Green Supply Chain Network Design with Emphasis on Inventory Decisions

Meysam Mahjoob\textsuperscript{a}, Seyed Sajjad Fazeli\textsuperscript{b,\textasteriskcentered}, Soodabeh Milanlouei\textsuperscript{c}, Ali Kamali Mohammadzadeh\textsuperscript{b} and Leyla Sadat Tavassoli\textsuperscript{d}

\textsuperscript{a}Department of Industrial Engineering, University of Tehran, Fooman, Rasht
\textsuperscript{b}Department of Industrial and System Engineering, Wayne State university, Detroit, MI
\textsuperscript{c}Center for Complex Network Research, Northeastern University, Boston, MA
\textsuperscript{d}Department of Industrial Manufacturing and Systems Engineering, University of Texas at Arlington, Arlington, TX

\begin{tabular}{l}
\textbf{ARTICLE INFO} \hline
\textbf{Keywords:} \\
Location-Routing-Inventory \\
Supply Chain Network \\
Evolutionary Algorithms \\
Bi-objective Optimization \\
Pareto Solution
\end{tabular}

\begin{tabular}{l}
\textbf{ABSTRACT} \hline
Excessive greenhouse gas emissions from the transportation sector have led companies to move towards a sustainable supply chain network design. In this study we present a new bi-objective non-linear formulation where multiple inventory components are integrated into the location and routing decisions throughout the supply chain network. To efficiently solve the proposed model, we implement an exact method and four evolutionary algorithms for small and large-scale instances. Extensive computational results and sensitivity analysis are performed to validate the efficiency of the proposed approaches, both quantitatively and qualitatively. Besides, we run a statistical analysis to investigate whether there is any statistically significant difference between solution methods.

1. Introduction

The intense competition in the global markets has forced companies to manage and design their supply chains in a more productive way. Designing a cost-efficient supply chain network asks for simultaneous decision making on strategic, tactical, and operation levels. Facilities’ locations are considered as strategic decisions that strongly affect the tactical and operational aspects of a supply chain. Inventory policies and transportation activities are among the critical decisions at the tactical and operational levels. The integrity of these decisions is an essential factor that significantly reduces supply chain costs and leads to higher customer satisfaction. In recent decades, there have been extensive studies on the Location-Routing-Inventory problem (LRIP). The majority of these studies considered the same approach towards the inventory components. For example, the order quantity is assumed to be a constant across the supply chain network while, in reality, it could vary among distribution centers based on the customer’s demand. The same situation could be observed for the Safety Stock (SS) level at different distribution centers. Besides, the inventory decisions could be significantly affected by uncertainty in distribution centers’ demand. One of the first attempts to model a Location-Routing-Inventory problem was made by [1], where the authors considered a two-level supply chain consisting of customers and multiple depots with limited capacities. They incorporated the continuous review policy in their model. Later, the same model was solved by an innovative approach based on Tabu search and simulated annealing algorithms developed by [2]. Authors in [3] presented an innovative approach to incorporate non-linear inventory and routing costs into the facility location models considering randomness in customers’ demand. The study by [4] introduced a novel mathematical model that simultaneously take into account the location, routing, and inventory decisions considering uncertainty in demand. The order quantity was the only inventory decision included in the model.

Other than designing a cost-efficient supply chain network, due to the growing concern of global warming, companies strive to include the environmental aspects into their operations. It is reported that more than 28% of the total greenhouse gas emissions come from the transportation sector in U.S., which is mainly released from road activities [5, 6]. Besides, distribution centers could be a source of carbon emission caused by power consumption and volume of inventory [7]. Green logistics offers sustainable production and distribution strategies to the companies. Considering richer objectives and more operational constraints involved with sustainable logistics issues impose new challenges that lead to more complex combinatorial optimization problems. There are only a few studies that incorporated the environmental aspects into the LRIP. The research in [8] developed a new multi-objective mathematical model that addresses routing, inventory, and location decisions. They validate their model by implementing a real case study and investigating various factors that affect CO\textsubscript{2} emissions in a supply chain network. LRIP is an extension of the facility location problem, and it is NP-Hard [4]. Numerous researchers proposed different methodologies to overcome the computational complexity of the LRIP. There are only a few studies that developed exact algorithms such as Benders Decomposition and Lagrangian relaxation to solve the LRIP [9, 10]. The majority of literature focused on heuristic ([4, 11, 1, 12]),
meta-heuristic ([13, 14, 15, 16]) and hybrid ([17, 8]) methods.

In this paper, considering uncertainties in Distribution Centers (DC) demand, lead time, and many other sources, we propose an innovative mathematical formulation that simultaneously considers location, inventory, and routing decisions under mixed uncertainty. The formulation specifically considers the important inventory decisions that arise in real-world applications which hasn’t been fully investigated on the literature. Also, we take the green approach by adding a new objective function considering the vehicles’ load and their emissions. To solve the proposed non-linear bi-objective mixed-integer model efficiently, we first linearize and then solve the small-scale instances with an exact method. For large-scale instances, we implement different meta-heuristic algorithms and statistically compare their performance to select the best solution approach.

The remainder of this paper is organized as follows: Section 2 provides problem definition and mathematical formulation of the problem along with a subsequent reformulation. Section 3 represents the solution methodologies, where we present exact and heuristic methods. Section 4 provides details about the test instances, the numerical results, and statistical analysis for comparing the proposed approaches. Finally, Section 5 provides concluding remarks.

2. Problem Definition and Model Formulation

2.1. Problem Definition

We consider a three-echelon supply chain network where the first echelon contains a supplier that receives the orders from the distribution centers. The second level includes distribution centers, where they receive orders from retailers in the third echelon. Once the distribution centers receive orders from the retailers, they send the order to the supplier based on the continuous inventory policy. The continuous policy for DC $k$ is denoted as $(q_k, r)$. In this policy, the inventory of DCs is continuously reviewed. Whenever the inventory amount becomes less than or equal to the reorder point ($r$), an order of size $q_k$ is placed to the supplier. In reality, the demand in a supply chain is affected by many sources of uncertainty. Authors in [18] suggested that the normality assumption is a reasonable approximation for sufficiently large demand values. We assume that the demand of retailer $i$ follows normal distribution $D_i \sim N(\mu_i, \sigma_i)$. We define $n_i$ as order frequency and $d_{ik}$ as the order size of DC $k$. So, the annual inventory received by distribution $k$ is $n_i q_k$. Considering multiple retailers, the annual average demand and variance for all retailers assigned to DC $k$ are equal to $\sum_i n_i \mu_i y_{i,k}$ respectively and $\sum_i \sigma_i^2 y_{i,k}$ where $y_{i,k}$ is a binary variable equal to one, if retailer $i$ is assigned to DC $k$. Due to uncertainty in demand, the value of $n_i q_k$ and $\sum_i \mu_i y_{i,k}$ could be dissimilar which will result in different scenarios for the SS and Cycle Inventory (CI). Figure 1 shows the inventory status under different scenarios. The $L_k$ denotes the lead time for distribution $k$.

The goal is to locate a set of distribution centers, determine the inventory decisions, and select the best possible routes for a fleet of vehicles to satisfy the retailers demand such that the total cost of the system is minimized. The main assumptions used to formulate the model are as follow:

- Each retailer has an uncertain demand which follows a normal distribution.
- Each distribution center follows continuous review policy $(q_k, r_k)$.
- Vehicles are different in terms of capacity (heterogeneous fleet).
- Vehicles return to the start point after delivering the products.
- The inbound transportation (from supplier to distribution center) is considered to be direct (no routing), and the outbound transportation (from distribution to retailers) is decided based on routing strategy.
- One type of product is distributed in the network.
- The amount of $CO_2$ emission depends on the traveled distance, the load weight of vehicles, and the DC emissions caused by volume of flow.
- There is only one supplier in the network denoted as $s_0$.

2.2. Notation

- **Sets**
  - $K$: Set of distribution centers, with $k \in K$
  - $I$: Set of retailers, with $i \in I$
  - $V^{in}$: Set of vehicles used in the inbound transportation, with $v \in V^{in}$
  - $V^{out}$: Set of vehicles used in the outbound transportation (transport from suppliers to DCs), with $v \in V^{out}$
  - $V$: Set of all vehicles, with $v \in V$ and $V = V^{in} \cup V^{out}$
  - $S$: Set of all nodes including retailers and distribution centers with $s \in S = K \cup I$

- **Model parameters**
  - $f_k$: Fixed establishment cost for DC $k \in K$
  - $g_k$: Fixed transportation cost from supplier to DC $k \in K$
  - $a_k$: Fixed ordering cost of DC $k$ to the supplier
  - $b_k$: Supply cost from supplier $s_0$ to DC $k$
  - $d_{s_0,k}$: Distance between supplier $s_0$ and DC $k \in K$
Figure 1: Inventory status under different scenarios

- $l_k$: Lead time of DC $k \in K$
- $\mu_i$: Mean of annual demand for retailer $i \in I$
- $\sigma_i^2$: Variance of annual demand for retailer $i \in I$
- $d_{i,j}^t$: Distance between node $i$ and node $j$, with $i, j \in S$
- $c_{i,j}$: Transportation cost between node $i$ and node $j$, with $i, j \in S$
- $h_k$: Inventory holding cost of DC $k \in K$
- $\rho_i^0$: Amount of fuel consumption by vehicle $v \in V$ with no load
- $\rho_i^*$: Amount of fuel consumption by vehicle $v \in V$ with maximum load capacity
- $q_v$: Quantity of orders by DC $k \in K$
- $q_k$: Quantity of orders by DC $k \in K$ during a year
- $n_k$: Number of orders by DC $k \in K$ during a year
- $y_{i,k}$: 1 if retailer $i \in I$ is selected to be sourced by DC $k \in K$; 0 otherwise.
- $r_{k,v}$: 1 if vehicle $v \in V$ is assigned to DC $k \in K$ for transportation; 0 otherwise.
- $w_{i,j,v}$: 1 if edge $(i,j)$ is visited by vehicle $v \in V$, with $i, j \in S$, 0 otherwise.
- $u_{i,k}$: Flow of products from node $i$ to node $j$ carried by vehicle $v$, with $i, j \in S$
- $m_{i,v}$: Auxiliary binary variable for removing sub-tours related to retailer $i \in I$ and vehicle $v \in V$
- $f_k$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$
- $f_i$: Auxiliary binary variables associated with DC $k \in K$

2.3. Mathematical Formulation
The Non-linear bi-objective mixed integer linear programming model is defined as follows:

Min

$$z_1 = \sum_{k \in K} \left( f_k x_k + \sum_{v \in V^{in}} f_v w_{0,k,v} + \sum_{i \in I} \sum_{v \in V^{out}} f_v w_{k,i,v} \right)$$

Fixed DC establishment and fleet cost
\[
\beta \cdot \sum_{k \in K} \left( \sum_{v \in V^\text{in}} (g_k + b_k q_k)n_k + \sum_{v \in V^\text{out}} \sum_{i,j \in S} c_{i,j} w_{i,j,v} r_{k,v} n_k \right) + \sum_{k \in K} \sum_{i \in I} w_{k,i,v} \leq 1 \quad \forall v \in V^\text{out} \tag{10}
\]
\[
\sum_{j \in S} w_{i,j,v} + \sum_{j \in S} w_{k,j,v} - y_{i,k} \leq 1 \quad \forall k \in K, i \in I, v \in V^\text{out} \tag{11}
\]
\[
\sum_{j \in S} \sum_{v \in V^\text{out}} (u_{i,j,v} - u_{j,i,v}) \leq M (1 - y_{i,k}) + \frac{\mu_i}{n_k} \quad \forall i \in I, k \in K \tag{12}
\]
\[
\sum_{j \in S} \sum_{v \in V^\text{out}} (u_{i,j,v} - u_{j,i,v}) \geq -M (1 - y_{i,k}) + \frac{\mu_i}{n_k} \quad \forall i \in I, k \in K \tag{13}
\]
\[
u_{s_0,k,v} \leq o_v w_{s_0,k,v} \quad \forall k \in K, v \in V^\text{in} \tag{14}
\]
\[
u_{i,j,v} \leq o_i w_{i,j,v} \quad \forall i,j \in S, v \in V^\text{out} \tag{15}
\]
\[
\sum_{i \in I} \mu_i \cdot y_{i,k} \leq \delta'_i \quad \forall k \in K \tag{16}
\]
\[
q_k = \sum_{v \in V^\text{in}} u_{s_0,k,v} \quad \forall k \in K \tag{17}
\]
\[
r_{k,v} = \sum_{i \in I} w_{i,j,v} \quad \forall k \in K, v \in V^\text{in} \tag{18}
\]
\[
r_{k,v} = \sum_{i \in I} w_{i,j,v} \quad \forall k \in K, v \in V^\text{out} \tag{19}
\]
\[
M t_k \geq \sum_{i \in I} \mu_i y_{i,k} - n_k q_k \quad \forall k \in K \tag{20}
\]
\[
-M (1 - i' k) \leq \sum_{i \in I} \mu_i y_{i,k} - n_k q_k \quad \forall k \in K \tag{21}
\]
\[
M t'_k \geq n_k q_k - \sum_{i \in I} \mu_i y_{i,k} - Z_a \sqrt{l_k \sum_{i \in I} \sigma_i^2 y_{i,k}} \quad \forall k \in K \tag{22}
\]
\[
-M (1 - i'_k) \leq n_k q_k - \sum_{i \in I} \mu_i y_{i,k} - Z_a \sqrt{l_k \sum_{i \in I} \sigma_i^2 y_{i,k}} \quad \forall k \in K \tag{23}
\]
\[
x_k \in \mathbb{Z}, y_{i,k}, w_{i,j,v}, t_k, t'_k, r_{k,v}, \in [0,1] \forall k \in K, i,j \in S, v \in V \tag{24}
\]
\[
n_k, q_k, u_{i,j,v}, m_{i,v}, y_{i,k} \geq 0 \quad \forall k \in K, i,j \in S, v \in V \tag{25}
\]

In equation (1), the first objective function minimizes the total costs of the network, including fixed annual DCs' establishment cost, fixed ordering cost, and fixed vehicle’s cost and expected inventory cost. The second objective function (2) minimizes the total carbon emission from inbound and outbound transportation activities, also, the carbon emissions from the distribution centers caused by flow of inventory. Constraints (3) ensure that the retailers are only assigned to the established DCs. Constraints (4) guarantee that each retailer is sourced by only one DC. Constraints (5) ensure that the demand of each customer is delivered by only one vehicle. Constraints (6) ensure that the vehicles dispatched from supplier to DC if only DC is established. Constraints (7) and (8) remove sub-tours and maintain flow in outbound transportation. Constraints (9) and (10) make sure that at most one established DC is included in each tour. Constraints (11) preserve the...
connection between the location and routing decisions, meaning that if vehicle \( v \) is dispatched from DC \( k \) and visits retailer \( i \), then DC \( k \) should be assigned to retailer \( i \). Constraints (12) and (13) balance the product flow and the demand from DCs to retailers. Constraints (14) and (15) impose the vehicles’ capacity limitations in inbound and outbound transportation. Constraints (16) ensure the DC’s capacity is not violated. Constraints (17) guarantee that the total shipment by vehicles in inbound transportation is equal to the order quantity. Constraints (18) and (19) state that the vehicle \( v \) can be dispatched if it is assigned to DC \( k \). Constraints (20) to (23) are the mathematical representation of Figure 1. Constraints (24) to (25) are variables restrictions.

3. Methodology

The model proposed in the previous section is a non-linear bi-objective mixed-integer formulation. To efficiently solve the problem, we first linearize the formulation and then offer an exact method to solve the small-scale instances. For large scale instances, we suggest four different evolutionary algorithms.

3.1. Exact Method

One of the approaches that is often used to obtain the Pareto solutions in multi-objective problems is augmented \( \varepsilon \)-constraint \((\varepsilon\text{-}c)\) method [19]. In the \( \varepsilon\text{-}c\) method, one of the objectives is selected as the main objective and the others ones will be added to the constraint sets. Suppose that there are \( m \) objective functions, then we have:

\[
\begin{align*}
\min & \quad f_1(x) \\
\text{s.t} & \quad f_2(x) \leq e_2, \ldots, f_m(x) \leq e_m \\
& \quad x \in S
\end{align*}
\]

where \( x \) is a vector of decision variables and \( S \) is the feasible region. We can obtain the set of solutions by changing the right-hand side of the constrained objective functions \( (e_i) \). However, the drawback of the \( \varepsilon \) constraint is that once the optimal solution for the main objective is obtained, the method has no obligation to find the best value for other objectives. In order to resolve this issue, [20] suggested using a lexicographic optimization technique to ensure multiple Pareto solutions. The interested readers may refer to [20] for more details. To efficiently implement the \((\varepsilon\text{-}c)\), we reformulate the proposed model to linear mixed integer problem. The details about the linearization methods are provided in A.

3.2. Solution Representation

The first step in implementing a meta-heuristic algorithm is the solution representation. The designed chromosome for this problem is a string of random numbers between \((0,1)\) with the length of \( |K| + |J| + |V^{in}| + |V^{out}| \). For example, a sample chromosome with two DCs, four retailers, and three vehicles on both outbound and inbound routes is as follows:

\[
\begin{array}{cccccc}
\text{DCs} & \text{Retailers} & \text{Vehicles in inbound} & \text{Vehicles in outbound} \\
2 & 1 & 3 & 4 & 1 & 2 \\
2 & 1 & 2 & 1 & 3 & 1 & 2 \\
\end{array}
\]

In the next step, sub-strings are ordered based on their values to determine each cell’s priority. For example, in Figure 3, the first sub-string indicates the priority of each DC for establishment. The second sub-string illustrates the priority of the assignment of retailers to the DCs. The third and fourth sub-string prioritize the allocation of vehicles. To obtain a feasible solution, we designed an efficient decoding strategy. According to the chromosome in Figure 3, we select DC 2 for establishment. We assign the retailers DC 2 based on their order until DC’s capacity is violated. Then, we choose the next DC and continue as before until all the retailers are assigned to the DCs. We make sure that the assignments preserve DCs capacity constraints. Let’s assume that the capacity of DC 1 is 1000 units and the capacity of retailers 1 to 4 are 300, 700, 400, and 600 units, respectively. Thus, retailers 3 and 4 (based on their order in the chromosome) are assigned to the DC 2 and retailers 1 and 2 to DC 1. It should be noted that if the retailers order were [3,1,2,4] instead of [3,4,1,2], retailers 3 and 1 would have been assigned to DC 2, and retailer 1 to DC 1. Due to capacity violation, we could not have assigned retailer 4 to any of the DCs. So, this solution is considered to be an infeasible solution. We penalize the infeasible solutions with cost with respect to the number of unassigned retailers. Once the retailers’ assignment to DCs is determined, the mean and variance of demand of each DC are calculated based on the aggregated demand of retailers. The next step is the assignment of inbound and outbound transportation to the DCs. The number of vehicles is dependent on the order quantity and number of orders by each DC. Following the above example, the annual demand for DC 2 is 400+600=1000 units. If we consider one-time order during a year, the entire order should be delivered to the DC and then to retailers in one shipment. Suppose that the capacity of vehicles in inbound transportation is 300 units. So, it is impossible to deliver the products to the DC 2 with three vehicles. We denote \( n^\text{max}_{DCk} \) as the maximum number of orders by DC \( k \). So, for the different number of orders \( (n) \), we can estimate the sufficient number of vehicles. For example, for \( n = 2 \) for DC 2, the order quantity is 500, and the minimum number of vehicles is two. The method is repeated for other DCs to obtain the minimum number of vehicles. Based on the above example, the estimation of number of vehicles needed for each DC is provided in A.
|       | n = 1 | n = 2 | n = 3 | n = 4 | n = 5 | n = 6 |
|-------|-------|-------|-------|-------|-------|-------|
| DC 1  | -     | 2     | 2     | 1     | 1     | 1     |
| DC 2  | -     | 2     | 2     | 1     | 1     | 1     |

Table 1
Estimation for the number of vehicles at each DC based on different number of orders

Table 1. We implement the same method for the outbound vehicles to create a similar table to Table 1. Once the tables for both inbound and outbound transportation are formed, for every DC, we randomly select an $n$ such that it is feasible in both layers of transportation. In the next step, we assign the vehicles to the distribution based on their order in the chromosome. Suppose that, $n_{k=2} = 2$ and $n_{k=1} = 4$, then vehicles 3 and 2 are assigned to DC 2 and vehicle 1 to DC 1. Since the amount of orders for DC 2 in each shipment is 500 units, the amount of load for vehicle 3 and 2 are 300 and 200 units, respectively. Also, vehicle 1 is assigned to DC 1 with 125 units of load in each shipment. Once all decisions are made, the amount of CO$_2$ is calculated based on the vehicles’ load.

3.3. Evolutionary Algorithms

Evolutionary algorithms are stochastic techniques inspired by the process of natural selection, which are extensively applied to solve different classes of NP-hard problems. In the literature, there are several evolutionary algorithms developed for multi-objective optimization problems. These algorithms use the Pareto dominance concept to lead the search process and return the best solutions in the form of a Pareto optimal set, while preserving the convergence and the diversity in the solution set. To solve the proposed model in Section 2 efficiently, we implement four meta-heuristic algorithms with different characteristics. These algorithms are Non-dominated Sorting Genetic Algorithms II (NSGA-II), Non-dominated Ranked Genetic algorithm (NRGA), Strength Pareto Evolutionary Algorithm II (SPEA-II), and Pareto Envelope-based Selection Algorithm II (PESA-II). We intend to evaluate the performance of well-known population-based algorithms for the proposed multi-objective location-inventory-routing problem.

3.3.1. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

Proposed by [21] as a biological heuristics algorithm, NSGA-II is among the most widely used multi-objective evolutionary algorithms. The algorithm utilizes a population of individuals and employs an elitism based sorting method. NSGA-II consists of two main operators, namely non-dominated sorting and crowding distance procedures. The non-dominated sorting method ranks all the solutions into different non-dominated levels according to the Pareto dominance principle. The crowding distance method, on the other hand, preserves the diversity of the solutions by calculating the dispersion of any two neighboring solutions in each front. These two procedures shape the Pareto front at each iteration [22, 23]. After generating an initial parent population ($P_0$), all non-dominated individuals are sorted. Using the tournament selection strategy, crossover, and mutation operators, the offspring population ($Q_0$) is generated afterward. Combining the parent and offspring populations, the combined population ($R_t = P_t \cup Q_t$) is formed at each generation $t$. The Pareto front is obtained by non-dominated sorting of combined population and estimation of the crowded distance of solutions. The subsequent parent population ($P_{t+1}$) is created by selecting the best individuals according to the rank and crowded distance. This procedure continues until the termination criterion is met.

3.3.2. Non-dominated Ranked Genetic Algorithm (NRGA)

Introduced by [24], NRGA is analogous to NSGA-II with one exception in the selection mechanism. Instead of the tournament selection operator utilized in NSGA-II, NRGA employs a ranked-based roulette wheel (RBRW) selection operator, combined with a Pareto-based population-ranking algorithm. According to RBRW, a parent $i$ is selected with the following probability:

$$P_i = \frac{2 \times \text{Rank}_i}{N \times (N + 1)}$$

Where $P_i$ and $N$ indicate the probability of being selected and the number of individuals in the population, respectively.

3.3.3. Strength Pareto Evolutionary Algorithm II (SPEA-II)

Presented in 1999 and 2001, SPEA and SPEA-II are considered as extensions of the genetic algorithm. SPEA-II employs three main operators: a fine-grained fitness assignment strategy, a nearest neighbor density estimation method, and an enhanced archive truncation method that guarantees border solutions are preserved [25]. This fitness assignment procedure has two main parts: the raw fitness based on the concept of dominance and a density estimation based on the k-nearest neighbor method. Besides, a truncation method is utilized to keep a fixed number of individuals in the archive. The interested reader may refer to [26] for more detailed descriptions.

3.3.4. Pareto Envelope-based Selection Algorithm (PESA-II)

PESA-II is an evolutionary optimization algorithm for solving multi-objective optimization problems developed by [27]. This algorithm is a modified version of its predecessor, PESA, in which region-based selection is used for assigning selective fitness. The algorithm maintains two populations: a fixed size internal population, and an external population, also called the archive set. Containing only the non-dominated solutions, the archive is being maintained for the selection at each iteration. The core of the algorithm can be summarized in six main steps as follows:
1. initialize the exterarchive and evaluate the generated internal population
2. incorporate the initial population (non-dominated members from the archive) into the archive.
3. If termination criteria are met, then stop. Otherwise, erase the internal population and reiterate step 4 until the generation of the new solution.
4. From the archive, select new parents and produce a new child using crossover and mutation. Set the probability of this process as $p$
5. Make the selection of one parent with the probability $1 - p$, perform the mutation of this parent to produce a child.
6. Return to step 2.

Now a candidate solution may enter the archive if and only if it has non-dominancy over the internal population. Any dominated candidate is removed from the archive.

4. Experiments and Results

In this section, we compare the computational performance of the exact method with the four meta-heuristic algorithms explained in section 3. Experiments were implemented in GAMS software equipped with the CPLEX 12.8 solver using a laptop with Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz, and 32GB RAM. It is worth mentioning that the values used in the model are generated from probability distribution provided in Table 2.

4.1. Parameter Tuning

The performance of the meta-heuristic algorithms is highly dependent on the input parameters. One of the most efficient methods for tuning the algorithm parameters is the Taguchi method [28]. The Taguchi can return a large amount of information with the least number of experiments. The Taguchi method makes use of orthogonal arrays, which approximate the effects of factors on the response mean and variation. Factors such as Noise factors result in variability in the performance of a system and cannot be controlled during production. The goal of the Taguchi method is to reduce the effect of uncontrollable factors and determine the best level (with higher signal-to-noise ($S/N$) ratio) for controllable factors. The $S/N$ is calculated as follows:

$$S/N = -10 \cdot \log \left( \frac{1}{n} \sum_{i=1}^{n} y_i^2 \right)$$

Since in multi-objective problems, a set of non-dominated solutions is considered optimal solutions, various metrics are often used to compare the solution algorithms. In this study, we use the following metrics:

- **Quality Metric (QM)**: Once non-dominated solutions from the four algorithms are obtained and stored in an archive, each pair of solutions are compared to each other and dominated solutions are eliminated. The algorithm’s share in the archive shows its quality.

- **Spacing Metric (SM)**: The SM metric provides valuable information regarding the distribution of non-dominated solutions in the solution space. The SM is calculated as follows:

$$SM = \frac{\sum_{i=1}^{n-1} |\bar{d} - d_i|}{(n - 1)\bar{d}}$$

where $n$ is the number of Pareto solutions, $d_i$ is the euclidean distance between successive Pareto solutions, and $\bar{d}$ is the average of $d_i$s. An algorithm with lower SM is more preferable.

- **Mean Ideal Distance (MID)**: The MID represents the distance between the best solution and Pareto solutions and calculated as follows:

$$MID = \frac{\sum_{i=1}^{n} \sqrt{\left(\frac{f_{i,max} - f_{i,min}^\text{best}}{f_{i,max} - f_{i,min}}\right)^2 + \left(\frac{f_{i,min} - f_{i,min}^\text{best}}{f_{i,max} - f_{i,min}}\right)^2}}{n}$$

where $f_{i,max}$ and $f_{i,min}$ are maximum and minimum values among the Pareto solutions for objective $i$ respectively. The $f_{i,min}^\text{best}$ denotes the best solution for objective $i$.

- **Diversification Matrix (DM)**: The DM shows the diversity in the Pareto solutions which is calculated as follows:

$$DM = \sqrt{\sum_{i=1}^{n} \max(|x_i - y_i|)}$$

where $x_i$ and $y_i$ denote the Pareto solutions for objective $i$.

To determine efficient values for algorithm parameters, we implemented a three-level Taguchi design in Minitab 17, including 9 experiments for NSGA-II and NRGA and 27 experiments for SPEA-II and PESA-II input parameters. Table 3 shows the tuned parameters. The details regarding the Taguchi experiments is provided in B.

| Parameter | Value |
|-----------|-------|
| $f_1$     | U(500-1000) |
| $f_2$     | U(10-15)   |
| $f_3$     | U(10-15)   |
| $f_4$     | U(5-10)    |
| $f_5$     | U(6-10)    |
| $f_6$     | U(5-10)    |
| $f_7$     | U(400-1500) |
| $f_8$     | U(10-100)  |

Table 2: Model’s Parameters
### Table 3

Tuned Parameters for each algorithm

| Parameters                | NSGA-II | NRGA | SPEA-II | PESA-II |
|---------------------------|---------|------|---------|---------|
| Population Size           | 100     | 150  | 100     | 100     |
| Archive Size              | -       | -    | 300     | 100     |
| Crossover Percentage      | 0.7     | 0.7  | 0.9     | 0.7     |
| Mutation Percentage       | 0.3     | 0.2  | 0.2     | 0.2     |
| Mutation Rate             | 0.03    | 0.05 | 0.03    | 0.05    |
| Selection Pressure        | -       | -    | -       | 3       |
| Deletion Pressure         | -       | -    | -       | 3       |

### Table 4

Generated test problems

| Test No. | $|k|$ | $|I|$ | $|V^{in}|$ | $|V^{out}|$ |
|----------|-----|-----|----------|----------|
| 1        | 2   | 4   | 3        | 3        |
| 2        | 2   | 4   | 4        | 3        |
| 3        | 2   | 4   | 3        | 4        |
| 4        | 3   | 5   | 3        | 3        |
| 5        | 3   | 5   | 4        | 4        |
| 6        | 3   | 7   | 3        | 3        |
| 7        | 4   | 10  | 5        | 5        |
| 8        | 5   | 15  | 7        | 7        |
| 9        | 6   | 20  | 9        | 9        |
| 10       | 7   | 25  | 11       | 11       |
| 11       | 8   | 30  | 13       | 13       |
| 12       | 10  | 50  | 15       | 15       |

### 4.2. Instance Generation

To demonstrate algorithms’ performance, we generated 12 test problems where the first five are considered small-scale and the rest are large-scale problems. Table 4 provides the details about the instances, where $|k|, |I|, |V^{in}|$, and $|V^{out}|$ indicate the number of DCs, retailers, vehicles in inbound, and vehicle in outbound transportation, respectively.

### 4.3. Results and Discussion

To evaluate the performance of the algorithms, we compare the results obtained by four algorithms with the exact methods for the small-scale test problems. Table 5 represents the comparison between the evolutionary algorithms considering comparison metrics with $\varepsilon$-constraint for five different runs. The numerical results indicate that evolutionary algorithms could produce optimal or close to optimal solutions when compared to the exact method in small-scale problems. In addition, we illustrated the Pareto solutions obtained from NSGA-II and $\varepsilon$-constraint algorithms in Figure 4. The figure shows that except for one solution, the NSGA-II was able to find all the solutions found by the exact method for the test problem 5. Also, the NSGA-II could find an additional solution where the $\varepsilon$-constraint was not able to obtain due to reaching the stipulated time-limit (3 hours). Table 6 presents the numerical results for the large-scale test problems and Figure 5 compares the average results obtained by different evolutionary algorithms considering multiple metrics for all test problems. It should be noted that the termination criterion for all algorithms is reaching to $3 \times 10^5$ for the number of function evaluations. In addition, we compare the solution methods in terms of running time provided in Figure 6. The results show that in all instances, the PESA-II algorithm significantly outperformed other algorithms. Other than PESA-II, the NSGA-II performed faster than other algorithms in all test problems. Also, The $\varepsilon$-constraint could only outperformed NRGA and SPEA in the first three test problems. Table 6 presents the numerical results for the large-scale test problems. Figure 5 compares the quality of the solutions obtained by different evolutionary algorithms considering multiple metrics for all test problems.

#### 4.3.1. Statistical Analysis

In this section, we investigate whether there is any statistically significant difference across the proposed
Table 5: Numerical results for small-scale problems

| Tests problems | No. | QM | SM | MID | DM | NSGA-II | NRGA | SPEA-II | PESA-II | A |
|---------------|-----|----|----|-----|----|---------|------|---------|---------|---|
|               | 1   | 1  | 1  | 1   | 1  | 24.83   | 24.83| 24.83   | 24.83   | 24.83 |
|               | 2   | 1  | 1  | 1   | 1  | 16.07   | 16.07| 16.07   | 16.07   | 16.07 |
|               | 3   | 0.88| 0.88| 0.88| 0.88| 43.07   | 43.07| 43.07   | 43.07   | 43.07 |
|               | 4   | 1  | 1  | 1   | 1  | 92.47   | 92.47| 92.47   | 92.47   | 92.47 |
|               | 5   | 0.93| 0.93| 0.93| 0.93| 153.6   | 161.9| 188.4   | 119.9   | 153.5 |

Table 6: Numerical results for large-scale problems

| Tests problems | No. | QM | SM | MID | DM | NSGA-II | NRGA | SPEA-II | PESA-II |
|---------------|-----|----|----|-----|----|---------|------|---------|---------|
|               | 1   | 0  | 0  | 0   | 0  | 35.71   | 95.57| 78.17   | 109.1   |
|               | 2   | 0.36| 0.14| 0.5  | 0   | 43.28   | 92.83| 104     | 59.89   |
|               | 3   | 0.25| 0.15| 0.6  | 0   | 69.47   | 146.1| 84.63   | 76.16   |
|               | 4   | 0.29| 0.21| 0.46| 0.04| 20.98   | 220.6| 441.1   | 45.09   |
|               | 5   | 0   | 0.2| 0.8  | 0  | 39.72   | 100.8| 28.73   | 174.3   |

Methods used for each defined metrics in the methodology section. We use the non-parametric Kruskal-Wallis test [29] since the sample size (here test problems) is small, the normality assumption is not met, and the number of methods is larger than two. We use 0.05 as a threshold for the p-value of the test to decide whether to reject the null hypothesis, indicating that there are at least two methods whose distribution of metrics are different from each other for large scale problems. We use SPSS software version 26 to first examine the QM distribution. Applying Kruskal-Wallis test, the resulting p-value is < 0.0005 and the null hypothesis is rejected. To find the exact methods whose QM metric is different from other algorithms, we run a pairwise comparison test and show the results in Table 7. Each row tests the null hypothesis that Sample 1 and Sample 2 distributions are the same. Asymptotic significances (2-sided tests) are displayed in the fifth column. Significance values have been adjusted by the Bonferroni correction for multiple testing, shown in the last column. Methods NSGA-II and PESA-II with p-value of 0.001 and methods SPEA-II and PESA-II with the same p-value are found to be statistically significantly different.
different from each other in terms of QM values (Figure 7-a).

For SM, the p-value of Kruskal-Wallis test is 0.016, concluding that the null hypothesis is rejected. The pairwise comparison tests in Table 8 shows that the distribution of SM for methods PESA-II and NRGA are statistically significantly different from each other (p-value of 0.012) (Figure 7-b). Since a lower value for SM is more preferable, we conclude the PESA-II method performs better than other methods regarding this specific metric.

While applying Kruskal-Wallis test on MID results in a p-value of 0.064 and indicates that we cannot reject the null hypothesis, the same p-value for DM is 0.031. Using pairwise comparisons tests (Table 9), we find that methods PESA-II and NRGA are statistically significantly different from each other in terms of DM values with p-value of 0.023. Since a higher value for DM is more preferable, methods NRGA and NSGA-II are performing better than other methods in terms of DM metric (Figure 7-c).

4.4. Sensitivity Analysis

We analyzed the effect of changes in the mean of annual demand. We considered six scenarios where we randomly draw six numbers from the scenario intervals. As it is shown in Figure 8, the proposed model is quite sensitive to the demand parameter. At each scenario, multiple solutions are provided by solving the model. These sets of solutions enable the decision makers to significantly decrease the amount of CO₂ emission with a small sacrifice in the supply chain cost. This can help companies to move towards the sustainable supply chain which will help them to mitigate the adverse effects of greenhouse gas emission and improve their competitiveness in the market. In addition, it can be observed that with an increase in the demand parameter we see an increase in the supply chain cost as well as the CO₂ emissions. This is mainly due to the increase in transportation activities and the need for establishment of more DCs because of higher demand volume.

5. Conclusion

In this research, we proposed a new mathematical model to design a green supply chain under uncertain data parameters. The proposed model incorporates different inventory component into the location-routing decisions. Besides, we considered the carbon emissions caused by transportation activities as well as distribution centers into the proposed framework to suggest a sustainable supply chain. To efficiently solve the proposed model, we first reformulated the model to a linear mixed-integer problem and then implemented an exact method which adapts to the multi-objective problems. For large-scale instances, we proposed four evolutionary algorithms with different characteristics. Extensive numerical experiments are performed on randomly generated test problems. The performance of four evolutionary algorithms was tested.
by comparing it to the exact method. For the small-scale instances, evolutionary algorithms were able to obtain optimal or near-optimal solutions. In terms of run-time, PESA-II performed quicker comparing to other algorithms. We also performed a statistical analysis to compare the evolutionary algorithms in terms of the quality of the solutions. Our statistical analysis showed that methods NSGA-II and SPEA-II are performing better in terms of the QM metric. For the SM metric, the PESA method demonstrates better efficiency compared to other algorithms. Moreover, methods NSGA-II and NRGA demonstrated better performance in regard to DM metric. To validate the model, we performed a case study, where we tested the model under different scenarios for the demand parameter. The results showed that on any demand level, the proposed model suggests a set of solutions where with a minor increase in supply chain cost, we remarkably decrease the CO\textsubscript{2} emissions. These solutions also provide various options for decision-makers to select their preferred outcome. Future work could include incorporating multiple scenarios into the proposed model. However, this will significantly increase the complexity of the proposed model, which requires designing efficient heuristic methods. Another extension could be considering

| Sample 1 - Sample 2 | Test Statistics | Std. Error | Std. Test Statistic | Sig. | Adj.Sig. |
|---------------------|----------------|------------|---------------------|------|----------|
| PESA-II - NRGA      | 13.571         | 4.397      | 3.087               | 0.002| 0.012    |
| PESA-II - NSGA-II   | 9.429          | 4.397      | 2.144               | 0.032| 0.192    |
| PESA-II - SPEA-II   | -9.857         | 4.397      | -2.242              | 0.025| 0.150    |
| NRGA - NSGA-II      | 4.143          | 4.397      | 0.942               | 0.346| 1.000    |
| NRGA - SPEA-II      | 3.714          | 4.397      | 0.845               | 0.398| 1.000    |
| NSGA-II - SPEA-II   | -0.429         | 4.397      | -0.097              | 0.922| 1.000    |

Table 9

| Sample 1 - Sample 2 | Test Statistics | Std. Error | Std. Test Statistic | Sig. | Adj.Sig. |
|---------------------|----------------|------------|---------------------|------|----------|
| PESA-II - NRGA      | 12.714         | 4.397      | 2.892               | 0.004| 0.023    |
| PESA-II - NSGA-II   | 9.143          | 4.397      | 2.079               | 0.038| 0.226    |
| PESA-II - SPEA-II   | -7.571         | 4.397      | -1.722              | 0.085| 0.510    |
| NRGA - NSGA-II      | 3.571          | 4.397      | 0.812               | 0.417| 1.000    |
| NRGA - SPEA-II      | 5.142          | 4.397      | 1.170               | 0.242| 1.000    |
| NSGA-II - SPEA-II   | 1.571          | 4.397      | 0.357               | 0.721| 1.000    |

Figure 7: Variability in each comparison metric across different algorithms
multiple products into the proposed framework.

Appendix

A. Linearization

The proposed model in Section 2 is non-linear due to the following situations:

A.1. Multiplication of integer (binary) and continuous variables

**Proposition A.1.** Consider the term $q_k \cdot n_k$ in objective (1). For bounded continuous and integer variables $a$ with $a_k \leq a_k, a_k > 0, \forall k \in K$ and $n$ with $n_k \leq b_k, b_k \geq 1, \forall k \in K$, a non-negative bi-linear variables $\zeta_k$ is defined as follows:

$$
\zeta_k = q_k \cdot n_k \quad \forall k \in K
$$

with the following constraints added to the constraint set:

$$
\begin{align*}
\zeta_k \geq 0 & \quad \forall k \in K \\
\zeta_k \leq a_k n_k & \quad \forall k \in K \\
\zeta_k \leq b_k n_k & \quad \forall k \in K \\
\zeta_k \geq a_k n_k + b_k n_k - a_k b_k & \quad \forall k \in K
\end{align*}
$$

For other variables with the same nature, we apply the same approach.

**Proof.** For the proof, see [30].

A.2. Multiplication of multiple binary variables

**Proposition A.2.** Suppose that there are $n$ binary variables $x_i$ and the goal is to linearize the product of the variables as follows:

$$
z = \prod_{i=1}^{n} x_i
$$

where $z$ is a binary variable and denotes the value of the product. Then we have:

$$
z \leq x_i \quad \forall i = 1, \ldots, n
$$

$$
z \geq \sum_{i=1}^{n} x_i - (n - 1)
$$

**Proof.** For the proof, see [31].

Consider the term $t_k \cdot t'_k$ in the objective (1) where two binary variables are multiplied together. We define binary variables $\theta_k$ as follows:

$$
\theta_k = t_k \cdot t'_k \quad \forall k \in K
$$

Then, the following constraints are added to constraints set:

$$
\begin{align*}
\theta_k & \leq t_k \quad \forall k \in K \\
\theta_k & \leq t'_k \quad \forall k \in K \\
\theta_k & \geq t_k + t'_k - 1 \quad \forall k \in K
\end{align*}
$$

Please note that we can always convert a bounded integer variable to the summation of some binary variables and use Proposition A.2 for multiplication of integer and binary variables together.

A.3. Square root of binary variables

**Proposition A.3.** Consider term $\sqrt{\sum_{i=1}^{n} a_i x_i}$ with $n$ binary variables $x_i$ and $a_i$ are parameters. We define parameter $b_j$ as follows:

$$
b_j = \sum_{i, S_{ij}=1} a_j \quad \forall j \in \{1, \ldots, 2^n\}
$$

where $S$ is a binary matrix with $n$ rows and $2^n$ (number of subsets) columns. We define binary variable $y_j$ where it is 1 if state $j$ is occurred with:

$$
y_j \leq \sum_{i=1}^{n} x_i + \sum_{i, S_{ij}=0}^{n} (1 - x_i) \leq y_j + n - 1
$$

then we can replace $\sqrt{\sum_{i=1}^{n} a_i x_i}$ with $\sum_{j=1}^{2^n} b_j y_j$

**Proof.** For the proof, see [32].

Consider term $\sqrt{\sum_{i \in I} \sigma_i^2 y_{i,k}}$ following the above logic, we have:

$$
b_{j,k} = \sum_{i \in I, S_{i,j,k}=1} \sigma_i^2 \quad \forall k \in K, j \in \{1, \ldots, 2^n\}
$$

$$
\sqrt{\sum_{i \in I} \sigma_i^2 y_{i,k}} = \sum_{j=1}^{2^n} b_{j,k} y_{j,k}
$$

Figure 8: The Pareto solutions obtained under different scenarios for the mean of annual demand.
Table 10
Parameters for NSGA-II and NRGA

| Parameters                  | NSGA-II | NRGA   |
|-----------------------------|---------|--------|
| Low            | Medium  | High   | Low   | Medium  | High   |
| Population Size   | 50      | 100    | 150   | 50      | 100    | 150    |
| Crossover Percentage | 0.70    | 0.80   | 0.90  | 0.70    | 0.80   | 0.90   |
| Mutation Percentage  | 0.30    | 0.20   | 0.10  | 0.30    | 0.20   | 0.10   |
| Mutation Rate      | 0.03    | 0.05   | 0.07  | 0.03    | 0.05   | 0.07   |

Table 11
Parameters for SPEA-II and PESA-II

| Parameters                  | SPEA-II | PESA-II |
|-----------------------------|---------|---------|
| Low            | Medium  | High   | Low   | Medium  | High   |
| Population Size   | 50      | 100    | 150   | 50      | 100    | 150    |
| Archive Size     | 100     | 200    | 300   | 100     | 200    | 300    |
| Crossover Percentage | 0.70    | 0.80   | 0.90  | 0.70    | 0.80   | 0.90   |
| Mutation Percentage  | 0.30    | 0.20   | 0.10  | 0.30    | 0.20   | 0.10   |
| Mutation Rate      | 0.03    | 0.05   | 0.07  | 0.03    | 0.05   | 0.07   |
| Selection Pressure | -       | -      | -     | 2       | 3      | 4      |
| Deletion Pressure  | -       | -      | -     | 1       | 2      | 3      |

We define binary variable $\tau_{j,k}$ which is activated by the following constraints:

$$\sum_{i:S_{i,j,k}=1} y_{i,k} + \sum_{i:S_{i,j,k}=0} (1 - y_{i,k}) \geq |I| \cdot \tau_{j,k} \quad \forall k \in K$$

(35)

$$\sum_{i:S_{i,j,k}=1} y_{i,k} + \sum_{i:S_{i,j,k}=0} (1 - y_{i,k}) \leq \tau_{j,k} + |I| - 1 \quad \forall k \in K$$

(36)

Following the three propositions, we linearized objective (1) and (2) as well as constraints (11)-(14),(22) and (23) to efficiently solve the problem.

B. Taguchi Experiments

The objective of parameter tuning is to calibrate the input parameters value of evolutionary algorithms. In this research, a three-level Taguchi design including 9 experiments for NSGA-II and NRGA and 27 experiments for SPEA-II and PESA-II is performed. The low, medium and high levels of the parameters along with their range are provided in Tables 10 and 11.

Each experiment is a different combination of factors specified where parameters change based on a predefined range. Each algorithm is tested in each experiment three times and the efficiency metrics are calculated. The average values for these runs are shown in Tables 12 and 13.

We aim to transform each metric into a response. It should be noted that the higher values for QM and DM and lower values for MID and SM are preferable. Figure 9 is the matrix of results where $X_j$ and $O_i$ denote $j$th metric and $i$th experiment, respectively, and $r_{ij}$ is the corresponding value. Since the metrics have different interpretation, we use a normalization method to convert all the values to comparable values. $X^+_j$ and $X^-_j$ represent normalized values for metrics QM/DM and MID/SM, respectively.

$$X_j \rightarrow R_{ij} = \frac{r_{ij} - \min_{i=1:n}(r_{ij})}{\max_{i=1:n}(r_{ij}) - \min_{i=1:n}(r_{ij})}, \quad X_j \rightarrow R_{ij} = \frac{\max_{i=1:n}(r_{ij}) - r_{ij}}{\max_{i=1:n}(r_{ij}) - \min_{i=1:n}(r_{ij})} \quad (37)$$

Using Goal Programming, we rank the metrics based on their importance and consider a weight for each of them and calculate the weighted sum (Response) as follows:

$$Response_i = \sum_{j=1}^{4} R_{ij}w_j; \quad (w_{QM}, w_{MID}, w_{SM}, w_{DM}) = (10^2, 10, 1, 1) \quad (38)$$

The weights are defined based on the importance of each metric. Normalized values and associated Responses for each algorithm are shown in Tables 14 and 15.

Now, based on Response values, we calculate the S/N rates and determine the level of each input parameter as it is shown in Figure 10. The final values for adjusted parameters is provided in Table 3.
Table 12
Results from adjusting the parameters for NSGA-II and NRGA

| NO. | NSGA-II | NRGA |
|-----|---------|------|
|     | QM SM MID DM | QM SM MID DM |
| 1   | 0.96 168.03 0.80 193.54 | 0.80 147.97 0.79 186.67 |
| 2   | 0.69 107.57 0.79 177.33 | 0.77 146.32 0.79 184.48 |
| 3   | 0.84 163.82 0.79 188.95 | 0.73 133.59 0.78 182.52 |
| 4   | 0.98 176.86 0.79 195.40 | 0.87 150.96 0.79 191.21 |
| 5   | 0.84 132.69 0.80 185.02 | 0.73 155.65 0.81 188.12 |
| 6   | 0.98 179.64 0.79 195.48 | 0.84 150.47 0.80 187.48 |
| 7   | 0.93 159.19 0.80 191.67 | 0.87 168.84 0.80 194.29 |
| 8   | 0.87 159.41 0.80 191.54 | 0.84 168.28 0.79 192.81 |
| 9   | 0.91 169.89 0.80 192.04 | 0.96 168.03 0.80 193.54 |

Table 13
Results from adjusting the parameters for SPEA-II and PESA-II

| NO. | SPEA-II | PESA-II |
|-----|---------|---------|
|     | QM SM MID DM | QM SM MID DM |
| 1   | 0.84 138.29 0.80 185.55 | 0.64 101.05 0.78 168.91 |
| 2   | 0.68 112.08 0.79 176.22 | 0.62 77.42 0.79 173.09 |
| 3   | 0.82 156.65 0.80 188.79 | 0.71 115.38 0.80 180.04 |
| 4   | 0.84 161.96 0.80 191.84 | 0.47 122.86 0.83 189.39 |
| 5   | 0.93 166.22 0.79 192.64 | 0.82 152.46 0.80 189.91 |
| 6   | 0.69 149.58 0.80 192.13 | 0.48 81.78 0.82 161.79 |
| 7   | 0.87 146.77 0.79 186.84 | 0.41 68.92 0.78 138.88 |
| 8   | 0.81 146.00 0.80 188.93 | 0.44 102.36 0.79 162.33 |
| 9   | 0.84 151.97 0.79 189.42 | 0.43 69.03 0.83 154.05 |
| 10  | 0.76 128.16 0.80 182.69 | 0.44 48.47 0.80 145.41 |
| 11  | 0.81 132.30 0.80 180.82 | 0.51 53.41 0.79 171.51 |
| 12  | 0.82 155.13 0.80 188.37 | 0.52 91.80 0.78 149.59 |
| 13  | 0.87 156.95 0.80 190.81 | 0.40 90.66 0.83 166.53 |
| 14  | 0.87 157.50 0.79 189.20 | 0.69 118.82 0.79 181.36 |
| 15  | 0.84 158.91 0.80 194.76 | 0.72 109.28 0.80 174.61 |
| 16  | 0.89 154.30 0.80 191.12 | 0.64 119.73 0.82 179.62 |
| 17  | 0.84 145.79 0.80 187.45 | 0.63 83.460 0.78 159.3 |
| 18  | 0.91 161.72 0.80 193.39 | 0.60 149.84 0.82 204.1 |
| 19  | 0.78 142.58 0.80 188.10 | 0.50 81.93 0.79 149.84 |
| 20  | 0.86 154.53 0.80 189.93 | 0.44 57.72 0.81 160.63 |
| 21  | 0.84 144.64 0.79 184.78 | 0.54 128.17 0.80 182.44 |
| 22  | 0.80 148.62 0.80 188.84 | 0.64 118.59 0.80 177.79 |
| 23  | 0.77 145.81 0.80 185.54 | 0.62 100.56 0.78 172.19 |
| 24  | 0.87 156.38 0.80 191.46 | 0.42 88.44 0.82 178.65 |
| 25  | 0.91 156.46 0.79 190.69 | 0.56 93.29 0.80 174.68 |
| 26  | 0.91 164.51 0.80 193.37 | 0.48 67.00 0.81 160.09 |
| 27  | 0.80 147.19 0.80 191.86 | 0.48 88.29 0.80 160.05 |

Table 14
Results from adjusting the parameters for NSGA-II and NRGA

| NO. | NSGA-II | NRGA |
|-----|---------|------|
|     | QM SM MID DM | QM SM MID DM |
| 1   | 0.92 0.16 0.37 0.89 | 0.3 0.59 0.74 0.35 |
| 2   | 0 1 1 0 | 0.17 0.64 0.62 0.17 |
| 3   | 0.54 0.22 0.83 0.64 | 0.54 0.17 0.41 0.42 |
| 4   | 0.62 0.28 0.05 0.79 | 0.6 0.31 1 64.1 |
| 5   | 0.77 0.14 0.37 0.81 | 0.5 0.73 0.87 58.18 |
Table 15
Results from adjusting the parameters for SPEA-II and PESA-II

| NO. | SPEA-II | | | | | PESA-II | | | | |
|-----|---------|---|---|---|---|---|---|---|---|---|---|
|     | QM | SM | MID | DM | Response | QM | SM | MID | DM | Response |
| 1   | 0.64 | 0.52 | 0.16 | 0.5 | 66.28 | 0.57 | 0.49 | 0.9 | 0.46 | 67.38 |
| 2   | 0.55 | 0.18 | 0.41 | 0.68 | 59.72 | 0.73 | 0.36 | 0.58 | 0.63 | 80.13 |
| 3   | 0.66 | 0.08 | 0.42 | 0.84 | 71.37 | 0.16 | 0.28 | 0 | 0.77 | 17.03 |
| 4   | 1 | 0 | 0.69 | 0.89 | 107.81 | 1 | 0 | 0.63 | 0.78 | 107.09 |
| 5   | 0.04 | 0.31 | 0.43 | 0.86 | 9.29 | 0.19 | 0.68 | 0.31 | 0.35 | 22.72 |
| 6   | 0.75 | 0.36 | 1 | 0.57 | 86.09 | 0.02 | 0.8 | 1 | 0 | 12.7 |
| 7   | 0.54 | 0.37 | 0.1 | 0.69 | 56.18 | 0.08 | 0.48 | 0.73 | 0.36 | 15.74 |
| 8   | 0.66 | 0.26 | 0.84 | 0.71 | 75.65 | 0.05 | 0.8 | 0.1 | 0.23 | 6.96 |
| 9   | 0.31 | 0.7 | 0.25 | 0.35 | 34.08 | 0.08 | 1 | 0.53 | 0.1 | 14.75 |
| 10  | 0.54 | 0.63 | 0.44 | 0.25 | 59.37 | 0.25 | 0.95 | 0.68 | 0.5 | 33.71 |
| 11  | 0.55 | 0.2 | 0.18 | 0.66 | 57.42 | 0.28 | 0.58 | 0.95 | 0.16 | 38.03 |
| 12  | 0.75 | 0.17 | 0 | 0.79 | 76.12 | 0 | 0.59 | 0.08 | 0.42 | 1.81 |
| 13  | 0.75 | 0.16 | 0.89 | 0.7 | 84.93 | 0.68 | 0.32 | 0.73 | 0.65 | 76.36 |
| 14  | 0.66 | 0.14 | 0.08 | 1 | 68.21 | 0.76 | 0.42 | 0.63 | 0.55 | 83.65 |
| 15  | 0.84 | 0.22 | 0.09 | 0.8 | 86.02 | 0.57 | 0.31 | 0.3 | 0.62 | 61.32 |
| 16  | 0.66 | 0.38 | 0.44 | 0.61 | 71.58 | 0.53 | 0.66 | 0.85 | 0.31 | 62.33 |
| 17  | 0.93 | 0.08 | 0.56 | 0.93 | 99.63 | 0.47 | 0.03 | 0.17 | 1 | 49.49 |
| 18  | 0.39 | 0.44 | 0.33 | 0.64 | 43.88 | 0.22 | 0.68 | 0.81 | 0.17 | 31.04 |
| 19  | 0.73 | 0.22 | 0.57 | 0.74 | 79.88 | 0.1 | 0.91 | 0.49 | 0.33 | 15.65 |
| 20  | 0.64 | 0.4 | 0.91 | 0.46 | 73.67 | 0.33 | 0.23 | 0.64 | 0.67 | 40.4 |
| 21  | 0.48 | 0.33 | 0.22 | 0.68 | 51.64 | 0.56 | 0.33 | 0.6 | 0.6 | 62.4 |
| 22  | 0.36 | 0.38 | 0.6 | 0.5 | 42.58 | 0.52 | 0.5 | 0.85 | 0.51 | 61.6 |
| 23  | 0.75 | 0.18 | 0.27 | 0.82 | 78.84 | 0.04 | 0.62 | 0.17 | 0.61 | 7.14 |
| 24  | 0.93 | 0.18 | 0.88 | 0.78 | 102.8 | 0.37 | 0.57 | 0.51 | 0.55 | 42.74 |
| 25  | 0.93 | 0.03 | 0.41 | 0.92 | 98.05 | 0.17 | 0.82 | 0.45 | 0.33 | 22.74 |
| 26  | 0.48 | 0.35 | 0.59 | 0.84 | 55.5 | 0.17 | 0.62 | 0.57 | 0.32 | 24.17 |
Figure 10: S/N rates for each algorithm
References

[1] Shu-Chu Liu and SB Lee. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. The International Journal of Advanced Manufacturing Technology, 22(11-12):941–950, 2003.

[2] SC Liu and CC Lin. A heuristic method for the combined location routing and inventory problem. The International Journal of Advanced Manufacturing Technology, 26(4):372–381, 2005.

[3] Zuo-Jun Max Shen and Lian Qi. Incorporating inventory and routing costs in strategic location models. European journal of operational research, 179(2):372–389, 2007.

[4] Amir Ahmadi Javid and Nader Azad. Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E: Logistics and Transportation Review, 46(5):582–597, 2010.

[5] Seyed Sajjad Fazeli, Saravanan Venkatachalam, and Jonathon M Smerek. Efficient algorithms for autonomous electric vehicles' min-max routing problem. arXiv preprint arXiv:2008.03333, 2020.

[6] Seyed Sajjad Fazeli, Saravanan Venkatachalam, Ratna Babu Chinnam, and Alper Murat. Two-stage stochastic choice modeling approach for electric vehicle charging station network design in urban communities. IEEE Transactions on Intelligent Transportation Systems, 2020.

[7] Ali Diabat and David Simchi-Levi. A carbon-capped supply chain network problem. In 2009 IEEE international conference on industrial engineering and engineering management, pages 523–527. IEEE, 2009.

[8] M Zhalechian, R Tavakkoli-Moghaddam, B Zahiri, and M Mohammadi. Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. Transportation Research Part E: Logistics and Transportation Review, 89:182–214, 2016.

[9] Yugang Yu, Chengbin Chu, Haoxun Chen, and Feng Chu. Large scale stochastic inventory routing problems with split delivery and service level constraints. Annals of Operations Research, 197(1):135–158, 2012.

[10] Xiaojin Zheng, Meixia Yin, and Yanxia Zhang. Integrated optimization of location, inventory and routing in supply chain network design. Transportation Research Part B: Methodological, 121:1–20, 2019.

[11] Nova Indah Saragih, Nur Bahagia, Ibu Syabri, et al. A heuristic method for location-inventory-routing problem in a three-echelon supply chain system. Computers & Industrial Engineering, 127:875–886, 2019.

[12] Seyed Reza Sajjadi and S Hossein Cheraghi. Multi-products location-routing problem integrated with inventory under stochastic demand. International Journal of Industrial and Systems Engineering, 7(4):454–476, 2011.

[13] Xiang Yang, Hanwu Ma, and Dengfan Zhang. Research into ilirp for logistics distribution network of deteriorating item based on jtid. In International Conference on Information Computing and Applications, pages 152–160. Springer, 2010.

[14] N Nekooghadirli, Reza Tavakkoli-Moghaddam, Vahid Reza Ghezavati, and SH Javanmard. Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. Computers & Industrial Engineering, 76:204–221, 2014.

[15] Fatemeh Forouzanfar and Reza Tavakkoli-Moghaddam. Using a genetic algorithm to optimize the total cost for a location-routing-inventory problem in a supply chain with risk pooling. Journal of Applied Operational Research, 4(1):2–13, 2012.

[16] Weitiao Wu, Wei Zhou, Yue Lin, Yuanqi Xie, and Wenzhou Jin. A hybrid metaheuristic algorithm for location inventory routing problem with time windows and fuel consumption. Expert Systems with Applications, page 114034, 2020.

[17] Atye Ghorbani and Mohammad Reza Akbari Jokar. A hybrid imperialist competitive-simulated annealing algorithm for a multi-source multi-product location-routing-inventory problem. Computers & Industrial Engineering, 101:116–127, 2016.

[18] Douglas C Montgomery, George C Runger, and Norma F Hubele. Engineering statistics. John Wiley & Sons, 2009.

[19] Jared L Cohon. Multiobjective programming and planning, volume 140. Courier Corporation, 2004.

[20] George Mavrotas and Kostas Florios. An improved version of the augmented ε-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. Applied Mathematics and Computation, 219(18):9652–9669, 2013.

[21] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation, 6(2):182–197, 2002.

[22] Madjid Tavana, Kaveh Khalili-Damghani, Debora Di Caprio, and Zeynab Oveisi. An evolutionary computation approach to solving repairable multi-state multi-objective redundancy allocation problems. Neural Computing and Applications, 30(1):127–139, 2018.

[23] Leyla, Sadat Tavassoli, Nahal Sakhavand, and Seyed Sajjad Fazeli. Integrated preventive maintenance scheduling model with redundancy for cutting tools on a single machine. Engineering. Technology & Applied Science Research, 10(6):6542–6548, 2020.

[24] Omar Al Jadaan, Lakishmi Rajamani, and CR Rao. Non-dominated ranked genetic algorithm for solving multiobjective optimization problems. In NRGA ’14, Journal of Theoretical and Applied Information Technology. Citeseer, 2008.

[25] Alejandro Lara-Caballero, Sergio Gerardo de-l-os Cobos-Silva, Roman Anselmo Mora-Gutiérrez, Eric Alfredo Rincón-García, Miguel Ángel Gutiérrez-Andrade, and Pedro Lara-Velázquez. Multiobjective genetic algorithms for reinforcing equal population in congressional districts. Mathematical Problems in Engineering, 2019, 2019.
[26] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. TIK-report, 103, 2001.

[27] David W Corne, Nick R Jerram, Joshua D Knowles, and Martin J Oates. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, pages 283–290, 2001.

[28] Genichi Taguchi. Introduction to quality engineering: designing quality into products and processes. Technical report, 1986.

[29] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. Journal of the American statistical Association, 47(260):583–621, 1952.

[30] Akshay Gupte, Shabbir Ahmed, Myun Seok Cheon, and Santanu Dey. Solving mixed integer bilinear problems using milp formulations. SIAM Journal on Optimization, 23(2):721–744, 2013.

[31] Fred Glover and Eugene Woolsey. Further reduction of zero-one polynomial programming problems to zero-one linear programming problems. Operations Research, 21(1):156–161, 1973.

[32] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.