Multi-objective Ranking and Selection: Optimal Sampling Laws and Tractable Approximations via SCORE

Eric A. Applegate[1], Guy Feldman[2], Susan R. Hunter[1], and Raghu Pasupathy[2]

[1] *School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA*
[2] *Department of Statistics, Purdue University, West Lafayette, IN 47907, USA*

Abstract

Consider the context of selecting a set of Pareto-optimal systems from a finite set of systems based on three or more stochastic objectives. We characterize the exact asymptotically optimal sample allocation that maximizes the rate of decay of the probability of a misclassification event, and we provide a multi-objective Sampling Criteria for Optimization using Rate Estimators (SCORE) allocation for use when the number of non-Pareto systems is large relative to the number of Pareto systems. The SCORE allocation has three salient features: (a) it simultaneously controls the probabilities of misclassification by exclusion and inclusion; (b) it exploits phantom Pareto systems for computational efficiency, which we find using a dimension-sweep algorithm; and (c) it models dependence between the objectives. The SCORE allocation is fast and accurate for problems with three objectives or a small number of systems. For problems with four or more objectives and a large number of systems, where modeling dependence has diminishing returns relative to computational speed, we propose independent SCORE (iSCORE). Our numerical experience is promising: SCORE and iSCORE successfully solve MORS problems involving several thousand systems in three and four objectives.

Key words: multi-objective, ranking and selection, simulation optimization

1 Introduction

We consider *multi-objective ranking and selection* (MORS), in which a decision-maker wishes to select the set of "best" systems among a finite set of r systems whose expected performances can only be observed with stochastic error. A *system* refers to one of the r decision variable configurations under consideration. Each system's performance is assessed on the basis of a d-dimensional vector of objectives defined implicitly, for example, through a Monte Carlo simulation model that is capable of generating unbiased estimates of the objective vector. The solution to the MORS problem is the *Pareto set*, that is, the set of all *non-dominated* systems. A system is non-dominated if no other system is at least as good on all objectives and strictly better on at least one objective.

MORS problems arise abundantly when designing stochastic systems. Diverse examples of applications include plant breeding [Hunter and McClosky, 2016], earthmoving operations [Zhang, 2008], and supply chain management [Ding et al., 2006]; see Hunter et al. [2018] for additional application areas. In fact, a widely held viewpoint is that a substantial fraction of optimization problems in

Preprint compiled Sunday 29th July, 2018 at 4:07pm. Corresponding author: susanhunter@purdue.edu.

the "real world" involve more than one competing objective, and multi-objective optimization to identify a Pareto set is a fruitful and disciplined way to handle such contexts [Eichfelder, 2008].

Despite their widespread occurrence, and in contrast to single-objective ranking and selection (R&S) which has a long history of development [Fu and Henderson, 2017], MORS problems have received relatively little attention in the literature to date (see Table 1 and §2). The MOCBA framework [Lee et al., 2010] is the most well-known and, to our knowledge, one of only a few known *fixed-budget procedures* (see §2.1) designed to estimate the entire Pareto set when the number of objectives is three or greater. MOCBA is an insightful algorithm that performs well in a variety of MORS problem instances; however, a number of important MORS questions remain unresolved, leaving room for designing algorithms that improve on MOCBA.

Table 1: Key fixed-budget MORS procedures are classified by their contributions.

Dependence	$d = 2$ Stochastic Objectives	$d \geq 2$ Stochastic Objectives
No	Hunter and McClosky [2016]; M-MOBA[†] [Branke and Zhang, 2015, Branke et al., 2016]	MOCBA[†] [Lee et al., 2010] and its variants by Teng et al. [2010], Li et al. [2018]; Choi and Kim [2018][†]
Yes	Feldman and Hunter [2018]	This work, and some versions of MOCBA in Li et al. [2018]

[†] Requires a normality assumption on the random objective vectors.

1.1 Commentary on the Proposed Algorithm

To understand features of the proposed algorithm, let us first discuss the main challenge associated with solving the MORS problem. Imagine a scenario in which r systems are to be classified as Pareto or non-Pareto on d objectives, after expending a total of n simulation replications across all systems. Each system is deemed Pareto or non-Pareto once the objectives corresponding to each system have been *estimated* using the simulation budget assigned to it. And, since deeming a system Pareto or non-Pareto is a straightforward task, virtually all of the difficulty in solving the MORS problem lies in identifying how much of the simulation budget n should be allocated to each system toward minimizing the probability of misclassifying a Pareto system as non-Pareto, or vice versa. To define terminology, we say a *misclassification (MC) event* occurs if, after all of the simulation budget n has been expended, the estimated Pareto set is not equal to the true Pareto set. An MC event can happen in two ways: a truly Pareto system is estimated as non-Pareto, or a truly non-Pareto system is estimated as Pareto. Like Hunter and McClosky [2016], we call these MC events *misclassification by exclusion* (MCE) and *misclassification by inclusion* (MCI), respectively; Lee et al. [2010] refer to these same events as Type II error and Type I error, respectively.

It is easy to construct simulation budget allocation schemes that ensure the MC probabilities decay to zero as the simulation budget increases — since the number of systems r is finite, one only needs to ensure that as $n \to \infty$, each of the r systems is allocated a positive fraction of n. For example, equally allocating the simulation budget across all systems trivially ensures the MC probability decays to zero as $n \to \infty$. However, such schemes are known to be naïve because the resulting *decay rate* of the MC probability tends to be slow, a fact that is often reflected unambiguously during implementation. Thus, as we see it, the fundamental question in solving MORS problems is that of deciding the simulation budget allocation across the r systems which maximizes the MC probability decay rate. Identifying such allocations is a theoretically and computationally challenging question, leading to sophisticated procedures like MOCBA and our procedure, multi-objective Sampling Criteria for Optimization using Rate Estimators (SCORE). We now discuss the salient aspects of multi-objective SCORE that allow it to resolve the theoretical question of optimal allocation while keeping the resulting algorithm computationally efficient.

1. (Exact characterization of the MC probability decay rate.) We provide what appears to be the first exact characterization of the MC probability decay rate for MORS problems with three or more objectives; we call this characterization the *brute force rate.* Such characterization has been elusive due to the difficulty of analyzing the MC event, which includes the possibility of both MCE and MCI events. For example, faced with this challenge, (a) MOCBA assumes independence of the objectives and heuristically chooses the probability of one of the two events, MCE or MCI, as the sole criterion for allocating the remaining budget; and (b) Li et al. [2018] provide bounds on the rate. We resolve the question of identifying the decay rate expression while incorporating both MCE and MCI events and retaining dependence between the objectives. The brute force rate forms the basis for our approximations, and enables us to assess their quality for small problem instances.

2. (Phantom Pareto systems for approximating the brute force rate.) Computing the brute force rate for problems with more than a few systems turns out to be an especially difficult task. To facilitate such computation, leading to an implementable algorithm, we characterize *phantom Pareto systems* for three or more objectives. Phantom Pareto systems are fictitious systems

constructed by combining the objectives of strategically chosen Pareto systems. The phantom Pareto systems enable us to approximate the brute force rate in a certain asymptotic regime; we call the resulting rate the *phantom rate*.

3. (Identifying the phantom Pareto systems.) The phantom Pareto systems are integral to approximating the brute force rate as a function of the simulation budget allocation fractions across systems. However, identifying the phantom Pareto systems for a given set of systems is itself a non-trivial problem. We present an algorithm that identifies all phantom Pareto systems efficiently in $O(\log^{d-1} p)$ computing time, where p is the number of Pareto systems.
4. (Approximation to the optimal budget allocation.) Identifying the simulation budget allocation across systems that maximizes the MC probability decay rate expression involves solving a multistage optimization problem, a severe impediment to implementation on a large scale. A series of strategic approximations to the original multistage optimization problem, including the limiting SCORE regime, the phantom approximation to the brute force rate, and strategic constraint reduction, resolves this issue and allows efficient solution within SCORE.
5. (Effect of dependent objectives.) SCORE models dependence between the objective estimates within a system. The effect of modeling dependence reveals itself most clearly by comparing the MC probability decay rate of an allocation that models dependence versus one that does not. When there are three objectives or the number of systems is small, modeling dependence requires little cost and provides moderate gains in efficiency. However, for large problems with four or more objectives, the gains in efficiency diminish relative to the computational cost of solving for the SCORE allocation. Thus we also propose the independent SCORE (iSCORE) allocation for large problem instances in four or more objectives.
6. (Efficient computation.) Extensive numerical experimentation suggests SCORE's stable and efficient performance on a variety of MORS problems. For example, due mainly to the approximations, SCORE is able to solve MORS problems having many thousands of systems within seconds on a standard laptop computer, reflecting speeds that are appreciably faster than MOCBA when the Pareto set is small relative to the total number of systems. We find that the effect of the introduced approximations on the optimality gap of the resulting sampling allocation is negligible in the vast majority of problem instances.

1.2 Organization

After preliminary topics, the paper is organized according to the elements listed in the previous section. Our preliminary topics include notation and convention in §1.3, related R&S literature beyond Table 1 in §2, and our problem statement in §3. Then, we characterize the brute force rate and the optimal allocation in §4. We approximate the optimal allocation in the limiting SCORE regime in §5, and define the phantom Pareto systems in §5.2. We define the phantom approximation to the brute force rate in §6, and the SCORE framework appears in §7. We investigate computational complexity in §8, describe a sequential algorithm for implementation in §9, and conduct numerical experiments on the sequential algorithms in §10.

1.3 Notation and Convention

With few exceptions, constants are denoted by lower-case letters (a), random variables by capital letters (G), sets by script capital letters ($\mathcal{S}$), vectors by bold ($\mathbf{g}$), random vectors by capital bold ($\boldsymbol{G}$), and operators by blackboard bold ($\mathbb{P}\{\cdot\}$). When comparing two d-dimensional vectors $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d)$, we use the notation $\mathbf{x} \leqq \mathbf{y}$ to signify that $x_k \leq y_k$ for all $k \in \{1, \ldots, d\}$, and we use $\mathbf{x} \leq \mathbf{y}$ to signify that $\mathbf{x} \leqq \mathbf{y}$ but $\mathbf{x} \neq \mathbf{y}$. (The notation for vector orderings "$\leq$" and "$\leqq$" are standard in the multi-objective optimization literature; see, e.g., Ehrgott [2005], Wiecek et al. [2016].) We let $\mathbf{0}_{d\times p}$ and $\mathbf{1}_{d\times p}$ denote a d-by-p matrices containing zeros and ones, respectively. The symbol $\mathbb{I}_{\{\cdot\}}$ denotes the indicator function. Finally, although we broadly follow the multi-objective optimization convention that the term "efficient" corresponds to the decision space and "Pareto optimal" corresponds to the objective vector space [see, e.g., Ehrgott, 2005], in R&S, the system indices may be arbitrarily assigned. Thus we work entirely in the objective vector space, and we refer to systems with non-dominated objective vectors as Pareto systems.

2 Related R&S Literature

In what follows, we review related concepts and literature in three areas: (a) the notion of fixed-budget versus fixed-precision in R&S procedures; (b) asymptotically optimal allocations within R&S procedures; (c) other MORS procedures. Due to the infancy of MORS, most of the references in our discussion of (a) and (b) are in the context of single-objective R&S. This, however, serves our purposes of introducing the reader to the main issues underlying (a) and (b).

2.1 Fixed Budget and Fixed Precision Procedures for R&S

In the single-objective context, R&S has a long history and many efficient procedures exist [see Kim and Nelson, 2006, for an overview]. Broadly, single-objective R&S procedures consist of obtaining one or more simulation replications from every system, constructing estimators of the expected system performances, and using these estimators to declare one system as the estimated best.

Usually, R&S procedures provide some form of guarantee on the quality of the returned solution or on the simulation efficiency of the procedure. Hunter and Nelson [2017] classify R&S methods as *fixed-precision* procedures if their primary objective is to provide some form of probabilistic guarantee on the quality of the returned solution. Fixed-precision procedures usually attempt to expend as few simulation replications as possible while providing the probabilistic solution quality guarantee upon termination, executing until a certain termination criterion is met and for which the probabilistic guarantee holds. Hunter and Nelson [2017] classify R&S methods as *fixed-budget* procedures if, given some total simulation budget to expend, the procedure expends the simulation budget in a way that guarantees simulation efficiency. Fixed-budget procedures usually attempt to maximize the probability of correctly selecting the best systems while expending only as many simulation replications as the fixed simulation budget allows. For single-objective R&S, notable procedures exist in both categories, including fixed-precision procedures NSGS [Nelson et al., 2001], $\mathcal{KN}$ [Kim and Nelson, 2001], BIZ [Frazier, 2014], and GSP [Ni et al., 2017], and fixed-budget procedures Optimal Computing Budget Allocation (OCBA) [Chen et al., 2000], Expected Value of Information (EVI) [Chick et al., 2010], Knowledge Gradient (KG) [Frazier et al., 2008], and SCORE [Pasupathy et al., 2015], some of which are related [Ryzhov, 2016].

2.2 Previous Work on Asymptotically Optimal Allocations

Asymptotically optimal allocations provide a theoretical basis for one class of fixed-budget procedures that includes OCBA and SCORE. For single-objective, unconstrained R&S, Glynn and Juneja [2004] provide a large-deviations based asymptotically optimal allocation that maximizes the false selection probability decay rate. The false selection probability is the probability that a system other than the true best system will be estimated as best when the total simulation budget is expended. Glynn and Juneja [2004] also show that, under a normality assumption and assuming the allocation to the best system is much larger than the allocation to each suboptimal system, the asymptotically optimal allocation corresponds to OCBA.

Pasupathy et al. [2015] provide insight into the types of problems for which allocating a much larger proportion of the simulation budget to the best system than to each suboptimal system is an optimal strategy. They show that fixing the objective value of the best system and sending the total number of systems r to infinity under appropriate regularity conditions results in an asymptotically optimal allocation to each suboptimal system that is $\Theta(1/r)$ and an asymptotically optimal allocation to the best system that is $\Theta(1/\sqrt{r})$. The allocations to the suboptimal systems that result from this limiting regime are the SCORE allocations, which also correspond to OCBA under a normality assumption. (While the primary focus of Pasupathy et al. [2015] is on single-objective, stochastically constrained R&S, their results hold for unconstrained R&S.)

Building on this work, Feldman and Hunter [2018] demonstrate that for a fixed Pareto set with exactly two objectives, as the total number of systems goes to infinity under a normality assumption and appropriate regularity conditions — including that the non-Pareto systems are added "evenly" with respect to the Pareto systems — the asymptotically optimal allocation to each non-Pareto system is $\Theta(1/r)$, and the asymptotically optimal allocation to each Pareto system is $\Theta(1/\sqrt{r})$. Feldman and Hunter [2018] provide SCORE allocations for bi-objective R&S that account for correlation between the objectives. They show that the bi-objective SCORE allocations are fast and accurate for up to ten thousand systems. Feldman and Hunter [2018] provide the theoretical background and proof-of-concept for our work on multi-objective SCORE.

Although we discuss the key papers that lead to our work above, a few other papers exist on this topic. We categorize these papers by some of their differences in Table 2. In particular, we categorize the papers by the number of stochastic objectives (d) and stochastic constraints (c) for which they were designed, whether they account for dependence between the objectives and constraints, the distributions for which they provide a characterization of the asymptotically optimal allocation, the distributions for which they provide an implementation or example, and whether they contain an asymptotically optimal allocation obtained through a limiting SCORE regime. Finally, while we are aware of results in Glynn and Juneja [2011, 2015] regarding estimating rate functions, our numerical experience has been positive when estimating only the parameters of an assumed normal family.

Table 2: Some key papers on asymptotically optimal allocation are classified by their contributions.

Paper	Stochastic Obj. / Con.	Dependence	Dist'n Rate Pf. / Implementation[a]	SCORE limit, $r \to \infty$[a]
Glynn and Juneja [2004]	$d = 1, c = 0$	N/A	G / N, Bernoulli	No
Szechtman and Yücesan [2008]	$d = 0, c \geq 1$	No	G / N, Bernoulli	No
Hunter and Pasupathy [2013][b]	$d = 1, c \geq 1$	No	G / N	No
Pasupathy et al. [2015][c]	$d = 1, c \geq 0$	Yes	G / N	Yes for G
Hunter and McClosky [2016]	$d = 2, c = 0$	No	G / N, Chi-Squared	No
Feldman and Hunter [2018][d]	$d = 2, c = 0$	Yes	G / N	Yes for N, G by C
Li et al. [2018]	$d \geq 2, c = 0$	Yes	–[e]/ N, Bernoulli	No
This work[f]	$d \geq 2, c = 0$	Yes	G / N	N by C, G by C

[a] G stands for General and light-tailed; N stands for Normal; C stands for Conjecture.
[b] Subsumes preliminary work in the WSC paper Hunter and Pasupathy [2010] and thesis Hunter [2011].
[c] Subsumes preliminary work in the WSC papers Hunter et al. [2011], Pujowidianto et al. [2012].
[d] Subsumes preliminary work in the WSC paper Hunter and Feldman [2015].
[e] Provides bounds on the rate of decay of $\mathbb{P}\{\text{MC}\}$.
[f] Subsumes or replaces preliminary work in the WSC paper Feldman et al. [2015] and thesis Feldman [2017].

2.3 Other Work on MORS

Several other approaches to MORS exist, including fixed-precision procedures that identify the entire Pareto set and methods that identify only a subset of the Pareto systems by requesting the decision-maker to specify preferences in advance, e.g., through a utility function. Early work on fixed-precision procedures that identify the entire Pareto set include Batur and Choobineh [2010], Lee [2014], Wang and Wan [2017]. MORS procedures that identify a subset of the Pareto set, often by requiring the decision-maker to specify a utility function in advance, include the fixed-budget and fixed-precision procedures of Dudewicz and Taneja [1978, 1981], Butler et al. [2001], Frazier and Kazachkov [2011], Merrick et al. [2015], Mattila and Virtanen [2015].

3 Problem Setting and Formulation

We provide a formal problem statement and assumptions required in the remainder of the paper.

3.1 Problem Statement

We write the MORS problem as

$$\text{Problem } M\text{: Find } \operatorname{argmin}_{s \in \mathcal{S}} \mathbf{g}(s) := (g_1(s), \ldots, g_d(s))\,, \tag{1}$$

where $\mathbf{g}(s) \in \mathbb{R}^d$ is a vector representing the expected performance of system s on each of the d objectives, $\mathcal{S} := \{1, \ldots, r\}$ is a finite set of system indices, and $\mathcal{D} := \{1, \ldots, d\}$ is a finite set of objective indices. The minimum is taken with respect to the vector ordering $\leq$, where we say that system s *dominates* system s' and write $\mathbf{g}(s) \leq \mathbf{g}(s')$ if and only if $g_k(s) \leq g_k(s')$ for all $k \in \mathcal{D}$ and $\mathbf{g}(s) \neq \mathbf{g}(s')$. The solution to Problem M is the set of indices of the Pareto optimal systems, $\mathcal{P} := \{i \in \mathcal{S} : \nexists \text{ system } s \in \mathcal{S} \text{ such that } \mathbf{g}(s) \leq \mathbf{g}(i)\}$.

For all systems $s \in \mathcal{S}$, let $\boldsymbol{G}_m(s) := (G_{1m}(s), \ldots, G_{dm}(s))$ be the performance vector of system s on the mth simulation replication. Define the vector of sample means after observing n samples from system s as $\overline{\boldsymbol{G}}(s,n) = (\overline{G}_1(s,n), \ldots, \overline{G}_d(s,n)) := n^{-1} \sum_{m=1}^{n} \boldsymbol{G}_m(s)$. Let α_s be the proportion of the simulation budget n allocated to system $s \in \mathcal{S}$, and define $\widehat{\boldsymbol{G}}(s) := \overline{\boldsymbol{G}}(s, n\alpha_s)$ and $\widehat{G}_k(s) := \overline{G}_k(s, n\alpha_s)$ for all $s \in \mathcal{S}$, $k \in \mathcal{D}$. Using these estimators, after the budget n has been expended, construct the estimated Pareto set $\hat{\mathcal{P}} := \{i \in \mathcal{S} : \nexists \text{ system } s \in \mathcal{S} \text{ such that } \widehat{\boldsymbol{G}}(s) \leq \widehat{\boldsymbol{G}}(i)\}$.

Ideally, at the end of sampling, $\hat{\mathcal{P}} = \mathcal{P}$. If $\hat{\mathcal{P}} \neq \mathcal{P}$, we say that a *misclassification* (MC) event occurs. We seek a simulation budget allocation $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_r)$, $\sum_{s=1}^{r} \alpha_s = 1$ that maximizes the rate of decay of the probability of an MC event as the simulation budget n increases, thus providing an efficiency guarantee for solving Problem M.

3.2 Assumptions

First, we require that each Pareto system is distinguishable from every other system on each objective, which is a standard assumption in the asymptotically optimal allocation literature.

Assumption 1. *There exists* $\delta > 0$ *such that* $\min\{|g_k(s) - g_k(i)| : s \in \mathcal{S},\ i \in \mathcal{P},\ s \neq i,\ k \in \mathcal{D}\} > \delta$.

For brevity and simplicity in presenting our results, throughout the remainder of the paper, we assume that for each system $s \in \mathcal{S}$, the performance vectors $\boldsymbol{G}_m(s), m = 1, 2, \ldots$ are independent and identically distributed (i.i.d.) multivariate normal random variables. We further assume that all systems are simulated independently of each other. Notice that all results in §4 hold more generally under the standard assumptions for the Gärtner-Ellis Theorem [Dembo and Zeitouni, 1998]; see, e.g., Feldman and Hunter [2018] for assumptions similar to those we require.

Assumption 2. *For each system* $s \in \mathcal{S}$, $\boldsymbol{G}_m(s), m = 1, 2, \ldots$ *are i.i.d. MVN*$(\mathbf{g}(s), \Sigma(s))$ *random vectors, where* $\Sigma(s)$ *is a positive definite covariance matrix with diagonal entries* $\sigma_1^2(s), \ldots, \sigma_d^2(s)$ *and off-diagonal entries* $\rho_{k_1k_2}(s)\sigma_{k_1}(s)\sigma_{k_2}(s)$ *in the* (k_1, k_2) *position,* $\rho_{k_1k_2}(s) \in (-1, 1)$ *and* $k_1, k_2 \in \mathcal{D}$. *Further, the systems are simulated independently, thus* $\{\boldsymbol{G}_m(s) \colon s \in \mathcal{S}, m = 1, 2, \ldots\}$ *are mutually independent.*

This assumption guides the model that we use for sampling, but does not preclude the use of our methods in scenarios that violate these assumptions, such as when using common random numbers. Our algorithms technically are suboptimal in such a case, but may still provide significant improvement over naïve methods. The assumption of normality is widely used with success in the R&S literature; some discussion of the violation of such assumptions appears in Hunter and Pasupathy [2013] and Pasupathy et al. [2015].

Under Assumption 2, the probability measures governing $\overline{\boldsymbol{G}}(s,n)$ and $\overline{G}_k(s,n)$ obey a large deviations principle for all $s \in \mathcal{S}, k \in \mathcal{D}$. For all $s \in \mathcal{S}$, let the large deviations rate function corresponding to the random vector $\overline{\boldsymbol{G}}(s,n)$ be $I_s(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^d$, and let the large deviations rate function corresponding to the random variable $\overline{G}_k(s,n)$ be $J_{sk}(x)$ for $x \in \mathbb{R}$. Under Assumption 2, $I_s(\mathbf{x}) = (1/2)(\mathbf{g}(s) - \mathbf{x})^\intercal \Sigma(s)^{-1}(\mathbf{g}(s) - \mathbf{x})$ and $J_{sk}(x) = (g_k(s) - x)^2/(2\sigma_k^2(s))$ for all $s \in \mathcal{S}, k \in \mathcal{D}$.

4 An Exact Characterization of the Asymptotically Optimal Allocation

To obtain the MC probability decay rate, we formulate the MC event in terms of a brute-force enumeration of all the ways an MC event can occur. The optimal allocation strategy follows from optimizing the MC probability decay rate as a function of the simulation budget allocation $\boldsymbol{\alpha}$.

4.1 The "Brute-Force" Misclassification Probability Decay Rate

We begin by writing the MC event, MC $:= (\hat{\mathcal{P}} \neq \mathcal{P})$, in a way that facilitates analysis. Recall that there are two ways an MC event can occur: *misclassification by exclusion* (MCE), in which a truly Pareto system is falsely excluded from $\hat{\mathcal{P}}$, and *misclassification by inclusion* (MCI), in which a truly non-Pareto system is falsely included in $\hat{\mathcal{P}}$. Thus the MC event can be written as MC $=$ MCE$\cup$MCI. Feldman [2017] shows that MC $=$ MCE$_{\mathcal{P}} \cup$MCI, where MCE$_{\mathcal{P}}$ denotes the event that a truly Pareto system is estimated as dominated by another Pareto system, MCE$_{\mathcal{P}} := \cup_{i\in\mathcal{P}} \cup_{i'\in\mathcal{P}} \widehat{\boldsymbol{G}}(i') \leq \widehat{\boldsymbol{G}}(i)$, and MCI $:= \cup_{j\in\mathcal{P}^c} \cap_{i\in\mathcal{P}} \cup_{k\in\mathcal{D}} \widehat{G}_k(j) \leq \widehat{G}_k(i)$. Then assuming the limits exist, the $\mathbb{P}\{\text{MC}\}$ decay rate is $-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\text{MC}\} = \min\big(-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\text{MCE}\}, -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\text{MCI}\}\big)$. We obtain an expression for the $\mathbb{P}\{\text{MC}\}$ decay rate in parts, where the easiest part to obtain is the decay rate for $\mathbb{P}\{\text{MCE}\}$. Thus we obtain an expression for this rate first. Then, we spend the remainder of this section obtaining an expression for the $\mathbb{P}\{\text{MCI}\}$ decay rate.

Using an analysis similar to Glynn and Juneja [2004], Feldman [2017] shows that the $\mathbb{P}\{\text{MCE}\}$

decay rate equals the minimum among the pairwise decay rates of the probability that one Pareto system dominates another Pareto system [see also Li, 2012, Li et al., 2018]. That is, define the pairwise decay rates of the probability that Pareto system i' dominates Pareto system i as

$$R^{\mathrm{MCE}}_{i'i}(\alpha_{i'}, \alpha_i) := \inf_{\mathbf{x}_{i'} \leq \mathbf{x}_i} \alpha_i I_i(\mathbf{x}_i) + \alpha_{i'} I_{i'}(\mathbf{x}_{i'}) \text{ for all } i, i' \in \mathcal{P}, i \neq i'.$$

Then the $\mathbb{P}\{\mathrm{MCE}\}$ decay rate is $-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\mathrm{MCE}\} = \min_{i\in\mathcal{P}} \min_{i'\in\mathcal{P},\, i'\neq i} R^{\mathrm{MCE}}_{i'i}(\alpha_{i'}, \alpha_i)$.

Now, consider the $\mathbb{P}\{\mathrm{MCI}\}$ decay rate. To be falsely included in the estimated Pareto set, a non-Pareto system j must be estimated as better than *each Pareto system* on some objective. This event contains dependence, which makes it more difficult to analyze. In the context of exactly two objectives, Hunter and McClosky [2016], Feldman and Hunter [2018] overcome this difficulty by re-formulating the MCI event as an MCE-like event involving phantom Pareto systems. Unfortunately, the re-formulation that works with two objectives does not work when there are three or more objectives, for reasons we briefly discuss in §6. We take a different approach to analyzing the $\mathbb{P}\{\mathrm{MCI}\}$ decay rate, which we call the *brute force* formulation. This formulation, which involves a brute force enumeration of all possible ways that a non-Pareto system j can "beat" every Pareto system on at least one objective, enables an easier analysis of the rate of decay of $\mathbb{P}\{\mathrm{MCI}\}$.

To specify the brute force formulation, without loss of generality, let the system labels $\{1, \ldots, p\}$ correspond to the Pareto systems. Since we have at least one Pareto system, system 1 is Pareto, and $\mathcal{P} = \{1, \ldots, p\}$. Recall that for any non-Pareto system to be falsely included in the Pareto set, it must beat each Pareto system $j \in \mathcal{P}$ on some objective. Specifically, for such a false inclusion event to happen, the non-Pareto system needs to beat the Pareto system 1 along some objective $\kappa_1 \in \{1, 2, \ldots, d\}$, beat the Pareto system 2 along some objective $\kappa_2 \in \{1, 2, \ldots, d\}$, and so on, beating the Pareto system p along along some objective $\kappa_p \in \{1, 2, \ldots, d\}$. The set $\mathcal{K} = \{\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_p) \in \{1, 2, \ldots, d\}^p\}$ thus represents "all possible ways" for a non-Pareto system to be falsely included in the Pareto set. Now define the *brute force* MCI *event* as $\mathrm{MCI}_{\mathrm{bf}} := \cup_{j\in\mathcal{P}^c} \cup_{\boldsymbol{\kappa}\in\mathcal{K}} \cap_{i\in\mathcal{P}} \widehat{G}_{\kappa_i}(j) \leq \widehat{G}_{\kappa_i}(i)$. By the definition of $\mathcal{K}$, the following proposition holds.

Proposition 1. $\mathrm{MCI} = \mathrm{MCI}_{\mathrm{bf}}$.

Since the $\mathrm{MCI}_{\mathrm{bf}}$ event reformulates MCI as a union over all non-Pareto systems and all objective index vectors, the $\mathbb{P}\{\mathrm{MCI}\}$ decay rate can be expressed as the minimum decay rate of the probabilities that a non-Pareto system j is falsely included via the objectives specified by $\boldsymbol{\kappa}$. The following lemma states the $\mathbb{P}\{\text{MCI}\}$ decay rate using the brute force MCI event; for brevity, define

$$R_{j\boldsymbol{\kappa}}^{\text{MCI}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}}) := \inf_{x_{j\kappa_i} \leq x_{i\kappa_i} \forall i \in \mathcal{P}} \alpha_j I_j(\mathbf{x}_j) + \sum_{i \in \mathcal{P}} \alpha_i J_{i\kappa_i}(x_{i\kappa_i}),$$

where $\boldsymbol{\alpha}_{\mathcal{P}} := (\alpha_1, \ldots, \alpha_p)$ is the vector of simulation budget allocations for the Pareto systems. A complete proof appears in Feldman [2017]; we provide a proof sketch.

Lemma 1. *The* $\mathbb{P}\{\text{MCI}\}$ *decay rate is* $-\lim \frac{1}{n} \log \mathbb{P}\{\text{MCI}\} = \min_{j \in \mathcal{P}} \min_{\boldsymbol{\kappa} \in \mathcal{K}} R_{j\boldsymbol{\kappa}}^{\text{MCI}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$.

Proof sketch. Using the brute force formulation, if the limits exist, we have that the $\mathbb{P}\{\text{MCI}\}$ decay rate is $-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\text{MCI}\} = \min_{j \in \mathcal{P}} \min_{\boldsymbol{\kappa} \in \mathcal{K}} \big(-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\text{MCI}_{\text{bf}}(j, \boldsymbol{\kappa})\}\big)$, where $\text{MCI}_{\text{bf}}(j, \boldsymbol{\kappa}) := \cap_{i \in \mathcal{P}}\ \widehat{G}_{\kappa_i}(j) \leq \widehat{G}_{\kappa_i}(i)$. To derive the rate of decay of $\mathbb{P}\{\text{MCI}_{\text{bf}}(j, \boldsymbol{\kappa})\}$, we consider the random variables involved in the expression $\text{MCI}_{\text{bf}}(j, \boldsymbol{\kappa})$, which are $\widehat{\boldsymbol{G}}(j)$ and $\widehat{G}_{\kappa_1}(1), \ldots, \widehat{G}_{\kappa_p}(p)$. Since the Pareto systems are sampled independently, the random variables $\widehat{G}_{\kappa_1}(1), \ldots, \widehat{G}_{\kappa_p}(p)$ are mutually independent, and each of these random variables is independent of $\widehat{\boldsymbol{G}}(j)$. Thus the rate functions corresponding to these random variables add up in the expression for $R_{j\boldsymbol{\kappa}}^{\text{MCI}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$. □

Combining the decay rates for $\mathbb{P}\{\text{MCE}\}$ and $\mathbb{P}\{\text{MCI}\}$, we present this section's main theorem.

Theorem 1. *The* $\mathbb{P}\{\text{MC}\}$ *decay rate, which we call the* brute force rate, *is*

$$z^{\text{bf}}(\boldsymbol{\alpha}) := -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\text{MC}\} = \min\big(\min_{i \in \mathcal{P}} \min_{i' \in \mathcal{P}, i \neq i'} R_{i'i}^{\text{MCE}}(\alpha_{i'}, \alpha_i), \min_{j \in \mathcal{P}^c} \min_{\boldsymbol{\kappa} \in \mathcal{K}} R_{j\boldsymbol{\kappa}}^{\text{MCI}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})\big).$$

Theorem 1 states that the overall $\mathbb{P}\{\text{MC}\}$ decay rate is found by considering the decay rates of the probabilities of the most likely events among (a) the pairwise false exclusion events between Pareto systems, and (b) all possible ways a non-Pareto system can be falsely included in the Pareto set by being estimated as better than every Pareto system on at least one objective.

4.2 The Optimal Allocation Strategy

Maximizing the $\mathbb{P}\{\text{MC}\}$ decay rate in Theorem 1 involves solving the following Problem Q, having solution $\boldsymbol{\alpha}^*$:

$$
\begin{aligned}
\text{Problem } Q: \qquad & \text{maximize } \ z^{\text{bf}} \ \text{ s.t.} \\
& R^{\text{MCE}}_{i'i}(\alpha_{i'}, \alpha_i) \geq z^{\text{bf}} \text{ for all } i, i' \in \mathcal{P} \text{ such that } i \neq i', \\
& R^{\text{MCI}}_{j\boldsymbol{\kappa}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}}) \geq z^{\text{bf}} \text{ for all } j \in \mathcal{P}^c, \boldsymbol{\kappa} \in \mathcal{K}, \\
& \textstyle\sum_{s=1}^{r} \alpha_s = 1, \ \alpha_s \geq 0 \quad \text{for all } s \in \mathcal{S}.
\end{aligned}
$$

As in previous work on asymptotically optimal allocations, Problem Q is a concave maximization problem in the decision variable $\boldsymbol{\alpha}$ [e.g., Glynn and Juneja, 2004, Hunter and Pasupathy, 2013, Pasupathy et al., 2015, Feldman and Hunter, 2018]. However, solving Problem Q is especially computationally burdensome. To compute the values of the constraints corresponding to controlling the $\mathbb{P}\{\text{MCI}\}$ decay rate, we must calculate $|\mathcal{P}^c|d^p$ rates corresponding to the number of non-Pareto systems times the total number of $\boldsymbol{\kappa}$ vectors.

5 Approximating the Optimal Allocation in the Limiting SCORE Regime

The total number of rates to compute when solving Problem Q becomes prohibitively large relatively fast: for $d = 3$ dimensions and $r = 30$ total systems, $p = 10$ of which are Pareto systems, we must compute over one million rates. Considering this computational complexity, we provide heuristics that approximate the solution to Problem Q and require far less computational resources. Inspired by the SCORE family of allocations [Pasupathy et al., 2015, Feldman and Hunter, 2018], in this section, we consider an approximately optimal allocation for MORS problems with three or more objectives and many non-Pareto systems. While the bi-objective SCORE allocations provided by Feldman and Hunter [2018] are rigorously derived as a limiting solution to Problem Q in which the number of sub-optimal systems tend to infinity, for simplicity, we instead assume the existence of a limiting SCORE regime like the one in Feldman and Hunter [2018]. We discuss the conditions under which such a regime is likely to exist and the limiting optimal allocations that result. Then, we simplify the limiting optimal allocations by showing that the brute force rate can be approximated by a formulation involving phantom Pareto systems.

5.1 Approximating the Allocation to the Non-Pareto Systems

In this section, we simplify finding the optimal allocation as the solution to Problem Q by predetermining the relative allocations between the non-Pareto systems. Since the constraints that correspond to controlling the rate of decay of $\mathbb{P}\{\text{MCE}\}$ in Problem Q do not involve the non-Pareto systems, we relax Problem Q to consider only constraints that correspond to controlling the $\mathbb{P}\{\text{MCI}\}$ decay rate. That is, we instead consider

$$\begin{aligned} \text{Problem } \widetilde{Q}: \qquad & \text{maximize } \ \tilde{z} \ \text{ s.t.} \\ & R^{\text{MCI}}_{j\boldsymbol{\kappa}}(\tilde{\alpha}_j, \tilde{\boldsymbol{\alpha}}_{\mathcal{P}}) \geq \tilde{z} \text{ for all } j \in \mathcal{P}^c, \boldsymbol{\kappa} \in \mathcal{K}, \\ & \textstyle\sum_{s=1}^{r} \tilde{\alpha}_s = 1, \ \tilde{\alpha}_s \geq 0 \ \text{ for all } s \in \mathcal{S}. \end{aligned}$$

Under Assumption 2, we write the rate $R^{\text{MCI}}_{j\boldsymbol{\kappa}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$ as the solution to

$$\begin{aligned} \text{Problem } R^{\text{MCI}}_{j\boldsymbol{\kappa}}: \quad \text{minimize} \quad & \frac{\alpha_j}{2} \begin{bmatrix} \mathbf{g}(j) - \mathbf{x}_j \\ \mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) - \mathbf{x}_{\boldsymbol{\kappa}} \end{bmatrix}^{\intercal} \begin{bmatrix} \Sigma(j)^{-1} & \mathbf{0}_{d\times p} \\ \mathbf{0}_{p\times d} & \Sigma_{\mathcal{P}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{g}(j) - \mathbf{x}_j \\ \mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) - \mathbf{x}_{\boldsymbol{\kappa}} \end{bmatrix} \\ \text{s.t.} \quad & \begin{bmatrix} A(\boldsymbol{\kappa}) & \mathbb{I}_{p\times p} \end{bmatrix} \begin{bmatrix} \mathbf{g}(j) - \mathbf{x}_j \\ \mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) - \mathbf{x}_{\boldsymbol{\kappa}} \end{bmatrix} \leq \begin{bmatrix} \mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) - \mathbf{g}_j(\boldsymbol{\kappa}) \end{bmatrix}, \end{aligned}$$

which is a quadratic program with linear constraints in the decision variables $\mathbf{x}_j$ and $\mathbf{x}_{\boldsymbol{\kappa}}$, and the notation is described as follows: $\mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) := (g_{\kappa_1}(1), \ldots, g_{\kappa_p}(p))$ is a p-dimensional vector of Pareto system objective values; $\mathbf{x}_{\boldsymbol{\kappa}} := (x_{1\kappa_1}, \ldots, x_{p\kappa_p})$ is a p-dimensional vector of decision variables; $\Sigma(j)$ is the covariance matrix from Assumption 2; $\Sigma_{\mathcal{P}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$ is a diagonal matrix with entries $(\alpha_j/\alpha_1)\sigma^2_{\kappa_1}(1), \ldots, (\alpha_j/\alpha_p)\sigma^2_{\kappa_p}(p)$; $A(\boldsymbol{\kappa})$ is a p-by-d matrix in which the (i,k)th entry is $-\mathbb{I}_{\{\kappa_i = k\}}$; $\mathbb{I}_{p\times p}$ is a $p \times p$ identity matrix; and $\mathbf{g}_j(\boldsymbol{\kappa}) := (g_{\kappa_1}(j), \ldots, g_{\kappa_p}(j))$ is a p-dimensional vector containing the objective values of system j on the objectives specified by $\boldsymbol{\kappa}$. In the context of Problem $\widetilde{Q}$, the solution to Problem $R^{\text{MCI}}_{j\boldsymbol{\kappa}}$ is a function of $\tilde{\boldsymbol{\alpha}}$. We denote the solution as $(\mathbf{x}^*_j(\tilde{\alpha}_j, \tilde{\boldsymbol{\alpha}}_{\mathcal{P}}), \mathbf{x}^*_{\boldsymbol{\kappa}}(\tilde{\alpha}_j, \tilde{\boldsymbol{\alpha}}_{\mathcal{P}}))$.

We now make the following assumption on the existence of a regime in which the optimal allocation that results from solving Problem $\widetilde{Q}$, which we call $\tilde{\boldsymbol{\alpha}}^* = (\tilde{\alpha}^*_1, \ldots, \tilde{\alpha}^*_r)$, is such that $\tilde{\alpha}^*_j/\tilde{\alpha}^*_i \to 0$ in $\Sigma_{\mathcal{P}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$ for all non-Pareto systems $j \in \mathcal{P}^c$ and Pareto systems $i \in \mathcal{P}$ as the number of non-Pareto systems $|\mathcal{P}^c| \to \infty$. Further, this regime sends $\mathbf{x}^*_{\boldsymbol{\kappa}}(\tilde{\alpha}^*_j, \tilde{\boldsymbol{\alpha}}^*_{\mathcal{P}}) \to \mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa})$ in all Problems $R^{\text{MCI}}_{j\boldsymbol{\kappa}}$, thus implying that the Pareto systems receive so many samples at optimality that, relative to the non-Pareto systems, their objective values appear "fixed."

Assumption 3. *There exists a regime in which, by holding the Pareto systems fixed and adding non-Pareto systems so that $|\mathcal{P}^c| \to \infty$ according to certain regularity conditions (see Feldman and Hunter [2018] for the regularity conditions in the bi-objective case), we have $\tilde{\alpha}^*_j/\tilde{\alpha}^*_i \to 0$ for all $j \in \mathcal{P}^c, i \in \mathcal{P}$ in such a way that $\tilde{\alpha}^*_j = \Theta(1/|\mathcal{P}^c|)$ for all $j \in \mathcal{P}^c$, $\tilde{\alpha}^*_i = \Theta(1/\sqrt{|\mathcal{P}^c|})$ for all $i \in \mathcal{P}$, and $\lim_{|\mathcal{P}^c|\to\infty} R_{j\boldsymbol{\kappa}}(\tilde{\alpha}^*_j, \tilde{\boldsymbol{\alpha}}^*_{\mathcal{P}})/\tilde{\alpha}^*_j = \inf_{x_{j\kappa_i} \leq g_{\kappa_i}(i)\ \forall i \in \mathcal{P}} I_j(\mathbf{x}_j)$ for all $j \in \mathcal{P}^c, \boldsymbol{\kappa} \in \mathcal{K}$.*

This regime is likely to hold when non-Pareto systems are added "evenly" behind the existing Pareto systems, subject to a uniform upper bound on their true objective vector values. For readers interested in the regularity conditions and detailed mathematics surrounding such a regime, we suggest reading the bi-objective case presented by Feldman and Hunter [2018]. We re-emphasize that assuming this regime is useful for designing allocation polices that are close to optimal. Practically speaking, our allocation policy should work well for a variety of problems, which we explore in the numerical section.

Using the regime in Assumption 3, for each non-Pareto system $j \in \mathcal{P}^c$, set the *score* equal to $\mathbb{S}_j := \min_{\boldsymbol{\kappa}\in\mathcal{K}}\big(\inf_{x_{j\kappa_i}\leq g_{\kappa_i}(i)\ \forall i\in\mathcal{P}} I_j(\mathbf{x}_j)\big)$. Theorem 2 below provides the limiting relative optimal allocations between the non-Pareto systems. We simplify the score expressions in the next section.

Theorem 2. *Under Assumption 3, for all non-Pareto systems $j, j' \in \mathcal{P}^c$,*

$$\frac{\tilde{\alpha}^*_{j'}}{\tilde{\alpha}^*_j} = \frac{\mathbb{S}_j}{\mathbb{S}_{j'}} = \frac{\min_{\boldsymbol{\kappa}\in\mathcal{K}} \inf_{x_{j\kappa_i}\leq g_{\kappa_i}(i)\ \forall i\in\mathcal{P}} I_j(\mathbf{x}_j)}{\min_{\boldsymbol{\kappa}\in\mathcal{K}} \inf_{x_{j'\kappa_i}\leq g_{\kappa_i}(i)\ \forall i\in\mathcal{P}} I_{j'}(\mathbf{x}_{j'})}.$$

5.2 The Phantom Pareto System Simplification

The allocations in Theorem 2 still require the brute force computation of all the ways a non-Pareto system j can be falsely included in the Pareto set. In this section, we simplify the score calculation by removing unnecessary $\boldsymbol{\kappa}$ vectors. In the end, we will be left with a much more manageable calculation: instead of taking the minimum score over all of the $\boldsymbol{\kappa}$ vectors, we take the minimum score over all of the *phantom Pareto systems*. Phantom Pareto systems were introduced in Hunter and McClosky [2016] in the context of bi-objective R&S problems, where they are easy to identify. In three or more objectives, the phantom Pareto systems are harder to identify, but as we show below, they may be found by removing the redundant $\boldsymbol{\kappa}$ vectors.

To begin, we require additional notation. First, notice that we have not written the constraints of the infimum in the score calculation using vectors since $\mathbf{x}_j$ is a d-dimensional vector, while $\mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) := (g_{\kappa_1}(1), \ldots, g_{\kappa_p}(p))$ is a p-dimensional vector of Pareto system objective values specified by the objectives in $\boldsymbol{\kappa} \in \mathcal{K}$. Using new notation, we can write these constraints with vectors, as follows. Define the kth element of the d-dimensional vector

$$g_k^{\mathrm{bf}}(\boldsymbol{\kappa}) := \begin{cases} \min_{\{i\in\mathcal{P}:\, \kappa_i=k\}} g_k(i) & \text{if } k \in \{k \in \mathcal{D} : \exists\, i \in \mathcal{P} \text{ such that } \kappa_i = k\} \\ \infty & \text{otherwise.} \end{cases}$$

This notation essentially goes objective-by-objective and specifies the minimum values that the decision vector $\mathbf{x}_j$ must not exceed in the infimum in the score calculation. Then by construction, the following Lemma 2 holds; we characterize redundant $\boldsymbol{\kappa}$ vectors in Lemma 3.

Lemma 2. *The score* $\mathbb{S}_j(\boldsymbol{\kappa}) = \inf_{\mathbf{x}_j \le \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa})} I_j(\mathbf{x}_j)$ *for all* $j \in \mathcal{P}^c, \boldsymbol{\kappa} \in \mathcal{K}$.

Lemma 3. *If* $\boldsymbol{\kappa}, \boldsymbol{\kappa}' \in \mathcal{K}$ *are such that* $\mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}') \le \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa})$*, then* $\mathbb{S}_j(\boldsymbol{\kappa}) \le \mathbb{S}_j(\boldsymbol{\kappa}')$.

Proof. Suppose $\boldsymbol{\kappa}, \boldsymbol{\kappa}' \in \mathcal{K}$ are two vectors of objective indices such that $\mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}') \le \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa})$. Let $\mathcal{X}' := \{\mathbf{x}' : \mathbf{x}' \leqq \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}')\}$ and $\mathcal{X} := \{\mathbf{x} : \mathbf{x} \leqq \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa})\}$. Then $\mathcal{X}' \subseteq \mathcal{X}$. Since $\mathbf{g}(j) \notin \mathcal{X}$, using Lemma 2, $\mathbb{S}_j(\boldsymbol{\kappa}) = \inf_{\mathbf{x}_j \le \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa})} I_j(\mathbf{x}_j) \le \inf_{\mathbf{x}_j \le \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}')} I_j(\mathbf{x}_j) = \mathbb{S}_j(\boldsymbol{\kappa}')$. □

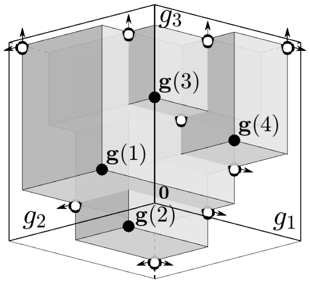

Figure 1: For $d = 3$, black circles represent Paretos and white circles represent phantom Paretos. Arrows imply the phantoms are located at infinity in that direction. (Figure inspired by Lacour et al. [2017].)

Now we define a minimal set of d-dimensional brute force points in the objective space as $\mathcal{G}^{\mathrm{ph}} := \{\mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}) : \boldsymbol{\kappa} \in \mathcal{K},\ \nexists\ \boldsymbol{\kappa}' \in \mathcal{K}$ such that $\mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}) \le \mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}')\}$. That is, the set $\mathcal{G}^{\mathrm{ph}}$ keeps only the points $\mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}), \boldsymbol{\kappa} \in \mathcal{K}$ that *do not dominate* any other points $\mathbf{g}_d^{\mathrm{bf}}(\boldsymbol{\kappa}'), \boldsymbol{\kappa}' \in \mathcal{K}$. The set $\mathcal{G}^{\mathrm{ph}}$ defines the *phantom Pareto systems* in d dimensions. We let the points in $\mathcal{G}^{\mathrm{ph}}$ be denoted as $\mathbf{g}^{\mathrm{ph}}(\ell) = (g_1^{\mathrm{ph}}(\ell), \ldots, g_d^{\mathrm{ph}}(\ell)) := (g_1(i_1(\ell)), \ldots, g_d(i_d(\ell)))$, where $i_k(\ell), k \in \mathcal{D}$ is the index of the Pareto system that contributes the kth objective value to the phantom Pareto system ℓ, and the letter ℓ is an index reserved for the phantom Pareto systems. The set $\mathcal{P}^{\mathrm{ph}}$ denotes the indices of the phantom Pareto systems, so that $\ell \in \mathcal{P}^{\mathrm{ph}}$. An illustration containing phantom Pareto systems appears in Figure 1. Using the phantom Pareto systems, the simplified score can be calculated without brute force enumeration, as in the following Theorem 3.

Theorem 3. *Under Assumption 3, for all non-Pareto systems* $j, j' \in \mathcal{P}^c$,

$$\frac{\tilde{\alpha}^*_{j'}}{\tilde{\alpha}^*_j} = \frac{\mathbb{S}_j}{\mathbb{S}_{j'}} = \frac{\min_{\ell \in \mathcal{P}^{\mathrm{ph}}} \inf_{\mathbf{x}_j \le \mathbf{g}^{\mathrm{ph}}(\ell)} I_j(\mathbf{x}_j)}{\min_{\ell \in \mathcal{P}^{\mathrm{ph}}} \inf_{\mathbf{x}_{j'} \le \mathbf{g}^{\mathrm{ph}}(\ell)} I_{j'}(\mathbf{x}_{j'})}.$$

For notational simplicity, define $\mathbb{S}_j(\ell) := \inf_{\mathbf{x}_j \le \mathbf{g}^{\mathrm{ph}}(\ell)} I_j(\mathbf{x}_j)$ for all $j \in \mathcal{P}^c, \ell \in \mathcal{P}^{\mathrm{ph}}$. We have only specified simplified calculations for the relative allocations between the non-Pareto systems; the allocations to the Pareto systems are determined heuristically in §7.

To calculate the scores, we must find the locations of the phantom Pareto systems. The problem of finding the phantom Pareto systems is related to Klee's measure problem for grounded boxes [see, e.g., Chan, 2013, Yildiz and Suri, 2012], and more specifically, to the problem of calculating the hypervolume indicator, which is a common performance metric in the deterministic multi-objective optimization literature [Lacour et al., 2017]. We provide an efficient algorithm for locating the phantom Pareto systems in the Online Supplement. Our algorithm is similar to the procedure described in Kaplan et al. [2008]. Importantly, Kaplan et al. [2008] prove that the number of phantom Pareto systems associated with a set of p Pareto systems in d-dimensional Euclidean space is $O(p^{\lfloor d/2 \rfloor})$; the query and storage/pre-processing complexities of their procedure, identical to our algorithm, are shown to be $O(\log^{d-1} p)$ and $O(p^{\lfloor d/2 \rfloor} \log^{d-1} p)$, respectively.

6 The Phantom Rate: An Approximation to the Brute Force Rate

Recall that our original expression for the rate of decay of $\mathbb{P}\{\text{MCI}\}$ required the brute force enumeration of all possible ways a non-Pareto system can be falsely included in the Pareto set. Now we consider a simplified and approximate overall rate of decay of $\mathbb{P}\{\text{MCI}\}$ defined by the phantom Pareto systems instead of the brute force enumeration vector $\boldsymbol{\kappa}$. To specify this rate, let $\mathcal{P}(\ell) = \{i_k(\ell) \in \mathcal{P} : \exists k \ni g_k(i) \in \{g_1^{\text{ph}}(\ell), \ldots, g_d^{\text{ph}}(\ell)\}\}$ denote the indices of all the Pareto systems $i_k(\ell) \in \mathcal{P}$ that contribute objective function value k to phantom Pareto system $\ell \in \mathcal{P}^{\text{ph}}$. Further, define the d-dimensional vector of variables $\mathbf{x}_\ell^{\text{ph}} = (x_{\ell 1}^{\text{ph}}, \ldots, x_{\ell d}^{\text{ph}})$. Then for each $j \in \mathcal{P}^c$ and $\ell \in \mathcal{P}^{\text{ph}}$, define the approximate rate of decay of $\mathbb{P}\{\text{MCI}\}$ using the phantom Pareto systems as

$$R_{j\ell}^{\text{ph}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}(\ell)}) := \inf_{\mathbf{x}_j \leq \mathbf{x}_\ell^{\text{ph}}} \alpha_j I_j(\mathbf{x}_j) + \sum_{k \in \mathcal{D}} \alpha_{i_k(\ell)} J_{i_k(\ell)k}(x_{\ell k}^{\text{ph}}), \tag{2}$$

where $\boldsymbol{\alpha}_{\mathcal{P}(\ell)}$ is the vector of proportional allocations for the Pareto systems in $\mathcal{P}(\ell)$. Using this approximation, we approximate the $\mathbb{P}\{\text{MC}\}$ decay rate in Theorem 1 as the *phantom rate*

$$z^{\text{ph}}(\boldsymbol{\alpha}) := \min\big(\min_{i \in \mathcal{P}} \min_{i' \in \mathcal{P}, i \neq i'} R_{i'i}^{\text{MCE}}(\alpha_{i'}, \alpha_i), \min_{j \in \mathcal{P}^c} \min_{\ell \in \mathcal{P}^{\text{ph}}} R_{j\ell}^{\text{ph}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}(\ell)})\big) \approx -\lim_{n \to \infty} \frac{1}{n} \log \mathbb{P}\{\text{MC}\}. \tag{3}$$

When there are two objectives, Feldman and Hunter [2018] prove that the rate in equation (3) is equal to the $\mathbb{P}\{\text{MC}\}$ decay rate. When there are three or more objectives, the rate in equation (3) is not necessarily equal to the $\mathbb{P}\{\text{MC}\}$ decay rate, as described in the following example.

Consider Problem N, which we define as a version of Problem M with two Pareto systems at $\mathbf{g}(1) = (2, 2.5, 5)$ and $\mathbf{g}(2) = (5, 3, 2)$, and one non-Pareto system at $\mathbf{g}(3) = (6, 5.3, 8)$, as shown in Figure 2. Each system's covariance matrix is the identity matrix, and all rates are calculated under equal allocation. The rates $R^{\text{MCE}}_{12}(1/3, 1/3) = 0.7708$, and $R^{\text{MCE}}_{21}(1/3, 1/3) = 0.7500$. The pairwise brute force and phantom $\mathbb{P}\{\text{MCI}\}$ rates are reported in Table 3. In this example, the overall brute force rate is $z^{\text{bf}} = 0.7367$, while the overall phantom rate is $z^{\text{ph}} = 0.6533$.

This discrepancy in rates occurs because in three or more dimensions, the phantom rate z^{ph} does not account for the ordering of the Pareto systems in the absence of an MCE event. The minimum in the brute force rate occurs at $\boldsymbol{\kappa} = (2, 1)$, which corresponds to non-Pareto system 3 being estimated as better than Pareto system 1 on objective 2 and better than Pareto system 2 on objective 1. Since the Pareto systems have similar values on objective 2, if they were to "switch places" on this objective, as shown in Figure 3, the estimated phantom Pareto system $\widehat{\boldsymbol{G}}^{\text{ph}}(4)$ would not correspond to the location of $\mathbf{g}^{\text{ph}}(4)$ from Figure 2. The non-Pareto system could be falsely included in the Pareto set if it were estimated as dominating $\widehat{\boldsymbol{G}}^{\text{ph}}(4)$. The brute force rate accounts for this possibility, while the phantom rate does not — the minimum in the phantom rate results from the possibility that the non-Pareto system is falsely estimated as dominating $\mathbf{g}^{\text{ph}}(5) = (\infty, 2.5, \infty)$, without considering the Pareto ordering on objective 2. Now, notice that this phenomenon does

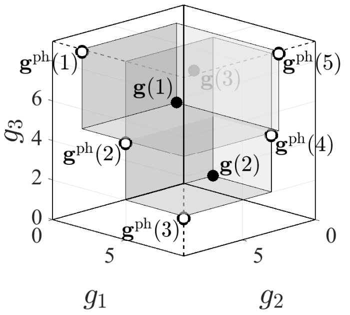

Figure 2: The figure shows the objective vector values for the systems in Problem N. We omit arrows from the phantoms with coordinates at infinity.

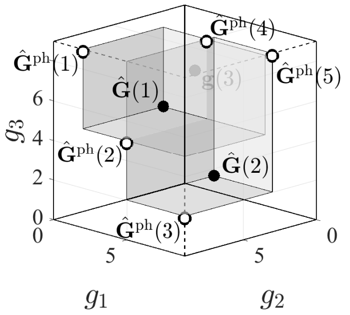

Figure 3: The figure shows estimated values for the Pareto systems in Problem N in which Pareto systems 1 and 2 have "switched places" on objective 2.

Table 3: The table reports the pairwise rates for Problem N under equal allocation, to four decimal places.

$\boldsymbol{\kappa}$ vectors for $R^{\text{MCI}}_{3\boldsymbol{\kappa}}(1/3, \mathbf{1/3})$									ℓ values for $R^{\text{ph}}_{3\ell}(1/3, \mathbf{1/3})$				
(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	1	2	3	4	5
1.3333	1.7422	4.3333	0.7367	0.7433	3.6533	0.8333	1.1908	3.0000	1.3333	0.8333	3.0000	1.1908	0.6533

not occur in any two-dimensional projections of this problem. Considering only objectives 2 and 3, if the Pareto systems switch places on objective 2, one dominates the other. Thus the MCI event probability in which the Pareto systems switch places on objective 2 is bounded below by the MCE event probability. Considering only objectives 1 and 2, system 2 is dominated by system 1.

Despite the potential discrepancies between the rates, the approximation in equation (3) is likely to be good under the limiting score regime, when the number of non-Pareto systems is large relative to the number of Pareto systems. This regime provides so many samples to the Pareto systems that they appear fixed relative to the non-Pareto systems. Thus the phantom Pareto systems also appear fixed, and events in which the Pareto systems are estimated "out of order" are highly unlikely.

Using the approximate rate of decay of $\mathbb{P}\{\text{MC}\}$ in equation (3), we can formulate a new version of Problem Q as Problem Q^{ph}, having solution $\boldsymbol{\alpha}^{\text{ph}}$:

$$\begin{aligned}\text{Problem } Q^{\text{ph}}: \quad & \text{maximize } \ z^{\text{ph}} \ \text{ s.t.} \\ & R^{\text{MCE}}_{i'i}(\alpha_{i'}, \alpha_i) \geq z^{\text{ph}} \text{ for all } i, i' \in \mathcal{P} \text{ such that } i \neq i', \\ & R^{\text{ph}}_{j\ell}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}(\ell)}) \geq z^{\text{ph}} \text{ for all } j \in \mathcal{P}^c, \ell \in \mathcal{P}^{\text{ph}}, \\ & \textstyle\sum_{s=1}^{r} \alpha_s = 1, \ \alpha_s \geq 0 \ \text{ for all } s \in \mathcal{S}.\end{aligned}$$

Since calculating the truly optimal allocation via Problem Q is often difficult, in the numerical sections (§8 and §10), we often approximate the rate of decay of $\mathbb{P}\{\text{MC}\}$ using equation (3) and approximate the asymptotically optimal allocation by solving Problem Q^{ph}.

7 The SCORE Allocation Framework

We are now ready to present the SCORE allocation framework. In this framework, we calculate the relative allocations to the non-Pareto systems using the scores in Theorem 3. We also calculate the rate of decay of $\mathbb{P}\{\text{MCI}\}$ using the phantom approximation in equation (2). Then, we approximate Problem Q^{ph} by strategically dropping constraints to reduce the computational complexity. Finally, we formulate the iSCORE allocation for large problems with four or more objectives.

7.1 The SCORE Allocation for Three Objectives or Small Problems

To begin, first, we implement the relative allocations to the non-Pareto systems specified by the scores in Theorem 3. For all non-Pareto systems $j \in \mathcal{P}^c$, define $\lambda^{\mathbb{S}}_j := \mathbb{S}_j^{-1} / \sum_{j' \in \mathcal{P}^c} \mathbb{S}_{j'}^{-1}$, and let

$\alpha_j = \lambda_j^{\mathbb{S}}(1 - \sum_{i=1}^{p} \alpha_i)$ be the allocation to non-Pareto system j as a function of the allocation to the Pareto systems. Everywhere an α_j appears in Problem Q^{ph}, we substitute this allocation.

To dramatically reduce the number of constraints in Problem Q^{ph}, first, we strategically drop constraints corresponding to $\mathbb{P}\{\text{MCI}\}$. To keep only the most relevant constraints, for each phantom Pareto system, we create a special set of non-Pareto systems, $\mathcal{J}^*(\ell)$, that are most likely to falsely exclude phantom ℓ. To create this set, first, for each phantom Pareto system, notice that there are at most d Pareto systems that contribute objective values to the phantom. Thus for each phantom Pareto system $\ell \in \mathcal{P}^{\text{ph}}$ and each Pareto system $i \in \mathcal{P}(\ell)$ that contributes objective value $k^*(i)$ to phantom Pareto system ℓ, calculate $j_i^*(\ell) = \operatorname{argmin}_{j \in \mathcal{P}^c}\big\{\mathbb{S}_j(\ell) : \mathbb{S}_j(\ell) \neq \inf_{x_{jk} \leq g_k^{\text{ph}}(\ell)\ \forall k \neq k^*(i)}\big(\inf_{x_{jk^*(i)}} I_j(\mathbf{x}_j)\big)\big\}$ as the "closest" non-Pareto system that competes with Pareto system i via phantom Pareto system ℓ. Then $\mathcal{J}^*(\ell) = \cup_{i \in \mathcal{P}(\ell)}\{j_i^*(\ell)\}$ is the set of up to d "closest" non-Pareto systems to phantom Pareto system ℓ; we keep only constraints corresponding to controlling the $\mathbb{P}\{\text{MCI}\}$ decay rate involving these systems.

For further computational speed, we strategically drop constraints corresponding to controlling the $\mathbb{P}\{\text{MCE}\}$ decay rate as well. To see which Pareto system pairs have the highest probabilities of creating MCE events, we define a score for each Pareto system. For all Pareto systems $i \in \mathcal{P}$, define the MCE score as $\mathbb{T}_i := \min_{i' \in \mathcal{P},\, i \neq i'}\ \inf_{\mathbf{x}_i \leq \mathbf{g}(i')} I_i(\mathbf{x}_i)$, and, for notational convenience, define $\mathbb{T}_i(i') := \inf_{\mathbf{x}_i \leq \mathbf{g}(i')} I_i(\mathbf{x}_i)$ for all $i \in \mathcal{P}, i' \in \mathcal{P}, i \neq i'$. Now we select constraints to keep by creating a special set of Pareto systems that are at risk of excluding Pareto system i, $\mathcal{M}^*(i) = \mathcal{M}^1(i) \cup \mathcal{M}^2(i) \cup \mathcal{M}^3(i)$, where given a Pareto system $i \in \mathcal{P}$, each set is defined as follows. First, to define $\mathcal{M}^1(i)$, just as we did for the non-Pareto systems above, we wish to retain constraints for up to d of the "closest" Pareto systems, while ensuring we retain at least one constraint corresponding to a Pareto system i' competing with Pareto system i on each objective k. Then for each objective $k \in \{1, \ldots, d\}$, define ${i'}_k^*(i) := \operatorname{argmin}_{i' \in \mathcal{P}}\big\{\mathbb{T}_{i'}(i)\colon \mathbb{T}_{i'}(i) \neq \inf_{x_{i'k'} \leq g_{k'}(i)\ \forall k' \neq k}\big(\inf_{x_{i'k}} I_{i'}(\mathbf{x}_{i'})\big)\big\}$ as the "closest" Pareto system that competes with Pareto system i on objective k, and let $\mathcal{M}^1(i) := \cup_{k \in \{1,\ldots,d\}}\{{i'}_k^*(i)\}$ be the set of up to d "closest" Pareto systems to Pareto system i. Since the scores in this context may not accurately reflect the true $\mathbb{P}\{\text{MCE}\}$ decay rate and $\mathbb{T}_{i'}(i) \neq \mathbb{T}_i(i')$ for Pareto systems $i, i' \in \mathcal{P}$, we retain symmetric constraints as well. That is, for all $i \in \mathcal{P}$, we let $\mathcal{M}^2(i) := \{i' \in \mathcal{P}\colon i \in \mathcal{M}^1(i'), i \neq i'\}$. Finally, to account for "clusters" of Pareto systems that may influence allocations, we include constraints corresponding to any Pareto systems i' whose scores are less than the 25th percentile p_{25} among the set of Pareto-with-Pareto MCE scores $\{\mathbb{T}_i(i')\colon i, i' \in \mathcal{P}, i \neq i'\}$. Thus we define $\mathcal{M}^3(i) := \{i' \in \mathcal{P}\colon \mathbb{T}_i(i') < p_{25}\}$. Recall that a small score implies the systems are "close," so loosely speaking, we ensure they receive adequate samples by retaining these constraints.

Our SCORE allocation framework results from solving

$$\begin{aligned}\text{Problem } Q_{\mathbb{S}}: \qquad & \text{maximize } \ z \ \text{ s.t.}\\ & R^{\text{MCE}}_{i'i}(\alpha_{i'}, \alpha_i) \geq z \text{ for all } i \in \mathcal{P}, i' \in \mathcal{M}^*(i)\\ & R^{\text{ph}}_{j^*\ell}(\lambda^{\mathbb{S}}_{j^*}(1 - \textstyle\sum_{i=1}^{p} \alpha_i), \boldsymbol{\alpha}_p) \geq z \text{ for all } \ell \in \mathcal{P}^{\text{ph}}, j^* \in \mathcal{J}^*(\ell),\\ & \textstyle\sum_{i=1}^{p} \alpha_i \leq 1, \ \alpha_i \geq 0 \ \text{ for all } i \in \mathcal{P},\end{aligned}$$

and setting $\alpha_j = \lambda^{\mathbb{S}}_j(1 - \sum_{i=1}^{p} \alpha_i)$ for all $j \in \mathcal{P}^c$. When there are three objectives or few systems, our experience with the SCORE allocation framework indicates that modeling dependence between the objectives has a mild implementation cost and also may yield mild benefits in terms of the $\mathbb{P}\{\text{MC}\}$ decay rate. Thus we tend to recommend modeling the dependence in these cases.

7.2 The iSCORE Allocation for Four Objectives and Large Problems

For large problems with four or more objectives, modeling the dependence between the objectives begins to incur some computational cost and reduced benefits in terms of the $\mathbb{P}\{\text{MC}\}$ decay rate. Thus in this section, we outline a further simplification of the SCORE framework that we call the independent SCORE (iSCORE) framework, which models the objectives as if they were independent. Our computational experience is that this framework is much faster to calculate — we must solve only one convex optimization problem and no quadratic programs under our normality assumption.

To approximate the rates in Problem $Q_{\mathbb{S}}$ using an independence assumption, first, notice that the rate of decay of $\mathbb{P}\{\text{MCE}\}$ can be approximated as follows:

$$\begin{aligned}R^{\text{MCE}}_{i'i}(\alpha_{i'}, \alpha_i) &= \inf_{\mathbf{x}_{i'} \leq \mathbf{x}_i} \alpha_i I_i(\mathbf{x}_i) + \alpha_{i'} I_{i'}(\mathbf{x}_{i'}) \approx \inf_{\mathbf{x}_{i'} \leq \mathbf{x}_i} \sum_{k \in \mathcal{D}} \alpha_i J_{ik}(x_{ik}) + \alpha_{i'} J_{i'k}(x_{i'k})\\ &\geq \sum_{k \in \mathcal{D}} \inf_{x_{i'k} \leq x_{ik}} \alpha_i J_{ik}(x_{ik}) + \alpha_{i'} J_{i'k}(x_{i'k}) = \sum_{k \in \mathcal{D}} \left(\tfrac{(g_k(i) - g_k(i'))^2 \mathbb{I}\{g_k(i') > g_k(i)\}}{2(\sigma_k^2(i)/\alpha_i + \sigma_k^2(i')/\alpha_{i'})} \right),\end{aligned}$$

where the last step follows by Glynn and Juneja [2004] under our normality assumption. Similar steps can be used to approximate the rate of decay of $\mathbb{P}\{\text{MCI}\}$. Thus we approximate the rates of decay of the probabilities of MCE and MCI, respectively, as

$$L_{i'i}^{\mathrm{MCE}}(\alpha_{i'},\alpha_i) := \sum_{k\in\mathcal{D}} \left(\frac{(g_k(i)-g_k(i'))^2\mathbb{I}\{g_k(i')>g_k(i)\}}{2(\sigma_k^2(i)/\alpha_i+\sigma_k^2(i')/\alpha_{i'})}\right) \text{ for all } i,i'\in\mathcal{P}, i\neq i',$$

$$L_{j\ell}^{\mathrm{MCI}}(\alpha_j,\boldsymbol{\alpha}_{\mathcal{P}(\ell)}) := \sum_{k\in\mathcal{D}} \left(\frac{(g_k(j)-g_k^{\mathrm{ph}}(\ell))^2\mathbb{I}\{g_k(j)>g_k^{\mathrm{ph}}(\ell)\}}{2(\sigma_k^2(j)/\alpha_j+\sigma_k^2(i_k(\ell))/\alpha_{i_k(\ell)})}\right) \text{ for all } j\in\mathcal{P}^c, \ell\in\mathcal{P}^{\mathrm{ph}};$$

recall that $i_k(\ell)$ is the index of the Pareto system that contributes the kth objective function value to phantom Pareto system ℓ. We also approximate the score calculations using independence. Let

$$\mathbb{S}_j^{\mathrm{indep}}(\ell) := \sum_{k\in\mathcal{D}} \left(\frac{(g_k(j)-g_k^{\mathrm{ph}}(\ell))^2\mathbb{I}\{g_k(j)>g_k^{\mathrm{ph}}(\ell)\}}{2\sigma_k^2(j)}\right), \quad \mathbb{T}_i^{\mathrm{indep}}(i') := \sum_{k\in\mathcal{D}} \left(\frac{(g_k(i)-g_k(i'))^2\mathbb{I}\{g_k(i)>g_k(i')\}}{2\sigma_k^2(i)}\right),$$

and $\mathbb{S}_j^{\mathrm{indep}} := \min_{\ell\in\mathcal{P}^{\mathrm{ph}}} \mathbb{S}_j^{\mathrm{indep}}(\ell)$. We form reduced constraint sets that are identical to $\mathcal{J}^*(\ell)$ and $\mathcal{M}^*(i)$, except that we use $\mathbb{S}_j^{\mathrm{indep}}(\ell)$ and $\mathbb{T}_i^{\mathrm{indep}}(i')$; thus we call the iSCORE reduced constraint sets $\mathcal{J}^{\mathrm{indep}}(\ell)$ and $\mathcal{M}^{\mathrm{indep}}(i)$. Then our proposed iSCORE allocation results from solving

$$\begin{aligned} \text{Problem } Q_{\mathbb{S}}^{\mathrm{indep}}: \qquad & \text{maximize } z \text{ s.t.} \\ & L_i^{\mathrm{MCE}}(\alpha_i,\alpha_{i'}) \geq z \text{ for all } i\in\mathcal{P}, i'\in\mathcal{M}^{\mathrm{indep}}(i) \\ & L_{j^*\ell}^{\mathrm{MCI}}(\lambda_{j^*}^{\mathrm{indep}}(1-\textstyle\sum_{i=1}^p \alpha_i),\boldsymbol{\alpha}_{\mathcal{P}(\ell)}) \geq z \text{ for all } \ell\in\mathcal{P}^{\mathrm{ph}}, j^*\in\mathcal{J}^{\mathrm{indep}}(\ell) \\ & \textstyle\sum_{i=1}^p \alpha_i \leq 1,\ \alpha_i\geq 0 \text{ for all } i\in\mathcal{P}, \end{aligned}$$

and setting $\alpha_j = \lambda_j^{\mathrm{indep}}(1-\sum_{i=1}^p \alpha_i)$ for all $j\in\mathcal{P}^c$, where $\lambda_j^{\mathrm{indep}} = (\mathbb{S}_j^{\mathrm{indep}})^{-1}/\sum_{j'\in\mathcal{P}^c}(\mathbb{S}_{j'}^{\mathrm{indep}})^{-1}$.

8 Time to Compute Proposed Allocations versus Optimality Gap

In this section, we assume we have access to the true rate functions and investigate the time it takes to solve for each proposed allocation on a suite of randomized test problems. When possible, we also investigate how close each allocation is to the asymptotically optimal allocation. The results in this section give us a sense of how long one update of the optimal allocation takes in the sequential implementation described in §9. We compare the following allocation strategies under the normality Assumption 2: (a) MVN True, which results from solving Problem Q and requires computing brute force rates; (b) MVN Indep., which results from solving Problem Q under the assumption of independence between the objectives; (c) MVN Phantom, which results from solving Problem Q^{ph}; (d) SCORE, which results from solving Problem $Q_{\mathbb{S}}$; (e) iSCORE, which results from solving Problem $Q_{\mathbb{S}}^{\mathrm{indep}}$; (f) MOCBA [Lee et al., 2010]; and (g) equal allocation. Solving

Problems Q^{ph}, Problem $Q_{\mathbb{S}}$, and Problem $Q_{\mathbb{S}}^{\mathrm{indep}}$ requires locating the phantom Pareto systems, which we locate using the algorithm in the Online Supplement.

Several of our proposed allocations require solving a bi-level optimization problem where, at each step in the "outer" optimization problem, we solve many quadratic problems that appear in the constraints. To speed up these computations, for the MVN phantom and SCORE allocations with $d = 3$ objectives, we pre-compute a look-up table of closed-form expressions for the solutions to the quadratic programs and feed gradients to the "outer" optimization routine. For $d \geq 4$ objectives and all allocations that require brute force enumeration via the $\boldsymbol{\kappa}$ vectors, our SCORE code solves as many quadratic programs as we have constraints at every step in the "outer" optimization routine, which is considerably slower than the closed-form expressions.

We generate a randomized test problem suite by two different methods, as follows. In the *fixed Pareto method*, we generate p Pareto systems uniformly on a d-sphere of radius 6 at center $100 \times \mathbf{1}_{d\times 1}$. Then, we generate non-Pareto systems by generating points uniformly in the d-ball of radius 6, rejecting any points that are not dominated by the p Pareto systems. In the *variable Pareto method*, we generate systems uniformly inside the d-ball of radius 6 until the desired total number of systems r is achieved. Thus the variable Pareto method results in a random number of Pareto systems. In both methods, as in Assumption 1, we ensure $\min\{|g_k(s) - g_k(i)| \colon s \in \mathcal{S}, i \in \mathcal{P}, s \neq i, k \in \mathcal{D}\} > 1 \times 10^{-4}$. This separation ensures the rate functions are not too shallow for the solver. In each test problem, all systems have multivariate normal rate functions with unit variances and a common correlation between all objectives. To ensure positive semi-definite covariance matrices, the correlation is chosen uniformly at random between -0.4 and 1 for each test problem.

For each number of objectives $d \in \{3, 4, 5\}$ and number of systems r, we generate 10 MORS problems using the fixed and variable Pareto methods. For each set of problems, Tables 6–11 in the Online Supplement report: the median number of Pareto systems p; the median number of phantom Pareto systems $|\mathcal{P}^{\mathrm{ph}}|$; the median and 75th percentiles of the wall-clock time required to solve for each allocation $\boldsymbol{\alpha}$, where the percentiles are taken across each set of random problems; the median brute force rate of decay of the $\mathbb{P}\{\mathrm{MC}\}$, when possible; and the median phantom rate of decay of the $\mathbb{P}\{\mathrm{MC}\}$ across each set of random problems. Tables 4 and 5 report a subset of these results.

We make the following observations about our proposed allocations. First, the relatively small difference in the median rates of decay of $\mathbb{P}\{\mathrm{MC}\}$ for the MVN True, MVN Phantom, and SCORE

Table 4: For 10 MORS problems generated by the fixed Pareto method, each with d objectives and r systems, the table reports: the median number of Paretos and phantoms, sample quantiles of the wall-clock time T to solve for each allocation $\boldsymbol{\alpha}$ in minutes (m) and seconds (s); the median rate of decay of the $\mathbb{P}\{\text{MC}\}$ calculated by brute force ($Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$) or by the phantom approximation ($Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$).

			Med.		MVN	MVN	MVN				
d	r	p	$\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	True	Indep	Phantom	SCORE	iSCORE	MOCBA[†]	Equal
3	10	5	11	Median T	1m 41s	1m 45s	0.05s	0.03s	0.023s	0.005s	0s
				75th %-ile T	1m 55s	2m 2s	0.05s	0.05s	0.026s	0.006s	0s
				$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	950.316	702.155	950.297	924.602	695.678		393.455
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	948.024	702.155	950.297	924.602	695.678		393.455
	500	10	21	Median T	–	–	1m 37s	0.15s	0.09s	3.41s	0s
				75th %-ile T	–	–	2m 38s	0.17s	0.10s	3.45s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.171	0.167	0.170		0.0009
	10,000	10	21	Median T	–	–	–	0.64s	0.31s	25m 6s	0s
				75th %-ile T	–	–	–	0.70s	0.33s	25m 13s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.0004	0.0003		0.0000003
4	5,000	10	42	Median T	–	–	–	1m 4s	0.30s	7m 43s	0s
				75th %-ile T	–	–	–	1m 28s	0.36s	7m 44s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.0013	0.0011		0.000002
	10,000	10	44	Median T	–	–	–	1m 52s	0.57s	31m 5s	0s
				75th %-ile T	–	–	–	2m 20s	0.61s	31m 9s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.00009	0.00009		0.0000001
5	10,000	10	90	Median T	–	–	–	8m 52s	1.15s	38m 16s	0s
				75th %-ile T	–	–	–	11m 13s	1.21s	38m 31s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.0001	0.0001		0.0000002

Computed in MATLAB R2017a on a 3.5 Ghz Intel Core i7 processor with 16GB 2133 MHz LPDDR3 memory.
The symbol '–' indicates no data due to large computational time or memory limitations.
[†] We do not report rates for MOCBA since it alternates between allocations.

allocations indicate that our three primary simplifications, the SCORE limit, the phantom MCI rates, and the reduced number of MCE and MCI constraints in Problem $Q_{\mathbb{S}}$, are good approximations that make larger problem instances more computationally tractable. Further, although the smallest problems in Table 6 in the $r = 10, |\mathcal{P}| = 5$ row suffer a relatively large penalty for modeling the objectives as independent, this penalty seems to decrease as the number of systems increases, as assessed by the median phantom rates for SCORE, iSCORE, and equal allocation in rows where MVN True and MVN Phantom allocations cannot be calculated.

Further, notice that in nearly all 3-objective rows and nearly all rows with problems generated via the fixed Pareto method, the median times for SCORE and iSCORE are clearly faster than those MOCBA, although for small problem instances when computations are fast, SCORE, iSCORE, and

Table 5: For 10 MORS problems generated by the variable Pareto method, each with d objectives and r systems, the table reports: the median number of Paretos and phantoms, the sample quantiles of the wall-clock time T to solve for each allocation $\boldsymbol{\alpha}$ in minutes (m) and seconds (s); the median rate of decay of the $\mathbb{P}\{\text{MC}\}$ calculated by brute force ($Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$) or by the phantom approximation ($Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$).

		Med.	Med.		MVN	MVN	MVN				
d	r	p	$\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	True	Indep	Phantom	SCORE	iSCORE	MOCBA[†]	Equal
3	250	31	62	Median T	–	–	80s	1s	1s	0.93s	0s
				75th %-ile T	–	–	2m 14s	3s	2s	0.94s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.946	0.932	0.714		0.011
	5,000	165	330	Median T	–	–	–	12s	8s	5m 55s	0s
				75th %-ile T	–	–	–	13s	10s	5m 56s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.000008	0.000008		0.0000003
	10,000	245	490	Median T	–	–	–	47s	35s	23m 56s	0s
				75th %-ile T	–	–	–	1m 6s	38s	23m 59s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.00001	0.00001		0.0000002
4	50	21	89	Median T	–	–	6m 3s	56s	0.24s	0.055s	0s
				75th %-ile T	–	–	9m 42s	1m 11s	0.26s	0.056s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	16.659	16.252	12.460		1.269
	1,000	133	686	Median T	–	–	–	–	12s	18.60s	0s
				75th %-ile T	–	–	–	–	53s	18.65s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	–	0.015		0.00006
	2,000	208	1,062	Median T	–	–	–	–	33s	1m 13.3s	0s
				75th %-ile T	–	–	–	–	34s	1m 13.7s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	–	0.00006		0.000004
	5,000*	374	2,064	Median T	–	–	–	–	2m 36s	8m 32s	0s
				75th %-ile T	–	–	–	–	2m 42s	8m 38s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	–	0.00005		0.000002
5	50	30	278	Median T	–	–	18m 18s	4m 38s	2s	0.08s	0s
				75th %-ile T	–	–	21m 44s	23m 6s	7s	0.08s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	7.394	7.326	5.461		0.641
	2,000*	356	5,618	Median T	–	–	–	–	13m 31s	1m 45.2s	0s
				75th %-ile T	–	–	–	–	24m 25s	1m 45.8s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	–	0.0007		0.00002

Computed in MATLAB R2017a on a 3.5 Ghz Intel Core i7 processor with 16GB 2133 MHz LPDDR3 memory. The symbol '–' indicates no data due to large computational time or memory limitations.

[†] We do not report rates for MOCBA since it alternates between allocations.

[*] This row computed in MATLAB R2017a on a high performance computing cluster node with two 10-core Intel Xeon-E5 processors and 128GB of memory.

MOCBA are computationally comparable. Interestingly, in terms of computational time, MOCBA suffers a penalty for a large total number of systems, while iSCORE suffers a penalty only for a large Pareto set. This penalty is especially noticeable in the 5-objective, 2,000-system row of Table 5, in which iSCORE must contend with a median number of phantom Pareto systems equal to 5,618, which require a median time of 8 minutes and 54 seconds to retrieve. These results make sense in light of our complexity results for both the number of phantom Pareto systems and the algorithm that locates them.

9 A Sequential Algorithm for Implementation

Throughout the paper so far, we have assumed that we have access to the rate functions of all systems, which we certainly do not have in practice. In this section, we provide a sequential algorithm, listed as Algorithm 1, that is suitable for implementation of our proposed allocations.

Algorithm 1 is similar in spirit to the sequential allocation algorithm provided by Hunter and Pasupathy [2013], although it differs slightly in the details. Like the previous algorithm, to estimate the rate functions, we use plug-in estimators for the parameters of the assumed distributional family in Step 8 and solve an optimization problem in Step 9. We also use the estimated optimal allocation $\hat{\boldsymbol{\alpha}}^*$ as a probability mass function from which to select the next system to simulate in Steps 13 and 14, and implement a minimum-sample proportion $0 < \alpha_\varepsilon \ll 1/r$ in Step 4 to ensure all systems are sampled infinitely often when the total simulation budget is infinite. We differ slightly in how the minimum-sampling requirement is implemented. In Hunter and Pasupathy [2013], the

Algorithm 1: A sequential algorithm for implementing the proposed allocations

```
   Input: initial sample size $\delta_0 > d \geq 2$; sample size between allocation vector updates $\delta \geq 1$; a
          minimum-sample proportion $0 < \alpha_\varepsilon \ll 1/r$ where $r = |\mathbb{S}|$ is the total number of systems and
          $\mathbb{S}$ is the set of system indices; total simulation budget $b \geq r \times \delta_0 + \delta$
1  Initialize: collect $\delta_0$ replications from each system $s \in \mathbb{S}$; $n \leftarrow r \times \delta_0, n_s \leftarrow \delta_0$ for all $s \in \mathbb{S}$
2  repeat
3      Initialize: $\delta_\varepsilon = 0$, $\mathbb{S}_\varepsilon \leftarrow \emptyset$, $\mathbb{S} \leftarrow \{1, \ldots, r\}$
4      foreach $s \in \mathbb{S}$ if $n_s/n < \alpha_\varepsilon$ then $\mathbb{S}_\varepsilon \leftarrow \mathbb{S}_\varepsilon \cup \{s\}$          /find systems needing simulation
5      if $0 \leq |\mathbb{S}_\varepsilon| < \delta$ then
6          if $|\mathbb{S}_\varepsilon| \geq 1$ then foreach $s_\varepsilon \in \mathbb{S}_\varepsilon$ do
7              collect one simulation replication from system $s_\varepsilon$, $n_{s_\varepsilon} \leftarrow n_{s_\varepsilon} + 1$, $\delta_\varepsilon \leftarrow \delta_\varepsilon + 1$
8          Calculate: update the rate function estimators for all systems $s \in \mathbb{S}$ by updating plug-in
            estimators for parameters in the assumed distributional family
9          Solve: an estimated version of Problem $Q^{\text{ph}}$, $Q_{\mathbb{S}}$, or $Q_{\mathbb{S}}^{\text{indep}}$, to obtain estimated allocation $\hat{\boldsymbol{\alpha}}^*$
10     else
11         $\mathbb{S} \leftarrow \mathbb{S}_\varepsilon$, $\hat{\boldsymbol{\alpha}}^* \leftarrow (1/|\mathbb{S}_\varepsilon|, \ldots, 1/|\mathbb{S}_\varepsilon|)$          /simulate only systems in $\mathbb{S}_\varepsilon$ with equal pr.
12     for $m = 1, \ldots, \delta - \delta_\varepsilon$ do                              /spend $\delta - \delta_\varepsilon$ replications left
13         Sample: randomly select a system index $X_m$ from $\mathbb{S}$, where for each $m$, $X_m$ is an i.i.d.
            random variable with probability mass function $\hat{\boldsymbol{\alpha}}^*$ supported on $\mathbb{S}$
14         Simulate: collect one simulation replication from system $X_m$ and $n_{X_m} \leftarrow n_{X_m} + 1$
15     Update: $n \leftarrow n + \delta$ and $\bar{\boldsymbol{\alpha}}_n \leftarrow (n_1/n, n_2/n, \ldots, n_r/n)$
16 until $n \geq b$ or other termination criteria met
```

sequential algorithm is written as non-terminating, and any systems not meeting the minimum-sampling requirement are simulated beyond the "stage-wise" sampling budget δ. We write our algorithm with optional termination criteria and ensure that the minimum-sampling requirement is met within the stage-wise sampling budget δ. Thus the algorithm is easier to terminate at a specific, known simulation budget, designed with multiples of δ in mind. Optionally, the algorithm may be terminated by other criteria, such as a certain amount of wall-clock time having passed.

10 Numerical Performance of the Sequential Implementations

In this section, we implement our proposed simulation budget allocations using sequential Algorithm 1, and compare their performances with MOCBA and equal allocation on two test problems.

10.1 Test Problem 1

Our first test problem, from Lee et al. [2010], has three objectives, 25 systems, 5 Pareto systems, and appears in Figure 4. The objective vector values are listed in Table 12 of the Online Supplement. We consider three versions of Test Problem 1: a version in which all objectives are independent and the covariance matrix for all systems $s \in \mathcal{S}$ is $\Sigma(s) = 64 \times \mathbb{I}_{3\times 3}$, where $\mathbb{I}_{3\times 3}$ is a 3-by-3 identity matrix, and two versions in which the variances are the same as the independent case, but there is common correlation of $\rho = -0.4$ and $\rho = 0.8$ between all objectives across all systems.

We implement our sequential Algorithm 1 and MOCBA with the following parameters, respectively. In Algorithm 1, we set the initial sample size to $\delta_0 = 5$, the number of samples between allocation updates to $\delta = 10$, and the minimum-sample proportion $\alpha_\varepsilon = 1 \times 10^{-8} \ll 1/r$, and the total simulation budget to $b \leq 75,000$. (Notice that given our values of δ_0 and b, extra sampling due to the minimum-sample proportion α_ε does not occur.) For MOCBA, using notation from Lee et al. [2010], we set the number of initial samples $N_0 = 5$, the number of samples between allocation updates to $\Delta = 10$, and the maximum samples to a single system $\tau = \Delta/2 = 5$.

For each allocation scheme, let $\bar{\boldsymbol{\alpha}}_n = (n_1/n, n_2/n, \ldots, n_r/n)$ denote the vector of proportional allocations expended by the sequential algorithm as a function of the sample size n. Figure 5 shows sample quantiles of the approximate optimality gap of $\bar{\boldsymbol{\alpha}}_n$, $z^{\text{ph}}(\boldsymbol{\alpha}^{\text{ph}}) - z^{\text{ph}}(\bar{\boldsymbol{\alpha}}_n)$, calculated across 5,000 independent replications of each sequential algorithm on Test Problem 1 with $\rho = 0$. There is dependence across the values of n in Figure 5. From the perspective of the optimality gap of

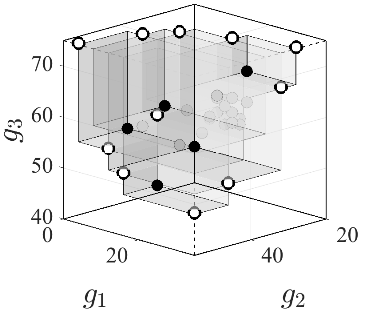

Figure 4: Test Problem 1: For $d = 3$, the figure shows 5 Paretos in black, 11 phantom Paretos in white (arrows omitted), and 20 non-Paretos in gray.

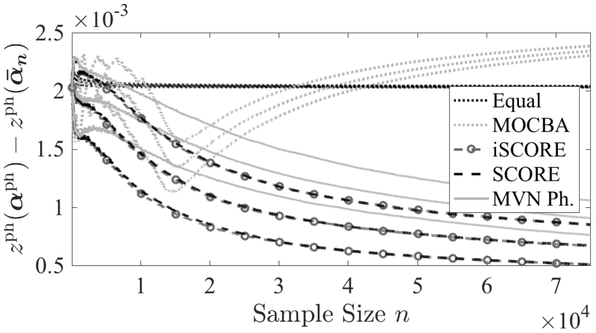

Figure 5: Test Problem 1, $\rho = 0$: Sample quantiles (.25, .5, .75) of the optimality gap over 5,000 independent runs of each sequential allocation strategy.

the allocation expended, SCORE and iSCORE appear to perform the best. MOCBA appears to perform well initially, but eventually veers off into an allocation that is worse than equal. We believe this event occurs because at some point, the bounds that MOCBA uses to determine whether to control for $\mathbb{P}\{\text{MCE}\}$ or $\mathbb{P}\{\text{MCI}\}$ are both estimated to be zero, to the numerical precision of the computer. By default in this case, MOCBA allocates to control $\mathbb{P}\{\text{MCE}\}$ only, which is suboptimal with respect to the phantom rate in this test problem.

Figure 6 shows the estimated probability of misclassification as a function of n for each sequential allocation strategy, calculated across 10,000 independent replications of each algorithm. Notably, SCORE and iSCORE perform nearly identically to MVN Phantom. SCORE, iSCORE, and MVN Phantom slightly out-perform MOCBA in terms of the overall estimated $\mathbb{P}\{\text{MC}\}$. In this test problem, correlation has a minor effect on the estimated $\mathbb{P}\{\text{MC}\}$. Finally, we remark that by the time the allocation scheme for MOCBA goes awry in Figure 5, the estimated $\mathbb{P}\{\text{MC}\}$ is already very small for MOCBA, SCORE, and iSCORE — nearly zero, as shown in Figure 6.

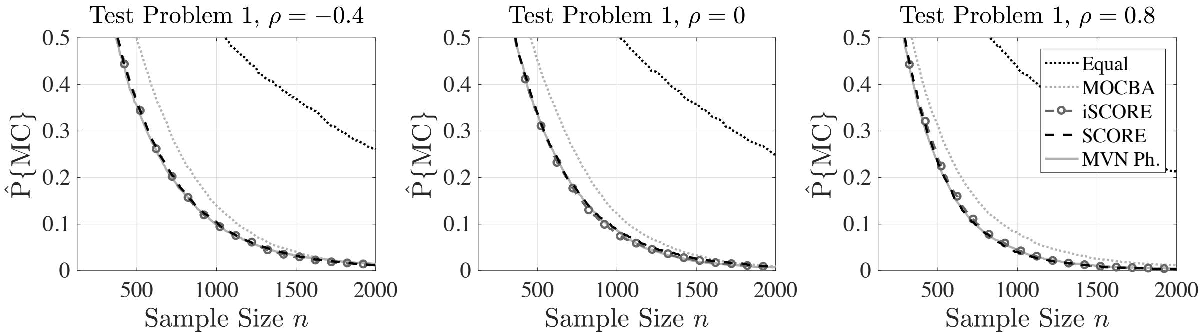

Figure 6: Test Problem 1, with $\rho = -0.4, \rho = 0$, and $\rho = 0.8$: Estimated $\mathbb{P}\{\text{MC}\}$ for sequential allocation strategies, calculated across 10,000 independent sample paths for each algorithm.

10.2 Test Problem 2

Our second test problem has four objectives, created by generating 500 true system objective vector values as a multivariate normal cloud with center $100 \times \mathbf{1}_{4\times 1}$, all standard deviations equal to 10, and all correlations equal to 0.5. The generated cloud is shown in three out of four objectives in Figure 7; all other three-objective projections look similar. The minimum distance between any two Pareto systems on any objectives is approximately 0.0953, and the minimum distance between a Pareto system and a non-Pareto system on any objective is approximately 0.0129. Fixing the 500 systems with objective vector values shown in Figure 7, we set all systems' covariance matrices to the identity matrix. Due to the size of Test Problem 2, we implement only iSCORE and MOCBA with the following parameters, respectively: iSCORE has $\delta_0 = 15$, $\delta = 150$, $\alpha_\varepsilon = 1 \times 10^{-8} \ll 1/r$, and total simulation budget $b \leq 15,000$; MOCBA has $N_0 = 15$, $\Delta = 150$, and $\tau = \Delta/2 = 75$. As in Test Problem 1, extra sampling due to α_ε does not occur in iSCORE.

Figures 8 and 9 display the results for Test Problem 2. On this test problem, Figure 8 shows that the median optimality gap of the actual allocation for iSCORE is consistently smaller than that of MOCBA. Notice that the sample size is not as large as for Test Problem 1, so it is not clear if MOCBA turns suboptimal for larger n. The 75th percentile line for iSCORE appears to level off slightly for larger sample sizes; we believe this performance is due to several bad sample paths that would eventually be corrected by forcing samples to certain systems via the minimum-sample vector α_ε. Figure 9 shows that MOCBA and iSCORE are close in terms of the estimated $\mathbb{P}\{\text{MC}\}$, but iSCORE appears to perform slightly better.

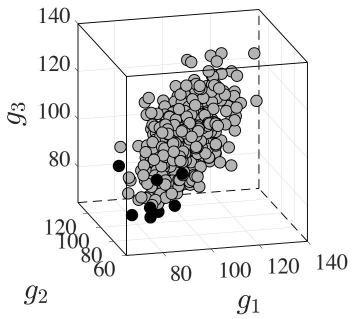

Figure 7: Test Problem 2: For $d = 4$ total objectives, the figure shows objectives 1, 2, and 3 with 8 Paretos in black.

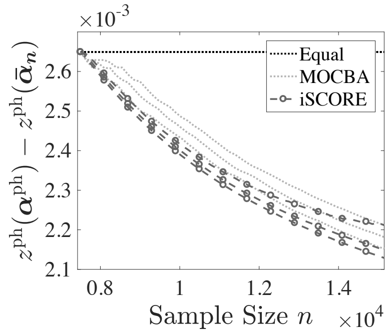

Figure 8: Test Problem 2: Sample quantiles (.25, .5, .75) of the optimality gap over 10,000 independent runs per algorithm.

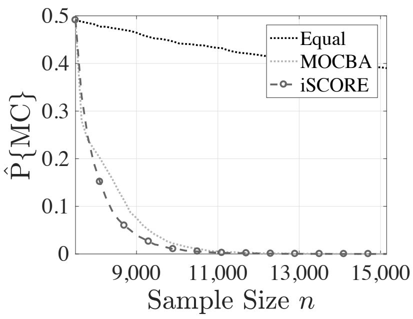

Figure 9: Test Problem 2: Estimated $\mathbb{P}\{\text{MC}\}$, calculated across 10,000 independent runs per algorithm.

11 Concluding Remarks

The question of efficiently identifying the entire Pareto set in MORS problems is challenging primarily due to the need to decide how to dedicate a given simulation budget across the competing systems so that the likelihood of misclassifying the systems is minimized. We demonstrate that this latter question of simulation budget allocation can be posed (without approximation) as a concave maximization problem through the introduction of fictitious systems constructed by combining objectives from selected systems. Solving this concave maximization problem, however, becomes prohibitively expensive even for MORS problems in three and four dimensions having a modest number of competing systems. This computational issue becomes pronounced during implementation, when the simulation budget allocation optimization problem needs to be solved repeatedly across iterations. SCORE and iSCORE are MORS solution algorithms that address the computational issue in a disciplined way. Through a series of approximations obtained by the strategic relaxation of constraints and asymptotic approximation, the simulation budget allocation optimization problem is reduced to a form that can be solved with dramatically less computational effort. Extensive numerical implementation reveals that SCORE and iSCORE can reliably solve many MORS problems with several thousand systems in three or greater number of dimensions.

Acknowledgments

E. A. Applegate and S. R. Hunter were supported in part by the National Science Foundation under grant CMMI-1554144.

References

D. Batur and F. F. Choobineh. Mean-variance based ranking and selection. In B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, editors, *Proceedings of the 2010 Winter Simulation Conference*, Piscataway, NJ, 2010. IEEE.

J. Branke and W. Zhang. A new myopic sequential sampling algorithm for multi-objective problems. In L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, editors, *Proceedings of the 2015 Winter Simulation Conference*, pages 3589–3598, Piscataway, NJ, 2015. IEEE.

J. Branke, W. Zhang, and Y. Tao. Multiobjective ranking and selection based on hypervolume. In T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, editors, *Proceedings of the 2016 Winter Simulation Conference*, pages 859–870, Piscataway, NJ, 2016. IEEE.

J. C. Butler, D. J. Morrice, and P. Mullarkey. A multiple attribute utility theory approach to ranking and selection. *Management Science*, 47(6):800–816, 2001.

T. M. Chan. Klee's measure problem made easy. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 410–419, Piscataway, NJ, 2013. IEEE. doi: 10.1109/FOCS.2013.51.

C.-H. Chen, J. Lin, E. Yücesan, and S. E. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, 10(3):251–270, 2000. doi: 10.1023/A:1008349927281.

S. E. Chick, J. Branke, and C. Schmidt. Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, 22(1):71–80, 2010.

S. H. Choi and T. G. Kim. Pareto set selection for multiobjective stochastic simulation model. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018. doi: 10.1109/TSMC.2018.2846680.

A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer, New York, 2nd edition, 1998.

H. Ding, L. Benyoucef, and X. Xie. A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. *Engineering Applications of Artificial Intelligence*, 19:609–623, 2006. doi: 10.1016/j.engappai.2005.12.008.

E. J. Dudewicz and V. S. Taneja. Multivariate ranking and selection without reduction to a univariate problem. In H. J. Highland, N. R. Nielsen, and L. G. Hull, editors, *Proceedings of the 1978 Winter Simulation Conference*, pages 207–210, Piscataway, NJ, 1978. IEEE.

E. J. Dudewicz and V. S. Taneja. A multivariate solution of the multivariate ranking and selection problem. *Communications in Statistics – Theory and Methods*, 10(18):1849–1868, 1981. doi: 10.1080/03610928108828154.

M. Ehrgott. *Multicriteria Optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Heidelberg, 2nd edition, 2005.

G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer, Berlin Heidelberg, 2008.

G. Feldman. *Sampling laws for multi-objective simulation optimization on finite sets*. PhD thesis, Purdue University, 2017.

G. Feldman and S. R. Hunter. SCORE allocations for bi-objective ranking and selection. *ACM Transactions on Modeling and Computer Simulation*, 28(1):7:1–7:28, January 2018. doi: 10.1145/3158666.

G. Feldman, S. R. Hunter, and R. Pasupathy. Multi-objective simulation optimization on finite sets: optimal allocation via scalarization. In L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, editors, *Proceedings of the 2015 Winter Simulation Conference*, pages 3610–3621, Piscataway, NJ, 2015. IEEE. doi: 10.1109/WSC.2015.7408520.

P. I. Frazier. A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operations Research*, 62(4):926–942, 2014. doi: 10.1287/opre.2014.1282.

P. I. Frazier and A. M. Kazachkov. Guessing preferences: a new approach to multi-attribute ranking and selection. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the 2011 Winter Simulation Conference*, pages 4324 – 4336, Piscataway, NJ, 2011. IEEE.

P. I. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optim.*, 47(5):2410–2439, 2008.

M. Fu and S. G. Henderson. History of seeking better solutions, aka simulation optimizaiton. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, editors, *Proceedings of the 2017 Winter Simulation Conference*, pages 131–157, Piscataway, NJ, 2017. IEEE.

P. Glynn and S. Juneja. Ordinal optimization – empirical large deviations rate estimators, and stochastic multi-armed bandits, 2015. URL http://arxiv.org/abs/1507.04564.

P. W. Glynn and S. Juneja. A large deviations perspective on ordinal optimization. In R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, editors, *Proceedings of the 2004 Winter Simulation Conference*, pages 577–585, Piscataway, NJ, 2004. IEEE. doi: 10.1109/WSC.2004.1371364.

P. W. Glynn and S. Juneja. Ordinal optimization: a nonparametric framework. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the 2011 Winter Simulation Conference*, pages 4057 – 4064, Piscataway, NJ, 2011. IEEE. doi: 10.1109/WSC.2011.6148095.

S. R. Hunter. *Sampling laws for stochastically constrained simulation optimization on finite sets*. PhD thesis, Virginia Polytechnic Institute and State University, 2011.

S. R. Hunter and G. Feldman. Optimal sampling laws for bi-objective simulation optimization on finite sets. In L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, editors, *Proceedings of the 2015 Winter Simulation Conference*, pages 3749–3757, Piscataway, NJ, 2015. IEEE. doi: 10.1109/WSC.2015.7408532.

S. R. Hunter and B. McClosky. Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation. *IIE Transactions*, 48(6):565–578, 2016. doi: 10.1080/0740817X.2015.1096430.

S. R. Hunter and B. L. Nelson. Parallel ranking and selection. In A. Tolk, J. Fowler, G. Shao, and E. Yücesan, editors, *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*, Simulation Foundations, Methods and Applications, chapter 12, pages 249–275. Springer International, Switzerland, 2017. doi: 10.1007/978-3-319-64182-9.

S. R. Hunter and R. Pasupathy. Large-deviation sampling laws for constrained simulation optimization on finite sets. In B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, editors, *Proceedings of the 2010 Winter Simulation Conference*, pages 995–1002, Piscataway, NJ, 2010. IEEE. doi: 10.1109/WSC.2010.5679092.

S. R. Hunter and R. Pasupathy. Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS Journal on Computing*, 25(3):527–542, Summer 2013. doi: 10.1287/ijoc.1120.0519.

S. R. Hunter, N. A. Pujowidianto, C.-H. Chen, L. H. Lee, R. Pasupathy, and C. M. Yap. Optimal sampling laws for constrained simulation optimization on finite sets: the bivariate normal case. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the 2011 Winter Simulation Conference*, pages 4294–4302, Piscataway, NJ, 2011. IEEE. doi: 10.1109/WSC.2011.6148116.

S. R. Hunter, E. A. Applegate, V. Arora, B. Chong, K. Cooper, O. Rincón-Guevara, and C. Vivas-Valencia. An introduction to multi-objective simulation optimization. *Optimization Online*, 2018. URL http://www.optimization-online.org/DB_HTML/2017/03/5903.html.

H. Kaplan, N. Rubin, M. Sharir, and E. Verbin. Efficient colored orthogonal range counting. *SIAM J. Comput.*, 38(3):982–1011, 2008. doi: 10.1137/070684483.

S.-H. Kim and B. L. Nelson. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11(3):251–273, 2001.

S.-H. Kim and B. L. Nelson. Selecting the best system. In S. G. Henderson and B. L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science, Volume 13, pages 501–534. Elsevier, Amsterdam, The Netherlands, 2006.

R. Lacour, K. Klamroth, and C. M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79:347–360, March 2017. doi: 10.1016/j.cor.2016.06.021.

J. S. Lee. *Advances in simulation: validity and efficiency*. PhD thesis, Georgia Institute of Technology, 2014. URL http://hdl.handle.net/1853/53457.

L. H. Lee, E. P. Chew, S. Teng, and D. Goldsman. Finding the non-dominated Pareto set for multi-objective simulation models. *IIE Transactions*, 42:656–674, 2010. doi: 10.1080/07408171003705367.

J. Li. *Optimal computing budget allocation for multi-objective simulation optimization*. PhD thesis, National University of Singapore, 2012.

J. Li, W. Liu, G. Pedrielli, L. H. Lee, and E. P. Chew. Optimal computing budget allocation to select the non-dominated systems – a large deviations perspective. *IEEE Transactions on Automatic Control*, 2018. doi: 10.1109/TAC.2017.2779603.

V. Mattila and K. Virtanen. Ranking and selection for multiple performance measures using incomplete preference information. *European Journal of Operational Research*, 242:568–579, 2015. doi: 10.1016/j.ejor.2014.10.028.

J. R. W. Merrick, D. Morrice, and J. C. Butler. Using multiattribute utility theory to avoid bad outcomes by focusing on the best systems in ranking and selection. *Journal of Simulation*, 9(3):238–248, 2015. doi: 10.1057/jos.2014.34.

B. L. Nelson, J. Swann, D. Goldsman, and W. Song. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research*, 49(6):950–963, 2001.

E. C. Ni, D. F. Ciocan, S. G. Henderson, and S. R. Hunter. Efficient ranking and selection in parallel computing environments. *Operations Research*, 65(3):821–836, May-June 2017. doi: 10.1287/opre.2016.1577.

R. Pasupathy, S. R. Hunter, N. A. Pujowidianto, L. H. Lee, and C.-H. Chen. Stochastically constrained ranking and selection via SCORE. *ACM Transactions on Modeling and Computer Simulation*, 25(1):1:1–1:26, January 2015. doi: 10.1145/2630066.

N. A. Pujowidianto, S. R. Hunter, R. Pasupathy, L. H. Lee, and C.-H. Chen. Closed-form sampling laws for stochastically constrained simulation optimization on large finite sets. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 168–177, Piscataway, NJ, 2012. IEEE. doi: 10.1109/WSC.2012.6465141.

I. O. Ryzhov. On the convergence rates of expected improvement methods. *Operations Research*, 64(6): 1515–1528, 2016. doi: 10.1287/opre.2016.1494.

R. Szechtman and E. Yücesan. A new perspective on feasibility determination. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 273–280, Piscataway, NJ, 2008. IEEE. doi: 10.1109/WSC.2008.4736078.

S. Teng, L. H. Lee, and E. P. Chew. Integration of indifference-zone with multi-objective computing budget allocation. *European Journal of Operational Research*, 203:419–429, 2010. doi: 10.1016/j.ejor.2009.08.008.

W. Wang and H. Wan. Sequential probability ratio test for multiple-objective ranking and selection. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, editors, *Proceedings of the 2017 Winter Simulation Conference*, pages 1998–2009, Piscataway, NJ, 2017. IEEE. doi: 10.1109/WSC.2017.8247934.

M. M. Wiecek, M. Ehrgott, and A. Engau. Continuous multiobjective programming. In S. Greco, M. Ehrgott, and J. R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 233 of *International Series in Operations Research & Management Science*, pages 739–815. Springer New York, New York, 2016. doi: 10.1007/978-1-4939-3094-4_18.

H. Yildiz and S. Suri. On Klee's measure problem for grounded boxes. In *Proceedings of the Twenty-Eighth Annual Symposium on Computational Geometry*, pages 111–1120, New York, NY, 2012. ACM. doi: 10.1145/2261250.2261267.

H. Zhang. Multi-objective simulation-optimization for earthmoving operations. *Automation in Construction*, 18:79–86, 2008. doi: 10.1016/j.autcon.2008.05.0023.

Online Supplement for Ranking and Selection with Many Objectives: Optimal Sampling Laws and Tractable Approximations via SCORE

Eric A. Applegate[1], Guy Feldman[2], Susan R. Hunter[1], and Raghu Pasupathy[2]

[1]School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA
[2]Department of Statisics, Purdue University, West Lafayette, IN 47907, USA

A Efficiently Locating the Phantom Pareto Systems

To solve for our proposed allocations, we require a way to identify the phantom Pareto systems; preferably, we would do so without using brute force enumeration. That is, we would like to know how to identify all phantom Pareto systems implicit to a set of Pareto objective vectors $\mathcal{G}^* = \{\mathbf{g}(1), \mathbf{g}(2), \ldots, \mathbf{g}(p)\}, \mathbf{g}(i) \in \mathbb{R}^d$ for all $i \in \{1, \ldots, p\}$ of p non-dominated points in d-dimensional Euclidean space. Towards answering this question in our context, we present pair of algorithms called Sweep and DimensionSweep, listed as Algorithms 2 and 3, respectively, that resemble the procedure described in Kaplan et al. [2008].

Sweep identifies "interior" phantom Pareto systems, that is, phantom Pareto systems whose coordinates are all finite, through strategic and recursive projection onto lower dimensional hyperplanes that are orthogonal to the axes. The specific set of operations that lead to the identification of the interior phantom Pareto systems is as follows. Consider a set of d'-dimensional points $\mathcal{G}^* = \{\mathbf{g}(1), \mathbf{g}(2), \ldots, \mathbf{g}(p')\}$ with $\mathbf{g}(i) \in \mathbb{R}^{d'}$ for each $i \in \{1, \ldots, p'\}$, where $1 \leq d' \leq d$ and $1 \leq p' \leq p$. If $d' = 1$, the set of phantoms is simply $\min_{1 \leq i \leq p'} g_1(i)$, and the procedure terminates. If $d' > 1$, then select an arbitrary dimension $k^* \in \{1, 2, \ldots, d'\}$ and sort the points in $\mathcal{G}^*$ in decreasing order by their values on the k^*-th objective. By convention, we select $k^* = d'$, and let the resulting ordered set be denoted $\mathcal{G}_{d'} = \{\mathbf{g}([1]), \mathbf{g}([2]), \ldots, \mathbf{g}([p'])\}$, where $[1] = \operatorname{argmax}_{1 \leq i \leq p'} g_{k^*}(i)$ denotes the index of the system with the largest objective value on objective k^*. Assume, for ease of exposition, that the points $\{\mathbf{g}([1]), \mathbf{g}([2]), \ldots, \mathbf{g}([p'])\}$ have distinct values along the k^*-th dimension. Now consider the $(d'-1)$-dimensional hyperplanes $\mathcal{Y}(i) := \{\mathbf{y} \in \mathbb{R}^{d'-1} \colon y_{k^*} = g_{k^*}([i])\}, i = 1, 2, \ldots, p'$, each of which is orthogonal to the k^*-th axis. For each $i = 1, 2, \ldots, p'$, project the $p' - i$ points $\{\mathbf{g}([i+1]), \mathbf{g}([i+2]), \ldots, \mathbf{g}([p'])\}$ onto the $(d-1)$-dimensional hyperplane $\mathcal{Y}(i)$, and calculate the Pareto points to get a new ordered set $\mathcal{G}^*_{d'-1}$ containing up to $p' - i$ Pareto systems, each lying

Corresponding author: susanhunter@purdue.edu.

in $d-1$ dimensional Euclidean space. Now repeat the described procedure with each input set $\mathcal{G}^*_{d'-1} \subset \mathbb{R}^{d'-1}, i = 1, 2, \ldots, p'$, in turn yielding several projected sets in $(d'-2)$-dimensional space. In this way, the process is repeated to yield several sequences of sets projected onto hyperplanes in successively lower dimensions, with the procedure stopping when the incumbent dimension of the input set is 1, at which time the minimum of the input set, augmented with the sequence of projected coordinates, is returned as the potential phantom candidate. A phantom candidate is kept in Step 11 only if it is dominated by the current sweep point. (Note that in the Sweep algorithm Step 6, we iterate up to $p' - (d'-1)$ instead of p' since we need at least $d'-1$ points projected into the hyperplane to make a phantom.)

Algorithm 2: $\mathcal{G}^{\text{ph}}_{\text{sweep}}$=Sweep($\mathcal{G}^*$)

```
   Input: set of points, $\mathcal{G}^* = \{\mathbf{g}(1), \mathbf{g}(2), \ldots, \mathbf{g}(p')\}$, where $\mathbf{g}(i) = (g_1(i), \ldots, g_{d'}(i))$ for all $i = 1, \ldots, p'$.
   Output: a set of $d'$-dimensional phantom Pareto systems $\mathcal{G}^{\text{ph}}_{\text{sweep}}$
1  if $d'$=1 then
2      $\mathcal{G}^{\text{ph}}_{\text{sweep}} \leftarrow \min_{1 \leq i \leq p'} g_1(i)$
3  else
4      $k^* \leftarrow d'$                                   /choose $k^*$ as largest objective
5      Sort the points in $\mathcal{G}^*$ in decreasing order on objective $k^*$, yielding the ordered set
         $\mathcal{G}_{d'} \leftarrow \{\mathbf{g}([1]), \mathbf{g}([2]), \ldots, \mathbf{g}([p'])\}$, where $[1] = \text{argmax}_{1 \leq i \leq p'} g_{k^*}(i), \ldots, [p'] = \text{argmin}_{1 \leq i \leq p'} g_{k^*}(i)$.
6      for $i = 1$ to $p' - (d'-1)$ do
7          Initialize $g_{\max} \leftarrow g_{k^*}([i])$ and $\mathcal{G}_{d'} \leftarrow \{\mathbf{g}([i+1]), \mathbf{g}([i+2]), \ldots, \mathbf{g}([p'])\}$
8          $\mathcal{G}_{d'-1} \leftarrow \{\mathbf{g}'(j) \colon \mathbf{g}'(j) = (g_1(j), \ldots, g_{d'-1}(j))$ for all $j$ indexing points in $\mathcal{G}_{d'}\}$
9          $\mathcal{G}^*_{d'-1}$ =GetParetos($\mathcal{G}_{d'-1}$)
10         $\mathcal{G}^{\text{ph}}_{d'-1}$ =Sweep($\mathcal{G}^*_{d'-1}$)              /points in $\mathcal{G}^{\text{ph}}_{d'-1}$ are $(d'-1)$-dimensional phantoms
11         $\mathcal{G}^{\text{ph}}_{d'-1} \leftarrow \mathcal{G}^{\text{ph}}_{d'-1} \setminus \{\mathbf{g} \in \mathcal{G}^{\text{ph}}_{d'-1} : (g_1([i]), \ldots, g_{d'-1}([i])) \nleq \mathbf{g}\}$
12         $\mathcal{G}^{\text{ph}}_{d'} \leftarrow \{\mathbf{g}'(j) : \mathbf{g}'(j) = (g_1(j), \ldots, g_{d'-1}(j), g_{\max})$ for all $j$ indexing points in $\mathcal{G}^{\text{ph}}_{d'-1}\}$
13         $\mathcal{G}^{\text{ph}}_{\text{sweep}} \leftarrow \mathcal{G}^{\text{ph}}_{\text{sweep}} \cup \mathcal{G}^{\text{ph}}_{d'}$
14 return $\mathcal{G}^{\text{ph}}_{\text{sweep}}$
```

The Sweep procedure as listed in Algorithm 1 identifies all phantoms in a given finite set $\mathcal{G}^*$ that is a subset of the d'-dimensional Euclidean space. Recall, however, that the procedure identifies only phantoms whose coordinates are all finite. In other words, if the set $\mathcal{G}^*$ has a phantom that has $i < d$ coordinates equal to infinity, then such a phantom needs to be identified by executing the Sweep procedure with points constructed using the appropriate $d' = k - i$ coordinates. Identifying *all* phantoms of a given set of Pareto points, each of which is in d-dimensional Euclidean space, thus entails executing the Sweep procedure with all possible combinations of points constructed from subsets of the d-coordinate choices. Such repeated calling of the Sweep procedure with all possible combinations of points constructed from subsets of the d-coordinate choices is performed using the "driver" procedure DimensionSweep, listed in Algorithm 3.

Algorithm 3: $\mathcal{G}^{\text{ph}}$=DimensionSweep($\mathcal{G}^*$)

```
Input: set of $d$-dimensional Pareto objective vectors $\mathcal{G}^*$
Output: the set of objective vectors corresponding to the phantom Pareto points $\mathcal{G}^{\text{ph}}$
Initialize $\mathcal{G}^{\text{ph}} = \emptyset$ and determine number of Pareto points $p$ and number of dimensions $d$ from $\mathcal{G}$
for $i = 1$ to $d$ do
    Projections $m \leftarrow \binom{d}{i}$                    /i is the number of finite objectives in a phantom
    Determine the $m$ combinations of $i$-dimensional indices, store as $C_1, C_2, \ldots, C_m$
    for $j = 1$ to $m$ do
        Reduce points in $\mathcal{G}^*$ to dimensions of $C_j$, store in set $\mathcal{A}$
        $\mathcal{A}^*$ =GetParetos($\mathcal{A}$)
        $\mathcal{A}_i^{\text{ph}}$ =Sweep($\mathcal{A}^*$)
        Append $\infty$ to dimensions not in $C_j$ for points in $\mathcal{A}_i^{\text{ph}}$ to create phantoms $\mathcal{A}^{\text{ph}}$
        $\mathcal{G}^{\text{ph}} = \mathcal{G}^{\text{ph}} \cup \mathcal{A}^{\text{ph}}$
return $\mathcal{G}^{\text{ph}}$
```

Due to the similarity of the DimensionSweep procedure with the procedure outlined in Kaplan et al. [2008], we omit a formal proof of the assertion that the DimensionSweep procedure, aided crucially by the Sweep procedure, identifies all phantom Pareto systems associated with a given finite set of Pareto systems.

B Time to Compute versus Optimality Gap: Supplemental Tables

In this section, we provide tables that show the complete numerical results from the experiments described in §8. For all tables in this section, the metrics reported in the tables include the median number of Pareto systems; the median number of phantom Pareto systems; the median and 75th percentile wall-clock time T to solve for each allocation $\boldsymbol{\alpha}$ in hours (h), minutes (m), and seconds (s); the median $\mathbb{P}\{\text{MC}\}$ decay rate calculated by brute force ($Z_{0.5}^{\text{bf}}(\boldsymbol{\alpha})\times 10^4$), where available, and by the phantom approximation ($Z_{0.5}^{\text{ph}}(\boldsymbol{\alpha})\times 10^4$) otherwise. (If we are not able to calculate the brute force rate, then this row is omitted from the table.) Further, the following notes apply to all tables in this section: (a) Unless indicated otherwise, all computation was performed in MATLAB R2017a on a laptop with a 3.5 Ghz Intel Core i7 processor with 16GB 2133 MHz LPDDR3 memory; (b) The symbol '–' indicates no data due to large computational time or memory limitations; and (c) We do not report rates for MOCBA since it alternates between allocations. We remark that sometimes the SCORE phantom rate is smaller than the iSCORE phantom rate. We believe this phenomenon is primarily due to the phantom rate approximation, although we also note here that all reported rates are to the tolerance of our solver.

B.1 Tables for Problems Generated via the Fixed Pareto Method

We begin by considering the test problems generated by the fixed Pareto method. Limiting the number of Pareto systems in this way limits the number of phantom Pareto systems, the number of MCI constraints, and the total number of MCE constraints in the MVN Phantom, SCORE, and iSCORE allocation strategies. Table 6 shows time and rate results for 3-objective problems generated by the fixed Pareto method. Recall that we use hard-coded look-up tables for the score and rate calculations with 3-objective problems. Tables 7 and 8 show similar time and rate results for 4-objective and 5-objective problems with 10 fixed Pareto systems, except the score and rate calculations in these tables required solving many quadratic programs.

The first row in Table 6 contains data on 10 problems, each with 5 Pareto systems and 5 non-Pareto systems. This row of small problems allows us to compare the MVN True and MVN Independent allocation strategies with MVN Phantom, SCORE, and iSCORE. Recall that the MVN True and MVN Independent allocation strategies involve calculating all of the brute force rates. Running these strategies on anything other than a small problem is computationally infeasible. We note that the median rates between MVN True, MVN Phantom, and SCORE show that the simplifications in MVN Phantom and SCORE do not cause a great loss in the $\mathbb{P}\{\text{MC}\}$ decay rate. We also see similar, yet lower, rates for MVN Independent and iSCORE, both of which treat objectives as independent.

Unlike MVN True and MVN Independent, the MVN Phantom allocation strategy can be computed for systems up to $r = 500$ in Table 6. Since MVN Phantom computes rate constraints for all non-Pareto systems with all phantom Pareto systems, there exists a point at which the number of constraints becomes too large for our solver to handle within memory limits. This happens for problems with size $r \geq 1,000$. The increased number of rate constraints in MVN Phantom also account for its slower computation times compared to SCORE and iSCORE.

Table 6 also shows that the median SCORE and iSCORE computation times are comparable to MOCBA for problem sizes $r \in \{10, 20, 50\}$ and faster than MOCBA for problem sizes greater than $r = 50$. The computations involved in MOCBA and iSCORE are similar in the sense that all systems

Table 6: The table reports metrics for 10 MORS problems with $d = 3$ objectives generated randomly using the fixed Pareto method.

	Med.	Med.		MVN	MVN	MVN				
r	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	True	Indep.	Phantom	SCORE	iSCORE	MOCBA	Equal
10	5	11	Median T	1m 41s	1m 45s	0.05s	0.03s	0.023s	0.005s	0s
			75th %-ile T	1m 55s	2m 2s	0.05s	0.05s	0.026s	0.006s	0s
			$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	950.316	702.155	950.297	924.602	695.678		393.455
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	948.024	702.155	950.297	924.602	695.678		393.455
20	10	21	Median T	–	–	0.17s	0.09s	0.05s	0.013s	0s
			75th %-ile T	–	–	0.19s	0.15s	0.06s	0.017s	0s
			$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	162.768	158.207	126.502		24.804
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	162.768	158.207	126.502		24.804
50	10	21	Median T	–	–	0.7s	0.09s	0.055s	0.044s	0s
			75th %-ile T	–	–	0.8s	0.11s	0.058s	0.045s	0s
			$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	45.618	44.755	44.305		3.944
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	45.618	44.755	44.305		3.944
100	10	21	Median T	–	–	2.1s	0.08s	0.06s	0.15s	0s
			75th %-ile T	–	–	2.5s	0.11s	0.06s	0.16s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	17.481	16.987	16.424		0.474
250	10	21	Median T	–	–	11s	0.12s	0.07s	0.86s	0s
			75th %-ile T	–	–	13s	0.16s	0.08s	0.88s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	5.944	5.713	4.666		0.126
500	10	21	Median T	–	–	1m 37s	0.15s	0.09s	3.41s	0s
			75th %-ile T	–	–	2m 38s	0.17s	0.10s	3.45s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.171	0.167	0.170		0.0009
1,000	10	21	Median T	–	–	–	0.17s	0.11s	13.6s	0s
			75th %-ile T	–	–	–	0.19s	0.12s	13.7s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.096	0.093		0.0004
2,000	10	21	Median T	–	–	–	0.22s	0.12s	54.3s	0s
			75th %-ile T	–	–	–	0.24s	0.18s	54.4s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.034	0.032		0.00008
5,000	10	21	Median T	–	–	–	0.47s	0.19s	6m 12s	0s
			75th %-ile T	–	–	–	0.49s	0.21s	6m 14s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.007	0.006		0.000005
10,000	10	21	Median T	–	–	–	0.64s	0.31s	25m 6s	0s
			75th %-ile T	–	–	–	0.70s	0.33s	25m 13s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.0004	0.0003		0.0000003

are considered when determining the allocations in MOCBA and the proportional allocations for the non-Pareto systems in iSCORE. MOCBA, however, also does additional calculations for each system in the problem to determine systems most likely to dominate another and on which objective that might happen. MOCBA also considers all systems in determining the $ae1$ and $ae2$ bounds. We suspect that the increased number of these computations are what causes MOCBA to take additional time on large problems.

Tables 7 and 8 show similar time and rate results for 4-objective and 5-objective problems

Table 7: The table reports metrics for 10 MORS problems with $d = 4$ objectives generated randomly using the fixed Pareto method.

	Med.	Med.		MVN				
r	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}^{\mathrm{ph}}\rvert$	Metric	Phantom	SCORE	iSCORE	MOCBA	Equal
20	10	44	Median T	18s	6s	0.07s	0.016s	0s
			75th %-ile T	24s	8s	0.08s	0.019s	0s
			$Z^{\mathrm{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	174.471	172.541	147.024		29.961
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	174.471	172.541	144.768		29.961
50	10	44	Median T	1m 15s	10s	0.10s	0.057s	0s
			75th %-ile T	1m 39s	12s	0.11s	0.060s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	4.656	4.515	3.156		0.214
100	10	44	Median T	1m 52s	12s	0.11s	0.198s	0s
			75th %-ile T	4m 9s	16s	0.11s	0.203s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	10.922	10.553	8.010		0.431
250	10	43	Median T	–	17s	0.14s	1.14s	0s
			75th %-ile T	–	21s	0.20s	1.15s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.976	0.882		0.011
500	10	43	Median T	–	24s	0.16s	4.5s	0s
			75th %-ile T	–	29s	0.17s	4.5s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.061	0.057		0.0003
1,000	10	44	Median T	–	26s	0.18s	17.5s	0s
			75th %-ile T	–	39s	0.20s	17.6s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.013	0.010		0.00005
2,000	10	43	Median T	–	31s	0.21s	1m 10.0s	0s
			75th %-ile T	–	53s	0.23s	1m 10.2s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.008	0.006		0.00001
5,000	10	42	Median T	–	1m 4s	0.30s	7m 43s	0s
			75th %-ile T	–	1m 28s	0.36s	7m 44s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.0013	0.0011		0.000002
10,000	10	44	Median T	–	1m 52s	0.57s	31m 5s	0s
			75th %-ile T	–	2m 20s	0.61s	31m 9s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.00009	0.00009		0.0000001

with 10 fixed Pareto systems. In these problems, we solve many quadratic programs, instead of using hard-coded look-up tables to determine the scores and rate constraints in the MVN Phantom and SCORE allocation strategies. We observe that iSCORE computation times are comparable to MOCBA for problem sizes $r \in \{10, 20, 50, 100\}$ and are faster than MOCBA for problem sizes greater than $r = 100$ in both 4-objective and 5-objective problems of this type. While SCORE suffers on computation time for smaller problems due to the quadratic programs calculations, it also overtakes MOCBA on 4-objective problems larger than $r = 1,000$ and 5-objective problems larger than $r = 2,000$. As mentioned above, we suspect that MOCBA takes longer for large problems due to the increased number of computations between systems. Comparing Tables 6 and 7, we see that an additional objective can add up to 1 minute of extra time in 5,000 system problems and around

Table 8: The table reports metrics for 10 MORS problems with $d = 5$ objectives generated randomly using the fixed Pareto method.

	Med.	Med.		MVN				
r	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}^{\mathrm{ph}}\rvert$	Metric	Phantom	SCORE	iSCORE	MOCBA	Equal
20	10	87	Median T	52s	12s	0.1s	0.019s	0s
			75th %-ile T	1m 3s	16s	0.2s	0.023s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	73.662	72.232	59.983		10.867
50	10	85	Median T	3m 45s	26s	0.22s	0.068s	0s
			75th %-ile T	6m 35s	28s	0.27s	0.071s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	6.129	4.422	5.783		0.504
100	10	88	Median T	17m 21s	35s	0.26s	0.247s	0s
			75th %-ile T	35m 24s	46s	0.32s	0.273s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	1.158	1.143	1.043		0.037
250	10	85	Median T	–	1m 1s	0.28s	1.41s	0s
			75th %-ile T	–	1m 16s	0.31s	1.42s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.095	0.092		0.001
500	10	84	Median T	–	57s	0.32s	5.53s	0s
			75th %-ile T	–	1m 8s	0.33s	5.55s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.034	0.027		0.0002
1,000	10	89	Median T	–	1m 28s	0.35s	21.91s	0s
			75th %-ile T	–	2m 2s	0.40s	21.94s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.021	0.021		0.0001
2,000	10	95	Median T	–	2m 55s	0.47s	1m 27.5s	0s
			75th %-ile T	–	3m 47s	0.52s	1m 27.8s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.008	0.006		0.00002
5,000	10	83	Median T	–	4m 15s	0.65s	9m 26s	0s
			75th %-ile T	–	4m 24s	0.68s	9m 32s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.001	0.0006		0.0000008
10,000	10	90	Median T	–	8m 52s	1.15s	38m 16s	0s
			75th %-ile T	–	11m 13s	1.21s	38m 31s	0s
			$Z^{\mathrm{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.0001	0.0001		0.0000002

6 minutes of extra time in 10,000 system problems for MOCBA.

The MVN Phantom allocation strategy runs into memory limitations for problems larger than $r = 100$ in both 4 and 5 objectives and has larger computation times for the small problems due to the quadratic program calculations. Consider a 4-objective problem with 250 systems, 10 Pareto systems, and 43 phantom Pareto systems. The MVN Phantom allocation strategy will include $10 * 9 = 90$ MCE rate constraints and $240 * 43 = 10,320$ MCI rate constraints to consider in `fmincon`. Additionally, each of the rates will need to be determined using a quadratic program.

B.2 Tables for Problems Generated via the Variable Pareto Method

We now consider the test problems generated by the variable Pareto method. In contrast to the fixed Pareto method, the variable Pareto method implies each test problem has a different number of Pareto systems, which results in a greater variation in the number of phantom Pareto systems for problems with the same overall size r. Thus, the number of MCE and MCI rate constraints for the MVN Phantom, SCORE, and iSCORE allocation strategies can be much higher than those in the fixed Pareto problems. Table 9 shows time and rate results for 3-objective problems with variable Pareto systems. Recall that we use hard-coded score and rate calculations with 3-objective problems. Tables 10 and 11 show time and rate results for 4-objective and 5-objective problems with variable Pareto systems, respectively. To solve for our proposed allocations in these tables, we

Table 9: The table reports metrics for 10 MORS problems with $d = 3$ objectives generated randomly using the variable Pareto method.

	Med.	Med.		MVN	MVN	MVN				
r	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	True	Indep	Phantom	SCORE	iSCORE	MOCBA	Equal
20	10	20	Median T	–	–	0.23s	0.09s	0.04s	0.017s	0s
			75th %-ile T	–	–	0.24s	0.13s	0.06s	0.021s	0s
			$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	17.997	17.951	15.081		3.497
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	17.997	17.951	15.081		3.497
50	11	23	Median T	–	–	1.0s	0.1s	0.07s	0.053s	0s
			75th %-ile T	–	–	1.3s	0.2s	0.09s	0.056s	0s
			$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	–	–		–
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	16.167	15.882	15.905		1.122
100	19	38	Median T	–	–	3s	0.4s	0.2s	0.166s	0s
			75th %-ile T	–	–	8s	1.2s	0.5s	0.172s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.941	0.809	0.857		0.033
250	31	62	Median T	–	–	80s	1s	1s	0.93s	0s
			75th %-ile T	–	–	2m 14s	3s	2s	0.94s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.946	0.932	0.714		0.011
500	43	87	Median T	–	–	–	4s	3s	3.67s	0s
			75th %-ile T	–	–	–	7s	5s	3.71s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.146	0.108		0.0011
1,000	62	125	Median T	–	–	–	7s	2s	14.3s	0s
			75th %-ile T	–	–	–	11s	10s	14.4s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.018	0.009		0.0001
2,000	93	187	Median T	–	–	–	7s	2s	57.4s	0s
			75th %-ile T	–	–	–	9s	4s	57.6s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.005	0.007		0.00002
5,000	165	330	Median T	–	–	–	12s	8s	5m 55s	0s
			75th %-ile T	–	–	–	13s	10s	5m 56s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.000008	0.000008		0.0000003
10,000	245	490	Median T	–	–	–	47s	35s	23m 56s	0s
			75th %-ile T	–	–	–	1m 6s	38s	23m 59s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–	0.00001	0.00001		0.0000002

solve many quadratic programs.

In Table 9, notice that SCORE and iSCORE computation times are comparable to MOCBA for problem sizes up to $r = 500$ and are faster than MOCBA for problem sizes greater than $r = 500$. The MVN Phantom allocation strategy runs into memory limitations for problems larger than $r = 250$ and is slower than all other allocation strategies due to the increased number of rate constraints considered. We include columns for MVN True and MVN Independent in this table to be consistent with Table 6, but computing the brute force allocations was computationally infeasible.

Table 10 shows time and rate results for 4-objective problems with variable Pareto systems. Since we solve many quadratic programs to determine our proposed allocations, the SCORE computation

Table 10: The table reports metrics for 10 MORS problems with $d = 4$ objectives generated randomly using the variable Pareto method.

r	Med. $\lvert\mathcal{P}\rvert$	Med. $\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	MVN Phantom	SCORE	iSCORE	MOCBA†	Equal
			Median T	26s	11s	0.07s	0.016s	0s
20	10	42	75th %-ile T	36s	13s	0.11s	0.022s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	91.091	89.554	89.554		11.612
			Median T	6m 3s	56s	0.24s	0.055s	0s
50	21	89	75th %-ile T	9m 42s	1m 11s	0.26s	0.056s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	16.659	16.252	12.460		1.269
			Median T	–	3m 47s	0.7s	0.21s	0s
100	31	143	75th %-ile T	–	16m 26s	1s	0.22s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.703	0.703		0.023
			Median T	–	1h 38m	12s	1.20s	0s
250	59	274	75th %-ile T	–	3h 54m	31s	1.20s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	0.145	0.120		0.003
			Median T	–	–	7s	4.67s	0s
500	87	432	75th %-ile T	–	–	37s	4.70s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.017		0.0003
			Median T	–	–	12s	18.60s	0s
1,000	133	686	75th %-ile T	–	–	53s	18.65s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.015		0.00006
			Median T	–	–	33s	1m 13.3s	0s
2,000	208	1,062	75th %-ile T	–	–	34s	1m 13.7s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.00006		0.000004
			Median T	–	–	2m 36s	8m 32s	0s
5,000*	374	2,064	75th %-ile T	–	–	2m 42s	8m 38s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.00005		0.000002
			Median T	–	–	–	32m 36s	0s
10,000*	564	3,133	75th %-ile T	–	–	–	32m 39s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–		0.0000002

* Computation for this row performed in MATLAB R2017a on a high performance computing cluster node with two 10-Core Intel Xeon-E5 processors and 128GB of memory

times are slower — in the range of hours for problems of size $r = 250$. Also, MVN Phantom begins to run into memory limitations due to the large number of rate constraints for problems of size $r = 100$. iSCORE is competitive with MOCBA for smaller problems and is faster than MOCBA for problems of size $r = 2,000$ and greater. Note, however, that iSCORE also succumbs to memory limitations for problems of size $r = 10,000$.

Table 11 shows time and rate results for 5-objective problems with variable Pareto systems. Since we solve many quadratic programs to determine our proposed allocations, the SCORE computation times are slower — in the range of hours for problems of size $r = 100$. Also, MVN Phantom begins to have memory limitations due to the large number of rate constraints for problems of size $r = 100$. iSCORE, while still faster than MVN Phantom and SCORE, is not as competitive with MOCBA for problems in this table. This result is largely due to the computation of the phantom Pareto

Table 11: The table reports metrics for 10 MORS problems with $d = 5$ objectives generated randomly using the variable Pareto method.

	Med.	Med.		MVN				
r	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	Phantom	SCORE	iSCORE	MOCBA†	Equal
			Median T	1m 16s	1m 16s	0.31s	0.02s	0s
20	16	123	75th %-ile T	1m 34s	1m 55s	0.58s	0.03s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	27.536	27.416	26.661		3.943
			Median T	18m 18s	4m 38s	2s	0.08s	0s
50	30	278	75th %-ile T	21m 44s	23m 6s	7s	0.08s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	7.394	7.326	5.461		0.641
			Median T	–	1h 40m	10s	0.24s	0s
100	48	536	75th %-ile T	–	2h 37m	19s	0.24s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	1.111	0.844		0.031
			Median T	–	–	21s	1.38s	0s
250	88	1,110	75th %-ile T	–	–	39s	1.44s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.039		0.0005
			Median T	–	–	1m 5s	5.35s	0s
500	146	1,997	75th %-ile T	–	–	3m 6s	5.38s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.014		0.0001
			Median T	–	–	3m 56s	21.18s	0s
1,000	228	3,373	75th %-ile T	–	–	4m 5s	21.23s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.0001		0.00004
			Median T	–	–	13m 31s	1m 45.2s	0s
2,000*	356	5,618	75th %-ile T	–	–	24m 25s	1m 45.8s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	0.0007		0.00002
			Median T	–	–	–	9m 43s	0s
5,000*	648	11,282	75th %-ile T	–	–	–	9m 44s	0s
			$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^4$	–	–	–		–

* Computation for this row performed in MATLAB R2017a on a high performance computing cluster node with two 10-Core Intel Xeon-E5 processors and 128GB of memory systems. Recall from §6 that the number of phantom Pareto systems, and the complexity of finding them, increases with the number of dimensions and Pareto systems. Thus, while the score and rate calculations with iSCORE are still quite fast, the bulk of the time spent to find an allocation is in finding the phantom Pareto systems. Note that iSCORE succumbs to memory limitations for problems of size $r = 5,000$.

C Test Problem 1: Objective Values and Supplemental Results

The objective vector values for Test Problem 1 appear in Table 12. Systems with indices 1, 2, 4, 5, and 9 are Pareto.

Table 12: Objective vector values for Test Problem 1 from Lee et al. [2010].

	System Index $s \in \mathcal{S}$																								
Obj.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
g_1	8	12	14	16	4	18	10	20	22	24	26	28	30	32	26	28	32	30	34	26	28	32	30	32	30
g_2	36	32	38	46	42	40	44	34	28	40	38	40	42	44	40	42	38	40	42	44	38	40	46	44	40
g_3	60	52	54	48	56	62	58	64	68	62	64	66	62	64	66	64	66	62	64	60	66	62	64	66	64

Figure 10 shows the performance of each sequential allocation strategy on Test Problem 1 in terms of the estimated probabilities of MC, MCE, and MCI events. Each row of the figure represents a different correlation that was applied to the systems within the problem. Notice that MOCBA and our proposed allocation strategies perform similarly with regards to estimated $\mathbb{P}\{\text{MCE}\}$, with MOCBA performing slightly better. Our proposed allocation strategies have a lower estimated $\mathbb{P}\{\text{MCI}\}$ and lower estimated overall $\mathbb{P}\{\text{MC}\}$.

D Test Problem 2: Supplemental Results

Figure 11 shows the performance of each sequential allocation strategy on Test Problem 2 in terms of the estimated probabilities of MC, MCE, and MCI events. On this test problem, iSCORE and MOCBA both do well, with iSCORE having a slightly better overall performance on the estimated $\mathbb{P}\{\text{MC}\}$ and estimated $\mathbb{P}\{\text{MCI}\}$. Both iSCORE and MOCBA perform significantly better than equal allocation.

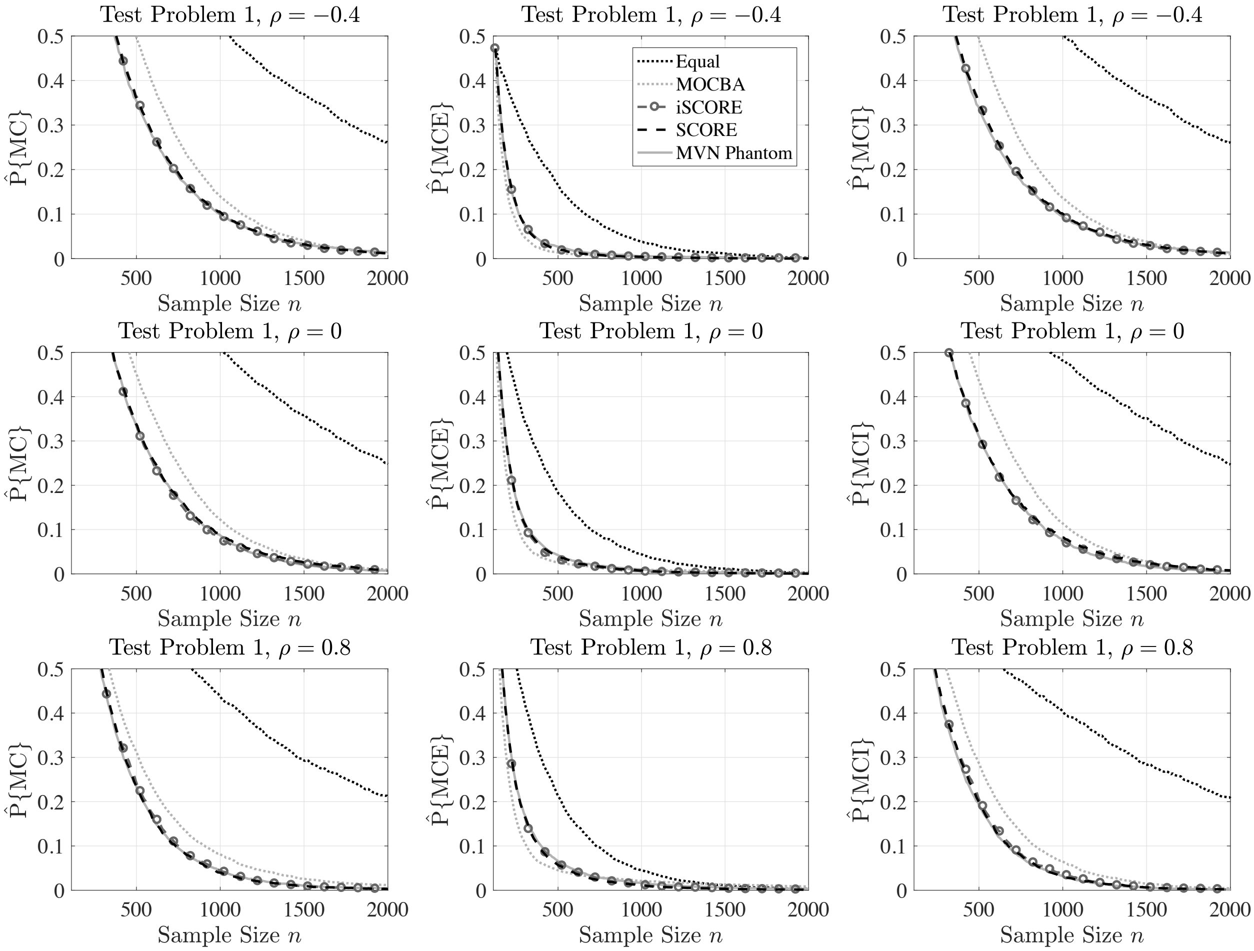

Figure 10: Test Problem 1, with $\rho = -0.4, \rho = 0$, and $\rho = 0.8$: Estimated $\mathbb{P}\{\mathrm{MC}\}$ for sequential allocation strategies, calculated across 10,000 independent sample paths for each algorithm.

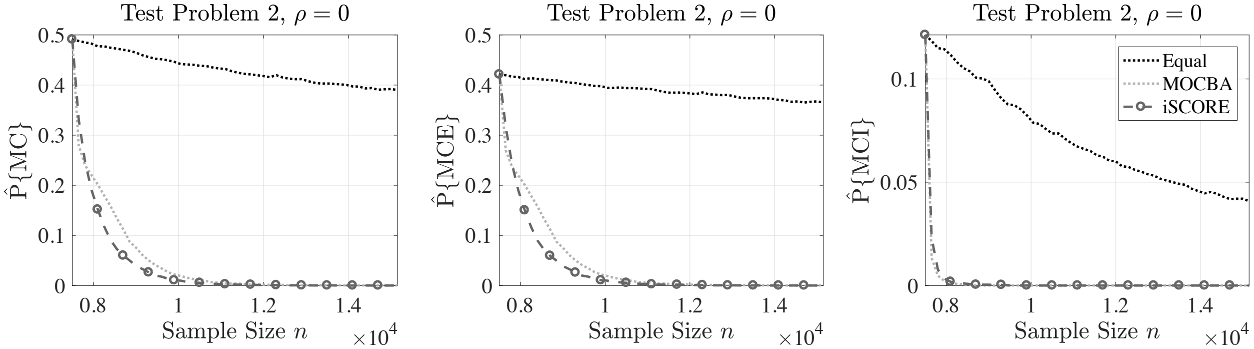

Figure 11: Test Problem 2: Average misclassification performance of sequential allocation strategies on Test Problem 2, calculated across 10,000 independent sample paths for each algorithm.