Online Supplement to "A New Framework for Multi-Objective Simulation-Based Goal Programming Using Retrospective Optimization"

Shing Chih Tsai

Department of Industrial and Information Management, National Cheng Kung University

Hao-Chun Lu

Department of Management Science, National Yang Ming Chiao Tung University

Honggang Wang

Department of Technology and Operations, California State Polytechnic University

---

A Appendix

A.1 The Details of Modified SPLINE Algorithm

We provide pseudocode for the modified SPLINE algorithm used within the WGP-SPLINE framework (see Table 1). The SPLINE algorithm, as listed in Table A1, iterates between two routines: (i) SPLI (search with piecewise-linear interpolation; see Table A2) generates the next candidate solution based on continuous piecewise-linear interpolation of the objective function $\widehat{g}(\boldsymbol{X})$; (ii) NE (neighborhood enumeration) ensures that the selected solution is not worse, in terms of the sample-path function, than any of the $2d$ neighboring solutions that are exactly one unit away. Here, d represents the dimension of the decision vector $\boldsymbol{X}$. In our modified algorithm, we replace the original SPLI call with the modified SPLI that computes subgradients and generates cuts. We also maintain a global set of cuts $\mathcal{C}$ that is updated in each SPLI call, ensuring that these cuts are considered in both SPLI and NE. When evaluating neighboring solutions in NE, we ensure they satisfy the accumulated cuts.

At the beginning of the modified SPLI, we add small perturbations (uniformly distributed within a small cube centered at the input integer point $\boldsymbol{X}_0$) to obtain a non-integer starting point. In the modified SPLI, specifically in Step 5, we replace the gradient computation with subgradient computation using the modified PLI (Piecewise-Linear Interpolation) algorithm (see Table A3).

We use the subgradient γ as the search direction. Here, we introduce the subgradient rules for max-functions. For $h(x) = \max\{f(x), 0\}$: (i) If $f(x) > 0$, the subgradient is $\nabla f(x)$; (ii) If $f(x) = 0$, the subgradient is any vector $\lambda \nabla f(x)$, where $\lambda \in [0, 1]$; (iii) If $f(x) < 0$, the subgradient is the zero vector $\mathbf{0}$. By appropriately choosing λ, we avoid situations where the algorithm stalls due to a zero subgradient. This allows the algorithm to seamlessly handle points where the function is not differentiable without special cases or exceptions. At each iteration, we use the subgradient γ to generate a cutting plane (cut): $\widehat{g}_{m_k}(\boldsymbol{X}) \geq \widehat{g}_{m_k}(\boldsymbol{X}_{best}) + \gamma^T(\boldsymbol{X} - \boldsymbol{X}_{best})$. We store each cut for use in refining the problem. We accumulate the cuts generated in each iteration and incorporate these cuts into the optimization problem solved during the line search. We adjust the PLI algorithm to compute subgradients instead of gradients, accommodating the non-differentiability introduced by the max-functions in the WGP objective function. In the modified PLI algorithm, we compute subgradients at each vertex of the simplex surrounding $\boldsymbol{X}$ and use the weights derived from the fractional parts of $\boldsymbol{X}$ to aggregate these subgradients. This process estimates the overall subgradient at $\boldsymbol{X}$ without relying on differences in function values, which is crucial in handling non-differentiable points effectively.

We provide a detailed explanation of the NE operation in the context of the modified WGP-SPLINE algorithm. The NE operation ensures that the current solution is not inferior to any of its immediate neighboring solutions in terms of the sample-path objective function $\widehat{g}_{m_k}(\boldsymbol{X})$. This is crucial because the SPLI routine may not always find the best integer solution due to the non-differentiability and approximation errors in the objective function. Let $N^1(\boldsymbol{X})$ denote the set of integer points that lie within one unit distance from the integer point $\boldsymbol{X}$ in any coordinate direction. For each feasible solution $\boldsymbol{X}'$ in $N^1(\boldsymbol{X}'_{new})$, we evaluate the objective function $\widehat{g}_{m_k}(\boldsymbol{X}')$ using the simulation oracle. The total number of simulation observations used in NE, denoted N''_k, does not exceed $(2d+1)m_k$, where m_k is the sample size for each evaluation. After evaluating all feasible points in $N^1(\boldsymbol{X}'_{new})$, there are three possible scenarios: (i) If a unique best solution $\boldsymbol{X}$ is identified, NE adopts this solution; (ii) If one or more points, including $\boldsymbol{X}'_{new}$, are equally optimal, NE retains $\boldsymbol{X}'_{new}$; (iii) If there are multiple points that are tied for best and better than $\boldsymbol{X}'_{new}$, then NE selects any one of these tied solutions. In all scenarios, the solution returned by NE is denoted as $\boldsymbol{X}_{new}$. In the modified algorithm, NE takes into account the cuts generated from subgradients to ensure that the neighborhood search respects the refined feasible region. By systematically evaluating the

Table A1: The framework of SPLINE algorithm.

Require: SPLINE algorithm parameters $\{m_k, v_k\}$; and initial solution $\boldsymbol{X}_{old} \in \mathbb{X}$.
Provide: updated solution $\boldsymbol{X}_{new} \in \mathbb{X}$; total number N_k of oracle calls expended.

1. Set $N_k = 0$ and cuts set $\mathcal{C} = \emptyset$ {initialize number of oracle calls}
2. **repeat**
3. Call modified SPLI: $[N_k^{'}, \boldsymbol{X}_{new}^{'}, \widehat{g}_{m_k}(\boldsymbol{X}_{new}^{'})] = \text{SPLI}(\boldsymbol{X}_{old}, m_k, v_k, \mathcal{C})$ {SPLI algorithm}
4. Update cuts set $\mathcal{C}$ with cuts from SPLI.
5. Call NE: $[N_k^{''}, \boldsymbol{X}_{new}, \widehat{g}_{m_k}(\boldsymbol{X}_{new})] = \text{NE}(\boldsymbol{X}_{new}^{'}, m_k, \mathcal{C})$ {neighborhood enumeration considering cuts}
6. Set $N_k = N_k + N_k^{'} + N_k^{''}$ {update the number of oracle calls}
7. **until** $\widehat{g}_{m_k}(\boldsymbol{X}_{new}) = \widehat{g}_{m_k}(\boldsymbol{X}_{new}^{'})$ or $N_k > v_k$ {found a local solution or timed out}

neighborhood, NE adds robustness to the algorithm, preventing it from missing better solutions due to the limitations of the interpolation or subgradient approximation.

A.2 Ranking and Selection Framework for Solving WGP and MGP

In this section, we present an alternative R&S framework for addressing the WGP and MGP formulations corresponding to the inventory optimization problem described in Section 4. The proposed framework assumes independent solutions and consists of two phases: "feasibility check" and "selection". In Phase I, the framework performs a fully sequential feasibility check procedure (FCP) similar to that proposed by Batur and Kim (2010), to evaluate the candidate solutions. The FCP collects one observation from all surviving solutions at each sampling stage, and then determines the feasibility of each solution by comparing it against computed thresholds. More specifically, we define feasibility with respect to the stochastic constraints: $Z_r(\boldsymbol{R}, \boldsymbol{Q}) \leq t_r, \quad \forall r \in \{1, 2, 3\}$. If a solution is deemed infeasible for the rth constraint (i.e., $Z_r(\boldsymbol{R}, \boldsymbol{Q}) > t_r$), we then set $d_r^- = 0$ and compute the estimator of $d_r^+ = \max\{Z_r(\boldsymbol{R}, \boldsymbol{Q}) - t_r, 0\}$ in Phase II. Otherwise, if the solution is determined to be feasible for the rth constraint (i.e., $Z_r(\boldsymbol{R}, \boldsymbol{Q}) \leq t_r$), we then set $d_r^+ = 0$ in Phase II. Note that our FCP is different from the procedure of Batur and Kim (2010) in that our FCP will not be terminated unless the feasibility of all constraints for all solutions are identified. By contrast, Batur and Kim (2010) claim that the solution is infeasible and remove it from consideration if at least one constraint is determined as infeasible. We first describe the FCP as follows.

Table A2: The Search with Piecewise-Linear Interpolation (SPLI) algorithm.

Require: SPLI algorithm parameters $\{m_k, v_k\}$; and solution $\boldsymbol{X}_0 \in \mathbb{X}$.
Provide: solution $\boldsymbol{X}_{best} \in \mathbb{X}$ with $\widehat{g}_{m_k}(\boldsymbol{X}_{best}) < \widehat{g}_{m_k}(\boldsymbol{X}_0)$, or $\boldsymbol{X}_{best} = \boldsymbol{X}_0$.

1. Define parameters: initial step $s_0 = 2.0$ and multiplier $c = 2.0$
2. Set $\boldsymbol{X}_{best} = \boldsymbol{X}_0$ and $N_k^{'} = 0$.
3. Continue {begin new line search with new gradient value}
4. Set $\boldsymbol{X}_1 = \text{PERTURB}(\boldsymbol{X}_{best})$ {perturb initial solution to a non-integer point}
5. Call modified PLI$(\boldsymbol{X}_1, m_k)$ to compute subgradient γ
6. **If** $\|\gamma\|$ is undefined or zero, **return** $[N_k^{'}, \boldsymbol{X}_{best}, \widehat{g}_{m_k}(\boldsymbol{X}_{best})]$ {stop if no direction}
7. Generate cut using γ and add to $\mathcal{C}$
8. Set $N_k^{'} = N_k^{'} + (d+1)m_k$
9. **If** $N_k^{'} > v_k$, **return** $[N_k^{'}, \boldsymbol{X}_{best}, \widehat{g}_{m_k}(\boldsymbol{X}_{best})]$ {stop if too many oracle calls}
10. Initialize line search parameters: $b = 0$ and $\boldsymbol{X}_0 = \boldsymbol{X}_{best}$
11. **repeat line search:**
12. Set $b = b + 1$, $s = c^{b-1} \times s_0$, and set $\boldsymbol{X}_1 = \boldsymbol{X}_0 - s \times \gamma / \|\gamma\|$
13. Project $\boldsymbol{X}_1$ onto feasible region considering cuts $\mathcal{C}$
14. **If** $\boldsymbol{X}_1$ is infeasible, **return** $[N_k^{'}, \boldsymbol{X}_{best}, \widehat{g}_{m_k}(\boldsymbol{X}_{best})]$ {stop if infeasible}
15. Evaluate $\widehat{g}_{m_k}(\boldsymbol{X}_1)$ and set $N_k^{'} = N_k^{'} + m_k$
16. **until** $\boldsymbol{X}_1 \neq \boldsymbol{X}_{best}$ or $N_k^{'} > v_k$
17. **If** $b \leq 2$, **return** $[N_k^{'}, \boldsymbol{X}_{best}, \widehat{g}_{m_k}(\boldsymbol{X}_{best})]$ {stop if line search ended quickly}
18. Go to Step 3

Table A3: The Piecewise-Linear Interpolation (PLI) algorithm.

Require: PLI algorithm parameters $\{m_k\}$; and point $\boldsymbol{X} = (x^1, x^2, \ldots, x^d)$.
Provide: gradient γ of the function $\widehat{g}_{m_k}(\boldsymbol{X})$.

1. Set $\boldsymbol{X}_0^j = \lfloor x^j \rfloor$, for $j = 1, 2, \ldots, d$ {integer floor}
2. Set $z^j = x^j - \boldsymbol{X}_0^j$, for $j = 1, 2, \ldots, d$ {fractional parts}
3. Sort $z = (z^1, z^2, \ldots, z^d)$ to get $1 = z^{p(0)} > z^{p(1)} \geq \cdots \geq z^{p(d)} \geq z^{p(d+1)} = 0$
4. Set $\boldsymbol{X}_j = \boldsymbol{X}_{j-1} + e_{p(j)}$, for $j = 1, 2, \ldots, d$ {identify the remaining vertices}
5. Set $w^j = z^{p(j)} - z^{p(j+1)}$, for $j = 0, 1, \ldots, d$ {calculate the weight associated with each vertex j}
6. Set $\delta = 0$
7. **If** $\delta < d + 1$ **return**
8. Compute subgradient $\gamma^{p(j)}$ for $j = 1, 2, \ldots, d$, using subgradient rules.
9. **return** γ with jth component γ^j, for $j = 1, 2, \ldots, d$

Let $\mathcal{K}$ denote the total number of solutions whose feasibility needs to be identified. In our problem, $\mathcal{K}$ corresponds to the number of candidate solutions $(\boldsymbol{R}, \boldsymbol{Q})$ that satisfy the constraints specified by Equation (3), assuming the problem size is manageable. If the problem size is too large to be handled within the specified computational budget, we randomly select a subset of $\mathcal{K}$ candidate solutions from the solution space for evaluation. Let $\mathcal{C} = \{1, 2, 3\}$ represent the set of stochastic constraints, corresponding to the three objectives in our problem. For each constraint $r \in \mathcal{C}$, the target level t_r represents the threshold value for the rth constraint. We define Z_{rj} as the jth simulation observation associated with the performance measure Z_r (i.e., the rth objective function).

Feasibility Check Procedure (FCP)

Setup: Choose initial sample size $n_0 \geq 2$ and confidence level $0 < 1 - \alpha_1 < 1$. Determine the vector of tolerance levels $\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3)^T$. Let M contain all solutions and $F = \emptyset$. Set $L_i = \emptyset$ for $i = 1, 2, \ldots, \mathcal{K}$ and $\eta = 1/2 \left[\left(\frac{2\alpha_1}{3\mathcal{K}} \right)^{-2/(n_0-1)} - 1 \right]$. Obtain observation Z_{rj} for $r \in \{1, 2, 3\}$, $j = 1, 2, \ldots, n_0$, and compute its corresponding sample variance $S_r^2(n_0)$ for each solution among M. Specify the observation counter $n_i = n_0$ for $i = 1, 2, \ldots, \mathcal{K}$ and proceed to **Feasibility Check**.
Feasibility Check: For each solution i among M,

- If there is an $r \in \mathcal{C}$ such that $\sum_{j=1}^{n_i} (Z_{rj} - t_r) \geq \max \left\{ 0, \frac{\eta(n_0-1)S_r^2(n_0)}{\psi_r} - \frac{\psi_r}{2} n_i \right\}$, then we claim that the solution is infeasible with respect to the rth constraint. We also move the constraint index r to L_i. If the feasibility for all constraints has been determined, we then remove the solution index from M.
- Else if $\sum_{j=1}^{n_i} (Z_{rj} - t_r) \leq -\max \left\{ 0, \frac{\eta(n_0-1)S_r^2(n_0)}{\psi_r} - \frac{\psi_r}{2} n_i \right\}$ for all $r \in \mathcal{C}$, then we claim that the solution is feasible and move the solution index from M to F.

Terminating Rule: If $|M| = 0$, then stop **FCP**. Else, sample one additional observation $Z_{r(n_i+1)}$ from each candidate solution belonging to the set M. Set $n_i = n_i + 1$ and go back to **Feasibility Check**.

Remark A.1. Assuming that the observations Z_{rj}, for $r \in \{1, 2, 3\}$ and $j = 1, 2, \ldots$ follow a normal distribution, we can prove, following a similar derivation as in Batur and Kim (2010), that the FCP guarantees to determine the feasibility of favorably feasible constraints (i.e., $Z_r(\boldsymbol{R}, \boldsymbol{Q}) \leq t_r + \psi_r$)

and unfavorably infeasible constraints (i.e., $Z_r(\boldsymbol{R}, \boldsymbol{Q}) > t_r - \psi_r$) with a specified confidence level of $1 - \alpha_1$.

If the set F contains at least one solution after applying FCP, then we have found the optimal solutions for both the WGP and MGP formulations, and do not need to proceed to Phase II. However, if F is empty, we select the solution with the best stochastic objective measure for WGP or MGP (i.e., minimizing Equation (5) or (8)) among all the candidate solutions in Phase II. For a given solution i, we only need to consider the positive deviations d_r^+ for the constraints assigned to L_i; we set $d_r^+ = 0$ for constraints not in L_i. Since the sample sizes collected from each candidate solution after applying FCP may differ, we suggest using the clean-up procedure of Boesel et al. (2003) in Phase II. This procedure can handle unequal sample sizes and provide a statistical guarantee for identifying the best solution among the visited ones. Let n_i denote the accumulated sample size for solution i in Phase I. We denote the observation of the positive deviation (associated with the rth performance measure) obtained from the jth simulation replication as $d_{rj}^+ = \max\{Z_{rj} - t_r, 0\}$. The steps of the ranking and selection framework for WGP (WGP-RSF) are detailed below.

Ranking and Selection Framework (WGP-RSF)

Setup: Select the overall nominal confidence level $1/\mathcal{K} \leq 1 - \alpha < 1$ and choose $\alpha_1 > 0$ and $\alpha_2 > 0$ such that $\alpha_1 + \alpha_2 = \alpha$. Implement the steps described in the **Setup** of **FCP**.

Feasibility Check: Same as in **FCP**.

Terminating Rule for Phase I: Same as in **FCP**. If $|F| > 0$, then terminate this framework and choose any solution from F as the optimal solution of Problem WGP. Otherwise (i.e., $|F| = 0$), proceed to Phase II. Suppose that the accumulated sample size for solution i in Phase I is denoted as n_i.

Setup for Phase II: Specify the indifference-zone parameter δ (with respect to the weighted sum of positive deviations). For solution i, compute the sample variance $S^2(n_i)$ with respect to $\sum_{r=1}^{3}[w_r^+ \times d_r^+]$ based on observations $\{d_{rj}^+, j = 1, 2, \ldots, n_i\}$. Note that we simply set $d_r^+ = 0$ for $r \notin L_i$.

Selection: Set $n_{\min} = \min_i\{n_i\}$ and $h = h(2, (1-\alpha_2)^{1/(\mathcal{K}-1)}, n_{\min})$ which is the extended Rinott's

(1978) constant. Determine the total required sample size for each solution $i = 1, 2, \ldots, \mathcal{K}$:

$$N_i = \max\left\{n_i, \left\lceil \left(\frac{hS(n_i)}{\delta}\right)^2 \right\rceil\right\}.$$

Take $N_i - n_i$ additional observations of d_{rj}^+, for $r \in L_i$, from each solution i. Select the best solution i with the smallest overall sample mean $(1/N_i)\sum_{j=1}^{N_i}\sum_{r=1}^{3}[w_r^+ \times d_{rj}^+]$.

Remark A.2. We denote the weighted sum of positive deviations for solution $(\boldsymbol{R}, \boldsymbol{Q})$ as $WS(\boldsymbol{R}, \boldsymbol{Q}) = \sum_{r=1}^{3}[w_r^+ \times d_r^+]$. Let $(\boldsymbol{R}^*, \boldsymbol{Q}^*)$ be the solution that minimizes $WS(\boldsymbol{R}, \boldsymbol{Q})$ while satisfying $Z_r(\boldsymbol{R}, \boldsymbol{Q}) > t_r - \psi_r$ for at least one performance measure among all $\mathcal{K}$ solutions. The finite-time statistical validity of WGP-RSF can be proven by applying Bonferroni's inequality. Specifically, we can demonstrate that

$$\Pr\{\text{Select } (\boldsymbol{R}, \boldsymbol{Q}) \text{ such that } WS(\boldsymbol{R}, \boldsymbol{Q}) < WS(\boldsymbol{R}^*, \boldsymbol{Q}^*) + \delta\} \geq 1 - \alpha.$$

This ensures that the selected solution is statistically guaranteed to have a performance that is no worse than the best among all visited solutions with a certain confidence level.

The R&S framework for MGP (MGP-RSF) is similar to that for WGP but differs in the objective function. Instead of minimizing the weighted sum of positive deviations $\sum_{r=1}^{3}[w_r^+ \times d_r^+]$, we minimize the maximum weighted positive deviation $\max_{r\in\{1,2,3\}}[w_r^+ \times d_r^+]$. A statistical guarantee similar to that described in Remark A.2 also holds for the MGP-RSF framework.

References

Batur, D., & Kim, S.-H. (2010). Finding feasible systems in the presence of constraints on multiple performance measures. *ACM Transactions on Modeling and Computer Simulation*, *20*(3), 1–26.

Boesel, J., Nelson, B.L., & Kim, S.-H. (2003). Using ranking and selection to Using ranking and selection to"clean up" after simulation optimization. *Operations Research*, *51*(5), 814–825.

Rinott, Y. (1978). On two-stage selection procedures and related probability-inequalities. *Communications in Statistics - Theory and Methods*, *A7*, 799–811.