcal{E}$ goes to zero w.p.1, and the distance between the estimated weakly Pareto set and the true weakly Pareto set goes to zero w.p.1.

**Theorem 1** (Bonnel and Collonge, 2014, p. 417, 421). Under Assumptions I(1)–I(4), the following hold:

1. the Hausdorff distance between the estimated efficient set $\hat{\mathcal{E}}_n$ and the true efficient set $\mathcal{E}$ goes to zero w.p.I as $n \to \infty$, that is, $P\{\lim_{n \to \infty} \Pi(\hat{\mathcal{E}}_n, \mathcal{E}) = 0\} = 1$;
2. the distance between the estimated weakly Pareto set $\hat{\mathcal{P}}_n^w$ and the true weakly Pareto set $\mathcal{P}^w$ goes to zero w.p.I as $n \to \infty$, that is, $P\{\lim_{n \to \infty} \mathcal{D}(\hat{\mathcal{P}}_n^w, \mathcal{P}^w) = 0\} = 1$.

To ensure there are no weakly Pareto points “missed,” such that the symmetric distance between the estimated weakly Pareto set and the true weakly Pareto set also converges to zero w.p.1, we require the additional assumptions of a convex feasible region $\mathcal{D}$ and strict convexity of each function $G_k(x, \xi)$ on $\mathcal{D}$ for a.e. $\xi \in \Xi$. Recall that if Problem $M$ is convex, then the local Pareto set is equal to the global Pareto set.
Theorem 2 (Bonnel and Collonge, 2014, p. 417). Under Assumptions I(1)–I(6), the Hausdorff distance between the estimated weakly Pareto set $\hat{P}_w^n$ and the true weakly Pareto set $P_w$ goes to zero w.p.1 as $n \to \infty$, that is,

$$P\{\lim_{n \to \infty} \mathbb{H}(\hat{P}_w^n, P_w) = 0\} = 1.$$ 

Thus Bonnel and Collonge (2014) provide consistency results for SAA estimators in the multi-objective context. We note here that Kim and Ryu (2011a) discuss the uniform convergence of gradient estimates in the context of SAA for MOSO, and Fliege and Xu (2011) provide some convergence rate results.

4.1.2 Convergence under Scalarization

Since prominent MOO methods use scalarization, it is also desirable to obtain consistency results on scalarized SAA sample-path functions. We now point the reader to several results regarding consistency under scalarization. First, Bonnel and Collonge (2014) use the linear weighted sum method to prove consistency of the estimated weakly Pareto set (presented in Theorem 2), and in doing so, provide some consistency results for the corresponding linear weighted scalarized problem under Assumptions 1(1)–1(6). Fliege and Xu (2011) propose and analyze the properties of a new smoothing Chebyshev scalarization, previously defined in §2.2.2. The smoothing Chebyshev sample-path problem is

$$\text{Problem } \hat{S}_\text{smooth}^\infty(w, t): \text{ minimize } \min_{x \in D} p\left((w_1(\bar{G}_1(x, n) - z^*_1), \ldots, w_d(\bar{G}_d(x, n) - z^*_d)), t\right),$$

where the function $p(y, t) := t \log \sum_{k=1}^d e^{y_k/t}$ for parameter $t > 0$ and $p(y, 0) := 0$, $w \in \mathbb{R}^d$ is an appropriately defined weight vector, and $z^* \in \mathbb{R}^d$ is a reference vector. Finally, Kim and Ryu (2011a) investigate the convergence properties of the estimated Pareto and efficient sets under the single-objective product formulation by Audet et al. (2008), defined in §2.2.4. An SAA sample-path problem for this scalarization is

$$\text{Problem } \hat{S}_\text{prod}(r): \text{ minimize } \hat{\phi}(x; r) := \frac{1}{n} \sum_{i=1}^n \left[ - \prod_{k=1}^d \left( \max\{r_k - G_k(x, \xi_i), 0\} \right)^2 \right].$$

Under regularity conditions including Assumptions 1(1)–1(4), Kim and Ryu (2011a) show that the uniform LLN holds for $\hat{\phi}(\cdot, r)$.

4.1.3 Algorithms

We note two SAA algorithms for MOSO, both of which are designed for the bi-objective case. First, Kim and Ryu (2011a) provide an SAA algorithm adapted from the deterministic bi-objective trust-region method in Ryu and Kim (2014), a version of which also appears in Kim and Ryu (2011b). This algorithm broadly iterates through the following steps: (1) select the most “isolated” point $x$ in a set of solutions, (2) create surrogate models for each objective and the scalarized single-objective product formulation $\hat{\phi}(\cdot, r)$ in the trust region around $x$, (3) solve the resulting $d + 1$ trust-region sub-problems, (4) and keep “good” points according to “sufficient reduction.” Under certain regularity conditions, the set of solutions produced by the SAA algorithm in Kim and Ryu (2011a) converges into a set of locally Pareto optimal solutions w.p.1. as $n \to \infty$. Second, Bonnel and Collonge (2015) present an algorithm for a convex bi-objective SO problem that uses SAA. Since the goal of this algorithm is to minimize a function of the objectives and return a single solution, we do not go into detail. Thus to our knowledge, Kim and Ryu (2011a) is the only available SAA algorithm for continuous bi-objective SO that seeks multiple Pareto points.
4.2 **Multiple Objective Probabilistic Branch and Bound (MOPBnB)**

As we saw in the previous section, SAA methods require some regularity conditions on the objective functions for convergence. When far less is known about the objective functions, or the search space contains both continuous and integer-ordered variables, we may require a more general solution method.

PBnB algorithms provide a more general framework for optimization. First developed by Zabinsky et al. (2011), these algorithms first sample uniformly in sub-regions of the appropriately bounded feasible space. The algorithm progresses by choosing the sub-regions containing the estimated “best” points for further division and sampling, and pruning regions containing only the estimated “worst” points. MOPBnB is a multi-objective version of the PBnB algorithm in which the retained regions in each iteration contain points that are estimated as being near the efficient frontier. The pruned regions contain only estimated non-Pareto points.

Like R&S methods, assuming the function observations at each point have a normal distribution with a known variance structure, MOPBnB provides a probabilistic guarantee on the returned estimated Pareto set. Specifically, MOPBnB guarantees that at each iteration, the estimated Pareto set contains points that lie within an appropriately relaxed true Pareto set with high probability. MOPBnB further guarantees that as the algorithm progresses, with high probability, the proportion of “good” solutions in the estimated Pareto set increases.

4.3 **Applications of Continuous MOSO Methods**

MOSO problems with continuous or mixed decision variables arise in a variety of contexts. Kim (2014) states that an oil drilling application motivated the algorithm in Kim and Ryu (2011a). In Fliege and Xu (2011, p. 158), the authors use multi-objective SAA to solve a stochastic collaborative game applied to the energy market. The objective of each player is to maximize their own profit. In Huang (2016, p. 90), MOPBnB is used to allocate portable ultrasound machines across orthopedic clinics. The goal is to find an allocation of ultrasound machines that minimizes both the expected cost and an expected health utility loss. Also, in Arora and Ventresca (2017), MOPBnB is used to find an action-based model for a given target network.

5 **Integer-Ordered Multi-Objective Simulation Optimization**

In this section, we consider solving Problem $M$ when the decision variable space is integer-ordered, $\mathcal{D} \subseteq \mathbb{Z}^q$. Thus the Pareto set $\mathcal{P}$ and the efficient set $\mathcal{E}$ are countable. While the definition of a global Pareto set remains clear in this context, the notion of a “local Pareto set” is less clear. Before we discuss methods for solving such problems, we first define the notion of a local Pareto set when the decision space is integer-ordered.

Following the lead of Wang et al. (2013), who define a flexible neighborhood structure for local optimality in the context of integer-ordered SOSO, we define a flexible neighborhood structure for “locally” Pareto optimal sets. First, define a neighborhood of size $a$ around the point $x$ as follows.

**Definition 5.** For $x \in \mathcal{D} \subseteq \mathbb{Z}^q$ and $a \geq 0$, the neighborhood $B_a(x) \subseteq \mathbb{Z}^q$ is

$$B_a(x) := \{x' \in \mathbb{Z}^q : ||x - x'|| \leq a\}.$$  

Then recall from Definition 3 that a vector $x^* \in \mathcal{D}$ is locally Pareto optimal if there does not exist a point $x \in B_a(x^*)$ such that $g(x) \preceq g(x^*)$. We wish to collect up locally Pareto optimal points
“near” each other into a set. Thus we require the notion of a neighborhood of size \( a \) around a set \( S \), which we define as follows.

**Definition 6.** For \( S \subseteq D \subseteq \mathbb{Z}^d \) and \( a \geq 0 \), the neighborhood \( N_a(S) \subseteq \mathbb{Z}^d \) is

\[
N_a(S) := \{ x \in \mathbb{Z}^d : d(x, S) \leq a \} = \bigcup_{x \in S} B_a(x).
\]

Now we define a \( N_a \)-local Pareto set as a set \( \mathcal{P}_a \) in which none of the points in \( \mathcal{P}_a \) dominate other points in \( \mathcal{P}_a \), and all points within a neighborhood of \( \mathcal{P}_a \), but not in \( \mathcal{P}_a \), are dominated by at least one point in \( \mathcal{P}_a \) (see Li et al., 2015a, p. 1155, who define the \( N_1 \)-local Pareto set).

**Definition 7.** A set \( \mathcal{P}_a \) is an \( N_a \)-local Pareto set if for all \( x^* \in \mathcal{P}_a \), there does not exist \( x \in N_a(\mathcal{P}_a) \cap D \) such that \( g(x) \preceq g(x^*) \), and for all \( x \in (N_a(\mathcal{P}_a) \setminus \mathcal{P}_a) \cap D \), there exists \( x^* \in \mathcal{P}_a \) such that \( g(x^*) \preceq g(x) \).

Notice that this definition is general in the sense that if the distance metric \( a \) is large enough, we retrieve the global Pareto-optimal set.

In this section, we consider algorithms that converge to an \( N_1 \)-local Pareto set w.p.1. To the best of our knowledge, only one such algorithm exists, MO-COMPASS (Li et al., 2015a). In §5.2, we discuss applications of MOSO algorithms on integer-ordered spaces.

### 5.1 MO-COMPASS

MO-COMPASS is a multi-objective version of the popular COMPASS algorithm for integer-ordered SOSO problems (Hong and Nelson, 2006; Xu et al., 2010). Like COMPASS, MO-COMPASS finds \( N_1 \)-local Pareto solutions to integer-ordered MOSO problems using a neighborhood structure called the most promising region (MPR). In the multi-objective context, the MPR is defined as the collection of decision points that are closer to the current estimated \( N_1 \)-local Pareto set than to any previously-visited estimated non-Pareto point. At iteration \( \nu \), it is constructed by taking the union over the MPRs corresponding to each point in the estimated Pareto set \( X^* \in \hat{\mathcal{P}}_\nu \). Formally, for a bounded feasible set \( D \), the MPR is

\[
C_\nu := \bigcup_{x^* \in \hat{\mathcal{P}}_\nu} \{ x \in D : \text{for all } x' \in \mathcal{V}_\nu \setminus \hat{\mathcal{P}}_\nu, \|x - X^*\| \leq \|x - x'\| \},
\]

where \( \mathcal{V}_\nu \) is the set of all feasible points visited so far. In each iteration, MO-COMPASS uniformly selects new candidate points from the MPR, applies a simulation allocation rule (SAR) to determine how many samples to obtain from each point visited so far, estimates the objective values of each point, updates the estimated local Pareto set, and updates the MPR. Thus the MPR is adaptive, converging w.p.1 to an \( N_1 \)-local Pareto set as candidate points are evaluated. MO-COMPASS may be terminated when a pre-specified simulation budget is exhausted or some other criteria are met.

Like COMPASS, versions of MO-COMPASS exist for the case of a fully constrained, partially constrained, or unconstrained feasible region \( D \). For simplicity, we consider only the fully constrained, MO-COMPASS algorithm in the following Algorithm 1 and refer the reader to Li et al. (2015a) for details of the other cases.

Under mild conditions, the estimated Pareto set \( \hat{\mathcal{P}}_\nu \) returned by the fully-constrained MO-COMPASS algorithm converges to a \( N_1 \)-local Pareto set w.p.1, in the sense that \( \mathbb{P}\{\text{FS}(\hat{\mathcal{P}}_\nu, N_1(\mathcal{P}_\nu)) \text{ i.o.} \} = 0 \), where \( \text{FS}(\hat{\mathcal{P}}_\nu, N_1(\mathcal{P}_\nu)) \) is the event that the observed estimated Pareto set \( \hat{\mathcal{P}}_\nu \) is not equal to the Pareto set across the points in an \( N_1 \)-neighborhood of \( \mathcal{P}_\nu \). The conditions for this convergence are as follows.
Algorithm 1: MO-COMPASS for fully-constrained $\mathcal{D}$

**Require:** simulation allocation rule SAR; number of points to visit $m$ at each iteration;

**Initialize:** iteration $\nu = 0$, set of visited points $\mathcal{V}_0 = \emptyset$, and most promising region $\mathcal{C}_0 = \mathcal{D}$

**while** not stopped **do**
  
  Set $\nu = \nu + 1$.

  Uniformly and independently sample $X_\nu = \{x_{\nu1}, \ldots, x_{\nu m}\}$ from $C_{\nu-1}$.

  Update the set of visited points $\mathcal{V}_\nu = \mathcal{V}_{\nu-1} \cup X_\nu$.

  **for all** $x \in \mathcal{V}_\nu$ **do**
    
    Apply the SAR to determine the amount to sample $a_\nu(x)$.

    Take $a_\nu(x)$ observations at $x$.

    Update $n_\nu(x) = \sum_{i=1}^{\nu} a_i(x)$ and the objective vector estimator $\hat{G}(x, n_\nu(x))$.
  
  **end**

  Update $\hat{P}_\nu = \{X^* \in \mathcal{V}_\nu : \exists x \in \mathcal{V}_\nu \ni G(x, n_\nu(x)) \leq G(X^*, n_\nu(X^*))\}$.

  Update $\mathcal{C}_\nu = \bigcup_{X^* \in \hat{P}_\nu} \{x \in \mathcal{D} : \text{for all } x' \in \mathcal{V}_\nu \setminus \hat{P}_\nu, ||x - X^*|| \leq ||x - x'||\}$.

**end**

1. No two points have exactly the same performance on any objective, that is, there exists a $\delta > 0$ such that for all $x, x' \in \mathcal{D}$ and for all objectives $k \in \{1, \ldots, d\}$, $x \neq x'$ implies $|g_k(x) - g_k(x')| \geq \delta$.

2. For every $x \in \mathcal{D}$ and objective $k \in \{1, \ldots, d\}$, $\hat{G}_k(x, n) \rightarrow g_k(x)$ w.p.1 as $n \rightarrow \infty$.

3. The SAR guarantees that we obtain at least one simulation observation at all the newly-sampled points in each iteration, that is, $a_\nu(x) \geq 1$ if $x$ is a newly-visited point at iteration $\nu$; and each visited point is sampled infinitely often, so that $\lim_{\nu \rightarrow \infty} n_\nu(x) = \infty$ for all $x \in \bigcup_{\nu=0}^{\infty} \mathcal{V}_\nu$.

Note that MO-COMPASS and the conditions for its convergence allow the use of common random numbers. The convergence of MO-COMPASS for the partially constrained and unconstrained cases requires additional conditions.

### 5.2 Applications of Integer-Ordered MOSO Methods

Notably, MO-COMPASS has been incorporated into two software frameworks. The first is D-SIMPAIR (Li et al., 2015a), which is a decision support system for inventory management. The second software framework is Object-Oriented Discrete Event Simulation (O2DES), described in Li et al. (2015). Both papers provide examples of finding optimal aircraft spare part inventory policies in which the two objectives are expected service level and expected cost. Li et al. (2016) also uses MO-COMPASS in the context of an inventory management problem. The goal of this problem is to find optimal inventory policies that minimize the expected total cumulated inventory, while also minimizing the expected cumulated backlog.

### 6 Concluding Remarks

As may be apparent from our discussion of existing theory, methods, and algorithms for MOSO problems, a tremendous amount of work remains in this area. We make the following remarks about important open questions and a future research agenda.

1. Bi-objective SO problems have special structure and may provide a good starting point for theoretical and algorithmic development. Specifically, in the bi-objective context, ordering...
the Pareto points on one objective implies their ordering on the second, any correlation between the objectives at a decision point \( x \) is easy to express as a single number \( \rho(x) \), and the nadir vector can be constructed from the ideal vector. Thus bi-objective SO methods may require less complexity than algorithms for higher dimensions. Further, bi-objective problems are an important class of problems for application areas (see, e.g., Hunter and McClosky, 2016), and thus methods developed for this class of problems are likely to be useful.

2. Probabilistic guarantee methods for MORS are notably absent from the literature. Given the importance, prominence, and popularity of single-objective R&S methods that provide a probabilistic guarantee, we believe MORS methods with a probabilistic guarantee will be an important class of algorithms within MOSO methods.

3. To fully harness the power of existing SOSO algorithms, common MOO scalarization techniques should be adapted to the MOSO context. While there has been some adaptation seen in, e.g., Kim and Ryu (2011a), Fliege and Xu (2011), and Bonnel and Collonge (2015), a significant amount of work remains in this area. As a notable example, the U.S. Department of Energy is using the \( \varepsilon \)-constraint method to solve MOSO problems in the context of the Advanced Process Engineering Co-Simulator (Subramanyan et al., 2011). However to the best of our knowledge, apart from the stochastically constrained SO methods in Table 1, there is no theoretical or methodological development for this scalarization technique in the MOSO context.

4. Because MOSO problems are may be more computationally intensive than MOO problems, adaptive methods to find evenly-distributed points in the Pareto frontier are likely to be crucial for solving MOSO problems efficiently. (See §2.2.5 for a few references to such methods in the MOO context.)

APPENDIX

A Multi-Objective Evolutionary Algorithms

To understand how MOEAs are used to solve MOSO problems, we first describe how they are used to solve deterministic MOO problems. In the deterministic context, evolutionary algorithms are among the most widely used methods for solving MOO problems (Coello Coello et al., 2007a). MOEAs are easy to implement and are flexible in the sense that they adapt well to different problem types, including problems with integer-ordered, continuous, or mixed variables (see, e.g., Brockhoff, 2011 and Deb, 2009).

The idea underlying MOEAs is to constantly improve, in terms of objective function value, a population of decision points using competition under the influence of “natural selection.” The general procedure for an MOEA involves repeated use of the three main evolutionary operators: selection of “parent” decision points, recombination to produce “child” decision points, and mutation. In the selection step, a “fitness” measure based on the objective functions is used to evaluate the population of potential parents. This step mimics natural selection, in the sense that “better” parents are more likely to be selected to produce children. In the MOO context, a single fitness value called transformed fitness is usually created by transforming the vector of objective function values into a scalar (Coello Coello et al., 2007b). After selection, recombination is applied to two or more selected decision points, producing one or more new “child” decision points. Mutation may then be applied to one or more decision points, each time resulting in a new decision point. The new population’s transformed fitness is then evaluated, and decision points are selected to be kept in the next
iteration. The entire process is repeated until a termination condition is met. Since this algorithm uses the recombination and mutation operators to explore the search space, it can be used when little is known about the problem structure. The success of MOEAs has also inspired researchers to develop multi-objective versions of other well known single-objective non-evolutionary methods such as simulated annealing, ant colony optimization, and particle swarm optimization (Coello Coello et al., 2007a).

Though MOEAs have been the popular choice for MOO problems, successful implementation is dependent on multiple algorithm parameters. These parameters include the size of the population, the number of generations, the mutation probability, and the type of recombination used. Deb and Agrawal (1999) study the complex interactions between these parameters and how their selection is dependent on the problem being solved. Brockhoff (2011) and Wegener (2003) emphasize that there is a lack of theory to support convergence of such algorithms, rigorous runtime analyses, and quality of the obtained set of solutions.

To use MOEAs in a MOSO context, during the transformed fitness evaluation step, a simulation oracle is queried with some pre-specified simulation budget. For example, Lee et al. (2008) use MOEA as a search procedure, and use MOCBA (see §3) to efficiently allocate simulation replications to decision points in the population. MOCBA is also used to identify the non-dominated set of solutions in the current population. Applications in which MOEAs have been used to solve MOSO problems include Baesler and Sepulveda (2000), Baesler and Sepulveda (2001), and Lee et al. (2007).

Acknowledgments

This paper began in the special topics class IE 690 Simulation Optimization at Purdue University in Fall 2016. The authors thank class members Zibo Zhao and Zhouyang Lou for participating in early drafts of §2 and §3, respectively, and for commenting on draft versions of this paper created during the course. The authors also thank David Eckman and Mario Ventresca for commenting on draft versions of the current paper. The work of S. R. Hunter and E. A. Applegate was supported in part by the National Science Foundation, under grant CMMI-1554144. The work of V. Arora was supported in part by the National Science Foundation, under grant CMMI-1549608.

References


G. Feldman and S. R. Hunter. 2016. SCORE allocations for bi-objective ranking and selection. *Opti-


J. Luo, L. J. Hong, B. L. Nelson, and Y. Wu. 2015. Fully sequential procedures for large-scale ranking-and-


