Available online at : http://josi.ft.unand.ac.id/

Jurnal Optimasi Sistem Industri

| ISSN (Print) 2088-4842 | ISSN (Online) 2442-8795 |

Research Article

Goal Programming and Monte Carlo Simulation for Optimizing Inbound Scheduling in Resource-Constrained Warehouses

Savio Enoson [1], Andreas Handojo [1,*], Tanti Octavia [2], Demetrio Franklyn Edryan [3]

[1] Department of Informatics, Petra Christian University, Surabaya, Indonesia
[2] Department of Industrial Engineering, Petra Christian University, Surabaya, Indonesia
[3] Department of Computer Science, University of British Columbia, Vancouver, Canada

*Corresponding Author: handojo@petra.ac.id

DOI: 10.25077/josi.v24.n2.p324-344.2025 **Submitted:** March 19, 2025 **Accepted:** October 29, 2025 **Published:** December 30, 2025

ABSTRACT

Resource efficiency lies at the heart of logistics performance, with unloading operations in storage facilities serving as a critical determinant of overall productivity. In less developed regions, the widespread reliance on basic rules-based systems such as FIFO often proves inadequate for handling operational complexities, leading to bottlenecks and inefficiencies. Small and medium-sized enterprises (SMEs), constrained by limited resources, are compelled to optimize existing infrastructure rather than invest in costly upgrades. To address this challenge, the present study introduces a goal-oriented programming model designed to assign trucks to loading docks within specific time slots, thereby enhancing time efficiency. The model evaluates performance across four key metrics: waiting time, loading time, overtime, and equity. By leveraging goal programming, numerical prioritization of these objectives becomes possible, enabling flexible adjustments to meet operational needs. Furthermore, Monte Carlo simulation (MCS) is employed to incorporate variability into the dataset and assess model robustness under real-world uncertainty. Experimental results reveal that the proposed approach consistently outperforms traditional systems, delivering significant improvements in time efficiency. These findings highlight the potential of goal programming as a practical solution for planning in resource-constrained environments. The resulting model offers an adaptive, reliable framework that warehouse managers can implement without incurring substantial infrastructure costs.

Keywords: inbound scheduling, time efficiency, goal programming, Monte Carlo simulation, SMEs

INTRODUCTION

Modern supply chains are highly complex and interlinked networks of systems and processes that work together to ensure the timely delivery of goods while minimizing operational costs [1]. This inherent complexity often results in the neglect of certain processes, even when optimization could bring significant efficiencies. One such process is the loading and unloading of goods in storerooms or distribution centers. As enterprises expand, the volume of goods handled increases significantly; however, the scheduling of unloading operations is often overlooked because of its seemingly simple nature. Inefficiencies at this early stage of the supply chain can propagate downstream, causing disruptions throughout the entire system [2]. Although advanced solutions such as Amazon's automated ranking systems [3] have been developed to address these problems, they remain largely unavailable to SMEs and even some larger companies in developing economies. This inaccessibility is mainly attributed to high installation and maintenance costs and to resource constraints such as unstable terrain or limited access to installations by road [4]. As a result, many companies still rely on traditional rules-based approaches, such as FIFO, as their core operating strategy. Though effective to some extent, FIFO fails to consider the broader implications of any one decision, in this case assigning a truck to a loading dock. This can often lead to repeated work, prolonged waiting times, and high overtime. A large body of solutions have been proposed to solve this problem; and as more research is being done on supply chain management, the need for an optimization method within the reach of smaller businesses is an increasingly relevant topic [5].

Recently, advancements in operations research and computational hardware have introduced powerful, easily accessible tools for all manner of supply chain operations. Techniques that are now ubiquitous like linear programming, genetic algorithms, and computer-based simulations used to be out of reach for the average warehouse manager. It required hiring experts or consulting external parties, and what software was available had a high barrier of entry. However, the increased availability and range of software have made these techniques easily available even for non-domain experts. One example is the integration of linear programming and genetic algorithms-based optimization into the Excel solver tool - seemingly simple solutions can lead to significant efficiency gains [6].

Linear programming is an optimization technique in which a mathematical function is defined to maximize or minimize a value within a linear constraint. It is based on mathematics, which makes it a reliable and long-standing method used for many different applications in the field of operational research. LP is particularly well suited for solving problems of resource allocation with clear parameters, such as minimizing transport costs or allocation of labor. However, LP in its most basic form is constrained to single-objective formulas, limiting its range of functionality in scenarios where decisions are trade-offs between competing or contradicting goals. Goal Programming (GP) is an extension of Linear Programming (LP) that facilitates multi-objective optimization by minimizing deviations from predefined target values rather than optimizing the values themselves [7]. This characteristic makes GP particularly suitable for inbound scheduling processes, where multiple, often conflicting objectives must be simultaneously considered and balanced.

Monte Carlo simulation (MCS) is a computational technique which uses repeated random sampling to approximate the variability in a system. Although conceptually simple, MCS is widely used for the model of complex systems characterized by uncertainty in key variables. In this study, MCS is used to complement the goal-oriented model by introducing stochastic variability into time-dependent parameters such as truck arrival delays and equipment breakdowns. A large number of pseudo-randomized data sets are used to approximate the variation of daily operations. This additional dimension of testing is necessary to test the robustness and reliability of the model; to ensure that the model remains reliable in the real world [8]. The process of MCS involves defining ranges of deviations for certain variables, creating different datasets, and then evaluating the performance of the model for each dataset. Results of all iterations are then aggregated and analyzed to assess the performance of the model. The output of the simulation allows the extraction of key statistical measures such as mean, standard deviation and upper and lower control limits (UCL and LCL).

As mentioned above, there is a large body of research on supply chain optimization. In large-scale operations, even small, marginal efficiency gains can result in significant cost reductions. However, the methods examined in previous studies are often difficult to implement in SMEs. The most common scheduling method used by small and medium-sized enterprises is FIFO. Although this is undoubtedly the easiest and most natural choice for start-ups and even for medium-sized companies handling small volumes of freight, it quickly becomes inadequate in the face of any complexity. In particular, the FIFO does not take into account competing objectives, variability and possible system limitations. Due to its upstream position on the network, planning in-bound shipments is of particular importance, as bottlenecks may slow down the whole operation [9].

 DOI: 10.25077/josi.v24.n2.p324-344.2025

Previous studies have examined different techniques for scheduling warehouses. Yu and Co. [10] used a hybrid genetic algorithm to minimise the time to complete the crossing system. Although the results are significant, the method lacks a clear way to address the trade-off between processing time and other factors. On the contrary, Wang et al. [11] addresses a similar problem by means of multi-objective optimisation (MOO). The model manages three competing objectives by finding a front that is the most optimal and which provides similar improvements in time efficiency compared to the other two. The goal programming has the basic characteristics of a MOE, but it provides a more accessible framework for non-specialists and is therefore more relevant to SMEs. Alakayev et al. [12] used GP to balance competing objectives in the scheduling problem, but failed to take into account the variability of key parameters such as arrival and processing times [12]. To close this gap and to test the reliability of the model, the MCS may be used. Although simple in concept, the use of MCS for this purpose has shown positive results, as illustrated by the work of Saeed et al. [13] where MCS is used in the UAV route and delivery system, which allows a more detailed understanding of the results of the model.

Other methods, more aligned with the field of computer science as supposed to mathematics have also been explored. A study by Yong et al. and Kmiecik explores the efficacy of genetic algorithms for optimizing warehouse explorations but discovers that the models is prone towards bias and may not be fully capable of addressing the full complexity of warehouse operations [14, 15]. Another study by Zeng tries to mitigate biases by employing a chaotic genetic algorithm in optimizing automated warehouse logistics but finds that it lacks the capability for multi-objective optimization [16]. Another seldom discussed aspect of these studies is the systems therein often assume that the workers have the ability to make changes on the fly. This is not always the case, especially for SMEs where human resources might be limited. Moreover, while advanced optimization techniques have been proven effective, their applicability in developing economies is limited by the lack of infrastructure and resources [17, 18]. While many studies are exploring more complex systems with a large amount of decision variables, the opposite approach of maximizing existing resources lacks substantive research.

The objective of this study is to address these shortcomings by developing and testing a GP model to optimize the time efficiency of in-sourcing logistics for SMEs. Unlike previous approaches, the proposed model explicitly assumes that existing infrastructure is non-changeable, which is the case most often for SMEs. To compensate for this constraint, the model has to maximize the use of existing resources while balancing several objectives. The optimal solution is inherently sensitive to changes in the expected input variables. To accommodate this, the model introduces a variable arrival and processing time using MCS, which effectively produces a more reliable result, already tested at different stress levels and with clear upper and lower limits.

The proposed framework is intended to provide practical and evidence-based solutions for operational managers in SMEs seeking to increase operational efficiency while maintaining scalability in order to support a more complex operating environment. Moreover, the integration of genetic programming allows non-specialists to adapt and adapt the model without the need for advanced technical expertise. This feature is particularly beneficial for businesses in developing economies, where resource constraints often prevent the adoption of advanced technologies.

METHODS

A baseline setting of storage facilities is essential to the development of the proposed model. Historical data on arrival of trucks and loading times were collected, as well as system constrains such as maximum permissible weight and volume of material handling and operating hours of the facility. This data set has been used to extract the key parameters for the simulation, such as the truck size and the distribution of the load type. However, in practice, the data often fails to capture edge cases, including arrival time spikes, which are critical to the assessment of the robustness of the model. To overcome this limitation, simulated data sets were generated for modelling evaluation under a wide range of possible scenarios [19].

To replicate the arrival patterns of trucks in the real world, a bimodal distribution function was used to approximate the arrival times of each simulated shipment. This function reflects the two peaks that are normally observed in the morning and afternoon. This approach is in line with established findings from research in the field of logistics and supply chain, where distribution models are commonly used to represent load factors and can be adapted to simulate the arrival patterns of warehouses [20]. In this study, a bimodal distribution function, denoted $f(t)$, is constructed by the normalization of two Gaussian functions, $G(t, h, p, s)$, each defined by its peak time, associated probability, and spread parameters.

$$G(t, h, p, s) = p * e^{-\frac{(t-h*60)^2}{2(30s)^2}} \tag{1}$$

G	= Bell curve representing the probability distribution with a given peak time and spread.
t	= Time, defined in minutes past 00:00; ranges from 0 to 1440.
p	= Relative height or amplitude for the peak probability.
h	= Center of a peak, defined in hours past 00:00, then converted to minutes in the formula.
s	= Controls the width of the bell curve, scaled using a standard deviation value of 30 minutes.

In order to mimic real-life arrival patterns, 2 bell curves are combined to create a major and minor peak shown in $g(t)$. This is then normalized by diving $g(t)$ by the single highest value found in the range to obtain the probability of a truck arriving at a given t. The latter equation is shown as $f(t)$.

$$g(t) = G(t, h_1, p_1, s) + G(t, h_2, p_2, s) \tag{2}$$

$$f(t) = \frac{g(t)}{max_{0 \le \tau \le 1440}\, g(\tau)} \tag{3}$$

The resulting bimodal distribution function represents the density of truck arrivals at a given time of day. Figure 1 and Figure 2 shows distributions for arrivals clustering around the afternoon and morning time periods respectively.

To assign trucks to their designated loading docks, this study uses an LP approach that formulates each truck-dock-time pair into binary decision variables. Time efficiency is measured in three key metrics: waiting time, unloading

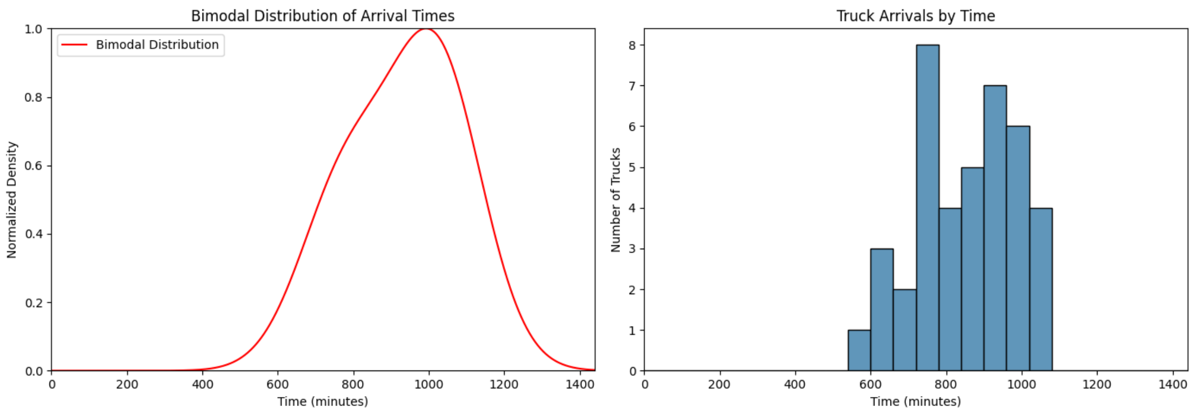

Figure 1. Distribution function for arrivals clustering around afternoon hours.

 DOI: 10.25077/josi.v24.n2.p324-344.2025

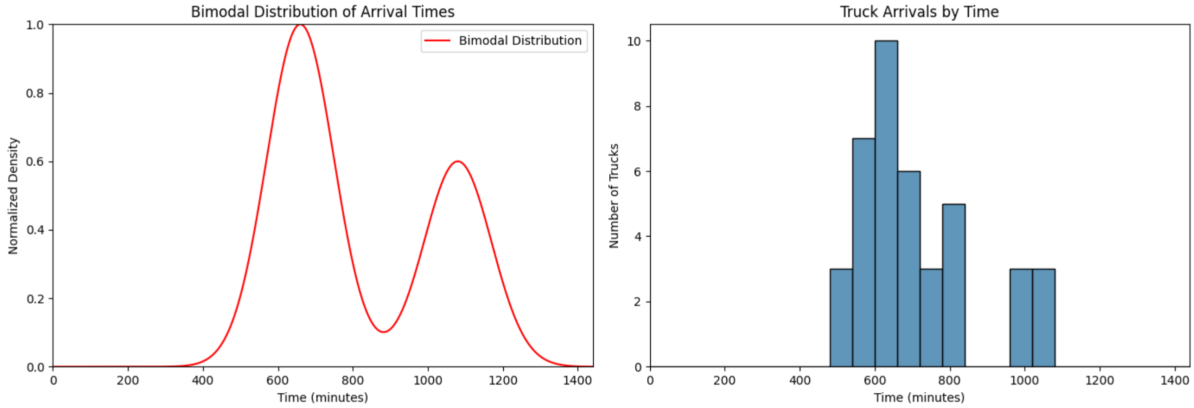

Figure 2. Distribution function for arrivals clustering around morning hours

time, and overtime, each defined as an objective function. A fourth goal; fairness is also defined, which represents the sum of deviations from the average waiting time in each schedule. This goal helps suppress long individual waiting times and make it reasonably fair towards drivers, more closely aligning with real-life ethics. An LP model is first formulated and solved for each model, this will give us the optimal values for said goal, which we will use to construct the GP model—the GP model's objective minimizes the sum of deviations from each goal multiplied by their respective weights.

This weight is determined using a combination of domain knowledge and statistical analysis. The financial cost associated with each goal is asymmetrical by nature—high total wait or unloading time might not cost the facility as much financially as overtime would. Unloading time is also highly subject to the number of servers and capacity of each server; thus having a much smaller optimal value when compared to the other goals. To gauge each goal's responsiveness towards optimization, a version of the GP model with equal weights is solved using several testing datasets. It is found that waiting time and overtime responds much more towards optimization. These 2 insights are used to determine the weights for the model—where top priority is given to minimizing overtime, with waiting time, fairness, and unloading time following suit. It is noted that more advanced methods of determining these weights should be explored in future studies on a similar topic.

Once solved, model's output can then be interpreted as the start times for each truck in a given dataset, from which the overall time efficiency can be measured. The performance of the resulting schedule is then compared against the FIFO system's performance when handling the same dataset, providing insight into the potential improvement towards the current system. Finally, the model is validated using MCS, which introduces variability in truck arrival and service times to assess the schedule's performance under possible fluctuations from expected processing and arrival times. 100 iterations with increasing levels of deviation are aggregated to obtain results in the form of mean, standard deviation, UCL, and LCL values. Figure 3 outlines the processes involved to create the model in chronological order.

Simulation Environment

The warehouse facility consists of 2 loading docks and 3 warehouses. Goods can be transported from any loading dock to any warehouse; however, the transport times will vary due to distance and available resources. Different types of goods must be transported to different pre-assigned warehouses based on the item's classification—therefore the total expected unloading time for trucks will vary for each loading dock.

Trucks that weigh over 4,000 kilograms must be serviced at loading dock 2 due to equipment constraints. The weight and volume of shipments are calculated based on the size, load percentage, and load type of the truck. The facility's

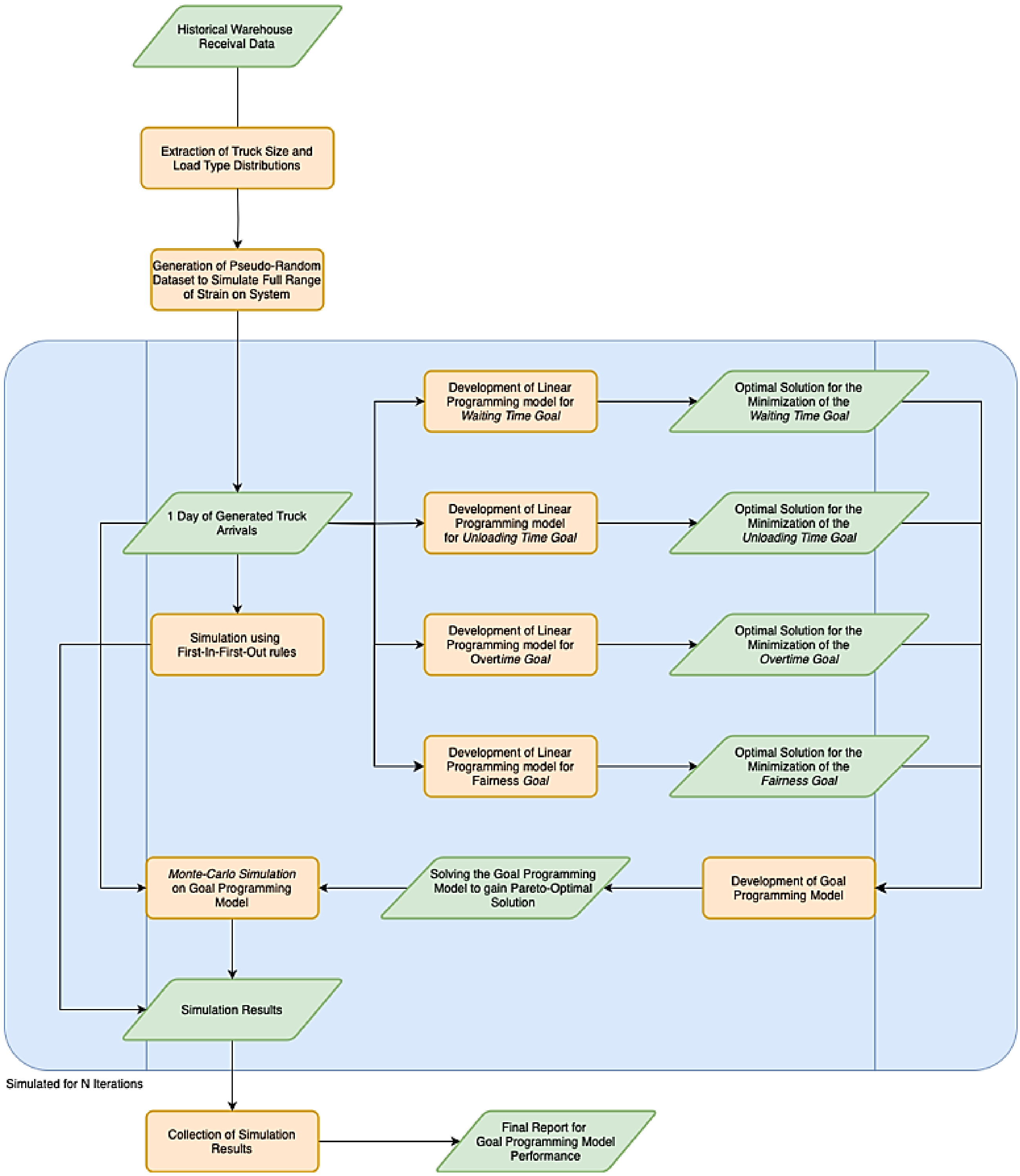

Figure 3. Research Flowchart

capacity to move items in between trucks, loading docks, and warehouses are measured in either liters per minute, or kilograms per minute.

Time is measured in minutes past 00:00, and each simulation instance is 24 hours long, hence there are 1,440 units of time. The facility opens at 08:00 and closes at 18:00 every day. Waiting time for trucks that arrive before opening hours will only be measured starting from 08:00. Every minute every truck spends unloading past 18:00 will count

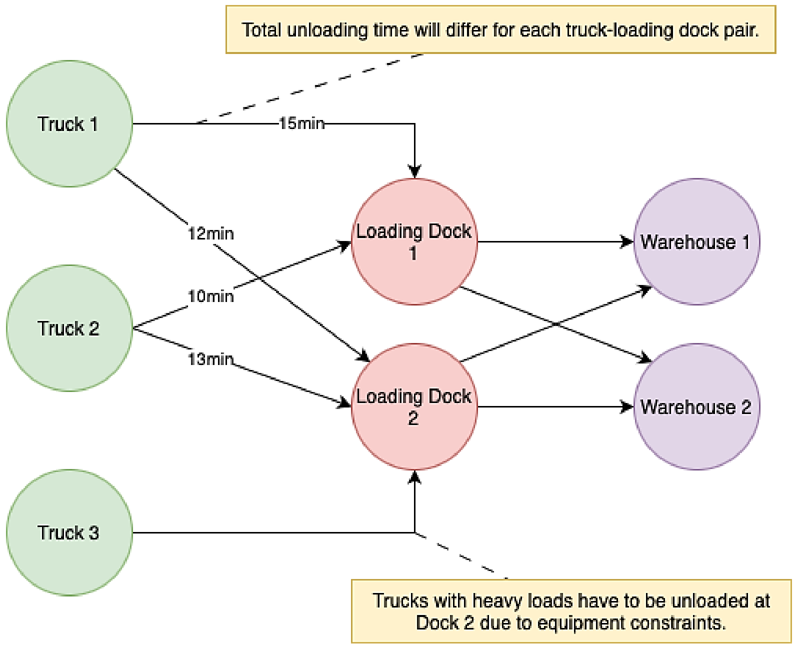

Figure 4. Graph model of warehouse inbound scheduling.

as overtime. Trucks which arrive past opening hours will not be processed in the same day. Standard operating procedures require shipments to pass through checkpoints which inspect manifest records for each truck. Overtime personnel, however, only includes those working directly on the receival process. The simulation reflects this by limiting truck arrivals to before 18:00.

Aside from the stated parameters, the model works under the following assumptions:

- Trucks that are scheduled to arrive will do so within the same day. The model does not account for no-shows on the supplier's end.
- Trucks that arrive past opening hours will be rejected.
- Loading docks can only serve 1 truck at a time regardless of size.

Figure 4 shows a model of the simulated warehouse inbound scheduling process.

Dataset Generation

Using proof of delivery documents from 2024, distributions for truck size and load types are extracted as shown in Table 1 and Table 2. Shipments typically utilize 60% to 100% of a truck's capacity. Based on a truck's size and load attributes, the simulation approximates the weight and volume of goods it is carrying. The formula for the volume and weight of Truck a is expressed as follows:

$$V_a = \max\_volume_a * load_{\%_a} * volume\_modifier_a \quad (4)$$

$$W_a = \max\_weight_a * load_{\%_a} * weight\_modifier_a \quad (5)$$

where

V = Volume

W = Weight

Table 1. Truck Size Distribution

Truck Size	Occurrence	Max. Volume	Max. Weight
Small (S)	15%	4,000	1,200
Medium (M)	25%	6,000	2,750
Large (L)	50%	9,000	4,500
Extra-Large (XL)	10%	14,500	8,000

Table 2. Truck Load Type Distribution

Load Type	Occurrence	Subtype	Subtype Occurrence	Weight and Volume Modifiers
High-Value Goods (HV)	20%	Cigarettes (Cg)	40%	1.0; 0.5
		Soft-Drinks (Sd)	50%	1.0; 0.3
		Miscellaneous (M)	10%	1.0;0.5
Fast-Moving Goods (F)	75%	Light-Snacks (Ls)	50%	0.3; 1.0
		Cookies / Biscuits (Ck)	35%	0.5; 0.7
		Candy (Cd)	15%	0.7; 0.4
Slow-Moving Goods (S)	5%	Apparel (A)	40%	1.0; 0.7
		Shelf-Stable Goods (SS)	40%	1.0; 0.3
		Toys (T)	20%	0.6; 0.7

The total unloading time for a truck-loading dock pair is a sum of the expected amount of time to unload items from the truck and finish transporting the goods to the assigned warehouse based on the item's classification. When calculating unloading times for a truck, the lowest value between volume and weight unloading times will be chosen.

The unloading time of Truck a at Loading Dock b is as follows:

$$U_{ab} = \min\left(\frac{V_a}{Sv_b}, \frac{W_a}{Sw_b}\right) + \min\left(\frac{V_a}{Sv_{b'}}, \frac{W_a}{Sw_{b'}}\right) \tag{6}$$

where

U = Unload rate

Sv = Service rate by volume

Sw = Service rate by weight

b' = Transportation rate from warehouse b to the designated warehouse

With all the parameters set, the previously defined bimodal distribution function is used to generate a list of trucks with unique arrival times, clustered around a certain peak hour. Then, a random number is drawn which determines the truck's size—this process is repeated for the load type and subtype. The weight, volume, and unloading times for each truck are then calculated and appended to the list. The resulting dataset consists of a list of trucks with arrival time, size, load type, volume, weight, and unloading time attributes which roughly align with the given distributions. Table 3 shows a sample of a generated dataset. Different random seeds can be utilized to generate different permutations of a given scenario. Figure 5 shows the difference in size and load type distributions on different random seeds.

To thoroughly test the model's performance, several scenarios with different seeds will be generated for each permutation of the adjustable dataset generation parameters. These include the volume of incoming trucks, the peak

 DOI: 10.25077/josi.v24.n2.p324-344.2025

Table 3. Sample Dataset Header

Truck #	Size	Type	Subtype	Load	Weight	Volume	Arrival Time	Unload Time 1	Unload Time 2
1	M	F	Cd	92.74%	1622.90	1708.77	546	29.00	18.00
2	L	HV	Cg	53.45%	2405.17	2577.93	1078	44.00	28.00
3	S	F	Ck	76.04%	570.34	1452.37	657	10.00	6.00
4	L	HV	Cg	84.53%	3803.82	3720.79	569	69.00	44.00
5	XL	F	Cd	82.15%	4600.37	4246.40	990	84.00	53.00

Table 4. Dataset generation parameters for each testing scenario

Number of Truck	Peak	Spread	Scenario
Low	Morning	High	1-10
		Low	11-20
	Afternoon	High	21-30
		Low	31-40
Medium	Morning	High	41-50
		Low	51-60
	Afternoon	High	61-70
		Low	71-80
High	Morning	High	81-90
		Low	91-100
	Afternoon	High	101-110
		Low	111-120

arrival times, and the spread or distribution of arrival times. This generates a total of 12 scenario types. For each type, 10 individual scenarios will be generated with randomized seeds, with a total of 120 scenarios overall used in the testing. The parameters for each scenario are detailed in Table 4.

Figure 5. Difference in size and type distributions on different random seeds

Goal Programming

Goal Programming (GP) is a multi-objective optimization technique that extends traditional Linear Programming (LP) by allowing decision-makers to balance multiple, often conflicting, objectives [21]. While it is possible to perform the task of time efficiency optimization using a traditional LP model, it fails to account for the complex trade-offs inherent in warehouse operations. For example, minimizing waiting time might lead to increased overtime, while reducing overtime could result in longer maximum wait times. GP addresses this challenge by introducing weighted goals, allowing decision-makers to prioritize objectives based on their relative importance [22]. The goal programming model developed in this study can be formulated as follows:

Objective Function:

$$Min. Z = w_1 * d_w^+ + w_2 * d_u^+ + w_3 * d_o^+ + w_4 * d_f^+ \tag{6}$$

Decision Variables:

$$X_{abt} = \begin{cases} 1, & \text{if truck a is assigned to dock } b \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

Deviation Variables:

d_w^+ = Positive deviation from ideal waiting time goal
d_u^+ = Positive deviation from ideal unloading time goal
d_o^+ = Positive deviation from ideal overtime goal
d_f^+ = Positive deviation from ideal fairness score goal

Subject to:

Assignment Constraint:

$$b∈Dt∈TXabt=1 \ ∀a∈A \tag{9}$$

Weight Constraint:

$$X_{a1t} = 0 \qquad \forall a \in A \ where \ W_a \geq 4000, \ \forall t \in T \tag{10}$$

Service Mutual Exclusion Constraint:

$$\sum_{a \in A} \sum_{t'=\max(t-Ut_a+1, t_{start})}^{t} X_{abt' \leq 1} \qquad \forall b \in D, \forall t \in T \tag{11}$$

Arrival Time Constraint:

$$X_{abt} = 0 \qquad \forall a \in A, \forall b \in D, \forall t < \tau_a \tag{12}$$

Waiting Time Goal Constraint:

$$\sum_{a,b,t} (St_a - \max(\tau_a, t_{start})) * X_{abt} - D_w^+ \leq W^{ideal} \tag{13}$$

Unloading Time Goal Constraint:

$$\sum_{a,b,t} Ut_{ab} * X_{abt} - D_u^+ \leq U^{ideal} \tag{14}$$

 DOI: 10.25077/josi.v24.n2.p324-344.2025

Overtime Goal Constraint

$$\sum_{a,b,t} \max(St_a + Ut_a - t_{end}, 0) * X_{abt} - D_o^+ \leq O^{ideal} \tag{15}$$

Fairness Goal Constraint

$$\sum_{a,b,t} \max(Wt_a - \overline{X_w}, 0) * X_{abt} - D_f^+ \leq F^{ideal} \tag{16}$$

where

A = Set of all trucks in a dataset
D = Set of all loading docks
T = Set of all time slots (0 to 1440)
τ_a = Arrival time of truck a
Ut_a = Unload time for truck a at its assigned loading dock
Wt_a = Wait time for truck a
St_a = Time when truck a starts unloading
t_{start} = Start of workday (08:00)
t_{end} = End of workday (18:00)
$\overline{X_w}$ = Average waiting time
W^{ideal} = Ideal value for waiting time
U^{ideal} = Ideal value for unloading time
O^{ideal} = Ideal value for overtime
F^{ideal} = Ideal value for fairness score
w_1 = Weight for waiting time goal
w_2 = Weight for unloading time goal
w_3 = Weight for overtime goal
w_4 = Weight for fairness score goal

The objective function (7) is the minimization of the sum of deviations from the waiting time, unloading time, overtime, and fairness goals. Positive deviation is measured by calculating the difference between the model's score for a given goal and the ideal value obtained by solving a Linear Programming model with a single goal as its objective value. The decision variables for this model consists of only binary variables, indicating whether truck a is processed at loading dock *b* at time *t* (8).

The model is bound by several constraints: A truck can only be assigned to 1 dock at any time (9). Trucks with over 4,000 kilograms in weight must be processed at loading dock 2 (10). Once a truck has been assigned to a dock, it needs to be fully unloaded at that dock (11). A truck can only be processed after it has arrived (12). The combination of (9) and (11) ensures that the resulting schedule can only place one truck in each dock-time slot pair, and a dock can immediately be assigned to a valid truck afterwards.

The formula for total waiting time is the sum of the difference between a truck's start time, and whichever is the greater value between the truck's arrival and the start of the workday (13). The formula for total unloading time is the sum of unloading time for all trucks (14). The formula for total overtime is the sum of minutes spent unloading past the end of the workday (15). Lastly, the formula for the fairness goal is the difference between a truck's waiting time and the average wait time (16).

Evaluation Metrics

To evaluate the performance of the GP model, the results will be compared with the results of a FIFO system handling the same dataset, wherein 3 key metrics will be measured: waiting time, unloading time, and overtime. Maximum and average waiting times will also be measured as control variables. The following are definitions and formulas for each metric.

Total Waiting Time

Waiting time is the amount of time a truck spends waiting for its turn to unload after arriving at the facility. Total waiting time is the sum of waiting times for all trucks.

$$\text{Total Waiting Time} = \Sigma_{a \in A} \, Wt_a \tag{17}$$

Total Unloading Time

Unloading time is the amount of time required to unload and transport a truck's load to its designated warehouse. Total unloading time is the sum of unload times for all trucks.

$$\text{Total Unload Time} = \Sigma_{a \in A} \, Ut_a \tag{18}$$

Total Overtime

Overtime is the amount of time required to unload and transport a truck's load to its designated warehouse past operational hours. Total overtime is the sum of overtime for all trucks.

$$\text{Total Overtime} = \Sigma_{a \in A} \, Ot_a \tag{19}$$

Average Waiting Time

Average waiting time is the average amount of time a truck spends waiting before unloading begins.

$$\text{Average Wait Time} = \frac{\Sigma_{a \in A} Wt_a}{|A|} \tag{20}$$

Maximum Waiting Time

Maximum amount of time a truck spends waiting before unloading begins.

$$\text{Maximum Wait Time} = max_{a \in A}(Wt_a) \tag{21}$$

Monte Carlo Simulation

Monte Carlo Simulation is a computational technique used to model and analyze complex systems by introducing randomness and variability into the input parameters, in this case arrival and unloading times. It is particularly useful for evaluating the performance of optimization models under real-world uncertainties [23, 24].

To evaluate the robustness of the GP model, a Monte Carlo Simulation with 100 iterations is conducted for each dataset. The degree to which variables are perturbed is classified into three categories; low (±3 minutes), medium (±5 minutes), and high (±8 minutes). An iteration's assigned class is randomized with a normal distribution function.

 DOI: 10.25077/josi.v24.n2.p324-344.2025

On each iteration, we calculate the evaluation metrics defined in (14, 15, 16, 17, 18) and derive the following statistical measures:

Mean

Mean is the average value of a metric across all iterations. Mean represents the average outcome for the simulation, providing a baseline estimate for the schedule's efficacy.

$$\overline{X_\iota} = \frac{1}{N}\sum_{i=1}^{N} i \tag{22}$$

i = {waiting time, unloading time, overtime, average wait time, maximum wait time}

Standard Deviation (Sigma)

Standard deviation or sigma is a measure of metric's variability across iterations. It can be interpreted as the spread of the results; a higher value would indicate a higher degree of variability and therefore risk of underperformance.

$$\text{Sigma } (\sigma)_i = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\sum i - \overline{X_\iota})^2} \tag{23}$$

Upper Control Limit (UCL) and Lower Control Limit (LCL)

UCL and LCL are limits that define a range in which a metric is expected to vary under the given conditions. They are calculated based on the *3-sigma rule*, which assumes that 99.7% of the data falls within ~3σ of the mean [25]. They are calculated using the following formula:

$$UCL_i = \overline{X_\iota} + 3 * \sigma \tag{24}$$

$$LCL_i = \overline{X_\iota} - 3 * \sigma \tag{25}$$

RESULT AND DISCUSSION

The GP model in this study is built in python with the PuLP library version 3.0.2, using the GNU Linear Programming Kit (GLPK) solver algorithm [26]. Simulation results are stored and aggregated using a MongoDB community database [27]. The solved model's decision variables can be interpreted as a schedule detailing which trucks should be unloaded at which dock and at what time. This schedule is tested against a traditional FIFO system, using the aforementioned time efficiency metrics as the basis for comparison.

Single-Iteration Simulation Results

To illustrate the effectiveness of the GP model, we first compare its performance with FIFO for a single iteration. The input data consists of a list of trucks as shown in Table 3—in this case, the list consists of 40 trucks with arrival times generated using peak hours at 10:00 and 16:00. Figure 6 shows a graphical representation for both schedules side by side. The X index represents time measured in minutes past 00:00 stopping at 24:00, and the Y axis represents discrete identifiers for each truck in the dataset. Each truck is represented as a bar, with gray areas representing wait times, and colored areas indicating unloading times at the designated dock.

The figure shows how the model rearranges the assignment to better utilize resources. The model notably favors placing trucks with low unloading times at the more efficient loading dock to minimize waiting times, and it also makes the choice of putting off large shipments to avoid cascading effects it would cause when numerous trucks with

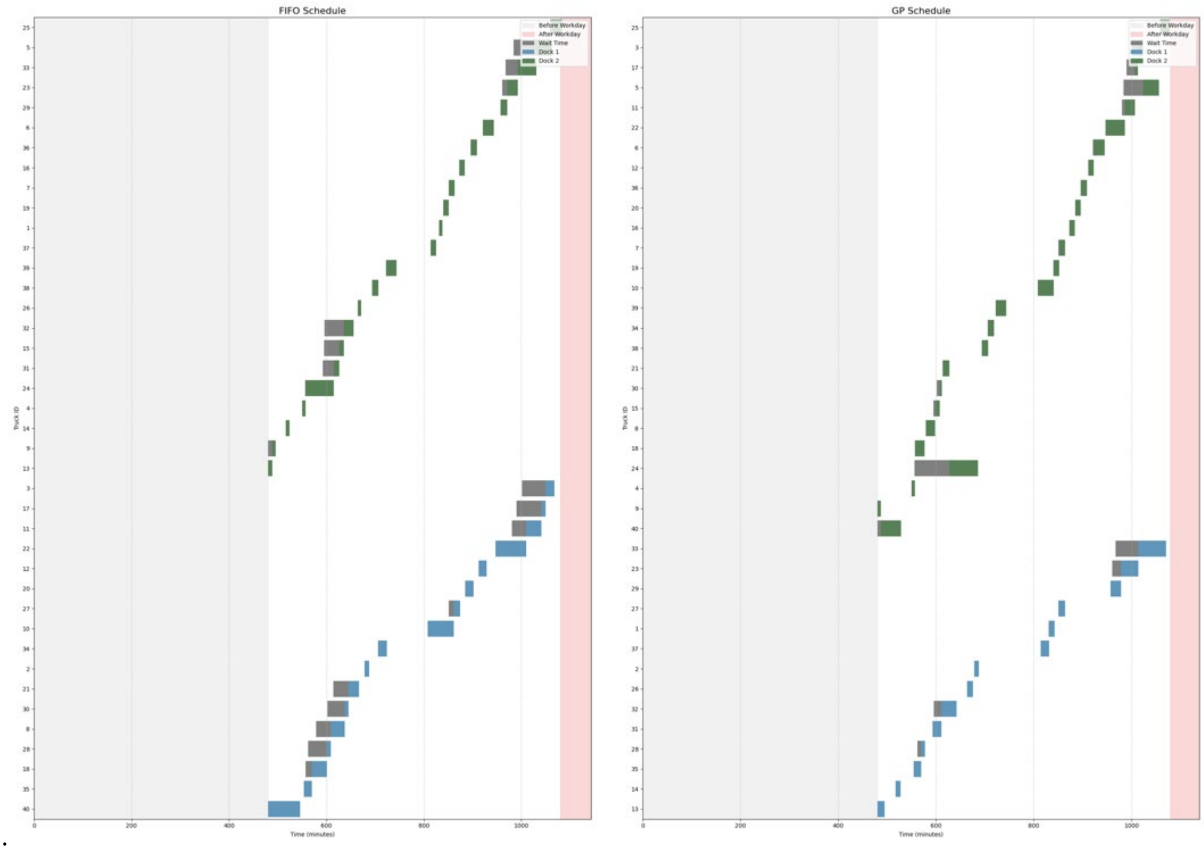

Figure 6. Graphical representation of the FIFO schedule (left) and GP schedule (right)

lower unloading times must wait for their turn at the loading dock, as seen in Truck #24 and Truck #5 in Figure 6. This could possibly cause unacceptable waiting times in higher-volume scenarios. However, placing a maximum waiting time constraint would cause the model to be infeasible in some scenarios. From an operations perspective, the best interpretation of such occurrence would be that the shipment should be rescheduled; however, the model still considers a high maximum waiting time to be preferable compared to a higher overall wait time for many trucks. To summarize; results are promising, with a substantial decrease in overall waiting time—and even though some trucks are made to wait, the overall average and maximum waiting times still see a marginal decrease. Details on the metrics gained from this simulation are shown in Table 5.

Table 5. Single-Iteration Simulation Results

Schedule	**Metric**	**Result**
First-In-First-Out	Waiting Time	477 minutes
	Unloading Time	794 minutes
	Overtime	72 minutes
	Average Waiting Time	11.93 minutes
	Maximum Waiting Time	62 minutes
Goal Programming	Waiting Time	304 minutes
	Unloading Time	764 minutes
	Overtime	60 minutes
	Average Waiting Time	7.6 minutes
	Maximum Waiting Time	51 minutes

 DOI: 10.25077/josi.v24.n2.p324-344.2025

Table 6. Monte-Carlo Simulation Results

Schedule	Metric	Mean	Sigma	UCL	LCL
First-In-First-Out	Waiting Time	492.88	66.83	505.98	479.78
	Unloading Time	797.68	23.95	802.37	792.99
	Overtime	79.58	31.72	86.30	73.36
Goal Programming	Waiting Time	382.15	82.43	398.31	365.99
	Unloading Time	764.95	23.45	769.55	760.35
	Overtime	70.83	16.23	74.01	67.65

Monte-Carlo Simulation Results

The results from the previous round of simulation shows that the GP model has good potential, successfully improving all metrics by a considerable margin without apparent drawbacks. However, the schedule is only optimal given that trucks will arrive exactly as scheduled and take the exact amount of time expected to finish unloading. Both require extraordinarily rare circumstances to occur in real-life conditions. Several decisions such as postponing large shipments incur a risk of cascading effects when there are changes to the input variable. To test the schedule against likely deviations from the schedule, we perturb the dataset by a certain amount and test the schedule against it, again noting its performance when compared to a FIFO system. Over enough iterations, it allows us to predict the real-world viability of the schedule. The results from this are shown below in Table 6.

Despite having an overall worse score, the GP model still proves to be an improvement from a FIFO system. Waiting time sees the most significant decrease at 22%, though notably the sigma value is much higher, indicating that more extreme changes to the schedule that might be caused by unforeseen circumstances could cripple the performance of the model. A high sigma value typically indicates that a metric has a high rate of variability, meaning that sway in performance between instances is higher in the GP schedule as supposed to FIFO. Unloading time and overtime also sees a marginal improvement, with a 4% and 11% decrease respectively.

The UCL and LCL values are calculated within a confidence interval of 95%, meaning that 95% of trucks will have scores ranging between the 2 values. Table 6 shows that the size of the range is roughly equal for all metrics, though with a lower overall value. This shows that both schedules respond similarly towards delays from the expected arrival and unloading times. The GP schedule has a higher sigma value, particularly for waiting time—this is likely due to the previously mentioned behavior where trucks with high unloading times are given less priority to reduce overall wait times.

Fairness can be further enforced by increasing the weight of the fairness goal. However, achieving a meaningful difference in this metric would likely require a significant tradeoff in one of the other goals. A company's best interest in this case would be to add an external rule to reschedule trucks with very high waiting times. This presents another use case for the model to be integrated earlier in the procurement process; as a decision support tool to determine whether or not a shipment should be scheduled for a certain date.

Large-Scale Simulation Results

To truly see the viability of a model, it must be tested against the full range of possible inputs. Input datasets are grouped into 12 broad types based on peak hours, volume of trucks, and density or spread of truck arrivals. Variations of each type are generated with a total of 120 simulations. MCS is then applied to each scenario to obtain

Table 7. Aggregate results for low-volume simulation scenarios

Schedule	Metric	Morning				Afternoon			
		Low Spread		High Spread		Low Spread		High Spread	
		Mean	Sigma	Mean	Sigma	Mean	Sigma	Mean	Sigma
FIFO	Wait Time	181.61	34.55	111.80	24.63	218.81	40.54	126.49	30.87
	Unload Time	486.19	15.94	445.06	16.63	442.70	16.16	465.13	16.03
	Overtime	7.80	2.48	12.85	2.83	67.51	14.74	20.71	7.18
GP	Wait Time	135.14	28.54	94.68	23.49	161.94	30.77	108.49	27.43
	Unload Time	455.55	14.80	412.41	15.20	413.09	14.95	435.68	15.20
	Overtime	7.41	2.49	9.12	2.36	49.74	10.12	17.97	5.63

Table 8. Percentage changes for low-volume simulation scenarios

Metric	Morning		Afternoon	
	Low Spread	High Spread	Low Spread	High Spread
Wait Time	19.53%	4.64%	19.11%	9.45%
Unload Time	6.29%	7.49%	6.79%	6.50%
Overtime	17.23%	21.56%	17.23%	3.78%

the mean and sigma values for each schedule. Results are grouped into 3 based on the number of inbound trucks parameter.

Low-Volume Simulation Scenario Results

Scenarios #1 to #40 feature datasets with low arrival volume (20 trucks). Aggregate results are shown below in Table 7 and Table 8. I can be seen that in low-volume scenarios, the GP schedule shows better time efficiency, consistently reducing waiting time by 4.6% - 19.5%, unloading time by 6.3% - 7.5%, and overtime by 3.8% - 21.5%. The GP schedules also show a slightly lower sigma value, indicating that the fairness problem discussed in the previous segment likely does not occur often in scenarios with low arrival volumes.

Medium-Volume Simulation Scenario Results

Scenarios #41 to #80 feature datasets with low arrival volume (30 trucks). Aggregate results are shown below in Table 9 and Table 10. As before, the GP model improves time efficiency of all metrics. There is a notable increase for waiting time, with values greater than 28% across the board. This shows that the model benefits from higher complexity—

Table 9. Aggregate results for medium-volume simulation scenarios

Schedule	Metric	Morning				Afternoon			
		Low Spread		High Spread		Low Spread		High Spread	
		Mean	Sigma	Mean	Sigma	Mean	Sigma	Mean	Sigma
FIFO	Wait Time	780.87	107.46	438.08	74.69	715.95	95.91	835.50	93.66
	Unload Time	691.02	21.05	623.87	19.77	678.22	20.81	708.50	20.52
	Overtime	62.37	18.82	20.94	9.53	172.85	38.64	248.48	43.28
GP	Wait Time	506.91	79.96	300.66	57.09	488.33	78.40	532.34	75.64
	Unload Time	671.82	18.75	603.70	18.53	659.03	18.33	678.10	19.02
	Overtime	55.53	12.56	22.91	8.66	128.77	24.49	161.83	26.54

 DOI: 10.25077/josi.v24.n2.p324-344.2025

Table 10. Percentage changes for medium-volume simulation scenarios

Metric	Morning		Afternoon	
	Low Spread	High Spread	Low Spread	High Spread
Wait Time	33.74%	28.74%	28.28%	31.52%
Unload Time	2.79%	3.30%	2.88%	4.29%
Overtime	3.44%	1.05%	16.27%	19.53%

or rather the inefficiency of simple systems like FIFO scales with system complexity. Improvement in overtime are more stable than the previous round of testing, with 16.2% and 19.5% for scenarios with peak arrival times in the afternoon. Unloading time shows no major improvement, indicating that the metric is more so related to the load carried by each truck, an immutable factor as far as the model is concerned.

High-Volume Simulation Scenario Results

Scenarios #81 to #120 feature datasets with low arrival volume (40 trucks). Aggregate results are shown below in Table 11 and Table 12. The final round of testing again shows major improvements for all metrics, again with higher values for waiting time and overtime—further supporting the previous hypothesis that the model's effectiveness scales with system complexity. It is likely that as the system grows more complex and more potential decisions are involved, the probability of an unfavorable scheduling choice grows ever higher for the FIFO system.

Summary of Results

The results from the 120 simulations demonstrate a significant improvement in time efficiency when using the GP optimized schedule compared to the traditional FIFO system. In each scenario, the schedule generated by the GP model consistently shows better time efficiency compared to FIFO. This effect scales with complexity, with higher

Table 11. Aggregate results for high-volume simulation scenarios

Schedule	Metric	Morning				Afternoon			
		Low Spread		High Spread		Low Spread		High Spread	
		Mean	Sigma	Mean	Sigma	Mean	Sigma	Mean	Sigma
FIFO	**Wait Time**	2790.7	213.1	1915.0	173.2	2290.6	190.4	2020.5	197.4
	Unload Time	874.7	22.6	921.0	23.6	923.6	23.0	912.3	23.2
	Overtime	420.1	68.6	433.3	70.7	937.4	108.8	788.3	100.7
GP	**Wait Time**	1540.5	120.6	1226.5	120.5	1367.0	134.2	1235.6	135.6
	Unload Time	852.2	21.3	898.5	21.6	898.0	22.3	887.5	21.9
	Overtime	233.4	27.9	297.9	35.6	574.5	57.7	471.1	50.1

Table 12. Percentage changes for high-volume simulation scenarios

Metric	Morning		Afternoon	
	Low Spread	High Spread	Low Spread	High Spread
Wait Time	43.08%	36.32%	38.68%	36.27%
Unload Time	2.55%	2.52%	2.80%	2.71%
Overtime	17.23%	8.71%	34.50%	29.93%

Table 13. Summary of aggregate simulation results

Metric	Time Difference	Percentage Change
Mean Wait Time	386.53 minutes	27.33%
Mean Unloading Time	25.96 minutes	4.27%
Mean Overtime	111.39 minutes	17.44 %

and higher improvements observed as the volume of truck arrivals increase. Sensitivity analysis indicates that the two scheduling methods have similar levels of robustness, both being prone to cascading delays, with GP being particularly vulnerable to having exceedingly high maximum waiting times in certain scenarios. Through testing, the model is proven to be largely beneficial, with considerable improvements to the overall time efficiency of the system. Table 13 shows the aggregate simulation results across all scenarios.

On average, the GP model reduces mean waiting time by 27.33%, mean unloading time by 4.27%, and overtime by 17.44%. These improvements are substantial, indicating that the GP model was able to effectively and consistently provide a robust, performant solution that balances multiple objectives at the same time.

CONCLUSION

The GP model developed in this study provides an effective approach for optimizing time efficiency in warehouse scheduling operations. By minimizing the weighted sum of total waiting time, total unloading time, total overtime, and fairness deviation goal scores, the model produces a solution that balances multiple conflicting objectives. Testing results demonstrate that the GP model consistently outperforms a traditional FIFO system, yielding lower values for total waiting time, unloading time, and overtime to various degrees throughout testing. The results of this study can be easily utilized by warehouse managers and logistics planners–the GP model can be implemented without making changes to the existing system, and the ability to dynamically adjust weights in the objective function allows managers to prioritize specific goals based on real-time operational needs. While the model is successful in improving time efficiency, it often forgoes fairness in high-volume datasets, leading to outliers that receive unreasonably longer wait times to suppress the overall average waiting time. This is likely a limitation of the system itself, wherein constraining the maximum waiting time further might lead to infeasible models or less-acceptable results in other metrics. From an operational perspective, such scenarios should highlight the need for enhancements to the facility's processing capacity rather than scheduling efficiency. Optimization for SMEs is a crucial in promoting economic growth and ensuring fairness in an ever more competitive market. As such, future research in this field would prove to be beneficial. A more complex or non-linear approach might be able to handle fairness more adequately than the GP model in this study. Future work can build on the foundation laid by this study through exploring the use of non-linear constraints, dynamic scheduling methods, or predictive methods in case studies where a larger dataset is available to train a model on.

ACKNOWLEDGMENTS

The authors of this study would like to sincerely express our gratitude to the individuals and organizations who have directly or indirectly contributed towards the creation of this research, including our colleagues and professors at Petra Christian University and the University of British Columbia.

CONFLICT OF INTEREST

The authors confirm that there are no conflicts of interest related to the publication of this paper. All authors have made significant contributions to the research and approved the final manuscript. This work was conducted without any financial, personal, or professional influences on its content or outcomes.

FUNDING

The authors received no financial support for the research, authorship, and/or publication of this article.

DATA AVAILABILITY STATEMENT

Simulation results are stored and aggregated using a MongoDB community database and available at: https://www.mongodb.com/products/community.

References

[1] A. Greasley, "Supply chain management," in Routledge eBooks, 2019, pp. 84–92. doi: 10.4324/9780429290602-15.

[2] Hamontree, S. Prompakdee, and J. Koiwanit, "Resource scheduling problem in distribution center," IOP Conference Series Materials Science and Engineering, vol. 639, no. 1, p. 012017, Oct. 2019, doi: 10.1088/1757-899x/639/1/012017.

[3] Z. Lin, "Big Data Analytics in Supply Chain Optimization and Risk Management: A case study of Amazon," *SHS Web of Conferences*, vol. 208, p. 04024, Jan. 2024, doi: 10.1051/shsconf/202420804024.

[4] J. Yao, "Supply chain scheduling optimisation in mass customisation based on dynamic profit preference and application case study," *Production Planning & Control*, vol. 22, no. 7, pp. 690–707, Jan. 2011, doi: 10.1080/09537287.2010.537577.

[5] H. Liu, F. Wang, J. Zhao, J. Yang, C. Tan, and L. Zhou, "Performance analysis of picking path strategies in Chevron Layout Warehouse," *Mathematics*, vol. 10, no. 3, p. 395, Jan. 2022, doi: 10.3390/math10030395.

[6] Journal, "Reviewing the role of Mathematical Optimization in Operations Research: Algorithms, applications, and challenges," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 08, no. 02, pp. 1–11, Feb. 2024, doi: 10.55041/ijsrem28578.

[7] Aktas and M. Kabak, "A goal programming model for grid-connected hybrid energy system operations," *SN Applied Sciences*, vol. 2, no. 1, Dec. 2019, doi: 10.1007/s42452-019-1878-1.

[8] M. F. Kahraman, H. Bilge, and S. Öztürk, "Uncertainty analysis of milling parameters using Monte Carlo simulation, the Taguchi optimization method and data-driven modeling," *Materials Testing*, vol. 61, no. 5, pp. 477–483, Apr. 2019, doi: 10.3139/120.111344.

[9] P. Centobelli, G. Converso, T. Murino, and L. C. Santillo, "Flow shop scheduling algorithm to optimize warehouse activities," *International Journal of Industrial Engineering Computations*, pp. 49–66, Aug. 2015, doi: 10.5267/j.ijiec.2015.8.001.

[10] W. Yu, C. Ha, and S. Park, "A Hybrid Genetic Algorithm for Integrated Truck Scheduling and Product Routing on the Cross-Docking System with Multiple Receiving and Shipping Docks," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–17, Mar. 2021, doi: 10.1155/2021/2026834.

[11] Y. Wang, W. Liu, C. Wang, F. Fadzil, S. Lauria, and X. Liu, "A Novel Multi-Objective Optimization Approach with Flexible Operation Planning Strategy for Truck Scheduling," *International Journal of Network Dynamics and Intelligence*, p. 100002, Jun. 2023, doi: 10.53941/ijndi.2023.100002.

[12] H. M. Alakaş, T. Eren, A. Yelek, and E. H. Özder, "Goal programming models for high-speed train crew scheduling problem," *Soft Computing*, vol. 28, no. 7–8, pp. 5921–5936, Dec. 2023, doi: 10.1007/s00500-023-09466-1.

[13] M. Sajid, H. Mittal, S. Pare, and M. Prasad, "Routing and scheduling optimization for UAV assisted delivery system: A hybrid approach," *Applied Soft Computing*, vol. 126, p. 109225, Jul. 2022, doi: 10.1016/j.asoc.2022.109225.

[14] W. Y. Ha and Z.-P. Jiang, "Optimization of Cube storage warehouse scheduling using genetic algorithms," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pp. 1–6, Aug. 2023, doi: 10.1109/case56687.2023.10260388.

[15] M. Kmiecik, "Creating a genetic algorithm for third-party logistics' warehouse delivery scheduling via a large language model," *Journal of Modelling in Management*, Jan. 2025, doi: 10.1108/jm2-06-2024-0192.

[16] F. Zeng, Y. Huang, J. Ding, X. Li, and Y. Liu, "Research on Scheduling Optimization Algorithm of Picking Job in Logistics Automated Warehouse Based on Chaotic Genetic Algorithm," *International Conference on Mechatronics, IoT and Industrial Informatics (ICMIII)*, pp. 320–324, Jun. 2023, doi: 10.1109/icmiii58949.2023.00067.

[17] W. O. Akerejola, E. U. Okpara, P. Ohikhena, and P. O. Emenike, "Availability of infrastructure and adoption of point of sales of selected small and medium enterprises (SMEs) in Lagos State, Nigeria," *International Journal of Academic Research in Business and Social Sciences*, vol. 9, no. 1, Jan. 2019, doi: 10.6007/ijarbss/v9-i1/5370.

[18] L. Hongbo, K. A. Lucien, Y. K. Raphael, and A. A. Boris, "Contribution of Small Medium Enterprises (SMEs) to Economic Development: Comparative Study of China and Cote d'lvoire," *International Journal of Academic Research in Business and Social Sciences*, vol. 8, no. 11, Dec. 2018, doi: 10.6007/ijarbss/v8-i11/5393.

[19] C. Wu, Y. Wu, Z. Li, and W. Zong, "A linear programming model for transportation planning in the arbitrary truck load strategy to supply products with unbalanced demand," *International Conference on Pure, Applied, and Computational Mathematics (PACM)*, p. 53, Jun. 2023, doi: 10.1117/12.2679309.

[20] H. B. Obaid, T. B. Trafalis, M. M. Abushaega, A. Altherwi, and A. Hamzi, "Optimizing dynamic evacuation using Mixed-Integer linear programming," *Mathematics*, vol. 13, no. 1, p. 12, Dec. 2024, doi: 10.3390/math13010012.

[21] I. Kacem, D. La Torre, and H. Masri, "Preface: Recent advances in multiple objective optimization and goal programming," *Annals of Operations Research*, vol. 296, no. 1–2, pp. 1–5, Dec. 2020, doi: 10.1007/s10479-020-03848-6.

[22] Ş. Gür, M. Pınarbaşı, H. M. Alakaş, and T. Eren, "Operating room scheduling with surgical team: a new approach with constraint programming and goal programming," Central European Journal of Operations Research, vol. 31, no. 4, pp. 1061–1085, Dec. 2022, doi: 10.1007/s10100-022-00835-z.

[23] I. Kouroudis, M. Gößwein, and A. Gagliardi, "Utilizing Data-Driven optimization to automate the parametrization of kinetic Monte Carlo models," *The Journal of Physical Chemistry A*, vol. 127, no. 28, pp. 5967–5978, Jul. 2023, doi: 10.1021/acs.jpca.3c02482.

[24] Y. Zeng, Y. Sun, T. Xu, and S. Su, "A reliability evaluation method for complex systems based on the editable GSPN and adaptive Monte Carlo simulation," *Systems Engineering*, vol. 27, no. 3, pp. 520–531, Nov. 2023, doi: 10.1002/sys.21736.

[25] K. P. Freeman, J. R. Cook, and E. H. Hooijberg, "Introduction to statistical quality control," *Journal of the American Veterinary Medical Association*, vol. 258, no. 7, pp. 733–739, Mar. 2021, doi: 10.2460/javma.258.7.733.

[26] PuLP, version 3.0.2, Python Package Index, 2023. [Online]. Available: https://pypi.org/project/PuLP. [Accessed: Date of Access].

[27] MongoDB, MongoDB Community Edition, MongoDB, Inc., [Online]. Available: https://www.mongodb.com/products/community.

AUTHORS BIOGRAPHY

Savio Enoson is an undergraduate student at Petra Christian University currently majoring in Business Information Systems. He places a strong emphasis on multidisciplinary learning, with projects related to industrial engineering, commerce, and operations management. This study was inspired by his experiences in industry, as well as his experience studying at the University of British Columbia, Canada during an exchange program. His research interest includes supply chain management, machine learning, and data science.

Andreas Handojo is an associate professor and head of field studies at the Department of Informatics in Petra Christian University, Indonesia. He received his Doctor of Industrial Engineering degree in Supply Chain and Logistics Management; and master's degree in management of technology from the Sepuluh Nopember Institute of Technology, Indonesia. He achieved his bachelor's degree in electrical engineering at Petra Christian University. His research interests are machine learning related to engineering and industrial application.

Tanti Octavia is an associate professor specializing in Supply Chain and Inventory Management. She achieved a master's degree in industrial engineering and management from the Asian Institute of Technology, Thailand and a bachelor's degree in the industrial engineering department from Petra Christian University, Indonesia. She actively works in research and publishing, providing her expertise to help create effective operating methods across diverse sectors. Her research interests are industrial engineering related to supply chain and inventory management.

Demetrio Franklyn Edryan is an undergraduate student at the University of British Columbia, Canada majoring in computer science. He has an extensive background in mathematics and software engineering, with notable achievements in the mathematics national Olympiad. He is passionate about exploring analytical approaches that make use of technological advancements to solve problems in industry. His research interests are primarily on the field of data science and cybersecurity.