Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

A SIMULATION-INFUSED OPTIMIZATION APPROACH FOR DECOMPOSING NONLINEAR SYSTEMS

Zeyu Liu[1]

[1]Dept. of Industrial and Mgmt. Systems Eng., West Virginia University, Morgantown, WV, USA

ABSTRACT

With the rapid advancements in modern computing technologies, simulation has been increasingly adopted to model complex real-world systems. Yet, such digital computational power remains underutilized in decision-making tasks, mainly due to the difficulties in integrating gray or black box simulation models with algebraic optimization. In this study, we address this by proposing a novel and generic simulation-infused optimization approach. Through decomposition, nonlinear subsystems are surrogated by simulation and approximated by cutting planes in a row generation algorithm for holistic optimization. Exploration and exploitation are introduced to the row generation algorithm via a novel parametric method. The proposed simulation-infused decomposition approach is applied to a multi-depot vehicle routing problem. Proof-of-concept experiments are conducted to validate the performance against benchmarks, showing drastic reductions in computational time and improvements upon conventional optimization methods when real-world nonlinear systems are involved.

1 INTRODUCTION

In recent years, the computational power of simulation and digital twins has revolutionized many fields, including manufacturing (Li et al. 2022), urban development (Ferré-Bigorra et al. 2022), agriculture (Pylianidis et al. 2021), and so on. However, it still remains an open question how virtual systems can be effectively utilized to improve physical ones (de Paula Ferreira et al. 2020). Since simulation and digital twins are most suited for descriptive and predictive tasks, it is often challenging to adopt them as prescriptive tools for decision-making problems.

To address this issue, many have combined simulation with various optimization techniques. Systematic reviews have defined clear scopes of these methods (Amaran et al. 2016). Most recent advancements in simulation-based optimization can be categorized into three main classes. First, studies have adopted a separate approach where simulation is used independently to estimate parameters or generate scenarios for the optimization (Lanzarone et al. 2018; Khalili-Damghani et al. 2022; Högdahl and Bohlin 2023). Second, Monte Carlo methods are applied to optimize systems with repeated simulations, typically involving heuristic algorithms (Babamiri and Marofi 2021; Sakki et al. 2022). Third, simulation is combined with heuristics in sample average approximations (SAA) to estimate subsystem objectives under stochastic parameters (Li and Zhang 2018; Wang et al. 2023).

As such, in the current literature, the central optimization technique, in a simulation environment, is mostly achieved through heuristics with Monte Carlo methods, either directly optimizing the entire system (Babamiri and Marofi 2021; Sakki et al. 2022), or solving subsystems inside an SAA algorithm (Li and Zhang 2018; Wang et al. 2023). As pointed out, due to the lack of "algebraic models", it is difficult to employ existing optimization techniques in a simulation environment (Amaran et al. 2016). This has substantially limited the potential of taking advantage of the computational power of state-of-the-art simulation methods. Complex optimization problems, such as those with integer variables and nonlinear constraints, will often benefit from a tractable optimization-simulation framework that produces good-quality solutions.

One of the most widely applicable optimization methods for complex problems is decomposition, e.g., the Benders decomposition (Benders 1962), where a system is partitioned into two or more subsystems to resolve nonlinearity, or to adopt a "divide-and-conquer" approach for fast computations. However, in many cases, evaluating the subsystems remains to be difficult, especially when integer and nonlinear components are involved (Rahmaniani et al. 2017). Although simulation can easily resolve nonlinear computations, a significant gap still exists in the literature when jointly using simulation models with optimization to solve real-world problems. A recent advancement in the logic-based Benders decomposition (LBBD) has attempted to address this by generating valid Benders cuts from binary variables through simulation (Forbes et al. 2024). Yet, more efforts are still needed, especially when gray/black box simulation software is used to model complex systems.

In this study, we adopt a novel approach to address the research gap. We develop a generic decomposition framework that unifies simulation and optimization through linear approximations of nonlinear subsystems. Specifically, we consider problems of the form

$$\min_{x\in X, y\in Y} \quad \left\{c^T x + f(x,y) :\ Ax = b, g(x,y) \geq 0\right\}. \tag{1}$$

We define $x \in X$ and $y \in Y$ as the decision variables, where X and Y are n_1 and n_2 dimensional mixed integer regions of x and y. We let X be the bounds of x but allow Y to be a complex region that may include more constraints with respect to y. In addition, $c \in \mathbb{R}^{n_1}$ and $b \in \mathbb{R}^{m_1}$ are vectors, and A is a real-valued $m_1 \times n_1$ matrix. We let $f : X \times Y \to \mathbb{R}$ and $g : X \times Y \to \mathbb{R}^{m_2}$ be two real-valued functions that are not necessarily linear. As such, the above problem is inherently hard to solve using conventional methods due to nonlinearity and the mixed integer variables.

This study contributes to the current simulation and optimization fields in the following ways. First, we develop a novel decomposition method that infuses simulation directly into optimization to reduce the duality gap between a nonlinear model and its relaxation. Second, we apply the proposed approach in a proof-of-concept study to solve a multi-depot vehicle routing problem (MDVRP) that faces challenges with conventional methods. Third, we conduct comprehensive experiments and sensitivity analyses to benchmark our approach and to show the benefit of simulation in real-world systems. In the following, Section 2 discusses the generic methods and algorithms. Section 3 applies the developed techniques to MDVRP, including the implementations of decomposition, simulation models, and cutting planes. Section 4 shows data, experiments, and results, including benchmarking, sensitivity analyses, and real-world simulations. Section 5 summarizes the findings and provides future research directions.

2 METHOD

We propose a generic solution approach for problem (1) by surrogating the nonlinear functions and some of the mixed integer variables with simulation. Then, we approximate the nonlinear regions with novel cutting plane methods. Near-optimal solutions are reached through an iterative optimization algorithm by continuously adding cutting planes. The approach is illustrated in Figure 1.

To separate linear and nonlinear components in (1), we define a master problem (MP) as

$$\min_{x\in X} \quad \left\{c^T x + \mathscr{Q}(x) : Ax = b\right\}, \tag{2}$$

where $\mathscr{Q}(\bar{x})$ is the subproblem (SP) for a given $\bar{x}$, i.e.,

$$\mathscr{Q}(\bar{x}) = \min_{y\in Y} \quad \left\{f(\bar{x},y) : g(\bar{x},y) \geq 0\right\}. \tag{3}$$

The above partitioning method has been widely considered in the literature, such as the regular and the generalized Benders decomposition (Benders 1962; Geoffrion 1972). The central idea of existing methods is to formulate tight linear approximations of $\mathscr{Q}(\bar{x})$ through the strong duality theory. Thus, the pitfall of

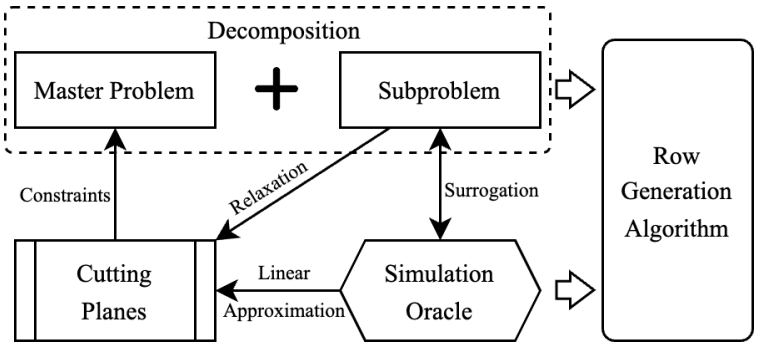


Figure 1: A demonstration of the proposed simulation-infused decomposition approach.

using established methods to solve the partition (2) and (3) is that strong duality does not always hold for the SP with integer variables and nonlinear constraints, resulting in low-quality solutions or failed convergence.

In this study, we develop a novel approach to reduce the duality gap by infusing simulation directly into the formulation of a cutting plane. Note that we primarily focus on optimality cuts over feasibility cuts. In case of infeasible MP solutions, existing feasibility cuts such as no-good cuts and combinatorial Benders cuts can be readily integrated into our method (Codato and Fischetti 2006). Let $\Omega : X \times Y \to \mathbb{R}$ be an "oracle" function, i.e., a simulation surrogate model, where given $\bar{x}$ and $\bar{y}$, $\Omega(\bar{x},\bar{y})$ closely approximates the SP objective function $f(\bar{x},\bar{y})$. We then define an operator $\tilde{\Omega}(\bar{x})$ that optimizes $\Omega(x,y)$ over the Y space:

$$\tilde{\Omega}(\bar{x}) = \min_{y} \quad \Omega(\bar{x},y) \approx \min_{y\in Y} \quad \big\{f(\bar{x},y) : g(\bar{x},y) \geq 0\big\} = \mathscr{Q}(\bar{x}).$$

Essentially, $\tilde{\Omega}(\bar{x})$ defines an optimization via simulation approach, which a large body of literature has already addressed (Amaran et al. 2016). Most commonly, heuristic algorithms are adopted where the solution y is iteratively improved through Monte Carlo simulations.

Next, we introduce two important assumptions to our method.

Assumption 1 The function $g(x,y)$ can be written in the form $g(x,y) = g'(y) - h(x,y)$, where $h(x,y)$ is linear in x for a given y.

Assumption 2 The dual multipliers π of the constraint $g(\bar{x},y) \geq 0$ can be obtained from the relaxed SP

$$\mathscr{L}(\bar{x}) = \max_{\pi} \quad \min_{y\in Y} \quad \big\{f(\bar{x},y) - \pi \cdot g(\bar{x},y)\big\}. \tag{4}$$

Assumption 1 ensures that the generated cutting planes are linear through the function $h(x,y)$. Note that $g'(y)$ can be nonlinear as the simulation will resolve the computation. Note that when $h(x,y)$ is nonlinear in x, the proposed approach also applies, but requires solving a nonlinear MP (2) using existing methods. Assumption 2 is easily satisfied as many algorithms have been proposed to obtain dual multipliers of linear and nonlinear systems. As such, we will use the dual multipliers π to formulate the cutting plane. Let $h(x,\bar{y})$ be the linear function of x in $g(x,y)$, excluding $g'(y)$ that contains stand-alone terms of y. Define $\theta \in \mathbb{R}$ as the variable that approximates $\mathscr{Q}(\bar{x})$. We generate cutting planes of the form

$$\theta \geq \pi^T \cdot h(x,\bar{y}) + \tilde{\Omega}(\bar{x}) - \pi^T \cdot h(\bar{x},\bar{y}). \tag{5}$$

We design a row generation algorithm, namely the simulation-infused decomposition (SID) algorithm, to iteratively add cuts to the MP. Initially, we generate an MP solution $\bar{x}$ without considering the SP. Then, we optimize the simulation oracle $\Omega(\bar{x},y)$ and the relaxed problem (4) to obtain an approximated optimal objective value $\tilde{\Omega}(\bar{x})$ and the multipliers π. Cut (5) is generated and added to the MP. The MP is then re-optimized with the added cut. The process continues until some termination condition is met, e.g.,

$\theta \geq \tilde{\Omega}(\bar{x})$ or a certain number of iterations is reached. Suppose that $k = 1, \ldots, K$ cuts are added to the MP sequentially. The final MP to be solved is

$$\begin{aligned} \min \quad & c^T x + \theta \\ \text{s.t.} \quad & Ax = b, \\ & \theta \geq \pi_k^T \cdot h(x, \bar{y}_k) + \tilde{\Omega}(\bar{x}_k) - \pi_k^T \cdot h(\bar{x}_k, \bar{y}_k) \quad \forall\ k = 1, \ldots, K, \\ & x \in X, \quad \theta \text{ unrestricted}, \end{aligned}$$

which yields the MP objective value, the MP solution x^*, and the SP objective θ^*. The SP solution is given by identifying $y^* = \arg\min_y \{\Omega(x^*, y) : \tilde{\Omega}(x^*) = \theta^*\}$ from the heuristic method.

Further, we improve the SID algorithms by introducing the concept of "learning rates". From preliminary experiments, we find that cuts of form (5) usually converge slowly due to the nonlinearity. To encourage convergence, we define $0 < \alpha < 1$ to reduce the benefit of exploring the X space via the following α-cut:

$$\theta \geq \alpha \pi^T \cdot h(x, \bar{y}) + \tilde{\Omega}(\bar{x}) - \alpha \pi^T \cdot h(\bar{x}, \bar{y}). \tag{6}$$

When α approaches 1, cut (6) becomes identical to cut (5), which leads to the conventional way of generating cuts. When α approaches 0, the constraints approach $\theta \geq \tilde{\Omega}(\bar{x})$ no matter the value of x, thus reducing the algorithm's desire to further explore the X domain. In this case, the final solution $\bar{\theta}$ may be an overestimate of a better solution that is already visited by the algorithm. Thus, at convergence, the best solution is identified from the history: $(x^*, y^*) = \arg\min_{(x^k, y^k)} \{c^T x^k + \Omega(x^k, y^k) : \tilde{\Omega}(x^k) \leq \bar{\theta}, k = 1, \ldots, K\}$, where K is the total algorithm iterations. We summarize the SID algorithm with and without the learning rates in Algorithm 1.

Algorithm 1: The simulation-infused decomposition (SID) algorithm.

```
Initialize the MP (2) and the history H;
k ← 0;
Repeat
    Solve the MP to obtain x̄^k and θ̄^k;
    Optimize the simulation oracle Ω̃(x̄^k) using heuristics methods to obtain ȳ^k;
    H ← H ∪ {(x̄^k, ȳ^k, θ̄^k, Ω̃(x̄^k))};
    if termination condition met then
        break;
    end
    Solve the relaxed SP 𝒮(x̄^k) to obtain the dual multipliers π;
    Add cut (5) or (6) to the MP;
    k ← k + 1;
end
Obtain the final solution (x*, y*) from H;
```

Here, we borrow the idea of "exploration versus exploitation" from the machine learning community to optimization and simulation. By tuning the learning rate in the α-cut, we are able to find a balance between solution quality and convergence speed. This improvement benefits the row generation algorithm as a whole. When more cuts are added, the MP, which is very likely an MIP, becomes harder to solve. The α-cut would avoid adding intractably many cuts by limiting the algorithm exploration, but at the same time still improve the MP solutions by exploiting the simulated information.

Note that our method shares similarities with LBBD and a recently proposed branch-and-check algorithm (Hooker and Ottosson 2003; Forbes et al. 2024). Yet, our approach is more streamlined through an iterative row generation algorithm and does not require the prior conversion of integer variables to binary ones, which may be costly for large-scale problems. Our method also emphasizes on the integration of optimization with

gray/black box simulation oracles, generalizable to a broad range of approximate optimization algorithms and state-of-the-art simulation software.

3 NUMERICAL STUDY

MDVRP has been well investigated in the literature (Jayarathna et al. 2021). Most recent studies focus on decomposition and heuristic approaches to minimize the total travel time and operational costs (Zhen et al. 2020; Şahin and Yaman 2022). Through MDVRP, we conduct a proof-of-concept study to validate the proposed simulation-infused decomposition approach. MDVRP is chosen for several seasons. First, MDVRP can be explicitly formulated and solved using MIP, enabling benchmarking. Its NP-hardness allows us to showcase the computational advantage of our approach. Second, MDVRP benefits greatly from geographic information systems (GIS) embedded in simulation frameworks, which calculate traveling distances more accurately, using real-world road networks instead of the Euclidean or Manhattan distance. Third, we formulate MDVRP using a two-stage structure, allowing easy application of Algorithm 1. This also provides a convenient framework to expand MDVRP with uncertain parameters in future studies, where dynamic decisions are made in stochastic environments.

We consider an urban delivery problem where a number of hubs dispatch vehicles to visit customers. Let $G = (V, E)$ be a complete undirected graph representing the delivery network. The set of nodes $V = V^h \cup V^c$ consists of both hubs V^h and customers V^c. Before the delivery, every customer is first assigned to a hub. Each hub $j \in V^h$ will dispatch one vehicle, with volume capacity $\bar{v}_j$, to visit a subset of customers. Each customer $i \in V^c$ owns a parcel with volume v_i. We set a nominal cost of $c_{i,j}$ for assigning customer i to hub j. Then, vehicles visit the subset of assigned customers in a traveling salesman fashion. The travel times between nodes are denoted by $\tau_{i,j}$, $\forall\ i, j \in V$.

We let $x_{i,j} \in \{0,1\}$, $\forall\ i \in V^c, j \in V^h$, be the decision-making variable for customer assignment, where $x_{i,j} = 1$ suggests that vehicle j must visit customer i during the delivery and $x_{i,j} = 0$ otherwise. We let $y_{h,i,j} \in \{0,1\}$, $\forall\ h \in V^h$, $i, j \in V^c \cup \{h\}$, $i \neq j$, be the routing variable of vehicle h and $u_{h,i} \in \mathbb{Z}_+$, $\forall\ h \in V^h$, $i \in V^c$, be the order of visit to eliminate subtours. The objective is to minimize the customer assignment cost and the total travel time $t \in \mathbb{R}_+$. Note that the total travel time t is the longest time taken among all vehicles. The mixed integer model can be formulated as follows:

$$\min \quad \sum_{i \in V^c, j \in V^h} c_{i,j} x_{i,j} + t \tag{7}$$

$$\text{s.t.} \quad \sum_{j \in V^h} x_{i,j} = 1 \quad \forall\ i \in V^c, \tag{8}$$

$$\sum_{i \in V^c} v_i x_{i,j} \leq \bar{v}_j \quad \forall\ j \in V^h, \tag{9}$$

$$t \geq \sum_{i,j \in V^c \cup \{h\}, i \neq j} \tau_{i,j} y_{h,i,j} \quad \forall\ h \in V^h, \tag{10}$$

$$y_{h,i,j} \leq x_{j,h} \quad \forall\ h \in V^h, i \in V^c \cup \{h\}, j \in V^c, \tag{11}$$

$$\sum_{h \in V^h, j \in V^c \cup \{h\}, j \neq i} y_{h,j,i} = 1 \quad \forall\ i \in V^c, \tag{12}$$

$$\sum_{i \in V^c} y_{h,i,h} = \sum_{k \in V^c} y_{h,h,k} \quad \forall\ h \in V^h, \tag{13}$$

$$\sum_{i \in V^c \cup \{h\}, i \neq j} y_{h,i,j} = \sum_{k \in V^c \cup \{h\}, k \neq j} y_{h,j,k} \quad \forall\ h \in V^h, j \in V^c, \tag{14}$$

$$u_{h,i} - u_{h,j} + 1 \leq |V^c| \cdot (1 - y_{h,i,j}) \quad \forall\ h \in V^h, i, j \in V^c, i \neq j, \tag{15}$$

$$x_{i,j} \in \{0,1\} \quad \forall\ i \in V^c, j \in V^h, \tag{16}$$

$$y_{h,i,j} \in \{0,1\} \quad \forall\ h \in V^h, i, j \in V, i \neq j, \quad u_{h,i} \in \mathbb{Z}_+ \quad \forall\ h \in V^h, i \in V^c, \quad t \in \mathbb{R}_+. \tag{17}$$

In the formulation, constraints (8) and (9) assign customers to hubs within the volume limits. Constraint (10) calculates the maximum delivery time among all hub vehicles. Constraint (11) couples the assignment and delivery stages. Constraints (12)-(15) define the route of each vehicle and eliminate subtours. Note that in MDVRP, the variable y is different from y in (3), which is the only SP variable. In MDVRP, we use a tuple (y,u,t) to represent the SP variables. As such, the function $f(x,y,u,t) = t$ and the function $g_{h,i,j}(x,y,u,t) = x_{j,h} - y_{h,i,j}$, $\forall\ h \in V^h, i \in V^c \cup \{h\}, j \in V^c$.

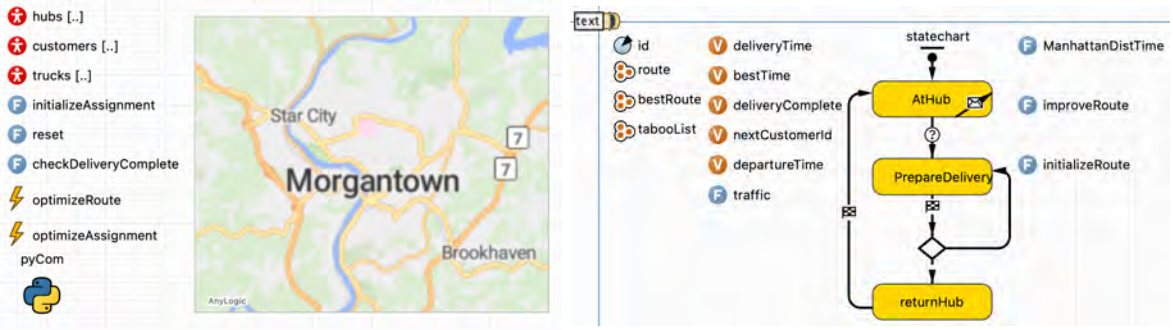
In the MDVRP, there is a natural separation between the assignment decisions and the routing decisions. We decompose the MDVRP into the corresponding stages. Let $\mathcal{Q}(x)$ be the second stage (i.e., SP) objective function. The first stage (i.e., MP) includes constraints for customer assignment:

$$\min_{x} \sum_{i \in V^c, j \in V^h} c_{i,j} x_{i,j} + \mathcal{Q}(x), \quad \text{s.t. } (8)-(9), \text{ and } (16).$$

The second stage SP reconciles vehicle routing decisions for a given $\bar{x}$:

$$\mathcal{Q}(\bar{x}) = \min_{t,y,u} t, \quad \text{s.t. } y_{h,i,j} \le \bar{x}_{j,h} \quad \forall\ h \in V^h, i \in V^c \cup \{h\}, j \in V^c, (10), \text{ and } (12)-(17). \tag{18}$$

According to Algorithm 1, we solve the MP as an MIP and optimize the second stage model through simulation. We construct an urban delivery simulation environment as the oracle using the AnyLogic software (version 8.8.6) (AnyLogic. 2024a). The first stage MP is solved using Gurobi (version 11.0.0) via the Python interface. Communications between AnyLogic and Python are realized through the Pypeline application programming interface (API) (AnyLogic. 2024b). Agent-based modeling is used to build the simulation model. Four types of agents, `Main`, `Hub`, `Customer`, and `Truck` are defined. Since the simulation model primarily optimizes vehicle routing, statecharts are used in the `Truck` agent to ensure solution feasibility. Figure 2 shows the key components of the simulation model.



(a) The `Main` agent

(b) The `Truck` agent

Figure 2: Key components of the simulation model.

To implement Algorithm 1, we first solve the MP to obtain $\bar{x}$. Then, we construct an initial solution for each vehicle $h \in V^h$ in the SP by randomly shuffling the set $\{i \in V^c : x_{i,h} = 1\}$. The route is optimized using a simple neighborhood search heuristic algorithm through repeated simulations. The neighborhood search algorithm is demonstrated in Algorithm 2. The algorithm eventually returns the optimized objective $\bar{\mathcal{Q}}(\bar{x})$ after K iterations.

After obtaining the simulation results, we construct optimality cuts for the MP. As discussed, a wide range of cutting plane methods have been developed in the literature and can be used to implement cuts (5) and (6). Due to the complex nature of the SP, i.e., mixed integer variables, there is no guarantee that any cutting plane method would reach optimality. Here, we consider three families of cuts that can accommodate integer variables: the Benders cut (Benders 1962), the Lagrangian cut (Zou et al. 2019), and the integer cut (Laporte and Louveaux 1993).

Algorithm 2: The neighborhood search algorithm to optimize vehicle routes.

Input: an initial route $r = (r_1, r_2, \dots)$ for truck $h \in V^h$, $r_i \in V^c$ and $x_{r_i,h} = 1$;
Initialize the incumbent objective $z = 0$, best objective $\bar{z} >> 0$, best route $\bar{r} = r$ and total iterations K;
for $k = 1, \dots, K$ **do**
    $z \leftarrow$ `simulate`(r);
    if $z \leq \bar{z}$ **then**
        $\bar{r} \leftarrow r$, $\bar{z} \leftarrow z$;
    end
    $r \leftarrow \bar{r}$;
    Randomly select $\bar{r}_i$ and $\bar{r}_j$, $\bar{r}_i \neq \bar{r}_j$;
    $r_i \leftarrow \bar{r}_j$, $r_j \leftarrow \bar{r}_i$;
end
Output: the optimized route $\bar{r}$ and objective $\bar{\mathscr{Q}}(\bar{x}) = \bar{z}$.

Let $\theta \in \mathbb{R}$ be the approximation of $\mathscr{Q}(x)$. For the Benders and Lagrangian cus, we obtain an estimate of the dual multipliers using the LP relaxation of the SP. We use λ to denote the dual multipliers for constraints (18). The Benders cut is formulated as

$$\theta \geq \alpha \sum_{h \in V^h, i \in V^c \cup \{h\}, j \in V^c} \lambda_{h,i,j}(x_{j,h} - \bar{x}_{j,h}) + \bar{\mathscr{Q}}(\bar{x}). \tag{19}$$

For the Lagrangian cut, a slightly different reformulation is needed by replicating $x_{i,j}$ with another variable $\xi_{i,j}$, $0 \leq \xi_{i,j} \leq 1$. The constraints, $x_{i,j} = \xi_{i,j} \quad \forall\, i \in V^c, j \in V^h$, is added to the SP. Let μ be the dual multipliers for the above replication constraints and $\bar{\xi}$ be the obtained variable. The Lagrangian cut is formulated as

$$\theta \geq \alpha \sum_{i \in V^c, j \in V^h} \mu_{i,j}(x_{i,j} - \bar{\xi}_{i,j}) + \bar{\mathscr{Q}}(\bar{x}). \tag{20}$$

The integer cut does not require solving the LP relaxation of SP. Let L be a lower bound to θ and S be the number of $x_{i,j} = 1$. Define $w_{i,j} = 1$ if $x_{i,j} = 1$ and $w_{i,j} = -1$ if $x_{i,j} = 0$. The integer cut is

$$\theta \geq (\bar{\mathscr{Q}}(\bar{x}) - L) \sum_{i \in V^c, j \in V^h} \alpha \cdot w_{i,j} x_{i,j} - (\bar{\mathscr{Q}}(\bar{x}) - L) \cdot \alpha \cdot (S-1) + L. \tag{21}$$

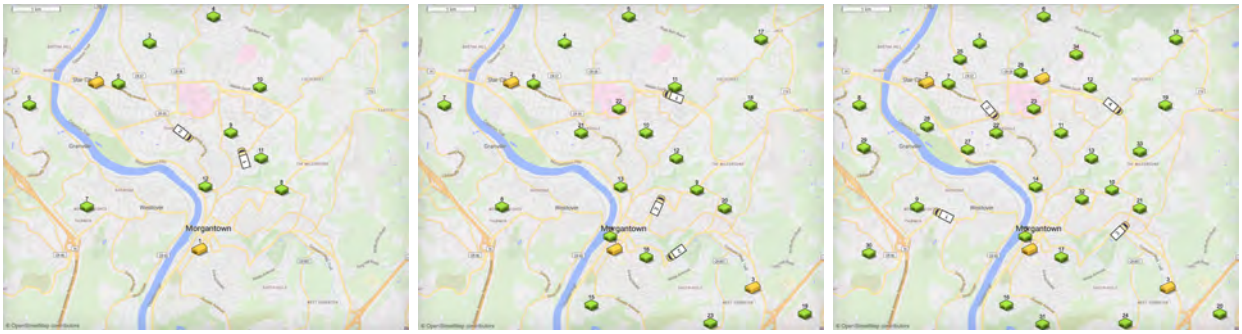
Note that since the integer cut does not require dual multipliers, the learning rate α is added as an additional coefficient to x.

4 DATA & EXPERIMENTS

We collect real-world GIS data from the city of Morgantown, West Virginia. Three instances are constructed with 12 ($|V^h| = 2, |V^c| = 10$), 23 ($|V^h| = 3, |V^c| = 20$), and 34 ($|V^h| = 4, |V^c| = 30$) nodes. GIS coordinates of local post offices are used as hubs. Customers are randomly selected from local schools, hospitals, commercial buildings, apartment complexes, and residential houses. Figure 3 shows the GIS information in the simulation model. Customer package volumes are randomly sampled using a uniform distribution $\mathscr{U}(10, 70)$. Vehicle capacities are randomly sampled using a uniform distribution $\mathscr{U}(250, 650)$. Customer assignment costs are set to be $c_{i,j} = 1$, $\forall\, i \in V^c, j \in V^h$.

4.1 Benchmarking

We compare the performance of the proposed approach with the benchmark, i.e., solving the MIP formulation (7)-(17) directly. Since we cannot incorporate the real-world road system directly into the algebraic MIP model, we set the travel time in the simulation to be identical to the MIP, calculated using



(a) The 12-node instance. (b) The 23-node instance. (c) The 34-node instance.

Figure 3: GIS locations of three testing datasets.

the Manhattan distance divided by the vehicle speed of 35 miles per hour. Note that through the following proof-of-concept experiments, we aim to show the advantages of the SID algorithm in optimizing nonlinear systems and its computational tractability, instead of claiming that SID is the fastest algorithm for a particular MDVRP, which many studies have achieved through heuristics (Jayarathna et al. 2021).

The MIP model (7)-(17) is solved directly using Gurobi. Model parameters are set as default. A solution time limit of 3600 seconds is imposed on all three instances. Hyperparameters of the SID algorithm are shown in Table 1. Hyperparameters are tuned through a simple grid search to select the best-performing sets. Note that for the large 34-node instance, we set the `MIPGap` of the MP to be 10%. That is, we only find near-optimal solutions for the MP to accelerate convergence. The random seed is set to 1 in the simulation model to ensure repeatability.

Table 1: Hyperparameters of the SID algorithm used in benchmarking.

Instance	K	Cuts	α	`MIPGap`	`Presolve`
12-node	750	Benders	$\frac{1}{5}$	1×10^{-4}	-1
23-node	1500	Benders	$\frac{1}{15}$	1×10^{-4}	-1
34-node	2500	Benders	$\frac{1}{25}$	1×10^{-1}	2

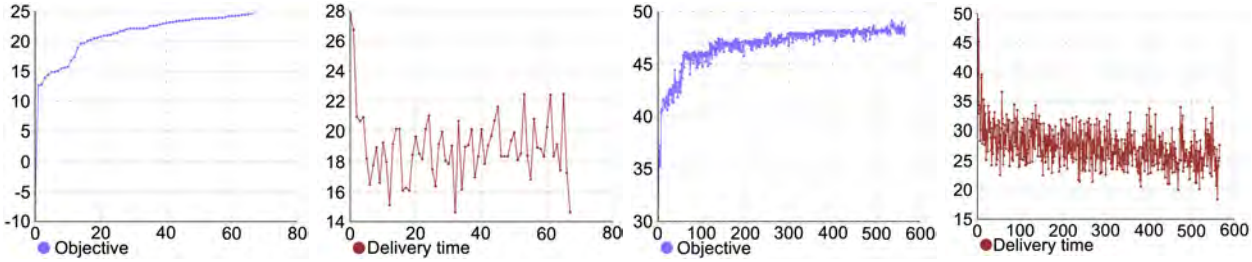
The comparison between MIP and the SID algorithm is shown in Table 2. For the small instance, i.e., 12-node, the SID algorithm is able to find the optimal solution, but takes longer to converge due to its iterative nature. For larger instances, the SID algorithm converges 67.11-69.58% faster than the MIP, with 2.69-8.82% gaps in the objective value (less than 4 minutes of delivery time). Overall, the SID algorithm shows competitive performance against state-of-the-art commercial solvers.

In addition, in Table 2, we compare SID with a recently proposed LBBD method in the literature (Forbes et al. 2024). The LBBD method is run for the same time as SID to compare the objective values. Results show that SID outperforms significantly for all instances, since SID utilizes more information from the dual problem, i.e., the multipliers, than the LBBD method, which is based on the no-good cuts and only includes the simulated objective $\tilde{\Omega}(\bar{x}^k)$.

Table 2: Performance comparisons between MIP, LBBD, and the SID algorithm.

Instance	MIP		SID				LBBD (Forbes et al. 2024)	
	Objective	Runtime (s)	Objective	Gap (%)	Runtime (s)	Impr. (%)	Objective	Runtime (s)
12-node	24.49	0.52	24.49	0.00	14.00	–	32.96	14.00
23-node	36.81	1160.28	37.80	2.69	353.00	**69.58**	52.65	353.00
34-node	44.44	3600.00	48.36	8.82	1184.00	**67.11**	68.02	1184.00

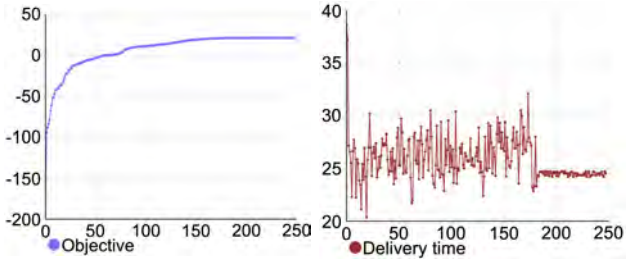
Figure 4 illustrates the detailed convergence for the 12- and 34-node instances. When `MIPGap` in MP is set to 1×10^{-4}, i.e., negligible, for the 12-node instance, each added cut is valid in the sense that it provides a better lower bound of $\mathscr{Q}(x)$. For the 34-node instance, a non-zero MIP gap allows faster solutions, leading to more thorough explorations over the MP feasible region. It also results in fluctuations in the cut qualities, but the objective still improves over time. For both instances, the SP objectives estimated from the oracle $\tilde{\Omega}(\bar{x})$ trend downwards, indicating that better solutions are found as the algorithm progresses.



(a) MP objective, 12-node (b) $\tilde{\Omega}(\bar{x})$, 12-node (c) MP objective, 34-node (d) $\tilde{\Omega}(\bar{x})$, 34-node

Figure 4: The MP objectives and the simulation oracle $\tilde{\Omega}(\bar{x})$ during the iterations.

Figure 5 shows the behavior of Benders cuts using the relaxed problem $\mathscr{L}(\bar{x})$ instead of the simulation oracle $\tilde{\Omega}(\bar{x})$. Due to the duality gap in $\mathscr{L}$, the algorithm cannot guarantee convergence, as θ stops improving after a while (Figure 5a). The MP loops within a set of suboptimal solutions, as illustrated by $\tilde{\Omega}(\bar{x})$ (Figure 5b), since the same constraints are repeatedly added to the MP without improvements.



(a) MP objective (b) $\bar{\mathscr{Q}}(\bar{x})$

Figure 5: The MP objectives and the simulation oracle $\tilde{\Omega}(\bar{x})$ for the 12-node instance using Benders cuts with $\mathscr{L}(\bar{x})$. In this case, the simulation is run independently from the optimization to evaluate the solutions.

Comparing Figures 4 and 5, for the SID algorithm to perform as intended, it is crucial to have a simulation oracle that reduces the duality gap in the cutting planes. As such, we have shown that the SID algorithm converges to near-optimal, sometimes optimal, solutions within tractable computation time. Under moderate assumptions, the SID algorithm is generalizable to a broad class of problems, where nonlinear systems are surrogated by linear approximations to iteratively improve the solutions.

4.2 Real-World Road Network

In this experiment, we show the benefit of infusing simulation into optimization. Instead of calculating distances between customers using the idealistic Manhattan distance, we simulate the distances and travel times using a real-world road network provided in AnyLogic by Open Street Map (OpenStreetMaps. 2024). The vehicle speed is still set to be 35 miles per hour. We solve the 12-node, 23-node, and 34-node instances and compare the solutions with those obtained from using the Manhattan distance. Figure 6 shows a

comparison between the solutions for the 12-node instance. We can immediately observe that the solution from real-world road networks differs significantly from the Manhattan distance.

```
Assignment: {1=[7, 8, 9, 11, 12], 2=[3, 4, 5, 6, 10]}
Route of truck 1: [12, 9, 11, 8, 7]
Route of truck 2: [6, 5, 10, 4, 3]
```

(a) Solution from Manhattan distance.

```
Assignment: {1=[7, 8, 9, 10, 11, 12], 2=[3, 4, 5, 6]}
Route of truck 1: [7, 8, 10, 9, 11, 12]
Route of truck 2: [6, 3, 4, 5]
```

(b) Solution from real-world road network.

Figure 6: Solutions to the 12-node instance, compared between the Manhattan distance and the real-world road network in simulation.

Further, we simulate the solutions obtained from the Manhattan distance-based MIP in the real-world road network environment. Table 3 shows the results. The runtimes for the real-world instances are longer than those using the Manhattan distance, due to the extra cost of running simulation on a GIS map. There is a 2.45-8.02% difference in the total delivery time between the two types of solutions. That is, without the simulation environment, solutions obtained from conventional distance-based methods lead to worse delivery time due to inaccuracies in distance estimations. This gives the SID algorithm a unique advantage to incorporate nonlinearities into optimization without drastically increasing the computational complexity.

Table 3: Comparisons between solutions from the Manhattan distance and the real-world road network.

Instance	Manhattan Distance			Real-World Network		
	$\bar{\mathcal{Q}}(\bar{x})$ from MIP	Simulated $\bar{\mathcal{Q}}(\bar{x})$	Gap (%)	$\tilde{\Omega}(\bar{x})$	Impr. (%)	Runtime (s)
12-node	14.49	21.70	49.76	19.96	8.02	35.00
23-node	16.81	27.33	62.58	26.66	2.45	1631.00
34-node	14.44	27.64	91.41	26.03	5.82	3423.00

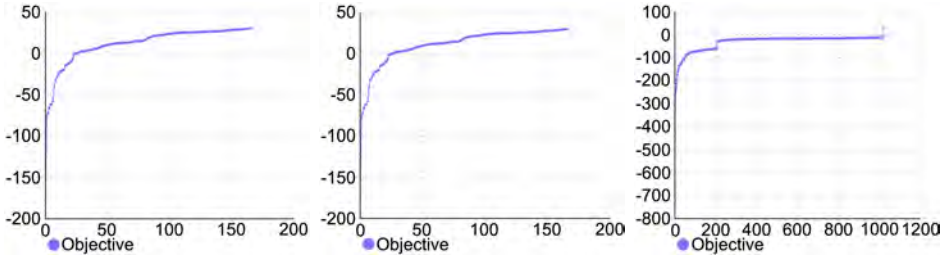
4.3 Sensitivity Analysis

In this experiment, we showcase the impact of hyperparameters in the SID algorithm. Particularly, we focus on the learning rate α and the type of cuts. As a novel concept introduced in this study, the learning rate α controls "exploration" and "exploitation" during the iterative evaluations of MP solutions. We investigate the sensitivity of the convergence rate and solution quality under different learning rates and types of cuts. We solve the 12-node instance under different hyperparameters to avoid intractably long computational times. Table 4 shows the final objective values and the iterations taken to converge under the Benders cut (19), Lagrangian cut (20), and integer cut (21), with $\alpha \in \{0.1, 0.2, 1\}$.

Table 4: Sensitivity analysis on the objective values and the convergence iterations over different cuts and α.

Cut	Objective Values			Iterations		
	$\alpha = 1$	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 0.2$	$\alpha = 0.1$
Benders	29.97	30.37	35.61	166	59	6
Lagrangian	29.97	30.12	35.60	167	59	6
Integer	30.12	30.35	29.56	1022	300	42

In general, as α approaches 1, the algorithm takes significantly longer to converge, but the final objective value improves. This is because the α-cuts allow for more explorations in the X domain. As α decreases, the algorithm converges faster but the solution quality reduces, suggesting that the generated cuts are more aggressive. This trade-off provides practitioners with the flexibility to adjust the hyperparameters for unique problem characteristics. In addition, the Benders cut behaves similarly to the Lagrangian cut, whereas the integer cut converge much slower, but eventually find the best solution among all.



(a) Benders cut (b) Lagrangian cut (c) Integer cut

Figure 7: The convergence of cuts (19), (20), and (21) with $\alpha = 1$.

In addition, we show the convergence steps with $\alpha = 1$ in Figure 7, for cuts (19), (20), and (21). Comparing Figures 7a and 7b, the resemblance of the Benders and Lagrangian cuts in Table 4 can be explained, where both improve the MP in a similar fashion at every iteration. In Figure 7c, the jump at the end for the integer cut signifies that the newly found solution has reduced the gaps between θ and $\bar{\mathcal{Q}}(\bar{x})$.

5 CONCLUSION

In this study, we develop a generic simulation-infused approach to solve decomposition models where nonlinearity in variables and constraints causes intractable computations. We build simulation surrogate models to approximate the optimal SP solutions. Combined with dual multipliers from a relaxation, simulated SP solutions are used to construct cutting planes to the MP, as under-approximations of the true values. As such, integrality and nonlinearity can be partially resolved through simulation. We conduct a proof-of-concept study on an MDVRP to demonstrate the implementations of our method in a real-world problem. Comprehensive experiments are conducted to show significant improvements (over 67%) in the computation time compared with state-of-the-art benchmarks. With the embedded GIS system in simulation, we are able to solve the instances on a real-world road network, where simulation-infused optimization outperforms benchmarking methods by up to 8%. In addition, through sensitivity analysis, we show the impact of hyperparameters such as the type of cut, as well as the learning rate α that expands the scope of cutting plane methods with the trade-offs between exploration and exploitation.

This study paves the way for a number of future research directions. First, our method extends naturally to stochastic problems and can be combined with SAA, where simulation yields a much more accurate uncertainty estimation than scenario-based algebraic models. Second, theoretical studies can be conducted to analyze the tightness of different types of cuts in the simulation environment, the convergence property of the SID algorithm, and additional nonlinear elements in the MP. Third, the SID algorithm can be validated on more classes of problems for its performance as a generic solution approach, such as using simulation to model vehicle-level road traffic, infectious disease transmissions, and stakeholder interactions in multi-echelon supply chains.

REFERENCES

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury. 2016. "Simulation Optimization: A Review of Algorithms and Applications". *Annals of Operations Research* 240:351–380 https://doi.org/https://doi.org/10.1007/s10479-015-2019-x.

AnyLogic. 2024a. *AnyLogic Simulation Software*. https://www.anylogic.com/, accessed 13rd February 2024.

AnyLogic. 2024b. *Python in AnyLogic: Pypeline library*. https://anylogic.help/advanced/code/python.html, accessed 13rd February 2024.

Babamiri, O. and S. Marofi. 2021. "A Multi-Objective Simulation–Optimization Approach for Water Resource Planning of Reservoir–River Systems Based on a Coupled Quantity–Quality Model". *Environmental Earth Sciences* 80(11):389 https://doi.org/https://doi.org/10.1007/s12665-021-09681-9.

Benders, J. F. 1962. "Partitioning Procedures for Solving Mixed-Variables Programming Problems". *Numerische Mathematik* 4(1):238–252 https://doi.org/https://doi.org/10.1007/s10287-004-0020-y.

Codato, G. and M. Fischetti. 2006. "Combinatorial Benders' Cuts for Mixed-Integer Linear Programming". *Operations Research* 54(4):756–766 https://doi.org/https://doi.org/10.1287/opre.1060.0286.

de Paula Ferreira, W., F. Armellini, and L. A. De Santa-Eulalia. 2020. "Simulation in Industry 4.0: A State-Of-The-Art Review". *Computers & Industrial Engineering* 149:106868 https://doi.org/https://doi.org/10.1016/j.cie.2020.106868.

Ferré-Bigorra, J., M. Casals, and M. Gangolells. 2022. "The Adoption of Urban Digital Twins". *Cities* 131:103905 https://doi.org/https://doi.org/10.1016/j.cities.2022.103905.

Forbes, M., M. Harris, H. Jansen, F. van der Schoot, and T. Taimre. 2024. "Combining Optimisation and Simulation Using Logic-Based Benders Decomposition". *European Journal of Operational Research* 312(3):840–854 https://doi.org/https://doi.org/10.1016/j.ejor.2023.07.032.

Geoffrion, A. M. 1972. "Generalized Benders Decomposition". *Journal of Optimization Theory and Applications* 10:237–260 https://doi.org/https://doi.org/10.1007/BF00934810.

Högdahl, J. and M. Bohlin. 2023. "A Combined Simulation-Optimization Approach for Robust Timetabling on Main Railway Lines". *Transportation Science* 57(1):52–81 https://doi.org/https://doi.org/10.1287/trsc.2022.1158.

Hooker, J. N. and G. Ottosson. 2003. "Logic-Based Benders Decomposition". *Mathematical Programming* 96(1):33–60 https://doi.org/https://doi.org/10.1007/s10107-003-0375-9.

Jayarathna, D., G. Lanel, and Z. Juman. 2021. "Survey on Ten Years of Multi-Depot Vehicle Routing Problems: Mathematical Models, Solution Methods and Real-Life Applications". *Sustainable Development Research* 3(1):36–47 https://doi.org/https://doi.org/10.30560/sdr.v3n1p36.

Khalili-Damghani, K., M. Tavana, and P. Ghasemi. 2022. "A Stochastic Bi-objective Simulation–Optimization Model for Cascade Disaster Location-Allocation-Distribution Problems". *Annals of Operations Research* 309(1):103–141 https://doi.org/https://doi.org/10.1007/s10479-021-04191-0.

Lanzarone, E., E. Galluccio, V. Bélanger, V. Nicoletta, and A. Ruiz. 2018. "A Recursive Optimization-Simulation Approach for the Ambulance Location and Dispatching Problem". In *2018 Winter Simulation Conference (WSC)*, 2530–2541 https://doi.org/https://doi.org/10.1109/WSC.2018.8632522.

Laporte, G. and F. V. Louveaux. 1993. "The Integer L-Shaped Method for Stochastic Integer Programs with Complete Recourse". *Operations Research Letters* 13(3):133–142 https://doi.org/https://doi.org/10.1016/0167-6377(93)90002-X.

Li, L., B. Lei, and C. Mao. 2022. "Digital Twin in Smart Manufacturing". *Journal of Industrial Information Integration* 26:100289 https://doi.org/https://doi.org/10.1016/j.jii.2021.100289.

Li, X. and K. Zhang. 2018. "A Sample Average Approximation Approach for Supply Chain Network Design with Facility Disruptions". *Computers & Industrial Engineering* 126:243–251 https://doi.org/https://doi.org/10.1016/j.cie.2018.09.039.

OpenStreetMaps. 2024. *Open Street Maps*. https://www.openstreetmap.org/#map=4/34.78/-84.55, accessed 16th February 2024.

Pylianidis, C., S. Osinga, and I. N. Athanasiadis. 2021. "Introducing Digital Twins to Agriculture". *Computers and Electronics in Agriculture* 184:105942 https://doi.org/https://doi.org/10.1016/j.compag.2020.105942.

Rahmaniani, R., T. G. Crainic, M. Gendreau, and W. Rei. 2017. "The Benders Decomposition Algorithm: A Literature Review". *European Journal of Operational Research* 259(3):801–817 https://doi.org/https://doi.org/10.1016/j.ejor.2016.12.005.

Şahin, M. K. and H. Yaman. 2022. "A Branch and Price Algorithm for the Heterogeneous Fleet Multi-Depot Multi-Trip Vehicle Routing Problem with Time Windows". *Transportation Science* 56(6):1636–1657 https://doi.org/https://doi.org/10.1287/trsc.2022.1146.

Sakki, G. K., I. Tsoukalas, P. Kossieris, C. Makropoulos, and A. Efstratiadis. 2022. "Stochastic Simulation-Optimization Framework for the Design and Assessment of Renewable Energy Systems under Uncertainty". *Renewable and Sustainable Energy Reviews* 168:112886 https://doi.org/https://doi.org/10.1016/j.rser.2022.112886.

Wang, X., G. Jones, and X. Li. 2023. "A Two-Stage Stochastic Model for Drone Delivery System with Uncertainty in Customer Demands". In *2023 Winter Simulation Conference (WSC)*, 1876–1887 https://doi.org/https://doi.org/10.1109/WSC60868.2023.10407713.

Zhen, L., C. Ma, K. Wang, L. Xiao, and W. Zhang. 2020. "Multi-Depot Multi-Trip Vehicle Routing Problem with Time Windows and Release Dates". *Transportation Research Part E: Logistics and Transportation Review* 135:101866 https://doi.org/https://doi.org/10.1016/j.tre.2020.101866.

Zou, J., S. Ahmed, and X. A. Sun. 2019. "Stochastic Dual Dynamic Integer Programming". *Mathematical Programming* 175:461–502 https://doi.org/https://doi.org/10.1007/s10107-018-1249-5.

AUTHOR BIOGRAPHIES

ZEYU LIU is an Assistant Professor in the Department of Industrial and Management Systems Engineering at West Virginia University. He received his Ph.D. in Industrial Engineering from The University of Tennessee, Knoxville in 2022. His research interests include optimization under uncertainty, data-driven analytics, reinforcement learning, and agent-based simulation, with applications in healthcare, infrastructure resilience, energy systems, transportation systems, and space logistics. Liu received the Harvey J. Greenberg Research Award from the INFORMS Computing Society in 2022. He is a member of INFORMS and IISE. His email address is zeyu.liu@mail.wvu.edu and his website is https://directory.statler.wvu.edu/faculty-staff-directory/zeyu-liu.