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

Eric A. Applegate[a], Guy Feldman[b], Susan R. Hunter[a], and Raghu Pasupathy[b]

[a]School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA;
[b]Department of Statistics, Purdue University, West Lafayette, IN 47907, USA

ARTICLE HISTORY
Compiled June 12, 2019

ABSTRACT
Consider the multi-objective ranking and selection (MORS) problem in which we select the Pareto-optimal set from a finite set of systems evaluated on three or more stochastic objectives. Solving this problem is difficult because we must determine how to allocate a simulation budget among the systems to minimize the probability that any systems are misclassified. Toward determining such a simulation budget allocation, we characterize the exact asymptotically optimal sample allocation that maximizes the misclassification-probability decay rate, and we provide an implementable allocation called MO-SCORE. The MO-SCORE allocation has three salient features: (a) it simultaneously controls the probabilities of misclassification by exclusion and inclusion; (b) it uses a fast dimension-sweep algorithm to identify phantom Pareto systems crucial for computational efficiency; and (c) it models dependence between the objectives. The MO-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 MO-SCORE (iMO-SCORE). Our numerical experience is extensive and promising: MO-SCORE and iMO-SCORE successfully solve MORS problems involving several thousand systems in three and four objectives.

KEYWORDS
multi-objective, ranking and selection, simulation optimization

1. Introduction

We consider the *multi-objective ranking and selection* (MORS) problem, in which a decision-maker wishes to select the set of "best" systems from a finite set of systems whose expected performances can only be observed with stochastic error. A *system* refers to one of the possible decision variable configurations under consideration. Each system's performance is assessed on the basis of multiple objective functions that are defined implicitly, for example, through a Monte Carlo simulation model that is capable of generating unbiased estimates of each objective. The solution to the MORS problem is the *Pareto set*, that is, the set of all non-dominated systems. We say that 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.

CONTACT S. R. Hunter. Email: susanhunter@purdue.edu

MORS problems often arise when designing stochastic systems (Butler, Morrice, & Mullarkey, 2001). Diverse examples of applications include plant breeding (Hunter & McClosky, 2016), earthmoving operations (Zhang, 2008), and supply chain management (Ding, Benyoucef, & Xie, 2006); Hunter et al. (2019) discuss additional application areas. In fact, a widely held viewpoint is that a substantial fraction of optimization problems in 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, MORS problems have received relatively little attention in the literature to date.

MORS procedures that do exist follow the same structure and conventions as single-objective R&S procedures, which have a long history of development (Fu & Henderson, 2017; Kim & Nelson, 2006). Single-objective R&S procedures broadly consist of constructing expected system performance estimators by obtaining one or more simulation replications from every system, perhaps sequentially, before using the constructed estimators to declare one system as the estimated best. In MORS procedures, the constructed estimators are used to declare one or more systems as members of the estimated Pareto set. Usually, both single-objective R&S and MORS procedures provide some form of guarantee on the quality of the estimated best system or on the simulation efficiency of the procedure. Hunter and Nelson (2017) classify procedures as *fixed-precision* or *fixed-budget*, depending on the guarantee provided. Fixed-precision procedures usually attempt to expend as few simulation replications as possible while providing a probabilistic solution quality guarantee upon termination, executing until a certain termination criterion is met and for which the probabilistic guarantee holds. Fixed-budget procedures, in contrast, usually attempt to maximize the probability of correctly selecting the best system or systems while expending only as many simulation replications as the fixed simulation budget allows.

While some work on fixed-precision MORS procedures exists (Batur, Wang, & Choobineh, 2018; J. S. Lee, 2014; Wang & Wan, 2017), most of the work on MORS to date consists of fixed-budget procedures, which is our focus. Many existing fixed-budget MORS procedures attempt to extend existing well-known single-objective R&S procedures into a setting with two or more objectives, *which is a complicated, highly-nontrivial task*. For example, arguably the most popular and well-known MORS procedure, MOCBA (L. H. Lee, Chew, Teng, & Goldsman, 2010), can be considered a multi-objective version of OCBA (Chen, Lin, Yücesan, & Chick, 2000); M-MOBA (Branke & Zhang, 2015; Branke, Zhang, & Tao, 2016) can be considered a bi-objective version of the Expected Value of Information (Chick, Branke, & Schmidt, 2010) and Knowledge Gradient (Frazier, Powell, & Dayanik, 2008) procedures, and bi-objective SCORE (Feldman & Hunter, 2018) extends the SCORE framework of Pasupathy, Hunter, Pujowidianto, Lee, and Chen (2015) into a bi-objective context. Some of these procedures are related (Ryzhov, 2016).

Table 1 classifies key fixed-budget MORS procedures by the number of objectives each procedure can handle and whether the procedures are able to model dependence between the objectives. We find MOCBA to be 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. In this paper, we answer some of these unresolved questions, and in doing so, we extend the bi-objective SCORE framework of Feldman and Hunter (2018) into three or more objectives, placing our work in the bottom-right cell of Table 1. For reasons we discuss in the sequel, the nature of the problem is much harder in a higher number of objectives, which makes this extension non-trivial.

Table 1. The table classifies key fixed-budget MORS procedures by their contributions.

Dependence	$d = 2$ Stochastic Objectives	$d \geq 2$ Stochastic Objectives
No	Hunter and McClosky (2016); M-MOBA† (Branke & Zhang, 2015; Branke et al., 2016)	MOCBA† (L. H. Lee et al., 2010) and variants by Teng, Lee, and Chew (2010); Choi and Kim (2018)†
Yes	Feldman and Hunter (2018)	This work and versions of MOCBA in Li, Liu, Pedrielli, Lee, and Chew (2018)

† Requires a normality assumption on the random objective vectors.

Finally, we remark here that some procedures identify only a subset of the Pareto set, often by requiring the decision-maker to specify a utility function in advance. Such procedures include the fixed-budget and fixed-precision procedures of Butler et al. (2001); Dudewicz and Taneja (1978, 1981); Frazier and Kazachkov (2011); Mattila and Virtanen (2015); Merrick, Morrice, and Butler (2015).

1.1. Unresolved Questions and Our Contributions

Consider a simple procedure designed to classify r systems as Pareto or non-Pareto on three or more objectives. Given a total simulation budget n, the procedure divides the the simulation replications among the systems, obtains the required number of simulation replications from each system, constructs estimated objective function values for each system on each objective, and declares each system as Pareto or non-Pareto based on the estimated objective function values. Since declaring each system as Pareto or non-Pareto is a relatively straightforward task, nearly all of the difficulty in designing this MORS procedure lies in determining how much of the simulation budget n should be allocated to each system so that the probability of a misclassification event is minimized. We say that 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 occurs if 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; L. H. 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 the total simulation budget n tends to infinity, 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 often is reflected unambiguously during implementation. This fact leads to the following fundamental question of MORS, which is currently unresolved in the literature: *What sampling allocation across the r systems maximizes the MC probability decay rate when the number of objectives is greater than two?*

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 (MO-SCORE). We

now discuss the salient aspects of MO-SCORE that allow it to resolve the theoretical question of optimal allocation while keeping the resulting algorithm computationally efficient; the structure of our paper mirrors the following list.

(1) *We provide the first exact characterization of the MC probability decay rate for MORS problems with three or more objectives,* which we call 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. Since computing the brute-force rate for problems with more than a few systems is difficult, we use the brute-force rate as the basis for our approximations and to assess their quality on small problems.

(2) *We characterize* phantom Pareto systems *for three or more objectives, which we use to approximate the brute-force rate.* Phantom Pareto systems are fictitious systems constructed by combining the objectives of strategically chosen Pareto systems (Hunter & McClosky, 2016). The phantom Pareto systems are formally defined for three or more objectives in §4.2 and illustrated by an example three-objective problem in Figure 1. The phantom Pareto systems enable us to approximate the brute-force rate in a certain asymptotic regime, thus facilitating an implementable allocation called the *phantom allocation.*

(3) *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.* Identifying the phantom Pareto systems for a given set of systems is itself a non-trivial problem. Since the phantom Pareto systems are integral to our approximations, having an efficient algorithm to locate them is crucial.

(4) *We provide the MO-SCORE allocation, which is a tractable approximation to the asymptotically optimal allocation.* Identifying the asymptotically optimal allocation involves solving a large bi-level optimization problem, which impedes implementation. A series of strategic approximations, including sending the number of non-Pareto systems to infinity, the phantom approximation to the brute-force rate, and strategic constraint reduction, resolves this issue.

(5) *Our MO-SCORE allocation 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 MO-SCORE allocation. Thus we also propose the independent MO-SCORE (iMO-SCORE) allocation for large problem instances in four or more objectives.

(6) *We perform extensive numerical experimentation that suggests MO-SCORE's stable and efficient performance on a variety of MORS problems.* For example, mainly due to the approximations, MO-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. Previous Work on Asymptotically Optimal Allocations

Our work contributes to the body of literature on asymptotically optimal allocations derived in a large deviations regime. Table 2 categorizes papers in this area by some of their differences, including 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 in which $r \to \infty$. The key papers from Table 2 that lead to our work are Glynn and Juneja (2004), Pasupathy et al. (2015), and Feldman and Hunter (2018), which we discuss below.

First, for single-objective, unconstrained R&S, Glynn and Juneja (2004) provide an 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 prove that, under certain regularity conditions, fixing the objective value of the best system and sending the total number of systems r to infinity results in an asymptotically optimal allocation to each suboptimal system that is $\Theta(1/r)$ and to the best system that is $\Theta(1/\sqrt{r})$ (see §1.3

Table 2. Some key papers on asymptotically optimal allocation using a large deviations analysis are classified by their contributions.

Paper	Stochastic Obj. / Con.	Dependence	Dist'n Rate Pf. / Implementation[a]	Limit as $r \to \infty$[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	G
Hunter and McClosky (2016)	$d = 2, c = 0$	No	G / N, Chi-Sq.	No
Feldman and Hunter (2018)[d]	$d = 2, c = 0$	Yes	G / N	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, 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, Hunter, Pasupathy, Lee, and Chen (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, Hunter, and Pasupathy (2015) and thesis Feldman (2017).

for notation). 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.

Building on this work, Feldman and Hunter (2018) derive the asymptotically optimal sampling allocation that maximizes the MC probability decay rate in the case of MORS with exactly two objectives. Then, under certain regularity conditions and a normality assumption, they prove that fixing the Pareto set and sending the total number of systems r to infinity results in an asymptotically optimal allocation to each non-Pareto system that is $\Theta(1/r)$ and to each Pareto system that is $\Theta(1/\sqrt{r})$. Feldman and Hunter (2018) conjecture that these results hold for general distributions, and provide SCORE allocations for bi-objective R&S that account for correlation between the objectives. Due to several simplifications employed by MOCBA, the bi-objective SCORE allocations do not correspond to MOCBA.

Feldman and Hunter (2018) provide the theoretical background and proof-of-concept for our work in more than two objectives. However, the proof techniques in Feldman and Hunter (2018) rely on the fact that the exact MC probability decay rate can be characterized using phantom Pareto systems, and the probability of some MCI events can be bounded below by the probability of MCE events. These arguments no longer hold in more than two objectives. (We briefly discuss why in §5.) Therefore in this paper, after we derive the asymptotically optimal allocation using the new brute-force formulation, we focus on finding good approximations that work well computationally, instead of the difficult details of proving the limiting SCORE regime. We remark here that while we are aware of results in Glynn and Juneja (2011, 2015) regarding estimating rate functions, our numerical experience has been overwhelmingly positive when estimating only the parameters of an assumed normal family.

Finally, the recent work of Li et al. (2018) provides a number of updates to the MOCBA allocation of L. H. Lee et al. (2010) based on a large deviations analysis. The authors first derive upper and lower bounds on the MC probability, followed by upper and lower bounds on the large deviations decay rate for these bounding probabilities. The authors then present an optimization problem to determine the budget allocation that maximizes the lower bound. The authors propose three new allocation schemes based on this analysis and simplifications of it: MOCBA*, MOCBA#, and MOCBA+, which are different from our phantom, MO-SCORE, and iMO-SCORE allocations. We compare our allocations with MOCBA*, MOCBA#, and MOCBA+ in the numerical section.

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 $\mathbf{x} \neq \mathbf{y}$. (This notation is standard in, e.g., Ehrgott (2005); Wiecek, Ehrgott, and Engau (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. For a sequence of real numbers $\{a_n\}$, we say that $a_n = \Theta(1)$ if $0 < \liminf a_n \leq \limsup a_n < \infty$.

2. Problem Setting and Formulation

In this section, we provide a formal problem statement and discuss our assumptions.

2.1. Problem Statement

We write the MORS problem as

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

where $\mathbf{g}(s) \in \mathbb{R}^d$ is a vector representing the expected performance of system s on each of the d objectives, $\mathcal{S} := \{1,\dots,r\}$ is a finite set of system indices, and $\mathcal{D} := \{1,\dots,d\}$ 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 $g_k(s) \leq g_k(s')$ 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, $\mathcal{P} := \{i \in \mathcal{S} : \nexists s \in \mathcal{S} \text{ such that } \mathbf{g}(s) \leq \mathbf{g}(i)\}$.

For all systems $s \in \mathcal{S}$, let $\boldsymbol{G}_m(s) := (G_{1m}(s),\dots,G_{dm}(s))$ 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 $\bar{\boldsymbol{G}}(s,n) = (\overline{G}_1(s,n),\dots,\overline{G}_d(s,n)) := n^{-1}\sum_{m=1}^{n}\boldsymbol{G}_m(s)$. Let α_s be the proportion of the simulation budget n allocated to system $s \in \mathcal{S}$, and define $\widehat{\boldsymbol{G}}(s) := \bar{\boldsymbol{G}}(s,n\alpha_s)$ 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 an MC event occurs. We seek a simulation budget allocation $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_r)$, $\sum_{s=1}^{r}\alpha_s = 1$ that maximizes the MC event probability decay rate as the simulation budget n increases, thus providing an efficiency guarantee for solving Problem M.

2.2. Assumptions

First, we require that each Pareto system is distinguishable from every other system on each objective, which is standard 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, we assume that for each system $s \in \mathcal{S}$, the performance vectors $\boldsymbol{G}_m(s), m = 1,2,\dots$ are independent and identically distributed (i.i.d.) multivariate normal random variables. We further assume that all systems are simulated independently of each other. *We remark here that all results in §3 hold more generally under the standard assumptions required for the Gärtner-Ellis Theorem (Dembo & Zeitouni, 1998, p. 43). For compactness, we do not include the assumptions here. Instead, we refer the interested reader to Feldman (2017, p. 11–13).*

Assumption 2. *For each system* $s \in \mathcal{S}$, $\boldsymbol{G}_m(s), m = 1,2,\dots$ *are i.i.d.* $N(\mathbf{g}(s),\Sigma(s))$ *random vectors, where* $\Sigma(s)$ *is a positive definite covariance matrix with diagonal entries* $\sigma_1^2(s),\dots,\sigma_d^2(s)$ *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* $\{\boldsymbol{G}_m(s)\colon s \in \mathcal{S}, m = 1,2,\dots\}$ *are mutually independent.*

Assumption 2 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 normality assumption is widely used with success in R&S; 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 $\bar{\boldsymbol{G}}(s,n)$ and $\bar{G}_k(s,n)$ 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 $\bar{\boldsymbol{G}}(s,n)$ be $I_s(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^d$, and let the large deviations rate function corresponding to the random variable $\bar{G}_k(s,n)$ 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 $s \in \mathcal{S}, k \in \mathcal{D}$.

3. 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, $\boldsymbol{\alpha}$.

3.1. The Brute-Force Rate: The Misclassification Probability Decay Rate

We begin by writing the MC event, $\mathrm{MC} := (\hat{\mathcal{P}} \neq \mathcal{P})$, in a way that facilitates analysis. Recall that there are two ways an MC event can occur: MCE, in which a truly Pareto system is falsely excluded from $\hat{\mathcal{P}}$, and MCI, in which a truly non-Pareto system is falsely included in $\hat{\mathcal{P}}$. Thus the MC event can be written as $\mathrm{MC} = \mathrm{MCE} \cup \mathrm{MCI}$. Feldman (2017, p. 108–112) shows that $\mathrm{MC} = \mathrm{MCE}_{\mathcal{P}} \cup \mathrm{MCI}$, where $\mathrm{MCE}_{\mathcal{P}}$ denotes the event that a truly Pareto system is estimated as dominated by another Pareto system, $\mathrm{MCE}_{\mathcal{P}} := \cup_{i \in \mathcal{P}} \cup_{i' \in \mathcal{P}} \widehat{\boldsymbol{G}}(i') \leq \widehat{\boldsymbol{G}}(i)$, and $\mathrm{MCI} := \cup_{j \in \mathcal{P}^c} \cap_{i \in \mathcal{P}} \cup_{k \in \mathcal{D}} \widehat{G}_k(j) \leq \widehat{G}_k(i)$. Since $\mathrm{MCE} = \mathrm{MCE}_{\mathcal{P}}$, henceforth, we denote $\mathrm{MCE}_{\mathcal{P}}$ as MCE. Then assuming the limits exist, the $\mathbb{P}\{\mathrm{MC}\}$ decay rate is

$$-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\mathrm{MC}\} = \min\left(-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\mathrm{MCE}\}, -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\mathrm{MCI}\} \right). \tag{2}$$

Feldman (2017, p. 113–114) shows that the $\mathbb{P}\{\mathrm{MCE}\}$ decay rate equals the minimum among the pairwise decay rates of the probability that one Pareto system dominates another (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

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

Then the $\mathbb{P}\{\mathrm{MCE}\}$ decay rate is

$$-\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\{\mathrm{MCE}\} = \min_{i \in \mathcal{P}} \min_{i' \in \mathcal{P},\, i' \neq i} R_{i'i}^{\mathrm{MCE}}(\alpha_{i'}, \alpha_i). \tag{3}$$

In the rest of this section, we obtain an expression for the $\mathbb{P}\{\mathrm{MCI}\}$ decay rate, which

is more complicated. 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, 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 discuss in §5. We take a different approach to analyzing the $\mathbb{P}\{\text{MCI}\}$ 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 analysis of the $\mathbb{P}\{\text{MCI}\}$ decay rate in three or more objectives.

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 the set of Pareto system indices is $\mathcal{P} = \{1,\ldots,p\}$. Recall that for any non-Pareto system to be falsely included in the Pareto set, it must beat each Pareto system on at least one 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 $\kappa_p \in \{1,2,\ldots,d\}$. Letting this vector of objectives be denoted $\boldsymbol{\kappa} := (\kappa_1,\kappa_2,\ldots,\kappa_p)$, the set $\mathcal{K} = \{\boldsymbol{\kappa}\colon \boldsymbol{\kappa} \in \{1,2,\ldots,d\}^p\}$ represents all possible ways that a non-Pareto system can be falsely included in the Pareto set. Now define the *brute-force* MCI *event* as

$$\text{MCI}^{\text{bf}} := \cup_{j\in\mathcal{P}^c} \cup_{\boldsymbol{\kappa}\in\mathcal{K}} \cap_{i\in\mathcal{P}} \widehat{G}_{\kappa_i}(j) \le \widehat{G}_{\kappa_i}(i),$$

where 'bf' denotes 'brute-force.' By the definition of $\mathcal{K}$, the following proposition holds.

Proposition 3.1 (Feldman, 2017, p. 116)**.** $\text{MCI} = \text{MCI}^{\text{bf}}$.

Since the MCI^{bf} event reformulates MCI as a union over all non-Pareto systems and all objective index vectors, the $\mathbb{P}\{\text{MCI}\}$ 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

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

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, p. 118–120); we provide only a proof sketch in the online appendix.

Lemma 3.2 (Feldman, 2017, p. 118)**.** *The* $\mathbb{P}\{\text{MCI}\}$ *decay rate is*

$$-\lim \frac{1}{n} \log \mathbb{P}\{\text{MCI}\} = \min_{j\in\mathcal{P}^c} \min_{\boldsymbol{\kappa}\in\mathcal{K}} R^{\text{MCI}}_{j\boldsymbol{\kappa}}(\alpha_j,\boldsymbol{\alpha}_{\mathcal{P}}). \tag{4}$$

Finally, this section's main theorem results from combining the decay rates for $\mathbb{P}\{\text{MCE}\}$ and $\mathbb{P}\{\text{MCI}\}$ in equations (3) and (4), respectively, with equation (2).

Theorem 3.3. *The* $\mathbb{P}\{\text{MC}\}$ *decay rate, which we call the* brute-force rate*, is*

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

Theorem 3.3 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.

3.2. The Optimal Allocation Strategy

Maximizing the $\mathbb{P}\{\text{MC}\}$ decay rate in Theorem 3.3 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., Feldman & Hunter, 2018; Glynn & Juneja, 2004; Hunter & Pasupathy, 2013; Pasupathy et al., 2015). 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. The total number of rates to compute when solving Problem Q quickly becomes prohibitively large: for $d = 3$ dimensions and $r = 30$ total systems, $p = 10$ of which are Pareto systems, we must compute over one million rates.

4. Approximating the Optimal Allocation in the Limiting SCORE Regime

Considering the computational complexity of solving Problem Q, we provide heuristics that approximate the solution to Problem Q and require far less computational resources. Inspired by the SCORE family of allocations, 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 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 optimal allocations by showing that the brute-force rate can be approximated by a formulation involving phantom Pareto systems.

4.1. Approximating the Allocation to the Non-Pareto Systems

In this section, we simplify finding the optimal allocation as the solution to Problem Q by pre-determining 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, $R^{\text{MCI}}_{j\boldsymbol{\kappa}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$ is the solution to the quadratic program

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

where $\mathbf{x}_j$ and $\mathbf{x}_{\boldsymbol{\kappa}} := (x_{1\kappa_1}, \ldots, x_{p\kappa_p})$ are the decision variables, $\mathbf{g}_{\mathcal{P}}(\boldsymbol{\kappa}) := (g_{\kappa_1}(1), \ldots, g_{\kappa_p}(p))$ is a vector of Pareto system objective values specified by $\boldsymbol{\kappa}$, $\Sigma_{\mathcal{P}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}})$ 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 $\boldsymbol{\kappa}$.

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 (Limiting SCORE Regime)**.** *The systems are arranged such that all systems receive positive allocation in Problem $\widetilde{Q}$ (see, e.g., Feldman & Hunter, 2018, p. 12, Assumption 8). Further, 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 & 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 $\lim_{|\mathcal{P}^c|\to\infty} R^{\text{MCI}}_{j\boldsymbol{\kappa}}(\tilde{\alpha}^*_j, \tilde{\boldsymbol{\alpha}}^*_{\mathcal{P}})/\tilde{\alpha}^*_j = \inf_{x_{j\kappa_i} \leq g_{\kappa_i}(i)\ \forall i \in \mathcal{P}} I_j(\mathbf{x}_j)$ 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. Our allocation policy should work well for a variety of problems, which we explore in §7 and §9.

Using the regime in Assumption 3, for each non-Pareto system $j \in \mathcal{P}^c$ and objective

vector $\boldsymbol{\kappa} \in \mathcal{K}$, let $\mathbb{S}_j(\boldsymbol{\kappa}) := \inf_{x_{j\kappa_i} \le g_{\kappa_i}(i)\ \forall i \in \mathcal{P}} I_j(\mathbf{x}_j)$. Then, set the *score* equal to $\mathbb{S}_j := \min_{\boldsymbol{\kappa} \in \mathcal{K}} \mathbb{S}_j(\boldsymbol{\kappa})$ for all $j \in \mathcal{P}^c$. Under Assumption 3, the optimal value of Problem $\widetilde{Q}$ equalizes the rates so that $\tilde{z}^* = \min_{\boldsymbol{\kappa} \in \mathcal{K}} R^{\mathrm{MCI}}_{j\boldsymbol{\kappa}}(\tilde{\alpha}^*_j, \tilde{\boldsymbol{\alpha}}^*_{\mathcal{P}})$ for each $j \in \mathcal{P}^c$, which implies that in the limit, $\tilde{z}^*/\tilde{\alpha}^*_j = \mathbb{S}_j$ for each $j \in \mathcal{P}^c$. This result leads to Theorem 4.1 below, which provides the limiting relative optimal allocations between the non-Pareto systems. We simplify the score expressions in the next section.

Theorem 4.1. *Under Assumption 3, for all non-Pareto systems $j, j' \in \mathcal{P}^c$,*

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

4.2. The Phantom Pareto System Simplification of the Scores

The allocations in Theorem 4.1 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 are 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. We now write these constraints with vectors, as follows. Define the kth element of the d-dimensional vector $\mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}) = (g^{\mathrm{bf}}_1(\boldsymbol{\kappa}), \ldots, g^{\mathrm{bf}}_d(\boldsymbol{\kappa}))$ as

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

This notation essentially goes objective-by-objective and specifies the minimum values that the decision vector $\mathbf{x}_j$ must not exceed in the score calculation. Then by construction, the following Lemma 4.2 holds.

Lemma 4.2. $\mathbb{S}_j(\boldsymbol{\kappa}) = \inf_{\mathbf{x}_j \le \mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa})} I_j(\mathbf{x}_j)$ *for each* $j \in \mathcal{P}^c, \boldsymbol{\kappa} \in \mathcal{K}$.

Given the results in Lemma 4.2, notice that we do not need all $\boldsymbol{\kappa}$ vectors to determine the score: some vectors are redundant. Lemma 4.3 characterizes the redundant $\boldsymbol{\kappa}$ vectors. A proof for Lemma 4.3 appears in the Online Appendix.

Lemma 4.3. *If $\boldsymbol{\kappa}, \boldsymbol{\kappa}' \in \mathcal{K}$ are such that $\mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}') \le \mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa})$, then $\mathbb{S}_j(\boldsymbol{\kappa}) \le \mathbb{S}_j(\boldsymbol{\kappa}')$.*

Now we define a minimal set of d-dimensional brute-force points in the objective space as $\mathcal{G}^{\mathrm{ph}} := \{\mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}) : \boldsymbol{\kappa} \in \mathcal{K},\ \nexists\ \boldsymbol{\kappa}' \in \mathcal{K} \text{ such that } \mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}) \le \mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}')\}$. That is, the set $\mathcal{G}^{\mathrm{ph}}$ keeps only the points $\mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}), \boldsymbol{\kappa} \in \mathcal{K}$ that *do not dominate* any other points $\mathbf{g}^{\mathrm{bf}}_d(\boldsymbol{\kappa}'), \boldsymbol{\kappa}' \in \mathcal{K}$. The set $\mathcal{G}^{\mathrm{ph}}$ defines the *phantom Pareto systems* in d dimensions. Let $\mathcal{P}^{\mathrm{ph}}$ denote the set of phantom Pareto system indices and let ℓ index the phantom Pareto systems, so that $\ell \in \mathcal{P}^{\mathrm{ph}}$. Then let the points in the set of objective values of

the phantom Pareto systems $\mathcal{G}^{\text{ph}}$ be denoted

$$\mathbf{g}^{\text{ph}}(\ell) = (g_1^{\text{ph}}(\ell), \ldots, g_d^{\text{ph}}(\ell)),$$

where $g_k^{\text{ph}}(\ell) := g_k(i_k(\ell))$ if Pareto system $i_k(\ell), k \in \mathcal{D}$ contributes its kth objective value to the phantom Pareto system ℓ; otherwise, if no such Pareto system exists, then for that objective k, $i_k(\ell) := 0$ and $g_k^{\text{ph}}(\ell) = g_k(0) := \infty$. An illustration containing phantom Pareto systems in three objectives appears in Figure 1. Using the phantom Pareto systems, the simplified score can be calculated without brute-force enumeration, as specified in the following Theorem 4.4. Theorem 4.4 follows from Theorem 4.1 together with Lemmas 4.2 and 4.3.

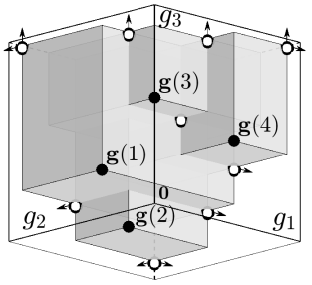

Figure 1. Black and white circles denote Pareto and phantom Pareto systems, respectively, in $d = 3$. Arrows show dimensions in which phantom values equal infinity. (Also see Lacour et al., 2017.)

Theorem 4.4. *Under Assumption 3, for all non-Pareto systems $j, j' \in \mathcal{P}^c$,*

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

where $\mathbb{S}_j(\ell) := \inf_{\mathbf{x}_j \leq \mathbf{g}^{\text{ph}}(\ell)} I_j(\mathbf{x}_j)$ for all $j \in \mathcal{P}^c, \ell \in \mathcal{P}^{\text{ph}}$.

We have simplified calculations only for the relative allocations between the non-Pareto systems; the allocations to the Pareto systems are determined heuristically in §6.

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 & Suri, 2012), and more specifically, to the problem of calculating the hypervolume indicator 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, Rubin, Sharir, and Verbin (2008). Importantly, Kaplan et al. (2008) prove that the number of phantom Pareto systems associated with a set of p Pareto systems in d objectives 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.

5. The Phantom Allocation: An Approximation to the Optimal Allocation

The phantom Pareto systems that result from removing redundant $\boldsymbol{\kappa}$ vectors in the limiting SCORE regime are also handy for approximating the brute-force rate and the optimal allocation. In this section, we formulate such an approximation.

5.1. The Phantom Rate: An Approximation to the Brute-Force Rate

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, we require additional notation. First, for each phantom Pareto system ℓ, let $\mathcal{P}(\ell)$ be the set containing the indices of the Pareto systems that contribute an objective value to defining phantom Pareto system ℓ. More formally, $\mathcal{P}(\ell) = \{i \in \mathcal{P} : \exists k \ni g_k(i) \in$

$\{g_1^{\text{ph}}(\ell), \ldots, g_d^{\text{ph}}(\ell)\}\}$ denotes the set of indices of all the Pareto systems $i_k(\ell) \in \mathcal{P}$ that contribute objective function value k to phantom Pareto system $\ell \in \mathcal{P}^{\text{ph}}$. Then, let $\boldsymbol{\alpha}_{\mathcal{P}(\ell)}$ be the vector of proportional allocations for the Pareto systems in $\mathcal{P}(\ell)$. Further, define the d-dimensional vector of variables $\mathbf{x}_\ell^{\text{ph}} = (x_{\ell 1}^{\text{ph}}, \ldots, x_{\ell d}^{\text{ph}})$. Now for each $j \in \mathcal{P}^c$ and $\ell \in \mathcal{P}^{\text{ph}}$, define the approximate rate of decay of $\mathbb{P}\{\text{MCI}\}$ as

$$R_{j\ell}^{\text{ph}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}(\ell)}) := \inf_{\mathbf{x}_j \le \mathbf{x}_\ell^{\text{ph}}} \big(\alpha_j I_j(\mathbf{x}_j) + \textstyle\sum_{k \in \mathcal{D}} \alpha_{i_k(\ell)} J_{i_k(\ell)k}(x_{\ell k}^{\text{ph}})\big), \tag{5}$$

where $\alpha_0 := 0$, and $J_{0k}(x) := 0$ for all $x \in \mathbb{R}, k \in \mathcal{D}$. Then we approximate the $\mathbb{P}\{\text{MC}\}$ decay rate in Theorem 3.3 as the *phantom rate*

$$\begin{aligned} z^{\text{ph}}(\boldsymbol{\alpha}) &:= \min\big(\min_{i \in \mathcal{P}} \min_{i' \in \mathcal{P}, i \neq i'} R_{i'i}^{\text{MCE}}(\alpha_{i'}, \alpha_i), \min_{j \in \mathcal{P}^c} \min_{\ell \in \mathcal{P}^{\text{ph}}} R_{j\ell}^{\text{ph}}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}(\ell)})\big) \\ &\approx -\lim_{n \to \infty} \tfrac{1}{n} \log \mathbb{P}\{\text{MC}\}. \end{aligned} \tag{6}$$

When there are two objectives, Feldman and Hunter (2018) prove that the rate in equation (6) is equal to the $\mathbb{P}\{\text{MC}\}$ decay rate. When there are three or more objectives, the rate in equation (6) is not necessarily equal to the $\mathbb{P}\{\text{MC}\}$ decay rate. In brief, this discrepancy occurs because in three or more objectives, the phantom rate does not account for the ordering of the Pareto systems in the absence of an MCE event. We include an example illustrating this effect in the online supplement. Notice, however, that the approximation in equation (6) 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 events in which the Pareto systems are estimated "out of order" are highly unlikely.

5.2. The Phantom Allocation Strategy

Using the approximate rate of decay of $\mathbb{P}\{\text{MC}\}$ in equation (6), we can formulate a new version of Problem Q as Problem Q^{ph}, having solution $\boldsymbol{\alpha}^{\text{ph}}$:

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

Since calculating the optimal allocation via Problem Q is difficult for anything but the smallest of problems, in §7 and §9, we often use the phantom approximations.

6. The MO-SCORE Allocation Framework

Having characterized the asymptotically optimal allocation and its approximation using the phantom Pareto systems, we now present the MO-SCORE allocation framework. In this framework, we calculate the relative allocations to the non-Pareto systems using the scores in Theorem 4.4. We also calculate the rate of decay of $\mathbb{P}\{\text{MCI}\}$ using the phantom approximation in equation (5). Then, we approximate Problem Q^{ph} by

strategically dropping constraints to reduce the computational complexity. Finally, we formulate the iMO-SCORE allocation for large problems with four or more objectives.

6.1. The MO-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 4.4. For all non-Pareto systems $j \in \mathcal{P}^c$, define $\lambda_j^{\mathbb{S}} := \mathbb{S}_j^{-1} / \sum_{j' \in \mathcal{P}^c} \mathbb{S}_{j'}^{-1}$, 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}, 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, notice that there are at most d Pareto systems that contribute objective values to each 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

$$j_i^*(\ell) = \text{argmin}_{j \in \mathcal{P}^c} \left\{ \mathbb{S}_j(\ell) : \mathbb{S}_j(\ell) \neq \inf_{x_{jk} \leq g_k^{\text{ph}}(\ell)\ \forall k \neq k^*(i)} \left(\inf_{x_{jk^*(i)}} I_j(\mathbf{x}_j) \right) \right\}$$

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 that control 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 $k \in \{1, \ldots, d\}$,

$$i'^*_k(i) := \text{argmin}_{i' \in \mathcal{P}} \left\{ \mathbb{T}_{i'}(i) \colon \mathbb{T}_{i'}(i) \neq \inf_{x_{i'k'} \leq g_{k'}(i)\ \forall k' \neq k} \left(\inf_{x_{i'k}} I_{i'}(\mathbf{x}_{i'}) \right) \right\}$$

is 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 MO-SCORE allocation framework results from setting $\alpha_j = \lambda_j^{\mathbb{S}}(1 - \sum_{i=1}^p \alpha_i)$ for all $j \in \mathcal{P}^c$ and solving

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

When there are three objectives or few systems, our experience with the MO-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.

6.2. The iMO-SCORE 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 we outline a further simplification of the MO-SCORE framework that we call the independent MO-SCORE (iMO-SCORE) 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

$$\begin{aligned} L^{\text{MCE}}_{i'i}(\alpha_{i'}, \alpha_i) &:= \sum_{k \in \mathcal{D}} \left(\tfrac{(g_k(i) - g_k(i'))^2 \mathbb{I}\{g_k(i') > g_k(i)\}}{2(\sigma_k^2(i)/\alpha_i + \sigma_k^2(i')/\alpha_{i'})} \right) \text{ for all } i, i' \in \mathcal{P}, i \neq i', \\ L^{\text{MCI}}_{j\ell}(\alpha_j, \boldsymbol{\alpha}_{\mathcal{P}(\ell)}) &:= \sum_{k \in \mathcal{D}} \left(\tfrac{(g_k(j) - g_k^{\text{ph}}(\ell))^2 \mathbb{I}\{g_k(j) > g_k^{\text{ph}}(\ell)\}}{2(\sigma_k^2(j)/\alpha_j + \sigma_k^2(i_k(\ell))/\alpha_{i_k(\ell)})} \right) \text{ for all } j \in \mathcal{P}^c, \ell \in \mathcal{P}^{\text{ph}}; \end{aligned}$$

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

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

and $\mathbb{S}_j^{\text{ind}} := \min_{\ell \in \mathcal{P}^{\text{ph}}} \mathbb{S}_j^{\text{ind}}(\ell)$. We form reduced constraint sets that are identical to $\mathcal{J}^*(\ell)$ and $\mathcal{M}^*(i)$, except that we use $\mathbb{S}_j^{\text{ind}}(\ell)$ and $\mathbb{T}_i^{\text{ind}}(i')$; thus we call the iMO-SCORE reduced constraint sets $\mathcal{J}^{\text{ind}}(\ell)$ and $\mathcal{M}^{\text{ind}}(i)$.

Our proposed iMO-SCORE allocation framework results from setting $\alpha_j = \lambda_j^{\text{ind}}(1 - \sum_{i=1}^p \alpha_i)$ for all $j \in \mathcal{P}^c$, where $\lambda_j^{\text{ind}} = (\mathbb{S}_j^{\text{ind}})^{-1} / \sum_{j' \in \mathcal{P}^c} (\mathbb{S}_{j'}^{\text{ind}})^{-1}$, and solving

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

7. 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 3-objective, 3-system test problem from Li et al. (2018) and a suite of randomized test problems. When possible, we also investigate how close each allocation is to the asymptotically optimal allocation. The results we present give us a sense of how long one update of the optimal allocation takes in the sequential implementation in §8. We compare the following allocation strategies under the normality Assumption 2:

- MVN True, in which we solve Problem Q using brute-force rates;
- MVN Phantom, in which we solve Problem Q^{ph};
- MO-SCORE, in which we solve Problem $Q_{\mathbb{S}}$;
- MVN Ind., in which we solve Problem Q assuming independent objectives;
- iMO-SCORE, in which we solve Problem $Q_{\mathbb{S}}^{\text{ind}}$;
- MOCBA, as described in L. H. Lee et al. (2010);
- LD-based MOCBA, which is described as MOCBA in Li et al. (2018);
- MOCBA*, MOCBA#, and MOCBA+ from Li et al. (2018);
- equal allocation.

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 MO-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 MO-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. Solving for the MVN Phantom, MO-SCORE, and iMO-SCORE allocations requires locating the phantom Pareto systems; we use the algorithm described in the online supplement. To implement LD-based MOCBA, MOCBA*, MOCBA#, and MOCBA+, we rely on MATLAB code supplied by the authors of Li et al. (2018).

We explore the performance of these algorithms on a variety of test problems. First, we implement the 3-objective, 3-system problem from Li et al. (2018), which has true objective function values $(3.0, 4.0, 2.2)$, $(3.5, 5.0, 3.0)$, and $(4.0, 3.5, 2.0)$, all variances equal to 1, and all correlations equal to zero. Then, we also generate a randomized test

problem suite using two different methods. 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 an increasing number of systems r, we generate 10 MORS problems using the fixed and variable Pareto methods. Then, we calculate statistics for each set of problems. Tables 3 and 4 report: the median number of Pareto systems p; the median number of phantom Pareto systems $|\mathcal{P}^{\text{ph}}|$; 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 $\mathbb{P}\{\text{MC}\}$, when possible; and the median phantom rate of decay of the $\mathbb{P}\{\text{MC}\}$ across each set of random problems.

We observe the following about our proposed allocations from Tables 3 and 4. First, the relatively small difference in the median rates of decay of $\mathbb{P}\{\text{MC}\}$ for the MVN True, MVN Phantom, and MO-SCORE 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 3 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 MO-SCORE, iMO-SCORE, and equal allocation in rows where MVN True and MVN Phantom allocations cannot be calculated.

Next, we compare MO-SCORE and iMO-SCORE with MOCBA. Notice that in nearly all 3-objective rows and nearly all rows with problems generated via the fixed Pareto method, the median times for MO-SCORE and iMO-SCORE are clearly faster than those of MOCBA. However, for small problem instances when computations are fast, MO-SCORE, iMO-SCORE, and MOCBA are computationally comparable. Interestingly, in terms of computational time, MOCBA suffers a penalty for a large total number of systems, while iMO-SCORE suffers a penalty only for a large Pareto set. This penalty is especially noticeable in the 5-objective, 2,000-system row of Table 4, in which iMO-SCORE 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.

Finally, we compare MO-SCORE and iMO-SCORE with LD-based MOCBA, MOCBA*, MOCBA#, and MOCBA+. The latter allocations achieve a better rate than the former allocations only in Table 3, row $d = 3$, $r = 10$, $p = 5$. However, for problems of this size, using the MVN Phantom allocation is fast and yields a very good rate, close to the brute force rate of MVN True. Therefore for small problems with few objectives, the MVN Phantom allocation may provide the best trade-off in terms of computational effort versus achieved asymptotic MC probability decay rate.

Table 3. The first three rows of the table report the computational time t and the optimality gaps for the 3-objective, 3-system problem from Li et al. (2018). Then, for 10 MORS problems generated by the fixed Pareto method, each with $d \geq 3$ objectives and $r \geq 10$ systems, the rest of 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 ($Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^5$) or by the phantom approximation ($Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$). We do not report rates for MOCBA since it alternates between allocations.

d	r	p	Med. $\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	MVN True	MVN Phantom	MO-SCORE	MVN Ind.	iMO-SCORE	MOCBA	LD-based MOCBA	MOCBA*	MOCBA#	MOBCA+	Equal
3†	3	2	5	t	0.527s	0.0254s	0.0122s	0.534s	0.0107s	0.0019s	0.190s	24.364s	3.178s	0.998s	0s
				$Z^{\text{bf}}(\boldsymbol{\alpha})\times 10^5$	2,294.714	2,294.714	2,294.714	2,294.714	2,294.714		2,144.662	2,294.684	2,294.684	2,294.668	2,083.333
				$Z^{\text{ph}}(\boldsymbol{\alpha})\times 10^5$	2,294.714	2,294.714	2,294.714	2,294.714	2,294.714		2,144.662	2,294.684	2,294.684	2,294.668	2,083.333
3	10	5	11	Median T	1m 44s	0.05s	0.03s	1m 56s	0.023s	0.005s	9.07s	–	5m 42s	9.06s	0s
				75th %-ile T	2m 4s	0.05s	0.05s	2m 36s	0.026s	0.006s	1m 14s	–	8m 22s	1m 14s	0s
				$Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	950.316	950.297	924.602	702.155	695.678		736.045	–	930.916	736.043	393.455
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	948.024	950.297	924.602	702.155	695.678		736.045	–	930.916	736.043	393.455
	500	10	21	Median T	–	1m 37s	0.15s	–	0.09s	3.41s	6m 4s	–	–	5m 58s	0s
				75th %-ile T	–	2m 38s	0.17s	–	0.10s	3.45s	12m 53s	–	–	12m 31s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	0.171	0.167	–	0.170		0.029	–	–	0.029	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
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.0004	–	0.0003		–	–	–	–	3×10^{-7}
4	5,000	10	42	Median T	–	–	1m 4s	–	0.30s	7m 43s	–	–	–	–	0s
				75th %-ile T	–	–	1m 28s	–	0.36s	7m 44s	–	–	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.0013	–	0.0011		–	–	–	–	2×10^{-6}
	10,000	10	44	Median T	–	–	1m 52s	–	0.57s	31m 5s	–	–	–	–	0s
				75th %-ile T	–	–	2m 20s	–	0.61s	31m 9s	–	–	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.00009	–	0.00009		–	–	–	–	1×10^{-7}
5	10,000	10	90	Median T	–	–	8m 52s	–	1.15s	38m 16s	–	–	–	–	0s
				75th %-ile T	–	–	11m 13s	–	1.21s	38m 31s	–	–	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.0001	–	0.0001		–	–	–	–	2×10^{-7}

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.

† This set of results contains only one test problem instance, the 3-objective, 3-system problem from Li et al. (2018) which has independent objectives.

Table 4. 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 ($Z^{\text{bf}}_{0.5}(\boldsymbol{\alpha})\times 10^5$) or by the phantom approximation ($Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$). The MVN True, MVN Ind., MOCBA#, and MOCBA* allocations are excluded from the table due to large computational time or memory limitations. We do not report rates for MOCBA since it alternates between allocations.

d	r	Med. p	Med. $\lvert\mathcal{P}^{\text{ph}}\rvert$	Metric	MVN Phantom	MO-SCORE	iMO-SCORE	LD-based MOCBA	LD-based MOCBA+	LD-based MOCBA+	Equal
3	250	31	62	Median T	1m 20s	1s	1s	0.93s	19m 4s	1h 14m	0s
				75th %-ile T	2m 14s	3s	2s	0.94s	2h 22m	3h 6m	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	0.946	0.932	0.714		0.240	0.248	0.011
	5,000	165	330	Median T	–	12s	8s	5m 55s	–	–	0s
				75th %-ile T	–	13s	10s	5m 56s	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	8×10^{-6}	8×10^{-6}		–	–	3×10^{-7}
	10,000	245	490	Median T	–	47s	35s	23m 56s	–	–	0s
				75th %-ile T	–	1m 6s	38s	23m 59s	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	0.00001	0.00001		–	–	2×10^{-7}
4	50	21	89	Median T	6m 3s	56s	0.24s	0.055s	2m 25s	2m 25s	0s
				75th %-ile T	9m 42s	1m 11s	0.26s	0.056s	3m 26s	3m 10s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	16.659	16.252	12.460		7.743	7.696	1.269
	1,000	133	686	Median T	–	–	12s	18.60s	–	–	0s
				75th %-ile T	–	–	53s	18.65s	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.015		–	–	0.00006
	2,000	208	1,062	Median T	–	–	33s	1m 13.3s	–	–	0s
				75th %-ile T	–	–	34s	1m 13.7s	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.00006		–	–	4×10^{-6}
	5,000*	374	2,064	Median T	–	–	2m 36s	8m 32s	–	–	0s
				75th %-ile T	–	–	2m 42s	8m 38s	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	0.00005		–	–	2×10^{-6}
5	50	30	278	Median T	18m 18s	4m 38s	2s	0.08s	4m 20s	4m 47s	0s
				75th %-ile T	21m 44s	23m 6s	7s	0.08s	49m 27s	54m 33s	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	7.394	7.326	5.461		0.996	0.995	0.641
	2,000*	356	5,618	Median T	–	–	13m 31s	1m 45.2s	–	–	0s
				75th %-ile T	–	–	24m 25s	1m 45.8s	–	–	0s
				$Z^{\text{ph}}_{0.5}(\boldsymbol{\alpha})\times 10^5$	–	–	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.

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

8. 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. We provide sequential Algorithm 1 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 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

Algorithm 1: A sequential algorithm for implementation

Input: initial sample size $\delta_0 > d \geq 2$; sample size between allocation updates $\delta \geq 1$; minimum-sample proportion $0 < \alpha_\varepsilon \ll 1/r$; total budget $b \geq r \times \delta_0 + \delta$

1 Initialize: collect δ_0 replications from each system $s \in \mathcal{S}$; $n \leftarrow r \times \delta_0, n_s \leftarrow \delta_0 \ \forall \ s \in \mathcal{S}$
2 **repeat**
3 Initialize: $\delta_\varepsilon = 0$, $\mathcal{S}_\varepsilon \leftarrow \emptyset$, $\mathcal{S} \leftarrow \{1, \ldots, r\}$
4 **foreach** $s \in \mathcal{S}$ **if** $n_s/n < \alpha_\varepsilon$ **then** $\mathcal{S}_\varepsilon \leftarrow \mathcal{S}_\varepsilon \cup \{s\}$ `/systems needing simulation`
5 **if** $0 \leq |\mathcal{S}_\varepsilon| < \delta$ **then**
6 **if** $|\mathcal{S}_\varepsilon| \geq 1$ **then foreach** $s_\varepsilon \in \mathcal{S}_\varepsilon$ **do**
7 collect a simulation replication from system s_ε, $n_{s_\varepsilon} \leftarrow n_{s_\varepsilon} + 1$, $\delta_\varepsilon \leftarrow \delta_\varepsilon + 1$
8 *Calculate:* update rate function estimators for all systems $s \in \mathcal{S}$ by updating estimators for parameters in the assumed distributional family
9 *Solve:* an estimated version of Problem Q^{ph}, $Q_{\mathbb{S}}$, or $Q_{\mathbb{S}}^{\text{ind}}$, to obtain $\hat{\boldsymbol{\alpha}}^*$
10 **else**
11 $\mathcal{S} \leftarrow \mathcal{S}_\varepsilon$, $\hat{\boldsymbol{\alpha}}^* \leftarrow (1/|\mathcal{S}_\varepsilon|, \ldots, 1/|\mathcal{S}_\varepsilon|)$ `/simulate from` $s \in \mathcal{S}_\varepsilon$ `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 $\mathcal{S}$, where for each m, X_m is an i.i.d. random variable with probability mass function $\hat{\boldsymbol{\alpha}}^*$ supported on $\mathcal{S}$
14 *Simulate:* collect one simulation replication from system X_m, $n_{X_m} \leftarrow n_{X_m} + 1$
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*

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 in how the minimum-sampling requirement is implemented. We write our algorithm to 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. Optionally, the algorithm may be terminated by a certain amount of wall-clock time having passed.

9. Numerical Performance of the Sequential Implementations

In this section, we implement our proposed allocations using sequential Algorithm 1 and compare them with competitors on two test problem sets.

9.1. Test Problem Set I

Our first test problem set is from L. H. Lee et al. (2010). There are three objectives, 25 systems, and 5 Pareto systems, shown in Figure 2. We consider three problems: version (a), 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 versions (b) and (c), 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, respectively. The objective values, allocations for MVN Phantom, MO-SCORE, iMO-SCORE, and MOCBA+ on version (a), and their corresponding rates are listed in the online supplement.

To match the parameter settings from L. H. Lee et al. (2010), we implement the sequential algorithms as follows. For MOCBA, we set the number of initial samples

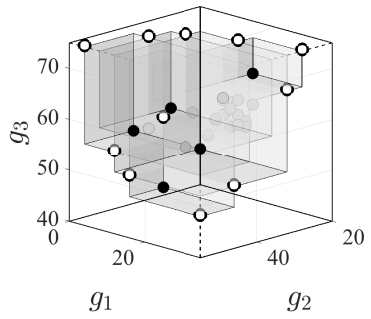

Figure 2. Test Problem Set I: For $d = 3$, 5 Paretos are black, 11 phantom Paretos are white, and 20 non-Paretos are gray.

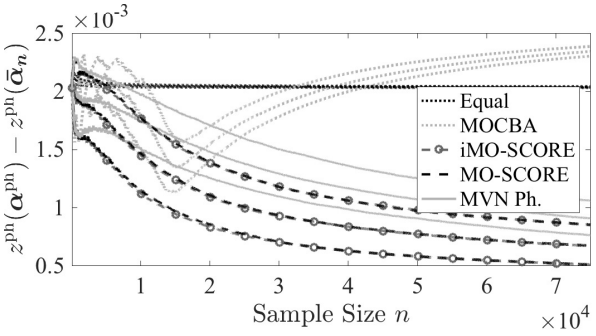

Figure 3. Test Problem Set I, version (a), $\rho = 0$: The figure shows sample quantiles (.25, .5, .75) of the optimality gap over 5,000 independent runs of each sequential allocation strategy.

$N_0 = 5$, the number of samples between allocation updates to $\Delta = 10$, and the maximum samples to a single system $\tau = \Delta/2 = 5$. For MOCBA+, we use code provided by the authors of Li et al. (2018) with the same parameter values ($N_0 = 5, \Delta = 10$). Due to large computational time, we do not implement LD-based MOCBA, MOCBA*, or MOCBA#. In our sequential 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 $\alpha_\varepsilon = 1 \times 10^{-8} \ll 1/r$. The total simulation budget is $b \leq 75{,}000$. (Given δ_0 and b, extra sampling due to α_ε does not occur.) For each allocation scheme, let $\bar{\boldsymbol{\alpha}}_n = (n_1/n, \ldots, n_r/n)$ denote the vector of proportional allocations expended by the sequential algorithm as a function of the sample size n.

Figure 3 shows sample quantiles of the approximate optimality gap of $\bar{\boldsymbol{\alpha}}_n$, $z^{\mathrm{ph}}(\boldsymbol{\alpha}^{\mathrm{ph}}) - z^{\mathrm{ph}}(\bar{\boldsymbol{\alpha}}_n)$, calculated across 5,000 independent replications of each sequential algorithm on the test problem version (a). Note that there is dependence across the values of n in Figure 3. (Due to large computational time, we exclude MOCBA+ from this figure. Further, graphing z^{bf} is too computationally intensive, thus we graph only the phantom rates, z^{ph}.) From the perspective of the optimality gap of the allocation expended, MO-SCORE and iMO-SCORE 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 for large enough sample size n, the bounds that MOCBA uses to determine whether to control for $\mathbb{P}\{\mathrm{MCE}\}$ or $\mathbb{P}\{\mathrm{MCI}\}$ are both estimated to be zero, to the numerical precision of the computer. By default in this case, MOCBA allocates to control $\mathbb{P}\{\mathrm{MCE}\}$ only, which is suboptimal with respect to the phantom rate in this test problem. However, by the time the allocation scheme for MOCBA goes awry in Figure 3, the actual estimated $\mathbb{P}\{\mathrm{MC}\}$ is already very small for MOCBA — nearly zero, as we will see from Figure 4.

Figure 4 shows the estimated MC probability as a function of n for each sequential allocation strategy, calculated across 10,000 independent replications of equal allocation, MOCBA, iMO-SCORE, MO-SCORE, and MVN Phantom. Due to large computational time, the results for MOCBA+ include only 1,000 replications. To account for this discrepancy in results, the line for MOCBA+ includes an approximate 95% confidence interval cloud calculated as $\hat{p} \pm 1.96 \times \sqrt{\hat{p}(1-\hat{p})/1000}$, where $\hat{p}$ denotes the estimated value of $\mathbb{P}\{\mathrm{MC}\}$, which is a function of n, in Figure 4. The online appendix contains complete results with estimated values of $\mathbb{P}\{\mathrm{MCE}\}$ and $\mathbb{P}\{\mathrm{MCI}\}$, in which case $\hat{p}$ may denote the estimated value of $\mathbb{P}\{\mathrm{MCE}\}$ or $\mathbb{P}\{\mathrm{MCI}\}$, respectively.

From Figure 4, we see that MO-SCORE and iMO-SCORE perform nearly identically

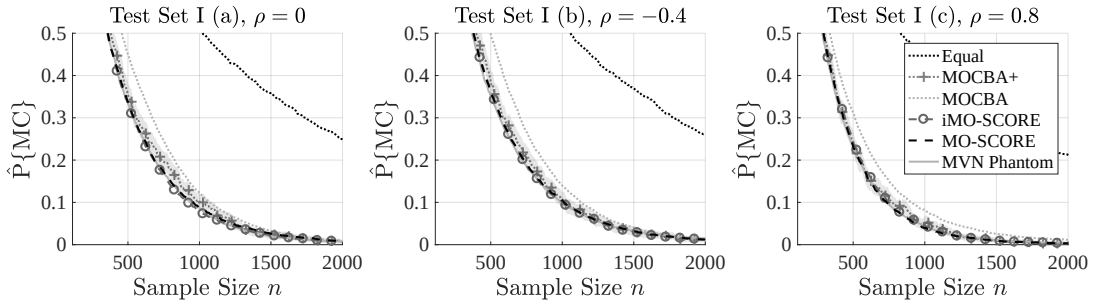

Figure 4. Test Problem Set I, versions (a) $\rho = -0.4$, (b) $\rho = 0$, and (c) $\rho = 0.8$: The figure shows the estimated $\mathbb{P}\{\mathrm{MC}\}$ calculated across 10,000 independent sample paths of each sequential allocation strategy except MOCBA+, which is calculated across 1,000 independent sample paths due to large computational time. Thus the MOCBA+ line includes a 95% confidence interval cloud in light gray.

to MVN Phantom. In terms of the overall estimated $\mathbb{P}\{\mathrm{MC}\}$, these three algorithms appear to slightly out-perform MOCBA+, which in turn appears to slightly out-perform MOCBA. Finally, in this test set, correlation seems to have a minor effect on the estimated $\mathbb{P}\{\mathrm{MC}\}$. The performance of MOCBA+ appears closer to that of MO-SCORE and iMO-SCORE in Test Set I (c), in which $\rho = 0.8$.

9.2. Test Problem II

Our second test problem setting 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 5; all other three-objective projections look similar. The minimum distance between any two Pareto systems on any objective is approximately 0.0953, and the minimum distance between a Pareto system and a non-Pareto system on any objective is approximately 0.0129. The true objective function values for this test problem are listed in the online appendix.

Fixing the 500 systems in Figure 5, we set all systems' covariance matrices to the identity matrix. Due to the size of Test Problem 2, we implement only iMO-SCORE and MOCBA. We require a larger δ_0 to estimate the covariance matrices, and we can afford fewer allocation updates. Thus iMO-SCORE has $\delta_0 = 15$, $\delta = 150$, $\alpha_\varepsilon = 1 \times 10^{-8} \ll 1/r$, and $b \leq 15,000$; MOCBA has $N_0 = 15$, $\Delta = 150$, and $\tau = \Delta/2 = 75$.

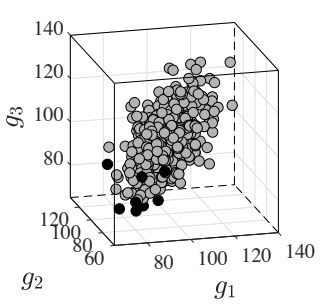

Figure 5. Test Problem II: For $d = 4$ total objectives, the figure shows objectives 1, 2, and 3 with 8 Paretos in black.

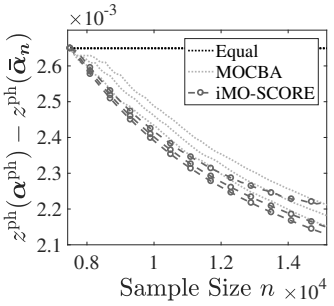

Figure 6. Test Problem II: The figure shows sample quantiles (.25, .5, .75) of the opt. gap over 10,000 independent runs per algorithm.

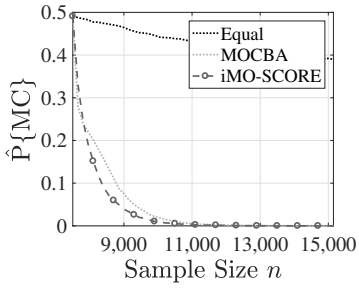

Figure 7. Test Problem II: The figure shows the estimated $\mathbb{P}\{\mathrm{MC}\}$, calculated across 10,000 independent runs per algorithm.

Figures 6 and 7 display the results for Test Problem II. On this test problem, Figure 6 shows that the median optimality gap of the actual allocation for iMO-SCORE is consistently smaller than that of MOCBA. Notice that the sample size is not as large as for Test Problem Set I, so it is not clear if MOCBA turns suboptimal for larger n. The 75th percentile line for iMO-SCORE 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 7 shows that MOCBA and iMO-SCORE are close in terms of the estimated $\mathbb{P}\{\text{MC}\}$, but iMO-SCORE appears to perform slightly better.

10. Concluding Remarks

The question of how to efficiently identify the entire Pareto set in MORS problems is challenging — primarily because we need to decide how to allocate a given simulation budget across the competing systems to minimize the likelihood of misclassifying any of the systems. We demonstrate that this question of simulation budget allocation can be posed, without approximation, as a concave maximization problem by specifying the decay rates of the MC event probabilities obtained through brute-force enumeration. Solving this concave maximization problem, however, becomes prohibitively expensive 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.

MO-SCORE and iMO-SCORE are MORS solution algorithms that address the computational issue in a disciplined way. Through a series of approximations obtained by asymptotic approximation and the strategic relaxation of constraints, 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 MO-SCORE and iMO-SCORE can reliably solve many MORS problems with several thousand systems in three or more objectives.

Funding

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

References

Batur, D., Wang, L., & Choobineh, F. F. (2018, November). Methods for system selection based on sequential mean-variance analysis. *INFORMS Journal on Computing*, *30*(4), 625–786.

Branke, J., & Zhang, W. (2015). 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, & M. D. Rossetti (Eds.), *Proceedings of the 2015 winter simulation conference* (pp. 3589–3598). Piscataway, NJ: IEEE.

Branke, J., Zhang, W., & Tao, Y. (2016). Multiobjective ranking and selection based on hypervolume. In T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, &

S. E. Chick (Eds.), *Proceedings of the 2016 winter simulation conference* (pp. 859–870). Piscataway, NJ: IEEE.

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

Chan, T. M. (2013). Klee's measure problem made easy. In *2013 ieee 54th annual symposium on foundations of computer science* (pp. 410–419). Piscataway, NJ: IEEE.

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

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

Choi, S. H., & Kim, T. G. (2018). Pareto set selection for multiobjective stochastic simulation model. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Dembo, A., & Zeitouni, O. (1998). *Large deviations techniques and applications* (2nd ed.). New York: Springer.

Ding, H., Benyoucef, L., & Xie, X. (2006). A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. *Engineering Applications of Artificial Intelligence*, *19*, 609–623.

Dudewicz, E. J., & Taneja, V. S. (1978). Multivariate ranking and selection without reduction to a univariate problem. In H. J. Highland, N. R. Nielsen, & L. G. Hull (Eds.), *Proceedings of the 1978 winter simulation conference* (pp. 207–210). Piscataway, NJ: IEEE.

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

Ehrgott, M. (2005). *Multicriteria optimization* (2nd ed., Vol. 491). Heidelberg: Springer.

Eichfelder, G. (2008). *Adaptive scalarization methods in multiobjective optimization.* Berlin Heidelberg: Springer.

Feldman, G. (2017). *Sampling laws for multi-objective simulation optimization on finite sets* (Unpublished doctoral dissertation). Purdue University, West Lafayette, IN, USA.

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

Feldman, G., Hunter, S. R., & Pasupathy, R. (2015). 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, & M. D. Rossetti (Eds.), *Proceedings of the 2015 winter simulation conference* (pp. 3610–3621). Piscataway, NJ: IEEE.

Frazier, P. I., & Kazachkov, A. M. (2011). Guessing preferences: a new approach to multi-attribute ranking and selection. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, & M. Fu (Eds.), *Proceedings of the 2011 winter simulation conference* (pp. 4324 – 4336). Piscataway, NJ: IEEE.

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

Fu, M., & Henderson, S. G. (2017). History of seeking better solutions, aka simulation optimization. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, & E. Page (Eds.), *Proceedings of the 2017 winter simulation conference* (pp. 131–157). Piscataway, NJ: IEEE.

Glynn, P. W., & Juneja, S. (2004). A large deviations perspective on ordinal optimization. In R. G. Ingalls, M. D. Rossetti, J. S. Smith, & B. A. Peters (Eds.), *Proceedings of the 2004 winter simulation conference* (pp. 577–585). Piscataway, NJ: IEEE.

Glynn, P. W., & Juneja, S. (2011). Ordinal optimization: a nonparametric framework. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, & M. Fu (Eds.), *Proceedings of the 2011 winter simulation conference* (pp. 4057 – 4064). Piscataway, NJ: IEEE.

Glynn, P. W., & Juneja, S. (2015). *Ordinal optimization – empirical large deviations rate estimators, and stochastic multi-armed bandits.* Retrieved from `http://arxiv.org/abs/1507.04564`

Hunter, S. R. (2011). *Sampling laws for stochastically constrained simulation optimization on finite sets* (Unpublished doctoral dissertation). Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.

Hunter, S. R., Applegate, E. A., Arora, V., Chong, B., Cooper, K., Rincón-Guevara, O., & Vivas-Valencia, C. (2019, January). An introduction to multi-objective simulation optimization. *ACM Transactions on Modeling and Computer Simulation*, *29*(1), 7:1–7:36.

Hunter, S. R., & Feldman, G. (2015). 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, & M. D. Rossetti (Eds.), *Proceedings of the 2015 winter simulation conference* (pp. 3749–3757). Piscataway, NJ: IEEE.

Hunter, S. R., & McClosky, B. (2016). Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation. *IIE Transactions*, *48*(6), 565–578.

Hunter, S. R., & Nelson, B. L. (2017). Parallel ranking and selection. In A. Tolk, J. Fowler, G. Shao, & E. Yücesan (Eds.), *Advances in modeling and simulation: Seminal research from 50 years of winter simulation conferences* (pp. 249–275). Switzerland: Springer International.

Hunter, S. R., & Pasupathy, R. (2010). Large-deviation sampling laws for constrained simulation optimization on finite sets. In B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, & E. Yücesan (Eds.), *Proceedings of the 2010 winter simulation conference* (pp. 995–1002). Piscataway, NJ: IEEE.

Hunter, S. R., & Pasupathy, R. (2013). Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS Journal on Computing*, *25*(3), 527–542.

Hunter, S. R., Pujowidianto, N. A., Chen, C., Lee, L. H., Pasupathy, R., & Yap, C. M. (2011). 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, & M. Fu (Eds.), *Proceedings of the 2011 winter simulation conference* (pp. 4294–4302). Piscataway, NJ: IEEE.

Kaplan, H., Rubin, N., Sharir, M., & Verbin, E. (2008). Efficient colored orthogonal range counting. *SIAM J. Comput.*, *38*(3), 982–1011.

Kim, S.-H., & Nelson, B. L. (2006). Selecting the best system. In S. G. Henderson & B. L. Nelson (Eds.), *Simulation* (pp. 501–534). Amsterdam, The Netherlands: Elsevier.

Lacour, R., Klamroth, K., & Fonseca, C. M. (2017, March). A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, *79*, 347–360.

Lee, J. S. (2014). *Advances in simulation: validity and efficiency* (Doctoral dissertation, Georgia Institute of Technology, Atlanta, GA, USA). Retrieved from `http://hdl.handle.net/1853/53457`

Lee, L. H., Chew, E. P., Teng, S., & Goldsman, D. (2010). Finding the non-dominated Pareto set for multi-objective simulation models. *IIE Transactions*, *42*, 656-674.

Li, J. (2012). *Optimal computing budget allocation for multi-objective simulation optimization* (Unpublished doctoral dissertation). National University of Singapore, Singapore.

Li, J., Liu, W., Pedrielli, G., Lee, L. H., & Chew, E. P. (2018, September). Optimal computing budget allocation to select the non-dominated systems – a large deviations perspective. *IEEE Transactions on Automatic Control*, *63*(9), 2913–2927.

Mattila, V., & Virtanen, K. (2015). Ranking and selection for multiple performance measures using incomplete preference information. *European Journal of Operational Research*, *242*, 568–579.

Merrick, J. R. W., Morrice, D., & Butler, J. C. (2015). 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.

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

Pujowidianto, N. A., Hunter, S. R., Pasupathy, R., Lee, L. H., & Chen, C. (2012). Closed-form sampling laws for stochastically constrained simulation optimization on large finite sets. In

C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, & A. M. Uhrmacher (Eds.), *Proceedings of the 2012 winter simulation conference* (pp. 168–177). Piscataway, NJ: IEEE.

Ryzhov, I. O. (2016). On the convergence rates of expected improvement methods. *Operations Research*, *64*(6), 1515-1528.

Szechtman, R., & Yücesan, E. (2008). A new perspective on feasibility determination. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, & J. W. Fowler (Eds.), *Proceedings of the 2008 winter simulation conference* (pp. 273–280). Piscataway, NJ: IEEE.

Teng, S., Lee, L. H., & Chew, E. P. (2010). Integration of indifference-zone with multi-objective computing budget allocation. *European Journal of Operational Research*, *203*(2), 419–429.

Wang, W., & Wan, H. (2017). Sequential probability ratio test for multiple-objective ranking and selection. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, & E. Page (Eds.), *Proceedings of the 2017 winter simulation conference* (pp. 1998–2009). Piscataway, NJ: IEEE.

Wiecek, M. M., Ehrgott, M., & Engau, A. (2016). Continuous multiobjective programming. In S. Greco, M. Ehrgott, & J. R. Figueira (Eds.), *Multiple criteria decision analysis: State of the art surveys* (Vol. 233, pp. 739–815). New York: Springer New York.

Yildiz, H., & Suri, S. (2012). On Klee's measure problem for grounded boxes. In *Proceedings of the twenty-eighth annual symposium on computational geometry* (pp. 111–1120). New York, NY: ACM.

Zhang, H. (2008). Multi-objective simulation-optimization for earthmoving operations. *Automation in Construction*, *18*, 79–86.

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

Eric A. Applegate[a], Guy Feldman[b], Susan R. Hunter[a], and Raghu Pasupathy[b]

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

Appendix A. Proof Sketch for Lemma 3.2

Using the brute-force formulation, if the limits exist, we have that the $\mathbb{P}\{\text{MCI}\}$ decay rate is

$$-\lim_{n\to\infty}\tfrac{1}{n}\log\mathbb{P}\{\text{MCI}\} = \min_{j\in\mathcal{P}}\min_{\boldsymbol{\kappa}\in\mathcal{K}}\big(-\lim_{n\to\infty}\tfrac{1}{n}\log\mathbb{P}\{\text{MCI}^{\text{bf}}(j,\boldsymbol{\kappa})\}\big),$$

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 $\mathbb{P}\{\text{MCI}^{\text{bf}}(j,\boldsymbol{\kappa})\}$ decay rate, 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 $\widehat{\boldsymbol{G}}(j)$. Applying Proposition 1.6.1 and Lemma 1.6.2 of Feldman (2017, p. 14–16) and the Contraction Crinciple (Dembo & Zeitouni, 1998, §4.2), the random vector $(\widehat{\boldsymbol{G}}(j),\widehat{G}_{\kappa_1}(1),\ldots,\widehat{G}_{\kappa_p}(p))$ obeys a Large Deviations Principle (LDP) with good rate function $\alpha_j I_j(\mathbf{x}_j) + \sum_{i\in\mathcal{P}}\alpha_i J_{i\kappa_i}(x_{i\kappa_i})$. Then, the result follows by applying the Gärtner-Ellis Theorem (Dembo & Zeitouni, 1998, p. 44).

Appendix B. Proof of Lemma 4.3

Proof. Suppose $\boldsymbol{\kappa},\boldsymbol{\kappa}'\in\mathcal{K}$ are two vectors of objective indices such that $\mathbf{g}_d^{\text{bf}}(\boldsymbol{\kappa}') \leq \mathbf{g}_d^{\text{bf}}(\boldsymbol{\kappa})$. Let $\mathcal{X}' := \{\mathbf{x}' : \mathbf{x}' \leqq \mathbf{g}_d^{\text{bf}}(\boldsymbol{\kappa}')\}$ and $\mathcal{X} := \{\mathbf{x} : \mathbf{x} \leqq \mathbf{g}_d^{\text{bf}}(\boldsymbol{\kappa})\}$. Then $\mathcal{X}'\subseteq\mathcal{X}$. Since $\mathbf{g}(j)\notin\mathcal{X}$, using Lemma 4.2,

$$\mathbb{S}_j(\boldsymbol{\kappa}) = \inf_{\mathbf{x}_j\leq\mathbf{g}_d^{\text{bf}}(\boldsymbol{\kappa})} I_j(\mathbf{x}_j) \leq \inf_{\mathbf{x}_j\leq\mathbf{g}_d^{\text{bf}}(\boldsymbol{\kappa}')} I_j(\mathbf{x}_j) = \mathbb{S}_j(\boldsymbol{\kappa}').$$ □

Appendix C. 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 a 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).

Corresponding author: susanhunter@purdue.edu.

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 $[1] = \text{argmax}_{1 \leq i \leq p'} g_{k^*}(i)$ 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 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

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 $\quad \mathcal{G}^{\text{ph}}_{\text{sweep}} \leftarrow \min_{1 \leq i \leq p'} g_1(i)$
3 **else**
4 $\quad k^* \leftarrow d'$ `/choose` k^* `as largest objective`
5 $\quad$ 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 $\quad$ **for** *$i = 1$ to $p' - (d'-1)$* **do**
7 $\quad\quad$ 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 $\quad\quad \mathcal{G}_{d'-1} \leftarrow \{\mathbf{g}'(j) \colon \mathbf{g}'(j) = (g_1(j), \ldots, g_{d'-1}(j))$ for all j indexing points in $\mathcal{G}_{d'}\}$
9 $\quad\quad \mathcal{G}^*_{d'-1}$ =GETPARETOS($\mathcal{G}_{d'-1}$)
10 $\quad\quad \mathcal{G}^{\text{ph}}_{d'-1}$ =SWEEP($\mathcal{G}^*_{d'-1}$) `/points in` $\mathcal{G}^{\text{ph}}_{d'-1}$ `are` $(d'-1)$`-dimensional phantoms`
11 $\quad\quad \mathcal{G}^{\text{ph}}_{d'-1} \leftarrow \mathcal{G}^{\text{ph}}_{d'-1} \setminus \{\mathbf{g} \in \mathcal{G}^{\text{ph}}_{d'-1} : (g_1([i]), \ldots, g_{d'-1}([i])) \lneq \mathbf{g}\}$
12 $\quad\quad \mathcal{G}^{\text{ph}}_{d'} \leftarrow \{\mathbf{g}'(j) : \mathbf{g}'(j) = (g_1(j), \ldots, g_{d'-1}(j), g_{\max})$ for all j indexing points in $\mathcal{G}^{\text{ph}}_{d'-1}\}$
13 $\quad\quad \mathcal{G}^{\text{ph}}_{\text{sweep}} \leftarrow \mathcal{G}^{\text{ph}}_{\text{sweep}} \cup \mathcal{G}^{\text{ph}}_{d'}$
14 **return** $\mathcal{G}^{\text{ph}}_{\text{sweep}}$

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.)

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 ∞ 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.

Appendix D. Example: Why the Brute-Force and Phantom Rates are Not Equal in Three or More Objectives

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 D1. Each system's covariance matrix is the identity matrix, and all rates are calculated under equal allocation. Rounded to three decimal places, the rates $R_{12}^{\text{MCE}}(1/3, 1/3) = 0.771$, and $R_{21}^{\text{MCE}}(1/3, 1/3) = 0.750$. The pairwise brute-force and phantom $\mathbb{P}\{\text{MCI}\}$ rates are reported in Table D1. In this example, the overall brute-force rate is $z^{\text{bf}} = 0.737$, while the overall phantom rate is $z^{\text{ph}} = 0.653$.

This discrepancy in rates occurs because in three or more objectives, the phantom rate 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 implies

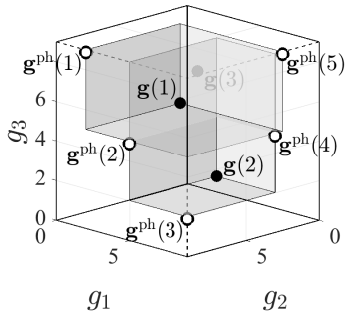

Figure D1. The figure shows the objective vector values for the systems in Problem N. We omit arrows from the phantoms with coordinates at infinity.

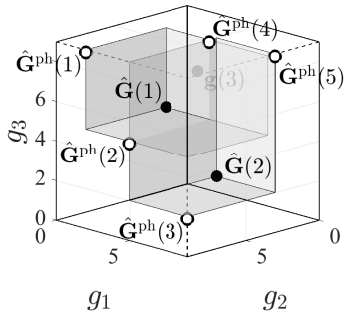

Figure D2. The figure shows estimated values for the Pareto systems in Problem N in which Pareto systems 1 and 2 have "switched places" on objective 2.

Table D1. Pairwise rates for Problem N under equal allocation, to three 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
1.333	1.774	4.333	0.737	0.743	3.653	0.833	1.191	3.000	1.333	0.833	3.000	1.191	0.653

non-Pareto system 3 is 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 D2, the estimated phantom Pareto system $\widehat{\boldsymbol{G}}^{\text{ph}}(4)$ would not correspond to the approximate location of $\mathbf{g}^{\text{ph}}(4)$ from Figure D1. 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 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.

Appendix E. Test Problem Set I: Supplemental Results

The objective vector values for Test Problem Set I appear in Table E1. Systems with indices 1, 2, 4, 5, and 9 are Pareto. For comparison, Table E1 also reports the (non-sequential) allocations to each system determined by MVN Phantom, MO-SCORE, iMO-SCORE, and MOBCA+. As shown in the table, the allocations differ only slightly. The brute force rates for the MVN Phantom, MO-SCORE, and iMO-SCORE allocations are identical to four decimal places; the rate for MOCBA+ is lower.

The MOCBA+ allocation was calculated using code provided by Li et al. (2018). We note here that due to operations that may divide by zero, the implementation of MOCBA+ uses a minimum-sample parameter that forces a certain minimum allocation to all systems. The identical allocations to systems 6 and 10–25 in MOCBA+ are due

Table E1. The table displays the objective vector values from L. H. Lee et al. (2010) used in Test Problem Set I. The rows of Pareto systems appear in gray. The table also shows the (non-sequential) MVN Phantom, MO-SCORE, iMO-SCORE, and MOCBA+ allocations and corresponding brute force rates for Test Problem Set I.

	Obj.†			% Allocations for Test Set I (a), $\rho = 0$				% Allocations for Test Set I (b), $\rho = -0.4$				% Allocations for Test Set I (c), $\rho = 0.8$			
System	g_1	g_2	g_3	MVN Phantom	MO-SCORE	iMO-SCORE	MOCBA+	MVN Phantom	MO-SCORE	iMO-SCORE	MOCBA+	MVN Phantom	MO-SCORE	iMO-SCORE	MOCBA+
1	8	36	60	2.363861	2.207141	2.207141	2.196812	2.374566	2.222935	2.207141	2.196460	2.361931	2.203656	2.207141	2.196462
2	12	32	52	20.833111	19.346949	19.346949	19.366660	20.921567	19.407769	19.346949	19.366828	20.820127	19.333487	19.346949	19.366840
3	14	38	54	14.335105	13.422298	13.422298	13.320156	14.402119	13.546537	13.422298	13.320356	14.321979	13.394942	13.422298	13.320371
4	16	46	48	2.363861	2.207141	2.207141	2.196812	2.374566	2.222935	2.207141	2.196539	2.361931	2.203656	2.207141	2.196462
5	4	42	56	16.983792	19.346949	19.346949	15.784161	17.060218	19.407769	19.346949	15.784357	16.970265	19.333487	19.346949	15.784372
6	18	40	62	0.908590	1.342230	1.342230	0.931428	0.547073	0.846659	1.342230	0.931429	0.987510	1.488327	1.342230	0.931429
7	10	44	58	16.983792	13.422298	13.422298	15.784161	17.060218	13.546537	13.422298	15.784357	16.970265	13.394942	13.422298	15.784372
8	20	34	64	14.335105	13.422298	13.422298	13.320156	14.402119	13.546537	13.422298	13.320356	14.321979	13.394942	13.422298	13.320371
9	22	28	68	2.363861	2.207141	2.207141	2.196812	2.374566	2.222935	2.207141	2.196460	2.361931	2.203656	2.207141	2.196462
10	24	40	62	0.544618	0.838894	0.838894	0.931428	0.547073	0.846659	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
11	26	38	64	0.988305	1.491366	1.491366	0.931428	0.992764	1.505171	1.491366	0.931429	0.987510	1.488327	1.491366	0.931429
12	28	40	66	0.544618	0.838894	0.838894	0.931428	0.536726	0.824572	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
13	30	42	62	0.345306	0.536892	0.536892	0.931428	0.346861	0.541861	0.536892	0.931429	0.345030	0.535798	0.536892	0.931429
14	32	44	64	0.238587	0.372842	0.372842	0.931428	0.238661	0.374312	0.372842	0.931429	0.238397	0.372082	0.372842	0.931429
15	26	40	66	0.544618	0.838894	0.838894	0.931428	0.536726	0.824572	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
16	28	42	64	0.345306	0.536892	0.536892	0.931428	0.346861	0.541861	0.536892	0.931429	0.345030	0.535798	0.536892	0.931429
17	32	38	66	0.988305	1.491366	1.491366	0.931428	0.990735	1.497249	1.491366	0.931429	0.987510	1.488327	1.491366	0.931429
18	30	40	62	0.544618	0.838894	0.838894	0.931428	0.547073	0.846659	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
19	34	42	64	0.345306	0.536892	0.536892	0.931428	0.346861	0.541861	0.536892	0.931429	0.345030	0.535798	0.536892	0.931429
20	26	44	60	0.544618	0.838894	0.838894	0.931428	0.536726	0.824572	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
21	28	38	66	0.988305	1.491366	1.491366	0.931428	0.990735	1.497249	1.491366	0.931429	0.987510	1.488327	1.491366	0.931429
22	32	40	62	0.544618	0.838894	0.838894	0.931428	0.547073	0.846659	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
23	30	46	64	0.238587	0.372842	0.372842	0.931428	0.204354	0.316086	0.372842	0.931429	0.238397	0.372082	0.372842	0.931429
24	32	44	66	0.238587	0.372842	0.372842	0.931428	0.226688	0.353388	0.372842	0.931429	0.238397	0.372082	0.372842	0.931429
25	30	40	64	0.544618	0.838894	0.838894	0.931428	0.547073	0.846659	0.838894	0.931429	0.544182	0.837184	0.838894	0.931429
Rate $z^{\text{bf}}(\boldsymbol{\alpha})$				0.037500	0.037500	0.037500	0.002466	0.037500	0.037500	0.037500	0.002466	0.037500	0.037500	0.037500	0.002466
Rate $z^{\text{ph}}(\boldsymbol{\alpha})$				0.037500	0.037500	0.037500	0.002466	0.037500	0.037500	0.037500	0.002466	0.037500	0.037500	0.037500	0.002466
Comp. time t‡				0.1s	< 0.1s	< 0.1s	23s	0.12s	< 0.1s	< 0.1s	21s	0.1s	< 0.1s	< 0.1s	27s

† For numerical stability, all sequential allocations and results were computed with the true objective vector values shifted to $(g_1 + 1000, g_2 + 1000, g_3 + 1000)$.
‡ Computed in MATLAB R2017a on a 3.5 Ghz Intel Core i7 processor with 16GB 2133 MHz LPDDR3 memory.

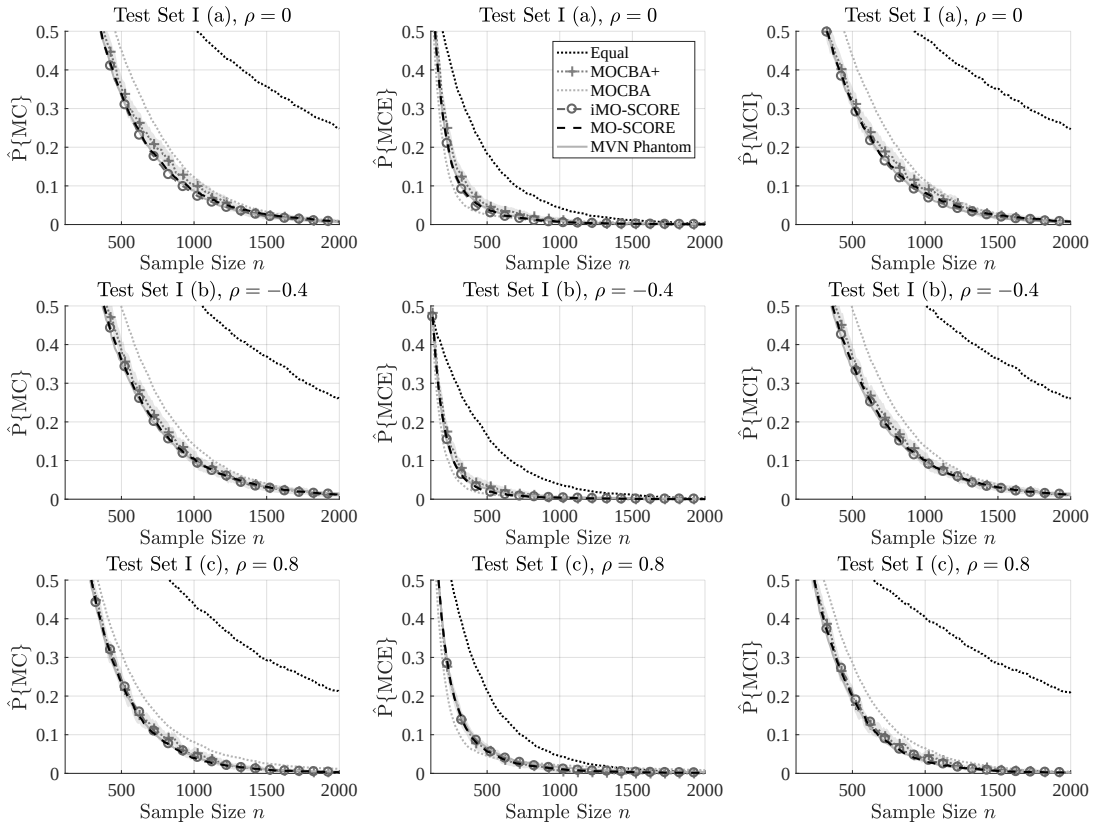

Figure E1. Test Problem Set I, versions (a) $\rho = -0.4$, (b) $\rho = 0$, and (c) $\rho = 0.8$: The figure shows the estimated $\mathbb{P}\{\mathrm{MC}\}, \mathbb{P}\{\mathrm{MCE}\}$, and $\mathbb{P}\{\mathrm{MCI}\}$ calculated across 10,000 independent sample paths of each sequential allocation strategy except MOCBA+, which is calculated across 1,000 independent sample paths. Thus the MOCBA+ line includes a 95% confidence interval cloud in light gray.

to this parameter. (Since this parameter exists in the non-sequential version of MOCBA+, it is not analogous to our α_ε.) Reducing this parameter seems to increase the computation time, therefore, we did not change it in the code. However, Test Problem Set I contains problems larger than the 3-objective test problems in Li et al. (2018). Therefore it is not clear whether updating this parameter would help the performance of MOCBA+ in Figure 4.

Finally, while a single update of MOCBA+ took 22 seconds on Test Set I (a), each update of the sequential allocation exhibited significant variance, from less than 40 seconds to about 8 minutes, depending on the nature of the random problems encountered during the sequential setting. For comparison, a single update of MVN Phantom ranged from less than 1 second to about a minute. Due to the increased amount of time for the sequential allocation updates in MOCBA+, we were only able to complete 1,000 runs of MOCBA+ for this test problem set.

Figure E1 shows the performance of each sequential allocation strategy on Test Problem Set I in terms of the estimated $\mathbb{P}\{\mathrm{MC}\}, \mathbb{P}\{\mathrm{MCE}\}$, and $\mathbb{P}\{\mathrm{MCI}\}$. MOCBA and our proposed allocation strategies perform similarly with regards to estimated $\mathbb{P}\{\mathrm{MCE}\}$, with MOCBA performing slightly better. Our proposed allocation strategies seem to have a lower estimated $\mathbb{P}\{\mathrm{MCI}\}$ and lower estimated overall $\mathbb{P}\{\mathrm{MC}\}$ compared to both MOCBA and MOCBA+ for Test Set I (a) and (b). The performances of these algorithms appear closer to each other on Test Set I (c).

Appendix F. Test Problem II: Supplemental Results

Figure F1 shows the performance of each sequential allocation strategy on Test Problem II in terms of the estimated $\mathbb{P}\{\text{MC}\}, \mathbb{P}\{\text{MCE}\}$, and $\mathbb{P}\{\text{MCI}\}$. On this test problem, iMO-SCORE and MOCBA both do well, with iMO-SCORE having a slightly better overall performance on the estimated $\mathbb{P}\{\text{MC}\}$ and estimated $\mathbb{P}\{\text{MCE}\}$.

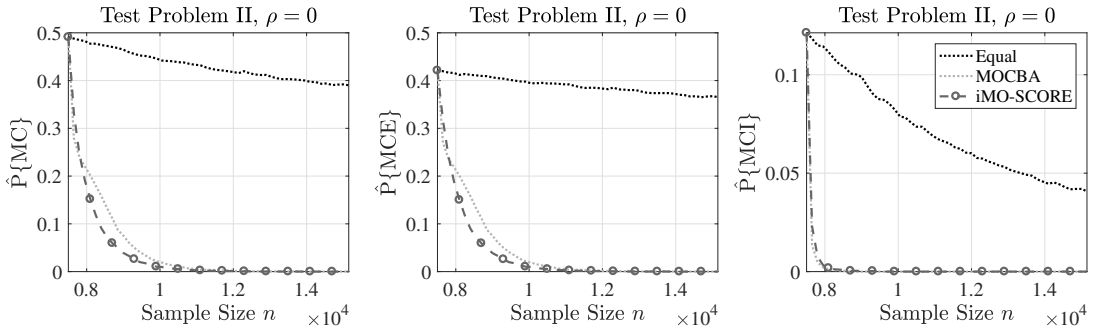

Figure F1. Test Problem II: The figure shows the estimated $\mathbb{P}\{\text{MC}\}, \mathbb{P}\{\text{MCE}\}$, and $\mathbb{P}\{\text{MCI}\}$ calculated across 10,000 independent sample paths for each sequential allocation strategy.

The following Table F1 presents the objective function values of the systems in Test Problem II. As was the case with Test Problem Set I, for numerical stability, all sequential allocations and results were computed with the true objective vector values shifted to $(g_1 + 1000, g_2 + 1000, g_3 + 1000)$.

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
1	70.76402064	92.54167954	92.24487835	89.41252266
2	72.98472792	80.55209548	75.00835128	64.05286572
3	73.44794249	88.06655288	81.86434643	89.16965672
4	74.19242797	89.81479498	87.89735012	100.58456380
5	76.39246424	111.09271350	94.26810557	94.15626728
6	76.54974812	96.28767891	84.62628046	87.23093539
7	77.52099288	82.96168605	81.37200517	92.32252553
8	78.99949135	85.35918640	89.80670601	77.25423473
9	79.11204242	93.16860142	77.23760003	83.30012274
10	79.60012562	90.90292361	90.10205460	100.22941900
11	80.41098852	84.43515883	86.41487114	88.15954444
12	80.45269968	80.45681700	77.54198415	78.88024530
13	80.53887217	90.34398157	92.30428304	77.12832945
14	81.16840775	82.49686598	72.84173860	92.96632442
15	81.62439135	90.93519352	78.35146197	85.12523716
16	81.75225758	85.40766062	87.09836032	92.34244237
17	81.94069999	83.87560095	85.15697766	82.15807279
18	82.09217747	83.35829564	77.67573565	70.66409221
19	82.35735073	77.08092051	90.05182921	90.66106663
20	82.58036250	96.32193992	84.18206846	91.68957794
21	82.98832844	83.87975109	96.41361364	82.13305860
22	83.04118434	92.66916163	78.45566859	78.51321236
23	83.32517372	109.19945370	94.68150929	94.16358862
24	83.49159152	98.58940109	95.22347367	93.41657445
25	83.56767087	104.10412860	86.27969492	98.59448307
26	83.57040917	89.10095605	73.94623788	94.61694679
27	83.58730469	84.10900774	79.46604780	91.21765509
28	84.72264614	95.34230977	77.92988116	98.77960840
29	84.74500583	88.75147695	88.27441301	98.04930626

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
30	84.77159839	101.80521620	109.82728330	100.21564920
31	84.86507526	100.46109410	85.99736541	90.39344387
32	84.87893066	93.00969019	100.95733090	101.23964270
33	84.94525209	84.79200085	77.60221833	79.55705668
34	84.97319664	84.47008182	74.30913576	77.31849979
35	85.14107321	88.08526307	86.49852470	100.87495390
36	85.35715903	93.89255077	92.47814736	99.09710263
37	85.42395204	97.14603079	111.95670150	99.14476475
38	85.58399797	61.36531379	83.59778225	80.56368267
39	85.68790523	104.41161140	96.13116554	82.67151588
40	85.94421176	107.67609320	90.08948528	115.69043970
41	86.00807052	91.16130465	89.52586135	96.22360862
42	86.19246171	101.56111030	90.12205395	81.46599284
43	86.29183430	104.33077590	116.05193650	96.28080322
44	86.59547389	98.67102372	96.63382388	102.73035710
45	86.74638061	80.50178087	86.75452679	95.12286375
46	86.74788405	89.51292206	77.20288466	78.85197973
47	86.85307131	88.50596012	80.45691477	107.10345390
48	86.90170040	104.19339880	91.89256147	108.72738130
49	86.97350143	87.31373677	92.37886417	77.45231399
50	86.98338816	86.05694499	106.84330190	96.14576950
51	86.99398122	99.02070679	83.02061289	101.90301840
52	87.04946523	84.66743855	97.67051247	85.89939941
53	87.20502862	107.03068420	94.51089208	106.07860170
54	87.27873740	102.32855990	93.14420975	104.58763260
55	87.34295950	87.35294431	88.49545946	80.33785003
56	87.38197222	114.44021130	111.10373270	100.07232690
57	87.46901208	103.13428520	86.32612006	93.77069725
58	87.70155537	96.50221690	90.61856675	95.20062293
59	87.70383039	83.38434424	87.50587564	102.11336790
60	87.70824861	83.32383213	80.30391710	92.21406778
61	87.84065346	87.32880078	93.07353444	87.44281177
62	87.89146125	89.55666150	99.78266863	92.89823419
63	88.00178250	110.05197120	92.48197784	104.42300220
64	88.22095251	87.19354039	99.76990468	96.25799004
65	88.32598268	103.06245070	104.18917180	85.69766770
66	88.38118730	88.85129569	91.44453880	100.44718860
67	88.64606711	102.69454210	98.83050928	108.97665880
68	88.65282058	87.41735321	96.14030063	94.24596667
69	88.65997055	106.06986310	100.04513180	112.67270290
70	88.73086380	102.24072190	94.62832469	96.47410077
71	88.76203504	94.38059925	91.09976269	104.09504950
72	89.12215321	111.03626030	100.82373950	106.86775600
73	89.22946036	109.45991630	103.06632250	110.35855090
74	89.26354463	92.02564394	88.94029239	96.24987454
75	89.28089900	98.35349617	105.06149660	88.75138668
76	89.29075383	101.91516380	109.86116230	104.48344870
77	89.30574321	97.46024372	90.56208305	96.42742319
78	89.43558834	66.98014953	100.36654690	93.37106726
79	89.52281573	105.15693560	100.86704160	99.38540044
80	89.56599297	100.57924800	96.55938475	91.09440316
81	89.59212503	81.03825220	82.48761391	95.20921885
82	89.62098255	86.78302067	105.94733620	93.14771569
83	89.62335857	98.34666938	87.80153735	94.72183413
84	90.12232956	100.63402230	100.77408530	91.62595737
85	90.16394638	80.06185035	100.38060400	92.04451714
86	90.24918463	102.39535490	89.11225662	106.88611120
87	90.51328530	82.32653450	96.05806626	96.37322655
88	90.68620231	102.17166560	95.22627351	114.16183290
89	90.74892645	88.99387415	99.06473801	103.52703110

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
90	90.76835607	98.12318477	79.99574976	84.33200087
91	90.81711063	100.91098760	103.10699150	98.95926633
92	91.05086051	92.77498210	93.75230143	103.37597740
93	91.24495462	96.92541882	106.66317270	91.61983774
94	91.26503408	85.90684522	86.43319354	93.72976383
95	91.31240612	94.15367647	86.73401503	94.61945078
96	91.31601083	102.10116200	93.23542031	99.08973780
97	91.33271161	87.16476634	81.79738921	90.07053482
98	91.39505145	85.61839533	102.06868980	83.87454636
99	91.42124523	114.28473760	100.89617540	100.38193500
100	91.59414342	90.52908887	90.98935932	104.86593290
101	91.62404704	88.41338761	95.13977982	90.72602946
102	91.64682163	95.00913772	105.19346200	94.33242849
103	91.69217433	105.46797250	96.77466785	87.96205703
104	91.85491604	93.60356300	89.83857632	107.17147110
105	91.87127630	97.47445403	103.27293380	106.74526430
106	91.93174750	85.94645836	97.94908509	102.31651020
107	91.99188545	92.81279223	84.15734118	89.22211160
108	91.99731242	96.75118560	102.68445200	99.34703331
109	92.02116825	94.78461618	81.08202397	90.77354595
110	92.10809806	90.49381160	84.03776155	86.24745971
111	92.12124054	99.50271747	99.82059243	98.24945620
112	92.21733910	96.33912773	86.88208758	93.52501810
113	92.40958491	79.36917392	88.55979531	97.12638060
114	92.66381166	105.02122830	93.84100162	105.37215270
115	92.69440451	91.96459053	99.90030528	98.11769403
116	92.73783851	103.42898610	108.82042070	108.52371900
117	92.94244448	77.44316747	91.66756301	77.79557602
118	92.95535926	96.68211564	86.30622804	88.93547164
119	92.97419037	78.22857302	109.24562070	95.20869759
120	93.03364135	100.28044910	100.06181160	108.86418050
121	93.05561703	106.05010390	103.73725050	96.50090205
122	93.15499635	81.96475552	97.53990798	92.07997502
123	93.17769492	83.82300842	84.62799104	72.25750842
124	93.23399868	84.32059473	87.04688345	97.01122226
125	93.26393727	100.36898580	96.97794098	90.26847910
126	93.46959260	78.40504262	90.80619826	96.25293456
127	93.47846291	97.01001095	104.68605190	98.06896367
128	93.52682144	106.46639380	111.25726470	105.60169210
129	93.53707455	107.61788610	110.08097750	105.91704330
130	93.55676258	92.76187375	88.61910112	102.71342290
131	93.60454867	94.20646770	95.50509479	94.84539498
132	93.60960070	99.56332213	108.89878140	96.95697247
133	93.67520449	100.10349170	108.97329560	113.84906030
134	93.69675311	89.09498407	94.17971873	88.01303199
135	93.72889003	98.90412416	98.26251843	91.78859958
136	93.77523234	90.93228385	100.39664820	82.27376011
137	93.95221814	80.60394624	95.59766966	89.26728095
138	94.03193088	106.59936140	109.10223090	91.61795879
139	94.05282769	89.62419443	79.79988637	97.43032719
140	94.14101584	97.26688386	105.88164490	90.94259727
141	94.17953573	108.66175830	109.59978300	92.31948498
142	94.21579058	106.63203590	102.36460360	88.07077979
143	94.24122405	83.99614492	88.40461555	92.95296712
144	94.37355402	93.05536947	79.77287452	102.14042070
145	94.37713757	101.82150850	95.71958921	93.81390965
146	94.47260856	100.91677270	97.86937514	89.55961889
147	94.52195163	109.81839000	102.61229800	98.98395943
148	94.67292265	94.80345985	96.95905403	93.59991652
149	94.68538654	99.32667975	107.26839270	100.49944350

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
150	94.75109646	96.16537870	106.44697390	87.49816521
151	94.90396585	91.13321477	98.98268446	96.55322648
152	94.90707741	105.03687080	93.28515655	91.57568253
153	95.00790425	82.45715985	90.83895542	87.61768605
154	95.01835600	90.58907859	92.71826892	100.30020660
155	95.09188339	104.72910010	79.96414153	81.96318907
156	95.09409302	91.03069698	97.95273601	94.74303409
157	95.10142064	111.85995310	105.04678790	111.12650770
158	95.13824456	90.60737877	114.73217670	103.75434490
159	95.18566898	96.42631573	92.54931289	95.36981286
160	95.19761246	115.06594150	111.04097370	110.07590540
161	95.20211222	109.98019780	107.41263960	97.42640112
162	95.21736582	108.76430660	102.42817520	112.40995620
163	95.22730049	109.76642250	107.99891070	101.61433600
164	95.29005072	95.22642006	94.76650709	97.88060963
165	95.39111722	101.40066360	98.24968276	105.06643170
166	95.42392456	85.30082741	98.48825954	99.29479765
167	95.46248746	100.41494090	104.04140610	104.27802230
168	95.67159468	101.96133780	98.71511171	108.32428900
169	95.71991815	97.52653675	108.23758050	101.91709770
170	95.76401069	105.80711660	99.28576326	97.24465044
171	95.87455836	93.04233194	93.17511719	113.46415720
172	95.89696453	96.93219024	102.30267710	84.80024810
173	95.91474220	105.64882860	108.89824500	97.72804927
174	95.93271228	113.64802310	89.83120742	122.80873410
175	95.98093195	112.58152870	105.81719070	97.05741663
176	95.98463478	88.81741872	94.24907509	97.13835370
177	96.01962152	95.47722643	94.78252235	105.39886100
178	96.03013188	96.55744487	98.99039262	104.64766780
179	96.05739965	104.03526440	109.20592030	92.80791066
180	96.08925012	95.91996056	99.50336209	88.04980959
181	96.20434699	109.66236990	104.46970310	108.49961110
182	96.22739908	103.85406910	107.18648860	107.29494340
183	96.29072636	95.80163084	103.78389890	91.59513795
184	96.35005558	103.37776990	95.20035448	102.02327230
185	96.37744287	104.36949730	102.81262560	100.60518820
186	96.47712099	108.44149860	109.39461820	83.46846630
187	96.59205874	85.95507024	88.82902840	99.19981778
188	96.61777351	93.52635260	93.17758877	92.46574219
189	96.67368472	93.93852499	97.76891089	99.86979179
190	96.71227876	99.44557498	101.70138100	92.81850113
191	96.74748533	91.27525246	105.56384230	96.23529707
192	96.76051427	93.09473341	91.96817533	94.24244793
193	96.77011703	105.18262870	103.73008500	91.82814307
194	96.80906208	102.20926020	96.58119720	111.39185960
195	97.21593021	91.14248789	85.92527795	91.12443816
196	97.31004376	89.44617876	96.30827987	96.81277539
197	97.39519652	101.37106590	106.60701460	95.48088423
198	97.42335935	98.38778929	79.87410028	98.68042566
199	97.45852340	93.13296907	94.23114787	96.47183367
200	97.46623503	89.81208790	104.63074370	94.80804719
201	97.54832423	110.33220740	97.90587444	103.59338240
202	97.59798751	89.35260238	93.31761790	98.37329252
203	97.61316281	108.10515010	106.65597140	85.36771421
204	97.64399741	106.01724620	109.23703680	97.99774997
205	97.68210009	95.00099616	97.84543873	99.79915911
206	97.77796560	107.08394280	112.70991280	99.17882805
207	97.78397938	112.71985720	97.18799388	96.96471570
208	97.81098715	98.84196103	97.16302500	84.34281404
209	97.91632984	88.78840061	83.70590856	82.83537715

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
210	97.92998003	95.54361394	83.55674127	93.75466051
211	98.01470548	105.11961680	103.73496030	103.96327870
212	98.15249260	109.93405410	97.07640111	99.49708786
213	98.26872747	103.77708430	105.85648220	102.74091730
214	98.32538311	82.95958711	105.10864300	82.72556208
215	98.36374114	93.81159857	83.98470927	102.52908310
216	98.45225955	99.48832453	102.28208220	85.15305818
217	98.53486514	97.10432677	86.25152697	79.97721933
218	98.53887536	86.48039210	85.17486128	82.91379410
219	98.58531687	103.80106410	102.83913840	99.28536587
220	98.61523008	78.23354843	89.98098539	96.47841128
221	98.62824368	105.02863120	100.88861850	96.24757992
222	98.67065184	99.07198852	106.33546170	112.85196030
223	98.71483140	83.10770181	86.78401296	86.81926670
224	98.75501492	92.67233264	104.70426800	101.12432380
225	98.77361663	100.46529620	108.70931050	97.36317727
226	98.77390008	75.37898931	90.61016147	91.97691009
227	98.78743020	86.72705175	83.91688476	105.27641720
228	98.78964346	98.72272781	107.49910130	103.49936650
229	98.84474426	98.96213642	108.63772160	120.68020330
230	98.89432481	92.67461440	105.59186590	82.82162207
231	98.92182845	99.34356911	88.34606712	84.23996662
232	98.98574814	98.66185414	82.83040008	93.00683631
233	99.06276310	99.69677311	96.89356876	98.24262305
234	99.09277235	103.20153800	110.13063470	113.38545690
235	99.10977964	93.05124380	96.93086680	91.70074521
236	99.15771584	106.88067440	100.28088060	92.22450685
237	99.15890135	107.95567080	95.97538221	97.13015875
238	99.19740221	93.87782759	95.34449387	85.03335103
239	99.19803181	93.78081929	110.28619870	98.37272744
240	99.30593238	89.56310366	101.95904580	90.44021616
241	99.49763963	104.53326620	116.35153470	108.49956420
242	99.52360095	104.86842250	109.58521330	97.11571339
243	99.56380770	95.69321630	122.71841530	88.15274074
244	99.59245845	83.88982198	88.73635237	98.87491663
245	99.59471034	109.52378200	104.48931290	116.20334060
246	99.61615575	98.09078794	111.33755960	94.45151920
247	99.77218367	92.34483346	98.30651236	98.01310903
248	99.78794680	102.04482800	95.49835833	90.82984272
249	99.80066556	103.71424200	113.32119250	95.42128182
250	99.85374293	94.63511625	89.82562485	104.87188600
251	99.91913398	96.31773691	98.38983321	87.45688122
252	99.99690058	82.12406666	101.01978460	99.39926792
253	100.00726430	103.32698790	109.91850640	91.69627545
254	100.00951310	113.27040450	106.16029210	103.24271370
255	100.04037780	94.97539431	92.11781735	85.28223121
256	100.05860660	92.15863926	103.14964450	99.15558464
257	100.06383650	113.95172900	117.05107420	99.87082845
258	100.10292080	92.59870711	107.83942010	112.77830700
259	100.14889600	92.46554556	101.76511340	97.71898919
260	100.18801610	99.62505870	90.93492057	95.37959837
261	100.19891950	105.41356050	98.82541121	104.28557530
262	100.21250500	98.56396393	98.61121740	88.63084676
263	100.22564420	103.10238020	97.95373575	101.30247290
264	100.23829780	109.67445050	96.66891651	104.33285650
265	100.24913680	96.51755683	108.12125200	107.62542150
266	100.26207680	105.68073210	108.68310070	108.19945330
267	100.31714340	106.70956390	105.16763940	126.28813550
268	100.41925870	99.34035114	95.60202333	82.26612344
269	100.44354450	96.27675534	96.67277528	102.56423130

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
270	100.49280570	92.08206633	91.60514777	77.37059098
271	100.49734300	98.77011865	100.09390730	90.16401902
272	100.53780390	84.58286618	99.65876903	101.54585510
273	100.79296800	91.30456393	88.01730397	95.88244983
274	100.80371160	103.82645530	90.12333024	91.59794663
275	100.86862520	98.38790788	100.88260520	98.63907509
276	100.87540960	95.43047267	98.14133544	113.99608060
277	100.90315480	80.63583458	84.75966226	89.68164702
278	100.91784450	94.61037660	107.00181050	87.53672000
279	100.91882360	92.42434457	104.16867030	96.63772185
280	100.96469260	104.34019130	96.17671162	114.94394340
281	100.97134870	91.89996310	86.64063772	108.86911850
282	100.97617670	93.59508488	84.63479972	94.95212515
283	101.08857650	112.39235470	109.78715520	96.16677518
284	101.16880920	99.64495736	92.91848742	98.99466884
285	101.24518530	85.24276252	99.60843479	97.24903149
286	101.31810600	96.75035925	101.35312330	94.12273175
287	101.31938200	110.51028820	104.40500150	105.06122950
288	101.33655340	106.98058150	99.08346356	94.96155430
289	101.34778830	101.17106920	103.88034090	84.51098658
290	101.36771710	107.08497370	102.59826040	105.64339360
291	101.37605850	114.96155740	108.11679330	100.01623510
292	101.39534420	103.95409130	99.72636607	100.09930570
293	101.41408160	97.24048285	91.92826732	102.55075320
294	101.41904070	115.91702370	112.15430010	103.44943220
295	101.70930730	100.07163170	112.41198420	103.33108750
296	101.80025240	105.35418430	82.50311576	93.85326218
297	101.81179310	90.38571279	115.86098570	103.18106110
298	101.86862460	106.01616350	111.43718060	96.45390415
299	101.89007180	100.55535290	99.11600165	105.74684750
300	101.91117390	91.81792853	90.54406116	99.85638413
301	101.95739750	96.83925794	109.99061060	106.09245490
302	101.98808710	115.69642930	106.51127470	94.44491675
303	102.07180740	103.51327090	102.64424520	109.98459640
304	102.10859300	86.80553067	106.03545930	96.38710639
305	102.13558530	96.54617000	104.07966530	89.62209844
306	102.23697400	102.36466420	106.51944710	91.76744228
307	102.30508940	90.47217568	103.02534010	88.28194196
308	102.31172270	113.94876200	105.49463830	106.37782860
309	102.37492330	106.95497550	104.72532770	122.86648970
310	102.37765620	103.10205200	99.31464215	106.34361960
311	102.50249530	116.60014010	106.21389290	93.62681046
312	102.51071670	96.02776679	103.20322460	103.33399420
313	102.52316100	107.94317980	106.20062250	99.99714029
314	102.65431540	100.87621410	105.71835460	109.61011780
315	102.69734910	97.96914526	116.04194660	101.31031860
316	102.72124370	89.88394060	95.61281714	100.28858980
317	102.72995750	93.91775126	103.73341370	100.64072570
318	102.75932150	100.61159230	96.76189539	99.48190083
319	102.78609330	85.56781575	91.44247712	95.98347619
320	102.88619120	110.76025420	115.54512310	97.58113684
321	102.89605690	95.54970914	93.12632391	87.95361528
322	102.90572950	103.41079060	99.91620827	109.01612990
323	102.91512690	103.76562600	85.74106916	111.65055650
324	102.94842150	92.43598238	88.13106050	102.45126410
325	103.00475230	109.36549360	98.88805250	103.82365730
326	103.10936980	92.84098316	86.14311049	84.68961664
327	103.21564740	99.56442909	105.04104850	103.97320350
328	103.22822460	98.93367591	100.39274130	110.14449760
329	103.26899800	106.11181110	98.88420152	96.39534823

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
330	103.31659400	100.99847010	94.48380261	104.70965310
331	103.36050480	102.30274330	100.70109780	93.43308669
332	103.47377800	107.29341800	104.22491470	109.85270340
333	103.48109010	95.89894892	101.04421350	85.65728216
334	103.50817750	93.37288044	85.28183809	102.08359450
335	103.51437080	111.95237220	106.44717450	113.50936320
336	103.53844810	105.15557050	94.34878053	98.98158570
337	103.54068400	93.27164673	97.93948255	106.32025330
338	103.54971860	96.38455765	88.64410482	103.32169170
339	103.62953610	104.76373240	100.29569080	87.77531585
340	103.73025020	105.16551170	109.48315590	94.82682585
341	103.99550860	106.05891240	104.48230350	104.33862610
342	104.07991550	102.31990280	109.48024280	107.37853990
343	104.10227540	103.33564910	105.04954900	103.24362360
344	104.13638390	93.55182178	104.91829430	106.73237930
345	104.15976470	102.50887020	80.75393995	90.95222119
346	104.17266070	111.22610060	98.77345417	97.67180147
347	104.19684870	114.69114730	105.15789690	102.65057310
348	104.34306050	84.01033736	97.37340821	94.24009031
349	104.35834860	98.59587673	122.12575650	89.99057433
350	104.38431510	96.87666319	94.61123172	93.87791319
351	104.49616890	106.10414820	90.67806944	98.12742215
352	104.50635230	106.59899880	128.94429200	115.48407810
353	104.58110150	100.48398950	91.53487654	89.86079726
354	104.65260890	96.63615717	104.74084330	107.47537640
355	104.65683410	92.21520823	100.10987630	98.19467561
356	104.76588720	103.60647940	97.58674231	99.98685677
357	104.83479870	110.49621240	109.38953930	112.82531500
358	105.11388420	108.68864010	82.48838172	95.17117138
359	105.12030890	105.60775510	98.87161379	104.30674390
360	105.29153750	102.11470900	101.87135630	105.88843950
361	105.30663120	93.14946657	97.08758571	97.55325594
362	105.31126770	102.23940380	102.23161670	93.51774750
363	105.33353120	95.87699073	94.38895340	104.92567090
364	105.42239330	111.62086500	112.39160220	95.74562396
365	105.42729650	96.95309678	121.74491060	87.25632703
366	105.48392910	96.15044694	95.80241210	89.44671529
367	105.56562670	94.61715262	95.86694769	111.21373970
368	105.68678190	101.86363590	89.98133281	88.40368655
369	105.70450850	105.18982640	90.53587844	98.63408197
370	105.71008310	106.00802690	113.16454150	112.86452880
371	105.71448370	115.96694590	106.16989540	101.56884880
372	105.73722300	113.84336300	95.90275879	106.76229400
373	105.76676400	97.03919625	90.83703327	105.62559850
374	105.78217530	111.50183920	106.38782990	102.10842210
375	105.81708990	101.65743040	108.40249040	115.41044510
376	105.98203150	95.39224876	94.06270025	89.58798358
377	106.24173180	110.42071870	105.49499550	109.27928310
378	106.37017170	106.85874210	116.92536550	117.05614250
379	106.48541790	109.75894670	102.80593140	110.08579730
380	106.49508680	94.91729551	93.07829265	86.71221111
381	106.53669700	120.37633740	125.60606340	110.43960140
382	106.60228340	109.25750990	116.82137090	101.42704310
383	106.69104670	114.14693840	109.14605530	108.05400560
384	106.76688690	97.05425386	104.01194350	107.44581890
385	106.85517890	101.76139530	105.44839660	111.52859330
386	107.07618910	105.32173730	87.67650818	97.48561119
387	107.17563160	118.24778890	111.73910310	95.55145698
388	107.18226620	104.11948230	118.42353030	112.09304240
389	107.24275090	100.74827320	90.07311487	101.73869410

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
390	107.37145070	89.89851691	93.68713543	100.62451740
391	107.40348650	102.41831620	108.08682320	102.28929390
392	107.51649680	107.01711890	104.55790900	112.13348340
393	107.57588200	102.96124010	99.66011827	109.50417860
394	107.58889660	99.72573760	97.04070694	90.75026028
395	107.61404770	117.67622830	115.36086850	121.96742440
396	107.63316560	105.58266460	90.48968753	95.19400643
397	107.63693600	115.26535540	111.35876160	106.43760840
398	107.67931140	116.01639510	109.50481740	112.50582940
399	107.87816500	104.41602200	103.66453420	137.93090150
400	108.10542600	113.46405580	106.96444980	111.03614480
401	108.27751770	109.80962460	106.16105880	102.85393880
402	108.28027800	93.16713701	92.54350175	107.87845460
403	108.33229810	110.64922240	93.02401095	106.78991840
404	108.37115070	105.41076910	117.73191310	110.02737740
405	108.42623530	115.21538800	107.91186260	115.45482800
406	108.44233990	95.92055548	105.12118950	102.17719190
407	108.49198140	100.71602700	108.10642790	103.98619700
408	108.57908930	114.99899360	103.97199780	96.94716796
409	108.60637230	115.83676310	87.62521130	94.40042016
410	108.63766990	97.44494534	101.58256330	93.68291583
411	108.65458960	109.68277930	101.58606390	106.51059360
412	108.71349410	109.66784100	103.98752750	95.92077154
413	108.72392010	101.11830860	94.75477267	94.03234511
414	108.73468370	94.09021140	106.51717090	109.13436200
415	108.76766690	97.08514972	105.16457570	103.88320000
416	108.83263040	102.57453760	108.16978260	100.77395010
417	108.85237100	101.64198770	97.73344190	89.58284115
418	108.89548580	104.39350080	114.27627170	113.73674890
419	108.89854110	108.72240860	107.00528100	117.97229080
420	108.91916560	108.85301070	112.71987840	89.81385313
421	109.03191560	97.83029312	94.80043325	94.45860445
422	109.05415980	97.49127261	105.76961050	101.83277840
423	109.12261780	95.44924915	90.30612924	106.02767820
424	109.28449080	119.66339270	117.15620470	117.10044800
425	109.41476380	99.41171317	96.66845359	99.94314993
426	109.53410020	87.27715981	99.86523829	108.33666510
427	109.60670200	100.62996950	97.64693232	105.34880190
428	109.63004700	109.45968390	114.31616410	116.26354590
429	109.70395820	88.34868259	108.16286550	101.24515140
430	109.71282210	100.90149580	108.80259850	100.15497690
431	109.88987830	103.98689000	107.23448230	108.02392990
432	109.90216230	99.10317796	103.42226760	96.79727347
433	109.93465540	102.85690610	108.88277310	106.21791430
434	110.00027370	85.91286389	107.89656250	105.84360320
435	110.01990920	102.04505230	98.05880854	97.05429192
436	110.29978550	104.21191930	101.56612830	109.81580680
437	110.35219430	106.22301750	105.38555190	103.18629270
438	110.36383310	95.28483953	106.18279180	113.39686680
439	110.67293740	102.61036140	125.79446710	107.26548780
440	110.71368040	99.50286576	101.32739880	106.69707170
441	110.79784340	106.02350620	115.83147830	116.48154360
442	110.96839400	106.21238780	107.14699120	109.11631750
443	111.03830610	86.48583254	92.36112826	102.38242110
444	111.07487420	92.81045718	97.54437620	81.68066713
445	111.08701890	108.86909260	114.57308730	110.15052410
446	111.41754820	106.50111380	96.56464592	94.27778763
447	111.48409660	98.49124050	105.67637870	97.31420932
448	111.94069210	106.92821960	97.17268021	99.41010677
449	112.00757130	95.06523710	132.03743630	101.62378400

Table F1.: The table shows the objective vector values for Test Problem II. The rows of the 8 Pareto systems appear in gray.

System	Objective g_1	Objective g_2	Objective g_3	Objective g_4
450	112.11073550	100.56362180	110.56286470	102.11357090
451	112.46477630	104.69738100	116.53127150	108.85416130
452	112.54094950	110.16398190	109.52150880	114.03996250
453	112.55347230	111.29325540	97.08776244	100.86773440
454	112.62137290	99.11996874	101.28107280	115.40253930
455	112.68103260	105.12918970	112.09191930	107.51945780
456	112.91897250	102.32272210	119.86293530	97.25717833
457	113.08102290	114.99585740	115.68260260	122.07904890
458	113.10876460	101.36670000	91.94907064	84.01508753
459	113.28910460	96.14598940	120.34881370	96.14539409
460	113.37856950	107.10846020	104.02347500	106.53098160
461	113.47324140	109.97576980	107.07141790	115.47900340
462	113.60598920	116.01152170	113.67710320	96.80772637
463	113.63378560	106.77021990	122.12728510	112.31651110
464	113.92509200	114.74053150	96.52412324	103.87384890
465	114.13876500	110.45996630	108.58074550	125.91277700
466	114.68100360	91.68179880	106.46388990	95.58376019
467	114.69277170	106.65931850	89.98474932	91.27427495
468	114.75000030	112.52868720	116.90379420	98.69440067
469	114.79010300	111.75676600	121.36631270	118.00230550
470	114.80723310	105.41724770	113.40842310	97.39756121
471	115.00231820	113.78564160	120.26953580	115.93489110
472	115.02803640	110.70451480	110.58587350	117.65704880
473	115.07875150	99.80757249	97.26835297	110.90578060
474	115.82706800	90.66069533	97.00944661	108.55782160
475	116.04075890	110.47616990	101.19285210	117.56535990
476	116.16136510	112.39966060	116.82615290	103.86810510
477	116.37872800	117.26880000	118.21771080	114.74525170
478	117.06010140	99.24290390	109.61799670	111.42092870
479	117.22378730	108.27770490	111.43966170	117.35841390
480	117.48199070	102.94464050	115.37047800	100.63564950
481	117.75569500	94.05283378	88.29035908	105.23290040
482	117.83068540	109.41483500	122.17348100	121.11260360
483	118.15950000	107.71353930	115.34086620	130.30036780
484	118.45905940	117.77083410	114.07557090	145.27256450
485	118.48276530	128.53460090	124.03044450	119.77882440
486	120.39957880	113.82059230	110.61472890	109.35503010
487	120.56032070	109.86477010	108.31839410	100.38149800
488	120.87460360	97.51505937	111.27018610	90.75551250
489	121.48005870	109.35311180	102.01904920	101.18460260
490	121.64208890	102.98230390	100.02222510	114.31256330
491	121.72578450	103.69808650	104.07907840	101.88522980
492	121.92132790	124.83903360	125.86399940	119.10370230
493	122.07635830	105.81319220	113.54472790	115.58675940
494	122.88561340	106.25197650	105.00070440	108.64766360
495	123.35679190	100.23151100	112.13483840	113.85738950
496	123.59857650	110.74389170	113.84651370	99.60335761
497	123.96085560	116.72523270	120.55149340	112.12745040
498	124.51604800	110.43812880	120.50760240	117.37860200
499	130.73530300	103.53059590	111.53474340	106.87085960
500	136.12463360	127.03720230	124.06689130	110.49094970