

Green Location Routing: Competitive Bi-objective Bi-level Model and Solution

Berivan Azeez

University of Kurdistan

Arsalan Rahmani

`a.rahmani@uok.ac.ir`

University of Kurdistan

Meysam Hosseini

University of Kurdistan

Research Article

Keywords: Bi-level Programming, Multi-objective Programming, Green Location Routing Problem, Chemical Reaction Optimization, Non-dominated Sorting Genetic Algorithm, Tabu Search Algorithm

Posted Date: September 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-8050581/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Green Location Routing: Competitive Bi-objective Bi-level Model and Solution

Berivan F. Azeez^{1†}, Arsalan Rahmani^{2*}, Meysam Hosseini^{3†}

¹Department of Mathematics, University of Kurdistan , Sanandaj, Iran.

^{2*}Department of Mathematics, University of Kurdistan , Sanandaj, Iran.

³Department of Mathematics, University of Kurdistan , Sanandaj,
Campus of Bijar, Iran.

*Corresponding author(s). E-mail(s): a.rahmani@uok.ac.ir;

Contributing authors: berivan.faris91@gmail.com;

me.hosseini@uok.ac.ir;

[†]These authors contributed equally to this work.

Abstract

Recent years have shown how prioritizing and respecting environmental expertise can make an important contribution to moving a company's and country's sectors along the path to sustainable development. In the current study, we considered a class of competitive green facility location-routing problems in which two competitive logistically linked firms: manufacturer (producer) and customer (distributor) determine their choice one after another. First, the plant as the lead firm determines the number and location of service depot. Thereafter, the distribution company as being the follower is/orders the desired number of depots to distribute the products to the customers. To achieve maximum benefits with for leaders under follower ship in a multi-objective sense, a multi-objective bi-level mixed-integer linear programming model is developed. Type of the leaders and the Objective of the leader (and their objective) of the leader are the following: first aim is to minimize the fixed costs of depot operation, transportation costs as well as the costs of emissions from transportation and locating of the depots. One of the objectives of the follower is to maximize the market capture profit with respect to transportation costs while at the same time the same follower aims to minimize the emissions cost of vehicles operating with his logistics zone. Furthermore, an efficient meta-heuristic, specifically a Bi-level Multi-objective Chemical Reaction Optimization, was applied to the proposed method. The fundamental principle in bi-level multi objective chemical reaction optimization is to identify effective solutions within a realistic time frame

inside the bi-level framework. Experimental results on well-established benchmarks demonstrate that our methodology substantially surpasses the current bi-level variant of the Non-dominated Sorting Genetic Algorithm and Tabu search techniques formulated for this context.

Keywords: Bi-level Programming; Multi-objective Programming; Green Location Routing Problem; Chemical Reaction Optimization; Non-dominated Sorting Genetic Algorithm; Tabu Search Algorithm

1 Introduction

As global environmental concerns grow, industries are now focusing on sustainability and are aiming at minimizing the CO₂ emission and directly operating costs. Climate Change due to the accumulation of green house gases in the atmosphere, we are witnessing one of the most pressing challenges of our time. It has been tremendously influenced by human activities, in particular by the industrial and transport-related activities. Specifically, the transport sector is one of the leading producers of CO₂ that contributes to some of the most pressing environmental issues, including global warming, air pollution, and public health emergencies. As transportation is the core element of any contemporary supply chain and logistics system, its environmental effect is crucial. Carbon dioxide released to the environment not only hastens depletion of the ozone layer, and increased exposure to ultraviolet radiation, respiratory diseases and other public health risks. Thus, there is clearly an imperative for industries and companies to carry out green logistics that aims at both reducing CO₂ emissions and associated costs. In response to this a greater amount of focus has been put towards logistics and transport systems as candidates for sustainability enhancements. At the heart of these systems is the Vehicle Routing Problem (VRP), a classic optimization problem that is concerned with finding the optimal routes for a fleet of vehicles that have to serve a set of customers. The VRP was first defined by Dantzig and Ramser (1959) [1] and numerous applications of the VRP have been found in industry. The main goal of VRP is to control the operation cost, including the fuel consuming, travel time, and efficient customer services.

A generalization of the VRP, Location Routing Problem (LRP) aims at finding the optimized values for both depot location and vehicle routing. While the VRP is dedicated to routing only, the LRP is a more general problem in that it combines the decision making for facility location with the routing. This integrated approach increases logistics efficiency by jointly optimizing these two important supply chain components in terms of facility location and transportation network based location [2]. As a result, LRP allows firms to lower their transportation costs as well as facility expenses. In recent years, LRP has received a lot of attention for its potential use to solve sustainability issues, as corporations want to both decrease their environmental impact by optimally positioning the depots and vehicle routes.

As more and more attentions are paid to the issue of sustainability, the Green Location Routing Problem (GLRP) is a generalization of the LRP in which environmental

concerns are taken into account more explicitly. In the same line, the GLRP extends LRP with green logistics considerations, as for instance fuel consumption, greenhouse gas emissions, or the penetration of eco-friendly vehicles and fuels. Incorporating environmental goals into traditional logistics paradigm, the GLRP is a sustainable way for supply chain planning [3]. The GLRP represents the need of the industry to adapt to its priorities moving forward, with a need to balance the business of the business (keeping the lights on) whilst being accountable environmentally. Some researches attempted to pursue the optimal solution of the GLRP problem by adopting a variety of mathematical models and optimization methods.

For example, [4] tackled the Green Vehicle Routing Problem (GVRP), a particular case of the GLRP, which considers time windows in a scenario of electric vehicle (EV) logistics. Their work addressed the routing problem of on-line EVs, offering limited driving ranges and needing the recharging facilities. In their work, they optimized vehicle routes and services considering EV aspects, such as charging duration's and battery capacity [5]. And also, [6] presented a GVRP model rooted in a structure of density - clustering and aimed at reducing the fuel consumption as well as the emissions through the geographic - optimized routing. Their model included some significant green factors such as energy consumption, as well as emission reduction indicating the increasing trend on sustainability in the logistics planning [7].

Outside the realm of vehicles routing, [8] investigated the efficacy of environmentally friendly fuels in addressing the fuel consumption and emissions in the GLRP context. In [9], the authors introduced a model in which minimizing both cost and footprint is achieved using alternative fuel types, namely biofuels and hydrogen. However, [10] extended the LRP by incorporating social responsibility in the logistics planning since sustainability involves more than the environmental aspect. Such a model is economical and social although their theoretical model was within social and economic factors, which falls within the scope of the triple-bottom-line sustainability framework while the latter include economic, social and environmental considerations [11].

In spite of these contributions, current GLRP models do not take into account the full range of hierarchical level decision-making structures that characterize realistic logistics systems. In reality, the logistics planning problem is a multi-level decision process, where facility location and vehicle routing are usually made in separate times and based on different objectives. For example, the decision to locate depots strategically tends to be a long-term one, whereas routing decisions are generally more operational and dynamic in nature. Nevertheless, the majority of the encountered models are not able to adequately capture this hierarchical complexity, but treat facility location and routing in the sense of a single-level optimization problem.

Motivated by this gap in the literature, in this paper, we present a bi-level and bi-objective optimization model for the GLRP. At the upper level, the model aims to minimize CO₂ emissions and facility installation costs, and at the lower level, it prefers emissions reduction and maximize profit in distribution stage. In this paper, we develop a new bi-objective decision making framework for the GLRP that caters to both economic and environmental objectives during planning and scheduling. In order to solve such a NP-hard problem, in this paper a new meta heuristic Bi level Multi objective with Chemical Reaction Optimization(BMCRO) is proposed. We show evidence that

BMCRO outperforms Non-dominated Sorting Genetic Algorithm (*NSGA-II*) and Tabu Search (*TS*) in terms of quality and computational time. This study adds to the emerging green logistics literature by presenting for the first time a bi-level optimization model in this particular setting. Table (1) outlines the existing optimization models that demonstrates novelty of the proposed approach.

Table 1: Summary of Optimization Models in Related Works with Problem Level

Reference	Objective (Single/ Multi)	Problem Level	LP/ NLP (Single/ Bi)	Green Logis- tics (GL)	Stochastic/ Fuzzy	Solution Method
Dantzig & Ramser (1959) [1]	Single	Single-Level	LP	No	Deterministic	Exact(Simplex Method)
Golden & Wail (2007) [12]	Multi	Single-Level	LP	No	Deterministic	Heuristic (Clarke-Wright Savings)
Lee & Dong (2017) [3]	Multi	Single-Level	NLP	Yes (GL)	Stochastic	Metaheuristic (Genetic Algorithm)
Deb et al. (2002) [13]	Multi	Single-Level	LP	No	Deterministic	Metaheuristic (NSGA-II)
Camacho-Vallejo et al. (2022) [14]	Multi	Bi-Level	LP	Yes (GL)	Stochastic	Metaheuristic (Tabu Search)
Abbasi et al. (2022) [15]	Multi	Bi-Level	NLP	Yes (GL)	Stochastic	Metaheuristic (CRO)
Prins & Prodhon (2014) [16]	Multi	Single-Level	LP	Yes (GL)	Fuzzy	Heuristic (Greedy Algorithm)
Glover & Laguna (1997) [17]	Single	Single-Level	LP	No	Deterministic	Metaheuristic (Tabu Search)
Pisinger & Ropke (2007) [18]	Multi	Single-Level	LP	Yes (GL)	Deterministic	Metaheuristic (Adaptive Large Neighborhood Search)
Toth & Vigo (2002) [19]	Multi	Single-Level	LP	No	Deterministic	Exact (Branch-and-Cut)
Hoff et al. (2010) [20]	Multi	Single-Level	NLP	Yes (GL)	Fuzzy	Metaheuristic (Ant Colony Optimization)
Eskandarpour et al. (2015) [21]	Multi	Single-Level	NLP	Yes (GL)	Stochastic	Metaheuristic (Simulated Annealing)
Tavakkoli-Moghaddam (2015) [22]	Multi	Bi-Level	NLP	Yes (GL)	Stochastic	Metaheuristic (CRO)
Laporte et al. (2014) [23]	Multi	Single-Level	LP	Yes (GL)	Fuzzy	Particle Swarm Optimization
Braysy & Gendreau (2002) [24]	Single	Single-Level	LP	No	Deterministic	Tabu Search
Ganesh & Narendran (2008) [25]	Single	Single-Level	LP	No	Deterministic	Heuristic (Agglomerative + Simulated Annealing)
Vidal et al. (2015) [26]	Single	Single-Level	LP	No	Deterministic	Hybrid Metaheuristic (Iterated Local Search + Genetic Search with Dynamic Programming)
Braysy & Gendreau (2005) [27]	Multi	Single-Level	LP	No	Deterministic	Metaheuristic (Tabu Search)
Vidal et al. (2012) [28]	Multi	Single-Level	NLP	Yes (GL)	Stochastic	Metaheuristic (Hybrid Genetic Algorithm)
Schneider et al. (2014) [29]	Multi	Single-Level	NLP	Yes (GL)	Fuzzy	Metaheuristic (Ant Colony Optimization)
Andelmin & Bartolini (2017) [30]	Single	Single-Level	LP	Yes	Deterministic	Exact (Branch-and-Cut)
Romero-Barrera et al. (2024) [31]	Single	Single-Level	NLP	No	Stochastic	Metaheuristic (EnvAdapt-CRO-SL)
Aghaei et al. (2024) [32]	Multi	Bi-level	NLP	Yes	Robust, Fuzzy	Fuzzy-goal interactive method
Prakash & Akhtar (2025) [33]	Multi	Single-Level	NLP	Yes	Deterministic	HEMO (Hybrid Environmental Multi-objective Optimization)
Our Work	Multi	Bi-level	MILP	Yes	Deterministic	Metaheuristic CRO

2 Problem Description

In this section, the mathematical model related to the problem is given. In Table (2), the sets, indices, parameters, and decision variables are listed. Let J , L , I and FA represent the set of customers, types of vehicles, facilities, and factories, respectively. Also, let $V(L)$ be the set of available vehicles of each type l and let $k(l)$ denote the k -th type of vehicle l , where $k(l) \in V(L)$.

In this system, the depot receives the finished goods from the factory. The final transport from the depot to the direct customers is then. The demand from the customers can be met if it can be done economically. In this duopoly market, a supplier (the leader company) and the distributor (the follower company) make decisions. There is perfect information sharing between the leader and follower, as the leader's decisions rely on precise forecasts and information provided by the follower. This assumption simplifies the hierarchical interaction and enables the optimistic solution approach, where the follower chooses solutions that favor the leader.

The current bi-level model assumes that the leader has access to accurate information provided by the follower, which is a common optimistic assumption in bi-level programming to ensure tractability and solution feasibility. However, we acknowledge that in real-world settings, the follower may possess private information regarding operational costs, vehicle capabilities, or local market conditions that the leader does not fully observe.

The decision-making procedure can be split into two stages. In the first stage, the leading company determines the locations of the depots. In the downstream stage, the follower company observes the decision of the leader company and then makes its own decision to decide how to deliver the products to the customers. And the rules lead the two parties to shape their actions in ways that lower the possibility of pollution of the environment. The proposed bi-level model explicitly captures the hierarchical nature of decision-making by distinguishing between the strategic and tactical levels. The upper-level decisions correspond to long-term strategic choices such as depot location selection, which inherently involve multi-year commitments and significant capital investments. These decisions are modeled as binary variables indicating whether a depot is opened or not, reflecting their relatively static nature over the planning horizon. Conversely, the lower-level decisions represent tactical routing choices that are more dynamic and can adapt frequently to operational conditions, such as daily or weekly customer demands and vehicle routing schedules. This temporal distinction is embedded in the model structure by separating the decision variables and constraints for each level, where the leader's decisions set the feasible framework within which the follower optimizes routing and distribution. In the supply chain, as with everything in impact, we measure the environmental effects of all things in terms of damaging GHG emissions, such as CO₂, CFCs, and NO_x. The opened depots and the product transportation in the network lead to pollution. Vehicles are of different types, pollute to different degrees, and have different capacities, and they are not homogeneous. The emissions produced by each truck are proportional to the distance driven and the volume of goods carried. So we think of the cost, the amount of goods moved the distance. The leader firm's problem of decision making is to take actions such that two conditions are satisfied. The former involves minimizing depot and transportation costs, and the latter involves minimizing pollution. It is worth remembering that the follower company is aimed at maximizing its profits from customer service. Further, the following company would like to reduce transportation-related emissions and transportation-related costs. It is assumed that, in a follower role, the follower company states as a result of the leader company's decision. The latter could involve an exhaustive projection of each of the leader's decisions, seeking the optimal counteractions.

To present the proposed model, the symbols representing the sets, parameters, and variables used are first introduced in Table 2.

Table 2: List of Sets,Parameters and Variables

Notation	Description
Sets	
FA	The set of factories with the indexes
I	The set of potential facilities(depots) i
L	The set types of vehicles with indexes l
$V(L)$	The collection related to type L vehicles with the indexes $k(l)$
J	The set of Customers with the indexes s, j
Parameters	
f_i	The fixed operating cost for opening the depot i
c_{ri}	The cost of conveying each unit of product from the factory r to depot i
e_{ri}	The environmental impact related to transporting a unit of products from the factory r to depot i
e_i	The environmental impact related to opening a depot at the node i
$CapF_r(CapF'_i)$	The capacity of a factory r (depot i) for handling products
$a(b)$	The parameter related to the influence coefficient of the transportation(opening depot) environmental impact rate in the objective function
B	Maximum funds available to carry out the transport activity
E_{max}	The maximum allowed emission caused by transporting and opening depots
d_j	The amount of demand related to the j -th customer
h_j	The Profit associated with each customer j
c_{rl}	Rental cost of vehicle type l
cc_l	The per-unit cost of conveying products via vehicle type l
d_{ij}	The distance of node i to node j
$\alpha_l(\beta_l)$	Environmental impact related to conveying products by distributor firm via vehicle type l
M	Sufficient large positive constant
t_{sj}	The travel time of a vehicle from node s to node j
V_{max}	Maximum duration of the route
U_{max}	Maximum available capacity associated with the type of vehicle type l
$[l_s u_s]$	The customer unloading service time s
τ_s	The customer unloading service time s
Variables	
x_i	1 if the location i is open by the leader as a depot; otherwise, 0
y_{ri}	Products quantities shipped from the plant r to the depot i by the leader
$w_{k(l)}$	Binary inclusion of the arc $s - j$ in the route of the vehicle k of type l
$z_{k(l)}$	Binary selection for vehicle k of type l by depot i
$\delta_j^{k(l)}$	Binary variable regarding passing the vehicle k of type l through node j
t_j	The non negative variable service starts the time at the node j .

according to the explanation given, the problem model is expressed as below (see P_1).
Upper-level

$$P_1 : Z_1 = \min \sum_{i \in I} f_i x_i + \sum_{r \in FA} \sum_{i \in I} c_{ri} y_{ri} \quad (1)$$

$$\min \quad a \sum_{r \in FA} \sum_{i \in I} e_{ri} y_{ri} + b \sum_{i \in I} e'_i x_i \quad (2)$$

subject to:

$$\sum_{i \in I} y_{ri} \leq Capacity_{F_r}, \quad \forall r \quad (3)$$

$$\sum_{r \in FA} y_{ri} \leq Capacity_{F'_i}, \quad \forall i \quad (4)$$

$$y_{ri} \leq x_i, \quad \forall i, r \quad (5)$$

$$\sum_{i \in I} f_i x_i + \sum_{r \in FA} \sum_{i \in I} c_{ri} y_{ri} + a \sum_{r \in FA} \sum_{i \in I} e_{ri} y_{ri} + b \sum_{i \in I} e_i x_i \leq B \quad \forall s \quad (6)$$

$$\sum_{r \in FA} \sum_{i \in I} e_{ri} y_{ri} \leq E_{max} \quad (7)$$

$$y_{ri} \geq 0, \quad x_i \in \{0, 1\} \quad \forall i, r \quad (8)$$

Lower-Level

$$\max \left\{ \sum_{k(l) \in V(l)} \sum_{j \in J} d_j h_j w_{sj}^{k(l)} - \sum_{l \in L} \sum_{k(l) \in V(l)} \left[\sum_{j \in J} c r_l z_j^{k(l)} + \sum_{s \in J} \sum_{j \in J} c c_l d_{ij} w_{sj}^{k(l)} + \sum_{i \in I} c c_l d_{0i} w_{0i}^{k(l)} \right] \right\} \quad (9)$$

$$\min \left\{ \sum_{l \in L} \sum_{k(l) \in V(l)} \left(\sum_{s \in J} \sum_{j \in J} \alpha_l d_{ij} w_{sj}^{k(l)} + \sum_{i \in I} \beta_l d_{0i} w_{0i}^{k(l)} \right) \right\} \quad (10)$$

Subject to:

$$\sum_{i \in I} w_{0i}^{k(l)} = \sum_{j \in J} w_{ij}^{k(l)} \quad \forall \quad K(l) \in V(l), l \in L \quad (11)$$

$$\sum_{j \in J} w_{j0}^{k(l)} = \sum_{i \in I} w_{0i}^{k(l)} \quad \forall \quad K(l) \in V(l), l \in L \quad (12)$$

$$\sum_{l \in L} \sum_{k(l) \in V(l)} \sum_{j \in J} z_j^{k(l)} = 1 \quad (13)$$

$$\sum_{j \in J} z_j^{k(l)} \leq M \delta_j^{k(l)} \quad \forall \quad K(l) \in V(l), j \in J \quad (14)$$

$$\sum_{i \in I} w_{ij}^{k(l)} + \sum_{s \in J} w_{sj}^{k(l)} \leq 1 + M(1 - \delta_j^{k(l)}) \quad \forall \quad K(l) \in V(l), j \in J \quad (15)$$

$$-\sum_{i \in I} w_{ij}^{k(l)} - \sum_{s \in J} w_{sj}^{k(l)} \leq 1 + M(1 - \delta_j^{k(l)}) \quad \forall \quad K(l) \in V(l), j \in J \quad (16)$$

$$\sum_{j \in J} w_{sj}^{k(l)} = \sum_{j \in J} w_{js}^{k(l)} \quad \forall \quad K(l) \in V(l), l \in L \quad (17)$$

$$\sum_{s \in J^*} \sum_{\substack{j \in J \\ j \neq s}} t_{sj} w_{sj}^{k(l)} \leq V^{max} \quad \forall \quad K(l) \in V(l), l \in L \quad (18)$$

$$\sum_{s \in J^*} \sum_{\substack{j \in J \\ j \neq s}} d_j w_{sj}^{k(l)} \leq U_l^{max} \quad \forall \quad K(l) \in V(l), l \in L \quad (19)$$

$$\sum_{j \in J} \sum_{k(l) \in V(l)} z_j^{k(l)} \leq |V(l)| \quad \forall \quad l \in L \quad (20)$$

$$\sum_{s \in W} \sum_{j \in W} w_{sj}^{k(l)} \leq \sum_{j \in W} z_j^{k(l)} \gamma^l \quad \forall \quad K(l) \in V(l), l \in L \quad (21)$$

$$W_{0i}'^{k(l)} \leq x_i \quad \forall \quad K(l) \in V(l), l \in L, i \in I \quad (22)$$

$$\sum_{r \in FA} y_{ri} = \sum_{l \in L} \sum_{k(l) \in V(l)} \sum_{j \in J} d_j w_{ij}^{k(l)} \quad \forall \quad i \in I \quad (23)$$

$$t_s + S_s + t_{sj} - (u_s + \tau_s)(1 - w_{sj}^{k(l)}) \leq t_j \quad \forall \quad s, j \in J \quad (24)$$

$$l_s(1 - x_s) \leq t_s \leq u_s(1 - x_s) + T_{max} x_s \quad \forall \quad s \in J \quad (25)$$

$$\sum_{s \in J - \{j\}} g_{js} \leq \sum_{s \in J - \{j\}} g_{sj} - d_j(1 - x_j) + \sum_{s \in N} d_s x_j \quad \forall \quad j \in J \quad (26)$$

$$g_{sj} \leq U_l^{max} \sum_{j \in J} z_j^{k(l)} \quad \forall \quad s, j \in J \quad (27)$$

$$w_{ij}^{k(l)}, w_{0i}'^{k(l)}, w_{sj}^{k(l)}, \delta_j^{k(l)}, z_j^{k(l)} \in \{0, 1\} \quad \forall \quad K(l) \in V(l), l \in L, s, j \in J \quad (28)$$

$$t_j, g_{sj} \geq 0 \quad \forall \quad s, j \in J \quad (29)$$

The model described by equations (2.1)-(2.29) is a bi-objective bi-level linear programming problem. Equation (2.1) represents the leader's objective function, which includes fixed usage costs of the depots and transportation costs. The second leader's objective function measures the *CO2* emissions caused by the route of each type of vehicle and the *CO2* generated by constructing and shipping processes from the facilities. Inequalities (2.3) and (2.4) represent the maximum capacity of the factories and depots used, respectively. Constraint (2.5) specifies that all transports from the factories must be made to a node where the depot is open. Constraint (2.6) sets a maximum available budget for the Leader Company, and inequality (2.7) indicates the maximum level of pollution permitted by the company. Equation (2.8) indicates that the leader's variables are non-negative and binary, respectively. In Equation (2.9), the first objective function of the follower pertains to maximizing profit. The first term represents the total income for each commodity demanded by customers, while the remaining terms include the total rental cost of the vehicles, the overall transportation costs of all vehicles, and the transportation costs of the commodities from the depots. The

second objective function (2.10) for the followers measures the CO₂ emissions caused by the routes of each type of vehicle and the CO₂ generated by shipping processes. Constraints (2.11) and (2.12) ensure that each vehicle departs from and arrives at the depot exactly once. Constraint (2.13) stipulates that selected customers are visited only once. Constraints (2.14) to (2.16) define the relationships between the depot, vehicle type, and customers. Equation (2.17) enforces flow conservation. Constraint (2.18) sets the maximum route duration for each vehicle, while inequality (2.19) specifies the maximum available capacity for each vehicle type. Equation (2.20) limits the maximum availability of vehicles for each type. Constraint (2.21) ensures that only in-use vehicles have designated routes. Constraint (2.22) requires that a vehicle starts moving from an active depot, while also defining the relationship between the first and second levels. Constraint (2.23) links the two levels by determining the volume of serviced customer's demand. Constraint (2.24) sets the arrival time of the vehicle at a customer node, while inequality (2.25) defines the maximum and minimum permissible times for meeting customer demands. Constraint (2.26) specifies the vehicle load at each vertex, with expression (2.27) setting the maximum allowable vehicle load. Note that if the customer is at the same vertex as the depot, the customer's demand is not considered in the vehicle load. Constraints (2.28) and (2.29) define the sign of the variables.

In the proposed bi-objective bi-level problem (*BBO*), each player optimizes its own objective under consideration of the other one's decisions. More specifically, the leader and follower each try to minimize or maximize their cost while taking into account the CO₂ emissions at each stage. There is correlated behavior between the leader's and follower's targets, since the costs and environmental effects are linked through common constraints. In line with many bi-level models, we take an optimistic viewpoint where, if there is more than one optimal solution for a given leader's decision from the member's point of view, the follower will select the most favorable one for the leader. This results in a level of cooperation where both players try to achieve more reductions in emissions while not destroying the leader's profitability. The reason of this assumption is that, although incoherent from a theory point of view, the coincidence of cost-based optimization and environmental sustainability is reasonable since we consider they have common interest of the decision makers. Such a choice reduces the complexity of the bi-level formulation and corresponds to the optimistic decisions commonly applied in the literature. Considering the cost and emission simultaneously, this model is more consistent with realistic decision making, in which economic benefits and negative externalizations should be CO₂ considered at the same time.

The Proposed Model is described with the following properties:

1. The second problem is expressed without any variable of the first one in its constraints.
2. For any upper-level feasible solution, there is at least one lower-level feasible solution.

A well-known special case of this problem is the discrete competitive facility location problem. See also [34]. As a result, the underlying structural constitutive model of interest is postulated as:

$$\begin{aligned}
P_1 : \quad & \min_{x, x', y, y'} F_1(x, x', y, y') \\
& \min_{x, x', y, y'} F_2(x, x', y, y') \\
\text{s.t.} \quad & G_1(x', x) \leq 0, \\
& \max_{x, x', y, y'} f_1(x, x', y, y') \\
& \min_{x, x', y, y'} f_2(x, x', y, y') \\
& G_2(x, x', y, y') \leq 0,
\end{aligned}$$

where x', y' and x, y represent vectors of continuous and binary variables at different hierarchical levels, respectively. The functions F , G , f , and g are assumed to be linear. Bard and Falk highlighted the following difficulties in the computation of algorithms for solving BLPP. In their 1990 paper [35], they made the following three general observations with respect to bilevel mixed integer programming problems:

Remark 1: The solution of the relaxed Bi-objective Bi-Level Programming Problem (*BLPP*), in which the upper- and lower-level problems as linear programming problems, is not a valid upper bound for the solution of the mixed-integer BLPP.

Remark 2: Admissible solutions in the induced region of the relaxed BLPP cannot in general be disregarded.

Remark 3: Feasible integer solutions of the relaxed BLPP considering the restrict-edness of some lower-level variables cannot be simply neglected. In this context, BLPP refers to a form of the problem where lower-level variables are relaxed to be continuous.

3 The Solution Approach

This section present the fundamental principles of the Chemical reaction optimization (*CRO*) algorithm, elucidates the *BBO* model that addresses cost and CO_2 emissions, and examines the coding and decoding mechanisms of the *CRO* algorithm, highlighting solution representation and transformation during the optimisation process.

3.1 Preliminary Considerations Regarding CRO

This subsection describes the basic ideas and mechanisms of the *CRO* algorithm. Motivated by the motion principles of chemical reactions, *CRO* provides a new searching protocol that efficiently balance the exploration and the exploitation in the search space. A step-by-step process for how to implement the algorithm is structured below:

3.1.1 Chemical Reaction Optimization for Single-Objective Optimization

CRO is a *nature-inspired meta-heuristic* algorithm which was initially proposed for single-objective optimization based on the dynamics of chemical reactions [36, 37]. In this work the *CRO* methodology is generalized for bi-objective optimization tasks. A candidate solution is viewed as a molecule in the space whose structure is determined by a set of decision variables $(x_1, x_2, \dots, x_n)$ and is characterized by an energy, $\mathbf{E}(M) = [f_1(\mathbf{x}), f_2(\mathbf{x})]$, whose components denote usual objectives, cost and CO_2 emissions, for example. The *CRO* scheme is an efficient methodology for searching and utilizing the potential solution space through the molecular interaction and transformation simulations. The algorithm is based on the principle of the minimization of the single objective function ($f(\mathbf{x})$) combined with reaction operators that describe chemical processes and allow the population of solutions evolve from one generation to the next towards optimality. Candidate solutions are modeled as molecules in the framework of *CRO*. A molecule M is represented by a vector:

$$M = (x_1, x_2, \dots, x_n)$$

where (x_i) is the decision variables. Each molecule’s performance is then assessed on the optimal function. The expected value is denoted by $f(x_1, \dots, x_n)$. Where $E(M)$ is the energy (fitness) of the molecule M .

Reaction Operators

The *CRO* employs several types of reaction operators to convert molecules and to navigate solution space:

- **Synthesis:** This operation merges the input molecules M_1 and M_2 to yield a new molecule M_3 by: $M_3 = \alpha M_1 + (1 - \alpha)M_2$ where α is a blending parameter in $[0, 1]$.
- **Decomposition:** A molecule M can be decomposed to simpler elements and a selective refinement of the decision variables.
- **Interchange:** The operator that exchanges decision variables of two molecules to promote diversity in the solution space. For example:

$$M_1 = (x_1, y_1) \quad \text{and} \quad M_2 = (x_2, y_2) \quad \Rightarrow \quad M_1 = (x_2, y_1) \quad \text{and} \quad M_2 = (x_1, y_2)$$

- **Ionization:** This operation will slightly perturb the decision-making variables of a molecule to find local changes:

$$M' = (x_1 + \Delta x_1, x_2 + \Delta x_2, \dots, x_n + \Delta x_n)$$

and Δx_i minor disturbances in the variables.

Evaluation and Optimization Process

The objective function value ($E(M)$) is calculated for each molecule after reaction operators are applied. The *CRO* procedure successively chooses molecules, applies

reaction operators, and updates the population until a stopping criterion (e.g., a maximum number of iterations) is satisfied.

- end while $t \geq \text{max iterations}$ end do
- Randomly choose a molecule (M_i) from the population.
- Act by reaction operators to convert (M_i).
- Check if the new molecule is accepted ($E(M_i)$).
- Merge the current and new populations according to Pareto dominance or other rules.

This process is iterated until the current solution finds a local optima and the algorithm converges to the best solution which is denoting by best molecule.

The Pseudocode for CRO Algorithm

Algorithm 1 CRO Algorithm

Require: Problem definition, population size N , max iterations T , reaction parameters

Ensure: Best solution found, *Best Solution*

```
1: Initialization:
2: Define problem parameters:  $N$ ,  $T$ , reaction probabilities, energy thresholds.
3: Generate an initial population of  $N$  molecules  $M_i$  with random solutions.
4: Evaluate the fitness  $f(M_i)$  of each molecule.
5: Assign potential and kinetic energy levels to all molecules.
6: Store all molecules in population List.
7: Set Best Solution to the molecule with the best fitness in population List.
8: for  $t = 1$  to  $T$  do ▷ Main Loop
9:   for each molecule  $M$  in population List do ▷ Reaction Phase
10:    Choose a reaction type (uni molecular or inter molecular) based on
    probabilities.
11:    if Unimolecular Reaction then
12:      Perform decomposition or isomerization:
13:      (a) For decomposition, split  $M$  into  $M_1$  and  $M_2$ , and evaluate
         $f(M_1)$ ,  $f(M_2)$ .
14:      (b) For isomerization, perturb  $M$  to generate  $M'$ , and evaluate
         $f(M')$ .
15:      Update population List with  $M_1$ ,  $M_2$ , or  $M'$  if they dominate  $M$ .
16:    else if Intermolecular Reaction then
17:      Select another molecule  $M_2$  from populationList.
18:      Perform collision or synthesis:
19:      (a) For collision, combine  $M$  and  $M_2$  to create  $M'$ .
20:      (b) For synthesis, merge  $M$  and  $M_2$  into  $M'$ .
21:      Evaluate  $f(M')$ , and replace  $M$  and/or  $M_2$  with  $M'$  if it dominates.
22:    end if
23:  end for
24:  Update Best Solution with the molecule having the best fitness in population
    List.
25: end for
26: return Best Solution.
```

3.2 Applying CRO of the Proposed Model

The *CRO* algorithm efficiently searches the solution space by seeking out good trade offs with well balance in fermentation and exploration in the process of different new materials. Energy-based operators it employs prevent it from prematurely getting trapped and from neglecting parts of the solution space. In addition, *CRO* preserves

diversity among the solutions and since capturing a number of solutions is very important in finding different alternatives and prevents stagnation. These features render *CRO* a strong and powerful optimization method, particularly in relation to methods that are slow to converge or lack exploration.

3.2.1 CRO Algorithm for Bi-Objective Bi-Level Optimization

CRO optimization technique composed of several stages that imitates how chemical reactions are took place in life, molecules evolved in order to get optimized, such as synthesis, decomposition, exchange, ionization etc. The operators provide the ability to search the solution space and aid in the search for optimal solutions. The *CRO* algorithm can be extended to tackle *BBO* problems, wherein there are two decision making levels, namely the leader and the follower, optimizing their objectives concurrently. The leader in these types of problems has the ability to make decisions at the upper level, and a follower can make decisions at the lower level according to the leader’s decisions. The *CRO* takes into account the hierarchical nature of the problem (decomposition into sub-problems with molecules as candidate solutions). The operators of *CRO* are adapted to search and optimize the objectives of the leader and follower simultaneously. And examples include synthesis, which assembles solutions to search the common solution space, decomposition, which targets only one decision-maker’s objectives, and ionization, which introduces local perturbations. This process is repeated multiple times until for which we find Pareto-optimal solutions, beneficial for both decision-makers without compromising the rank ordering. The *CRO*’s features of balancing exploration with exploitation, and hierarchical search make it very suitable for multi-objective optimization. Nevertheless, the computational burden that emerges from bi-level problems, in particular when multiple objectives are considered, needs appropriate parametrization of population size and reaction rates for their efficient operation. Despite these challenges, *CRO* has evolved to be a formative methodology for solving complex optimization problems in subject areas spanning from engineering design to financial planning.

Molecule Representation and Variables

In a *BBO*, molecules are employed to represent possible answers and each molecule is formed with two units that are similar to the *leader’s* and *follower’s* decision variables. Every molecule (M) in *CRO* is a solution vector, splitted into two, having decision variables for the leader (upper-level decision maker) and for the follower (lower-level decision maker). The molecular weight (M) of a molecule is given by:

$$M = (x, y)$$

Where:

- (x) is the decision variables of the leader.
- (y) is the decision variables for the follower.

As a pair (x) and (y), these are the candidate solution of the bi-level optimization problem.

Interaction of Molecules with Objectives

In a *BBO* each player optimize their objectives according to both (x) and (y). For example:

- The *leader* seeks to minimize two objectives ($F_1(x, y)$) and ($F_2(x, y)$), where (x) is used to determine the leader’s objectives and the decisions of the follower (y).
- The single *follower* will minimize their own objective ($G(x, y)$), the response to the leader’s decision (x) via (y).

CRO Reaction Operators and Variables

Using reaction operators ? synthesis, decomposition, interchange, and ionization ? *CRO* transforms molecules (solutions) and explores the search space. The leader and follower variables are updated by each operator to reach optimal solutions:

- **Synthesis:** Merge the leader and follower variables to obtain new solutions. For example, by synthesizing the two molecules ($M_1 = (x_1, y_1)$) and ($M_2 = (x_2, y_2)$) a new molecule (M_3) is created by crossing elements of (x_1), (x_2), (y_1), and (y_2):

$$M_3 = (\alpha x_1 + (1 - \alpha)x_2, \beta y_1 + (1 - \beta)y_2) \quad (30)$$

where (α) and (β) are weighting factors.

- **Decomposition** This operator decomposes a molecule into simpler parts, hence uncoupling the leader and follower variables. By decomposing a molecule $M = (x, y)$, the algorithm is able to refine x and y independently, providing a better focus on one of the leader or follower’s task.
- **Interchange:** *CRO* exchanges portions of decision variables within two molecules. For example, it could swap leader variables x of two molecules while maintaining their follower variables y , which will introduce solution diversity for the space in:

$$M_1 = (x_1, y_1), \quad M_2 = (x_2, y_2) \quad \rightarrow \quad M_1 = (x_2, y_1), \quad M_2 = (x_1, y_2)$$

- **Ionization:** This operator randomly modifies some decision variables values of the molecule at hand, probing local variability of solution. For example:

$$M_1 = (x, y) \quad \rightarrow \quad M'_1 = (x + \Delta x, y + \Delta y)$$

where Δx and Δy are perturbations used for sampling nearby solutions.

Evaluation of Molecules

After applying the reaction operators, each molecule is assessed according to the leader and follower objectives. Since the problem is bi-objective, Pareto Dominance is applied for solution comparison. A molecule M_1 dominates M_2 if it has better performance in at least one objective and is equally or better than the other. This guarantees that both objects are taken into account when choosing the molecules for subsequent reactions.

3.2.2 Coding and Decoding

In a CRO framework, how to code upper-level and lower-level decision variables in terms that can represent a solution in a sensible way is even more complicated. Such a representation is necessary in order to meaningfully search through the solution space in the optimization process.

Representation of Solutions solutions are encoded as molecules in the *CRO* algorithm. Every molecule, (M), is represented by a vector:

$$M = (x, y) \quad (31)$$

where: $x = (x_1, x_2, \dots, x_n)$ is a vector that stands for the upper-level decision variables. These decision variables include options with material selection and investment options, then impact the construction cost and CO₂ emissions. ($y = (y_1, y_2, \dots, y_m)$) are the lower-level decision variables. These are operational or production scheduling decisions in order to maximize the profit and minimize CO₂ emission in operation phase or product life cycle. The energy of each molecule is related to the objective functions of the bi-objective optimization problem. Namely, the energies are given by:

- **Upper-Level Energy:**

$$E_{\text{upper}} = \alpha \cdot \text{Cost}(x) + \beta \cdot \text{CO}_2(x)$$

where (α) and (β) are the scaling factors that weight the objectives.

- **Lower-Level Energy:**

$$E_{\text{lower}} = \gamma \cdot (-\text{Profit}(y)) + \delta \cdot \text{CO}_2(y)$$

where (γ) and (δ) balance the objectives at the lower level.

The total energy of each molecule, E is minimized during the optimization with: $E_{\text{total}} = E_{\text{upper}} + E_{\text{lower}}$ solutions closer to the global lower bound are increasingly favorable as they require a lower amount of construction cost, profit, and CO₂ emissions trade-offs. Decoding Solutions from *CRO* The decoupling decoding (re-activation) operation is to translate the encoded decision variables back to practical construction plans and operation strategies. Following identification of a group of molecules which are potential solutions using the *CRO* algorithm, the following decoding steps are performed:

Upper-Level Decoding The decision variables at an upper level (x) are mapped into actual construction decisions. This includes:

- Specifying material purchases.
- Establishing project timelines.
- Investing in emissions-reducing technologies.

These decisions illustrate the compromise between the cost of construction and CO₂ emissions.

Lower-Level Decoding The lower-level decision variables (y) are decoded into operational responses which tend to maximize profit in parallel with minimizing CO₂ emissions. This includes:

- Determining optimal product quantities.
- Establishing pricing strategies.
- Implementing energy efficiency measures.

These decoding procedures convert the solutions of the *CRO* algorithm into realistic, meaningful construction and operational actions.

Algorithm 2 CRO Algorithm for The Proposed Model

Require: Outer and inner-level objectives f_1 , f_2 , population size N , max iterations T , reaction parameters

Ensure: Pareto-optimal solutions $\mathcal{P}$ for the bi-objective problem

```

1: Initialization:
2: Define parameters:  $N$ ,  $T$ , reaction probabilities, energy thresholds.
3: Create an initial population of  $N$  molecules with random outer-level solutions.
4: Solve the inner-level optimization for each molecule.
5: Compute  $f_1$  and  $f_2$  for all molecules.
6: Assign energy levels (potential and kinetic) to each molecule.
7: Initialize Pareto front  $\mathcal{P}$  with non-dominated solutions.
8: for  $t = 1$  to  $T$  do ▷ Main Loop
9:   For each molecule  $M$  in the population, choose a reaction type (unimolecular
   or intermolecular).
10:   if Unimolecular Reaction then
11:     Perform decomposition or isomerization:
12:     (a) For decomposition, split  $M$  into two new molecules and evaluate
   them.
13:     (b) For isomerization, perturb  $M$  to generate a new molecule  $M'$ 
   and evaluate it.
14:     Replace  $M$  in the population if new molecules dominate or improve
   diversity.
15:   else if Intermolecular Reaction then
16:     Select another molecule  $M_2$  and perform collision or synthesis:
17:     (a) For collision, combine  $M$  and  $M_2$  into a new molecule  $M'$ .
18:     (b) For synthesis, merge  $M$  and  $M_2$  into a single molecule  $M'$ .
19:     Evaluate  $M'$  and update the population if  $M'$  dominates or improves
   diversity.
20:   end if
21:   Update Pareto front  $\mathcal{P}$  with non-dominated solutions in the population.
22: end for
23: 13. Return Pareto front  $\mathcal{P}$ .

```

The above algorithm 2, addresses bi-objective, bi-level optimization, with certain fluency and Underlying influence in using *CRO*. It provides initialization, reaction propagation and iterative refinement to explore the solution space by optimizing hierarchical objectives.

3.3 Computational Complexity

In this subsection, we analyze the time computational complexity of the proposed *BMCRO*. If N_u and G_u represent the upper population size and number of generations at the upper level, and also N_l , G_l and M the lower population size, number of generations in the lower layer and number of objectives, respectively. The time complexity for a deepened single-objective algorithm can be formulated as:

Upper-level complexity $\times$ Lower-level complexity. If $G_u = G_l$, the complexity is:

$$(G_u \times N_u) \times (G_l \times N_l) \approx O(GN)^2.$$

As for multi-objective, the sorting takes most of the upper procedure. So now you need to care about the time complexity. As per, the time complexity of the QNDSA becomes:

1. **The best case**—where each individual non-dominates the other individuals in the entire population. Therefore the optimum solution will take time complexity of $O(N \log N)$.
2. The **average case**, which is when we have a Number of Fronts (NF) such that $NF < N$. Its average case time complexity is $O(N \log N)$ as well.
3. **Worst case scenario**: if all fronts are single-solution. The worst-case complexity is $O(N^2)$. It is worth mentioning that very few of such situation happens in multi-objective optimization.

The multi-objective BLOP complexity becomes:

$$O(N \log N) \times [(G_u \times N)] \approx O(N^2 G \log N)$$

or in the best and average case, and:

$$O(N^2) \times (G \times N) \approx O(GN^3)$$

the (rare in multi-objective optimization) worst case. The majority of the classical multi-objective ranking methods can be based on a non-dominated sorting $O(MN)^2$ [38], [39]). Hence, the total multi-objective bi-level classical procedure will:

$$O(MGN^3).$$

Clearly, applying NSGA-II with TS into the bi-level EA can greatly enhance the efficiency of the algorithm and save computation significantly.

4 Computational Algorithm

In this section outlines the computational and software setup used for this investigation. The system in use is a Windows 10 Home derivative (version 22H2), built with Windows Feature Experience Pack 1000.19060.1000.0. The computer's hardware is an Intel(R) Core(TM) i7-8565U CPU at 1.80 GHz and it's capable of up to 1.99 GHz, with a 64-bit, x64-based processor. 8GB 7.86 GB usable of installed RAM, which is suitable for moderate multitasking and light with moderate-sized data. It is also pen and touch enabled and also has 10 touch points making it suitable for various interactive applications. The main software package utilized for post processing and analysis is MATLAB, which has been developed and adapted (under software revision date of July 29, 2023). The installed size is 1.32 KB (1,360 bytes) with disk usage of 4.00 KB (4,096 bytes), and is an efficient environment for running computational tasks on such hardware.

The algorithm is tested on data with random sizes which are generated based on the data generation rules defined in 4.1 by the method in the examples are users to illustrate the Accuracy of the model in 4.2 and compare with models in 4.3 This section also discusses the sensitivity analysis of the model's key parameters in 4.4.

4.1 Data Generation Rule

In our research, we integrated the geographical locations of the strategic nodes?customers, depots, and factories?using MATLAB language, namely, we simulated the coordinates of the locations of each node as representations of a matrix. Three matrices: `posc`, `posd` and `posf` were used to describe the different types of entities, there each row of the matrices represented the position of an individual entity, and the columns correspond to (x) and (y) coordinates in a two-dimensional representation. Customer locations were assigned to `posc` and each row contained a specific spatial location in the data set. This organizational structure based on matrices facilitated spatially systematical organization of the various types of entities and the decoupling to easily and efficiently manipulate the data. With these positional matrices, we established a strong platform for subsequent distance calculations and spatial analysis. Pairwise distances between these matrices were calculated using MATLAB's `pdist2` function, allowing for a structured analysis of spatial inter-relationships between the entities. The statement (`D = pdist2(X,Y)`) creates a distance matrix (`D`) in which every element stores the distance between each point in matrices (`X`) and (`Y`). Distances between the set of customer and depot locations, for instance, were calculated using `disDtoC = pdist2(posc,posd)`, providing essential information for reasoning about how these two sets interact. Also, we computed intra-set distances for customers by using `disCtoC = pdist2(posc, posc)` and generated a square matrix that showed spatial ties between customer locations. In addition to the distance calculations, the dataset boundaries were determined by the minimum and maximum values in all the combined matrices (`MINLIM = min([posf; posd; posc])` and `MAXLIM = max([posf; posd; posc])`), representing the spatial extent of the data. Additionally the number of elements of all the relevant arrays was computed for further analysis using `numel`, which gave scalars for each array (e.g., `nd = numel(posd)`, `nf =`

`numel(posf)`, `nc = numel(posc)`). The computed minimum distances and their specie counts give the underlying rules for our analysis purposes in our study-scale context. Also, the parameter used for the distance metrics is an important issue of spatial analysis. The Euclidean distance metric was the one used here, as it is the measure of direct straight-line distance from any point to any another, well fitted to precision on the direct proximity in space. But, conditional on logistical constraints, other distance metrics as, for example, Manhattan distance, which is the sum of the absolute differences in coordinates, may be examined, as well. The broad scope of distance measures accepted by the `pdist2` function in MATLAB enables flexible analysis pipelines corresponding to the spatial topology and mobilities distinctive of different classes of entity.

4.2 Accuracy of the model

We have evaluated the accuracy and convenience of the proposed model through testing its performance using three optimization algorithms as: *CRO*, *NSGA – II*, and (*TS*) in our study. The purpose was to investigate if the model generates meaningful and logical outcomes and successfully addresses the optimization problem in question. After solving the model, we noted the following two important facts that confirm its feasibility: a routing problem between factories and centers and a green location problem between centers and customers. These results are as expected; the model appears to incorporate the salient interactions and constraints of the domain. Figure (1) shows the distribution of factories, centers and customers for two different problem sizes. Furthermore, Figure (2) depicts the connection among factories, facilities and customers, which also corroborates the validity in modeling the underlying optimizing processes of the model. The outcome of this comparative analysis not only demonstrates the importance of using a variety of the algorithms in the assessment of the model but also validates the optimal performance of the model to deal with the complexity of the task at hand. These findings highlight the effectiveness of the proposed method for simulating real operational issues.

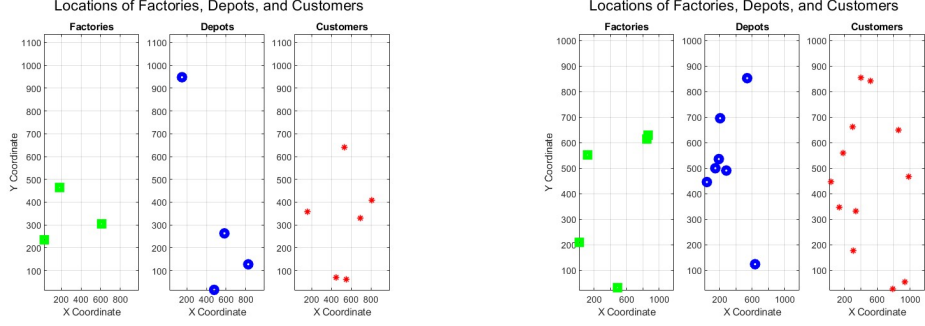


Fig. 1: Distribution Of the Factories, depotes and Costumers

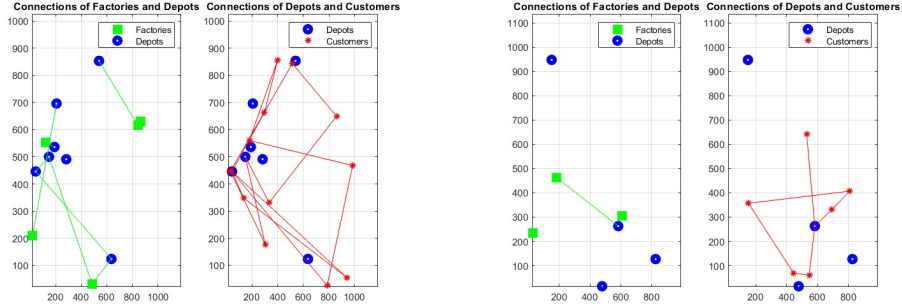


Fig. 2: Connection Of the Factories, depotes and Costumers

4.3 Comparison of the Methods

The *CRO* algorithm comparison with the others powerful optimization methods, Tabu Search *TS* and Non-dominate sorting genetic algorithm *NSGA – II*, was performed on three data sets (small, medium, and large) respectively. In this review, we were interested in the performance and scalability of each method as it is applied to different instances of the problem. Comparison was made in several aspects: The number of solutions, the time took to achieve these solutions. The MID (Mean Ideal Distance), diversity. Leader objective value, Follower objectives value. The comparisons of the algorithms and the time intervals are summarized in Table 3, Table 4 and Figure 3, respectively. This analysis allows us to gain an understanding of which method can cope with higher degrees of complexity and size of data.

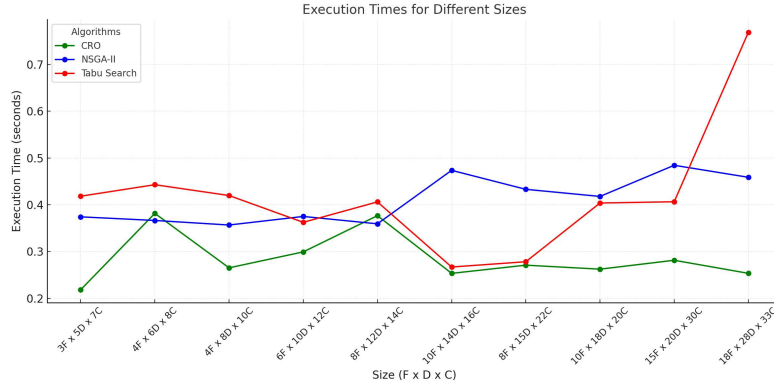
The comparison of results of *CRO*, *NSGA – II* and *TS* demonstrates the strengths and drawbacks of each in terms of trade-off between the two objectives: maximizing profit (OBJ1) and minimizing CO_2 emissions (OBJ2). *CRO* exhibits better computational efficiency, in terms of faster computational time in different problem sizes, however compromise with regard to leader and follower solutions with regard to OBJ1 and OBJ2 are in favor

Table 3: Comparative Table

Time		CRO							NSGA-II							Tabu Search						
Size	NO	Time	MID	Diversity	Spacing	<i>Leader</i> <i>OBJ2,OBJ1</i>	<i>Follower</i> <i>OBJ2,OBJ1</i>	NO	Time	MID	Diversity	Spacing	<i>Leader</i> <i>OBJ2,OBJ1</i>	<i>Follower</i> <i>OBJ2,OBJ1</i>	NO	Time	MID	Diversity	Spacing	<i>Leader</i> <i>OBJ2,OBJ1</i>	<i>Follower</i> <i>OBJ2,OBJ1</i>	
$F \times D \times C$ _{3 5 7}	3	0.218285 sec	1.1111	59.111	0.001152	107,2549	582,1954	10	0.374175 sec	1.4717	181.65	0.77864	210,2589	485,1683	12	0.41828sec	1.4245	59.453	0.86544	291,2568	697,1541	
$F \times D \times C$ _{4 5 8}	5	0.381588 sec	1.3418	146.64	0.057351	238,3311	1079, 1181	10	0.366444 sec	1.3486	218.07	0.45415	238,3396	976,1254	3	0.442882 sec	1.2748	76.694	0.92595	223,3193	761,1469	
$F \times D \times C$ _{4 8 10}	4	0.265184 sec	1.3418	142.36	0.24314	228,4582	972,1848	10	0.356717 sec	1.4359	267.09	0.32395	204,4579	1385,1395	3	0.419768 sec	1.4075	118.79	0.25576	311,4601	831,2029	
$F \times D \times C$ _{6 10 12}	6	0.299248 sec	1.5729	59.111	0.91737	465,3152	465,3152	9	0.375012 sec	1.6058	327.33	0.25414	174,6050	1363,1983	4	0.362457 sec	164.38	1.5424	0.84396	331,5922	1190,2096	
$F \times D \times C$ _{8 12 14}	6	0.376700 sec	1.5834	240.72	0.31166	312,6785	1610,2100	10	0.359279 sec	1.9536	383.4	0.22709	285, 6769	1676,1894	4	0.406104 sec	1.9136	1.9136	0.23083	326,6496	1762,1838	
$F \times D \times C$ _{10 14 16}	5	0.253389 sec	1.9964	406.76	0.27648	483,8638	1874,2597	10	0.473595 sec	1.9431	415.59	0.17495	471,8686	1716, 2695	4	0.266926 sec	2.0035	217.07	0.17154	608,8694	2090,2361	
$F \times D \times C$ _{8 15 22}	7	0.270885 sec	1.234	407.11	0.23316	455,9268	1880,3318	10	0.433007 sec	1.9167	434.96	0.23585	428,9253	1978,3190	4	0.278216 sec	1.9863	231.11	0.1293	595,9341	2362, 2826	
$F \times D \times C$ _{10 18 20}	6	0.262352 sec	2.1825	330.69	0.20823	563,9052	2491,3878	10	0.417815 sec	2.0533	436.97	0.1617	478,9308	2173	5	0.403734 sec	2.1868	239.52	0.18757	577,9073	2272,4067	
$F \times D \times C$ _{15 20 30}	3	0.281357 sec	2.4367	231.44	0.12388	859,10282	3700,4612	3	0.484370 sec	1.9633	192.23	0.21683	763,10165	3617,4735	3	0.406431 sec	2.2821	192.22	0.21618	972,10204	2866,5386	
$F \times D \times C$ _{18 28 33}	4	0.253424 sec	2.2854	344.49	0.22966	701,14268	3381,6085	3	0.458729 sec	2.3147	261.7	0.24187	604,14329	3882,5744	3	0.768795 sec	2.1345	261.03	0.16288	695,14182	3811,5645	

Table 4: Examination of Time Intervals

Size \ Time	CRO	NSGA-II	Tabu Search
$F \times D \times C$ 3 5 7	0.218285 sec	0.374175 sec	0.41828 sec
$F \times D \times C$ 4 6 8	0.381588 sec	0.366444 sec	0.442882 sec
$F \times D \times C$ 4 8 10	0.265184 sec	0.356717 sec	0.419768 sec
$F \times D \times C$ 6 10 12	0.299248 sec	0.375012 sec	0.362457 sec
$F \times D \times C$ 8 12 14	0.376700 sec	0.359279 sec	0.406104 sec
$F \times D \times C$ 10 14 16	0.253389 sec	0.473595 sec	0.266926 sec
$F \times D \times C$ 8 15 22	0.270885 sec	0.433007 sec	0.278216 sec
$F \times D \times C$ 10 18 20	0.262352 sec	0.417815 sec	0.403734 sec
$F \times D \times C$ 15 20 30	0.281357 sec	0.484370 sec	0.406431 sec
$F \times D \times C$ 18 28 33	0.253424 sec	0.458729 sec	0.768795 sec

**Fig. 3:** Examination of time intervals

of lesser diversity in larger problems. *NSGA-II* is better at searching for the solution which is indicated by its higher diversity and a well-maintained MID (Mean Ideal Distance), to portray a balanced convergence toward the optimal trade-offs between OBJ1 and OBJ2. Though considerably less efficient for such as complex situations, *Tabu Search* provides the lowest MID values, indicative of its ability to converges to exact solutions. Its diversity scores are, however, lower than our approach and it may not perform well in problems in which a larger number of trade-off solutions is to be explored. In general *CRO* is a very efficient algorithm, especially in situations where computational speed is a priority. *CRO* is rapidly converging to acceptable solutions, motivated by molecular energy transitions, and has both good computational efficiency and an excellent initial performance. This makes it well-suited for time-sensitive work that requires quick, efficient results. *NSGA-II* is generalist, tonsup profit maximization and (CO_2) reduction whereas *TabuSearch* leads to precision on

less complex landscapes. only the former is efficient when fast approximate solution is required, *NSGA-II* can be used for a global solution and *TabuSearch* prefers to be used for more focused attempts with high precision.

4.4 Sensitivity Analysis

In this section, we concentrate on the sensitivity analysis of the model P_1 , particularly analyzing the impacts of three selected parameters on the performance of the model. Sensitivity analysis is carried out on the factors which are not well established in the literature.

4.5 Sensitivity Analysis of cri

In the subsequent subsection, we conduct a sensitivity analysis for the parameter cri under consideration, varying its value from 1 to 20. The impact of this parameter variation is analyzed, and the results are presented in the following table (5): The analysis of sensitivity reveals for CRO , a steady response to the varied transportation cost (cri) that affects leader and follower objectives. Leader's objectives such as minimization of construction cost and CO2 emission were subject to considerable variability, while the construction cost show continuously increasing trend by increasing (cri) very sharper in extreme levels, the CO2 emission had moderate to high peak at middle levels and drop off slightly for extreme levels. Also, follower objectives showed a high degree of sensitivity, profitability decreased sharply as transportation cost increased, and a slight recovery at some (cri) values were noticed. Follower per capita CO2 emissions grew steadily higher, as followers were being squeezed by high transportation costs. In general, CRO was robust across different values of cri and able to preserve feasible solutions despite changes in the objectives. But the point is to show the need to manage transportation costs to achieve a balance of economic and environmental performance.

4.6 Sensitivity Analysis for Environmental Impact

In the following subsection, a sensitivity analysis is performed for the parameter eri of interest by varying its value from 1 to 20. The effects of these variations are examined, and the findings are summarized in the accompanying Table (6) The sensitivity analysis of from environmental impact eri of CRO algorithm showed the significant response in the leader objective and the follower objective. The goals of leaders, such as reducing environmental costs and impacts, showed moderate variability. The environmental cost increased steadily according to eri with larger spikes in its peaks in higher values ($Cri = 20$), while to environmental impact varied regularly to higher values reflecting the challenges of adjusting to strict conditions. Follower goal dynamics were highly sensitive, as performance criteria like environmental accommodation oscillated as the eri level varied. Profitability (OBJ1 for followers) exhibited steep increases for intermediate levels ($Cri = 8$) but a sharp decrease at extreme levels, indicating complex tension between environmental constraints and economic sustainability. Follower environmental goals (OBJ2) continued to rise, indicating that the pressure for a high level of environmental impacts did not decrease. In general, the performance of the

Table 5: Sensitivity Analysis of cri

Algorithm	NSGA-II				CRO				MOTS			
Cri	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>
1	6848	1912	8022	6263	6630	1752	10028	5266	6430	1984	9024	6220
4	13522	2007	5073	9705	10345	1871	4623	10125	12993	2061	1975	12733
5	12403	1930	3241	10507	13370	1891	598	13150	9229	1760	4739	1959
8	12876	2130	2227	12202	12747	1741	1892	12457	9428	2022	5211	9208
10	11608	2175	4301	10837	10472	1643	5375	9813	14814	4003	1257	14176
15	11666	2185	3735	11506	9380	2098	6021	9220	9432	1094	7034	953
20	14631	1623	1315	14351	13342	1689	3074	12682	12964	1786	2982	12794

Table 6: Sensitivity Analysis of eri

Algorithm	NSGA-II				CRO				MOTS			
eri	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>
1	8177	2050	8046	7033	8247	1966	7082	7997	6612	1918	8257	6472
4	6377	2855	7653	6067	8561	3484	7069	6721	6457	2537	8043	5677
5	6925	3003	8596	6126	6847	3449	8795	5927	6686	3842	8236	6586
8	6245	2147	9329	6026	7005	2820	8500	6855	7075	4995	9006	6369
10	6965	4137	9305	6050	9077	3920	7888	7417	8844	6494	8750	6525
15	15446	6367	7568	7831	9712	6612	8446	6953	10255	7122	8417	6852
20	10709	5303	8522	6743	12078	6367	9108	6087	10101	6875	8783	6572

CRO is robust for different *eri* settings and feasible solution can still be obtained even when the objectives drastically change. Nevertheless, the study highlights the need to address the negative impact on the environment in order to maintain an integrated optimization of economic and environmental dimensions.

4.7 Sensitivity Analysis for Cost Conveying via Vehicles

In the following sub-section, a sensitivity analysis is performed on the parameter; *ccl* in question, by changing its value in a range from 1 to 20 systematically. You have made a proper analysis of the effects of these two variations and the results will be shown in the table (7). The sensitivity test results for *CRO* algorithm showed that the *CRO* is sensitive with per-unit cost of products conveyed by vehicle type *ccl* of leader and follower objectives. There was large variation of preference on leader objectives such as reduction of environmental costs and pollution. Environmental costs continued to increase with increasing *ccl*, with the largest relative changes made at larger *ccl*. Environmental effects had some stability with small variations. Followers' goals were found to be highly sensitive to variations in *ccl*. The overall profit decline seen at very high *ccl* values can on occasion be reversed at moderate levels, suggesting a more intricate relation between cost pegs and long-term performance. Environmental effects of followers were all higher than the leaders, reflecting the continuing environmental issues of increased transportation costs. In general, the *CRO* algorithm presented a robust performance along different *ccl* levels, as it was able to preserve feasible solutions, even when objectives varied. The work underscores the necessity of strong economic pressure for the trade-offs which must be made between the economic and environmental goals to not give in to a purely transshipment policy.

5 Conclusion and Future Work

The research presented in this paper addresses the pressing need for sustainable logistics by formulating a novel Competitive Bi-level Multi-objective Green Location Routing Problem. This problem models the hierarchical decision-making process between two competing firms?the leader and the follower?where the leader determines depot locations and the follower optimizes product distribution routes. The proposed bi-level mixed-integer linear programming framework simultaneously minimizes economic costs and environmental impacts, specifically CO2 emissions, reflecting the dual priorities of cost efficiency and sustainability in modern supply chains. To solve this complex bi-objective bi-level problem, the study introduces the Bi-level Multi-objective Chemical Reaction Optimization algorithm, a metaheuristic inspired by chemical reaction processes. The method effectively balances exploration and exploitation in the solution space, maintains diversity, and avoids premature convergence, which are critical for addressing the multi-level and multi-objective nature of the problem. The algorithm?s design includes specialized reaction operators?synthesis, decomposition, interchange, and ionization?that operate on both leader and follower decision variables to iteratively improve solution quality. We provide a thorough comparative analysis of the optimization algorithms, BMCRO algorithm, TS, and

Table 7: Sensitivity Analysis Cost of Conveying Products via Vehicle Types

Algorithm	NSGA-II				CRO				MOTS			
ccl	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>	<i>Leader OBJ1</i>	<i>Leader OBJ2</i>	<i>Follower OBJ1</i>	<i>Follower OBJ2</i>
1	8410	2106	8211	7143	8333	2034	8104	7002	6710	1903	8311	6522
4	6501	2914	7717	6200	6400	2800	7604	6112	6601	2605	8112	5700
5	7243	3170	8600	6240	7123	3050	8528	6132	6832	3800	83043	6603
8	6447	2256	9507	6112	6338	2143	9456	6072	7238	5882	9234	6316
10	7123	4285	9465	6133	7054	4112	9304	6000	8908	6554	8765	6474
15	15505	6500	7708	7906	15052	6405	7610	7832	10308	7106	8550	6800
20	10800	5451	8605	6834	10743	5300	8503	6700	10450	6900	8800	6603

NSGA-II when dealing with bi-level multi-objective problems using datasets in different scales. Every technique has its advantages: the proposed algorithm with its high computational efficiency, NSGA-II that maintains good trade-offs among objectives, and TS with its accuracy over simple problem landscapes. A sensitivity analysis highlights the importance of under-examined parameters in algorithm performance, shedding light on model tuning and robustness. The proposed model assumes perfect information sharing between the leader and follower, as the leader's decisions rely on precise forecasts and information provided by the follower. In practice, information asymmetry can lead to suboptimal or unstable outcomes because the leader might make decisions based on incomplete or inaccurate data, and the follower may strategically withhold or misrepresent information. This can cause non-cooperative behaviors and deviations from the optimistic equilibrium. To capture these effects, future model enhancements could incorporate game-theoretic frameworks with incomplete information or signaling mechanisms, allowing analysis of how information asymmetry influences equilibrium stability and solution robustness. On the other hand, the current model focuses on a single follower scenario to maintain clarity and computational tractability. Introducing multiple followers who simultaneously respond to the leader's strategic decisions would significantly increase the complexity, as it transforms the lower-level problem into a multi-follower game with potentially competing objectives. In such a setting, the leader's problem becomes a multi-leader-follower or Stackelberg-Nash game, where followers compete among themselves while responding to the leader. This would require modeling the followers' interactions explicitly, possibly through equilibrium concepts such as Nash or generalized Nash equilibria. Extending the model to multiple followers is a promising direction for future research, enabling the study of competitive dynamics among transportation providers and their impact on overall supply chain sustainability and efficiency. Another important aspect that should be considered for future work is the multi-follower issue. The current bi-level model assumes a single follower and does not explicitly consider horizontal competition among multiple logistics providers. Introducing horizontal competition among followers would change the hierarchical game structure into a more complex multi-follower Stackelberg game with competitive interactions at the follower level. This competition could lead to different equilibrium outcomes, such as multiple equilibria or more aggressive follower strategies, potentially affecting the leader's optimal depot location and routing decisions. The leader might need to anticipate competitive behaviors and design contracts or incentives accordingly. Modeling such horizontal competition would require integrating game-theoretic solution concepts for multi-follower games and possibly employing equilibrium computation algorithms tailored for these settings. We recognize that, regarding stability over time amid changing market conditions or regulations, the model's static formulation represents a snapshot decision-making scenario. However, the bi-level structure inherently supports iterative re-optimization: as external conditions evolve, the leader can revise strategic decisions in subsequent planning cycles, and the follower adjusts routing accordingly. This iterative process maintains hierarchical stability by continuously aligning strategic and tactical decisions. To explicitly address dynamic environments, future research could integrate adaptive or robust optimization techniques or model

repeated interactions as a dynamic game. Nonetheless, the current model provides a credible and stable hierarchical framework by embedding the follower's best response within the leader's decision process and assuming cooperative behavior.

A list of abbreviations used in the manuscript and their meanings:

VRP	Vehicle Routing Problem
LRP	Location Routing Problem
GLRP	Green Location Routing Problem
GVRP	Green Vehicle Routing Problem
EV	Electrical Vehicle
BMCRO	Bi-level Multi objective Chemical Reaction Optimization
NSGA-II	Non-dominated Sorting Genetic Algorithm
TS	Tabu Search
CRO	Chemical Reaction optimization
BLPP	Bi-objective Bi-level Programming Problem.

Declarations

Ethics approval and consent to participate

This study was approved by the institutional ethics committee. All participants provided written informed consent before participating in the study. All methods were carried out in accordance with relevant guidelines and regulations.

Consent for publication

Not applicable.

Availability of data and materials

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Authors' contributions

Arsalan Rahmani contributed to the article through ideation, data analysis, theoretical framework development, simulations, and preparation of the manuscript. Berivan Faris

Azeez reviewed and supervised the work. Meysam Huseini conducted data analysis, editing, and assisted with writing and editing. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

References

- [1] Dantzig B.G., Ramser J.H. (1959) The Truck Dispatching Problem. *Manag. Sci.* **6**(1), 80?91.
- [2] Sinuany-Stern Z. (2021) *Handbook of Operations Research and Management Science in Higher Education*, vol. 309. Springer Nature.
- [3] Lee D.-H., Dong L. (2017) Green Logistics: An Overview. *Transp. Res. Part D: Transp. Environ.* **50**, 1?21.
- [4] Jabir E., Panicker V.V., Sridharan R. (2015) Multi-objective Optimization Model for a Green Vehicle Routing Problem. *Procedia-Soc. Behav. Sci.* **189**, 33?39.
- [5] Wang L., Gao S., Wang K., Li T., Li L., Chen Z. (2020) Time-Dependent Electric Vehicle Routing Problem with Time Windows and Path Flexibility. *J. Adv. Transp.* **2020**, 3030197.
- [6] Chen S., Meng Q., Choi T.-M. (2022) Transportation Research Part E Logistics and Transportation Review: 25 Years in Retrospect. *Transp. Res. Part E: Logist. Transp. Rev.* **161**, 102709.
- [7] Macrina G., Di Puglia Pugliese L., Guerriero F. (2020) The Green-Vehicle Routing Problem: A Survey. *Model. Optim. Green Logist.* **1**, 1?26.
- [8] Salimifard K., Raeesi R. (2014) A Green Routing Problem: Optimising CO₂ Emissions and Costs from a Bi-Fuel Vehicle Fleet. *Int. J. Adv. Oper. Manag.* **6**(1), 27?57.
- [9] Salimi Fard K., Raeesi R. (2017) An Environmentally Friendly Routing Problem Focusing on Fuel Consumption and CO₂ Emissions. *J. Clean. Prod.* **156**, 19?31.
- [10] Dukkanci O., Karsu O., Kara B.Y. (2022) Planning Sustainable Routes: Economic, Environmental and Welfare Concerns. *Eur. J. Oper. Res.* **301**(1), 110?123.
- [11] Dukkanci O., Kara B.Y. (2014) A Bi-Objective Location-Routing Problem with Demand Allocation. *Ann. Oper. Res.* **232**(1), 159?180.
- [12] Golden B.L., Raghavan S., Wasil E.A. (2008) The Vehicle Routing Problem: Latest Advances and New Challenges (Vol. 43). *Springer Science & Business Media*.

- [13] Deb K., Pratap A., Agarwal S., Meyarivan T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182?197.
- [14] Camacho-Vallejo J.-F., Lopez-Vera L., Smith A.E., Gonzalez-Velarde J.-L. (2022) A Tabu Search Algorithm to Solve a Green Logistics Bi-Objective Bi-Level Problem. *Ann. Oper. Res.* **316**(2), 927?953.
- [15] Abbassi M., Chaabani A., Ben Sai L. (2022) An Efficient Chemical Reaction Algorithm for Multi-Objective Combinatorial Bi-Level Optimization. *Eng. Optim.* **54**(4), 665?686.
- [16] Prins C., Prodhon C. (2014) Vehicle Routing Problem with Environmental Considerations. In: *Handbook of Metaheuristics*. Springer.
- [17] Glover F., Laguna M. (1997) *Tabu Search*. Kluwer Academic Publishers.
- [18] Pisinger D., Ropke S. (2007) A General Heuristic for Vehicle Routing Problems. *Comput. Oper. Res.* **34**(8), 2403?2435.
- [19] Toth P., Vigo D. (2002) Vehicle Routing Problem. *Discrete Optim.* **1**(1?3), 3?26.
- [20] Hoff E., Andersson H., Christiansen M., Hasle G., Lokketangen A. (2010) Ant Colony Optimization for Vehicle Routing Problems. *J. Optim. Theory Appl.* **145**(3), 287?302.
- [21] Eskandarpour M. (2015) Green Logistics: Integrating Environmental Criteria. *Eur. J. Oper. Res.* **242**(1), 4?16.
- [22] Tavakkoli-Moghaddam R. (2015) Chemical Reaction Optimization Algorithm for Multi-Objective Optimization Problems. *Appl. Soft Comput.* **36**, 437?448.
- [23] Laporte G.A. (2014) Particle Swarm Optimization for Green Logistics. *J. Heuristics* **20**(6), 629?648.
- [24] Braysy O., Gendreau M. (2002) Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows. *Top* **10**(2), 211?237.
- [25] Ganesh K., Narendran T.T. (2008) Taste: A two-phase heuristic to solve a routing problem with simultaneous delivery and pick-up. *Int. J. Adv. Manuf. Technol.* **37**(11), 1221?-1231.
- [26] Vidal T., Battarra M., Subramanian A., Erdogan G. (2015) Hybrid metaheuristics for the clustered vehicle routing problem. *Comput. Oper. Res.* **58**, 87?-99.
- [27] Braysy O., Gendreau M. (2005) Tabu search for vehicle routing problems. *Comput. Oper. Res.* **32**(7), 1659?-1686.

- [28] Vidal T., Crainic T.G., Gendreau M. (2012) A hybrid genetic algorithm for vehicle routing problem. *Comput. Oper. Res.* **39**(10), 2534?-2549.
- [29] Schneider L., Ramos A., Martinez H. (2014) Green vehicle routing problem with time windows. *Eur. J. Oper. Res.* **237**(3), 1014?-1025.
- [30] Andelmin J., Bartolini E. (2017) An exact algorithm for the green vehicle routing problem. *Transp. Sci.* **51**(4), 1288?-1303.
- [31] Romero-Barrera A.J., Cruz-Piris L., Tejedor-Romero M., Gimenez-Guzman J.M., Marsa-Maestre I. (2024) A bi-level programming approach to design a sustainable supply chain network under uncertainty: a real-life case study. *Clean Technol. Environ. Policy*, 1?-31.
- [32] Fishani B.A., Mahmoodirad A., Niroomand S., Hajjaghaei-Keshteli M. (2024) A bi-level programming approach to design a sustainable supply chain network under uncertainty: a real-life case study. *Clean Technol. Environ. Policy*, 1?-31.
- [33] Prakash A., Akhtar A.K. (2025) A hybrid environmental multi-objective optimization algorithm for eco-friendly vehicle routing in smart cities. *Land Archit.* **4**(1), 1.
- [34] Beresnev V.L. (2009) Upper bounds for objective functions of discrete competitive facility location problems. *J. Appl. Ind. Math.* **3**(4), 419-?432.
- [35] Bard J.F., Moore J.T. (1992) An algorithm for the discrete bilevel programming problem. *Naval Res. Logist. (NRL)* **39**(3), 419-?435.
- [36] Lam A.Y.S., Xu J., Li V.O.K. (2010) Chemical reaction optimization for population transition in peer-to-peer live streaming. *Proc. IEEE Congr. Evol. Comput.*, 1?8.
- [37] Lam A.Y.S., Li V.O.K. (2012) Chemical reaction optimization: A tutorial. *Memetic Comput.* **4**(1), 3?17.
- [38] Abbassi M., Chaabani A., Ben Said L. (2022) An efficient chemical reaction algorithm for multi-objective combinatorial bi-level optimization. *Eng. Optim.* **54**(4), 665?686.
- [39] Rebai S., Kessentini M., Wang H., Maxim B. (2020) Web service design defects detection: A bi-level multi-objective approach. *Inf. Softw. Technol.* **121**, 106255.