Bi-objective Simulation Optimization on Integer Lattices using the Epsilon-Constraint Method in a Retrospective Approximation Framework

Kyle Cooper
School of Industrial Engineering, Purdue University and Tata Consultancy Services, coopel49@purdue.edu,

Susan R. Hunter
School of Industrial Engineering, Purdue University, susanhunter@purdue.edu,

Kalyani Nagaraj
School of Industrial Engineering & Management, Oklahoma State University, kalyani.nagaraj@okstate.edu,

We propose the Retrospective Partitioned Epsilon-constraint with Relaxed Local Enumeration (R-PεRLE) algorithm to solve the bi-objective simulation optimization problem on integer lattices. In this nonlinear optimization problem, both objectives can only be observed with stochastic error, the decision variables are integer-valued, and a local solution is called a local efficient set. R-PεRLE uses a retrospective approximation (RA) framework to repeatedly call the PεRLE sample-path solver at a sequence of increasing sample sizes, using the solution from the previous RA iteration as a warm start for the current RA iteration. The PεRLE algorithm comprises a call to each of the Pε and RLE algorithms, in sequence. First, Pε searches for new points to add to the sample-path local efficient set by solving multiple constrained single-objective optimization problems. Pε places constraints that locate new sample-path local efficient points that are a function of the standard error away, in the objective space, from those already obtained. Then, the set of sample-path local efficient points found by Pε is sent to RLE, which is a local crawling algorithm that ensures the set is a sample-path approximate local efficient set. As the number of RA iterations increases, R-PεRLE provably converges to a local efficient set with probability one under appropriate regularity conditions. We also propose a naïve, provably-convergent benchmark algorithm for problems with two or more objectives, called R-MinRLE. R-MinRLE is identical to R-PεRLE except that it replaces the Pε algorithm with an algorithm that updates one local minimum on each objective before invoking RLE. R-PεRLE performs favorably relative to R-MinRLE and the current state of the art, MO-COMPASS, in our numerical experiments.

Key words: multi-objective simulation optimization, epsilon-constraint method

History: Authors’ preprint compiled Monday 4th June, 2018.

1. Introduction

We consider the context of multi-objective simulation optimization (MOSO) on integer lattices, that is, nonlinear optimization in which two or more simultaneous objectives can only be observed with error, and each decision variable can only take on integer values. The solution to a MOSO problem is the set of feasible decision points for which no other feasible
decision point is at least as good on all objectives and strictly better on at least one objective. We refer to this set, and the decision points therein, as efficient; the image of this set and the points therein are Pareto. MOSO problems that involve characterizing the efficient or Pareto set frequently arise when Monte Carlo simulation models are used to design stochastic systems, there are multiple conflicting objectives, and the decision variables are natural numbers. For example, as noted by Hunter et al. (2018), applications arise in a variety of fields including aviation (Li et al. 2015b), healthcare (Chen and Wang 2016), environment and sustainability (Singh and Minsker 2008), logistics and supply chain management (Chew et al. 2009), and manufacturing (Andersson et al. 2007); decision variables include quantities such as the number of people to employ or the amount of stock to order.

When tasked with solving a MOSO problem, decision-makers may select from a variety of solution approaches. For example, the decision-makers may create a utility function and re-formulate the MOSO problem as a single-objective simulation optimization (SO) problem. Then, they may use existing single-objective SO methods to find a solution (see, e.g., Pasupathy and Ghosh 2013, Fu 2015, for an overview). Alternatively, the decision-makers may wish to explore the structure of the MOSO problem by identifying an entire efficient set, from which they may later select a single efficient point for implementation. Our interest lies in solution methods that identify an entire local efficient set (defined in §2.1.2).

Few provably convergent algorithms exist to identify an entire local efficient set in the context of MOSO on integer lattices (Hunter et al. 2018). When the feasible set is finite, “small,” and the decision variables may be categorical, methods that locate the global efficient set are called multi-objective ranking and selection methods. These methods include MOCBA (Lee et al. 2010), SCORE (Feldman and Hunter 2018), and M-MOBA (Branke and Zhang 2015, Branke et al. 2016). When the feasible set is not necessarily finite and a neighborhood structure can be defined on the feasible set, as is the case with an integer lattice, existing methods and algorithms include (a) Adaptive Pareto Search (Gutjahr 2009), which is a framework based on sample average approximation (SAA, see, e.g., Shapiro et al. 2009); (b) Multi-Objective Probabilistic Branch and Bound (MOPBnB, Huang and Zabinsky 2014), which is a globally convergent method, and (c) MO-COMPASS (Li et al. 2015a), which is the current state-of-the-art algorithm for identifying a local efficient set as the solution to a MOSO problem on an integer lattice. Under appropriate regularity conditions, MO-COMPASS provably converges to a local efficient set with probability one.
We propose a new efficient and provably-convergent algorithm called Retrospective Partitioned Epsilon-constraint with Relaxed Local Enumeration (R-PεRLE, written as R-PERLE when special characters are not allowed) to solve bi-objective SO problems on integer lattices. Our algorithm is a competitor to MO-COMPASS for MOSO with exactly two objectives. R-PεRLE employs three key concepts: a version of SAA called retrospective approximation (RA) for overall algorithmic efficiency (see, e.g., Pasupathy and Ghosh 2013), the ε-constraint method (see, e.g., Miettinen 1999) which enables us to find sample-path local efficient points using a pseudo-gradient-based single-objective solver, and relaxed local enumeration (RLE) to certify the solution returned in each RA iteration is, in some sense, sample-path optimal.

To explore each of these key concepts in turn, first, consider RA. RA is an algorithmic framework that requires solving a sequence of sample-path problems (formulated in §3.1) at increasing sample sizes. One RA iteration comprises solving a sample-path problem at one sample size. To ensure sampling efficiency, the solution from the previous RA iteration is used as a warm start in the next RA iteration, which has a higher sample size. Thus large sample sizes are not wasted on suboptimal points in early RA iterations. Instead, they are saved for later RA iterations in which the warm start likely is close to the true solution.

Within each RA iteration, we propose the (deterministic) PεRLE algorithm to solve the sample-path bi-objective problem on an integer lattice. PεRLE comprises two sub-algorithms with the primary goals of pseudogradient-based search and certification, respectively: Pε (‘P epsilon’) and RLE. First, the Pε algorithm partitions the objective space and solves a set of sample-path ε-constraint problems (defined in §5) at carefully-chosen constraint values, denoted by ε. This technique, called the ε-constraint method, is a standard scalarization method for solving multi-objective optimization problems (Miettinen 1999); we choose the values of ε to make infeasible all regions of the decision space that map to objective vectors closer than a function of the standard error away from known sample-path local Pareto points. We use the single-objective, pseudo-gradient-based SPLINE algorithm (Wang et al. 2013, listed in §5.3) to solve each ε-constraint problem. Then, the set of local efficient points found by Pε is sent to the RLE algorithm, which enumerates the neighborhood of the set and crawls to new local efficient points as required, up until it can certify that the set is a sample-path approximate local efficient set (defined in §3.2). For sampling efficiency in our RA framework, “completeness” of the local efficient set in RLE also depends on the standard errors of the objective values of the points in the set.
1.1. Contributions

We view the specific contributions of this work as follows:

1. R-P\(\varepsilon\)-RLE can solve a wide class of bi-objective SO problems having integer-valued decision variables and deterministic constraints. The importance of developing algorithms for this class of problems is demonstrated by the abundance of so-called integer-ordered problems on the simopt.org website (Henderson and Pasupathy 2018), and by the wide variety of application areas in which bi-objective SO problems arise (Hunter et al. 2018).

2. R-P\(\varepsilon\)-RLE, which shows promising numerical performance, adapts the \(\varepsilon\)-constraint method for use with bi-objective SO problems in an algorithmically efficient way. First, since an RA framework prescribes obtaining the same number of simulation replications at every point within an RA iteration, the required simulation replications can be obtained in parallel with common random numbers (CRN, see, e.g., Law 2015). Solving sample-path \(\varepsilon\)-constraint problems inside P\(\varepsilon\) can also be completed in parallel, and in the limit, the number of \(\varepsilon\)-constraint problems solved in each RA iteration corresponds to the cardinality of the local efficient set, which we assume is finite. Further, the P\(\varepsilon\) algorithm employs the pseudo-gradient-based SPLINE algorithm to quickly locate local efficient points as the solution to each \(\varepsilon\)-constraint problem. Finally, both P\(\varepsilon\) and RLE employ relaxations to ensure we only solve each sample-path problem to an error tolerance commensurate with our sampling error. The relaxations inside P\(\varepsilon\) ensure that at the end of each RA iteration, we return an “even” approximation of the local Pareto set, where the granularity is a function of the standard errors of the sample-path local Pareto points found.

3. While we propose R-P\(\varepsilon\)-RLE for mainstream use, we also propose and discuss the convergence properties of two other RA algorithms, R-P\(\varepsilon\) and R-MinRLE.

   (a) The R-P\(\varepsilon\) algorithm is identical to R-P\(\varepsilon\)-RLE, except that RLE is never invoked. To show that P\(\varepsilon\) usually provides good starting points to RLE, we demonstrate that R-P\(\varepsilon\) converges under regularity conditions that are more restrictive than those required for the convergence of R-P\(\varepsilon\)-RLE.

   (b) The R-MinRLE algorithm is identical to R-P\(\varepsilon\)-RLE, except that the P\(\varepsilon\) algorithm is replaced by the MIN algorithm. The MIN algorithm brings up the sample sizes of all estimated efficient points from the last RA iteration, updates the sample-path local minimizers on each objective, and removes sample-path dominated points.
Then, it invokes RLE. The R-MINRLE algorithm can solve problems with two or more objectives. We view R-MINRLE as a somewhat naïve, provably convergent, pseudogradient-based benchmark algorithm for MOSO problems on integer lattices with two or more objectives. Since the proofs of convergence for R-PE RLE and R-MINRLE rely only on having invoked RLE, PE can be considered an “accelerator” for RLE in two objectives that out-performs the naïve accelerator, the MIN algorithm. In the future, other accelerators can be developed for RLE, where the convergence guarantee is provided by this paper.

1.2. Notation and Terminology

With few exceptions, constants are denoted by lower-case letters \((a)\), random variables by capital letters \((X)\), sets by script capital letters \((\mathcal{A})\), vectors by bold \((\mathbf{x})\), random vectors by capital bold \((\mathbf{X})\), families of sets by Fraktur \((\mathcal{A} \in \mathfrak{A})\), and operators by blackboard bold \((\mathbb{E}[X])\). The set of all \(q\)-dimensional integer-valued vectors is \(\mathbb{Z}^q \subset \mathbb{R}^q\). The set of all \(d\)-dimensional extended real-valued vectors is \(\mathbb{R}^d\). The \(d\)-dimensional vector \((0, \ldots, 0)\) is denoted \(\mathbf{0}_d\). The complement of the set \(A\) is \(A^c\). The sum of two sets \(A \subseteq \mathbb{R}^d\) and \(B \subseteq \mathbb{R}^d\) is the Minkowski sum, \(A + B := \{a + b : a \in A, b \in B\}\). For a sequence of events \(\{A_n\}\) defined in a probability space, \(A_n i.o.\) if infinitely many of \(A_n\) occur, where \(A_n i.o. = \limsup_{n} A_n = \cap_{i=1}^{\infty} \cup_{j=i}^{\infty} A_j\). Finally, let \(A \subset \mathbb{R}^q\) and \(B \subset \mathbb{R}^q\) be nonempty, bounded sets. Then (a) \(d(x, x') = ||x - x'||\) is the Euclidean distance between two points \(x, x' \in \mathbb{R}^d\); (b) \(d(x, B) = \inf_{x' \in B} ||x - x'||\) is the distance from the point \(x \in \mathbb{R}^d\) to the set \(B\); (c) \(d(A, B) = \sup_{x \in A} d(x, B)\) is the Hausdorff distance between sets \(A\) and \(B\). Also, to help the reader, we provide acronyms in Table 1.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Defined in</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO / MOSO</td>
<td>Simulation Optimization / Multi-Objective Simulation Optimization</td>
<td>§1</td>
</tr>
<tr>
<td>SAA / RA</td>
<td>Sample Average Approximation / Retrospective Approximation</td>
<td>§1, §3</td>
</tr>
<tr>
<td>CRN</td>
<td>Common Random Numbers</td>
<td>§1.1</td>
</tr>
<tr>
<td>PE</td>
<td>Partitioned Epsilon (algorithm)</td>
<td>§1, §5</td>
</tr>
<tr>
<td>RLE</td>
<td>Relaxed Local Enumeration (algorithm)</td>
<td>§1, §6</td>
</tr>
<tr>
<td>LWEPE / LEP</td>
<td>Local Weakly Efficient Point / Local Efficient Point</td>
<td>§2.1.1</td>
</tr>
<tr>
<td>GWEPE / GEP</td>
<td>Global Weakly Efficient Point / Global Efficient Point</td>
<td>§2.1.1</td>
</tr>
<tr>
<td>LWES / LES</td>
<td>Local Weakly Efficient Set / Local Efficient Set</td>
<td>§2.1.2</td>
</tr>
<tr>
<td>GWES / GES</td>
<td>Global Weakly Efficient Set / Global Efficient Set</td>
<td>§2.1.2</td>
</tr>
<tr>
<td>LPS</td>
<td>Local Pareto Set</td>
<td>§2.1.2</td>
</tr>
<tr>
<td>ALES</td>
<td>sample-path Approximate (\mathcal{N}_a)-Local Efficient Set</td>
<td>§3.2</td>
</tr>
<tr>
<td>NCN</td>
<td>sample-path Non-Conforming Neighborhood</td>
<td>§6.1</td>
</tr>
</tbody>
</table>
2. Problem Context: Preliminaries for MOSO on Integer Lattices

We formulate the MOSO problem on integer lattices with \( d \) simultaneous objectives as

\[
\text{Problem } M_d: \quad \text{minimize}_{x \in \mathcal{X}} \{ g(x) = (g_1(x), \ldots, g_d(x)) := (E[G_1(x, \xi)], \ldots, E[G_d(x, \xi))] \},
\]

where \( g: \mathcal{X} \to \mathbb{R}^d \) is an unknown vector-valued function and the nonempty feasible set \( \mathcal{X} \subseteq \mathbb{Z}^q \) is a subset of an integer lattice. Any deterministic constraints present in Problem \( M_d \) are defined through the feasible set \( \mathcal{X} \). In what follows, we define optimality concepts for Problem \( M_d \) and provide a formal problem statement. Although the focus of R-P\( \varepsilon \)RLE is the bi-objective case of \( d = 2 \), we retain the generality of \( d \) objectives whenever possible because our benchmark algorithm, R-MiNRLE, is defined for \( d \geq 2 \) objectives.

2.1. Optimality Concepts

Since it may not be clear what we mean by a local solution in the context of MOSO, in this section, we formulate optimality concepts for Problem \( M_d \). Our presentation of these concepts follows Hunter et al. (2018), Li et al. (2015a), Wang et al. (2013).

Since our focus is on local optimality, we define a flexible neighborhood structure on which our local optimality holds. For \( x \in \mathcal{D} \subseteq \mathbb{Z}^q \) and neighborhood size parameter \( a \in \mathbb{R}, \ a \geq 1, \) define the \( \mathcal{N}_a \)-neighborhood of a point \( x \) as \( \mathcal{N}_a(x) := \{ x' \in \mathbb{Z}^q : d(x, x') \leq a \} \). Further define the \( \mathcal{N}_a \)-neighborhood of a set as the union of the \( \mathcal{N}_a \)-neighborhoods of all the points belonging to the set. That is, for \( S \subset D \subseteq \mathbb{Z}^q \), the \( \mathcal{N}_a \)-neighborhood of set \( S \) is \( \mathcal{N}_a(S) := \bigcup_{x \in S} \mathcal{N}_a(x) \).

Then for any set \( A \), define \( \mathcal{N}_a^c(A) := \mathcal{N}_a(A) \setminus A \) as the deleted neighborhood of \( A \).

In addition to the neighborhood definitions, we also require a way to compare vectors in the objective function space. Thus we define the following notions of dominance.

**Definition 1.** Let \( x_1, x_2 \in \mathcal{X} \) and \( d \geq 2 \). We say the vector \( g(x_1) \)

1. weakly dominates \( g(x_2) \), written as \( g(x_1) \preceq g(x_2) \), if \( g_k(x_1) \leq g_k(x_2) \) for all \( k = 1, \ldots, d; \)
2. dominates \( g(x_2) \), written as \( g(x_1) \prec g(x_2) \), if \( g(x_1) \preceq g(x_2) \) and \( g(x_1) \neq g(x_2); \)
3. strictly dominates \( g(x_2) \), written as \( g(x_1) < g(x_2) \), if \( g_k(x_1) < g_k(x_2) \) for all \( k = 1, \ldots, d. \)

We also define that \( g(x_2) \) is weakly dominated by \( g(x_1) \), written as \( g(x_2) \succeq g(x_1) \), if \( g_k(x_1) \geq g_k(x_2) \) for all \( k \in \{1, \ldots, d\}. \) Using analogous definitions, denote that \( g(x_2) \) is dominated or strictly dominated by \( g(x_1) \) as \( g(x_2) \succeq g(x_1) \) or \( g(x_2) > g(x_1) \), respectively. Using these definitions, we define concepts related to optimal points and optimal sets, which are illustrated in Figure 1 for an \( \mathcal{N}_1 \)-neighborhood structure.
2.1.1. Minimizers and Efficient Points Following Wang et al. (2013), for each objective $k \in \{1, \ldots, d\}$, we define local minimizers of the $k$th objective function as follows.

**Definition 2 (Wang et al. 2013).** Given an objective function $g_k : \mathcal{X} \rightarrow \mathbb{R}$, a point $x_k^{\text{min}} \in \mathcal{X}$ is an $N_a$-local minimizer of $g_k$ if $g_k(x_k^{\text{min}}) \leq g_k(x)$ for all $x \in N_a(x_k^{\text{min}}) \cap \mathcal{X}$.

We further define local efficient points and local efficient points as follows.

**Definition 3.** A point $x^w \in \mathcal{X}$ is an $N_a$-local weakly efficient point (LWEP) if
1. $\not\exists x \in N_a(x^w) \cap \mathcal{X}$ such that $g(x) < g(x^w)$; or equivalently, if
2. $\forall x \in N_a(x^w) \cap \mathcal{X}$, $g(x) \not\leq g(x^w)$, that is, $\exists k \in \{1, \ldots, d\}$ such that $g_k(x^w) \leq g_k(x)$.

**Definition 4.** A point $x^* \in \mathcal{X}$ is an $N_a$-local efficient point (LEP) if
1. $\not\exists x \in N_a(x^*) \cap \mathcal{X}$ such that $g(x) \leq g(x^*)$; or, equivalently, if
2. $\forall x \in N_a(x^*) \cap \mathcal{X}$, $g(x) \not< g(x^*)$, that is, one of the following holds: (a) $\exists k \in \{1, \ldots, d\}$ such that $g_k(x^*) < g_k(x)$, or (b) $g(x) = g(x^*)$.

Notice that every $N_a$-local minimizer of some objective $g_k$, $k \in \{1, \ldots, d\}$, is an $N_a$-LWEP, and if the $N_a$-local minimizer is the unique minimum in its neighborhood, then it is also an $N_a$-LEP. Further, all $N_a$-LEP’s are $N_a$-LWEP’s. We define a global minimizer, a global weakly efficient point (GWEP), and a global efficient point (GEP) as an $N_a$-local minimizer, $N_a$-LWEP, and $N_a$-LEP, respectively, in which we set the neighborhood size parameter $a = \infty$.

2.1.2. Efficient and Pareto Sets We collect the various types of efficient points defined in the previous section into various types of efficient sets, as follows.

**Definition 5.** A set $\mathcal{W}_a \subseteq \mathcal{X}, |\mathcal{W}_a| \geq 1$, is an $N_a$-local weakly efficient set (LWES) if
1. $\forall x^w \in \mathcal{W}_a$, $x^w$ is an $N_a$-LWEP, and
2. no points in $g(\mathcal{W}_a)$ strictly dominate other points in $g(\mathcal{W}_a)$, and
3. $\forall x \in N_a(\mathcal{W}_a) \cap \mathcal{X}$, $\exists x^w \in \mathcal{W}_a$ such that $g(x^w) \leq g(x)$.
Definition 6. A set \( L_a \subseteq X, |L_a| \geq 1 \), is an \( N_a \)-local efficient set (LES) if (a) \( \forall x^* \in L_a \), \( x^* \) is an \( N_a \)-LEP, (b) no points in \( g(L_a) \) dominate other points in \( g(L_a) \), and (c) \( \forall x \in N'_a(L_a) \cap X, \exists x^* \in L_a \) such that \( g(x^*) \leq g(x) \).

Notice that every \( N_a \)-LES is also an \( N_a \)-LWES. Finally, we define the global weakly efficient set (GWES), denoted \( E^w \), and the global efficient set (GES), denoted \( E \), as an \( N_a \)-LWES and \( N_a \)-LES, respectively, in which the neighborhood size is \( a = \infty \). Although our definitions exist primarily in the decision space so far, we also define a \( N_a \)-local Pareto set (LPS) as the image of an \( N_a \)-LES, \( g(L_a) \).

We remark here that under our definitions, there may exist \( N_a \)-LWEP’s that do not belong to an \( N_a \)-LWES. To see an example of such a case, consider Figure 1, and notice that \( g(x^*_1) \) is not dominated by the image of any points in the \( N_1 \)-neighborhood of \( x^*_g \), which are the points \( g(x^\min_1), g(x^\min_2) \), and \( g(x^\min_1) \). Therefore, \( x^*_g \) is an \( N_1 \)-LWEP. (It is also an \( N_1 \)-LEP.) However, \( \{x^*_g\} \) is not an \( N_1 \)-LWES because \( g(x^*_g) \) does not dominate \( g(x^\min_1), g(x^\min_2) \), or \( g(x^\min_1) \). In this example, it is not possible to construct an \( N_1 \)-LWES using only the points \( x^\min_1, x^*_g, \) and \( x^\min_2 \) because the images of these points do not dominate the images of any other feasible points, and therefore cannot dominate the images of the points in their deleted \( N_1 \)-neighborhood. Thus any \( N_1 \)-LWES including \( x^*_g \) must also include a member of the GES whose image does not dominate its image, such as \( x^\min_1, x^*_b, \) or \( x^\min_2 \). But, including any of these points in the candidate \( N_1 \)-LWES with \( x^*_g \) implies that there exist neighborhood points that violate the definition of an \( N_1 \)-LWES. Thus \( x^*_g \) does not belong to an \( N_1 \)-LWES.

2.2. Problem Statement

Using the optimality concepts defined in the previous section, we consider the following problem statement: Given a neighborhood size \( a \) and an oracle capable of producing estimators \( \hat{G}_n(x) \) of \( g(x) \) such that \( \hat{G}_n(x) \to g(x) \) w.p.1 as the sampling effort \( n \to \infty \) for each \( x \in X \subseteq \mathbb{Z}^q \), find a local solution to Problem \( M_d \), which is an \( N_a \)-LES.

3. Solution Context: Retrospective Approximation

To address our problem statement, we employ an RA framework, which is a version of SAA. First, we discuss the sample-path problem and solution in the context of SAA, as well as its use inside an RA framework. Then, we discuss an approximate sample-path solution designed to enhance the efficiency of our RA framework for MOSO on integer lattices.
3.1. The Sample-Path Problem and Solution

SAA is a solution framework for Problem $M_d$ defined by the sample-path problem

Problem $\overline{M}_{d,n}$:

$$\text{minimize}_{x \in \mathcal{X}} \left\{ \bar{G}_n(x) = (\bar{G}_{1,n}(x), \ldots, \bar{G}_{d,n}(x)) := \left( \frac{1}{n} \sum_{i=1}^{n} G_1(x, \xi_i), \ldots, \frac{1}{n} \sum_{i=1}^{n} G_d(x, \xi_i) \right) \right\},$$

where $\bar{G}_n(x)$ is an estimator of $g(x)$, and at each feasible point $x \in \mathcal{X}$ the oracle generates $n$ copies of the random objective vector $G(x, \xi_i) := (G_1(x, \xi_i), \ldots, G_d(x, \xi_i))$ for all $i = 1, \ldots, n$. For fixed values of the random variables $\xi_i$, $i = 1, \ldots, n$, solving the sample-path Problem $\overline{M}_{d,n}$ is a deterministic optimization problem. Thus we define sample-path versions of all optimality concepts in §2.1 by replacing the objective function values $g(x)$ and $g_k(x)$ with $\bar{G}_n(x)$ and $\bar{G}_{k,n}(x)$, respectively, for all $k \in \{1, \ldots, d\}$. We denote sample-path $\mathcal{N}_a$-local minimizers, sample-path $\mathcal{N}_a$-LWEP’s, and sample-path $\mathcal{N}_a$-LEP’s as $X_{k,n}^\text{min}$, $X_n^\text{w}$, and $X_n^*$, respectively. A local solution to Problem $\overline{M}_{d,n}$ is a sample-path $\mathcal{N}_a$-LES.

In the special case of SAA called RA, instead of solving Problem $\overline{M}_{d,n}$ at a static sample size $n$, we solve a sequence of sample-path problems. These sample-path problems are denoted Problem $\overline{M}_{d,m_\nu}$ at sample size $m_\nu$, where $\{m_\nu, \nu = 1, 2, \ldots\}$ is a sequence of increasing sample sizes and $\nu$ is the RA iteration number. The solution to Problem $\overline{M}_{d,m_{\nu-1}}$ is used as a warm start to find the solution to Problem $\overline{M}_{d,m_\nu}$. (Henceforth, within an RA iteration $\nu$, we usually denote the sample size as $n = m_\nu$ to reduce our use of double subscripts.)

3.2. An Approximate Sample-Path Solution

Assuming the sample-path Problem $\overline{M}_{d,n}$ solved within an RA iteration is well-behaved, one can solve it to optimality by locating a complete sample-path $\mathcal{N}_a$-LES. However, when sample sizes are small, there may be a relatively large error in estimating the objective vectors. Thus it may be inefficient to chase down all members of a sample-path $\mathcal{N}_a$-LES if the points are unlikely to be true $\mathcal{N}_a$-LES members — for a convergent algorithm, non-$\mathcal{N}_a$-LES members eventually will be eliminated from consideration in a future RA iteration. Instead, imagine that we solve the sample-path problem only to within a certain error tolerance that is commensurate with the amount of error we have in estimating the objective function values (see, e.g., Pasupathy 2010, for similar concepts in the stochastic root-finding context). To employ such a concept, we would like a relaxed definition of a sample-path $\mathcal{N}_a$-LES that will enable us to stop our search for a sample-path $\mathcal{N}_a$-LES within an RA iteration early. To this end, we define an approximate version of local optimality for Problem $\overline{M}_{d,n}$, as follows.
Definition 7. A set \( A \subseteq X \) is a sample-path approximate \( N_a\)-LES (ALES) for the sample-path Problem \( \overline{d} \), if no points in \( G_n(A) \) dominate other points in \( G_n(A) \) and, given a vector-valued completeness function \( \delta: X \rightarrow \mathbb{R}^d \) such that \( 0_d \leq \delta(x) \) for all \( x \in X \),

(a) \( \forall x^w \in A, x^w \) is a sample-path \( N_a\)-WEP (\( \forall x \in N_a(x^w) \cap X, G_n(x) \not\subset G_n(x^w) \)),

(b) \( \forall x \in N_a^* (A) \cap X \), (i) \( \exists x^w \in A \) such that \( G_n(x^w) \leq G_n(x) \), or (ii) \( \exists x^w \in A \) such that \( (\bar{G}_n(x^w) \leq \bar{G}_n(x^w)) \) and \( \bar{G}_n(x^w) - \delta(x^w) \leq \bar{G}_n(x) + \delta(x) \), or (iii) \( \forall x^w \in A, \bar{G}_n(x) \not\subset \bar{G}_n(x) \), and \( \exists x^w \in A \) such that \( \bar{G}_n(x^w) - \delta(x^w) \leq \bar{G}_n(x) + \delta(x) \) or \( \bar{G}_n(x) - \delta(x) \leq \bar{G}_n(x^w) + \delta(x^w) \).

Definition 7 is similar to the definition of a sample-path \( N_a\)-WES, except for Part (b). Definition 7(b) requires that all feasible points in the deleted neighborhood of the ALES are either (i) weakly dominated by a point in the set, or (ii) dominate a point in the set by less than a certain amount, or (iii) do not weakly dominate any points in the set, and would either weakly dominate or be weakly dominated by a point in the set if both were moved by a certain amount. The “certain amount” is specified by the function \( \delta \). We call \( \delta(\cdot) = (\delta_1(\cdot), \ldots, \delta_d(\cdot)) \) the completeness function because it allows the ALES to have neighborhood points that violate the definition of a sample-path \( N_a\)-WES, and bigger values of \( \delta \) result in a “less-complete” ALES. If \( \delta_k(x) = \infty \) for all \( x \in X, k \in \{1, \ldots, d\} \), the ALES is a collection of sample-path \( N_a\)-WES’s that may or may not belong to a sample-path \( N_a\)-WES. If \( \delta(x) = 0_d \) for all \( x \in X \), the ALES is a sample-path \( N_a\)-WES.

We set the completeness function using the estimated standard errors of the objective function values \( \bar{G}_{k,n}(x) \) for all \( k \in \{1, \ldots, d\} \), \( x \in X \). Thus we require the following definitions. For all \( x \in X \), \( k \in \{1, \ldots, d\} \), define the variance \( \sigma_k^2(x) := \forall(G_k(x, \xi)) < \infty \). (The assumption of finite variances is made formal in §8.) Then, let the estimated standard deviation of the kth objective value at \( x \in X \) be \( \delta_{k,n}(x) := \sqrt{(n-1)^{-1} \sum_{i=1}^{n}(G_k(x, \xi_i) - \bar{G}_{k,n}(x))^2} \), and let the standard error of the kth estimated objective value be \( s.e.(\bar{G}_{k,n}(x)) := \delta_{k,n}(x)/n^{1/2} \). Further, for each objective \( k \in \{1, \ldots, d\} \) and for all \( x \in X \), \( \beta \in (0, \infty] \), define the function \( \hat{f}_k(x, \beta) := s.e.(\bar{G}_{k,n}(x))(n^{1/2-\beta}) \) if \( \beta \in (0, \infty) \) and \( \hat{f}_k(x, \infty) := 0 \). Let \( \hat{f}(x, \beta) := (\hat{f}_1(x, \beta), \ldots, \hat{f}_d(x, \beta)) \).

For the remainder of the paper, we consider the ALES completeness function specified by \( \hat{f} \). Since the value of this function is random for each \( x \in X \), henceforth, we denote the completeness function as \( \hat{\delta}(x) := \hat{f}(x, \beta) \) for all \( x \in X \), where \( \beta_\delta \in (0, \infty] \) is the completeness parameter. Smaller values of \( \beta_\delta \) correspond to larger values of \( \hat{\delta}(x) \), thus specifying a less-complete ALES. The value \( \beta_\delta = \infty \) implies that the ALES is a sample-path \( N_a\)-WES. Since
the completeness function is a function of the standard error, assuming the variances are finite, the value $\delta(x) \to 0$ w.p.1 for each $x \in \mathcal{X}$ as the sampling effort increases.

4. The Main Algorithm: R-P$\varepsilon$RLE for Two Objectives

We are now ready to provide an overview of our main RA algorithm, R-P$\varepsilon$RLE, which is listed in Algorithm 1. As previously discussed, R-P$\varepsilon$RLE employs an RA framework that solves the sequence of sample-path Problems $M_{2,m_\nu}$ at increasing sample sizes $\{m_\nu, \nu = 1, 2, \ldots\}$, where $\nu$ is the RA iteration number. Within an RA iteration, we propose the deterministic P$\varepsilon$RLE algorithm as the sample-path solver. For compactness, we implicitly define the P$\varepsilon$RLE algorithm as calling P$\varepsilon$ followed by RLE within one RA iteration $\nu$ (Algorithm 1 Steps 3 and 4). The solution to Problem $M_{2,m_\nu}$ found by P$\varepsilon$RLE on the $\nu$th RA iteration, denoted $\hat{A}_\nu$ in Algorithm 1, is guaranteed by RLE to be an ALES. For efficiency, R-P$\varepsilon$RLE uses the sample-path solution from the preceding RA iteration, $\hat{A}_{\nu-1}$ which is an ALES for Problem $M_{2,m_{\nu-1}}$, as an initial set of points for finding an ALES that solves Problem $M_{2,m_\nu}$. Given the amount of detail inherent in the algorithms P$\varepsilon$ and RLE, we address these algorithms separately in §5 and §6, respectively.

R-P$\varepsilon$RLE requires a few input parameters, in addition to an initial feasible point $x_0 \in \mathcal{X}$ and a sequence of sample sizes $\{m_\nu, \nu = 1, 2, \ldots\}$. First, R-P$\varepsilon$RLE requires a sequence of limits on oracle calls during search, $\{b_\nu, \nu = 1, 2, \ldots\}$, that prevents chase-offs in the case of “bad” sample-path realizations of Problem $M_{2,m_\nu}$. We set this sequence so that for large enough RA iteration numbers $\nu$, search “time outs” due to binding $b_\nu$ do not occur w.p.1. (Such a sequence is also required by SPLINE in §5.3.) R-P$\varepsilon$RLE also requires parameters $\beta = (\beta_\varepsilon, \beta_\delta)$, which essentially control the completeness of the ALES and are discussed in the sections that follow. We suppress the choice of neighborhood size, which is $a = 1$ by default. The convergence properties of R-P$\varepsilon$RLE under different parameter values is discussed in §8, and we specify default settings for these parameters in §9. Under the default settings, the initial feasible point $x_0 \in \mathcal{X}$ is the only required user-specified input parameter for R-P$\varepsilon$RLE.

---

**Algorithm 1:** The R-P$\varepsilon$RLE Algorithm for $d = 2$

**Input:** initial point $x_0 \in \mathcal{X}$; sequence of sample sizes to expend at each visited point, $\{m_\nu\}$; sequence of limits on oracle calls during search, $\{b_\nu\}$; $\varepsilon$-placement and ALES parameters, $\beta = (\beta_\varepsilon, \beta_\delta)$

1. Initialize: $\hat{A}_0 = \{x_0\}$ and set $x_0$ as a global variable

2. for $\nu = 1, 2, \ldots$ with CRN do

3. $\quad \hat{A}_\nu = \text{P} \varepsilon(\hat{A}_{\nu-1}, m_\nu, b_\nu, \beta_\varepsilon)$

4. $\quad \hat{A}_\nu = \text{R} \varepsilon(\hat{A}_\nu, m_\nu, b_\nu, \beta_\delta)$

/partition and solve $\varepsilon$-constraint problems

/guarantee the returned set is an ALES
5. The Pε Algorithm for Two Objectives

The Pε algorithm is the first algorithm in our proposed two-part sample-path solver, PεRLE. In RA iteration ν, the Pε algorithm uses the ε-constraint method to find a collection of points that are sample-path $\mathcal{N}_a$-LWEP’s for Problem $\overline{M}_{2,m_\nu}$; recall that $n = m_\nu$ is the current sample size inside RA iteration ν. Using the ε-constraint method, we re-formulate the sample-path Problem $\overline{M}_{2,n}$ into a set of constrained single-objective problems. Given an objective to minimize $k^* \in \{1, 2\}$ and an ε value, the sample-path ε-constraint problem is

$$\text{Problem } \bar{S}_{2,n}(k^*, \varepsilon): \text{ minimize } x \in \mathcal{X} \hat{G}_{k^*,n}(x) \text{ s.t. } \bar{G}_{k^{\text{con}},n}(x) \leq \varepsilon \text{ for } k^{\text{con}} \in \{1, 2\}, k^{\text{con}} \neq k^*.$$

On an integer lattice, the sample-path $\mathcal{N}_a$-local minimizer for Problem $\bar{S}_{2,n}(k^*, \varepsilon)$ is guaranteed to be an $\mathcal{N}_a$-LWEP for the original sample-path Problem $\overline{M}_{2,n}$ (see, e.g., Miettinen 1999, for the continuous context). Except in pathological cases, varying the ε values and solving each resulting sample-path ε-constraint problem results in locating multiple sample-path $\mathcal{N}_a$-LWEP’s for Problem $\overline{M}_{2,n}$.

The Pε algorithm operates as follows. First, it finds sample-path $\mathcal{N}_a$-local minimizers on each objective to bound the space where ε values are placed. Then it selects an objective $k^*$ to minimize and solves a collection of Problems $\bar{S}_{2,n}(k^*, \varepsilon)$ at a set of carefully-placed ε values. We use the $\hat{f}$ function defined in §3.2 to place the ε values a function of the standard error away from the images of known sample-path $\mathcal{N}_a$-LWEP’s. Such known sample-path $\mathcal{N}_a$-LWEP’s may have been carried forward from the previous RA iteration, or may have been found during the current RA iteration. Thus Pε yields a collection of sample-path $\mathcal{N}_a$-LWEP’s that includes a local minimizer on each objective and the sample-path $\mathcal{N}_a$-LWEP’s that result from solving Problem $\bar{S}_{2,n}(k^*, \varepsilon)$ for each chosen ε.

5.1. Pε Algorithm Listing

We now discuss the Pε algorithm (Algorithm 2) in more detail. First, to ensure that all sample-path ε-constraint problems have a non-empty feasible set, in Step 1, Pε obtains bounds on the objective values of a sample-path $\mathcal{N}_a$-LWES by obtaining a set, $\hat{A}_n^{0}$, containing updated sample-path $\mathcal{N}_a$-local minimizers on each objective. The MÍN algorithm called in Step 1 is listed in Algorithm 3 and discussed further in §5.2. The sample-path $\mathcal{N}_a$-local minimizers at current sample size n ensure that for the constrained objective $k^{\text{con}}$, no ε values are placed outside of the interval $(\bar{G}_{k^{\text{con}},n}(X_{k^{\text{con}},n}^{\text{min}}), \bar{G}_{k^{\text{con}},n}(X_{k^*,n}^{\text{min}}))$ for $k^{\text{con}} \neq k^*$. Thus for every ε-constraint problem posed, there exists a sample-path feasible point in the set $\hat{A}_n^{0}$.
Algorithm 2: $\hat{A}_{\text{new}} = P\varepsilon(\hat{A}_{\text{old}}, n, b, \beta_z)$

**Input:** estimated efficient set from the last RA iteration, $\hat{A}_{\text{old}} \subseteq X$; sample size, $n$; limit on oracle calls during search, $b$; epsilon placement parameter, $\beta_z$

**Output:** $\hat{A}_{\text{new}} \subseteq X$, a collection of sample-path $N_\alpha$-LWEP's

1. $\hat{A}_n^0 = \text{MIN}(\hat{A}_{\text{old}} \cup \{x_0\}, n, b)$ /SEARCH: update sample-path $N_\alpha$-local minimizers
2. $[\sim, \hat{A}_{\text{new}}, \sim] = \text{REMOVENONLWEP}(\hat{A}_n^0)$ /get the set of sample-path $N_\alpha$-LWEP's in $\hat{A}_n^0$
3. if $\hat{A}_n^0 = \emptyset$ then $\hat{A}_n^0 \leftarrow \hat{A}_n^0$ /MIN update timed out, no other sample-path $N_\alpha$-LWEP's exist
4. Initialize: $c^0 \leftarrow |\hat{A}_n^0|$ /set $c$'s using sample-path $N_\alpha$-LWEP's
5. for $k^* = 1, 2$ do /determine objective to minimize, $k^*$
   6. Initialize: $k^*_{\text{con}} \leftarrow k$ for $k \in \{1, 2\}$ such that $k \neq k^*$
   7. Sort $G_n(\hat{A}_n^0)$ on $k^*_{\text{con}}$ to get $(X_{11}^w, \ldots, X_{c^0}^w)$ where $\hat{G}_{k^*_{\text{con}}, n}(X_{11}^w) \leq \ldots \leq \hat{G}_{k^*_{\text{con}}, n}(X_{c^0}^w)$
   8. Set constraint lower bound $L_{k^*} \leftarrow \hat{G}_{k^*_{\text{con}}, n}(X_{11}^w) + \hat{f}_{k^*_{\text{con}}}(X_{11}^w, \beta_z)$
   9. for $i = 2, \ldots, c^0$
      10. [ \begin{align*}
          \varepsilon_{k^*}^i &= \hat{G}_{k^*_{\text{con}}, n}(X_{11}^w) - \hat{f}_{k^*_{\text{con}}}(X_{11}^w, \beta_z) \\
          \varepsilon_{k^*}^i(i) &= \hat{G}_{k^*_{\text{con}}, n}(X_{i}^w) + \hat{f}_{k^*_{\text{con}}}(X_{i}^w, \beta_z)
        \end{align*} ]
      11. $c_{k^*} \leftarrow \{ \varepsilon_{k^*}^i(i) : i \in \{2, \ldots, c^0\}, L_{k^*} - \varepsilon_{k^*}^i(i) < \varepsilon_{k^*}^i(i) \not\in (\varepsilon_{k^*}^i(i), \varepsilon_{k^*}^i(i)) \}$ for all $i \in \{2, \ldots, c^0\}$
      12. $C_{k^*} \leftarrow \{ c_{k^*} \}$
   13. $K^* \leftarrow \arg \min \{ C_{k^*} : k^* \in \{1, 2\} \}$ /choose least $\varepsilon$-constraints; break ties randomly
   14. $C \leftarrow C_{K^*}$ and $K^*_{\text{con}} \leftarrow k$ for $k \in \{1, 2\}$ such that $k \neq K^*$
   15. if $C > 0$ then /if $C > 0$
      16. Sort $C_{K^*}$ in ascending order to get the ordered list $(\varepsilon_1, \ldots, \varepsilon_C)$
      17. for $j = 1, 2, \ldots, C - 1$ /partition space
      18. [ \begin{align*}
          \varepsilon_j^i &= \max(L_{K^*}, \max(\varepsilon_{K^*}^i(i) : \varepsilon_{K^*}^i(i) < \varepsilon_j, i \in \{2, \ldots, c^0\}))
        \end{align*} ] /get traceback lower bound
      19. Initialize: $\varepsilon_{\text{new}} \leftarrow \varepsilon_j$, $A_j \leftarrow \emptyset$, $T \leftarrow \emptyset$
      20. while $\varepsilon_j^i < \varepsilon_{\text{new}}$ do /find new sample-path $N_\alpha$-LWEP's
          21. $X^0 \leftarrow \arg \min \{ \hat{G}_{K^*_{\text{con}}, n}(X) : X \in T \cup A_j, \hat{G}_{K^*_{\text{con}}, n}(X) \leq \varepsilon_{\text{new}} \}$
          22. $[X^0, T', N(A_j^0)] = \text{SPLINE}(K^*, X^0, \lambda_{K^*}(\varepsilon_{\text{new}}), n, b) /\text{SEARCH: solve } \varepsilon\text{-constrained}$
          23. $A_j \leftarrow A_j \cup \{X_n^0\}$ and $T \leftarrow T \cup T'$
          24. $\varepsilon_{\text{new}} \leftarrow \hat{G}_{K^*_{\text{con}}, n}(X_n^0) - \hat{f}_{K^*_{\text{con}}}(X_n^0, \beta_z)$ /traceback: set new $\varepsilon$ to disqualify
      25. $\hat{A}_{\text{LWEP}} = \bigcup_{j=1}^C A_j$ /collect new sample-path $N_\alpha$-LWEP's
      26. $\hat{A}_{\text{new}} = \text{REMOVEDOMINATED}(G_n(\hat{A}_n^0 \cup \hat{A}_{\text{LWEP}} \cup \{x_0\}))$ /remove dominated, do no harm

To select an objective $k^* \in \{1, 2\}$ to minimize, in Steps 2 through 14, $P\varepsilon$ places constraints a function of the standard error away from the images of known sample-path $N_\alpha$-LWEP’s on both objectives, and ultimately selects the objective that results in solving the least number of $\varepsilon$-constraint problems. This strategy ensures that we do not constrain an objective with relatively small standard errors, thus wasting simulation effort attempting to order many points on a high standard error objective. To determine the set of $\varepsilon$-constraint values for each objective, first, in Step 2, $P\varepsilon$ creates an initial set of known sample-path $N_\alpha$-LWEP’s, $\hat{A}_n^w$, from the points in $\hat{A}_n^0$ using the REMOVENONLWEP function (discussed in §6.2). If there are no such points in $\hat{A}_n^0$, the search budget value $b$ must have been binding in the MIN algorithm. Then in Step 3, we set $\varepsilon$ values based on $\hat{A}_n^0$. Since $b$ is non-binding in the limit, for large enough RA iteration number $\nu$, all $\varepsilon$ values will be set based on
known sample-path \( N_a \)-LWEP’s. In Steps 5 through 12, \( P_\varepsilon \) sorts the initial points and partitions the objective space by placing \( \varepsilon \) values in each part of the objective space where the standard error intervals of the initial points do not overlap. The standard error intervals in Steps 5 through 12 are defined using the function \( \hat{f} \) from §3.2. For all known sample-path \( N_a \)-LWEP’s, denoted \( X_n^w \), the \( P_\varepsilon \) algorithm does not place any new \( \varepsilon \) values in the interval 
\[
( \bar{G}_{k,con,n}(X_n^w) - \hat{f}_{k,con}(X_n^w, \beta_\varepsilon), \bar{G}_{k,con,n}(X_n^w) + \hat{f}_{k,con}(X_n^w, \beta_\varepsilon) ),
\]
where \( \beta_\varepsilon \in (0, \infty) \) is a parameter setting. The \( \beta_\varepsilon \) parameter controls how large the standard error intervals are; notice that larger standard error values and smaller \( \beta_\varepsilon \) values both result in wider intervals. Once both sets of \( \varepsilon \)-constraint values have been determined, Steps 13 and 14 select the objective to minimize, where ties are broken randomly.

If the chosen objective to minimize results in solving one or more \( \varepsilon \)-constraint problems, these problems are solved in Steps 15 through 25. First, \( P_\varepsilon \) partitions the objective space based on the \( \varepsilon \) values. Then, each \( \varepsilon \)-constraint problem is solved using the single-objective pseudo-gradient-based SPLINE algorithm, further discussed in §5.3. Within a partition of the objective space, once a new sample-path \( N_a \)-LWEP is found as the solution to the initial \( \varepsilon \)-constraint problem, \( P_\varepsilon \) performs what we call a traceback in Step 24 by attempting to place a new \( \varepsilon \) value that is both within the current partition and that sets as infeasible the newly-found sample-path \( N_a \)-LWEP. If the new \( \varepsilon \) value is outside the partition, the search ends; otherwise, the new \( \varepsilon \)-constraint problem is solved, and this process repeats until no more \( \varepsilon \) values can be placed in the current partition. The new sample-path \( N_a \)-LWEP’s found across all partitions are collected into a set of sample-path non-dominated \( N_a \)-LWEP’s for Problem \( M_{2,n} \) and returned by \( P_\varepsilon \) in Step 26. We remark here that as the RA iteration number \( \nu \to \infty \) in R-\( P_\varepsilon \)RLE, under appropriate regularity conditions, the \( P_\varepsilon \) algorithm solves as many \( \varepsilon \)-constraint problems as there are points in the \( N_a \)-LES to which the algorithm converges (see §8).

5.2. The Min Algorithm for Many Objectives

The Min algorithm for \( d \geq 2 \) objectives, called in \( P_\varepsilon \) Step 1 and listed in Algorithm 3, is a relatively simple algorithm that takes in any set of feasible points, brings up the sample sizes, updates the sample-path \( N_a \)-local minimizers on each objective, and removes any points whose images are sample-path dominated. The resulting set of sample-path non-dominated points and sample-path \( N_a \)-local minimizers are returned as the set \( \hat{A}_n^* \).
5.3. The SPLINE Algorithm for One Objective

The SPLINE algorithm of Wang et al. (2013) is the engine that underlies all of our deterministic single-objective searches, that is, solving the ε-constraint problems in the P ε algorithm, Step 22, and finding the sample-path N ε-local minimizers in the Min algorithm, Step 3. The SPLINE algorithm (Algorithm 4) finds a sample-path N ε-local minimizer of objective k on a feasible set X using sample size n. Our version of the SPLINE algorithm contains minor modifications that allow us to input an objective to minimize, input the feasible space, and output the search trajectory for later use in our algorithms.

The SPLINE algorithm consists of two primary steps: SPLI and NE, which are called iteratively until a sample-path N ε-local minimizer is found, or the search times out. The SPLI algorithm conducts a pseudogradient-based line search with piecewise linear interpolation. The NE algorithm performs neighborhood enumeration to either move to a better neighborhood point, or to certify a local minimum has been found. We refer the reader to Wang et al. (2013) for detailed listings and further explanations of SPLI and NE.

We remark here that Liuzzi et al. (2018) also provide a line search algorithm for integer lattices. Certainly, other algorithms could be used in place of SPLINE; we select SPLINE because of its impressive performance on single-objective SO problems in Wang et al. (2013).

---

### Algorithm 3: \( \mathcal{A}_n^* = \min(\hat{\mathcal{A}}_n, n, b) \)

**Input:** a set of feasible points \( \hat{\mathcal{A}}_n \subseteq \mathcal{A} \); sample size, \( n \); limit on SPLINE calls, \( b \)

**Output:** \( \mathcal{A}_n^* \), a candidate EALES with updated sample-path local minimizers at sample size \( n \)

for \( k = 1, 2, \ldots, d \) do

1. \( X_{\min, k, \text{old}} \leftarrow \arg\min \{ G_{k, n}(x) : x \in \hat{\mathcal{A}}_n \} \)
2. \( [X_{\min, k, \text{old}}, \sim] = \text{SPLINE}(k, X_{\min, k, \text{old}}, \mathcal{X}, n, b) \) /SEARCH: sample-path \( N_\epsilon \)-local minimizer
3. \( \mathcal{M}_n \leftarrow \bigcup_{k=1}^{d} \{ X_{\min, k, n} \} \) /points in \( \tilde{G}_n(\mathcal{M}_n) \) may dominate other points in \( \tilde{G}_n(\mathcal{M}_n) \)
4. \( \mathcal{A}_n^* = \text{REMOVE-dominated}(\tilde{G}_n(\mathcal{A}_n \cup \mathcal{M}_n \cup \{ x_0 \})) \)

---

### Algorithm 4: \([\mathbf{x}^*, T, \mathcal{N}(\mathbf{x}^*)] = \text{SPLINE}(k, \mathbf{x}_0, \mathcal{X}, n, b)\)

**Input:** objective \( k \); initial point \( \mathbf{x}_0 \in \mathcal{X} \); feasible set \( \mathcal{X} \); sample size \( n \); limit on search oracle calls, \( b \)

**Output:** local solution \( \mathbf{x}^* \) on \( \mathcal{X} \); sample-path search set, \( T \); neighborhood points, \( \mathcal{N}(\mathbf{x}^*) \)

1. Initialize: search oracle calls spent so far \( b^* \leftarrow 0 \), \( \mathbf{x}_\text{NE} \leftarrow \mathbf{x}_0 \), and \( T \leftarrow \{ \mathbf{x}_0 \} \)
2. Repeat

3. [\( \mathbf{b}^*, \mathbf{x}_\text{SPLI}, G_n(\mathbf{x}_\text{SPLI}) \] = SPLI(\( k, \mathbf{x}_\text{NE}, \mathcal{X}, n, b \)) /SEARCH: line search with interpolation
4. if \( G_{k, n}(\mathbf{x}_\text{SPLI}) > G_{k, n}(\mathbf{x}_\text{NE}) \) then \( \mathbf{x}_\text{SPLI} \leftarrow \mathbf{x}_\text{NE} \) /SPLI cannot cause harm
5. \( [\mathbf{b}^*, \mathbf{x}_\text{NE}, G_n(\mathbf{x}_\text{NE}), \mathcal{N}(\mathbf{x}_\text{NE})] = \text{NE}(k, \mathbf{x}_\text{SPLI}, \mathcal{X}, n) \) /neighborhood enumeration
6. \( T \leftarrow T \cup \{ \mathbf{x}_\text{SPLI}, \mathbf{x}_\text{NE} \} \) /update trajectory
7. \( b^* \leftarrow b^* + b^* + b^n \) /update oracle calls expended
8. until \( G_{k, n}(\mathbf{x}_\text{NE}) = G_{k, n}(\mathbf{x}_\text{SPLI}) \) or \( b^* > b \) /find a local solution or time out
6. The RLE Algorithm for Many Objectives

The RLE algorithm is the second algorithm in our proposed two-part sample-path solver, \( P_{\varepsilon} \)RLE. The collection of sample-path \( \mathcal{N}_a \)-LWEP’s found by the \( P_{\varepsilon} \) algorithm is sent to RLE to certify that this collection of points is indeed an ALES, or to create an ALES using this collection of points as an initial set. Without RLE to certify an ALES, an algorithm like \( P_{\varepsilon} \) that relies only on collecting sample-path \( \mathcal{N}_a \)-LWEP’s may “get stuck” by returning points that do not belong to the same, or to any, sample-path \( \mathcal{N}_a \)-LWES. For example, in Figure 1, the \( P_{\varepsilon} \) algorithm may return \( S = \{ x_{1,\text{min}}, x_{g,\text{min}}, x_{2,\text{min}} \} \), which is a set of \( \mathcal{N}_a \)-LWEP’s containing a minimum on each objective but that is not an \( \mathcal{N}_a \)-LWES. The RLE algorithm is designed to crawl out of the sample-path version of this scenario when the completeness function is small enough. In what follows, we first define a key concept used in RLE called the sample-path non-conforming neighborhood. Then, we discuss the RLE algorithm in detail.

6.1. The Sample-Path Non-Conforming Neighborhood

Suppose we are given a set of feasible points \( S \) such that none of the estimated images of points in \( S \) dominate the estimated images of other points in \( S \). The non-conforming neighborhood of \( S \), defined in Definition 8, is the set of points in the deleted neighborhood of \( S \) that prevent it from being an ALES.

**Definition 8.** Let \( S \subseteq \mathcal{X} \) be a collection of feasible points such that no points in \( \bar{G}_n(S) \) dominate other points in \( G_n(S) \). Then given a completeness function \( \delta: \mathcal{X} \rightarrow \mathbb{R}^d \) such that \( 0_d \leq \delta(x) \) for all \( x \in \mathcal{X} \), define the sample-path non-conforming neighborhood (NCN) of \( S \), \( \mathcal{N}_{a,\text{nc}}(S) \), as all feasible points in the deleted neighborhood of \( S \), \( x \in \mathcal{N}_a(S) \cap \mathcal{X} \), such that

(a) \( \exists \bar{x} \in S \) such that \( x \in \mathcal{N}_a(\bar{x}) \) and \( G_n(x) < G_n(\bar{x}) \), or

(b) \( i) \not\exists \bar{x} \in S \) such that \( G_n(\bar{x}) \leq G_n(x) \), and \( ii) \not\exists \bar{x} \in S \) such that \( (G_n(x) \leq G_n(\bar{x})) \) and \( G_n(x) - \delta(\bar{x}) \leq G_n(x) + \delta(x) \), and \( iii) \exists \bar{x} \in S \) such that \( G_n(x) \leq G_n(\bar{x}) \), or \( \not\exists \bar{x} \in S \) such that \( G_n(\bar{x}) - \delta(\bar{x}) \leq G_n(x) + \delta(x) \) or \( G_n(x) - \delta(x) \leq G_n(\bar{x}) + \delta(\bar{x}) \).

First, Definition 8(a) adds \( x \in \mathcal{N}_a(S) \cap \mathcal{X} \) to the NCN if it prevents a point in \( S \) from being a sample-path \( \mathcal{N}_a \)-LWEP. Definition 8(b) also adds a feasible deleted-neighborhood point to the NCN if it violates the conditions of Definition 7(b), that is, the point (i) is not weakly dominated by any points in \( S \), and (ii) does not dominate any points in \( S \) by less than a certain amount, and (iii) weakly dominates a point in \( S \), or would not either weakly dominate or be weakly dominated by a point in \( S \) if both were moved by a certain amount.
Given a singleton feasible set \( \mathcal{S} = \{ \tilde{x} \} \), Figure 2 shows regions of the objective space that correspond to a feasible point in the deleted neighborhood of \( \mathcal{S} \), \( x \in \mathcal{N}_{a}(\tilde{x}) \cap X \), being declared a member of the NCN. Definition 8(a) implies \( x \) is in the NCN if it is an \( \mathcal{N}_{a} \)-neighbor of \( \tilde{x} \) and \( G_{n}(x) \) is in the light gray region of Figure 2. Excluding points that meet the requirements of Definition 8(a), Definition 8(b) implies that \( x \) is in the NCN if its entire “\( \delta \) box,” defined by the corners \( G_{n}(x) \pm \delta(x) \), is completely contained in the dark gray shaded region of Figure 2. Thus there is no overlap between the \( \delta \) boxes of \( x \) and \( \tilde{x} \) on any objective. Henceforth, we use \( \hat{\delta}(x) = \hat{f}(x, \beta) \) for all \( x \in X \) (see §3.2) as the NCN completeness function.

6.2. RLE Algorithm Listing

We now discuss the RLE algorithm (Algorithm 5) in detail. To guarantee an ALES, in Step 1, RLE first removes any points in \( \mathcal{S} \) whose estimated objective vectors are dominated by the estimated objective vectors of other points in \( \mathcal{S} \), since these points cannot be members of an ALES. Then, in Step 2, RLE calculates the NCN of the remaining points using the function \( \text{GetNCN} \). If the NCN is empty in Step 2, then RLE certifies it has found an ALES and the algorithm terminates; otherwise, RLE enters a search phase.

The “outer” RLE search phase, which begins in Step 4, checks to see if any members of the NCN are also sample-path \( \mathcal{N}_{a} \)-LWEP’s using \( \text{RemoveNonLWEP} \). The \( \text{RemoveNonLWEP} \) function takes an input set of feasible points and outputs three quantities: (a) the number of simulation replications expended, (b) the set of sample-path \( \mathcal{N}_{a} \)-LWEP’s in the input set, and (c) a set of points in the neighborhood of the input set whose images sample-path dominate the images of the members of the input set. Thus when \( \text{RemoveNonLWEP} \) is passed a non-empty NCN, it enumerates the neighborhood of the NCN; these points are neighbors of neighbors of the original set. If the NCN contains sample-path \( \mathcal{N}_{a} \)-LWEP’s in
Algorithm 5: $\hat{A}_{ALES} = \text{RLE}(S, n, b, \beta_\delta)$

Input: set of points $S \subset X$; limit on search oracle calls, $b$; ALES completeness parameter $\beta_\delta$

Output: $\hat{A}_{ALES}$, which is an ALES for the sample-path problem with sample size $n$

1. $S = \text{REMOVE DOMINATED}(G_n(S \cup \{x_0\}))$
2. $N^{nc} = \text{GetNCN}(S, \beta_\delta)$ /get non-conforming neighborhood
3. Initialize: outer search oracle calls spent so far $b^* \leftarrow 0$
4. while $b^* \leq b$ and $N^{nc} \neq \emptyset$ do
5.   \begin{align*}
       \left[ b', N^{w}, N^{2}\right] &= \text{REMOVE NON LWEP}(N^{nc})
   \end{align*}
6.   $S \leftarrow S \cup N^{w}$ /add neighborhood sample-path $N_a$-LWEP’s to $S$
7. if $N^{w} = \emptyset$ then /the neighbors are dominated by their neighbors: $N^{2}_\delta \neq \emptyset$
8.   \begin{align*}
       X^{new} &\leftarrow N^{2}_\delta \\
       b^{**} &\leftarrow b^{**} + b'
       \end{align*}
9.   \begin{align*}
       \left[ b'', X^w, N^{2}\right] &= \text{REMOVE NON LWEP}(X^{new})
   \end{align*}
10. \begin{align*}
       X^{new} &\leftarrow N^{2}_\delta \\
       b^{**} &\leftarrow b^{**} + b''
       \end{align*}
11. $S \leftarrow S \cup X^w$ /keep progress if search times out
12. if $X^w = \emptyset$ then $S \leftarrow S \cup X^{new}$
13. $b^* \leftarrow b^* + b' + b''$
14. $S = \text{REMOVE DOMINATED}(G_n(S \cup \{x_0\}))$
15. $[b'', N^{nc}] = \text{GetNCN}(S, \beta_\delta)$ /get non-conforming neighborhood
16. return $\hat{A}_{ALES} \leftarrow S$

Step 5, RLE adds them to $S$ in Step 6. If no members of the NCN are sample-path $N_a$-LWEP’s in Step 7, there must exist neighbors of the NCN that dominate points in the NCN, denoted as $N^*_\delta$ in Steps 5 and 8. If this is the case, RLE enters an “inner” search phase.

The “inner” RLE search phase, which begins in Step 9, allows the algorithm to find new sample-path $N_a$-LWEP’s by traversing points whose estimated images are sample-path dominated. Once a sample-path $N_a$-LWEP is found or the inner search times out, the new points are added to $S$ in Steps 12 and 13. After removing points whose images are sample-path dominated by the images of other points in $S$, in Step 15, RLE checks the new set $S$ to see if it is an ALES. If not, this process repeats until a complete ALES is found, or a total outer search budget has been exhausted. Since the search budget sequence is non-binding in the limit, for large enough RA iteration number $\nu$, RLE guarantees an ALES.

7. Other Algorithms: R-P$\varepsilon$ and R-MinRLE

To assess and understand the performance of R-P$\varepsilon$RLE, we find it helpful to define and analyze two other RA algorithms: R-P$\varepsilon$ and R-MinRLE, which we discuss in this section. We discuss R-P$\varepsilon$ for two objectives first, followed by R-MinRLE for many objectives.

First, we define the R-P$\varepsilon$ algorithm for two objectives as identical to R-P$\varepsilon$RLE (Algorithm 1), except without the call to RLE in Step 4. To show that the P$\varepsilon$ algorithm delivers
Algorithm 6: The R-MinRLE Algorithm for \(d \geq 2\)

**Input:** initial point \(x_0 \in \mathcal{X}\); sequence of sample sizes to expend at each visited point, \(\{m_\nu\}\); sequence of limits on oracle calls during search, \(\{b_\nu\}\); ALES completeness parameter, \(\beta_\delta\)

1. Initialize: \(\hat{A}_0 = \{x_0\}\) and set \(x_0\) as a global variable
2. for \(\nu = 1, 2, \ldots\) with CRN do
   3. \(\hat{A}_{\min} = \text{MIN}(\hat{A}_{\nu-1}, m_\nu, b_\nu)\) /update the sample-path \(N_a\)-local minimizers
   4. \(\hat{A}_\nu = \text{RLE}(\hat{A}_{\min}, m_\nu, b_\nu, \beta_\delta)\) /guarantee the returned set is an ALES

“good points” to the RLE algorithm, in §8, we prove convergence of the R-P\(\varepsilon\) algorithm under a set of fairly restrictive assumptions; we remark that these assumptions are violated in the example in Figure 1. We also find the R-P\(\varepsilon\) algorithm useful for numerically analyzing the settings of the \(\beta_\varepsilon\) parameter in §10.2. We emphasize here that we do not recommend R-P\(\varepsilon\) for implementation. Unless the decision-maker has knowledge of special structure in the objective functions, one should always choose R-P\(\varepsilon\)RLE over R-P\(\varepsilon\) for bi-objective SO.

The R-MINRLE algorithm, listed in Algorithm 6, is arguably the most general algorithm we propose, since it is defined for \(d \geq 2\) objectives. R-MINRLE is like R-P\(\varepsilon\)RLE except that instead of obtaining a set of sample-path \(N_a\)-LWEP’s from P\(\varepsilon\) on each RA iteration, R-MINRLE uses the MIN algorithm (Algorithm 3) to update the sample-path \(N_a\)-local minimizer on each objective before invoking RLE. We define the sample-path solver MINRLE as calling MIN followed by RLE within an RA iteration; the MINRLE algorithm locates an ALES for sample-path Problem \(M_{d,m_\nu}\) for \(d \geq 2\) and each \(\nu = 1, 2, \ldots\).

We view the R-MINRLE algorithm as a naïve pseudo-gradient-based benchmark algorithm for many objectives. Loosely speaking, for two objectives, notice that R-MINRLE is likely to exhibit “outside-in” convergence behavior. That is, because R-MINRLE only guarantees locating the sample-path \(N_a\)-local minimizers on each RA iteration, if the completeness parameter \(\beta_\delta\) is “small” so that RLE crawls less, MINRLE locates the sample-path \(N_a\)-local minimizers and perhaps a few points nearby to complete an ALES. All externalities being equal, this ALES is less likely to contain points that map to the “center” of a sample-path \(N_a\)-LPS than a corresponding ALES located by P\(\varepsilon\)RLE. Therefore in our numerical experiments, comparisons with R-MINRLE demonstrate the usefulness of using the P\(\varepsilon\) algorithm as a precursor to RLE within an RA iteration, as opposed to the naïve MIN algorithm.

8. Asymptotic Behavior

We now study the asymptotic behavior of our RA algorithms and show that, under appropriate regularity conditions, R-P\(\varepsilon\)RLE, R-P\(\varepsilon\), and R-MINRLE converge to an \(N_a\)-LES w.p.1.
In the sections that follow, first, we discuss the assumptions required for our results to hold. Then, we prove the convergence of algorithms that rely on RLE, that is, R-PεRLE and R-MINRLE. The proof of convergence for algorithms that rely on RLE is general in the sense that, under our regularity conditions, the proof provides a convergence result for any RA algorithm designed to solve Problem $M_d$ whose sample-path solver invokes RLE last with the completeness function $\hat{\delta}$ and $\beta_{\delta} \in (0, \infty]$. We also prove the convergence of R-Pε under fairly restrictive assumptions. Finally, in §8.4, we provide sampling efficiency results. Throughout this section, we assume the search budget sequence $\{b_{\nu}, \nu = 1, 2, \ldots\}$ is non-binding w.p.1 for all $\nu$ large enough, under our regularity conditions. Therefore we ignore issues related to binding budget sequences for small $\nu$. We specify the default budget sequence in §9.

8.1. Preliminaries and Assumptions

We require several regularity conditions on both the true, unknown objective functions and the sample-path objective functions. We require Assumptions 1–3 in all of our results.

**Assumption 1.** (Wang et al. 2013, p. 12) For each $x \in \mathcal{X}$ and $k \in \{1, \ldots, d\}$, there exists $\alpha_k > 0$, dependent on $x$, such that $S_k(x, \alpha_k) := \{x' \in \mathcal{X}: g_k(x') \leq g_k(x) + \alpha_k\}$ is finite.

**Assumption 2.** (Wang et al. 2013, p. 12) For each $k \in \{1, \ldots, d\}$, we assume the following. Let $S_k(x, \alpha_k)$ be as in Assumption 1, and define $\hat{S}_{k,\nu}(x) := \{x' \in \mathcal{X}: \bar{G}_{k,m_{\nu}}(x') \leq \bar{G}_{k,m_{\nu}}(x)\}$ for all $\nu = 1, 2, \ldots$. Given $x \in \mathcal{X}$, there exists a sequence $\{p_{\nu}\}_{\nu=1}^{\infty}$ such that $\mathbb{P}\{\tilde{x} \in \hat{S}_{k,\nu}(x)\} \leq p_{\nu}$ and $\sum_{\nu=1}^{\infty} p_{\nu} < \infty$ for all $\tilde{x} \in \mathcal{X} \setminus S_k(x, \alpha_k)$.

**Assumption 3.** All variances are finite, that is, $\max_{k \in \{1, \ldots, d\}} \sigma_k^2(x) < \infty$ for all $x \in \mathcal{X}$.

First, Assumption 1 implies that at each feasible point $x \in \mathcal{X}$ and for every objective $k$, there exists a constant such that the level set created by adding the constant to $g_k(x)$ is finite. Since this property holds for every objective $k$, then under Assumption 1, the union of the level sets over the objectives $k$ is also finite. That is, for each $x \in \mathcal{X}$, define $\alpha := (\alpha_1, \ldots, \alpha_d)$, where $\alpha$ also depends on $x$. Define the set $\mathcal{S}(x, \alpha) := \bigcup_{k=1}^{d} S_k(x, \alpha_k)$ as the set of all feasible points that map to objective values that are not strictly dominated by the point $g(x) + \alpha$; notice that $g(\mathcal{S}(x, \alpha)) = \{g(x) + \alpha\} + \{y \in \mathbb{R}^d: 0_d < y\}$ is the set of points that $g(x) + \alpha$ strictly dominates for all $x \in \mathcal{X}$. Then under Assumption 1, $\mathcal{S}(x, \alpha)$ is finite for all $x \in \mathcal{X}$.

Assumption 1 further implies the nonempty GWES exists and all $\mathcal{N}_a$-LWES’s are finite. This result is presented without proof in Lemma 1, where we first define the following
notation. Given a neighborhood size parameter \( a \in (0, \infty) \), let \( \mathcal{W}_a \) and \( \mathcal{L}_a \) be the collection of all possible \( \mathcal{N}_a \)-LWES’s and \( \mathcal{N}_a \)-LES’s for Problem \( M_d \), respectively, where \( \mathcal{L}_a \subseteq \mathcal{W}_a \). Since the GWES is also an \( \mathcal{N}_a \)-LWES for \( a \geq 1 \), notice that \( |\mathcal{W}_a| \geq 1 \) if the GWES exists.

**Lemma 1.** Under Assumption 1, given \( a \geq 1 \), the following hold:
(a) The GWES, \( \mathcal{E}^w \subseteq \mathcal{X} \), exists and is nonempty.
(b) All \( \mathcal{N}_a \)-LWES’s are finite; that is, \( 1 \leq |\mathcal{W}_a| < \infty \) for all \( \mathcal{W}_a \in \mathcal{W}_a \).

Now, let us turn our attention to Assumption 2, which is a condition defined by Wang et al. (2013) ensuring that the probability of incorrectly estimating a level set decays sufficiently fast. Wang et al. (2013) provide a detailed discussion of the conditions under which Assumption 2 holds. For completeness, we include the conditions below as Lemma 2; recall that the variance \( \sigma_k^2(x) \) is \( \mathbb{V}(G_k(x, \xi)) \) for each \( x \in \mathcal{X} \) and objective \( k \in \{1, \ldots, d\} \). Essentially, Lemma 2 implies that Assumption 2 holds under a large-deviations regime or under the conditions of the Central Limit Theorem, whenever the sample size sequence increases at a sufficiently fast rate. For additional explanation of Assumption 2 and Lemma 2.

**Lemma 2.** (see Wang et al. 2013, p. 13–14) Assumption 2 holds if one of the following two sets of conditions holds:

**C1.** (a) For all \( k \in \{1, \ldots, d\} \), the sequence of random variables \( \{\hat{G}_{k, m_{\nu}}(x) - g_k(x)\} \) is governed by a large-deviation principle with rate function \( I_{k, x}(s) \) (Dembo and Zeitouni 1998); (b) each \( I_{k, x}(s) \) is such that for any \( \epsilon > 0 \), \( \inf_{x \in \mathcal{X}, k \in \{1, \ldots, d\}} \min(I_{k, x}(\epsilon), I_{k, x}(-\epsilon)) = \eta > 0 \); and (c) the sequence of sample sizes \( \{m_{\nu}\} \) increases faster than logarithmically, that is, \( \limsup_{\nu \to \infty}(m_{\nu})^{-1}(\log \nu)^{1+\Delta_1} = 0 \) for some \( \Delta_1 > 0 \).

**C2.** (a) For all \( k \in \{1, \ldots, d\} \), a central limit theorem holds on the sequence of random variables \( \{\hat{G}_{k, m_{\nu}}(x)\} \) for each \( x \in \mathcal{X} \), that is, \( \sqrt{m_{\nu}}(\sigma_k(x))^{-1}(\hat{G}_{k, m_{\nu}}(x) - g_k(x)) \Rightarrow Z \), where \( \sigma_k(x) > 0 \) satisfies \( \sup_{x \in \mathcal{X}} \sigma_k^2(x) < \infty \), and (b) as \( \nu \to \infty \), \( \sup_y |F_{k, x, m_{\nu}}(y) - \Phi(y)| = O(1/\sqrt{m_{\nu}}) \) for all \( x \in \mathcal{X} \), where \( F_{k, x, m_{\nu}}(\cdot) \) denotes the cumulative distribution function of the random variable \( \sqrt{m_{\nu}}(\sigma_k(x))^{-1}(\hat{G}_{k, m_{\nu}}(x) - g_k(x)) \), and (c) the sequences of sample sizes \( \{m_{\nu}\} \) satisfies \( \limsup_{\nu \to \infty}(m_{\nu})^{-1}\nu^{2+\Delta_2} = 0 \) for some \( \Delta_2 > 0 \).

The primary implication of Assumptions 1 and 2 is the convergence of the estimated level sets into the true level sets, as described in the following Lemma 3. Before we present the lemma, recall that \( S(x, \alpha) = \cup_{k=1}^d S_k(x, \alpha_k) \) is finite, and define \( \hat{S}_{\nu}(x) := \cup_{k=1}^d \hat{S}_{k, \nu}(x) \) as the
set of all decision points estimated as being at least as good as \( x \) on at least one objective. A proof sketch for Lemma 3 appears in the Online Appendix.

**Lemma 3.** Under Assumptions 1 and 2,
(a) (see Wang et al. 2013, p. 15) for each \( k \in \{1, \ldots, d\} \) and any \( x \in X \), the sets \( \hat{S}_{k, \nu}(x) \) converge almost surely into the set \( S_k(x, \alpha_k) \), that is,
\[
P\{\hat{S}_{k, \nu}(x) \not\subseteq S_k(x, \alpha_k) \text{ i.o.}\} = 0;
\]
(b) the sets \( \hat{S}_{\nu}(x) \) converge almost surely into the sets \( S(x, \alpha) \) for any \( x \in X \), that is,
\[
P\{\hat{S}_{\nu}(x) \not\subseteq S(x, \alpha) \text{ i.o.}\} = 0.
\]

Finally, we need our last required assumption, Assumption 3, because our \( \varepsilon \)-placement and ALES completeness parameters rely on the estimated standard errors of the objective function values. Notice that Assumption 3 is implied under the conditions of Lemma 2.

In addition to Assumptions 1–3 discussed above, some of our results require additional structure on the true, unknown objective functions in Problem \( M_d \). We present these assumptions as Assumptions 4–6, and then we discuss their implications.

**Assumption 4.** For all \( x, x' \in X \), if \( g(x) = g(x') \), then \( x = x' \).

**Assumption 5.** There exists \( \kappa > 0 \) such that
\[
\min_{k \in \{1, \ldots, d\}} \inf \{|g_k(x) - g_k(x')| : x, x' \in X, x \neq x'\} > \kappa.
\]

**Assumption 6.** Given \( a \in [1, \infty) \), all \( N_a\)-LWEP’s are GEP’s and there exists exactly one \( N_a\)-LES that solves Problem \( M_d \) which is also the GES, \( \mathcal{E} \).

Assumption 4 ensures that two or more decision points in the feasible space do not map to the same point in the objective space. Under this assumption, the following Lemma 4 holds regarding the existence of an \( N_a\)-LES within each \( N_a\)-LWES. We present the result without proof; intuitively, it follows because Assumption 4 prevents the points in the \( N_a\)-LWES from having identical objective vector values. Thus the set must contain \( N_a\)-LEP’s, from which the \( N_a\)-LES can be constructed.

**Lemma 4.** Under Assumptions 1 and 4, given \( a \in (0, \infty] \), all \( N_a\)-LWES’s contain an \( N_a\)-LES; that is, for all \( \mathcal{W}_a \in \mathfrak{W}_a \), there exists \( \mathcal{L}_{a}^* \in \mathfrak{L}_a \) such that \( \mathcal{W}_a \supseteq \mathcal{L}_{a}^* \).

Finally, we remark on Assumptions 5 and 6. Assumption 5, which subsumes Assumption 4, ensures that each feasible point is distinguishable on each objective. Under this assumption, every \( N_a\)-LWEP is an \( N_a\)-LEP, and every \( N_a\)-LWES is an \( N_a\)-LES. Assumption 6 is required for the convergence of R-P\( \varepsilon \), and stipulates that every \( N_a\)-LWEP is a GEP, and there exists
Cooper, Hunter, and Nagaraj: Bi-objective SO on Integer Lattices using the \(\varepsilon\)-Constraint Method in an RA Framework

8.2. Convergence of R-P\(\varepsilon\)RLE and R-MinRLE

We now consider the convergence of the algorithms R-P\(\varepsilon\)RLE and R-MinRLE under the regularity conditions discussed in the previous section. These algorithms invoke RLE to guarantee that the set of points returned at the end of each RA iteration is an ALES. Theorem 1 and its proof are presented for \(d \geq 2\) since R-MinRLE converges for two or more objectives; the proof appears in the Online Appendix. Indeed, given appropriate parameter values, the proof of convergence of Theorem 1 holds for any RA algorithm that invokes RLE as the last step in the sample-path solver.

Theorem 1. Let Assumptions 1–3 hold. For any neighborhood size \(a \in [1, \infty)\), initial point \(x_0 \in \mathcal{X}\), \(\varepsilon\)-placement rule, and completeness parameter \(\beta_\delta \in (0, \infty]\), R-P\(\varepsilon\)RLE (\(d = 2\)) and R-MinRLE (\(d \geq 2\)) generate a sequence of estimated solutions \(\{\hat{A}_\nu\}\) such that

(a) \(\{\hat{A}_\nu\}\) converges into an \(N_a\)-LWES almost surely, that is, \(\exists \mathcal{W}_a \in \mathfrak{W}_a\) such that \(P\{\hat{A}_\nu \not\subseteq \mathcal{W}_a \text{ i.o.}\} = 0\);

(b) under Assumption 4, \(\{\hat{A}_\nu\}\) contains an \(N_a\)-LES infinitely often almost surely, that is, \(\exists \mathcal{L}_a \in \mathfrak{L}_a\) such that \(P\{\mathcal{L}_a \not\subseteq \hat{A}_\nu \text{ i.o.}\} = 0\);

(c) under Assumption 5, \(\{\hat{A}_\nu\}\) converges to an \(N_a\)-LES almost surely, that is, \(\exists \mathcal{L}_a \in \mathfrak{L}_a\) such that \(P\{\hat{A}_\nu \neq \mathcal{L}_a \text{ i.o.}\} = 0\);

(d) under Assumptions 5 and 6, \(\{\hat{A}_\nu\}\) converges to the GES almost surely, that is, \(P\{\hat{A}_\nu \neq \mathcal{E} \text{ i.o.}\} = 0\).

Theorem 1 presents a series of convergence results that require increasingly stringent assumptions on the underlying Problem \(M_d\). At a minimum, under our required Assumptions 1–3, R-P\(\varepsilon\)RLE and R-MinRLE converge into an \(N_a\)-LWES almost surely. If more restrictive assumptions hold, our algorithms converge to an \(N_a\)-LES almost surely.

8.3. Convergence of R-P\(\varepsilon\)

We now consider the convergence of R-P\(\varepsilon\), which does not rely on RLE to certify that each RA iteration returns an ALES. Since R-P\(\varepsilon\) is an algorithm designed for exactly two objectives, henceforth in this section, we let \(d = 2\).

To show the convergence of R-P\(\varepsilon\) in Theorem 2, first, we notice that for each objective \(k \in \{1, 2\}\), the sequence of sample-path \(N_a\)-local minimizers produced by \(P\varepsilon\) in Step 1 across exactly one \(N_a\)-LES, which is also the GES. Under this assumption, the R-P\(\varepsilon\) algorithm cannot “get stuck” by returning parts of different \(N_a\)-LES’s.
RA iterations, defined as \( \{M_\nu, \nu = 1, 2, \ldots\} \) where \( M_\nu = \{X_{1,m_\nu}^{\text{min}}, X_{2,m_\nu}^{\text{min}}\} \) for all \( \nu = 1, 2, \ldots \), converges into the set of all true \( N_\nu \)-local minimizers of objective \( g_k \) over the feasible set \( X \), \( M_\nu^* \subseteq X \), almost surely as \( \nu \to \infty \). Since this result, presented in Theorem 5, follows almost directly from Wang et al. (2013) under Assumptions 1 and 2, we do not provide a proof.

The proof of convergence of R-P\( \varepsilon \), which requires our most restrictive assumptions on the underlying Problem \( M_2 \), appears in the Online Appendix.

**Lemma 5.** Under Assumptions 1 and 2, for \( d = 2 \), any neighborhood size \( a \in [1, \infty) \), initial point \( x_0 \in X \), and \( \varepsilon \)-placement rule, across RA iterations, \( P\varepsilon \) Step 1 generates a sequence of sample-path \( N_\nu \)-local minimizers \( \{M_\nu\} \) that converges into \( M_\nu^* \) almost surely, that is, \( \mathbb{P}\{M_\nu \not\subseteq M_\nu^* \text{ i.o.}\} = 0 \).

**Theorem 2.** Under Assumptions 1–6, for \( d = 2 \), any neighborhood size \( a \in [1, \infty) \), initial point \( x_0 \in X \), and \( \varepsilon \)-placement rule specified by \( \beta_\varepsilon \in (0, \infty) \), R-P\( \varepsilon \) generates a sequence of estimated solutions \( \{\hat{A}_\nu\} \) that converges almost surely to the global efficient set \( E \), in the sense that \( \mathbb{P}\{\hat{A}_\nu \neq E \text{ i.o.}\} = 0 \).

### 8.4. Sampling Efficiency

Finally, we provide a result on the sampling efficiency of our algorithms. This result provides insight into how to set the algorithm parameter values in §9 to achieve exponential convergence. In Theorem 3, let \( X^w \) denote the set of all \( N_\nu \)-LWEP’s for Problem \( M_d \), and let \( X^w_\nu \) denote the set of all sample-path \( N_\nu \)-LWEP’s on the \( \nu \)th RA iteration. Further, let \( \hat{A}_\nu \) denote the solution returned on the \( \nu \)th RA iteration of R-P\( \varepsilon \)RLE (\( d = 2 \)), R-P\( \varepsilon \) (\( d = 2 \)), or R-MINRLE (\( d \geq 2 \)) for any \( x_0 \in X \), \( \varepsilon \)-placement rule \( \beta_\varepsilon \in (0, \infty) \), and completeness parameter \( \beta_\delta \in (0, \infty) \). A proof of Theorem 3 appears in the Online Appendix.

**Theorem 3.** Let the neighborhood size \( a = 1 \) and suppose the feasible set \( X \subseteq \mathbb{Z}^q \) is finite with \( \max_{k \in \{1, \ldots, d\}} \sup_{x \in X} \sigma_k^2(x) < \infty \). For all objectives \( k \in \{1, \ldots, d\} \), let the sequence of random variables \( \{G_{k,m_\nu}(x) - g_k(x)\} \) be governed by a large-deviation principle with rate function \( I_{k,x}(s) \), as stipulated in Lemma 2. Then the following hold:

(a) \( \mathbb{P}\{X^w_\nu \not\subseteq X^w\} = O(e^{-\gamma m_\nu}) \) for some \( \gamma > 0 \).

(b) If the sequence of sample sizes increases to infinity at least linearly in R-P\( \varepsilon \)RLE, R-P\( \varepsilon \), and R-MINRLE, that is, if \( \limsup_{\nu \to \infty} m_\nu^{-1} \nu < \infty \), then

(i) \( \mathbb{P}\{\hat{A}_\nu \not\subseteq X^w\} = O(e^{-\gamma m_\nu}) \) for some \( \gamma > 0 \),

(ii) under Assumption 5 and 6, \( \mathbb{P}\{\hat{A}_\nu \neq E\} = O(e^{-\gamma m_\nu}) \) for some \( \gamma > 0 \).
9. Algorithm Parameters and Implementation

We now discuss the details of algorithm implementation and the choice of four algorithm parameters. First, for algorithmic efficiency, everywhere the oracle is called at a point \( \mathbf{x} \) with sample size \( n = m_\nu \) within an RA iteration, we assume the triple \((\mathbf{x}, G_n(\mathbf{x}), \hat{s.e.}(G_n(\mathbf{x}))\)) is stored and is made available to all relevant subroutines within an RA iteration. Thus everywhere a candidate ALES is passed between functions, we assume the estimated objective function values of the neighborhood points are made available to all relevant subroutines, especially to RLE. This practice enhances efficiency by removing the need to re-sample neighborhood points when checking whether a candidate ALES is truly an ALES. All stored points visited and simulation replications obtained is cleared between RA iterations.

While all of our definitions and algorithms allow for a flexible neighborhood specified by the parameter \( a \), as noted by Wang et al. (2013), there exists a tension between the relatively faster convergence enabled by \( a = 1 \) and the certification of the local solution as optimal in a larger neighborhood. By default, we set \( a = 1 \). For larger \( a \) and for high-dimensional feasible spaces, Wang et al. (2013) remark that the neighborhood enumeration NE routine inside SPLINE may be modified to return only a better point, rather than the best point in the neighborhood. Further, the algorithms GETNCN and REMOVENONLWEP inside RLE can be modified to return only a subset of the non-conforming points encountered in the neighborhood or the first sample-path \( N_\alpha \)-LWEP encountered, respectively, rather than all such points. These modifications do not affect the convergence properties of the algorithm.

We now discuss the default sample size sequences and relaxation parameters. By default, we set the monotone-increasing sample size sequence as
\[
m_\nu = \lceil 2 \times 1.1^\nu \rceil \quad \text{for all } \nu \geq 1.
\]
This sequence satisfies the requirements of Lemma 2 and Theorem 3 in §8. To ensure every search terminates in finite time, but that for large enough \( \nu \), the sample size limit \( b_\nu \) will not be reached, we set the sequence \( b_\nu = \lceil 8 \times 1.2^\nu \rceil \) for all \( \nu \geq 1 \). Notice that each search we conduct inside \( P_\varepsilon \) and RLE gets a fresh limiting sample size.

We control the placement of the \( \varepsilon \) values in \( P_\varepsilon \) and the completeness of the ALES returned by RLE using the parameters \( \beta_\varepsilon \) and \( \beta_\delta \). Since our algorithm converges for a wide variety of \( \beta_\varepsilon \) and \( \beta_\delta \) values, there is significant flexibility in setting these parameters. By default, we set \( \beta_\varepsilon = \beta_\delta = 0.5 \) in all of our algorithms. Thus we search for new sample-path \( N_1 \)-LEP’s that are more than one standard error away from the ones we have in every objective. We numerically explore the effect of these parameters on our algorithm performance in §10.
10. Numerical Experiments

First, we conduct numerical experiments that compare our main algorithm, \( R-P_\varepsilon RLE \), to the benchmark algorithm \( R-M in RLE \) and the current state-of-the-art, \( MO-COMPASS \). Then, we explore the performance of \( R-P_\varepsilon RLE \) across a variety of \( \beta = (\beta_\varepsilon, \beta_\delta) \) values.

10.1. Algorithm Performance with Default Parameter Values

We compare the performances of \( R-P_\varepsilon RLE \), \( R-M in RLE \), and \( MO-COMPASS \) on three increasingly complicated test problems. We configure our numerical experiments as follows. In each independent run of an algorithm, we use an initial point \( x_0 \) that is generated uniformly from the feasible set \( \mathcal{X} \), which is finite in our test problems. Within an algorithm run, we use CRN across points visited. \( R-P_\varepsilon RLE \) and \( R-M in RLE \) use the default parameter values described in the previous section. We configure \( MO-COMPASS \), including the simulation allocation rule (SAR), as close as possible to the settings used in Li et al. (2015a, p. 10). In the quantile plots that follow, the algorithm performance at each value of the total simulation budget \( t \) is dependent on its previous performance.

10.1.1. Test Problem A

Our first test problem is a modified version of a problem that appears in Kim and Ryu (2011). We define Problem \( T_A \) as

\[
\text{Problem } T_A: \quad \text{minimize}_{x \in \mathcal{X}} \begin{cases} 
  g_1(x) = \mathbb{E}[(x_1/10 - 2\xi_1)^2 + (x_2/10 - \xi_2)^2] \\
  g_2(x) = \mathbb{E}[x_1^2/100 + (x_2/10 - 2\xi_3)^2] 
\end{cases}
\]

where \( \mathcal{X} = \tilde{\mathcal{X}}_{A1} \times \tilde{\mathcal{X}}_{A2} \) and \( \tilde{\mathcal{X}}_{A1} = \tilde{\mathcal{X}}_{A2} = \{0, 1, 2, \ldots, 50\} \), \( |\mathcal{X}| = 2601 \), and \( \xi_i \) are independent chi-squared random variables with one degree of freedom so that \( \mathbb{E}[\xi_i] = 1 \) and \( \mathbb{V}(\xi_i) = 2 \) for all \( i \in \{1, 2, 3\} \). Thus the random objective values returned by the simulation oracle are independent for all \( x \in \mathcal{X} \). A picture of Problem \( T_A \) appears in Figure 3.

Problem \( T_A \) satisfies Assumptions 1–6; it has one \( \mathcal{N}_1 \)-LES which equals the GES. Therefore for this problem, by Theorem 1 Part (d), our algorithms converge to the GES as the total simulation work done, denoted by \( t \), goes to infinity. Let \( \hat{A}(t) \) denote the set returned by an algorithm after expending a total of \( t \) simulation replications. We measure the solution quality returned by each algorithm using sample quantiles of the coverage error. The coverage error is defined by Hunter et al. (2018) as the Hausdorff distance between the image of the set returned by the algorithm and the image of the true efficient set as a function of \( t \), \( d_H(g(\hat{A}(t)), g(\mathcal{E})) \). Figure 4 shows the sample quantiles of the coverage error for 1,000 independent runs each of \( R-P_\varepsilon RLE \), \( R-M in RLE \), and \( MO-COMPASS \) on Problem \( T_A \).
Figure 3: Problem $T_A$: Black circles represent points in the only $N_1$-LES which is also the GES (left) and their image (right).

Figure 4: Problem $T_A$: Sample quantiles (0.25, 0.50, 0.75) of the coverage error across 1,000 independent runs.

Figure 4 shows that R-$P\varepsilon$RLE and R-MinRLE out-perform MO-COMPASS on Problem $T_A$. The performances of R-$P\varepsilon$RLE and R-MinRLE are similar, with R-$P\varepsilon$RLE performing slightly better for lower simulation budgets $t$.

10.1.2. Test Problem B Our second test problem is a modified version of a test problem that appears in Ryu and Kim (2014). We define Problem $T_B$ as

$$\text{Problem } T_B: \quad \text{minimize}_{x \in \mathcal{X}} \begin{cases} g_1(x) &= \mathbb{E} [\xi_1 h_1(x_1)] \\ g_2(x) &= \mathbb{E} [\xi_1 \xi_2 f(x_2) h_2(h_1(x_1), f(x_2))] \end{cases}$$

where $\mathcal{X} = \tilde{X}_{B1} \times \tilde{X}_{B2}$ and $\tilde{X}_{B1} = \tilde{X}_{B2} = \{0, 1, \ldots, 100\}$, $|\mathcal{X}| = 10, 201$, $h_1(x_1) = 4x_1/100$, and $h_2(h_1, f)$ and $f(x_2)$ are defined as

$$h_2(h_1, f) = \begin{cases} 1 - (h_1/f)^{\alpha} & \text{if } h_1 \leq f, \\ 0 & \text{otherwise} \end{cases}$$

$$f(x_2) = \begin{cases} 4 - 3 \exp\left(-\frac{(x_2-20)^2}{2}\right) & \text{if } 0 \leq x_2 \leq 40, \\ 4 - 2 \exp\left(-\frac{(x_2-70)^2}{20}\right) & \text{if } 40 < x_2 \leq 100; \end{cases}$$

and $\alpha = 0.25 + 3.75(f(x_2) - 1)$. As in the previous test problem, $\xi_1$ and $\xi_2$ are independent chi-squared random variables with one degree of freedom. Unlike in Problem $T_A$, Problem $T_B$ has dependence between the random objective function values returned by the simulation oracle. A picture of Problem $T_B$ appears in Figure 5.

Problem $T_B$ has two $N_1$-LWES’s, one of which is the GWES. Since Problem $T_B$ has GWEP’s that are not also GEP’s, it satisfies only Assumptions 1–4. By Theorem 1 Parts (a) and (b), our algorithms converge into an $N_1$-LWES almost surely, and contain an $N_1$-LES almost surely. Nevertheless, for this problem, we use the local coverage error as our solution quality metric (Hunter et al. 2018). In our context, the local coverage error is the Hausdorff...
distance from the set $g(\hat{A}(t))$ to the nearest $N_1$-LPS as a function of the total simulation work done, $\min_{\mathcal{L}_1 \in \mathcal{L}_1} d_H(g(\hat{A}(t)), g(\mathcal{L}_1))$. This metric penalizes all algorithms for returning the points that are GWES members but not GES members, which may not be distinguishable with finite sample size. Figure 6 shows the sample quantiles of the local coverage error for 1,000 independent runs each of R-PεRLE, R-MINRLE, and MO-COMPASS on Problem $T_B$.

Figure 6 shows that R-PεRLE out-performs both R-MINRLE and MO-COMPASS on Problem $T_B$. R-MINRLE eventually out-performs MO-COMPASS, but initially suffers from high variance in its performance. We believe this behavior occurs because R-MINRLE crawls from the “outside in,” and the guaranteed sample path $N_1$-local minimizers on each objective may be members of the GWES and not the GES. Also, R-MINRLE may not retrieve the “middle” of the $N_1$-LPS until the sample sizes become large enough that the completeness function values are small enough for RLE to crawl there. Thus R-PεRLE’s ability to retrieve the middle of the $N_1$-LPS is likely a crucial aspect of its speedy convergence in Problem $T_B$.

10.1.3. Test Problem C Our last test problem, Problem $T_C$, is also a modified version of a test problem that appears in Ryu and Kim (2014). We define Problem $T_C$ as

$$\text{Problem } T_C: \text{ minimize}_{x \in \mathcal{X}} \begin{cases} g_1(x) = \mathbb{E} \left[ \sum_{i=1}^{2} -10\xi_i \exp \left\{ -0.2 \sqrt{x_i^2 + x_{i+1}^2} \right\} \right] \\ g_2(x) = \mathbb{E} \left[ \sum_{i=1}^{3} \xi_i |x_i|^{0.8} + 5\sin^3(x_i) \right] \end{cases}$$

where $\mathcal{X} = \tilde{\mathcal{X}}_{C1} \times \tilde{\mathcal{X}}_{C2} \times \tilde{\mathcal{X}}_{C3}$, $\tilde{\mathcal{X}}_{Ci} = \{-5, -4.5, -4.0, -3.5, \ldots, 5\}$ for all $i \in \{1, 2, 3\}$, $|\mathcal{X}| = 9,261$, and $\xi_1, \xi_2$, and $\xi_3$ are independent chi-squared random variables with one degree of freedom so that $\mathbb{E} [\xi_i] = 1$ and $\mathbb{V} (\xi_i) = 2$ for all $i \in \{1, 2, 3\}$. Like Problem $T_B$, Problem $T_C$ has dependence between the random objective function values returned by the simulation.
oracle. Problem $T_C$ appears in Figure 7. We map Problem $T_C$ to an integer lattice so that the $N_1$-neighborhood corresponds to points within distance 0.5 in the original feasible space.

Problem $T_C$ has multiple feasible points that map to the same objective vector value. Therefore Problem $T_C$ only satisfies Assumptions 1–3. By Theorem 1 Part (a), our algorithm returns a solution that converges into an $N_1$-LWES w.p.1., with no guarantees on completeness. Nevertheless, we use the local weakly coverage error as our solution quality metric, which we define as
\[
\min_{W_1 \in W_1} d_H(g(\hat{A}(t)), g(W_1)).
\]
Since all $N_1$-LES’s are also $N_1$-LWES’s, this metric is less stringent than local coverage error. Algorithm performances based on the local weakly coverage error, calculated across a collection of 516 unique $N_1$-LWES’s, appear in Figure 8. Our method for locating the $N_1$-LWES’s appears in the Online Appendix.

Figure 8 shows that both R-PrεRLE and R-MinRLE out-perform MO-COMPASS on Problem $T_C$. In many of the $N_1$-LWES’s, the $N_1$-LWES members are not neighbors. Thus the $N_1$-LWES members may be far away from each other in the feasible space, and often are isolated, as seen in Figure 7. We believe the relative efficiency of R-PrεRLE and R-MinRLE occurs because RLE crawls to find a sample-path $N_1$-LWEP that completes the sample-path $N_1$-LWES, even if the required sample-path $N_1$-LWEP is far away in the feasible space. MO-COMPASS operates by updating a region of the feasible space called the most promising region. We suspect that the isolated, scattered nature of the $N_1$-LWES members results in a low probability that these far-away $N_1$-LWEP’s are contained in the most promising region.

10.2. R-PrεRLE Performance Across a Range of $\beta$ Values
We explore R-PrεRLE’s performance on our test problems across a variety of $\beta = (\beta_\varepsilon, \beta_\delta)$ parameter values. Recall that for Pε, smaller $\beta_\varepsilon$ values result in solving fewer sample-path $\varepsilon$-constraint problems, and larger $\beta_\varepsilon$ values correspond to solving more sample-path
\( \min_{\mathbf{L}_1 \in \mathbf{L}_1} d_H(\hat{g}(\mathbf{A}(t)), g(\mathbf{L}_1)) \)

\( R-P_\varepsilon \)

\( R-P_\varepsilon \) \( R_{LE}, \beta \delta = 0.5 \)

\( d_H(\hat{g}(\mathbf{A}(t)), g(\mathbf{E})) \)

\( R-P_\varepsilon \) \( R_{LE}, \beta \varepsilon = 0.5 \)

\varepsilon\)-constraint problems. For the RLE algorithm, smaller \( \beta \delta \) implies less crawling and a less-complete ALES, and larger \( \beta \delta \) corresponds to more crawling and a more-complete ALES. Across 1,000 independent runs of \( R-P_\varepsilon \) or \( R-P_\varepsilon \) \( R_{LE} \) on Problems \( T_A, T_B, \) and \( T_C \), Figures 9, 10, and 11 show the sample quantiles of the respective coverage errors at the total simulation budget of \( t = 0.4 \times 10^6 \) (corresponding to the first \( t \)-axis tick mark in Figures 4, 6, and 8) across a variety of parameter settings. Each independent run uses CRN across the \( \beta \) values.

At total simulation budget \( t = 0.4 \times 10^6 \) on Problem \( T_A \), there seems to be a “sweet spot” for setting \( \beta \varepsilon \) in the interval (0.2, 0.4), as seen in the left and center panels of Figure 9. Relative to our sampling error, \( \beta \varepsilon < 0.2 \) causes the algorithm to find too few sample-path \( \mathcal{N}_1 \)-LWEP’s, while \( \beta \varepsilon > 0.4 \) cause the algorithm to find too many. Given that \( \beta \varepsilon = 0.5 \), the \( R-P_\varepsilon \) \( R_{LE} \) performance in the right panel of Figure 9 is fairly robust to different values of \( \beta \delta \). Notice that with \( \beta \varepsilon = 0.5 \), for small values of \( \beta \delta \), \( R-P_\varepsilon \) and \( R-P_\varepsilon \) \( R_{LE} \) return similar sets.

On Problem \( T_B \), however, solving more \( \varepsilon \)-constraint problems and crawling more in RLE seems to improve algorithm performance. We suspect that in this problem, correlation between the objectives and using CRN implies each sample-path problem is similar to the true problem. In this scenario, the ordering of the points in the sample-path problem is simi-
Figure 11  Problem $T_C$: Sample quantiles (0.25, 0.50, 0.75) of the local weakly coverage error at $t = 0.4 \times 10^6$ across 1,000 independent runs of $R-P_\varepsilon$ (left), $R-P_{\varepsilon RLE}$, $\beta_\delta = 0.5$ (center), $R-P_{\varepsilon RLE}$, $\beta_\varepsilon = 0.5$ (right).

lar to the ordering of the points in the true problem with high probability, except among the GWEP’s. Thus finding more $N_1$-LWEP’s in $P_\varepsilon$ and crawling more in RLE is usually better.

Problem $T_C$ is a difficult problem for which $R-P_\varepsilon$ is not guaranteed to converge. In the left panel of Figure 11, like in the left panel of Figure 9, $R-P_\varepsilon$ exhibits $u$-shaped behavior as a function of $\beta_\varepsilon$. However, the center and right panels of Figure 11 tell an interesting story for $R-P_{\varepsilon RLE}$. It seems that in Problem $T_C$, it is best to solve few $\varepsilon$-constraint problems (smaller $\beta_\varepsilon$) and let RLE do the work of finding the disconnected $N_1$-LWES members, with the sweet spot for $\beta_\delta$ shown in the right panel of Figure 11. Interestingly, that more effort should be expended in RLE and less effort in $P_\varepsilon$ explains the good performance of $R$-MinRLE in Figure 8. Finally, we remark that across all the problems, without prior knowledge of the problem structure, our default $\beta_\varepsilon$ and $\beta_\delta$ values seem reasonable.

11. Concluding Remarks

We propose $R-P_{\varepsilon RLE}$, a new, provably-convergent algorithm for bi-objective SO on integer lattices. We also propose $R$-MinRLE as a benchmark algorithm for MOSO on integer lattices with two or more objectives. $R-P_{\varepsilon RLE}$ out-performs both $R$-MinRLE and the current state-of-the-art algorithm, MO-COMPASS, on our test problems. This work points to a family of RA algorithms for MOSO on integer lattices that employ RLE for sample-path certification of an ALES, where the convergence guarantees are provided by Theorem 1.

Acknowledgments

The second and third authors were supported in part by the National Science Foundation through grant CMMI-1554144. An early version of this work appears in Cooper et al. (2017).

References


Online Appendices for
Bi-objective Simulation Optimization on Integer Lattices
using the Epsilon-Constraint Method
in a Retrospective Approximation Framework

Kyle Cooper
School of Industrial Engineering, Purdue University and Tata Consultancy Services, coope149@purdue.edu

Susan R. Hunter
School of Industrial Engineering, Purdue University, susanhunter@purdue.edu

Kalyani Nagaraj
School of Industrial Engineering & Management, Oklahoma State University, kalyani.nagaraj@okstate.edu

A. Proof Sketch of Lemma 3

Proof Sketch. The proof of Lemma 3 Part (a) is provided in Wang et al. (2013, p. 15) and follows from the first Borel-Cantelli lemma (Billingsley 1995, p. 59). By Lemma 3 Part (a), for each objective \( k \in \{1,\ldots,d\} \), there exists \( \tilde{\nu}_k \), dependent on \( \alpha, x \), and the random realization, such that for all \( \nu \geq \tilde{\nu}_k \), \( \hat{S}_{k,\nu}(x) \subseteq S_k(x,\alpha_k) \) w.p.1. Let \( \tilde{\nu} := \max_k \{\tilde{\nu}_k\} \), so that for all \( \nu \geq \tilde{\nu} \), \( \hat{S}_{\nu}(x) \subseteq S(x,\alpha) \) w.p.1. □

B. Proof of Theorem 1

Proof of Theorem 1 Part (a). For every \( \nu \), R-P\( \varepsilon \)RLE and R-M\( \varepsilon \)RLE return a set \( \hat{A}_\nu \) in finite time. Thus both algorithms produce an infinite sequence of solutions \( \{\hat{A}_\nu\} \). Further, notice that R-P\( \varepsilon \)RLE and R-M\( \varepsilon \)RLE never return a set \( \hat{A}_\nu \) containing a point whose estimated objective vector is dominated by \( \bar{G}_{m,nu}(x_0) \) (see Algorithm 5, RLE Steps 1 and 14). Now consider the union of the level sets corresponding to the starting point \( x_0, S(x_0,\alpha) \). By Lemma 3, there exists \( \tilde{\nu} \) such that for all \( \nu \geq \tilde{\nu} \), \( \hat{A}_\nu \subseteq \hat{S}_\nu(x_0) \subseteq S(x_0,\alpha) \) w.p.1. Since \( S(x_0,\alpha) \) is finite, then any sequence of estimated efficient points \( \{X^*_\nu : X^*_\nu \in \hat{A}_\nu \text{ for all } \nu = 1,2,\ldots\} \) is bounded w.p.1. Using an argument similar to that in Wang et al. (2013, Theorem 5.4, p. 15), we now prove that \( \{\hat{A}_\nu\} \) converges into an \( \mathcal{N}_a \)-LWES w.p.1.

Since \( S(x_0,\alpha) \) is finite and \( a \in (0,\infty) \), then \( \mathcal{N}_a(S(x_0,\alpha)) \cap \mathcal{X} \) is also finite. Thus for all \( k \in \{1,\ldots,d\} \), \( G_{k,nu}(\cdot) \) uniformly converges to \( g_k(\cdot) \) w.p.1 as \( \nu \rightarrow \infty \) on the set \( \mathcal{N}_a(S(x_0,\alpha)) \cap \mathcal{X} \). Let the set \( \mathcal{D}_k := \{(x,x') : x,x' \in \mathcal{N}_a(S(x_0,\alpha)) \cap \mathcal{X}, g_k(x') \neq g_k(x)\} \) be the set of all pairs
of feasible points in the level set neighborhood that have different true objective function values on objective \( k \), and let \( \kappa_1 = \min_{k \in \{1, \ldots, d\}} \inf_{\mathcal{D}_k} |g_k(x) - g_k(x')| > 0 \) be the smallest difference in objective values across these pairs; Assumption 1 implies \( \kappa_1 > 0 \). Then w.p.1, there exists \( \nu' \) (dependent on neighborhood size \( a \), initial point \( x_0 \), the constants \( \kappa_1 \) and \( \alpha \), and the random realization) such that for all \( \nu \geq \nu' \), \( \max_{k \in \{1, \ldots, d\}} |\bar{G}_{k,m_\nu}(x) - g_k(x)| < \kappa_1/4 \) for all \( x \in \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \). Since \( \beta_\Delta \in (0, \infty] \), the ALES completeness function \( \hat{\delta}_k() = \hat{f}_k() / m_\nu^\beta_\Delta \) uniformly converges to zero w.p.1 on the finite set \( \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \) as \( \nu \to \infty \) for all \( k \in \{1, \ldots, d\} \). Then w.p.1, there exists \( \nu'' \) (dependent on the same quantities as \( \nu' \) and dependent on \( \beta_\Delta \)) such that for all \( \nu \geq \nu'' \), \( \max_{k \in \{1, \ldots, d\}} |\hat{\delta}_k(x)| < \kappa_1/4 \) for all \( x \in \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \).

Henceforth, let \( \nu \geq \max\{\nu', \nu''\} \). Combining the above results, for all \( x \in \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \), \( \max_{k \in \{1, \ldots, d\}} |\bar{G}_{k,m_\nu}(x) - \hat{\delta}_k(x) - g_k(x)| < \kappa_1/2 \) w.p.1. Thus for all \( k \in \{1, \ldots, d\} \) and for all \( x, x' \in \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \), the following hold:

R1. if \( g_k(x) < g_k(x') \), then \( \bar{G}_{k,m_\nu}(x) + \hat{\delta}_k(x) < \bar{G}_{k,m_\nu}(x') - \hat{\delta}_k(x') \) w.p.1;

R2. if \( \bar{G}_{k,m_\nu}(x) + \hat{\delta}_k(x) \leq \bar{G}_{k,m_\nu}(x') - \hat{\delta}_k(x') \), then \( g_k(x) \leq g_k(x') \) w.p.1.

Further, for all \( x, x' \in \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \), if \( \bar{G}_{m_\nu}(x) \not\subseteq \bar{G}_{m_\nu}(x') \), then \( \exists k \in \{1, \ldots, d\} \) such that \( \bar{G}_{k,m_\nu}(x') < \bar{G}_{k,m_\nu}(x) \), implying that \( g_k(x') \leq g_k(x) \) w.p.1. This result, along with results R1 and R2 above, imply that for all \( x, x' \in \mathcal{N}_a(S(x_0, \alpha)) \cap \mathcal{X} \),

R3. if \( \bar{G}_{m_\nu}(x) \not\subseteq \bar{G}_{m_\nu}(x') \), then \( g(x) \not\subseteq g(x') \) w.p.1;

R4. if \( \bar{G}_{m_\nu}(x) + \delta(x) \subseteq \bar{G}_{m_\nu}(x') - \delta(x) \), then \( g(x) \subseteq g(x') \) w.p.1.

Now let \( \nu \geq \max\{\nu', \nu''\} \). Then by Lemma 3, the set of decision points \( \hat{A}_\nu \) returned by each algorithm lie in \( S(x_0, \alpha) \) w.p.1. We now consider all parts of the definition of an ALES (Definition 7). First, Algorithm 5 ensures no points in \( \bar{G}_n(\hat{A}_\nu) \) dominate other points in \( \bar{G}_n(\hat{A}_\nu) \) (RLE Steps 1 and 14). Thus result R3 above implies that no points in \( g(\hat{A}_\nu) \) strictly dominate other points in \( g(\hat{A}_\nu) \). Second, Algorithm 5 ensures each point in \( X_\nu \in \hat{A}_\nu \) is a sample-path \( \mathcal{N}_a \)-LWEP (e.g., RLE Steps 2, 5, 10, and 15). Thus result R3 above implies that all points in \( \hat{A}_\nu \) are \( \mathcal{N}_a \)-LWEP’s. Third, Algorithm 5 ensures that the NCN of \( \hat{A}_\nu \) is empty (RLE Steps 2 and 15). Then applying results R3 and R4 above, for all \( X \in \mathcal{N}_a(\hat{A}_\nu) \cap \mathcal{X} \),

(i) \( \exists X_\nu \in \hat{A} \) such that \( g(X_\nu) \leq g(X) \) w.p.1, or

(ii) \( \exists X_\nu \in \hat{A} \) such that \( g(X) \leq g(X_\nu) \) and \( \bar{G}_{m_\nu}(X_\nu) - \delta(X_\nu) \subseteq \bar{G}_{m_\nu}(X) + \delta(X) \) w.p.1, which happens with probability zero unless \( g(X) = g(X_\nu) \), or

(iii) employing the complements of the previous two conditions, \( \forall X_\nu \in \hat{A} \), \( g(X_\nu) \not\subseteq g(X) \not\subseteq g(X_\nu) \), and \( \exists X_\nu \in \hat{A} \) such that \( \bar{G}_{m_\nu}(X_\nu) - \delta(X_\nu) \subseteq \bar{G}_{m_\nu}(X) + \delta(X) \) or
$G_{m_\nu}(X) - \delta(X) \leq G_{m_\nu}(\hat{X}_\nu) + \delta(\hat{X}_\nu^w)$ w.p.1, which happens with probability zero. To see this, let $X \in N_a(\hat{A}_\nu) \cap \mathcal{X}$ and notice that the condition $\forall X_w^\nu \in \hat{A}$, $g(X_w^\nu) \neq g(X) \neq g(X_w^\nu)$ w.p.1 implies that $\forall X_w^\nu \in \hat{A}$, $\exists k, k_2 \in \{1, \ldots, d\}$ such that $g_{k_1}(X_w^\nu) > g_{k_1}(X)$ and $g_{k_2}(X_w^\nu) < g_{k_2}(X)$ w.p.1, and hence by result R1 above, $G_{k_1,m_\nu}(X_w^\nu) - \delta_{k_1}(X_w^\nu) > G_{k_1,m_\nu}(X) + \delta_{k_1}(X)$ and $G_{k_2,m_\nu}(X) - \delta_{k_2}(X) > G_{k_2,m_\nu}(X_w^\nu) + \delta_{k_2}(X_w^\nu)$ w.p.1.

Therefore when $\nu \geq \max\{\tilde{\nu}, \nu', \nu''\}$, R-P-$\varepsilon$RLE and R-MINRLE certify that w.p.1, all points in $\hat{A}_\nu$ are $N_a$-LWEP’s, no points in $g(\hat{A}_\nu)$ strictly dominate other points in $g(\hat{A}_\nu)$, and for all $X \in N_a(\hat{A}_\nu) \cap \mathcal{X}$, $\exists X_w^\nu \in \hat{A}$ such that $g(X_w^\nu) \leq g(X)$. Thus by Definition 5, $\hat{A}_\nu$ is an $N_a$-LWES w.p.1. Further, each $\hat{A}_\nu$ is such that there does not exist a pair of points $X_{\nu-1}^s \in \hat{A}_{\nu-1}$ and $X_{\nu}^s \in \hat{A}_\nu$ such that $g(X_{\nu-1}^s) < g(X_{\nu}^s)$ w.p.1. (The second part follows because we carry forward the points from $\hat{A}_{\nu-1}$ into the $\nu$th iteration, and we ensure that no points in $G_n(\hat{A}_\nu)$ dominate other points in $G_n(\hat{A}_\nu)$ in Algorithm 5, RLE Steps 1 and 14.) Therefore if $\nu \geq \max\{\tilde{\nu}, \nu', \nu''\}$, there exists $W_a \in \mathfrak{W}_a$ such that R-P-$\varepsilon$RLE and R-MINRLE returns $\hat{A}_\nu \subseteq W_a \subseteq S(x_0, \alpha)$ w.p.1.

Proof of Theorem 1 Part (b). Let $\nu \geq \max\{\tilde{\nu}, \nu', \nu''\}$. By the proof of Theorem 1 Part (a), $\hat{A}_\nu$ is an $N_a$-LWES w.p.1. Under Assumption 4, by Lemma 4, there exists $L_a \in \mathfrak{L}_a$ such that $\hat{A}_\nu \supseteq L_a$ w.p.1, and the result holds.

Proof of Theorem 1 Part (c). Assumption 5 implies that all $N_a$-LWES’s are $N_a$-LES’s. Thus $\mathfrak{W}_a = \mathfrak{L}_a$, and the result follows from Theorem 1 Parts (a) and (b).

Proof of Theorem 1 Part (d). Assumptions 5 and 6 imply that the $N_a$-LES in Theorem 1 Part (c) is the GES, and the result follows. □

C. Proof of Theorem 2

Proof. For every $\nu$, R-P-$\varepsilon$ returns a set $\hat{A}_\nu$ in finite time, thus producing an infinite sequence of solutions $\{\hat{A}_\nu\}$. Further, R-P-$\varepsilon$ never returns a set $\hat{A}_\nu$ containing a point whose estimated objective vector is dominated by $G_{m_\nu}(x_0)$ (see Algorithm 2, P-$\varepsilon$ Step 26). Recall that for all $\nu \geq \nu$, $\hat{A}_\nu \subseteq \hat{S}_\nu(x_0) \subseteq S(x_0, \alpha)$ w.p.1, and since $S(x_0, \alpha)$ is finite, then any sequence of estimated efficient points $\{X_{\nu}^s : X_{\nu}^s \in \hat{A}_\nu \text{ for all } \nu = 1, 2, \ldots\}$ is bounded w.p.1.

As in the proof of Theorem 1 Part (a), for all $k \in \{1, 2\}$, $\tilde{G}_{k,m_\nu}(\cdot)$ uniformly converges to $g_k(\cdot)$ w.p.1 as $\nu \rightarrow \infty$ on the finite set $N_a(S(x_0, \alpha)) \cap \mathcal{X}$. Since there are only two objectives and $\beta_\varepsilon \in (0, \infty)$, the maximum $\varepsilon$-placement distance $\max_{k \in \{1, 2\}} \tilde{\delta}_{k,m_\nu}(\cdot, \beta_\varepsilon) = \max_{k \in \{1, 2\}} \delta_{k,m_\nu}(\cdot)/m_\nu^{\beta_\varepsilon}$ also uniformly converges to zero w.p.1 as $\nu \rightarrow \infty$ on
\( N_a(S(x_0, \alpha)) \cap \mathcal{X} \). Let \( \kappa > 0 \) be as in Assumption 5. Then w.p.1, there exists \( \nu'_{P_\varepsilon} \) (dependent on \( a, x_0, \kappa, \alpha \), and the random realization) such that for all \( \nu \geq \nu'_{P_\varepsilon} \) and all \( x \in N_a(S(x_0, \alpha)) \cap \mathcal{X} \), we have \( \max_{k \in \{1, 2\}} |\hat{G}_{k, m_\nu}(x) - g_k(x)| < \kappa/4 \) w.p.1. Also w.p.1, there exists \( \nu''_{P_\varepsilon} \) (dependent on the same quantities as \( \nu'_{P_\varepsilon} \) and dependent on \( \beta_\varepsilon \)) such that for all \( \nu \geq \nu''_{P_\varepsilon} \), \( \max_{k \in \{1, 2\}} |\hat{f}_{k, \nu}(x, \beta_\varepsilon) - g_k(x)| < \kappa/4 \) w.p.1. Combining the above results, if \( \nu \geq \max\{\nu'_{P_\varepsilon}, \nu''_{P_\varepsilon}\} \), then for all \( x \in N_a(S(x_0, \alpha)) \cap \mathcal{X} \), \( \max_{k \in \{1, 2\}} |\hat{G}_{k, m_\nu}(x) - \hat{f}_{k, \nu}(x, \beta_\varepsilon)| - g_k(x) < \kappa/2 \) w.p.1. Henceforth, let \( \nu \geq \max\{\nu'_{P_\varepsilon}, \nu''_{P_\varepsilon}\} \). Then for all \( x, x' \in N(S(x_0, \alpha)) \cap \mathcal{X} \), the following hold w.p.1:

R5. \( \forall k \in \{1, 2\}, g_k(x) < g_k(x') \) if and only if \( \hat{G}_{k, m_\nu}(x) + \hat{f}_{k}(x, \beta_\varepsilon) < \hat{G}_{k, m_\nu}(x') - \hat{f}_{k}(x', \beta_\varepsilon) \);

R6. if \( G_{m_\nu}(x) \notin G_{m_\nu}(x') \), then \( g(x) \notin g(x') \), that is, \( \exists k \in \{1, 2\} \) such that \( g_k(x') < g_k(x) \).

Under Assumption 1, for any \( x_0 \in \mathcal{X}, \mathcal{E} \subseteq S(x_0, \alpha) \). Therefore results R5, R6 and Assumptions 5–6 imply that all points in \( \mathcal{E} \) are both sample-path \( \mathcal{N}_a \)-LWEP’s and sample-path global efficient points w.p.1. Further, all points in \( \mathcal{N}_a(\mathcal{E}) \) are not sample-path \( \mathcal{N}_a \)-LWEP’s w.p.1. Let \( c_\varepsilon := |\mathcal{E}| \geq 1 \), and for any objective \( k \in \{1, 2\} \), sort the elements of \( \mathcal{E} \) on objective \( k \) so that \( g_k(x_{k(1)}^*) < \ldots < g_k(x_{k(c_\varepsilon)}^*) \), where \( x_{k(i)}^* \) denotes the \( i \)th ordered element of \( \mathcal{E} \) on objective \( k \), \( i = 1, \ldots, c_\varepsilon \). If \( c_\varepsilon \geq 2 \), then result R5 implies that w.p.1 for all \( i = 1, \ldots, c_\varepsilon - 1 \),

\[
\hat{G}_{k, m_\nu}(x_{k(i)}^*) + \hat{f}_{k, \nu}(x_{k(i)}^*, \beta_\varepsilon) < \hat{G}_{k, m_\nu}(x_{k(i+1)}^*) - \hat{f}_{k, \nu}(x_{k(i+1)}^*).
\]  

(1)

By Lemma 5, w.p.1 there exists \( \nu''' \) (dependent on the same quantities as \( \nu''_{P_\varepsilon} \)) such that for all \( \nu \geq \nu''' \), the updated sample-path \( \mathcal{N}_a \)-local minimizers returned as part of P\( \varepsilon \) Step 1, which we call \( \mathcal{M}_\nu \), are such that \( \mathcal{M}_\nu \subseteq \mathcal{M}_a \). Under Assumptions 5–6, the set \( \mathcal{M}_a = \{x_1^{\min}, x_2^{\min}\} \) contains the unique global minimizers for each objective \( k \in \{1, 2\} \).

Henceforth, let \( \nu > \max\{\nu, \nu'_{P_\varepsilon}, \nu''_{P_\varepsilon}, \nu''_{P_\varepsilon}\} \), and let \( \{k^{\nu}_\nu, \nu = 1, 2, \ldots\} \) be any sequence of objectives minimized, where \( k^{\nu\nu}_\nu \neq k^{\nu}_\nu \) for each \( \nu \). Then by Lemma 3, the set of decision points \( \hat{A}_\nu \) returned by R-P\( \varepsilon \) lie in \( S(x_0, \alpha) \) w.p.1, as does the set of points used to set the \( \varepsilon \)-values in Algorithm 2, P\( \varepsilon \) Step 2, which is a set of sample-path \( \mathcal{N}_a \)-LWEP’s we call \( \hat{A}_w \). Since all points in \( \hat{A}_w \) are sample-path \( \mathcal{N}_a \)-LWEP’s, then results R5, R6 and Assumptions 5–6 ensure that \( \hat{A}_w \subseteq \mathcal{E} \) w.p.1; further, \( \mathcal{M}_\nu \subseteq \hat{A}_w \), where \( \mathcal{M}_\nu = \{x_1^{\min}, x_2^{\min}\} \) w.p.1. If \( c_\varepsilon \in \{1, 2\} \), the proof is complete, since \( \hat{A}_w \) is returned as \( \hat{A}_\nu \) in Algorithm 2, P\( \varepsilon \) Step 26, and no other points have entered the set w.p.1. Now suppose \( c_\varepsilon \geq 3 \). All points in \( \mathcal{M}_\nu \cup \mathcal{E} \) can be ordered on \( k^{\nu\nu}_\nu \) as in line (1). Points in \( \mathcal{E} \setminus \hat{A}_w \) are retrieved by Algorithm 2, P\( \varepsilon \) Steps 15–25, and carried forward to \( \hat{A}_{\nu+1} \); no other points enter the set w.p.1. Then it follows that for all \( \nu^* > \nu + 1 \), \( \hat{A}_{\nu^*} = \mathcal{E} \) w.p.1, and the result holds.
D. Proof of Theorem 3

Proof of Theorem 3 Part (a). Let $\mathcal{D} \subseteq \mathcal{X}$ be any subset of the feasible set. Since $\mathcal{X}$ is finite, $\mathcal{D}$ is finite. Let $\hat{\mathcal{B}}^*_{k,\nu}(\mathcal{D})$ denote the set of sample-path global minimizers of objective $g_k$, $k \in \{1, \ldots, d\}$ on the set $\mathcal{D}$, and let $\mathcal{B}^*_k(\mathcal{D})$ denote the corresponding set of true global minimizers on $\mathcal{D}$. Then under our assumptions, by Wang et al. (2013, p. 16), for all $k \in \{1, \ldots, d\}$ and all $\mathcal{D} \subseteq \mathcal{X}$, there exists $\eta > 0$ such that for large enough $\nu$, 

$$\mathbb{P}\{\hat{\mathcal{B}}^*_{k,\nu}(\mathcal{D}) \not\subseteq \mathcal{B}^*_k(\mathcal{D})\} \leq |\mathcal{D}| e^{-m\nu \eta}.$$ (2)

Recall that $\mathcal{X} \subset \mathbb{Z}^q$ and let $x \in \mathcal{X}$ be a feasible point. Letting $e_i$ denote a $q$-dimensional vector of zeros with one in the $i$th place, divide $\mathcal{N}_1(x)$ into $2q$ sub-neighborhoods that include $x$ and exactly one other neighborhood point in each direction, $\mathcal{N}_{1,\pm i}(x) := \{x, x + e_i\}$ and $\mathcal{N}_{1,\pm i}(x) := \{x, x - e_i\}$ for all $i \in \{1, \ldots, q\}$.

For every non-$\mathcal{N}_1$-LWEP $x \in \mathcal{X} \setminus \mathcal{X}^w$, there must exist $x' \in \mathcal{N}_1(x) \cap \mathcal{X}$ such that $g(x')$ strictly dominates $g(x)$. Then for every $x \in \mathcal{X} \setminus \mathcal{X}^w$, there exists $j \in \{-q, \ldots, -1, 1, \ldots, q\}$ and $x' \in \mathcal{N}_1(x) \cap \mathcal{X}$ such that $\mathcal{N}_{1,j}(x) = \{x, x'\}$ and $g(x')$ strictly dominates $g(x)$. Thus $x$ is not a global minimizer on $\mathcal{N}_{1,j}(x)$ on any objective. If $x \in \mathcal{X} \setminus \mathcal{X}^w$ is nonetheless estimated as an $\mathcal{N}_1$-LWEP, that is, $x \in \mathcal{X}_\nu^w$ on its $\mathcal{N}_1$-neighborhood, there must exist an objective $k \in \{1, \ldots, d\}$ such that $\hat{\mathcal{B}}^*_{k,\nu}(\mathcal{N}_{1,j}(x)) \not\subseteq \mathcal{B}^*_k(\mathcal{N}_{1,j}(x))$. Then for large enough $\nu$, 

$$\mathbb{P}\{\hat{\mathcal{X}}_\nu^w \not\subseteq \mathcal{X}^w\} \leq \sum_{x \in \mathcal{X} \setminus \mathcal{X}^w} \mathbb{P}\{x \in \hat{\mathcal{X}}_\nu^w\}$$

$$\leq \sum_{x \in \mathcal{X} \setminus \mathcal{X}^w} \sum_{j \in \{-q, \ldots, -1, 1, \ldots, q\}} \sum_{k \in \{1, \ldots, d\}} \mathbb{P}\{\hat{\mathcal{B}}^*_{k,\nu}(\mathcal{N}_{1,j}(x)) \not\subseteq \mathcal{B}^*_k(\mathcal{N}_{1,j}(x))\}$$

$$\leq \sum_{x \in \mathcal{X} \setminus \mathcal{X}^w} \sum_{j \in \{-q, \ldots, -1, 1, \ldots, q\}} \sum_{k \in \{1, \ldots, d\}} 2 e^{-m\nu \eta} \leq |\mathcal{X}| 4qde^{-m\nu \eta},$$

where $\eta > 0$ denotes the relevant constant from line (2).

Proof of Theorem 3 Part (b). We begin by noticing that item (i) follows from Theorem 3 Part (a), along with the assumption that sample sizes increase at least linearly and the fact that our algorithms guarantee $\mathcal{A}_\nu$ contains only sample-path $\mathcal{N}_1$-LWEP’s.

To prove item (ii), notice that under our assumptions, all $\mathcal{N}_1$-LWEP’s are global efficient points. Therefore by item (i), $\mathbb{P}\{\mathcal{A}_\nu \not\subseteq \mathcal{E}\} = O(e^{-\gamma m\nu})$ for some $\gamma > 0$. We now consider $\mathbb{P}\{\mathcal{E} \not\subseteq \mathcal{A}_\nu\}$, where $\mathbb{P}\{\mathcal{E} \not\subseteq \mathcal{A}_\nu\} \leq \sum_{x \in \mathcal{E}} \mathbb{P}\{x \not\in \hat{\mathcal{A}}_\nu\}$. Notice that if $x \in \mathcal{E}$ is not in $\hat{\mathcal{A}}_\nu$, then it must have been incorrectly estimated as dominated by at least one point in its neighborhood. Then by a proof similar to that of Theorem 3 Part (a), it follows that $\mathbb{P}\{\mathcal{E} \not\subseteq \hat{\mathcal{A}}_\nu\} = O(e^{-\gamma m\nu})$ for some $\gamma > 0$, which implies the result. □
E. Finding $\mathcal{N}_1$-LWES’s in Problem $T_C$

To calculate the collection of all possible $\mathcal{N}_1$-LWES’s in Problem $T_C$, we first locate all $\mathcal{N}_1$-LWEP’s; we find 512. Then, starting from each $\mathcal{N}_1$-LWEP, we run a program similar to RLE with no relaxation to find the smallest complete $\mathcal{N}_1$-LWES that contains the $\mathcal{N}_1$-LWEP. We refer to these $\mathcal{N}_1$-LWES’s as level-$1\mathcal{N}_1$-LWES’s; after removing duplicate sets, we find 39. Then, we take all possible unions of two level-1 $\mathcal{N}_1$-LWES’s, remove any dominated points, and check if this set is a new, unique $\mathcal{N}_1$-LWES. We refer to all new, unique $\mathcal{N}_1$-LWES’s that are found by taking the union of two level-1 $\mathcal{N}_1$-LWES’s as level-$2\mathcal{N}_1$-LWES’s. We repeat this process for level-3 and so on, up to level-8. We found one level-7 $\mathcal{N}_1$-LWES and no level-8 $\mathcal{N}_1$-LWES’s. The total number of unique $\mathcal{N}_1$-LWES’s found in this manner, up to level-8, was 516. All together, these 516 $\mathcal{N}_1$-LWES’s contain just 73 points; we call these points $\mathcal{N}_1$-LWES members in Figure 7. Recall that there are 512 $\mathcal{N}_1$-LWEP’s, so not all $\mathcal{N}_1$-LWEP’s are members of an $\mathcal{N}_1$-LWES.

References
