RAIRO-Oper. Res. 55 (2021) 1371–1399
https://doi.org/10.1051/ro/2021050

RAIRO Operations Research
www.rairo-ro.org

CAPACITATED LOCATION ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY UNDER THE RISK OF DISRUPTION

Milad Dehghan[1], Seyed Reza Hejazi[1], Maryam Karimi-Mamaghan[2], Mehrdad Mohammadi[2,*] and Amir Pirayesh[3]

Abstract. This paper develops a new mathematical model to study a location-routing problem with simultaneous pickup and delivery under the risk of disruption. A remarkable number of previous studies have assumed that network components (*e.g.*, routes, production factories, depots, etc.) are always available and can permanently serve the customers. This assumption is no longer valid when the network faces disruptions such as flood, earthquake, tsunami, terrorist attacks and workers strike. In case of any disruption in the network, tremendous cost is imposed on the stockholders. Incorporating disruption in the design phase of the network will alleviate the impact of these disasters and let the network resist disruption. In this study, a mixed integer programming (MIP) model is proposed that formulates a reliable capacitated location-routing problem with simultaneous pickup and delivery (RCLRP-SPD) services in supply chain distribution network. The objective function attempts to minimize the sum of location cost of depots, routing cost of vehicles and cost of unfulfilled demand of customers. Since the model is NP-Hard, three meta-heuristics are tailored for large-sized instances and the results show the outperformance of hybrid algorithms comparing to classic genetic algorithm. Finally, the obtained results are discussed and the paper is concluded.

Mathematics Subject Classification. 90C59.

Received April 25, 2019. Accepted March 30, 2021.

1. Introduction

Supply chain management (SCM) consists of efficiently planning, implementing and controlling the supply chain operations. Activities in SCM include transferring and storing raw materials, work-in-process inventory and final products from origin to customers [33]. Accordingly, design of a distribution network is a fundamental step in building an efficient supply chain. In designing a distribution network, decisions vary from the number of echelons in the network to the optimal location(s) of the facilities. These decisions are often classified into strategic, tactical, and operational decisions. A strategic decision, which is a long-term one, is not made on a regular basis and requires major investment in capital. Decisions such as the location and construction decisions fall into this category. A tactical or mid-term decision is made more frequently than a strategic decision. For

Keywords. Reliable capacitated location-routing problem, simultaneous pickup and delivery, disruptions, hybrid algorithms.

[1] Department of Industrial & System Engineering, Isfahan University of Technology, Isfahan, Iran.

[2] IMT Atlantique, Lab-STICC, UBL, F-29238 Brest, France.

[3] Centre of Excellence in Supply Chain and Transportation (CESIT), KEDGE Business School, Bordeaux, France.

[*]Corresponding author: mehrdad.mohammadi@imt-atlantique.fr

example, decisions in a vehicle routing problem (VRP) are tactical decisions [30]. Finally, those decisions that take place regularly are the operational or short-term decisions such as scheduling.

The location-routing problem (LRP) integrates the strategic location and tactical routing decisions [10]. The LRP is often defined as a special case of VRP, in which the optimal number and location of depots along with distribution routes must be determined simultaneously [8, 44]. Indeed, the LRP aims to locate the depots and route the vehicles to meet the demand of customers with a typical objective of minimizing the total cost of transportation (routing costs), fixed costs of locating depots and fixed costs of vehicles. A capacitated version of the LRP (CLRP) is provided by imposing capacity constraints on both depots and vehicles [14, 56].

Location and routing decisions are traditionally made in succession, where routing decisions follows location decisions. First, the location of the depots is determined *a priori* and then, the vehicles are routed through the customers and from the located depots. However, in real-world application, these decisions should be made simultaneously since ignoring routes when locating depots increases the cost of distribution and leads to sub-optimal solutions [51].

In most LRPs, one typical assumption is that depots are always available and can serve customers under any circumstances. However, this assumption is not valid in real settings when the depots are exposed to uncertainty risk [10]. The uncertainty in SCM can be generally categorized into operational uncertainty (risk) and disruption. Operational risks do not affect the functionality of supply chain's elements and only affect operational factors that are supposed to be fundamentally uncertain. They are typically inherent uncertainties in input parameters of a problem such as customer demand, purchasing prices for raw materials or required resources, production costs, transportation costs, lead times or transportation times. However, disruption risks can either completely or partly interrupt the functionality of supply chains' elements, typically for an uncertain amount of time [50]. Such disruptions are caused by nature or disasters such as earthquakes, hurricanes, terrorist attacks, economic and financial crises, labor strikes, or machine breakdowns [10,30]. Disruptions are mostly discrete and significant, whereas the uncertainties from operational sources are usually continuous and minor [1]. The occurrence of such major disruptions in the depots of a supply chain network results in an excessive transportation cost since the customers previously served by these failed depots have to be served by more distant depots [64]. Therefore, it is necessary to design a distribution system as reliable as possible to reduce the adverse effects of such disruptions and their resulting cost. In the context of supply chain design, according to [52], "a reliable system is one that can perform well even when parts of the system have failed". This ability is often called "reliability" [53].

Accordingly, this paper aims at developing a RCLRP-SPD, wherein the impact of disruption of depots is investigated in both the location of depots and the routing of the vehicles through the customers. Indeed, when disruption happens in the depots, the location of depots may change and this change would probably change the routing of the vehicles through the customers. Furthermore, the problem is modeled on simultaneous pickup and delivery, whereby the depot delivers the demand of the customers as well as picking up the products from the customers, such as unused or damaged products which need to be recovered or destroyed. The RCLRP-SPD assumes that customer demands are deterministic and when a depot is disrupted, it becomes totally unavailable to serve customers. When disruption occurs in a depot, the demand of its allocated customers is either serviced by other depots or lost. The goal is to determine the depot locations, outbound delivery routing and backup emergency depots under disruptions, in order to minimize the expected total cost (*i.e.*, the sum of fixed location costs, variable routing costs and demand loss penalties).

The RCLRP-SPD is NP-hard as it includes two NP-hard problems which are well-known as: the capacitated multi-depot location problem, the capacitated vehicle routing problem (CVRP) [14]. Therefore, in order to solve medium and large-size numerical instances effectively, it is necessary to develop meta-heuristic solution approaches. Different types of solution algorithms have been proposed for solving the CLRP in the literature, varying from local searches such as Variable Neighborhood Search (VNS) and Granular Tabu Search (GTS) [15] to global search algorithms such as genetic algorithms [29]. This paper proposes two hybrid solution algorithms that combines both local and global search mechanisms for having more efficient exploitation and exploration simultaneously.

The rest of the paper is organized as follows. Section 2 reviews the relevant papers in the literature. The problem is stated and formulated in Section 3. Next, Section 4 proposes a classic genetic and two hybrid solution approaches to solve the proposed MILP model. Afterward, the results are provided in Section 5. Finally, the paper is concluded in Section 6.

2. Literature review

This section reviews the latest and most relevant RCLRP and RCLRP-SPD papers. The literature review is presented in three sections: Section 2.1 reviews the location-routing problems, Section 2.2 reviews the reliable location-routing problems and finally, Section 2.3 reviews the solution approaches, particularly the heuristic and meta-heuristic algorithms employed in LRPs.

2.1. Location-routing problem (LRP)

Location-routing problems (LRPs) have been extensively studied by several authors in the literature and the interested readers are referred to the studies dealing with LRPs and their proposed classifications by Prodhon and Prins [49] and Drexl and Schneider [12].

Karaoglany *et al.* [23] studied LRP with simultaneous pickup and delivery, wherein the authors developed two mixed integer formulations (*i.e.*, a node-based and a flow-based formulation) incorporating a family of valid inequalities to strengthen the model. In order to solve the mathematical model, the authors developed a heuristic algorithm based on simulated annealing (SA) algorithm. Through the computational results, they showed that the flow-based formulation outperforms the node-based formulation in terms of the solution quality and running time for small-size instances (10–30 customers), while the node-based formulation performs better than the flow-based formulation for medium-size instances (50–100 customers). Zarandi *et al.* [61] examined the LRP with time windows under operational uncertainty, where customers demand and travel times are assumed to be fuzzy variables. The authors designed a fuzzy chance-constrained programming model using credibility theory, and a simulation-embedded SA algorithm was then presented to solve the problem. For initializing the solutions of the proposed SA algorithm, they used a heuristic method based on fuzzy *c*-means clustering with Mahalanobis distance and sweeping method.

Huang [19] presented a three-stage solution approach to deal with the multi-compartment capacitated LRP with pickup-delivery and stochastic demands. This three-stage solution approach solves the model to determine the depot locations, assign customers to the located depots and route the vehicles through customers subsequently. The objective is to minimize depot opening cost, vehicle cost, and travel costs and violation of the vehicle and depot capacity constraints. Pichka *et al.* [47] addressed the two-echelon open LRP (2E-OLRP). This problem is a new variant of the classical LRP that considers location and routing decisions in two-echelon supply chains in which third-party logistics providers are used. In this research, three new mathematical models and a hybrid SA algorithm are developed to solve the 2E-OLRP.

Zhang *et al.* [65] studied the multi-depot emergency facilities LRP with uncertain demand. To incorporate the uncertain characteristics of the emergency response system, they have developed an uncertain multi-objective programming model. Due to the computational complexity of the model, a hybrid intelligent algorithm is designed to solve the proposed model by combining uncertain simulation and a genetic algorithm. Moshref-Javadi and Lee [41] introduced the latency LRP (LLRP) aiming at the customers' waiting time minimization by optimally locating depots and routing of vehicles. To solve the model, the authors presented a memetic algorithm and a recursive granular algorithm.

Khalili-Damghani *et al.* [27] studied a bi-objective LRP to distribute perishable products in a supply chain. Considering the perishability of products, minimizing transportation costs is not the main objective in perishable supply chain planning. Therefore, the objectives are to minimize the total cost of the system and to balance the transportation cost of perishable products at distribution centers while the due-date of perishable products should be met. Timely delivery is an essential factor in the distribution process of perishable products. So, the authors considered a set of constraints to deliver the products no later than a predetermined time. Since the proposed model is NP-hard as well as bi-objective, they developed two solution procedures, *i.e.*, an exact method called epsilon-constraint and an evolutionary computation, called non-dominated sorting genetic algorithm (NSGA-II) to find out the near Pareto optimal solutions. Ghaffari-Nasab *et al.* [17] addressed a bi-objective CLRP with probabilistic travel times, where the aim is to minimize the total cost and the maximum delivery time to the customers. The authors presented mathematical programming formulations to model the problem using two stochastic programming approaches. The deterministic equivalents of the two stochastic models are extracted and solved by a variable neighborhood descent (VND).

2.2. Reliable location-routing problem (RLRP)

In deterministic LRP we make decisions about depots locations, customer allocation and vehicle routing to balance the trade-off between fixed set-up costs and variable transportation costs. However, considering disruption caused by natural disasters, terrorist attacks, or labor strikes, some of the located depots may become unavailable to serve customers. When disruption occurs in a depot, its customers may have to be re-allocated from their original depot to other depots with higher transportation costs [9]. Ahmadi-Javid and Seddighi [1] studied the LRP in a supply chain network composed of a set of producer-distributors that produced and distributed a single commodity to a number of customers. In this work, the production capacity of each producer (distributor) varies randomly because of numerous sources of disruptions. In addition, the vehicles utilized the distribution system are also disrupted randomly.

Xie *et al.* [55] studied the RLRP wherein all located depots are independently disrupted with a same probability. One of the assumptions made in this study is that customers receive delivery service in fixed groups. In other words, a vehicle is assigned to visit the customers in each group. Under this assumption, if one depot fails, then all of its customers will be reassigned to another backup depot within the same group, which is rather restrictive. Moreover, the vehicle delivery distance is approximated by the sum of the local travel distances within the group and the line-haul distance between this group and the affected depot. Zhang *et al.* [64] examined an LRP that includes a set of warehouses which are randomly disrupted. A scenario dependent MILP model is proposed to optimize the location of depots, delivery routing towards faraway areas and alternative planning. The authors proposed a meta-heuristic algorithm exploiting maximum-likelihood sampling, route-reallocation improvement, two-stage neighborhood search and SA algorithm.

Mohammadi *et al.* [38] developed a MINLP location-allocation model in a supply chain management problem to design reliable logistics networks that perform as well as possible under normal condition, and also perform relatively well when disruptions occur. They proposed a single objective scenario based robust optimization problem to minimize the nominal cost. In addition, to solve the proposed model, a new self-adaptive meta-heuristic algorithm based on genetic and imperialist competitive algorithms is developed. Li *et al.* [28] proposed two related models for the design of distribution networks exposed to the risk of disruption (*i.e.*, reliable p-median problem and reliable uncapacitated fixed-charge location problem). Both models accounted for heterogeneous facility failure probabilities and one layer of supplier backup. Both models are formulated as nonlinear integer programming models which are proven to be NP-hard. To solve the proposed model, the authors developed an algorithm based on Lagrangian relaxation. Wang *et al.* [54] examined a facility location problem under random facility disruption and presented a MINLP model for designing a reliable supply chain network that is exposed to the risk of disruption. They solved the proposed model using a Lagrangian relaxation-based algorithm.

2.3. Heuristic/meta-heuristic approaches

Among different evolutionary algorithms, genetic algorithms (GAs) are probably the most widely used class of evolutionary algorithms and have been applied as well to the VRPs and LRPs. A comprehensive survey of different types of GAs for these problems can be found in [20].

To solve CLRP, Derbel *et al.* [11] developed a GA-based hybrid algorithm and an iterated local search (ILS). Since GA could fail to converge to the global optimum and ILS could fall to the local optimum too quickly, they embedded ILS into GA to refine the search through successive iterations and maximize the chance of convergence to the optimal solution. Karaoglan and Altiparmak [21] proposed a memetic algorithm for solving a CLRP with Mixed Backhauls. They conducted an experimental study and compared the results with the branch-and-cut algorithm's lower bounds. Results showed that the memetic algorithm was able to find optimal or very good quality solutions within a reasonable computation time.

Escobar *et al.* [14] presented a two-phase hybrid heuristic (2-Phase HGTS), wherein the first phase constructs a solution (*i.e.*, Construction phase), and the second phase (*i.e.*, Improvement phase) employs a modified Granular Tabu Search (GTS) approach with different diversification strategies to modify the constructed solution. Furthermore, a random perturbation procedure is considered to prevent the algorithm from remaining in a local optimum for a given number of iterations. Ardjmand *et al.* [3] developed a new model for the location-routing problem. They proposed a new GA that solves the model within a reasonable computational time. Yu and Lin [56] proposed an SA heuristic with a special solution encoding scheme for solving the LRP with simultaneous pickup and delivery (LRPSPD). The solution encoding scheme can broaden the search space and facilitate the search for a better solution. The results showed the efficiency of the proposed SA to solve LRPSPD and its superiority over the existing exact approaches in terms of the quality of the solution. Yu *et al.* [57] presented an SA algorithm in which each solution is represented as a giant tour. Move, swap and 2-opt were used as neighborhoods and they operated on the giant tour. Therefore, a neighborhood move may change the position of the customer, open or close depots or reassign customers to the locations. Ferreira and de Queiroz [16] hybridized two heuristic algorithms based on the SA algorithm combined with a diversification procedure to solve a CLRP.

2.4. This paper's contributions

Table 1 summarizes the reviewed articles with their characteristics in two main aspects as problem characteristics and solution approach. In Table 1, SPD, LSbMH and PbMH stand for Simultaneous Pickup and Delivery, Local Search based Meta-Heuristics (*i.e.*, single solution based meta-heuristics) and Population based Meta-Heuristics, respectively. This Table 1 helps to well position our work comparing to the literature.

Based on Table 1, the main contributions of this paper that distinguish it from the existing studies in the literature are:

- Studying a reliable capacitated location-routing problem with simultaneous pickup and delivery (RCLRP-SPD) wherein depots are stochastically disrupted and become unavailable to serve the customer demands.
- Proposing a new mixed-integer linear programming (MILP) model for the proposed RCLRP-SPD.
- Developing an efficient scenario-based approach to cope with the stochastic disruption of depots.
- Proposing three tailored meta-heuristic algorithms (*i.e.*, two hybrid algorithms and a classical genetic algorithm) for solving the proposed MILP.

3. Problem statement and formulation

This section proposes a MILP model to formulate a RCLRP-SPD. Section 3.1 states the problem with its underlying assumptions. Afterward, Section 3.2 proposes the MILP model for the problem.

3.1. Problem statement

The goal of the proposed RCLRP-SPD is to choose and locate a set of depots (facilities) and to build vehicles' routes to meet customers delivery and pickup demands, such that the total expected cost of location, routing and disruption is minimized. In this problem, each potential depot could be stochastically disrupted, and each customer has certain demand. In Figures 1 and 2, two solutions of the CLRP and RCLRP are illustrated.

In Figure 1, three depots (*i.e.*, depots number 1, 4 and 5) are located (opened) and the customers are allocated to the located depots and served in a particular order. Furthermore, if each of the depots fails, its allocated customers will not be served anymore and this would impose high costs on the system. On the other hand, Figure 2 illustrates the reliable version of the CLRP, wherein the located depot number 4 is disrupted and some

Table 1. Position of this paper comparing to the literature.

Reference	Problem characteristics					Solution approach		
	Location	Routing	Depot Capacity	SPD	Disruption	LSbMH	PbMH	Hybrid
Yu *et al.* [57]	*	*	*			*		
Karaoglan *et al.* [23]	*	*	*	*		*		
Derbel *et al.* [11]	*	*	*			*	*	*
Escobar *et al.* [14]	*	*	*			*		*
Ahmadi-Javid and Seddighi [1]	*	*	*		*	*		
Zarandi *et al.* [61]	*	*				*		
Ghaffari-Nasab *et al.* [17]	*	*	*			*		
Escobar *et al.* [15]	*	*	*			*		*
Nadizadeh and Nasab [43]	*	*	*			*		*
Zhang *et al.* [64]	*	*	*		*	*		
Huang [19]	*	*	*	*		*		
Khalili-Damghani *et al.* [27]	*	*					*	
Karaoglan and Altiparmak [21]	*	*	*			*	*	
Marinakis [31]	*	*	*			*	*	
Yu and Lin [56]	*	*	*	*		*		
Lopes *et al.* [29]	*	*	*			*	*	*
Moshref-Javadi and Lee [41]	*	*				*	*	*
Xie *et al.* [55]	*	*			*			
Mohammadi *et al.* [38]	*				*		*	*
Majidi *et al.* [30]		*		*				
Dehghani *et al.* [10]	*		*		*			
Pichka *et al.* [47]	*	*	*			*		*
Zhang *et al.* [65]	*	*	*				*	*
Ferreira and de Queiroz [16]	*	*	*			*		
Yu *et al.* [58]	*	*	*			*	*	*
Pekel and Soner Kara [46]	*	*	*			*		*
Oudouar *et al.* [45]	*	*	*			*		*
This Study	*	*	*	*	*	*	*	*

of its customers are served by depot number 1 but with a higher transportation cost and finally some of them are served *via* the emergency depot.

The main assumptions of the proposed RCLRP-SPD are as follows:

- Only one depot and one vehicle can be allocated to each customer.
- Route of each vehicle starts and ends at the same depot.
- A single product is delivered and picked up in this problem.
- All vehicles have a same capacity (homogeneous fleety), but depots have different capacities.

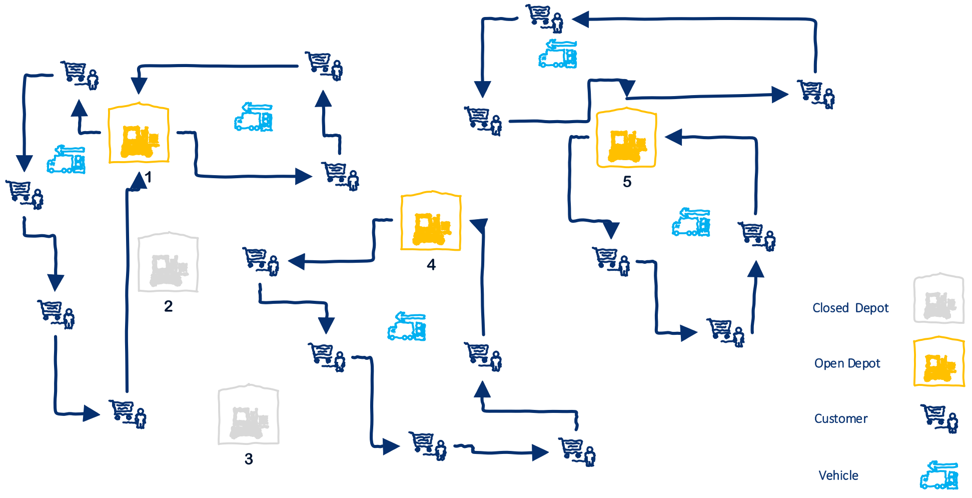


Figure 1. The scheme of a CLRP.

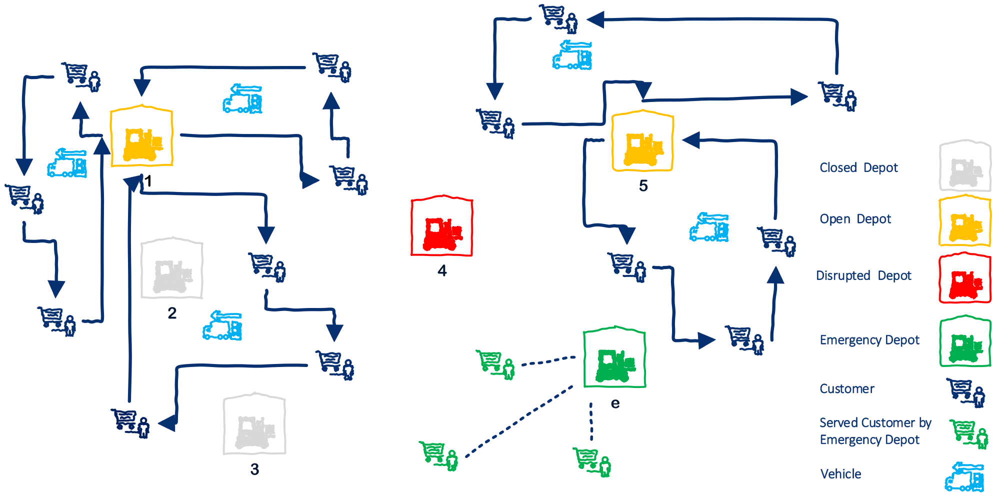


Figure 2. The scheme of a RCLRP.

- Customers have known pickup and delivery demands and predefined locations.
- Potential depots have known capacities and locations.
- There is a fixed cost for vehicles and depots.
- Several vehicles can be assigned to each located depot.
- Disruptions are assumed to be independent and depots can simultaneously be disrupted.
- Simultaneous pickup and delivery are assumed for each customer.
- Disruption probability of each depot is known.
- The penalty cost of the unmet demand is known.

In real-world situations, it is not possible to predict accurately the exact value of future parameters. Therefore, considering uncertainty makes the decision makers more capable to have better planning for the future.

The first step in dealing with uncertainty in modeling and optimizing supply chain related problems is to determine how to display uncertainty. Three well-known and widely-used methods for this purpose are [5]:

(I) *Distribution-based approach*, wherein the normal distribution is typically used with a specific mean and standard deviation to model uncertain parameters [2,31,34].
(II) *Fuzzy-based approach*, wherein the parameters are treated as fuzzy numbers with membership functions [36,42,43,60,62].
(III) *Scenario-based approach*, wherein some discrete scenarios with relevant levels of probability are used to describe the expected occurrence of specific results [38,64].

In this paper, disruption of depots is modeled by a set of discrete scenarios. Each scenario specifies a subset of depots that become non-operational or disrupted with a specific probability of occurrence. In case of a disruption, the affected customers whose allocated depots are disrupted, may be served by other located depots or emergency depots. Obviously, allowing more surviving (backup) depots to cover customers that were once serviced by a disrupted depot will increase the flexibility of the distribution system. Accordingly, Section 3.2 formulates a scenario based RCLRP-SPD.

3.2. Problem formulation

This section presents a scenario-based mathematical formulation for the studied problem. This formulation is based on the model proposed by Zhang *et al.* [64] for the RLRP but extends it to account for the multiple vehicles. Unlike [64], which has only considered delivery decisions, the presented formulation tackles with both pickup and delivery decisions. Our model aims to minimize the total expected costs of all scenarios including fixed costs, expected travel costs and expected penalty costs. Let $G = (V, A)$ be an undirected network, where V is the set of nodes, composed of a subset of geographically dispersed customers denoted by I and a subset of potential depots denoted by J. $A = \{(i, j)|i, j \in V, i \neq j\}$ is the set of arcs that connect each pair of nodes in V. Let S be the set of scenarios, each specifying a set of disrupted depots. Before proposing the MILP model, necessary notations are explained as follows.

Sets

I Set of customers.
J Set of potential depots.
S Set of scenarios.
H Set of vehicles.
e Emergency depot.

Parameters

fd_j Fixed cost of depot j.
cd_j Capacity of depot j.
q_j Failure probability of depot j.
d_i Delivery goods demand of customer i.
p_i Pickup goods demand of customer i.
θ_i Penalty of serving customer i using the emergency depot.
cv Capacity of vehicles.
fv Fixed cost of each vehicle.
tc_{ij} Traveling cost from node i to node j.
a_{js} If depot j in scenario s is not disrupted 1, otherwise 0. Disrupted depots are known *a priori*; therefore, a_{js} is a binary parameter.
pr_s Probability that scenario s occurs.
M A big number.

Decision variables

Y_j If depot j is open 1, otherwise 0.
Z_{ijs} If customer i is served by depot j in scenario s 1, otherwise 0.
X_{ikhjs} If node k is visited immediately after node i on a route originating from depot j by vehicle h in scenario s 1, otherwise 0.
T_{ijs} Delivery goods transported between node i and node j in scenario s.
W_{ijs} Pickup goods transported between node i and node j in scenario s.
U_{ihj} An auxiliary integer variable to eliminate the subtours.

Emergency depot e, has no fixed cost, a failure probability of 0 and infinite capacity. If a customer i is not served in a scenario, we assign it to the emergency depot e, which induces an unmet demand penalty θ_i.

In certain scenarios, if a depot survives, then it is “*normal*” (non-disrupted); if it is both normal and open, we call it “*available*”. Considering this point, customers can only obtain service from available depots in each scenario. Consider $J_{ns} = \{j | j \in J, a_{js} = 1\}$ as the set of normal depots in scenario s. In the strategic decision, assume that the set of depots selected to be opened is denoted by J_o. Then, the set of available depots in scenario s is $J_{as} = J_{ns} \cap J_o$. Since each scenario specifies a subset of depots which become non-operational or disrupted ($a_{js}, \forall j \in J, \forall s \in S$), the probability that scenario s occurs is represented using equation (3.1).

$$pr_s = \prod_{j \in J : a_{js} = 0} q_j . \prod_{j \in J : a_{js} = 1} (1 - q_j) . \tag{3.1}$$

A total of $2^{|J|}$ possible scenarios exist, which leads to an exponential growth in problem size and huge computational burden to solve the problem. By increasing the number of depots by one, we double the number of scenarios.

Based on the provided notations, the proposed MILP model for the RCLRP-SPD is as follow:

$$\min \mathrm{OFV} = \sum_{j \in J} fd_j Y_j + \sum_{s \in S} pr_s \left(\sum_{i \in I \cup J_{ns}} \sum_{k \in I \cup J_{ns}} \sum_{h \in H} \sum_{j \in J_{ns}} tc_{ik} X_{ikhjs} + \sum_{i \in I} Z_{ies} \theta_i + \sum_{j \in J_{ns}} \sum_{i \in I} \sum_{h \in H} X_{jihjs} fv \right) \tag{3.2}$$

subject to:

$$Z_{ijs} \leq Y_j \qquad \forall i \in I, j \in J_{ns}, s \in S \tag{3.3}$$

$$\sum_{j \in J_{ns} \cup e} Z_{ijs} = 1 \qquad \forall i \in I, s \in S \tag{3.4}$$

$$\sum_{i \in I} d_i Z_{ijs} \leq cd_j Y_j \qquad \forall j \in J_{ns}, s \in S \tag{3.5}$$

$$\sum_{i \in I} p_i Z_{ijs} \leq cd_j Y_j \qquad \forall j \in J_{ns}, s \in S \tag{3.6}$$

$$\sum_{\substack{k \in I \cup \{j\} \\ k \neq i}} \sum_{h \in H} X_{kihjs} = Z_{ijs} \qquad \forall i \in I, j \in J_{ns}, s \in S \tag{3.7}$$

$$\sum_{\substack{k \in I \cup J_{ns} \\ i \neq k}} \sum_{h \in H} \sum_{j \in J_{ns}} X_{kihjs} \leq 1 \qquad \forall i \in I, s \in S \tag{3.8}$$

$$\sum_{\substack{k \in I \cup J_{ns} \\ k \neq i}} \sum_{h \in H} \sum_{j \in J_{ns}} X_{kihjs} = \sum_{\substack{k \in I \cup J_{ns} \\ k \neq i}} \sum_{h \in H} \sum_{j \in J_{ns}} X_{ikhjs} \qquad \forall i \in I \cup J_{ns}, s \in S \tag{3.9}$$

$$\sum_{\substack{k \in I \\ k \neq i}} X_{kihjs} + \sum_{\substack{k \in J_{ns} \\ k = j}} X_{kihjs} = \sum_{\substack{k \in I \\ k \neq i}} X_{ikhjs} + \sum_{\substack{k \in J_{ns} \\ k = j}} X_{ikhjs} \qquad \forall i \in I, j \in J_{ns}, h \in H, s \in S \tag{3.10}$$

$$\sum_{\substack{i \in J_{ns} \\ i = j}} \sum_{k \in I} \sum_{j \in J_{ns}} X_{ikhjs} \leq 1 \qquad \forall h \in H, s \in S \tag{3.11}$$

$$\sum_{i \in I \cup J_{ns}} T_{ijs} - \sum_{i \in I \cup J_{ns}} T_{jis} = d_j \left(\sum_{i \in I \cup J_{ns}} \sum_{h \in H} \sum_{m \in J_{ns}} X_{ijhms} \right) \qquad \forall j \in I, s \in S \tag{3.12}$$

$$\sum_{i \in I \cup J_{ns}} W_{jis} - \sum_{i \in I \cup J_{ns}} W_{ijs} = p_j \left(\sum_{i \in I \cup J_{ns}} \sum_{h \in H} \sum_{m \in J_{ns}} X_{ijhms} \right) \qquad \forall j \in I, s \in S \tag{3.13}$$

$$W_{ijs} + T_{ijs} \leq cv \left(\sum_{h \in H} \sum_{m \in J_{ns}} X_{ijhms} \right) \qquad \forall i \in I \cup J_{ns}, j \in I \cup J_{ns}, s \in S \tag{3.14}$$

$$\sum_{i \in I} T_{ijs} = 0 \qquad \forall j \in J_{ns}, s \in S \tag{3.15}$$

$$\sum_{i \in I} W_{jis} = 0 \qquad \forall j \in J_{ns}, s \in S \tag{3.16}$$

$$U_{ihj} + 1 \leq U_{khj} + M\left(1 - X_{ikhjs}\right) \qquad \forall i, k \in I; i \neq k, j \in J_{ns}, h \in H, s \in S \tag{3.17}$$

$$Y_j \in \{0, 1\} \qquad \forall j \in J_{ns} \tag{3.18}$$

$$Z_{ijs} \in \{0, 1\} \qquad \forall i \in I, j \in J_{ns} \cup e, s \in S \tag{3.19}$$

$$X_{ikhjs} \in \{0, 1\} \qquad \forall i \in I \cup J_{ns}, k \in I \cup J_{ns}, h \in H, j \in J_{ns}, s \in S \tag{3.20}$$

$$T_{ijs}, W_{ijs} \geq 0 \qquad \forall i \in I \cup J_{ns}, j \in I \cup J_{ns}, s \in S \tag{3.21}$$

$$U_{ihj} \in \text{integer} \qquad \forall i \in I, j \in J_{ns}, h \in H. \tag{3.22}$$

The objective function (3.2) minimizes sum of the depots fixed costs (scenario-independent) and the expected costs of each scenario including routing costs, penalty costs and vehicles fixed costs (scenario-dependent) respectively. Constraint (3.3) ensures that a customer in any scenario could be assigned to a depot if the depot is opened. Constraint (3.4) ensures that each customer is assigned to exactly one depot in each scenario. Constraints (3.5) and (3.6) are depots capacity constraints in each scenario. Constraint (3.7) links the allocation and routing in each scenario. Constraint (3.8) ensures that, in each scenario, each customer is visited only once. Constraint (3.9) is the flow conservation constraint which forces that, in each scenario, the input flow to each customer/depot is equal to the output flow from that customer/depot. Constraint (3.10) checks the feasibility of the route in each scenario when starting the routes from the depots. Constraint (3.11) ensures that each vehicle is assigned at most to a single depot in each scenario. Constraints (3.12)–(3.14) control and impose the vehicles capacities in each scenario. Constraint (3.15) ensures that the amount of delivery demand when vehicles return to depots is zero. Constraint (3.16) ensures the amount of pickup demand when vehicles start routes from depots is zero. Constraint (3.17) is the sub-tour elimination constraint in each scenario. Constraints (3.18)–(3.22) are the integrality and non-negativity constraints.

4. Solution approaches

Metaheuristic algorithms have been widely used in the literature to solve optimization problems [13,24,40]. In this regard, the papers in the literature use either population-based evolutionary algorithms or single solution-based local search algorithms. The former is powerful in diversification and the latter is powerful in intensification. In this paper, we attempt to propose a hybrid algorithm which is powerful in both diversification and intensification [13,24]. Indeed, hybrid algorithms attempts to benefit from the strengths of different algorithm in a single algorithm [25,26,37,39,59,63]. Accordingly, we proposed a hybrid algorithm by the combination of genetic algorithm (GA) (as a powerful algorithm in diversification) and two local search algorithms (as powerful algorithms in intensification).

Genetic algorithm (GA), which was first developed by Holland [18], is one of the most well-known evolutionary optimization algorithms. GA evolves a population of randomly generated solutions with size P_{size} using crossover and mutation operators. At each iteration, a number of solutions are generated through crossover operator with rate C_R and a set of other solution are generated through mutation operator with rate M_R. In crossover, a pair of selected parents (using tournament selection mechanism) are crossed with the hope of creating better children. Despite crossover, mutation operator is employed on an individual solution in order to diversify the solution space and escape from the local optimum. These operators are employed to generate new solutions as a next offspring so that the current population and generated offspring are combined together for creating next generation. In this paper, we do not consider the elitism operator. GA can be used in various types of problems with appropriate genetic representation, fitness function, and evolution operators.

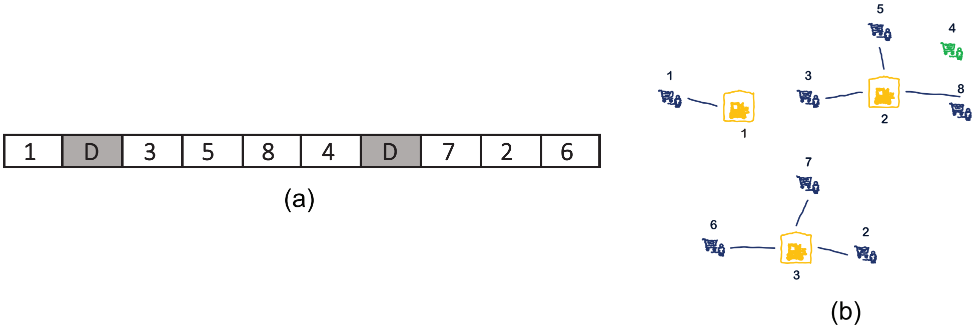
For solving the proposed RCLRP-SPD, we propose two efficient hybrid GA by hybridizing the GA with variable neighborhood descent (VND) (HGAVND) and the GA with a local search (HGALS) [4]. In addition, the proposed model aims at locating depots which are less sensitive to disruption and lead to the minimum cost no matter which disruption scenario occurs. Indeed, the final solution would be the most reliable solution when exposed to any disruption scenario. Therefore, the located depot(s) is (are) the same for all scenarios. Since in the RCLRP-SPD the open depots are the same for all scenarios, a two-stage mechanism is developed in the algorithms. The first and the second stages are called “Location-Allocation” stage and “Vehicle Routing” stage respectively. In the Location-Allocation stage, the depots are located and the customers are allocated to the opened depots. Once the first stage is solved, its results are fed to the second stage and the routing decisions are made in the second stage. Indeed, the routing is solved for each scenario independently. Besides to the proposed HGAVND and HGALS algorithms, we also employ classical GA to better show how these hybridizations improve the performance of the classical GA. Therefore, we consider three combinations of algorithms to solve the two “Location-Allocation” and “Vehicle Routing” stages as GA→GA, GA→HGAVND, and GA→HGALS. For instance, the combination GA→GA means that the GA is used for solving both stages; and the combination GA→HGAVND states that stage 1 is solved using GA and stage 2 is solved using HGAVND. As it can be seen, for solving the “Location-Allocation” stage (*i.e.*, stage 1), we only use classical GA but the “Vehicle Routing” stage (*i.e.*, stage 2) is solved using all three algorithms (*i.e.*, GA, HGAVND, and HGALS).

4.1. Solution representation

Any evolutionary algorithm’s performance remarkably depends on its representation of the solution. Since GA is a population-based algorithm, the solution representation should not consume much memory. The solution representation contains two different parts and each part corresponds to a stage of the proposed algorithms. These parts are described in the following.

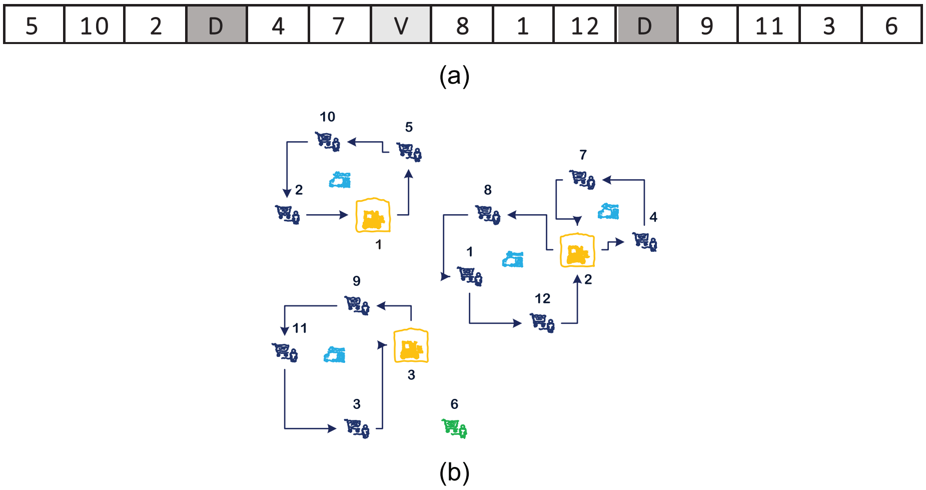
(i) *Location-Allocation*: Figure 3a shows the location-allocation decisions in a network with 8 customers. Numbers 1–8 represent the ycustomers, and each “D” shows a depot. The number of “D”s plus 1 demonstrates the number of located depots. In Figure 3, there are totally three located depots. Starting from the left, the customers are allocated to the first located depot until they will not exceed the capacity of the corresponding depot. For instance, customer number 1 has been allocated to the first depot. Then, customers number 3, 5, 8 and 4 should be allocated to the second located depot but only customers 3, 5 and 8 can be allocated since adding customer 4 will exceed the capacity of the second depot. Therefore, the demand of customer number 4 is unmet. This procedure continues until the most possible customers are allocated to the located depots. Figure 3b illustrates the scheme of the location-allocation decisions.

(ii) *Vehicle Routing*: Figure 4a shows the solution representation regarding the vehicle assignments and routing. Numbers 1–12 represent the customers. In this figure, each “V” signifies that the route is terminated and another route from the same depot is started, even though the accumulated demand has not exceeded vehicle capacity, while each “D” terminates the assignments of routes to the current depot and starts the



1	D	3	5	8	4	D	7	2	6

FIGURE 3. Location-Allocation representation. (a) Location-Allocation matrix. (b) Scheme of Location-Allocation.



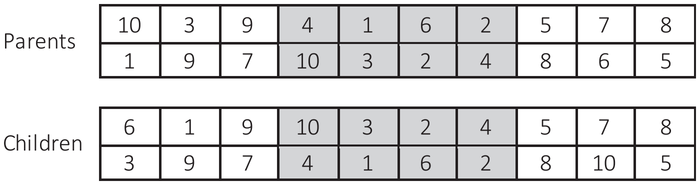
5	10	2	D	4	7	V	8	1	12	D	9	11	3	6

FIGURE 4. Vehicle Routing representation. (a) Vehicle Routing matrix. (b) Scheme of Vehicle Routing.

assignment to the next depot. Note that every "D" represents a new depot and a new vehicle. A vehicle has been assigned to the first and the third depots, but two vehicles have been assigned to the second located depot. The route of the third located depot serves customers 9, 11 and 3. Since adding customer 6 exceeds the vehicle capacity, the route is terminated. Note that the capacity of depots is not taken into consideration during the decoding process. Therefore, a per unit penalty cost P is added to the objective function value whenever the total demand served by a depot exceeds its capacity.

4.2. Initial population construction

Each stage of the proposed algorithms requires an initial population. The initial population of the first stage is generated in a greedy manner. We generate an initial population including greedy heuristic solution as well as a random solution. The Location-Allocation part of the solution representation is then constructed by a greedy method with the hope that a good initial solution can be found within a reasonable time. In this greedy algorithm, all potential depots are opened and each customer is allocated to the closest located depot according to an increasing order of $D_i + P_i$ until the depot capacity is not exceeded; if exceeded, the next closest depot is



Parents	10	3	9	4	1	6	2	5	7	8
	1	9	7	10	3	2	4	8	6	5
Children	6	1	9	10	3	2	4	5	7	8
	3	9	7	4	1	6	2	8	10	5

Figure 5. PMX crossover.

selected. This greedy solution is then improved through the algorithm by closing unnecessarily opened depots. The population of the second stage is generated randomly.

4.3. Fitness function

The fitness function is defined by the solution representation and measures the quality of each solution. The fitness of each stage is calculated as follow:

(i) *Location-Allocation*: in this stage, the fitness function includes depots fixed cost, direct distance cost between the depots and customers location, and penalty cost of unmet customers. Since we want to minimize the cost, lower value of the fitness function implies higher fitness.

(ii) *Vehicle Routing*: in the second stage, the fitness function includes vehicles fixed cost, vehicles routing cost, constant P if the depot capacity is exceeded and the penalty cost of unmet customers. Since we want to minimize the cost, lower value of the fitness function implies higher fitness.

4.4. Crossover operator

To obtain a near – optimum solution, genetic evolutionary operators are used to create a better solution and replace them with those that existed in the initial population. Crossover is the GA's main operator, sharing information between two randomly selected parents in the hope of better offspring. These offspring are compared with respect to the fitness function and passed onto the next generation. In this paper, partially mapped crossover (PMX) is employed in both parts of "*Location-Allocation*" and "*Vehicle Routing*". The PMX indexes in the middle part of the solution representation are exchanged and the missing stops replace their matching index in the second parent. Stop indexes are exchanged based on the mapping set, which is constructed by comparing the exchanging parts of two parents [6]. This popular crossover method is illustrated in Figure 5. Note that each "V" and "D" can be replaced by numbers.

4.5. Mutation operator

Mutation is a strategy of diversification that aims to explore new areas of solution space, avoid early convergence, and escape from local optima. Through the mutation operator, three "*Swap*", "2-*Opt-a*", and "2-*Opt-b*" moves in the both "*Location-Allocation*" and "*Vehicle Routing*" parts. Using the "*Swap*" operator, two genes from the chromosome are selected and swapped. Using the "2-*Opt-a*" move, two points in the chromosome are selected and the order of the in-between genes is reversed. Using the "2-*Opt-b*" move, two points in the chromosome are selected and the order of the in-between alleles is randomly changed. These mutation operators are illustrated in Figure 6.

4.6. Hybrid genetic algorithm

In this paper, we propose two hybrid GA by hybridizing the GA with VND (HGAVND) and the GA with a local search (HGALS). Both HGAVND and HGALS are employed to jointly solve routing in the "Vehicle Routing" part. Since the GA may fail to converge to a global optimum, we use VND and a local search to refine

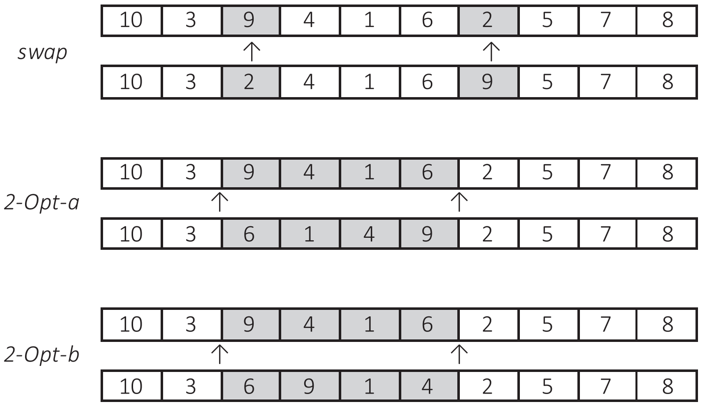


FIGURE 6. Examples of the "*swap*", "*2-Opt-a*" and "*2-Opt-b*" mutation operators.

the GA search through successive iterations and maximize the chance of convergence to an optimal solution [7,32,35,48].

The VND is a deterministic version of the variable neighborhood search (VNS) algorithm proposed by Mladenović and Hansen [35]. VND's basic idea is to explore a set of predefined neighborhood structures $(N(l)(l = 1, 2, \ldots, l_{\max}))$ successively in order to obtain a better solution. It explores a set of neighborhoods systematically in order to obtain various local optima and escape from local optima. The overview of the proposed hybrid GA-VND (HGAVND) algorithm is depicted in Figure 7. In the proposed HGAVND, we incorporate VND into the GA's general scheme. This allows us to take advantage of VND features in order to improve the population generated by the GA and thus to complement the genetic search. In Figure 7, *MaxIt_GA* is the maximum number of iterations of the GA (number of generations). In addition, *p_hybrid* is a factor to calculate the probability of doing VND in the proposed HGAVND.

It is noteworthy that implementing the VND at each iteration of the GA increases the computational time. Therefore, the VND is involved in the GA with a probability equal to $prob(i)$. This value increases by the number of iterations. Accordingly, we give more chance to the VND in the later iterations. Then, as we find the promising regions of the feasible solution space, we gradually increase the probability of employment of VND [48]. The pseudo code of the VND proposed for HGAVND algorithm is depicted in Figure 8, wherein Itr_H is the maximum iteration for the VND.

In the VND used to complement the GA in our hybrid approach, we use seven neighborhoods ($l_{\max} = 7$) in both hybridization algorithm, namely $N(1)$, $N(2)$, $N(3)$, $N(4)$, $N(5)$, $N(6)$ and $N(7)$, which are described in the following [14,21,32]:

- $N(1)$: randomly selects two customers assigned to two different routes and interchanges them.
- $N(2)$: randomly selects a customer and inserts it at a random position in a different route.
- $N(3)$: randomly selects two customers assigned to a route and interchanges them.
- $N(4)$: randomly selects two different routes and sequential customers are interchanged.
- $N(5)$: randomly selects two customers assigned to a route and customers of the in-between alleles are reversed.
- $N(6)$: three arcs which are on the same route are deleted. Then three new arcs are created.
- $N(7)$: randomly selects two sequential customers assigned to a route and adds them to the random position in another random route with same sequential.

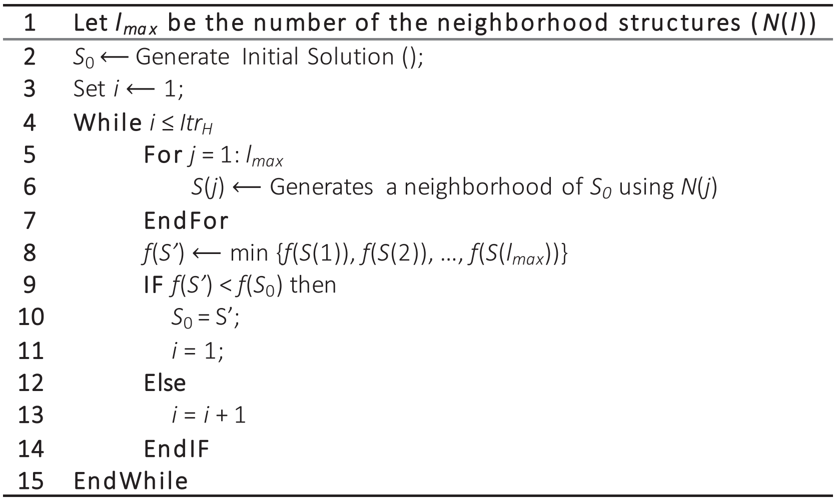
HGALS is similar to HGAVND, with difference in local search. In HGAVND we obtain local optima in every neighborhood structure, but in HGALS, all the neighborhood structures are applied together and we obtain local optima in all neighborhood structures. The pseudo code of the local search proposed for HGALS algorithm

```
Set parameters: $P_{size}$, $MaxIt\_GA$, $p\_hybrid$, $C_R$, $M_R$
Generate initial random population of size $P_{size}$
For $i$ = 1: MaxIt_GA
        For $j$ = 1: $C_R$*$P_{size}$
            $P_1$ & $P_2$: Select parents for crossover
            Cross $P_1$ & $P_2$ and generate two children $O_1$ & $O_2$
            Add $O_1$ & $O_2$ to the next generation
        EndFor
        For $k$ = 1: $M_R$*$P_{size}$
            $P_3$: Select a solution for mutation
            Mutate $P_3$
            Add created new solution to the next generation
        EndFor
        $\boldsymbol{prob(i)} = \frac{\mathbf{1}}{(\boldsymbol{p\_hybrid})^{\boldsymbol{MaxIt\_GA-i}}}$ ;
        IF $rnd() \le prob(i)$
            $P_4$: Get the current best solution for doing local search
            Do VND on $P_4$ and create new solution $P_5$                // Local Search by VND
            Add $P_5$ to the next generation
        EndIF
EndFor
```

Figure 7. HGAVND algorithm.

```
Let $l_{max}$ be the number of the neighborhood structures ($N(l)$)
$S_0$ ⟵ Generate/get Initial Solution ();
Set $l$ ⟵ 1;
While ($l \le l_{max}$) Do
Set $i$ ⟵ 1;
        While $i \le Itr_H$
                $S'$ ⟵ Generates a neighborhood of $S_0$ using $N(l)$
                IF $f(S') < f(S_0)$ then
                    $S_0 = S'$;
                    $i$ = 1;
                Else
                    $i = i + 1$;
                EndIF
        EndWhile
        $l$=$l$+1;
EndWhile
```

Figure 8. VND's pseudo code.



```
Let $l_{max}$ be the number of the neighborhood structures ($N(l)$)
$S_0 \longleftarrow$ Generate Initial Solution ();
Set $i \longleftarrow 1$;
While $i \leq Itr_H$
        For $j = 1: l_{max}$
            $S(j) \longleftarrow$ Generates a neighborhood of $S_0$ using $N(j)$
        EndFor
        $f(S') \longleftarrow \min \{f(S(1)), f(S(2)), \ldots, f(S(l_{max}))\}$
        IF $f(S') < f(S_0)$ then
            $S_0 = S'$;
            $i = 1$;
        Else
            $i = i + 1$
        EndIF
EndWhile
```

Figure 9. Local search's pseudo code.

is depicted in Figure 9, wherein Itr_H is the maximum number of iterations of the proposed local search in the HGALS.

5. Computational study

In this section, we present extensive computational results in order to assess the effectiveness of the proposed GA, HGAVND and HGALS algorithms for solving the RCLRP-SPD in two stages. The GA, HGAVND, and HGALS algorithms are coded using MATLAB R2014a and the mathematical model is solved with CPLEX 12.7.1.0 (with a time limit of 3 h) solver in GAMS 24.9.1 software on Intel®Core$^{\mathrm{TM}}$i7-720QM (1.6 GHz, 4 GB RAM).

5.1. Parameter setting

The efficiency and effectiveness of all meta-heuristic algorithms are significantly influenced by its parameters. In this study, the parameters optimization of the proposed GA, HGAVND and HGALS are performed based on Taguchi methods. The experimental design of Taguchi has been widely used for optimization problems. Taguchi design is based on two major tools: orthogonal array (OA) and signal-to-noise (S/N) ratio. The OA is a matrix of numbers containing experimental schemes based on different levels of factors. The S/N ratio is the measure of variation and guarantees the robustness of this kind of experimental design. The term "signal" represents "mean response variable" as the desired value and "noise" represents "standard deviation" as undesirable value [7]. Taguchi method is utilized *via* Minitab 18 software for design of experiments (DOE) and analyzing their results. The aim of the Taguchi method is to maximize the S/N ratio which is calculated by equation (5.1) for minimization problems for each parameter i on its related level j.

$$(\mathrm{S/N\ ratio})_{ij} = -10 \log_{10} \left(\frac{\sum Z_{ij}^2}{n} \right) \quad \forall i, j \tag{5.1}$$

where z_{ij} is the objective function value using parameter i on level j and n is the number of repetitions for the corresponding experiment.

Since the proposed algorithms are executed in two stages, the parameters of each stage are tuned separately as follows:

Table 2. Different levels of GA parameters in "*Location-Allocation*" parameters.

GA parameters "Location-Allocation"	Levels			
	1	2	3	4
Number of generations	50	75	**100**	200
P_{size}	**200**	250	275	300
C_R	0.6	0.7	**0.8**	0.9
M_R	0.2	0.3	0.4	**0.5**

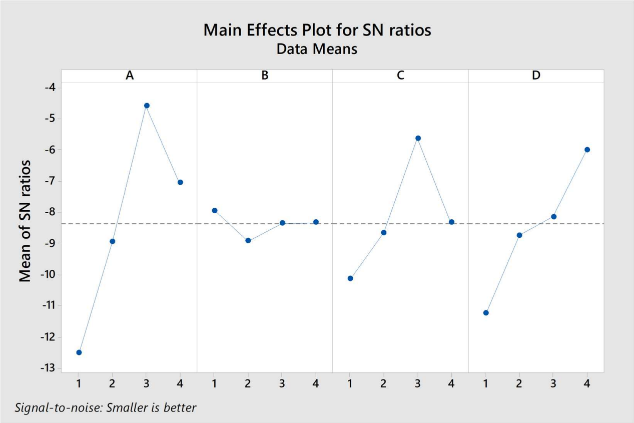


Figure 10. The S/N value index of hybrid algorithms for "*Location-Allocation*" parameters.

Table 3. Different levels of hybrid parameters in "*Vehicle Routing*" parameters.

GA parameters "Vehicle Routing"	Levels		
	1	2	3
Number of generations	100	**150**	200
P_{size}	150	**200**	250
C_R	0.7	0.8	**0.9**
M_R	0.2	**0.3**	0.4
Itr_H	50	75	**100**
p_hybrid	1.1	**1.15**	1.2

(i) *Location-Allocation*: since GA solves the first stage, different levels of the "*Location-Allocation*" parameters for tuning process is shown in Table 2. According to the S/N ratio plot shown in Figure 10, it is inferred that the number of generations, P_{size}, C_R and M_R are tuned at 100, 200, 0.8 and 0.5, respectively.

(ii) *Vehicle Routing*: since hybrid algorithms solve the second stage, different levels of the "*Vehicle Routing*" parameters for tuning process is shown in Table 3. According to the S/N ratio plot shown in Figure 11, it is inferred that the number of generation, population size, crossover rate, mutation rate, number of generation in hybridization algorithms and parameter that affects the probability of starting the hybridization algorithms, are tuned at 150, 200, 0.9, 0.4, 100 and 1.15, respectively.

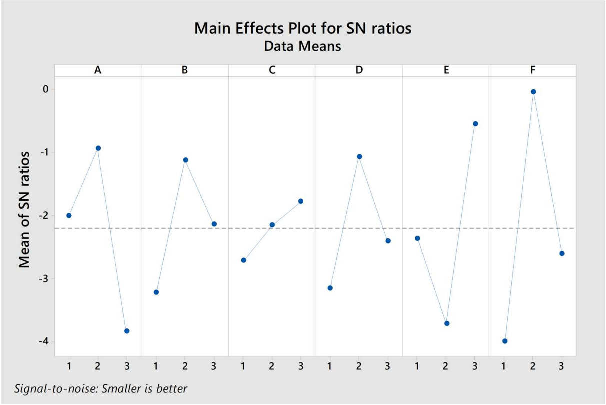


FIGURE 11. The S/N value index of hybrid algorithms for "*Vehicle Routing*" parameters.

5.2. Computational results

This section provides comprehensive numerical results to demonstrate the efficacy of the proposed MILP model and the solution approaches. Numerous instances are generated based on [64] and solved in this section. Regarding the failure probability, 5 levels are considered as $[0.01, 0.1]$, $[0.1, 0.2]$, $[0.2, 0.3]$, $[0.3, 0.4]$, $[0.8, 0.9]$, which are expressed as "a" to "e", respectively. These instances are labeled as $|I| - |J| - |H|$-a, where $|I|$ is the number of customers, $|J|$ is the number of potential depots, $|H|$ is the number of vehicles, and finally "a" denotes the level of failure probability. Each instance of the problem is executed 30 times and the results, including the best, the worst, the average and the standard deviation (σ) values are reported afterwards.

The potential location of customers and depots are randomly generated in a continuous space between $[1, 10]$. The cost between two nodes is calculated as the Euclidean distance between them. The delivery demand and pickup demand of each customer are randomly distributed in $[10, 100]$. The unmet-demand penalty ζ_i is chosen from $[100, 120]$. The fixed setup cost of each depot and vehicle are drawn uniformly from $[100, 500]$ and $[50, 100]$, respectively. We first calculate the $\overline{C}$ and $\overline{D}$ as:

$$\overline{C} = \frac{\sum_{i=1}^{I} d_i}{|J|} \tag{5.2}$$

$$\overline{D} = \frac{\sum_{i=1}^{I} d_i}{|H|}. \tag{5.3}$$

Each capacity cd_j andcv are then drawn uniformly from $[1.5\overline{C}, 2.5\overline{C}]$ and $[\overline{D}, 1.5\overline{D}]$, respectively. In medium and large instances, all parameters are generated like small instances, except cd_j, cv and θ_i that are drawn uniformly from $[1.5\overline{C}, 1.8\overline{C}]$, $2/3[\overline{D}, 1.5\overline{D}]$ and $[50, 100]$, respectively.

5.2.1. Analysis of small-size instances

Three sets of instances are solved with different failure probability from "a" to "e", where the failure probability increases from "a" to "e". Two number of potential depots are considered for small-size instances. Therefore, the number of all scenarios is then 2^2 and instances are solved in four scenarios. The results of CPLEX, GA, HGAVND, and HGALS have been provided in Table 4. Since for all 30 executions, the optimal solution is found, Table 4 contains only the best values (best, worst and average are the same and $\sigma = 0$). The results are compared in terms of objective function value, CPU time and the number of opened depots.

It can be seen that for all the small-size instances, all three GA, HGAVND, and HGALS reach the optimal solution; however, GA has lower CPU time since no extra local search operator is employed. In addition, the

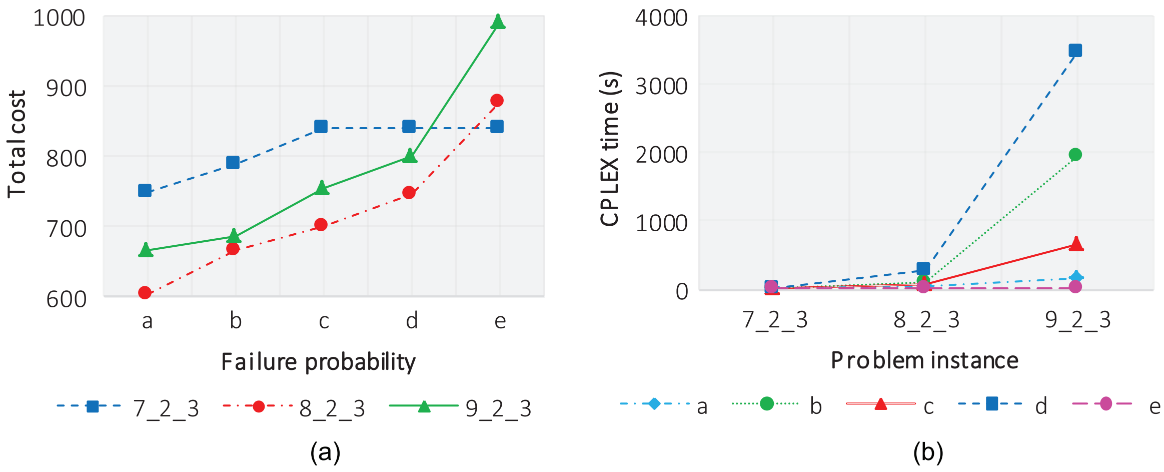


FIGURE 12. Small-scale instances analysis in CPLEX. (a) Total cost *vs.* Failure probability. (b) CPU time *vs.* Instance size.

proposed HGALS leads to smoothly lower CPU time comparing to HGAVND. In addition, in the solution of instances 7-2-3-c,d,e, 8-2-3-e and 9-2-3-e, no depot has been opened since the failure probability is high and the model prefer not to allocate customers to unreliable depots.

Two analyses are illustrated on the result of the CPLEX algorithm in Figure 12. Figure 12a illustrates that the total cost increases with increasing the failure probability, while there is no change in 7-2-3 ("c", "d" and "e") instances because in these instances the penalty cost is paid to all customers and no location and routing decisions are performed. Figure 12b connects the running times resulting from running the CPLEX on different sized instances, there is no growth in "e" instances because in these instances the penalty cost is paid to all customers and location and routing is not performed. Note that all the meta-heuristic algorithms have reached the optimal solution. In the majority of instances (11 out of 15), metaheuristic algorithms reach the optimal solution in significantly less CPU time comparing to CPLEX.

5.2.2. Analysis of medium instances

Four instances are solved with "a" failure probability because we mentioned in the introduction section that the occurrence probability of disruption is small. In addition, the same observation and results were obtained for medium and large instances comparing to small instances. Therefore, the results of medium and large instances only contain the failure probability "a" which is low and more close to reality. Furthermore, three potential depots are considered that leads to the total 2^3 number of different scenarios. Among them, we choose those scenarios that have the largest probability of occurrence. Here, instances are solved in four $(|J|+1)$ scenarios that have the largest probability of occurrence. The results of the best, the worst, the average and σ values are given in Table 5 and CPLEX time limit is 10 800 s. The column "DIFF (%)" shows the gap between the meta-heuristic cost and CPLEX cost and is calculated as DIFF $= \frac{\text{OFV}_{\text{meta-heuristic}} - \text{OFV}_{\text{CPLEX}}}{\text{OFV}_{\text{CPLEX}}}$. Figure 13a shows that only in the third instance (20-3-4-a), CPLEX has obtained a better result than all meta-heuristic algorithms. According to the Table 5 and "Open depot" column, we found out that CPLEX and solution algorithms have led to the different location of depots because in the solution algorithms, the "*Location-Allocation*" stage is solved first and this lead to a local optimum (this is a heuristic approach); but still closed-to-optimal solutions are found. Figure 13b connects the running times resulting from running the meta-heuristic algorithms on different instances and shows that hybrid algorithms require more time. Moreover, the results show that GA, HGAVND, and HGALS algorithms have satisfactory performance in terms of solution quality and CPU time.

Table 4. Computational results of the small instances.

Instance	CPLEX			GA		
	Time (s)	Cost	Open depot	Time (s)	Cost	Open depot
7-2-3-a	13.54	**748.98**	1	**2.67**	**748.98**	1
7-2-3-b	25.89	**788.82**	1	**2.66**	**788.82**	1
7-2-3-c	8.35	**839.00**	–	**2.79**	**839.00**	–
7-2-3-d	**1.01**	**839.00**	–	2.69	**839.00**	–
7-2-3-e	**1.02**	**839.00**	–	3.21	**839.00**	–
8-2-3-a	33.38	**604.87**	2	**3.50**	**604.87**	2
8-2-3-b	116.09	**665.74**	2	6.05	**665.74**	2
8-2-3-c	68.03	**701.02**	2	**5.97**	**701.02**	2
8-2-3-d	284.10	**745.83**	2	**6.20**	**745.83**	2
8-2-3-e	**2.00**	**878.00**	–	**7.15**	**878.00**	–
9-2-3-a	161.10	**666.25**	1	10.76	**666.25**	1
9-2-3-b	1948.02	**686.75**	1	10.67	**686.75**	1
9-2-3-c	661.61	**753.53**	1	10.43	**753.53**	1
9-2-3-d	3469.29	**799.40**	1	**10.59**	**799.40**	1
9-2-3-e	**0.64**	**992.00**	–	10.69	**992.00**	–

Instance	HGAVND			HGALS		
	Time (s)	Cost	Open depot	Time (s)	Cost	Open depot
7-2-3-a	6.19	**748.98**	1	5.44	**748.98**	1
7-2-3-b	4.21	**788.82**	1	4.62	**788.82**	1
7-2-3-c	6.01	**839.00**	–	5.40	**839.00**	–
7-2-3-d	5.23	**839.00**	–	5.40	**839.00**	–
7-2-3-e	6.10	**839.00**	–	5.51	**839.00**	–
8-2-3-a	6.16	**604.87**	2	5.45	**604.87**	2
8-2-3-b	8.48	**665.74**	2	**6.02**	**665.74**	2
8-2-3-c	8.82	**701.02**	2	7.96	**701.02**	2
8-2-3-d	7.88	**745.83**	2	11.69	**745.83**	2
8-2-3-e	7.51	**878.00**	–	8.44	**878.00**	–
9-2-3-a	**6.57**	**666.25**	1	7.21	**666.25**	1
9-2-3-b	9.23	**686.75**	1	**8.97**	**686.75**	1
9-2-3-c	9.91	**753.53**	1	**9.86**	**753.53**	1
9-2-3-d	10.81	**799.40**	1	10.66	**799.40**	1
9-2-3-e	**10.60**	**992.00**	–	11.87	**992.00**	–

5.2.3. Analysis of large instances and benchmark

Twelve instances with $|J|+1$ scenarios that have the largest probability of occurrence are solved in "a" failure probability and results are given in Table 6, in which the number and probability of scenarios are written below the instances as well. Figure 14 shows that HGAVND and HGALS algorithms significantly outperform GA in terms of solution quality. In the sixth instance (40-5-6-a), HGALS has obtained a disappointing result, but according to Table 6 and "Open depot" columns, we found out that they have a different location, because in the meta-heuristic algorithms, first the "*Location-Allocation*" is solved and this is a heuristic approach. Figure 14 shows that HGAVND is between the GA and HGALS in term of CPU time for small and medium instances. But this algorithm performs faster when dealing with large instances. Finally, the performance of the proposed algorithms is compared against the instances derived from Barreto's test set by Angelelli and Mansini's

TABLE 5. Computational results of the medium instances.

Instance	Solution approach	Time (s)	BestCost	WorstCost	AvgCost	Open depot of BestCost	DIFF (%)	σ
10-3-3-a	CPLEX	10 752.31	**663.85**	–	–	1	–	–
	GA	**14.00**	**663.85**	**663.85**	**663.85**	1	**0.00**	0
	HGAVND	32.54	**663.85**	**663.85**	**663.85**	1	**0.00**	0
	HGALS	24.93	**663.85**	**663.85**	**663.85**	1	**0.00**	0
15-3-3-a	CPLEX	10 800.00	528.85	–	–	1	–	–
	GA	**132.73**	530.98	536.25	533.84	1	0.40	**1.874**
	HGAVND	145.09	**525.59**	**532.86**	529.78	1	**–0.61**	3.430
	HGALS	146.26	**525.59**	534.92	**528.91**	1	**–0.61**	4.125
20-3-4-a	CPLEX	10 800.00	947.85	–	–	1	–	–
	GA	177.82	1022.58	1035.19	1028.65	1, 2	7.88	4.560
	HGAVND	193.23	**1021.96**	1035.72	1028.30	1, 2	**7.81**	5.800
	HGALS	**189.03**	1024.63	**1033.58**	**1027.85**	1, 2	8.10	**3.986**
25-3-4-a	CPLEX	10 800.00	975.68	–	–	1, 3	–	–
	GA	**187.84**	901.24	914.10	907.97	1, 3	–7.62	**4.078**
	HGAVND	202.79	900.86	915.27	907.84	1, 3	–7.48	5.966
	HGALS	212.43	**897.93**	**910.22**	**904.68**	1, 3	**–7.96**	5.587

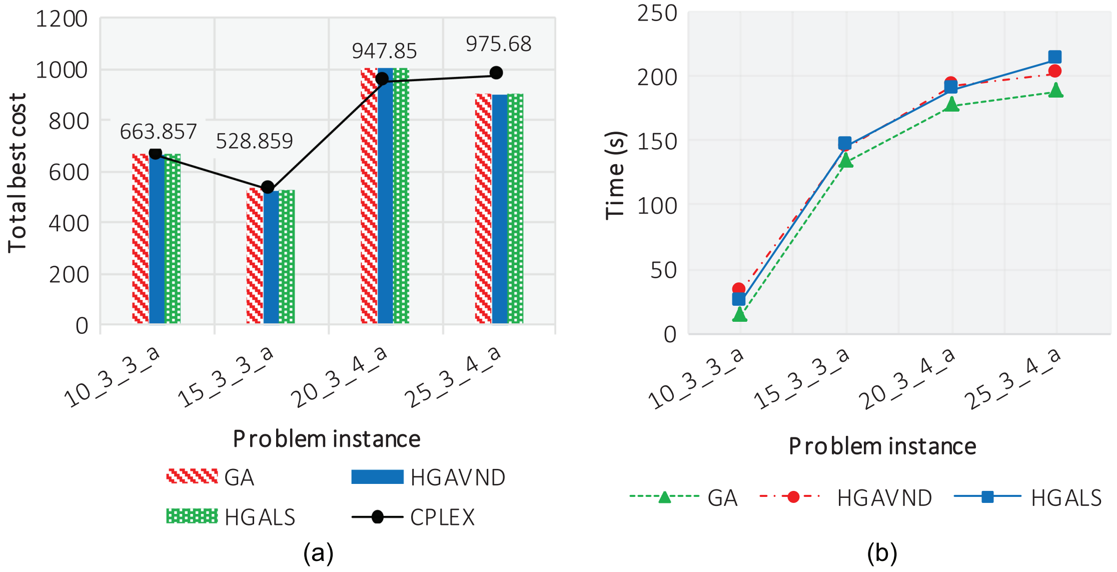


FIGURE 13. Medium-scale instances analysis. (a) Total best cost *vs.* Instance size. (b) CPU time *vs.* Instance size.

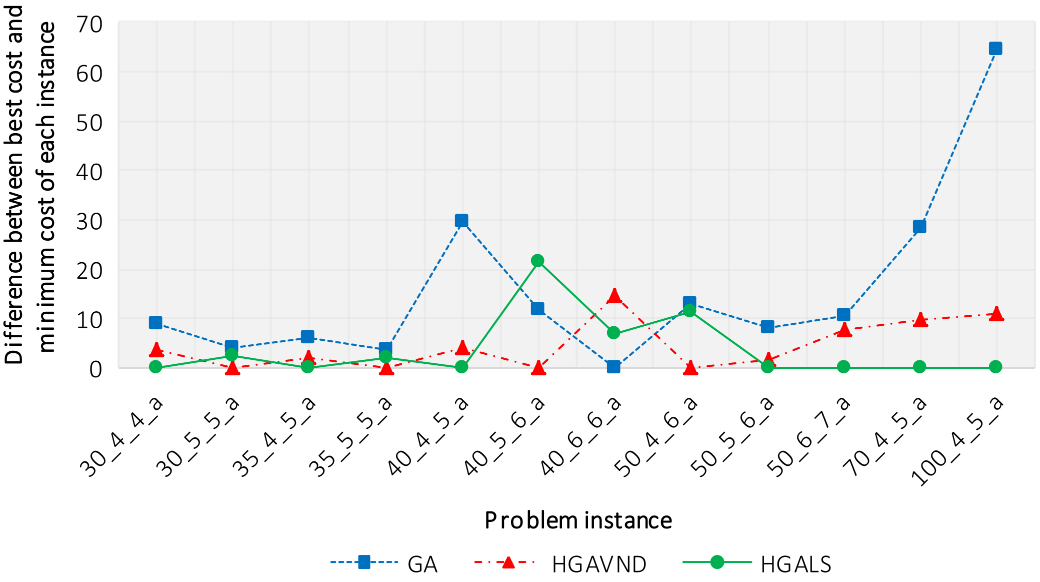


FIGURE 14. Best cost *vs.* Instance size.

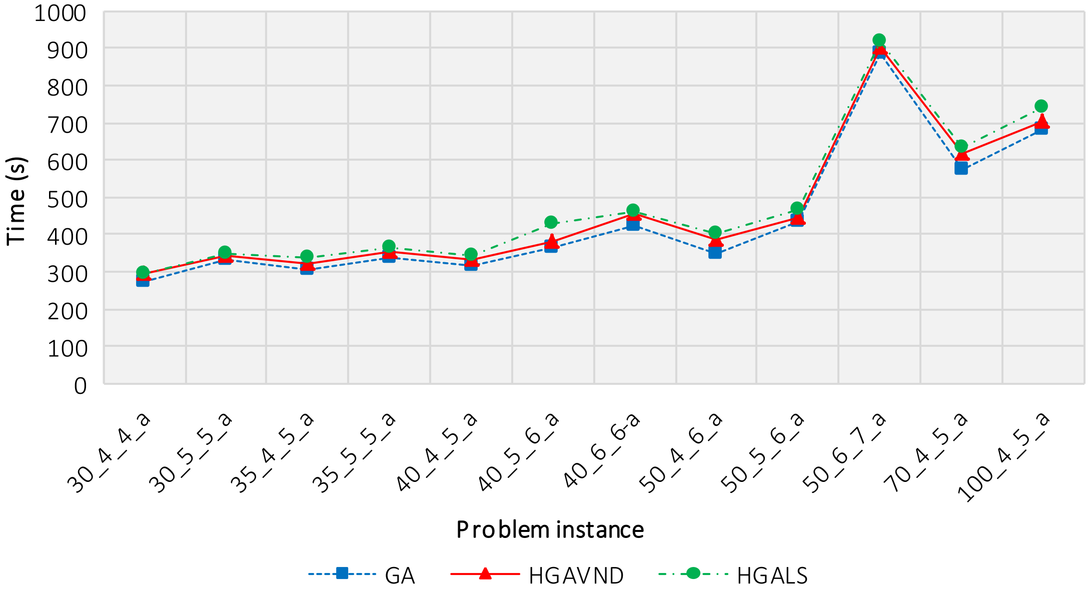


FIGURE 15. CPU time *vs.* Instance size.

separation approach (*i.e.*, BAM) benchmark in the literature [22] as reported in Table 7. This comparison shows the superiority of the proposed algorithms compared to those of in the literature.

5.2.4. Analysis in the different number of scenarios

According to the previous section, we choose HGALS and all results are obtained from HGALS. The reason for selecting HGALS is that the HGALS obtains better solutions in terms of the objective function value and

Table 6. Computational results of the large instances.

Instances	GA						HGAVND						HGALS					
	Time (s)	Best	Worst	Avg	σ	Depot	Time (s)	Best	Worst	Avg	σ	Depot	Time (s)	Best	Worst	Avg	σ	Depot
30-4-4-a 5(0.9888)	**274**	1475.5	1481.3	1478.9	2.54	1, 3, 4	292	1470.0	1481.6	1473.2	4.66	1,3,4	297	**1466.5**	1476.9	**1470.9**	**2.35**	1,3,4
30-5-5-a 6(0.9653)	**332**	1513.4	1537.9	1530.5	9.02	1, 3, 4	342	1509.3	1527.6	1518.3	7.34	1,3,4	350	**1511.8**	1527.2	**1521.7**	**5.45**	1,3,4
35-4-5-a 5(0.9949)	**304**	1182.8	1191.4	1187.3	3.37	1, 2, 3	320	1178.6	1189.4	1182.1	4.50	1,2,3	337	**1176.7**	1185.0	**1181.0**	**2.93**	1,2,3
35-5-5-a 6(0.9656)	**336**	1790.1	1812.8	1804.8	8.27	2, 3, 5	351	1786.4	1810.1	1800.1	10.47	2,3,5	364	**1788.2**	1806.9	**1799.2**	**6.95**	2,3,5
40-4-5-a 5(0.9946)	**318**	1568.6	1578.0	1572.4	3.64	1, 2, 4	330	1543.0	1567.1	1554.2	8.07	1,2,4	344	**1539.0**	1542.2	**1540.6**	**0.99**	1,2,4
40-5-6-a 6(0.9644)	**364**	1252.5	1265.1	1252.8	10.83	1, 4, 5	381	1220.9	1255.6	1241.3	12.99	1,4,5	428	**1242.2**	1265.4	**1254.8**	**8.65**	1,2,4,5
40-6-6-a 7(0.9610)	**423**	1663.0	1690.3	1675.0	17.08	2, 4, 5, 6	456	1667.6	1690.6	1679.7	7.93	2,4,5,6	459	**1659.7**	1680.5	**1671.1**	**9.08**	2,4,5,6
50-4-6-a 5(0.9756)	**350**	1441.7	1467.0	1457.0	10.05	2, 3, 4	387	1428.7	1465.6	1451.4	13.46	2,3,4	403	**1439.9**	1454.7	**1448.2**	**5.29**	2,3,4
50-5-6-a 6(0.9706)	**434**	1990.0	2027.0	2014.7	12.92	1, 2, 3, 4	446	1983.4	2017.8	2004.0	13.57	1,2,3,4	465	**1981.9**	2003.7	**1995.8**	**8.35**	1,2,3,4
50-6-7-a 7(0.9357)	**886**	1705.2	1775.6	1753.9	25.17	2, 3, 4, 5	902	1702.4	1777.5	1744.4	24.31	2,3,4,5	919	**1694.8**	1752.8	**1737.3**	**21.49**	2,3,4,5
70-4-5-a 5(0.9684)	**576**	1054.6	1066.1	1060.3	4.35	2, 4	617	1035.9	1047.3	1043.3	4.17	2,4	632	**1026.2**	1039.8	**1034.2**	**4.24**	2,4
100-4-5-a 5(0.9777)	**679**	1110.4	1124.0	1116.2	4.64	1, 2, 3	701	1056.7	1077.1	1066.0	7.02	1,2,3	740	**1045.8**	1060.2	**1051.0**	**4.45**	1,2,3

TABLE 7. Computational results for the BAM data-set.

Instances	DSS[1]	CPLEX			B&C		GA		HGAVND		HGALS	
		UB	Gap (%)	Time (s)	Obj.	Time (s)	Obj.	Time (s)	Obj.	Time (s)	Obj.	Time (s)
Srivastava86-8x2	W	**873.58**	0.00	1.59	**873.58**	0.00	**873.58**	3.97	**873.58**	5.33	**873.58**	5.83
Srivastava86-8x2	Z	**806.06**	0.00	2.89	**806.06**	0.00	**806.06**	4.06	**806.06**	5.87	**806.06**	5.81
Perl83-12x2	W	**243.98**	0.00	9.95	**243.98**	0.57	**243.98**	5.48	**243.98**	7.29	**243.98**	7.95
Perl83-12x2	Z	**243.98**	0.00	16.79	**243.98**	0.65	**243.98**	5.92	**243.98**	7.23	**243.98**	8.35
Gaskell67-21x5	W	**528.42**	0.00	5647.23	**528.42**	290.18	557.12	15.11	557.12	17.73	557.12	18.54
Gaskell67-21x5	Z	**513.30**	0.00	7902.63	**513.30**	89.43	**513.30**	14.06	**513.30**	17.00	**513.30**	19.11
Gaskell67-22x5	W	**653.80**	0.00	5135.28	**653.80**	3.76	**653.80**	14.75	**653.80**	18.36	**653.80**	19.02
Gaskell67-22x5	Z	**653.80**	0.00	6282.74	**653.80**	2.66	**653.80**	15.59	**653.80**	19.46	**653.80**	19.74
Min92-27x5	W	3326.81	12.63	14 400.00	**3142.02**	19.66	**3142.02**	24.15	**3142.02**	27.56	**3142.02**	29.05
Min92-27x5	Z	3301.85	11.62	14 400.00	**3142.02**	18.12	**3142.02**	22.29	**3142.02**	25.61	**3142.02**	27.41
Gaskell67-29x5	W	597.50	6.13	14 400.00	**592.10**	6.76	**592.10**	21.18	**592.10**	30.70	**592.10**	33.36
Gaskell67-29x5	Z	597.50	8.38	14 400.00	**592.10**	100.91	**592.10**	24.39	**592.10**	31.58	**592.10**	34.94
Gaskell67-32x5_1	W	757.52	21.91	14 400.00	**696.38**	5694.09	**696.38**	47.36	**696.38**	53.87	**696.38**	55.97
Gaskell67-32x5_1	Z	719.52	20.74	14 400.00	**643.37**	259.67	**643.37**	50.68	**643.37**	57.42	**643.37**	59.28
Gaskell67-32x5_2	W	643.94	14.98	14 400.00	**595.27**	15.68	**595.27**	32.20	**595.27**	37.15	**595.27**	37.49
Gaskell67-32x5_2	Z	606.90	12.04	14 400.00	**564.33**	30.90	**564.33**	31.08	**564.33**	35.55	**564.33**	37.80
Gaskell67-36x5	W	625.50	32.71	14 400.00	**540.37**	27.20	**540.37**	42.25	**540.37**	51.45	**540.37**	53.92
Gaskell67-36x5	Z	603.56	25.37	14 400.00	**540.37**	46.46	**540.37**	23.93	**540.37**	47.36	**540.37**	51.21
Ch69-50x5	W	N/A	–	–	**708.37**	14 400.00	**708.37**	198.26	**708.37**	209.78	**708.37**	213.73
Ch69-50x5	Z	N/A	–	–	701.91	14 400.00	701.63	212.67	701.63	221.71	**700.96**	232.20
Perl83-55x15	W	N/A	–	–	1330.85	14 400.00	1338.30	367.51	**1328.95**	392.82	**1328.95**	405.13
Perl83-55x15	Z	N/A	–	–	1338.30	14 400.00	1330.70	332.62	1325.72	388.10	**1322.61**	394.22
Ch69-75x10	W	N/A	–	–	1177.65	14 400.00	1168.34	484.06	1135.86	522.10	**1133.03**	529.49
Ch69-75x10	Z	N/A	–	–	1108.82	14 400.00	1110.49	508.44	**1077.56**	548.19	**1077.56**	565.07
Perl83-85x7	W	N/A	–	–	1901.09	14 400.00	1863.35	711.28	1855.91	731.47	**1855.55**	734.69
Perl83-85x7	Z	N/A	–	–	1893.28	14 400.00	1881.04	630.75	**1858.72**	696.28	1860.85	705.29
Daskin95-88x8	W	N/A	–	–	533.37	14 400.00	536.93	741.93	499.13	780.18	**498.58**	787.53
Daskin95-88x8	Z	N/A	–	–	487.81	14 400.00	482.75	765.76	**479.99**	801.63	**479.99**	809.28
Ch69-100x10	W	N/A	–	–	1079.41	14 400.00	1013.48	859.90	1015.43	933.67	**1012.80**	968.70
Ch69-100x10	Z	N/A	–	–	1038.26	14 400.00	1015.17	869.79	**998.50**	954.87	**998.50**	981.35
Average					962.15	5980.22	958.15	236.05	952.66	255.91	952.39	261.72

Notes. DSS[1]: Demand Separation Strategy. Italic numbers present the optimal solution. Bold numbers present the best solution in each instance.

TABLE 8. Computational results instances with the different number of scenarios.

Instance	Number of scenarios												
	1			5			11			16			Cost
	Time (s)	Cost	Depot	Time (s)	Cost	Depot	Time (s)	Cost	Depot	Time (s)	Cost	Depot	Increase (%)
30-4-4-a	180.06	1455.20	1 3 4	290.28	1472.41	1 3 4	521.56	1484.16	1 3 4	591.17	1481.93	1 3 4	1.83
35-4-5-a	198.19	1173.52	1 2 3	313.22	1183.22	1 2 3	531.56	1180.69	1 2 3	665.09	1183.02	1 2 3	0.81
40-4-5-a	210.32	1533.97	1 2 4	335.01	1569.26	1 2 4	574.21	1559.26	1 2 4	660.52	1570.52	1 2 4	2.38
50-4-6-a	244.61	1381.66	2 3 4	400.50	1455.78	2 3 4	626.36	1469.01	2 3 4	755.35	1460.36	2 3 4	5.69

the computational time. In addition, the HGALS has had also higher convergence rate (*i.e.*, lower standard deviation when being executed several times). Therefore, the analyses obtained from the HGALS algorithm would be more validated. Since the number of scenarios is different in Table 8, for a fair comparison of costs, the objective function must be calculated as follows:

$$\min\ \mathrm{OFV} = \sum_{i\in J} fd_j Y_j + \frac{1}{\sum_{s\in S} pr_s} \times \sum_{s\in S} pr_s \left(\sum_{i\in I\cup J_{ns}} \sum_{k\in I\cup J_{ns}} \sum_{h\in H} \sum_{j\in J_{ns}} tc_{ik} X_{ikhjs} + \sum_{i\in I} Z_{ies}\theta_i + \sum_{j\in J_{ns}} \sum_{i\in I} \sum_{h\in H} fvX_{jihjs} \right). \tag{5.4}$$

Four instances are solved in "a" failure probability. Four potential depots are considered and the number of all scenarios is 2^4, so we choose the different number of scenarios that have the largest probability of occurrence. The results are given in Table 8. Figure 16a shows that the smallest cost of each instance is in the one scenario, which does not take into account the disruptions. With increasing the number of scenarios, costs are rising, but there is not much difference between costs in 5, 11 and 16 scenarios. As shown in Figures 16a and 16b, we conclude that the number of scenarios that previously considered ($|J|+1$) was correct and suitable in terms of time.

The last column "Cost increase (%)" in Table 8 is calculated as $\frac{\mathrm{OFV}_{16\,\mathrm{scenarios}} - \mathrm{OFV}_{1\,\mathrm{scenario}}}{\mathrm{OFV}_{1\,\mathrm{scenario}}}$ that $\mathrm{OFV}_{1\,\mathrm{scenario}}$ represents the total cost if disruptions are not assumed and the $\mathrm{OFV}_{16\,\mathrm{scenarios}}$ represent the total cost if all disruptions are assumed. The results show that percentage of increased costs are between one to six percent that each manager in the supply chain is willing to accept it, because not considering the disruptions, imposed the staggering cost on the system.

Figures 17 and 18 illustrate the sensitivity of the total cost *vs.* the capacity of depots and vehicles, respectively. It can be seen in both Figures 17 and 18 that increasing the capacity will decreases the total cost. Actually, when the capacity of depots and vehicles increases, two event happen as: (1) less number of depots are opened and consequently lower fixed cost is imposed to the solution and (2) customers are allocated to closer depots and consequently lower transportation cost is paid in the solution.

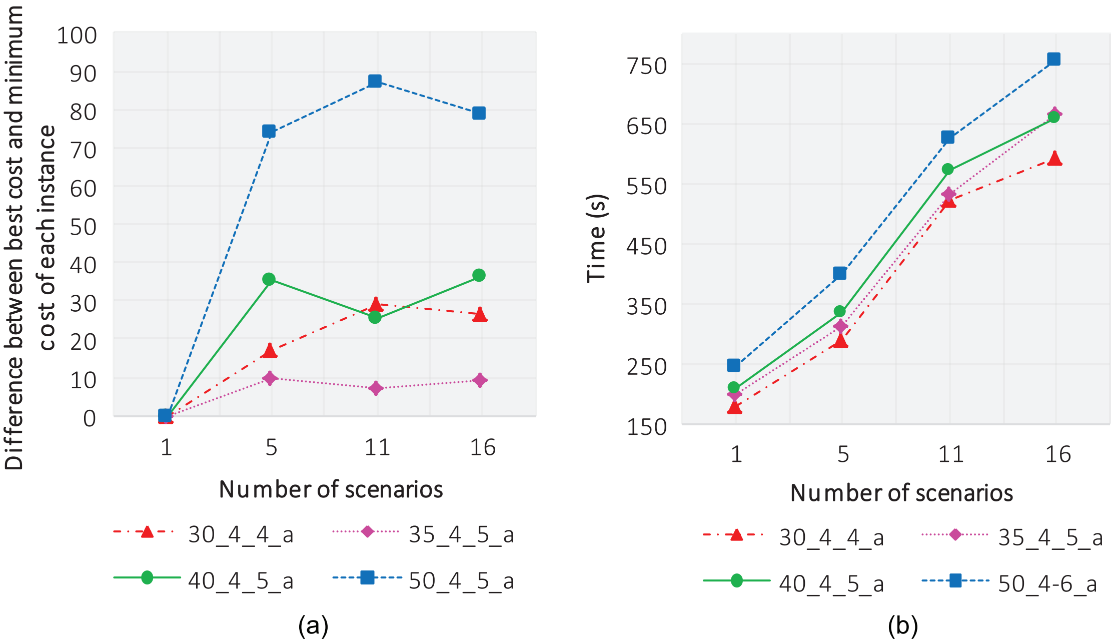


FIGURE 16. Analysis in a different number of scenarios. (a) Best cost *vs.* Number of scenarios. (b) CPU time *vs.* Number of scenarios.

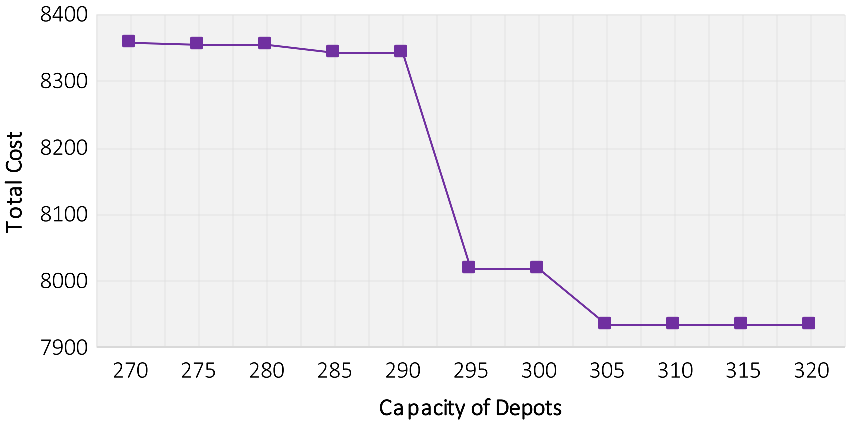


FIGURE 17. Total cost *vs.* Capacity of depots.

6. CONCLUSION

This paper develops a new mixed-integer mathematical model for a reliable capacitated location-routing problem with simultaneous pickup and delivery, wherein the depots are exposed to the risk of disruption. Any disruption in the location-routing network imposes a huge cost on the company and its stockholders. Considering disruption in the design phase of the network will alleviate this impact and let the network to resist disruption. The objective function of the proposed model minimizes the total expected cost of the network including the sum of location cost of depots, routing cost of vehicles and cost of unfulfilled demand of customers. The proposed

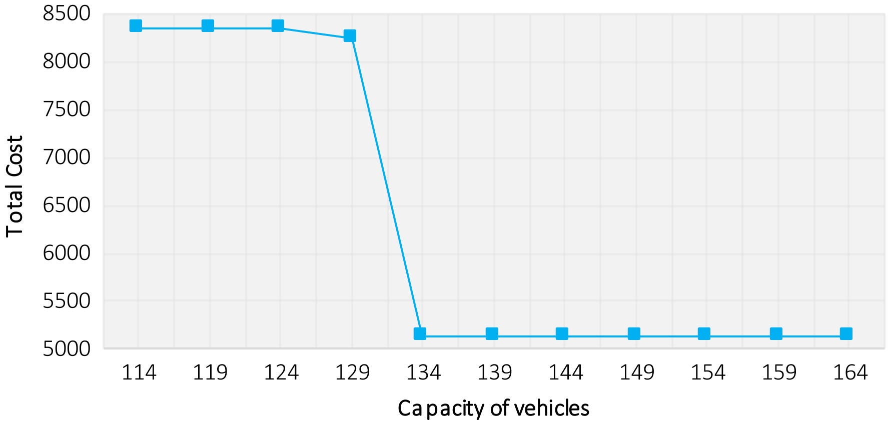


Figure 18. Total cost *vs.* Capacity of vehicles.

model is solved for small-sized instances and for large-sized problems, three efficient meta-heuristics are tailored and their performance is investigated through a comprehensive experimental research. All the meta-heuristics algorithms reach the optimal solution in the small instances and better solution comparing to CPLEX in the most cases in the medium instances, wherein the CPLEX is stopped after an enough huge computational time. The computational results demonstrate that the hybrid algorithms significantly outperform the classical genetic algorithm in terms of solution quality.

Possible future extensions include using multiple products, using time window for products, considering the location-routing-inventory problem and accounting for the cost of carbon emissions and developing other meta-heuristics for the problem. In addition, comparing the proposed hybrid meta-heuristic algorithms with other existing meta-heuristic algorithms such as Tabu Search (TS), Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithm would be an important future research direction.

References

[1] A. Ahmadi-Javid and A.H. Seddighi, A location-routing problem with disruption risk. *Transp. Res. Part E: Logistics Transp. Rev.* **53** (2013) 63–82.

[2] M. Albareda-Sambola, E. Fernández and G. Laporte, Heuristic and lower bound for a stochastic location-routing problem. *Eur. J. Oper. Res.* **179** (2007) 940–955.

[3] E. Ardjmand, G. Weckman, N. Park, P. Taherkhani and M. Singh, Applying genetic algorithm to a new location and routing model of hazardous materials. *Int. J. Prod. Res.* **53** (2015) 916–928.

[4] S. Babaie-Kafaki, R. Ghanbari and N. Mahdavi-Amiri, Hybridizations of genetic algorithms and neighborhood search metaheuristics for fuzzy bus terminal location problems. *Appl. Soft Comput.* **46** (2016) 220–229.

[5] C.-L. Chen and W.-C. Lee, Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. *Comput. Chem. Eng.* **28** (2004) 1131–1144.

[6] H. Chentli, R. Ouafi, and W.R. Cherif-Khettaf, A selective adaptive large neighborhood search heuristic for the profitable tour problem with simultaneous pickup and delivery services. *RAIRO:OR* **52** (2018) 1295–1328.

[7] H. Chentli, R. Ouafi and W.R. Cherif-Khettaf, Impact of iterated local search heuristic hybridization on vehicle routing problems: application to the capacitated profitable tour problem. In: International Conference on Operations Research and Enterprise Systems. Springer, Cham (2018) 80–101.

[8] W.R. Cherif-Khettaf, M.H. Rachid, C. Bloch and P. Chatonnay, New notation and classification scheme for vehicle routing problems. *RAIRO:OR* **49** (2015) 161–194.

[9] T. Cui, Y. Ouyang and Z.-J.M. Shen, Reliable facility location design under the risk of disruptions. *Oper. Res.* **58** (2010) 998–1011.

[10] E. Dehghani, M.S. Pishvaee and M.S. Jabalameli, A hybrid Markov process-mathematical programming approach for joint location-inventory problem under supply disruptions. *RAIRO:OR* **52** (2018) 1147–1173.

[11] H. Derbel, B. Jarboui, S. Hanafi and H. Chabchoub, Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Syst. App.* **39** (2012) 2865–2871.

[12] M. Drexl and M. Schneider, A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.* **241** (2015) 283–308.

[13] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing. Springer **53** (2003).

[14] J.W. Escobar, R. Linfati and P. Toth, A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Comput. Oper. Res.* **40** (2013) 70–79.

[15] J.W. Escobar, R. Linfati, M.G. Baldoquin and P. Toth, A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem. *Transp. Res. Part B: Methodol.* **67** (2014) 344–356.

[16] K.M. Ferreira and T.A. de Queiroz, Two effective simulated annealing algorithms for the Location-Routing Problem. *Appl. Soft Comput.* **70** (2018) 389–422.

[17] N. Ghaffari-Nasab, M.S. Jabalameli, M.B. Aryanezhad and A. Makui, Modeling and solving the bi-objective capacitated location-routing problem with probabilistic travel times. *Int. J. Adv. Manuf. Technol.* **67** (2013) 2007–2019.

[18] J. Holland, Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control Artif. Intell.* (1975).

[19] S.-H. Huang, Solving the multi-compartment capacitated location routing problem with pickup–delivery routes and stochastic demands. *Comput. Ind. Eng.* **87** (2015) 104–113.

[20] S. Karakatič and V. Podgorelec, A survey of genetic algorithms for solving multi depot vehicle routing problem. *Appl. Soft Comput.* **27** (2015) 519–532.

[21] A. Karaoglan and F. Altiparmak, A memetic algorithm for the capacitated location-routing problem with mixed backhauls. *Comput. Oper. Res.* **55** (2015) 200–216.

[22] I. Karaoglan, F. Altiparmak, I. Kara and B. Dengiz, A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery. *Eur. J. Oper. Res.* **211** (2011) 318–332.

[23] A. Karaoglan, F. Altiparmak, I. Kara and B. Dengiz, The location-routing problem with simultaneous pickup and delivery: formulations and a heuristic approach. *Omega* **40** (2012) 465–477.

[24] M. Karimi-Mamaghan, M. Mohammadi, B. Pasdeloup, R. Billot and P. Meyer, An online learning-based metaheuristic for solving combinatorial optimization problems. In: 21ème congrès annuel de la société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF) (2020).

[25] M. Karimi-Mamaghan, M. Mohammadi, P. Jula, A. Pirayesh and H. Ahmadi. A learning-based metaheuristic for a multi-objective agile inspection planning model under uncertainty. *Eur. J. Oper. Res.* **285** (2020) 513–537.

[26] M. Karimi-Mamaghan, M. Mohammadi, A. Pirayesh, A.M. Karimi-Mamaghan and H. Irani, Hub-and-spoke network design under congestion: a learning based metaheuristic. *Transp. Res. Part E: Logistics Transp. Rev.* **142** (2020) 102069.

[27] K. Khalili-Damghani, A.-R. Abtahi and A. Ghasemi, A new bi-objective location-routing problem for distribution of perishable products: evolutionary computation approach. *J. Math. Model. Algorithms Oper. Res.* **14** (2015) 287–312.

[28] Q. Li, B. Zeng and A. Savachkin, Reliable facility location design under disruptions. *Comput. Oper. Res.* **40** (2013) 901–909.

[29] R.B. Lopes, C. Ferreira and B.S. Santos, A simple and effective evolutionary algorithm for the capacitated location-routing problem. *Comput. Oper. Res.* **70** (2016) 155–162.

[30] S. Majidi, S.-M. Hosseini-Motlagh, S. Yaghoubi and A. Jokar, Fuzzy green vehicle routing problem with simultaneous pickup – delivery and time windows. *RAIRO:OR* **51** (2017) 1151–1176.

[31] Y. Marinakis, An improved particle swarm optimization algorithm for the capacitated location routing problem and for the location routing problem with stochastic demands. *Appl. Soft Comput.* **37** (2015) 680–701.

[32] E. Mehdizadeh, R. Tavakkoli-Moghaddam and M. Yazdani, A vibration damping optimization algorithm for a parallel machines scheduling problem with sequence-independent family setup times. *Appl. Math. Model.* **39** (2015) 6845–6859.

[33] M.T. Melo, S. Nickel and F. Saldanha-Da-Gama, Facility location and supply chain management – A review. *Eur. J. Oper. Res.* **196** (2009) 401–412.

[34] D.M. Miranda, J. Branke and S.V. Conceição, Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time. *Appl. Soft Comput.* **70** (2018) 66–79.

[35] N. Mladenović and P. Hansen, Variable neighborhood search. *Comput. Oper. Res.* **24** (1997) 1097–1100.

[36] M. Mohammadi and R. Tavakkoli-Moghaddam, Design of a fuzzy bi-objective reliable p-hub center problem. *J. Intell. Fuzzy Syst.* **30** (2016) 2563–2580.

[37] M. Mohammadi, R. Tavakkoli-Moghaddam, H. Tolouei and M. Yousefi, Solving a hub covering location problem under capacity constraints by a hybrid algorithm. *J. Appl. Oper. Res.* **2** (2010) 109–116.

[38] M. Mohammadi, R. Tavakkoli-Moghaddam, A. Siadat and J.-Y. Dantan, Design of a reliable logistics network with hub disruption under uncertainty. *Appl. Math. Model.* **40** (2016) 5621–5642.

[39] M. Mohammadi, P. Jula and R. Tavakkoli-Moghaddam, Design of a reliable multi-modal multi-commodity model for hazardous materials transportation under uncertainty. *Eur. J. Oper. Res.* **257** (2017) 792–809.

[40] M. Mohammadi, J.Y. Dantan, A. Siadat and R. Tavakkoli-Moghaddam, A bi-objective robust inspection planning model in a multi-stage serial production system. *Int. J. Prod. Res.* **56** (2018) 1432–1457.

[41] M. Moshref-Javadi and S. Lee, The latency location-routing problem. *Eur. J. Oper. Res.* **255** (2016) 604–619.

[42] A. Nadizadeh and B. Kafash, Fuzzy capacitated location-routing problem with simultaneous pickup and delivery demands. *Transp. Lett.* **11** (2019) 1–19.

[43] A. Nadizadeh and H.H. Nasab, Solving the dynamic capacitated location-routing problem with fuzzy demands by hybrid heuristic algorithm. *Eur. J. Oper. Res.* **238** (2014) 458–470.

[44] G. Nagy and S. Salhi, Location-routing: issues, models and methods. *Eur. J. Oper. Res.* **177** (2007) 649–672.

[45] F. Oudouar, M. Lazaar, Z. E. Miloud, A novel approach based on heuristics and a neural network to solve a capacitated location routing problem. *Simul. Model. Pract. Theory* **100** (2020) 102064.

[46] E. Pekel, S. Soner Kara, Solving fuzzy capacitated location routing problem using hybrid variable neighborhood search and evolutionary local search. *Appl. Soft. Comput.* **83** (2019) 105665.

[47] K. Pichka, A.H. Bajgiran, M.E. Petering, J. Jang and X. Yue, The two echelon open location routing problem: mathematical model and hybrid heuristic. *Comput. Ind. Eng.* **121** (2018) 97–112.

[48] O. Polat, C. B. Kalayci, O. Kulak and H.-O. Günther, A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. *Eur. J. Oper. Res.* **242** (2015) 369–382.

[49] C. Prodhon and C. Prins, A survey of recent research on location-routing problems. *Eur. J. Oper. Res.* **238** (2014) 1–17.

[50] J. Respen, N. Zufferey and J.-Y. Potvin, Impact of vehicle tracking on a routing problem with dynamic travel times. *RAIRO:OR* **53** (2019) 401–414.

[51] S. Salhi and G. K. Rand, The effect of ignoring routes when locating depots. *Eur. J. Oper. Res.* **39** (1989) 150–156.

[52] L.V. Snyder, *Supply chain robustness and reliability: models and algorithms*. Ph.D diss., Northwestern University (2003).

[53] L.V. Snyder and M.S. Daskin, Reliability models for facility location: the expected failure cost case. *Transp. Sci.* **39** (2005) 400–416.

[54] J. Wang, K. Su and Y. Wu, The reliable facility location problem under random disruptions. *Wireless Pers. Commun.* **102** (2018) 2483–2497.

[55] W. Xie, Y. Ouyang and S.C. Wong, Reliable location-routing design under probabilistic facility disruptions. *Transp. Sci.* **50** (2015) 1128–1138.

[56] F. Yu and S.-Y. Lin, Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing. *Int. J. Prod. Res.* **54** (2016) 526–549.

[57] F. Yu, S.-W. Lin, W. Lee and C.-J. Ting, A simulated annealing heuristic for the capacitated location routing problem. *Comput. Ind. Eng.* **58** (2010) 288–299.

[58] X. Yu, Y. Zhou, X.-F. Liu, A novel hybrid genetic algorithm for the location routing problem with tight capacity constraints. *Appl. Soft. Comput.* **85** (2019) 105760.

[59] B. Zahiri, S.A. Torabi, M. Mohammadi and M. Aghabegloo, A multi-stage stochastic programming approach for blood supply chain planning. *Comput. Ind. Eng.* **122** (2018) 1–14.

[60] M.H.F. Zarandi, A. Hemmati and S. Davari, The multi-depot capacitated location-routing problem with fuzzy travel times. *Expert Syst. App.* **38** (2011) 10075–10084.

[61] M.H.F. Zarandi, A. Hemmati, S. Davari and I.B. Turksen, Capacitated location-routing problem with time windows under uncertainty. *Knowl.-Based Syst.* **37** (2013) 480–489.

[62] Y. Zare Mehrjerdi and A. Nadizadeh, Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. *Eur. J. Oper. Res.* **229** (2013) 75–84.

[63] 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. *Transp. Res. Part E: Logistics Transp. Rev.* **89** (2016) 182–214.

[64] Y. Zhang, M. Qi, W.-H. Lin and L. Miao, A metaheuristic approach to the reliable location routing problem under disruptions. *Transp. Res. Part E: Logistics Transp. Rev.* **83** (2015) 90–110.

[65] B. Zhang, H. Li, S. Li and J. Peng, Sustainable multi-depot emergency facilities location-routing problem with uncertain information. *Appl. Math. Comput.* **333** (2018) 506–520.