The Inventory-Routing Problem Subject to Vehicle Failure

ABSTRACT

The problem of vehicle failure is introduced to the field of inventory-routing problems and formulated. Since the model is NP-hard, we have designed an adapted social-based algorithm with four scenarios to solve it. We have embedded the “global war” procedure into the algorithm to reduce the chance of being trapped in local optimum and premature convergence. For validation, the following benchmarking methods are enlisted: (i) comparing with four other metaheuristic algorithms which are all calibrated and their effectiveness and efficiency evaluated; (ii) comparing with a suggested lower-bound. Sensitivity analyses of the algorithm provide insights for academics and practitioners.

***Keywords**:* Vehicle routing; vehicle failure; time windows; inventory management; meta-heuristic.

1. INTRODUCTION

The inventory-routing problem (IRP) makes inventory management and the vehicle routing problem (the VRP) decisions concurrently (Bertazzi *et al.* 2013; Li *et al.* 2016). The objective of the IRP is to determine when orders should be delivered to customers, the delivery volume to each customer, and the sequence of customer visits in each time period so that total transport and inventory costs are minimised while the customer demands are met (Song and Furman 2013; Yu *et al.* 2008).

According to Jetlund and Karimi (2004), in the chemical industry, operational logistic costs including inventory and routing decisions contribute 20% to the purchasing costs. Tarantilis and Kiranoudis (2001) state that IRP decisions can effectively decrease logistics costs in transporting fresh milk from dairy farms to processing plants. The IRP has been widely applied to many other industries, including department stores such as Wal-Mart, Kmart and Dillard (Çetinkaya and Lee 2000; Zhao *et al.* 2007); the beverage industry, for example the Hey Song Corporation (Dondo and Cerdá 2009); LCD and LED producers (Zhalechian *et al.* 2016); healthcare (Niakan and Rahimi 2015); dispatching refined oil products to retailers with tankers and barges (Fagerholt 2004); crude oil transport planning from a supply centre to customer facilities (Shen *et al.* 2011); perishable goods (Le *et al.* 2013); and liquefied natural gas (Stålhane *et al.* 2012). Many other researchers including Moin and Salhi (2007), Andersson *et al.* (2010) and Liu and Lee (2011) have confirmed the wide applications of the IRP to practice.

This research is motivated by a gap in the IRP literature considering; it is the occurrence of vehicle failure; this paper could therefore help organisations to be prepared for unforeseen circumstances in their service delivery planning. Vehicle breakdown leads to disruption in a transport system, which causes some operational risks such as delay, customer dissatisfaction, and disruption in information and cash flow in the supply chain (Chopra and Sodhi 2004). Based on some historical data from a highway over an observation period, Harwood *et al.* (1989) show that 44.5% of truck accidents are caused by truck failure (e.g. body, tank, valve or fitting). Such inefficiencies can lead to a decrease in profitability (Paul *et al.* 2015) and may damage sales and social perceptions (Zhu and Levinson 2012), which might have serious effects on the performance of a supply chain network (Klibi and Martel 2012). Planners try to avoid delays by making appropriate transport decisions (Zhu and Levinson 2012) and preserving inventory in the right size and location (Chopra and Sodhi 2004). Disruption management has been investigated in many planning areas like flight scheduling, project scheduling, production planning and supply chain coordination;

it has also been studied in the VRP (Mu *et al.* 2011) but not in the IRP. The most important research questions are as follows: (i) How can vehicle failure problems be categorised in the context of the IRP? (ii) How can the IRP with vehicle failure be modelled and solved efficiently? (iii) What managerial insights can be extracted based on the impact of vehicle failure on the IRP to help practitioners?

We formulate the IRP with soft time windows, which is computationally a complex problem. Even simple versions of the IRP with time windows (IRPTW) belong to the group of nondeterministic polynomial (NP) time problems (Aghezzaf *et al.* 2006). As the problem size increases, heuristic and meta-heuristic algorithms are the only viable alternatives to solve the problem (Nguyen *et al.* 2012). In this paper, we propose a metaheuristic approach called adapted social-based algorithm (ASBA) with four scenarios and show that it works more efficient than other well-known meta-heuristic techniques. This paper contributes to the literature in the following ways: (1) introducing the concept of vehicle failure to the IRP and modelling it; (2) applying ASBA with four scenarios to solve the model; and (3) embedding the "global war" procedure into the algorithm to reduce the chance of being trapped in local optimum and premature convergence. We have also suggested a lower bound to improve the validation process. Finally, we provide some insights for practitioners and academics related to this problem.

The rest of the paper is organised as follows: Section 2 positions the research in the literature of the IRP and highlights its contribution. The details of the problem are described in Section 3 and then mathematically formulated. The mathematical formulation of the problem is complicated so that it cannot be solved even for small-sized problems. Therefore, the model cannot be utilised for validating of our proposed algorithm. However, the model is useful because we will use it to introduce a lower bound. This rare situation made us create an unusual validation approach. In order to avoid confusion, the structure of the other section of the paper is illustrated in Figure 1. Section 4 explains the structure of the proposed algorithms. Data generation for different test problems is described in Section 5. Preparation and validation of the proposed solution algorithms are explained in Section 6. In section 7, the research insights are presented. Finally, Section 8 concludes the research.

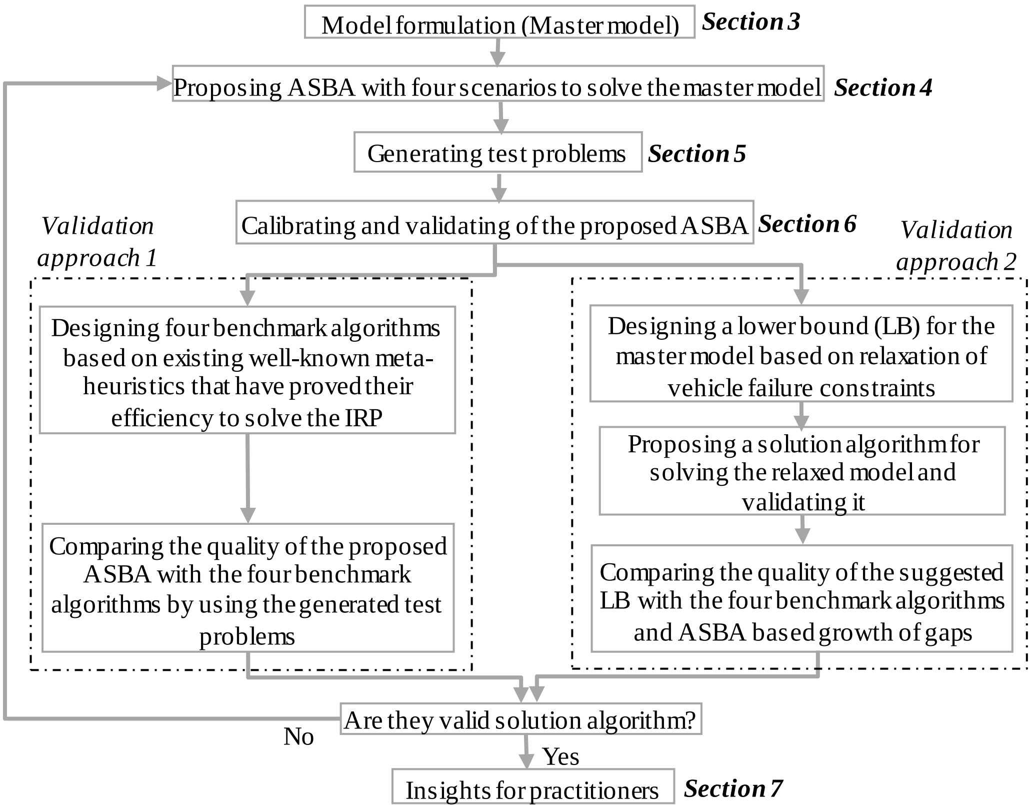

Model formulation (Master model)
Section 3
Proposing ASBA with four scenarios to solve the master model
Section 4
Generating test problems
Section 5
Calibrating and validating of the proposed ASBA
Section 6
Validation approach 1
Validation approach 2
Designing four benchmark algorithms based on existing well-known meta-heuristics that have proved their efficiency to solve the IRP
Comparing the quality of the proposed ASBA with the four benchmark algorithms by using the generated test problems
Designing a lower bound (LB) for the master model based on relaxation of vehicle failure constraints
Proposing a solution algorithm for solving the relaxed model and validating it
Comparing the quality of the suggested LB with the four benchmark algorithms and ASBA based growth of gaps
Are they valid solution algorithm?
No
Yes
Insights for practitioners
Section 7

Figure 1. The research design framework.

2. LITERATURE REVIEW

While no standard classifications for the IRP have been agreed by all researchers, Moin and Salhi (2007) and Andersson *et al.* (2010) provide comprehensive reviews for them. Focusing on various aspects of this research, we categorise the IRP literature in terms of (i) inventory, (ii) routing and (iii) inventory-routing assumptions and their features. These aspects, features and their possible types are depicted and coded in Table 1. Table 1 is partially adopted from Andersson *et al.* (2010). In fact, there are similarities between our categorisations and those of Andersson *et al.* (2010): (i) demand and inventory under "inventory" assumptions; (ii) routing, fleet composition and fleet size under "routing" assumptions; and (iii) time horizon under "inventory-routing". Other features – including (1) product type, holding cost and shortage cost under "inventory" assumptions, (2) vehicle speed, travel cost and time windows under "routing" assumptions, and (3) planning period under "inventory-routing" assumptions – are what we have added to this categorisation to position our research. Additionally, Andersson *et al.* (2010) investigated the related papers up to 2009 and we have updated the search up to 2018.

Following Table 1, we have identified features of the IRP papers in the literature to form Table 2. Table 2 lists the IRP research papers in the literature chronologically. In Table 2, in addition to the features mentioned in Table 1, we have also illustrated the used solution techniques. The last row of the table compares and contrasts our research with other studies in the literature.

Table 1. Classification of criteria involved in the IRP (partially adapted from Andersson *et al.* 2010).

ASSUMPTION	FEATURE	TYPE	CODE
Inventory related	**Demand**	Deterministic	De
		Stochastic	St
		Forecasted	Fo
	Inventory	Fixed	Fi
		Stock-out	So
		Lost sale	Ls
		Back-order	Bo
	Product type	Single-product	Sp
		Multi-product	Mp
	Holding cost	Constant	Co
		Period varying	Pv
	Shortage cost	Constant	Co
		Period varying	Pv
Routing related	**Routing**	Direct	Di
		Multi visit	Mv
		Continuous	Con
	Fleet size	Single-vehicle	Si
		Multi-vehicle	Mu
		Unconstrained	Uc
	Fleet composition	Homogenous	Ho
		Heterogeneous	He
	Vehicle speed	Yes	Y
		No	N
	Vehicle failure	Yes	Y
		No	N
	Travel cost	Certain	Cer
		Uncertain	Ucr
	Time windows	Soft	Sf
		Hard	Ha
Inventory-routing related	**Time horizon**	Instant	In
		Finite	Fn
		Infinite	If
	Planning period	Single-period	Spe
		Multi-period	Mpe

2.1. Basic assumptions

According to Table 2, we consider an IRP problem with these features: multi-period (with a finite time horizon), multi-product, holding and backlogging costs, soft time windows, multiple vehicle, heterogeneous fleet of vehicles, deterministic-forecasted demand, variable vehicle speed (before and after failure), capacitated retailers and customer inventory policy (here, order-up-to policy).

Table 2. Review of the IRP studies.

REFERENCES	Inventory related					Routing related						Inventory -routing		Solution method
	Demand	Inventory	Product type	Holding cost	Shortage cost	Routing	Fleet size	Fleet composition	Vehicle speed	Travel cost	Time windows	Time Horizon	Planning period	
Federgruen and Zipkin (1984)	St	So	Sp	Co	Co	Mv	Mu	He	N	Cer	-	In	Spe	Generalised Benders decomposition
Dror et *al*. (1985)	St	Fi	Sp	-	-	Mv	Mu	Ho	N	Cer	-	In	Mpe	i) Customer selection/ routing heuristic; ii) Single customer analysis
Federgruen et *al*. (1986)	St	So	Sp	-	Co	Mv	Mu	He	N	Cer	-	In	Spe	i) Routing/ allocation decomposition; ii) Interchange heuristic
Anily and Federgruen (1990)	De	Fi	Sp	Co	-	Mv	Uc	Ho	N	Cer	-	If	Mpe	i) Lower bound; ii) Fixed partitions
Chandra (1993)	De	Fi	Mp	Co	-	Mv	Uc	Ho	N	Cer	-	Fn	Mpe	i) Decomposition; ii) Sequential heuristic; iii) Local search
Carter et *al*. (1995)	De	Bo	Mp	Pv	Pv	Mv	Mu	Ho	N	Cer	-	Fn	Mpe	i) Allocation/ routing decomposition; ii) Iterative heuristic
Barnes-Schuster and Bassok (1997)	St	Bo	Sp	Co	Co	Di	Uc	Ho	N	Cer	-	If	Spe	i) Lower bound; ii) Analytical
Viswanathan and Mathur (1997)	De	Fi	Mp	Co	-	Mv	Mu	Ho	N	Cer	-	If	Spe	Constructive insertion heuristic
Berman and Larson (2001)	St	So	Sp		Co	Mv	Si	Ho	N	Cer	-	In	Spe	Stochastic dynamic programming
Bertazzi et *al*. (2002)	De	Fi	Mp	Co	-	Mv	Si	Ho	N	Cer	-	Fn	Spe	i) Constructive heuristic; ii) Local search
Gaur and Fisher (2004)	De	Fi	Sp	-	-	Mv	Mu	He	N	Cer	-	Fn	Mpe	i) Fixed partitioning policy; ii) Randomised sequential matching algorithm
Kleywegt et *al*. (2004)	St	Ls	Sp	Pv	Pv	Mv	Mu	Ho	N	Cer	-	If	Mpe	i) Markov decision process; ii) Iterative heuristic
Abdelmaguid and Dessouky (2006)	De	Bo	Sp	Co	Co	Mv	Mu	He	N	Cer	-	Fn	Mpe	Genetic algorithm (GA)
Aghezzaf et *al*. (2006)	De	Fi	Sp	Co	-	Mv	Mu	Ho	Y	Cer	-	If	Spe	Heuristic column generation
Archetti et *al*. (2007)	De	Fi	Sp	Pv	-	Mv	Si	Ho	N	Cer	-	Fn	Mpe	Branch and cut
Li et *al*. (2008b)	De	Fi	Sp	Co	-	Di	Si	Ho	N	Cer	-	If	Mpe	Construction algorithm
Raa and Aghezzaf (2008)	De	Fi	Sp		-	Mv	Mu	Ho	N	Cer	-	If	Spe	Heuristic column generation
Raa and Aghezzaf (2009)	De	Fi	Sp	Co	-	Mv	Mu	He	N	Cer	-	Fn	Spe	Heuristic column generation
Hemmelmayr et *al*. (2009)	De	Fi	Sp	-	-	Mv	Mu	Ho	N	Cer	-	Fn	Mpe	i) Giant tour with skipped customers; ii) Variable neighbourhood research (VNS)
Zhao et *al*. (2008)	De	Fi	Sp	Co	-	Di-Mv		Ho	N	Cer	-	If	Spe	i) Fixed partition-Power-of-tow; ii) VNS
Abdelmaguid et *al*. (2009)	St	Bo	Sp	Co	Co	Mv	Mu	He	N	Cer	-	Fn	Mpe	Constructive and improvement heuristics
Huang and Lin (2010)	St	So- Bo	Mp	-	Co	Mv	Mu	Ho	Y	Ucr	-	In	Spe	Modified ant colony
Liu and Lee (2011)	St	Ls	Sp	Co	Co	Mv	Si	Ho	Y	Cer	Sf	In	Spe	Combination of VNS and tabu search (TS)
Moin et *al*. (2011)	De	Bo	Sp	Co	-	Mv	Uc	Ho	N	Cer	-	Fn	Mpe	Hybrid GA
Shen et *al*. (2011)	De-Fo	Bo	Sp	Co	Co	Di	Mu	He	N	Cer	-	Fn	Mpe	Lagrangian relaxation approach
Yu et *al*. (2012)	De	Ls	Sp	Pv	-	Mv	Uc	Ho	N	Cer	-	Fn	Mpe	i) Hybrid approach; ii) Lagrangian relaxation; iii) Partial linearisation approach
Mjirda et *al*. (2012)	De	Fi	Mp	Co	-	Mv	Mu	Ho	N	Cer	-	Fn	Mpe	i) VNS; ii) Variable neighbourhood descent algorithm
Popović et *al*. (2012)	De	Fi	Mp	Co	-	Di-Mv	Uc	Ho	N	Cer	-	Fn	Mpe	i) VNS; ii) MILP; iii) Compartment transfer
Coelho et *al*. (2012)	De	Fi	Sp	Co	-	Di-Mv	Si	He	N	Cer	-	Fn	Mpe	Adaptive large neighbourhood search heuristic
Bertazzi et *al*. (2013)	St	So	Sp	Co	Co	Mv	Si	He	N	Cer	-	Fn	Mpe	Hybrid rollout algorithm
Coelho and Laporte (2014)	De	Fi	Sp	Co	-	Mv	Mu	He	N	Cer	-	Fn	Mpe	Branch-and-cut algorithm.
Mirzaei and Seifi (2015)	De	Ls	Sp	Co	-	Mv	Uc	He	N	Cer	-	Fn	Mpe	Hybrid SA and TS
Soysal et *al*. (2015)	St	Bo	Sp	Co	-	Mv	Uc	He	Y	Cer	-	Fn	Mpe	Simulation model algorithm
Singh et *al*. (2015)	De-Fo	So	Mp		Co	Mv	Mu	He	N	Cer	Ha	Fn	Mpe	Incremental decomposing approach
Santos et *al*. (2016)	De	Fi	Sp	Co	-	Mv	Mu	Ho	N	Cer	-	Fn	Mpe	Hybrid heuristic-based local search
Chitsaz et *al*. (2016)	De	Fi	Sp	Co	-	Mv	Mu	Ho	Y	Cer	-	If	Spe	Two-phase heuristic approach
Soysal (2016)	St	Ls	Mp	Co	Co	Mv	Mu	Ho	Y	Cer	-	Fn	Mpe	Simulation approach
Soysal et *al*. (2018)	St	Bo	Mp	Co	Co	Mv	Mu	Ho	Y	Cer	-	Fn	Mpe	Chance-constrained programming approach
Cheng et *al*. (2017)	De	So	Sp	Co	-	Mv	Mu	He	Y	Cer	-	Fn	Mpe	Branch-and-cut algorithm
This research	De-Fo	Bo	Mp	Pv	Pv	Mv	Mu	He	Y	Ucr	Sf	Fn	Mpe	i) ASBA; ii) SBA; iii) ICA; iv) HGS; v) HPV

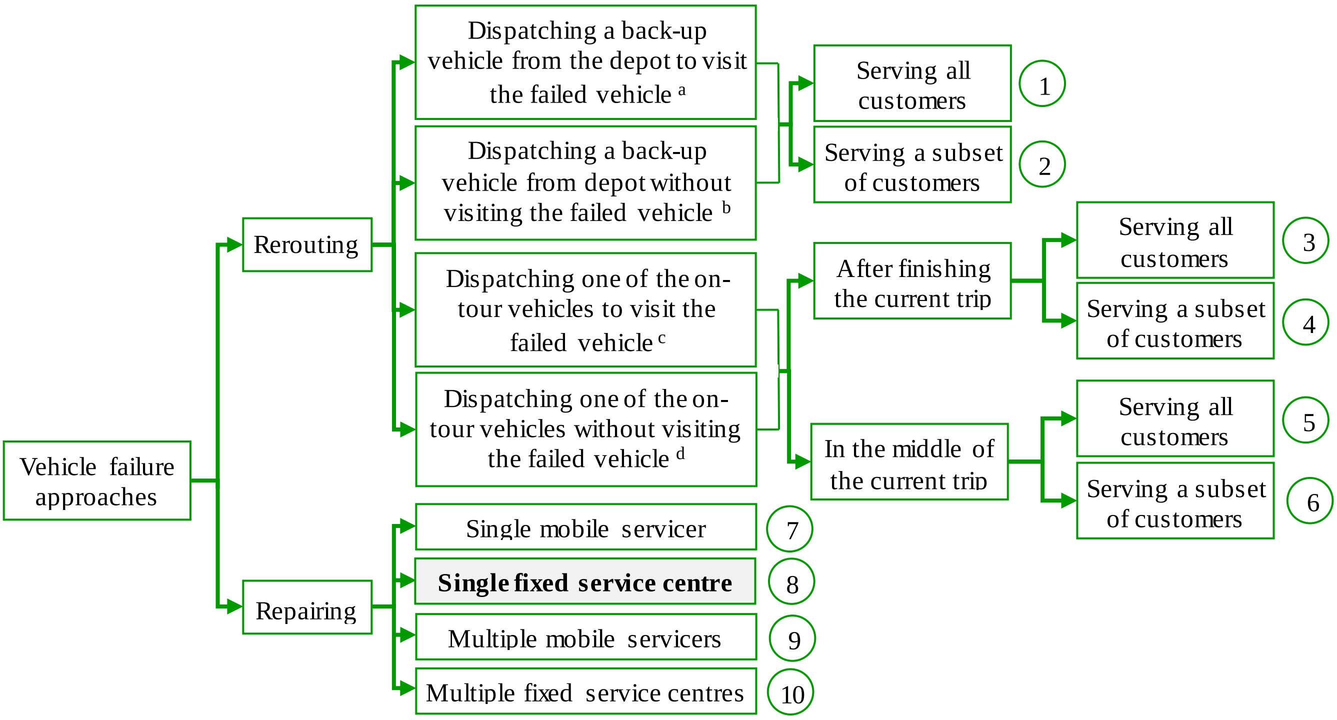
Note that we do not tend to make the problem unnecessarily complicated but we avoid simplifying assumptions. For instance, one may think that since it is assumed that all types of goods of a single retailer will be loaded together into the same vehicle, and the demands are predicted separately, this aspect does not really have any influence on the routing optimisation compared to the case with only one type of good (which may represent the aggregated demand of the retailer). This statement could be true when the variables do not interplay either in the objective function or in the constraints. Note that each product is treated individually in a multi-period setting (in which periods are interdependent) based on forecast demand. Then, demands for all products that belong to a retailer are aggregated to help plan for inventory-routing decisions in a single period. However, if we could ignore the multi-period assumption or exclude inventory holding and shortage costs (which are calculated based on forecasted demands for each product and in different periods), the aggregate, single product version of the problem does the same work.

2.2. Contribution

The contributions of the research in terms of problem definition and also methodology are explained below.

2.2.1. Vehicle failure in the IRP

The main contribution of the research is considering vehicle failure in the IRP. The concept of vehicle failure has already been investigated in various contexts such as (i) disruption in distribution networks and (ii) the VRP literature. However, to the best of the present researcher's knowledge (according to Table 2), vehicle failure has not yet been studied and applied in the IRP. We have visualised our own perception regarding various strategies of handling vehicle failure in Figure 2.


Vehicle failure approaches
Rerouting
Repairing
Dispatching a back-up vehicle from the depot to visit the failed vehicle [a]
Dispatching a back-up vehicle from depot without visiting the failed vehicle [b]
Dispatching one of the on-tour vehicles to visit the failed vehicle [c]
Dispatching one of the on-tour vehicles without visiting the failed vehicle [d]
Serving all customers
1
Serving a subset of customers
2
After finishing the current trip
Serving all customers
3
Serving a subset of customers
4
In the middle of the current trip
Serving all customers
5
Serving a subset of customers
6
Single mobile servicer
7
Single fixed service centre
8
Multiple mobile servicers
9
Multiple fixed service centres
10

Legend:
a: Availability of extra vehicles at the depot, customer-specific products
b: Availability of extra vehicles at the depot, non-customer-specific products
c: Unavailability of extra vehicles at the depot, customer-specific products
d: Unavailability of extra vehicles at the depot, non-customer-specific products

Figure 2. Vehicle failure approaches in distribution networks and the VRP studies.

According to Figure 2, there are ten approaches to choose from in order to handle the issue of vehicle failure. Rerouting (1 to 6) and repairing (7 to 10) are two main strategies which are briefly described as follows:

Rerouting: When a vehicle fails, unserved customers on the failed trip will be served by another vehicle, which is referred to as a "*backup vehicle*" (Li *et al.* 2007a; Li *et al.* 2007b; Zhang and Tang 2007; Li *et al.* 2008a; Li *et al.* 2009a; Li *et al.* 2009b; Wang *et al.* 2009; Yang and Wang 2009; Wang *et al.* 2010; Mu *et al.* 2011; Hu and Sun 2012; Minis *et al.* 2012; Ngai *et al.* 2012; Wang *et al.* 2012; Mamasis *et al.* 2013; Mu and Eglese 2013; Ahmadi and Seddighi 2013). This strategy makes vehicles in the system be rerouted. Rerouting can be either performed through dispatching a backup vehicle from the depot (approach 1) or dispatching one of the vehicles on tour (approach 2) to deliver the ordered products to the unserved customers. Moreover, in both approaches, the backup vehicle can load undelivered products either from the failed vehicle or the depot. Product types and the number of available vehicles play a significant role in selecting one of these approaches and visiting either the failed vehicle or depot for loading the undelivered products.

The types of shipping products in the literature include “customer-specific” and “non-customer-specific”. The first one refers to some specific cargoes ordered by some specific customers while the second refers to the common cargoes ordered by all customers (Eglese and Zambirinis 2018). In the case of customer-specific products, due to a lack of the products in the depot, the backup vehicle must move to the point of the failed vehicle to pick up its load before serving other customers (Li *et al.* 2008a; Minis *et al.* 2012). But in the case of non-customer-specific products, some extra products are available in the depot and it is not necessary for the backup vehicle to visit the failed vehicle to receive the undelivered products (Yang and Wang 2009; Mu *et al.* 2011; Mamasis *et al.* 2013). Additionally, availability of extra vehicles at the depot and time limitations could lead to dispatching the backup vehicle from the depot (Li *et al.* 2007a; Li *et al.* 2009a; Wang *et al.* 2010; Hu and Sun 2012) or the customers on the disrupted tour being served by one of the vehicles on their own trip (Yang and Wang 2009; Minis *et al.* 2012). In the case of dispatching one of the on-tour vehicles, in some situations, a backup vehicle could be rerouted right after finishing its current trip (Li *et al.* 2008; Mu *et al.* 2011); in the others, rerouting a vehicle is possible in the middle of its current trip (Yang and Wang 2009; Minis *et al.* 2012).

Finally, delivering to all of the unserved customers of the failed trip (Li *et al.* 2007a; Li *et al.* 2007b; Li *et al.* 2008a; Zhang and Tang 2007; Yang and Wang 2009; Mu *et al.* 2011; Ngai *et al.* 2012; Wang *et al.* 2012; Mu and Eglese 2013; Ahmadi and Seddighi 2013) or just serving some of them based on their priority (Li *et al.* 2009a; Li *et al.* 2009b; Wang *et al.* 2009; Wang *et al.* 2010; Hu and Sun 2012; Minis *et al.* 2012; Mamasis *et al.* 2013) determines which approach (1 to 6) should be considered (Eglese and Zambirinis 2018).

Repairing: In this approach, the failed vehicle is repaired; it then continues its planned trip. This approach is utilised for minor failures in which the service time is short. Repairing the failed vehicle could be done in two ways: (i) towing the failed vehicle to a fixed service centre (approaches 8 and 10); and (ii) the vehicle receives mobile repairing services at the failing point (approaches 7 and 9). Only Jbili *et al.* (2018) take this approach which investigates the failure of heavy vehicles travelling long distances between cities. In their study, the failed vehicle receives its repair service at the failing point (approach 7). Finally, approaches 7 and 8 can be developed with considering more than one mobile service (approach 9) or fixed service centre (approach 10).

Our research applies approach 8 to the IRP. We consider “fixed service centre strategy in vehicle failure” for vehicle failure. In other words, the failed vehicles are carried to a fixed service centre and repaired so that they can visit the remaining customers on their planned route.

2.2.2. Methodological

Since the research problem is NP-hard (Aghezzaf *et al.* 2006), developing heuristic and metaheuristic approaches to solve the IRP in real-world problems is a common approach. As mentioned in Table 2, a variety of heuristic and metaheuristic solutions have been applied to the IRP. In this research, for the first time, we develop a metaheuristic approach (i.e. ASBA with four scenarios) and compare it with four other benchmark metaheuristic algorithms (i.e. social-based algorithm (SBA), imperialist competitive algorithm (ICA), hybrid GA and SA (HGS) and hybrid PSO with VNS (HPV)) to solve the IRP. These algorithms are calibrated and their effectiveness and efficiency are evaluated. Our methodological contribution is not only applying ASBA to the IRP but also modifying the standard SBA as follows: We have embedded the "global war" procedure to the standard SBA that occurs after some iterations to reduce the chance of getting stuck in local optimum and premature convergence. The global war procedure stops the main loop of the algorithm, generates a new random population, and merges it with the old population. Then, the main loop starts again with a new population. Note that in order to analyse the effect of different initialisation processes in ASBA, four different policies are considered for ASBA as follows: ICA-based (IB), Classification-based (CB), Frog-based (FB) and Hybrid-ICA-frog-based (HIF).

3. PROBLEM DEFINITION AND MODELLING

In this section, we define the problem in detail. Then, the problem is modelled mathematically. The model is not solvable in reasonable times even by considering deterministic parameters. However, the model is useful because it helps us to:

(1) formulate details of constraints and show the complexity of the model (in terms of the number and types of variables and constraint) to justify why we need to design meta-heuristic techniques; and

(2) modify the mathematical model and accordingly design a lower bound for the problem to measure the quality of our designed meta-heuristic approaches.

The problem that we consider consists of a supplier as the depot and a set of geographically dispersed retailers as the customers of the distribution network. The problem is represented by a graph R(A, M) where A is the set of nodes and M is the set of arcs illustrating minimum-cost routes linking nodes in the network. Let 0 and N denote the depot and set of retailers, respectively. The retailers forecast the demand for multiple products for the next period according to past demands. The depot with a sufficiently large capacity serves the retailers through a heterogeneous fleet of vehicles; the capacity of each type of vehicle is given and all product types can be loaded to all types of vehicles. Dispatching each vehicle causes a fixed cost associated with the vehicle type; there is also a variable cost for distance travelled. Each retailer has a limited capacity to hold inventory and may suffer from shortage and holding costs (if the demand is underestimated or overestimated, respectively). The vehicles should meet the retailers during the determined soft time windows in the working hours.

In this problem, decisions regarding demand forecasting, dispatching the shipments and managing the vehicles throughout the distribution networks are made concurrently. The first two decisions (forecasting and dispatching) are the concomitant trials and tribulations in most distribution networks, but the third one (vehicle management) – as far as vehicle failure is concerned – is critical; this is particularly the case for unreliable transport systems including fleet and infrastructure. As we mentioned, the failed vehicles can be repaired in two possible ways: (a) a mobile servicer providing low-level services meets the down vehicles where they fail; or (b) the failed vehicles are carried to a fixed service centre that offers high-level services. In this research, we assume that the failed vehicles are repaired in the fixed service centre where high-level services are offered. Therefore, the service centre can repair the failed vehicles in less than a period. For example, an electrical problem leading to vehicle break down is repaired in 2 or 3 hours within a working day of 8 hours.

As illustrated in Figure 3, a vehicle failure may occur at point f with probability function $Pf\ (t)$ in each delivery tour; then it should be moved to the service centre. Then, the repaired vehicle continues to service the remaining retailers according to the previous sequence. The main goal of this problem is to find the optimal value of decision variables.

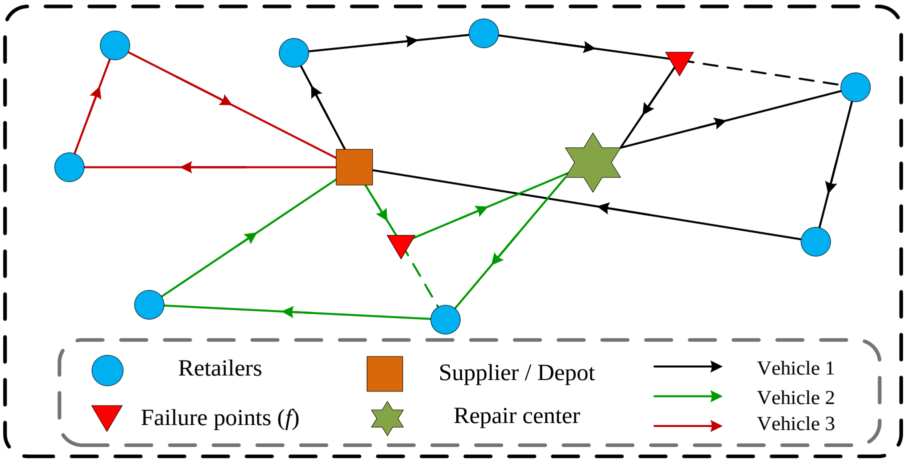


Figure 3. A graphic example of the IRP problem with vehicle failure in a distribution network during a specific period.

3.1. Assumptions

Before formulating the problem, the following assumptions are introduced:

- Since the number and location of retailers are known, the number and position of nodes are known and fixed.
- Each retailer must be visited at most once in each period by only one vehicle. As the demand of the retailers to the supplier is “less than truckload” (LTL) and the split of delivery is not allowed, the nodes should be visited no more than once.
- The retailers receive an independent certain demand in each period and they forecast the demand of multiple products for the next period according to past demands.
- The vehicles should arrive at the retailers’ location in specific time windows. If they arrive early or late, they will have to pay the penalty. Therefore, we use the soft time windows (Jia *et al.* 2014).
- The products are loaded together into the same vehicle. Although different products have different sizes and storage conditions, a standard pallet is considered for all products.
- Vehicle failure occurs no more than once in each tour and in each period; therefore, we assume that the vehicle failure pattern follows an exponential distribution. This distribution network is designed for urban distribution networks. Delivery should be done in a short time in networks like distributing dairy products, beverage, and refined oil.
- The vehicles periodically deliver the products to retailers (e.g. daily) and the vehicles can be failed with an increasing rate of a cumulative exponential distribution. The probability of failure has memory (i.e. it considers the time, which it has worked and by passing of time the failure probability increases). The possibility of failing more than once in a short period is approximately zero. Thus, we assume that vehicle failure does not occur more than once for each vehicle per period. The vehicles are repaired in the service centre during less than a period and the failed vehicles can continue their determined path after repair and meet remaining retailers according to previous sequences.
- Different vehicle capacities are considered and fleet composition is heterogeneous.
- Each driver receives a fixed schedule at the beginning of the day and has to meet the scheduled retailers. Therefore, each route (set of retailers) is served by one vehicle.
- The number of vehicles, vehicle dispatching cost/time, travelling cost/unit distance, holding cost, service time for each retailer, distances between network nodes, distances from network nodes to failure points, distances

from failure points to service centre, distances from service centre to network nodes, and vehicle speed are known. With respect to the location of the depot/supplier and retailers as well as some historical data, all of these parameters could be measured or calculated.

- Trips must begin and end at the supplier (i.e. depot). The main reason for this assumption is that the vehicles should be kept in their parking spaces at the depot and uploaded for the next period. Therefore, they should be returned to the supplier after delivering products to the retailers.
- Vehicles are available from the beginning of the day and we consider a working day as a period in this distribution network. Thus, the maximum available time for each vehicle is less than or equal to the working time per period (e.g. per day).

3.2. Notation

In order to formulate the problem, the following notation is used:

Sets and indices	***Description***
0	Index of the depot
M_k	Set of vehicles of type $k, m_k \in M_k, k \in K$
P	Set of periods $p \in P$
T	Set of time intervals $t \in T$ in each period
G	Set of product types $g \in G$
N	Set of retailers $i, j, i' \in N = \{1, 2, 3, ..., n\}$
A	Set of network nodes $A = \{0,1,2,3,...,n\} = \{N \cup \{0\}\}$
F	Set of failure points where the vehicles are broken down $f \in F$
$S_{m_k k}$	Any subset of retailers that vehicle m_k of type k meets $(S_{m_k k} \subset N)$
Parameters	
c_k	Average travel cost for distance unit by a vehicle of type $k \in K$
l_{ij}	Distance between node $i \in A$ and node $j \in A$
l'_{ij}	Distance between node $i \in A$ and the failure point which is located in link (i, j)
l''_{ij}	Distance from the failure point which is located in link (i, j) to service centre
l_{sj}	Distance between the service centre and node $j \in A$
f_{kp}	Fixed cost of the vehicle of type $k \in K$ in period $p \in P$
q_k	Capacity of vehicle type $k \in K$
ρ_k	Average travel cost/carrying cost from a failure point to the service centre for vehicle of type $k \in K$
Ψ_{kp}	Fixed cost of repairing the vehicle of type k in period $p \in P$
d_{igp}	Demand of retailer $i \in N$ for product $g \in G$ in period $p \in P$
$\hat{d}_{igp}$	Demand predicted by retailer $i \in N$ for product $g \in G$ in period $p \in P$, for initial period $\hat{d}_{ig0} = \hat{d}_{ig1}$
ς_{ig}	Target net stock at retailer $i \in N$ for product $g \in G$
ω_{ig}	Target inventory or inventory/work in progress (WIP) at retailer $i \in N$ for product $g \in G$
$v_{m_k ktp}$	Binary input variables, which presents that the vehicle $m_k \in M_k$ of type $k \in K$ may be prone to failure in period $p \in P$ and time $t \in T$ based on the cumulative distribution function (CDF)
$t'''_{m_k ktp}$	If vehicle $m_k \in M_k$ of type $k \in K$ in period $p \in P$ and time $t \in T$ may be prone to failure, $t'''_{m_k ktp}$ is equal to t; otherwise, it is equal to zero. based on the CDF
b_{igp}	Shortage cost per product unit at retailer $i \in N$ for product $g \in G$ in period $p \in P$
h_{igp}	Holding cost per product unit at retailer $i \in N$ for product $g \in G$ in period $p \in P$
φ_{ig}	Inventory capacity of retailer $i \in N$ for product $g \in G$
O_{igp}	The maximum capacity order of retailer $i \in N$ for product $g \in G$
$\gamma_{m_k kp}$	Average speed for moving a the failed vehicle $m_k \in M_k$ of type $k \in K$ to the service centre in period $p \in P$
$u_{m_k kp}$	Average speed for a vehicle $m_k \in M_k$ of type $k \in K$ to travel in period $p \in P$
$\vartheta_{im_k kp}$	Replenishment time for vehicle $m_k \in M_k$ of type $k \in K$ for serving retailer $i \in N$ in the period $p \in P$
$\phi_{m_k kp}$	Repair time of vehicle $m_k \in M_k$ of type $k \in K$ in the period $p \in P$
ϖ	Working time of a vehicle in a period (total time intervals in a period)
e_{ip}	The earliest arrival time of the time window for retailer $i \in N$ in period $p \in P$
τ_{ip}	The latest arrival time of the time window for retailer $i \in N$ in period $p \in P$
Γ_{ep}	Earliness penalty cost/ unit time in period $p \in P$

Γ_{lp}	Lateness penalty cost/ unit time in period $p \in P$
Γ'_{ip}	Time window violation penalty cost for retailer $i \in N$ in period $p \in P$
$\lambda_{m_k kp}$	Parameter of exponential distribution for the failure of vehicle $m_k \in M_k$ of type $k \in K$ in period $p \in P$
$\Delta_{m_k ktp}$	The CDF of the failure of vehicle $m_k \in M_k$ of type $k \in K$ at time $t \in T$ in period $p \in P$
$\eta_{m_k ktp}$	A random value that represents the possibility of the failure of vehicle $m_k \in M_k$ of type $k \in K$ at time $t \in T$ in period $p \in P$. In other word, the vehicle $m_k \in M_k$ of type $k \in K$ at time $t \in T$ in period $p \in P$ malfunctions when $\eta_{m_k ktp} \sim [0,1]$ is less than $\Delta_{m_k ktp}$ (i.e. CDF). In this study, $\eta_{m_k ktp}$ is generated through a continuous uniform distribution [0, 1].
ε	A small number which is less than one.

In the inventory part of the problem, although various replenishment policies are presented (Silver *et al.* 1998; Zipkin 2000), we should choose a policy to reduce disruptions such as demand variation and vehicle failure at a lower cost than holding extra inventory (Chen and Disney 2007). In this regard, the tendency of retailers to forecast orders based on previous demands leads to using the replenishment strategies known as the OUT policy (Cannella and Ciancimino 2010). In such a system, the inventory level equals on-hand inventory + WIP – backlog and is reviewed in each period. Consequently, an order quantity is determined to enhance the inventory level to a certain or base level. Accordingly, the amount of replenishment based on forecast orders is calculated by the retailers as follows:

$$o_{igp} = \hat{d}_{igp} + r_{1igp}(\varsigma_{ig} - s_{igp-1} + \pi_{igp-1}) + r_{2igp}(\omega_{ig} - w_{igp}) \qquad \forall i \in N, g \in G, p \in P \tag{1}$$

where $\hat{d}_{igp}$ is demand of retailer $i \in N$ for product $g \in G$ in period $p \in P$, which can be achieved in different ways, such as by single exponential smoothing method (Chen *et al.* 2000). r_{1ijp} and r_{2ijp} are constants to consider the effect of previous period inventory level and WIP to forecast the demand of product $g \in G$ in period $p \in P$, respectively. In other words, r_{1ijp} and r_{2ijp} are the correction factors for any shortfall of inventory and orders and goods in transit correction factor, which smooth the order level and reduce holding and shortage costs, respectively (Devika *et al.* 2016). The inventory status is equal to the net stock and WIP. Net stock is equal to on-hand inventory plus backlog. The WIP for retailer $i \in N$ for product $g \in G$ in period $p \in P$ is equal to o_{igp-1}. Two sets of variables including auxiliary and decision variables are defined as follows:

Auxiliary variables	***Description***
$\beta_{m_k ktp}$	Binary variable, which is 1 if vehicle $m_k \in M_k$ of type $k \in K$ fails at time t in period $p \in P$; 0 otherwise
s_{igp}	Net stock at retailer $i \in N$ for product $g \in G$ in period $p \in P$, for initial period $s_{ig0} = 0$
π_{igp}	Shortage at retailer $i \in N$ for product $g \in G$ in period $p \in P$, for initial period $\pi_{ig0} = 0$
o_{igp}	Amount of product $g \in G$ ordered by retailer $i \in N$ in period $p \in P$, for initial period, $o_{ig0} = \hat{d}_{ig0} = \hat{d}_{ig1}$
w_{igp}	The amount of WIP for retailer $i \in N$ for product $g \in G$ in period $p \in P$, $w_{igp} = o_{igp-1}$
$v'_{m_k ktp}$	Binary decision variables, which presents that the vehicle $m_k \in M_k$ of type $k \in K$ may be prone to failure in time interval $t \in T$ of period $p \in P$ based on routing decisions and $v_{m_k kp}$
α'_{jp}	Arrival time to retailer $j \in N$ in period $p \in P$
$y'_{m_k ktp}$	A virtual binary decision variable used to convert conditions (58) to some constraints (if $t'_{0m_k kp} \le t'''_{m_k ktp}$, then $y'_{m_k ktp} = 1$)
$y''_{m_k ktp}$	A virtual binary decision variable used to convert conditions (58) to some constraints (if $t'''_{m_k ktp} \le t''_{0m_k kp}$, then $y''_{m_k ktp} = 1$)
$w'_{jm_k kp}$	A virtual binary decision variable used to convert conditions (59) to some constraints (if $\sum_{t \in T} t'''_{m_k ktp} \beta_{m_k ktp} \le t''_{jm_k kp}$, then $w'_{jm_k kp} = 1$)

Decision variables	***Description***

$x_{ijm_k kp}$	Binary variable, which is 1 if vehicle $m_k \in M_k$ of type $k \in K$ moves from node $i \in A$ to node $j \in A$ in period $p \in P$; 0 otherwise
$x'_{ijm_k kp}$	Binary variable, which is 1 if vehicle $m_k \in M_k$ of type $k \in K$ fails in link (i, j) in period $p \in P$; 0 otherwise
$z_{im_k kp}$	Binary variable, which is 1 if vehicle $m_k \in M_k$ of type $k \in K$ retailer visits $i \in A$ in period $p \in P$; 0 otherwise
$y_{igm_k kp}$	Amount of product $g \in G$ delivered to retailer $i \in N$ by vehicle $m_k \in M_k$ of type $k \in K$ in period $p \in P$
$t'_{jm_k kp}$	The time interval that vehicle $m_k \in M_k$ of type $k \in K$ in period $p \in P$ departs retailers $j \in N$
$t''_{jm_k kp}$	The time interval that vehicle $m_k \in M_k$ of type $k \in K$ in period $p \in P$ arrives to node $j \in A$
r_{1igp}	A constant correcting error, which is based in the difference between the actual and desired stock decided by retailer $i \in N$ for product $g \in G$ in period $p \in P$
r_{2igp}	A constant correcting discrepancy between WIP and desired WIP by retailer $i \in A$ for product $g \in G$ in time period $p \in P$

3.3. Mathematical formulation

In this section, the mathematical model of the problem is presented. There will be terms in the objective function and two sets of constraints in the model that makes it non-linear. Moreover, the model contains binary variables.

$$\begin{aligned} z^* = Min \sum_{p\in P}\sum_{k\in K}\sum_{m_k\in M_k}\sum_{i\in A}\sum_{j\in A} x_{ijm_k kp}\left[\left(1-x'_{ijm_k kp}\right)c_k l_{ij} + x'_{ijm_k kp}\left(c_k l'_{ij} + \rho_k l''_{ij} + c_k ls_j\right)\right] + \\ \sum_{p\in P}\sum_{g\in G}\sum_{i\in N} h_{igp}(s_{igp} + o_{igp}) + \sum_{p\in P}\sum_{g\in G}\sum_{i\in N} b_{igp}\pi_{igp} + \\ \sum_{p\in P}\sum_{k\in K}\sum_{m_k\in M_k}\sum_{j\in N} f_{kp} x_{0jm_k kp} + \sum_{p\in P}\sum_{i\in N}\Gamma'_{ip} + \\ \sum_{p\in P}\sum_{k\in K}\sum_{m_k\in M_k}\sum_{t\in T}\beta_{m_k ktp}\psi_{kp} \end{aligned} \tag{2}$$

Subject to

$$\sum_{k\in K}\sum_{m_k\in M_k}\sum_{i\in A, i\neq j} x_{ijm_k kp} = 1 \qquad \forall j \in N, p \in P \tag{3}$$

$$\sum_{j\in A, i\neq j} x_{ijm_k kp} + \sum_{j\in A, i\neq j} x_{jim_k kp} = 2z_{im_k kp} \qquad \forall i \in A, m_k \in M_k, k \in K, p \in P \tag{4}$$

$$\sum_{i\in S_{m_k k}}\sum_{j\in S_{m_k k}, j\neq i} x_{ijm_k kp} \leq \sum_{i\in S_{m_k k}} z_{im_k kp} - z_{i' m_k kp} \qquad \forall S_{m_k k} \in N,\ m_k \in M_k,\ k \in K,\ p \in P,\ for\ some\ i' \in S_{m_k k} \tag{5}$$

$$\sum_{i\in N}\sum_{g\in G} y_{igm_k kp} \leq q_k \qquad \forall m_k \in M_k, k \in K, p \in P \tag{6}$$

$$\sum_{g\in G} y_{igm_k kp} \leq q_k z_{im_k kp} \qquad \forall i \in N, m_k \in M_k, k \in K, p \in P \tag{7}$$

$$o_{igp} = \hat{d}_{igp} + r_{1igp}(\varsigma_{ig} - s_{igp-1} + \pi_{igp-1}) + r_{2igp}(\omega_{ig} - o_{igp-1}) \qquad \forall i \in N, g \in G, p \in P \tag{8}$$

$$\hat{d}_{igp} = \alpha d_{igp-1} + (1-\alpha)\hat{d}_{igp-1} \qquad \forall i \in N, g \in G, p \in P \tag{9}$$

$$\sum_{k\in K}\sum_{m_k\in M_k} y_{igm_k kp} = o_{igp-1} \qquad \forall i \in N, g \in G, p \in P \tag{10}$$

$$o_{igp} \leq \mathrm{O}_{igp} \qquad \forall i \in N, g \in G, p \in P \tag{11}$$

$$s_{igp} = s_{igp-1} - \pi_{igp-1} + \pi_{igp} + o_{igp-1} - d_{igp} \qquad \forall i \in N, g \in G, p \in P \tag{12}$$

$$s_{igp} + o_{igp-1} \leq \varphi_{ig} \qquad \forall i \in N, g \in G, p \in P \tag{13}$$

$$t'_{jm_k kp} = \left(t'_{jm_k kp} + \frac{l_{ij}}{u_{m_k kp}} + \vartheta_{jm_k kp}\right) x_{ijm_k kp} \qquad \forall i \in A, j \in N, m_k \in M_k, k \in K, p \in P \tag{14}$$

$$t''_{jm_k kp} = \left(t'_{jm_k kp} + \frac{l_{ij}}{u_{m_k kp}}\right) x_{ijm_k kp} \qquad \forall i \in A, j \in A, m_k \in M_k, k \in K, p \in P \tag{15}$$

$$y'_{m_k ktp}(t'''_{m_k ktp} + \varepsilon - t'_{0m_k kp}) \geq 0 \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{16}$$

$$(1-y'_{m_k ktp})(t'''_{m_k ktp}+\varepsilon-t'_{0m_k kp})\leq 0 \qquad \forall m_k \in M_k, k\in K, t\in T, p\in P \tag{17}$$

$$y''_{m_k ktp}(t''_{0m_k kp}-t'''_{m_k ktp}-\varepsilon)\geq 0 \qquad \forall m_k \in M_k, k\in K, t\in T, p\in P \tag{18}$$

$$(1-y''_{m_k ktp})(t''_{0m_k kp}-t'''_{m_k ktp}-\varepsilon)\leq 0 \qquad \forall m_k \in M_k, k\in K, t\in T, p\in P \tag{19}$$

$$v'_{m_k ktp}=y''_{m_k ktp}y'_{m_k ktp} \qquad \forall m_k \in M_k, k\in K, t\in T, p\in P \tag{20}$$

$$\sum_{t\in T}\beta_{m_k ktp}\leq 1 \qquad \forall m_k \in M_k, k\in K, p\in P \tag{21}$$

$$M\left(\sum_{t\in T}\beta_{m_k ktp}\right)\geq\sum_{t\in T}v'_{m_k ktp} \qquad \forall m_k \in M_k, k\in K, p\in P \tag{22}$$

$$w'_{jm_k kp}\left(t''_{jm_k kp}+\varepsilon-\sum_{t\in T}t'''_{m_k ktp}\beta_{m_k ktp}\right)\geq 0 \qquad \forall j\in A, m_k \in M_k, k\in K, p\in P \tag{23}$$

$$(1-w'_{jm_k kp})\left(t''_{jm_k kp}+\varepsilon-\sum_{t\in T}t'''_{m_k ktp}\beta_{m_k ktp}\right)\leq 0 \qquad \forall j\in A, m_k \in M_k, k\in K, p\in P \tag{24}$$

$$x'_{ijm_k kp}w'_{jm_k kp}\left(\sum_{t\in T}t'''_{m_k ktp}\beta_{m_k ktp}-t'_{im_k kp}x_{ijm_k kp}\right)\geq 0 \qquad \forall i, j\in A, m_k \in M_k, k\in K, p\in P \tag{25}$$

$$(1-x'_{ijm_k kp})w'_{jm_k kp}\left(\sum_{t\in T}t'''_{m_k ktp}\beta_{m_k ktp}-t'_{im_k kp}x_{ijm_k kp}\right)\leq 0 \qquad \forall i, j\in A, m_k \in M_k, k\in K, p\in P \tag{26}$$

$$\begin{aligned}\alpha_{jp}={}&\sum_{m_k\in M_k}\sum_{k\in K}x_{0jm_k kp}t'_{0m_k kp}+\sum_{m_k\in M_k}\sum_{k\in K}\sum_{i\in N}x_{ijm_k kp}\left(\vartheta_{im_k kp}+\alpha_{ip}\right)+\sum_{m_k\in M_k}\sum_{k\in K}\sum_{i\in N}x_{ijm_k kp}\left(1-x'_{ijm_k kp}\right)\left(\frac{l_{ij}}{u_{m_k kp}}\right)+\\&\sum_{m_k\in M_k}\sum_{k\in K}\sum_{i\in N}x_{ijm_k kp}x'_{ijm_k kp}\left(\frac{l'_{ij}}{u_{m_k kp}}+\frac{l''_{ij}}{\gamma_{m_k kp}}+\phi_{m_k kp}+\frac{ls_j}{u_{m_k kp}}\right)\qquad \forall j\in N, p\in P\end{aligned} \tag{27}$$

$$\begin{aligned}&\sum_{j\in N}x_{j0m_k kp}\alpha_{jp}+\sum_{j\in N}x_{j0m_k kp}\left(1-x'_{j0m_k kp}\right)\left(\frac{l_{j0}}{u_{m_k kp}}\right)+\\&\sum_{j\in N}x_{j0m_k kp}x'_{j0m_k kp}\left(\frac{l'_{j0}}{u_{m_k kp}}+\frac{l''_{j0}}{\gamma_{m_k kp}}+\phi_{m_k kp}+\frac{ls_0}{u_{m_k kp}}\right)\leq\varpi \qquad \forall m_k \in M_k, k\in K, p\in P\end{aligned} \tag{28}$$

$$(\alpha_{ip}-\tau_{ip})\xi_{ip}\geq 0 \qquad \forall i\in N, p\in P \tag{29}$$

$$(\alpha_{ip}-\tau_{ip})(1-\xi_{ip})\leq 0 \qquad \forall i\in N, p\in P \tag{30}$$

$$(e_{ip}-\alpha_{ip})\zeta_{ip}\geq 0 \qquad \forall i\in N, p\in P \tag{31}$$

$$(e_{ip}-\alpha_{ip})(1-\zeta_{ip})\leq 0 \qquad \forall i\in N, p\in P \tag{32}$$

$$\Gamma'_{ip}=\Gamma_{ep}(e_{ip}-\alpha_{ip})(\zeta_{ip})+\Gamma_{lp}(\alpha_{ip}-\tau_{ip})(\xi_{ip}) \qquad \forall i\in N, p\in P \tag{33}$$

$$x_{ijm_k kp}\in\{0,1\} \qquad \forall (i,j)\in A, m_k \in M_k, k\in K, p\in P \tag{34}$$

$$z_{im_k kp}\in\{0,1\} \qquad \forall i\in A, m_k \in M_k, k\in K, p\in P \tag{35}$$

$$x'_{ijm_k kp}\in\{0,1\} \qquad \forall (i,j)\in A, m_k \in M_k, k\in K, p\in P \tag{36}$$

$$v'_{m_k ktp}\in\{0,1\} \qquad \forall m_k \in M_k, k\in K, t\in T, p\in P \tag{37}$$

$$w'_{jm_k kp}\in\{0,1\} \qquad \forall j\in A, m_k \in M_k, k\in K, p\in P \tag{38}$$

$$y'_{m_k ktp}\in\{0,1\} \qquad \forall m_k \in M_k, k\in K, t\in T, p\in P \tag{39}$$

$$y''_{m_k ktp} \in \{0,1\} \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{40}$$

$$\beta_{m_k ktp} \in \{0,1\} \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{41}$$

$$t'_{jm_k kp} \geq 0 \qquad \forall j \in A, m_k \in M_k, k \in K, p \in P \tag{42}$$

$$t''_{jm_k kp} \geq 0 \qquad \forall j \in A, m_k \in M_k, k \in K, p \in P \tag{43}$$

$$\zeta_{ip} \in \{0,1\} \qquad \forall i \in N, p \in P \tag{44}$$

$$\xi_{ip} \in \{0,1\} \qquad \forall i \in N, p \in P \tag{45}$$

$$s_{igp} \geq 0 \qquad \forall i \in N, g \in G, p \in P \tag{46}$$

$$\pi_{igp} \geq 0 \qquad \forall i \in N, g \in G, p \in P \tag{47}$$

$$o_{igp} \text{ and } \hat{d}_{igp} \geq 0 \qquad \forall i \in N, g \in G, p \in P \tag{48}$$

$$w_{igp} \geq 0 \qquad \forall i \in N, g \in G, p \in P \tag{49}$$

$$\alpha_{ip} \geq 0 \qquad \forall i \in N, g \in G, p \in P \tag{50}$$

$$y_{igm_k kp} \geq 0 \qquad \forall i \in N, g \in G, m_k \in M_k, k \in K, p \in P \tag{51}$$

$$0 \leq r_{1igp} \leq 1 \qquad \forall i \in N, g \in G, p \in P \tag{52}$$

$$0 \leq r_{2igp} \leq 1 \qquad \forall i \in N, g \in G, p \in P \tag{53}$$

In the objective function (2) the first term is the transport costs, which is comprised of two parts. The first part computes total transport costs related to passing no failed links. The second part calculates the moving costs of the failed vehicles from the beginning node of the failed link to the failure point and from the service centre to the next retailer and carrying costs from the failure point to the service centre. The carrying costs are calculated as ρ_k (average travel cost / carrying cost from a failure point to the service centre for the vehicle type $k \in K$) multiplied by l''_{ij} (the distance between failure point to service centre). The moving costs of the failed vehicle (i) from the beginning node of the failure link to the failure point and (ii) from the service centre to the next retailer are calculated. The calculation is based on the distance from network nodes to potential vehicle points in network (l'_{ij}) and surplus distance, which a vehicle (after being repaired in the service centre) should pass to the next retailer (l_{sj}). The next two terms denote inventory and backlogging costs, respectively. The forth summation calculates the fixed cost of each tour. The fifth term deals with the time window violation penalty cost. The last summation stands for the repair costs in service centres.

Constraints (3) ensure that each retailer must be visited only one time by all the vehicles and vehicle types in each time period. Constraints (4) and (5) are degree constraints and sub-tour elimination constraints, respectively. Furthermore, constraints (5) ensure that all routes should be started and ended from/to the depot. Constraints (6) force the capacity limit for the vehicles. Constraints (7) represent that if vehicle $m_k \in M_k$ of type $k \in K$ never visit retailer $i \in N$, the vehicle cannot have any delivery to the retailer; otherwise the vehicle's capacity must be observed. Constraints (8) calculate the order value, and constraints (9) forecast the demand of product $g \in G$ for the next period. Constraints (10) impose that the delivered product $g \in G$ to retailer $i \in N$ should be equal to the order of the customer registered in the previous period. Constraints (11) limit the customers' order capacity. Constraints (12) are about the inventory balance imposed on the net stock and shortage of retailers in each period. Constraints (13) ensure that the warehouse capacity of retailers is considered. Constraints (14) find the departure time of retailers in each tour. Moreover, constraints (15) calculate the arrival time to network nodes in each tour. Note that we use a discrete time index t in the model. Then, we use continuous functions for vehicle failures. The model checks this according to the time set T ($t \in T$). In other words, we discretise the continuous time using the set T and only check the vehicle failures at these times. Otherwise, on a continuous time, there would be infinitely many index t, which could not be solvable. According to the value of cumulative exponential distribution ($\Delta_{m_k ktp}$) and the random value ($\eta_{m_k ktp}$) for each time interval $t \in T$, time period $p \in P$ and vehicle $m_k \in M_k$ of type $k \in K$, the prone situations ($\nu_{m_k ktp}$) of vehicle failure are calculated. Conditions (54-56) mathematically demonstrate the process of finding these situations.

$$\Delta_{m_k ktp} = 1 - e^{-\lambda_{m_k kp} t} \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{54}$$

$$\nu_{m_k ktp}(\Delta_{m_k ktp} - \eta_{m_k ktp}) \ge 0 \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{55}$$

$$(1 - \nu_{m_k ktp})(\Delta_{m_k ktp} - \eta_{m_k ktp}) \le 0 \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{56}$$

$$t\nu_{m_k ktp} = t'''_{m_k ktp} \qquad \forall m_k \in M_k, k \in K, t \in T, p \in P \tag{57}$$

Constraints (55)-(56) impose that a vehicle $m_k \in M_k$ of type $k \in K$ can be failed only if the probability of the vehicle failure exceeds the failure threshold. An additional binary decision variable $\nu_{m_k ktp}$ is presented so that the vehicle $m_k \in M_k$ of type $k \in K$ may be prone to failure in period $p \in P$. Constraints (57) represent the time interval that vehicle $m_k \in M_k$ of type $k \in K$ may be prone to failure in period $p \in P$. According to conditions (58), a vehicle is prone to fail based on routing decisions and $\nu_{m_k ktp}$. In this regard, if the computed time interval for vehicle $m_k \in M_k$ of type $k \in K$ in period $p \in P$, which was prone to fail based on cumulative exponential distribution, is between its departure and arrival time from/ to depot, then the vehicle is prone to fail based on routing decisions.

$$if \quad t'_{0m_k kp} \le t'''_{m_k ktp} + \varepsilon \le t''_{0m_k kp} \quad then \quad v'_{m_k ktp} = 1 \qquad \forall\ m_k \in M_k, k \in K, t \in T, p \in P \tag{58}$$

Constraints (16)-(20) present the modified equivalent of Conditions (58). Constraints (21) ensure that no more than one vehicle failure can occur in a tour. Additionally, constraints (22) indicate that if one or more prone vehicle failure points exist in a tour, one of them must be selected as a vehicle failure point.

Conditions (59) and (60) help to find the location of vehicle failure points in all time periods. It is noted that we just know the tours which have a failure points $(\beta_{m_k ktp})$ and the time interval that the vehicle failure was occurred ($t'''_{m_k ktp}$). Accordingly, Conditions (59) identify the network nodes in the corresponding tour in which arrival time to them (which is computed without assuming failure point based on constraints (15)) is after occurring the vehicle failure point. Therefore, this node can be the end node of the link having a vehicle failure $(w'_{jm_k kp} = 1)$. Then, based on the nodes found through Conditions (59), Conditions (60) identify the link in the corresponding tour, which, terminating to one of the found nodes and the departure time from its originating node (which is computed without assuming failure point based on constraints (14)), is less than the time interval that the vehicle failure point occurs at. In this regard, constraints (23)-(26) present the modified equivalent of conditions (59) and (60).

$$if \quad \sum_{t \in T} t'''_{m_k ktp}\beta_{m_k ktp} \le t''_{jm_k kp} + \varepsilon \quad then \quad w'_{jm_k kp} = 1 \qquad \forall j \in A, m_k \in M_k, k \in K, p \in P \tag{59}$$

$$if \quad x_{ijm_k kp} w'_{jm_k kp} t'_{im_k kp} \le \sum_{t \in T} t'''_{m_k ktp}\beta_{m_k ktp} \quad then \quad x'_{ijm_k kp} = 1 \qquad \forall i, j \in A, m_k \in M_k, k \in K, p \in P \tag{60}$$

Constraints (27) calculate the arrival time to retailers considering vehicle failures. These constraints are comprised of four different terms. The first and second terms represent the arrival time to the exact prior node of retailer j which was visited. If this prior node is the depot, then the first term is used. Otherwise, the second term must be used. The second term considers the replenishment time because it is a retailer node rather than a depot. The third term is used if the link reaching to retailer j is a sound link ($x'_{ijm_k kp} = 0$) and calculates the travel time on this link. Finally, the forth term is designed to calculate the travel time on the link if it has a failure point. Hence, in the fourth term, the travel time from beginning node to failure point, travel time from failure point to service centre, repair time in service centre, and travel time from service centre to retailer j must be considered instead of travel time on the link. Constraints (28) illustrate that the working time assigned to vehicle k should not exceed the limit. According to this equation, the vehicles should come back to the depot during working time. In this regard, the first term of constraints (28) finds total travel time to the last retailer in each tour, and the second and third terms compute travel time between the last retailer and the depot. Constraints (29)-(33), a modified form of Condition (61), apply the time window violation penalty cost incurring when the arrival time to a node does not meet the considered time window. In other words, constraints (29)-(32) present the position of arrival time to node j in its related time windows, and constraints (33) calculate the violation penalty cost. Binary decision variables ξ_{ip} are added to transform the if-then conditions (61) to constraints (29)-(33). Constraints (34)-(53) impose the type of variables.

$$\Gamma'_{ip} = \begin{cases} \Gamma_{ep}(e_{ip} - \alpha_{ip}), & \text{if} \quad \alpha_{ip} < e_{ip} \\ 0, & \text{if} \ e_{ip} \le \alpha_{ip} \le \tau_{ip} \\ \Gamma_{lp}(\alpha_{ip} - \tau_{ip}) & \text{if} \ \tau_{ip} < \alpha_{ip} \end{cases} \qquad \forall i \in N, p \in P \qquad (61)$$

There are non-linear relations such as objective function, constraints (8), (15)-(20) and (23)-(33); some of them cannot be linearised. Of course, the mathematical model can be solved on a commercial optimisation software product to find a local optimum solution (not global) but this cannot necessarily provide a reliable upper bound for the objective function.

Another issue is that, we use a random parameter generated from a uniform distribution, and by comparing this value with the cumulative probability function of exponential distribution, we decide whether a failure is going to happen or not. However, the random parameters again need to be generated before the model is solved and the result of the model will depend heavily on the generated random parameter.

We suggest an approach for finding the best solution according to the amount of expected risk (probability × consequence) that a decision maker can tolerate. Our definition of *probability* is the chance of having some vehicle failures equal to or greater than a specific number along the considered time period. The probability can be extracted from related historical data or estimated with a distribution probability function. For instance, Figure 4 shows the corresponding probability in a network. Our definition of *consequence* is the difference between the objective of the worst-case scenario and the objective of the corresponding scenario (in terms of total cost). In order to compute the consequence of different scenarios, some steps should be taken. At the first step of calculating the consequence, the decision maker generates many random parameters and solves the model many times. Obviously, every time the model is solved, it may return a different solution depending on the random parameter used. At the second step, in addition to the different solutions, the decision maker needs to determine the most optimistic and pessimistic solutions. The most optimistic solution will occur when no vehicle failures happen in all periods. By assuming that all the random parameters are equal to one, the most optimistic solution will appear.

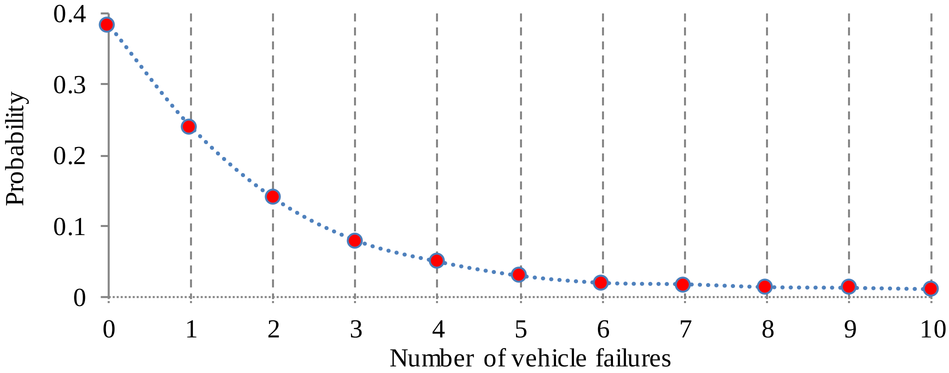


Figure 4. Probabilities of having a number of vehicle failures in network.

On the other hand, the most pessimistic solution is the solution that all used vehicles in all periods experience a vehicle failure in a location which has the furthest distance from the service centre. If all the random parameters equal zero, all of the used vehicles will have a vehicle failure and the most pessimistic solution will be obtained. At the third step, by subtracting the amount of each solution's objective of the pessimistic solution's objective, the consequence of them will be resulted. Because each solution has a discrete number of vehicle failures, apart from having no vehicle failure in the network (optimistic solution), a range of consequences may occur for any number of vehicle failures. For instance, Figure 5 shows the consequences of the solutions obtained for different numbers of vehicle failures. Finally, the decision maker can calculate their expected risk by multiplying the consequence of the solution and the probability of the solution. Figure 6 shows the expected risk of all solutions.

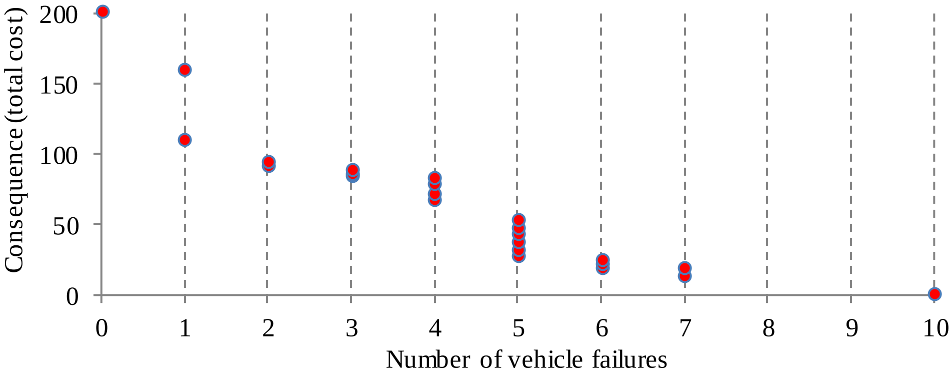


Figure 5. Consequences of having a number of vehicle failures in network.

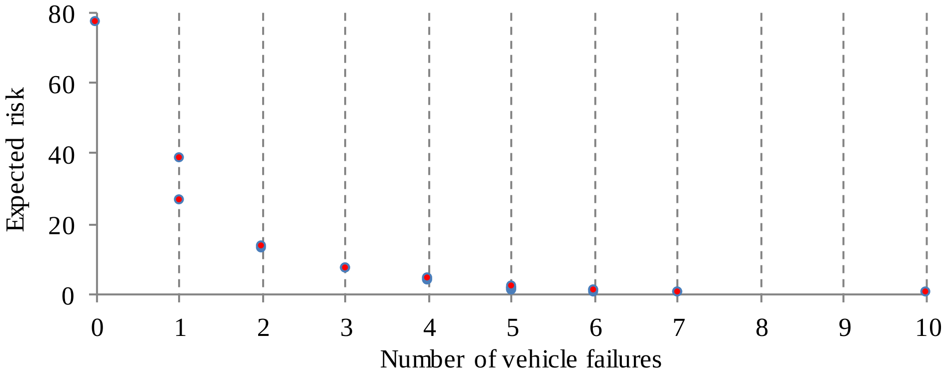


Figure 6. Risks of having a number of vehicle failures in network.

In addition to our proposed approach for finding the best solution in an uncertain model, there are some other techniques to tackle such problems. We also suggest robust optimisation introduced by Mulvey *et al.* (1995) to be used in large-scale systems. Interested readers may refer to Mulvery *et al.* (1995), Leung *et al.* (2007) and Yu and

Li (2000) to see a brief description of this technique. There are several other possible approaches proposed to handle the probabilistic parameters. Some of them are as follows:

1) The scenario-based optimisation method is based on some random instances of the uncertain parameter(s) to find the optimal solution where only the constraints are associated with uncertainty (Schildbach and Morari 2016);
2) Monte Carlo Simulation runs as an estimate of an actual system as one of the most popular tools to analyse uncertainty aspects (Fu 2002);
3) Simulation-based optimisation methods provide the flexibility to accommodate arbitrary stochastic parameters (Wan *et al.* 2005);
4) Fuzzy mathematical programming approaches are classified into two major groups: (i) possibilistic programming used when there is a lack of knowledge about the exact values of the model parameters (Inuiguchi and Ramik 2000); and (ii) flexible programming applied to cope with flexible target value of goals and constraints (Mula *et al.* 2006).

In terms of risk attitude, we assume a risk-neutral decision maker but the model has also the ability to produce solutions for risk averse and risk seeking decision makers. Accordingly, we solve the problems only for a known set of random parameters to generate an optimal solution for a risk-natural decision maker.

4. PROPOSED SOLUTION ALGORITHMS

According to Aghezzaf *et al.* (2006), the IRP belongs to the class of nondeterministic polynomial time (NP) problem. Considering the vehicle failure and the time windows issues in the IRP model, the formulated problem is computationally complex. Therefore, the heuristic approaches to solve these types of large-scale problems inevitably become the only possible alternative (Nguyen *et al.* 2012). Moreover, the model is nonlinear and nonconvex.

We develop four algorithms based on an adapted social-based metaheuristic approach that combines the evolutionary algorithm (EA) and socio-political process-based methods (Ramezani and Lotfi 2013). We have adapted a traditional SBA with some considerable changes and introduced it as a metaheuristic named adapted SBA (ASBA). To improve the performance of ASBA in the proposed problem, four scenarios are considered regarding its construction.

We show that our developed metaheuristic algorithms obtain near-optimal solutions within reasonable time. The algorithms minimise the total cost through: (i) determining the best routes that vehicles should pass through; (ii) directing vehicles with high failure potential to the service centre via the near passes; (iii) determining the best inventory level in retailers' stores in order to find the best trade-off between storage and backlogging costs; and (iv) identifying the sequence of retailers to be served by vehicles within the given time windows. In the following subsections, ASBA as a novel approach is presented comprehensively.

4.1. Encoding scheme and vehicle failure-related calculation

As this section is the same in all algorithms, we added the vehicle failure-related procedure in this section. In the proposed model, there are two types of variables: auxiliary and decision variables. The auxiliary variables are used for arrival time, distance between failure point and node, distance between failure point and service centre, net stock and shortage at the retailers, distance travelled by a vehicle and distance travelled by the failed vehicle. Binary variables are used for presenting the vehicle failures calculated in the solution process. The decision variables should be considered in the structure of the solution. Figure 7 presents a schematic illustration of the structure of the encoding process in each period.

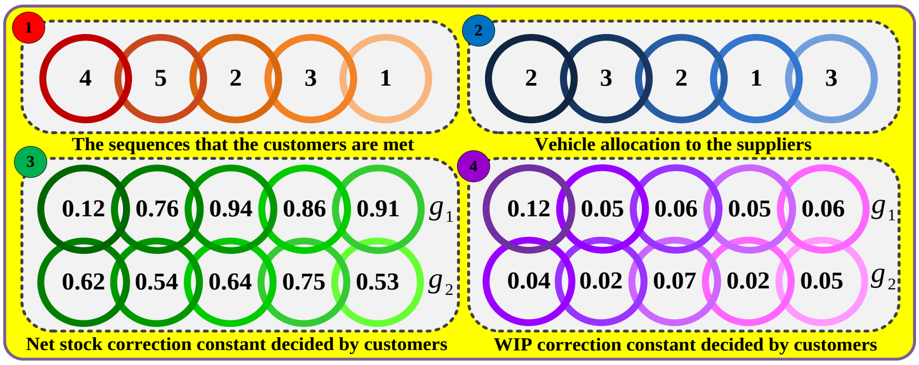

1
4
5
2
3
1
The sequences that the customers are met
2
2
3
2
1
3
Vehicle allocation to the suppliers
3
0.12
0.76
0.94
0.86
0.91
g_1
0.62
0.54
0.64
0.75
0.53
g_2
Net stock correction constant decided by customers
4
0.12
0.05
0.06
0.05
0.06
g_1
0.04
0.02
0.07
0.02
0.05
g_2
WIP correction constant decided by customers

Figure 7. A graphical representation of the structure of encoding process in a period.

The first part is related to the sequence of the retailers to be met by the vehicles covering the standard decision variables in the VRP (Homberger and Gehring 1999) and IRP (Moin *et al.* 2011). In other words, the related decision variables are $x_{ijm_k kp}$, $z_{im_k kp}$ and $y_{igm_k kp}$, respectively. The second part is used to assign the retailers to the vehicles and is related to multiple-vehicle issue (Govindan *et al.* 2014). Each number in the second section represents a vehicle that meets the retailers. For instance, the second vehicle serves the first and third retailers. Accordingly, a matrix with $|TNV|$ elements (TNV is the total number of vehicles from different types) is constructed. Each number represents a related vehicle. The third and fourth parts indicate the net stock and WIP correlation constants (i.e. r_{1ijp} and r_{2ijp}, respectively) determined by each retailer for each product in each time period.

To enable the continuous metaheuristic algorithms such as PSO to solve this discrete problem, the random-key (RK) technique is applied. An encoding scheme, RK has been frequently used in solving discrete problems with continuous approaches (Chang *et al.* 2009; Tavakkoli-Moghaddam *et al.* 2009). RK is applied to parse the primary solution. Accordingly, as illustrated in Figure 8, a matrix of $|N|$ elements, each from a uniform distribution $U(0,1)$ is generated so that the RK demonstrates the sequence of retailers that should be met. For instance, the encoded solution (1) {0.78, 0.85, 0.38, 0.49, 0.23} represents the parsed solution (2) {4, 5, 2, 3, 1} calculated by sorting the encoded solution. In the second sub-solution, the vehicles are assigned to retailers. Accordingly, a matrix with $|N|$ elements each from a uniform distribution $U(1,TNV)$ is generated where the RK illustrates the desired vehicle for each retailer. For example, the encoded solution (3) ({2.32, 2.78, 1.85, 1.07, 2.61}) represents the parsed solution (4) ({2, 3, 2, 1, 3}) acquired by rounding the encoded solution.

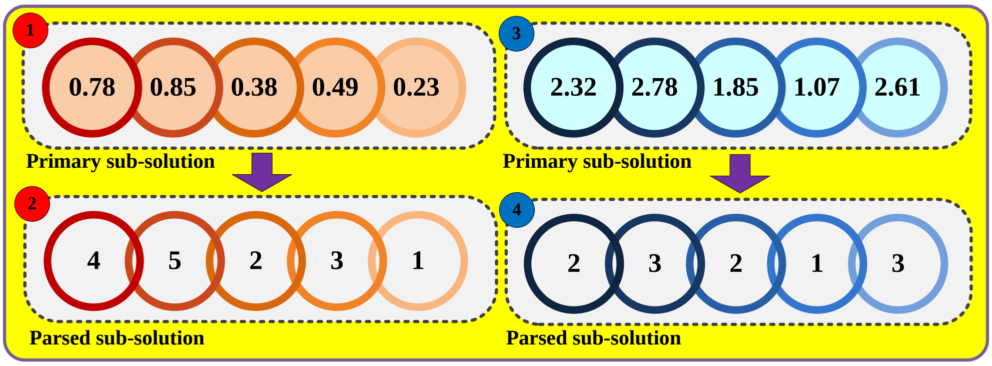


Figure 8. A schematic illustration of the RK technique in the first and second sub-solutions.

Vehicle failure imposes additional repair and carrying costs on the total cost and also increases the working time of the failed vehicles. The developed algorithms calculate the mentioned added costs and times through "vehicles failure-related calculation procedure". As this procedure is performed in all developed methods, it is presented in this section. The pseudo-code of vehicles' failure-related calculation procedure is presented as Figure A1 (Appendix). When the $\Delta_{m_k ktp}$ (as CDF of vehicle failure) exceeds $\eta_{m_k ktp}$ (which is generated through a continuous uniform distribution [0, 1]), the vehicle may fail at time t if it has not failed in previous time intervals. $\Delta_{m_k ktp}$ is a strictly increasing function; i.e. the probability of vehicle failure increases over time in each period and sets to zero again for the first time interval of the next period. In this regard, the algorithm finds the failure location (corresponding link has a failure point) at time t according to the arrival and departure time to/from network nodes in the corresponding tour. Then, the distances between the selected failure point and prior node, the failure point and service centre as well as the service centre and the next node are calculated. Finally, the total moving and carrying costs and also repair costs of the failed vehicle are calculated.

4.2. Initialisation procedure

The initialisation process is used to generate the *nPop* sample solutions as the first population. In this paper, the initialisation process introduced by Liu and Lee (2011) is used to randomly generate the initial solutions from the feasible space. According to our primary experiments, this initialisation process can reduce the computational time of different methods. The procedure of determining the initial solution is presented in Figure A2 (Appendix). This initialisation process is constructed based on the assimilation concept known as vehicle load assimilation (VLA). In this regard, first of all, the required number of vehicles to satisfy the total received orders is calculated and then the received orders are assigned to the selected vehicles so that the vehicles carry the load with the same capacity utilisation rate. Therefore, the VLA provides an initial solution so that the retailers are assigned to vehicles based on the concept of assimilation; the rest of the decision variables used in the model are generated randomly.

4.3. Adapted social-based algorithm (ASBA)

The social-based algorithm (SBA) was first introduced by Ramezani and Lotfi (2013) as a continuous algorithm. This method was constructed based on an evolutionary algorithm (EA) and socio-political process represented in an ICA. Although the EAs have been known as popular algorithms for their many interesting properties and have been widely applied in a variety of optimisation problems (Back *et al.* 1997), they might be incapable of obtaining an optimal solution in some problems (Grosan and Abraham 2007). Accordingly, the hybridisation of EAs with other algorithms as part of a larger system may lead to a very powerful search algorithm, which is able to handle the variety of problems in finding more qualified solutions (Grosan and Abraham 2007; Ramezani and Lotfi 2013).

ICA, which was introduced by Atashpaz-Gargari and Lucas (2007), is considered one of the cutting-edge EAs in the field of evolutionary computation. The evolutionary optimisation strategy used in ICA has been revealed as high performance in many NP-hard problems because of two aspects: (i) its convergence rate; and (ii) obtaining the global optima achievement (Aghezzaf *et al.* 2006). The algorithm benefits from the relationship between empires and colonies and imperialistic competition among these empires to finally converge to a state in which there exists only one empire as the best solution (Mozafari *et al.* 2012).

Obviously, EA and ICA have different optimisation perspectives. Therefore, the combination of these two algorithms can provide us with a more powerful method (i.e. SBA) as a novel hybrid algorithm that benefits from the eligibility of both algorithms. The concept of person in SBA is similar to the chromosome concept in GA terminology and the particles in the PSO method, which is actually an array of candidate solutions.

4.3.1. Social communities in SBA construction

In a human society, people from different walks of life can evolve a community. They construct different communities in their countries. Each person in considered an *N*-dimensional optimisation problem (i.e., 1×*N* array) as follows:

$$Person_i = \overline{p}_i = [p_{i1}, p_{i2}, \dots, p_{iN}] \tag{62}$$

where p_i is the variable that should be optimised. The cost of a person is calculated using a cost function of the variables $\overline{p}_i$ as $c_i = \mathrm{Cost}(person_i) = \mathrm{Cost}(\overline{p}_i)$. In this algorithm, four types of community are used:

- ***Republic***, where the best person is chosen as the president for a certain period of time;
- ***Autocracy***, where a person without any credibility is selected as the leader whom people do not try to obey;
- ***Monarchy***, where the powerful person is considered as the monarch whom the people are obliged to follow; and
- ***Multinational*** where communities are constructed based on a symbiotic relationship of countries that cooperate and transact with each other. Interested readers seeking more details of SBA can refer to Ramezani and Lotfi (2013). After some iteration, a global war iteratively occurs for the preservation of premature convergence. In the end, just one imperialist will survive. The framework of the proposed ASBA is described as follows.

4.3.2. ASBA Initialisation

The algorithm starts with generating the initial N_p as people through the VLA procedure. Then, the most powerful people, N_c, (people with minimum cost) are chosen as the leaders and the remaining people shape the people of these communities. The people are distributed among countries as their citizens based on the leader's power. Four different types of ASBA initialisation process are used as follows:

ICA-Based (IB) is developed based on original ICA initialisation. Accordingly, a number of countries that have N_{imp} most powerful countries (countries with minimum cost) are chosen as the imperialists. The leader countries proportionally absorb the rest of the people based on their powers. As illustrated in Figure 9, the more powerful empires can absorb the greater number of countries. A detailed and completed review of the ICA is presented byAtashpaz-Gargari and Lucas (2007). This is referred to as ASBA$_{\mathrm{IB}}$ in the results.

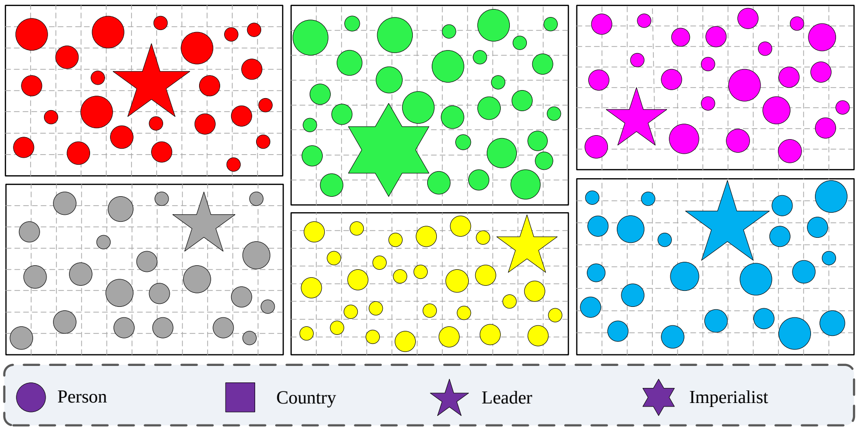


Figure 9. Generating the initial empires and countries: imperialists possess the countries based on their powers.

Classification-based (CB) is constructed based on the geographical determinism concept. In other words, people live in geographical positions and have to obey their local governments. In CB, the initialisation of the people are classified based on the nearest Euclidean norm of distances with their imperialist (Ramezani and Lotfi 2013). It is referred to as ASBA$_{\mathrm{CB}}$ in the results.

The **Frog-based (FB)** process uses an initialisation process introduced in the shuffled frog-leaping (SFL) algorithm (Eusuff and Lansey 2003). People are considered as an SFL population consisting of a set of frogs (solutions) partitioned into subsets referred to as memeplexes (Afzalan *et al.* 2012). For a meticulous and comprehensive review of SFL algorithm, readers are directed to Afzalan *et al.* (2012). This is referred to as ASBA_{FB} in the results.

Hybrid-ICA-frog-based (HIF) process combines the probabilistic idea used in ICA with the ordering idea used in SFL to assign the sorted people to each country based on the described power (Ramezani and Lotfi 2013). It is referred to as ASBA_{HIF} in the results.

The total power of a country is mainly affected by the power of the leader while the power of the people has a diluted impact on the total power of that country. Therefore, the total power of a country is defined as follows:

$$TP_i = \text{Cost}(Leader_i) + \xi mean\{\text{Cost}(People\ of\ \ country_i)\} \tag{63}$$

where ξ is a positive number less than 1. The total power of the country is determined by the leader when the value of ξ is small; however, as the value of ξ increases, the role and power of the people becomes more important.

Once initial empires are formed, the evolutionary algorithm operators (EAO) such as selection, crossover and mutation are used in each decade to improve the people of each country. The total number of decades is shown by N_d. The mutation procedure is performed on people of all countries, while the crossover procedure is randomly done to mate the people, leaders and empires with each other. The percentage of population carried out in the crossover is shown by P_c. It is also performed to maintain diversity in the population and preservation is carried out to stick to local optimal solutions (Arumugam and Rao 2007). Mutation is carried out on P_m fraction of the population. In this paper, the roulette wheel selection (RWS) procedure has been applied to select the population of crossover and mutation. Figure 10 illustrates the procedures of mutation (sections 1 to 4) and crossover (sections 5 to 9) used in this algorithm. Finally, the new solutions are evaluated and placed in the population based on the quality of their objective functions.

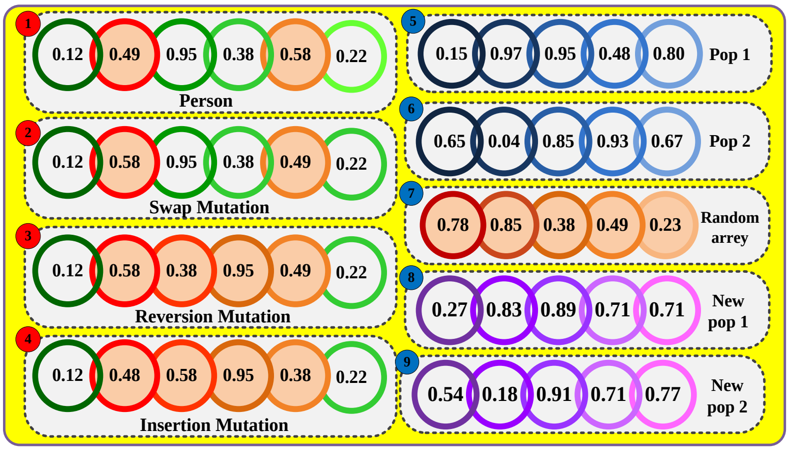


Figure 10. Crossover and Swap, Reversion and Insertion Mutations (“Pop” stands for population).

Since the solution has several parts, some schemes are considered to handle this issue. We considered lots of mutations and crossovers on different parts of solutions randomly. This process has two steps:

- ***First step:*** Selecting the current solutions: Either a leader or a random person is selected to use the mutation operator. One person or two and the leader are selected randomly to use on the crossover operator.
- ***Second step:*** Selecting a part(s) from the structure of the solution: One part or more of the solution are selected

randomly to be changed by an operator. For example, part (1) of a solution is selected and the sequence of vehicles is changed to produce a new person(s). Then the new solution(s) is (are) considered for the revolution process.

In monarchy countries, the countries of the empires start to move towards their relevant imperialist. This movement is a simple model of an assimilation procedure applied to some imperialists. Revolution brings some sudden random changes in the positions of some countries. In each country, the new people with lower cost than the leader are swapped with the leader. Similarly, this process is applied to each empire to select a better leader as a new empire. All empires try to take possession and control of the colonies of other empires. In this rat race, the power of the weaker empires gradually reduces and that of the more powerful one rises. Accordingly, one of the weakest countries of the weakest empires is picked and possessed among the empires based on empires' total power. In this regard, the RWS method is used for assigning the people or countries to the country or empire. Two types of assimilation are used as follows.

4.3.3. External revolution operations (ERO)

EROs are applied to the countries with revolution probability (P_e). The assimilation process is accomplished just on the monarchy countries of each imperialist. The country's movement toward an imperialist means that all the people in these countries move in the same way towards the empires. As illustrated in Figure 11, the leaders move toward the empires by x steps. The new position of a leader is calculated as follows:

$$x \sim U\left(0, Cof_{ea} \times d\right) \tag{64}$$

where x is a random variable distributed uniformly between 0 and $Cof_{ea} \times d$. Cof_{ea} is the external assimilation coefficient greater than 1; d is the distance between leader and empire.

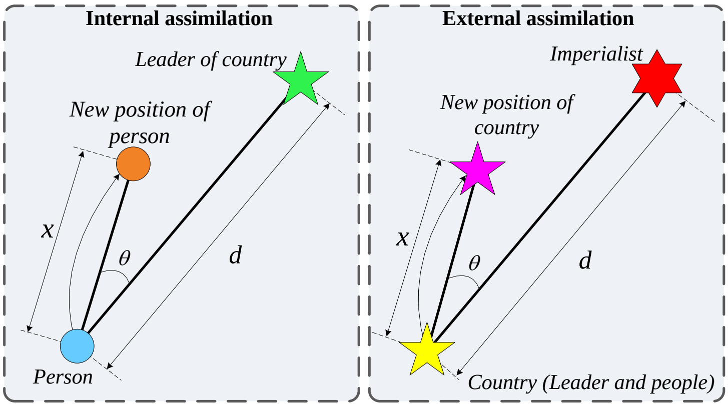


Figure 11. The external and internal assimilation procedure.

The countries do not move directly towards their empires. They move with a deviation of $\theta \sim U(-\lambda, \lambda)$ from the connecting line between the country and its imperialist by x units to increase the search area. Estimating λ as $\pi/4$ (rad) usually leads to a continuous convergence of the countries to the global optimum (Ramezani and Lotfi 2013). The revolution occurs in all the countries. As the revolution is against the empire in monarchy countries, the people of a country should move the same way.

4.3.4. Internal revolution operations (IRO)

IROs are performed on the people of each country with revolution probability (P_i). This assimilation is different in multinational communities and is implemented in two steps. Firstly, each person in the i^{th} country moves its i^{th}

position toward the i^{th} position of its leader as $x_i \sim U\left(0, Cof_{ia} \times d_i\right)$. d_i is the distance of the i^{th} position of a person to the i^{th} position of a leader and Cof_{ia} is the internal assimilation coefficient, which it is greater than 1.

The assimilation procedure occurs in all countries to direct the people toward the leaders, and the revolution procedure is applied to all countries to improve the position of the people. The internal and external forms of revolution are shown in Figure 12.

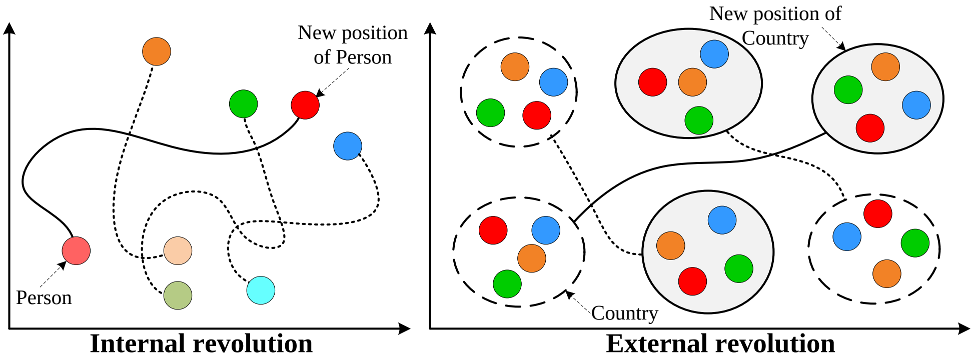


Figure 12. Internal and external revolution procedures.

4.3.5. Global war

ASBA benefits from the global war procedure as a universal phenomenon that affects the construction of both countries and societies simultaneously. After a number of iterations, a global war (I_{GW}) is introduced to produce a new population equal to the initial size of population for the preservation of premature convergence (Rabiee *et al.* 2012). Then, the new population is merged with the old population. Then, the unified population is sorted in ascending order based on cost functions. Subsequently, a number of populations equal to the old population are selected. This process is iteratively repeated a number of times (N_{GW}). In this paper, the stopping criteria or end of ASBA is considered when there is only one empire for all of the countries. The pseudo-code of ASBA is illustrated in Figure A3 (Appendix).

5. DATA GENERATION

There are a lot of test instances in the literature of the VRP such as Cordeau *et al.* (2001), Dondo and Cerdá (2009) and Liu and Shen (1999), which consider the usual inputs such as positions of depots, demand of customers, time windows, capacity and number of vehicles. In our research, we consider an OUT policy in the replenishment process including forecasting and inventory management. Moreover, vehicle failure is the main contribution of this paper and its related data are not considered in any of these test instances. Additionally, the following inputs are not considered in any of these popular and publicly available test instances: The position of vehicles at failure point (*x* and *y* coordinates); Position of service centre; Average travel/carrying cost of a failed vehicle; Fixed repair cost; Target net stock; Target WIP; Shortage cost of per product; Holding cost of products; Inventory capacity of retailer; Average speed of moving to a failed vehicle; Time to repair vehicle; and Parameter of exponential distribution for vehicle failure.

Therefore, the existing databases may help only partially and calculated objective functions cannot benchmark our algorithms; we have to generate lots of parameters. Therefore, we decided to generate some new numerical examples for this problem. Twenty-four random problems are generated to analyse the proposed methods. These problems are classified into three groups to analyse the capability of the algorithms in different sizes: large, medium, and small. In each test problem, there is a distribution network consisting of a depot, set different vehicles of various types (MK) and a set of retailers that order G types of products. The depot and retailers are located in a geographical area of $U(0,100)$ randomly. The number of time periods is 100. The test data and information generated for the problems are presented in Table 3. In addition, ω_{ig} is assumed as the average vehicle capacity. ς_{ig} is equal to the inventory capacity of the retailers considered for each product. α is assumed to be 0.25. $\lambda_{m_k kp}$ which is uniformly distributed in range (0.02, 0.5) and P_c is uniformly distributed as U [0.95, 0.98].

Table 3. Factors and their levels.

Factors		**Levels**	
$(N \times MK \times G)$	Small size	Prob. #1: $(4\times1\times2)$	Prob. #5: $(22\times5\times3)$
		Prob. #2: $(6\times2\times2)$	Prob. #6: $(30\times6\times4)$
		Prob. #3: $(10\times3\times3)$	Prob. #7: $(38\times6\times4)$
		Prob. #4: $(14\times4\times3)$	Prob. #8: $(48\times8\times4)$
	Medium size	Prob. #9: $(72\times10\times5)$	Prob. #13: $(130\times12\times6)$
		Prob. #10: $(92\times10\times5)$	Prob. #14: $(145\times14\times6)$
		Prob. #11: $(104\times10\times5)$	Prob. #15: $(160\times14\times6)$
		Prob. #12: $(115\times12\times5)$	Prob. #16: $(175\times15\times6)$
	Large size	Prob. #17: $(210\times15\times8)$	Prob. #21: $(270\times18\times8)$
		Prob. #18: $(224\times16\times8)$	Prob. #22: $(288\times20\times10)$
		Prob. #19: $(240\times16\times8)$	Prob. #23: $(305\times20\times10)$
		Prob. #20: $(255\times18\times8)$	Prob. #24: $(320\times24\times10)$
Demand ordered by retailers (d_{igp})		$N(100,20)$: Normal distribution (μ,σ)	
Average cost of travelling (C_k)			U (0.5,1.5)
Average cost of carrying the failed vehicles (ρ_k)			U (5,15)
Waiting penalty cost (Γ_{ep})			U (1,5)
Lateness penalty cost (Γ_{lp})			U (5,10)
Shortage cost of each unit (b_{igp})			U (4.5,12.5)
Holding cost of each unit (h_{igp})			U (1.5,3.5)
The fixed cost of the vehicle (f_{kp})			$q_k \times U(10,\ 50)$
The fixed repair cost (ψ_{kp})			$q_k \times U(40,\ 100)$
The average speed $(u_{m_k kp})$			$\min q_k \times (U(8,12)/q_k)$
The average speed for the failed vehicles $(\gamma_{m_k kp})$			$\min q_k \times (U(1,4)/q_k)$
The replenishment time ($\vartheta_{im_k kp}$)			$U(0.02,\ 0.05) \times W$
The vehicle repair time ($\phi_{m_k kp}$)			$U(0.12,\ 0.25) \times W$
Earliest arrival time (e_{ip})			$U(0.1,\ 0.85) \times W$
Latest arrival time (τ_{ip})			$e_{ip} + U(0.05,\ 0.15) \times W$

The inventory capacity of the retailers for each product (φ_{ig}) is estimated as follows:

$$\varphi_{ig} = \sum_{p \in P} d_{igp} \times \frac{U(1,4)}{P}, \quad \forall i \in N, g \in G \tag{86}$$

In order to determine q_k, the maximum of summation of total demands in all periods is calculated and then λ_k is generated as a random coefficient of each vehicle type k based on uniform distribution of U[1, 10].

Finally, the capacity of vehicles is estimated as follows:

$$q_k^{mean} = \frac{\sum_{p \in P} \sum_{i \in N} \sum_{g \in G} d_{igp} \times (1+\beta)}{P} \times \frac{\lambda_k}{\sum_{k \in K} \lambda_k}, \qquad \forall k \in K \tag{87}$$

$$q_k^{min} = \left[(1\text{-}\delta) \times q_d^{mean}\right] \text{ and } q_k^{Max} = \left[(1+\delta) \times q_d^{mean}\right], \qquad \forall k \in K \tag{88}$$

$$q_k = U(q_k^{min}, q_k^{Max}), \qquad \forall k \in K \tag{89}$$

where β and δ are coefficients with U[0, 0.1] and U[0, 0.1], respectively. W is considered to be eight hours a day.

6. PREPARATION AND VALIDATION OF SOLUTION ALGORITHMS

In this section, we conduct an experiment to find the initial solution. Then, parameter tuning is performed. After that, the validation approaches for proposed solution algorithms are introduced. Finally, the results of the validation are presented. Note that we implement the algorithms in MATLAB 2013a software product and run on 3 parallel 2.6 GHz PCs with 2GB of RAM.

6.1. Evaluation of the proposed initial procedure

VLA as the developed initial procedure is evaluated. In this regard, the assignment procedure used in the initialisation process is performed through GA. As illustrated in Figure 13, VLA can help algorithms to find the optimum solution with lower CPU time. Consequently, the VLA can provide the algorithm with more qualified initial solutions, which may find better solutions based on their capabilities in a shorter time. This scheme can be useful especially in large-scale problems and can help to save a great deal of time.

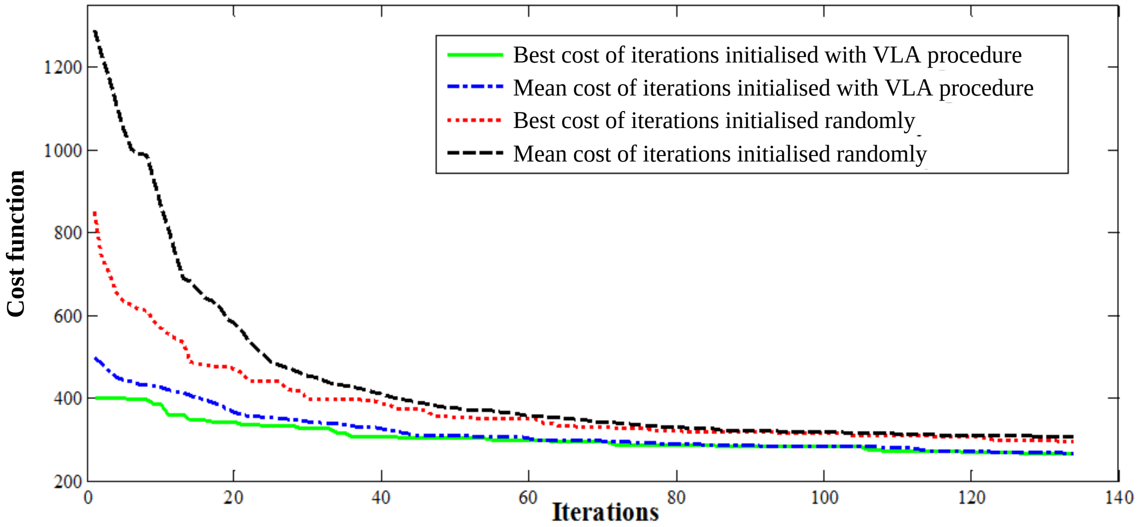


Figure 13. Effect of VLA on the best cost and mean cost of population over the different iterations.

6.2. Parameter tuning

The efficiency of metaheuristic algorithms depends significantly on choosing values for their parameters. Parameter tuning can eradicate premature convergence, deepen searching around interesting regions, and increase diversity in the search space (Behnamian and Fatemi Ghomi 2011). Therefore, this process can improve the performance of algorithms through determining miscellaneous parameters before the final runs. In this paper, response surface methods (RSM), introduced by Box and Wilson in the early 1950s, are used to determine the values of the parameters (Myers *et al.* 2009). We use the polynomial response surface function (y) introduced by Neter *et al.* (1996) to find the optimal values of parameters and save time.

RSM as one design of experiments (DOE) methods considers a lower bound (X_l) and an upper bound (X_h) for each variable. The number of experiments (N_E) contains 2^J factorial points (n_f) or a fraction of it (i.e. 2^{J-1} or 2^{J-2} (Montgomery and Myers 2002)) (2J) axial points (n_{ax}) coded as ± 1, and n_{cp} central points for each algorithm as illustrated in Table 4. As presented in Table 4, when the number of factors like ICA, SBA, HGS, and ASBA is too high, a fraction of factorial points are considered (Myers *et al.* 2009). For instance, there are eight factors in the problem for ICA, which lead to 2^8+16+10=282 experiments in the full scheme. As the number of experiments is too high, we can use 1/4 fraction scheme, which results in only 2^{8-2}+16+8=88 experiments.

Table 4. Level of factors in the algorithms with the number of experiments.

Algorithm	Factors and their levels (X_l,X_h)									$N_E = (n_f, n_{ax}, n_{cp})$
ICA	$nPop$	N_{Imp}	$Maxdc$	ξ	P_{AS}	P_R	I_{GW}	N_{GW}		88=(2^{8-2},16,8)
	(100,200)	(4,8)	(100,200)	0.03,0.05)	(0.2,0.4)	(0.2,0.4)	(40,100)	(1,2)		
SBA	$nPop$	$N_{country}$	N_{Imp}	ξ	N_d	P_i	P_e			88=(2^{7-2},14,10)
	(150,300)	(20,40)	(4,8)	0.03,0.05)	100,200	(0.2,0.4)	(0.2,0.4)			
HGS	$MaxIt$	$nPop$	P_c	P_m	$LSiter$	Max_{ipt}	T_0	T_f		88=(2^{8-2},16,8)
	(80,120)	(60,100)	0.62,0.72	(0.2,0.34)	(50,80)	(4,8)	(10,20)	(0.001,0.01)		
HPV	$MaxIt$	$nPop$	LS_{iter}							20=(2^3,6,6)
	(300,600)	(40,100)	(8,12)							
ASBA	$nPop$	$N_{country}$	N_{Imp}	ξ	N_d	P_i	P_e	I_{GW}	N_{GW}	70=(46,18,6) Min Run Ress V
	(100,200)	(20,40)	(4,8)	0.03,0.05)	100,200	(0.2,0.4)	(0.2,0.4)	(40,100)	(1,2)	

Note that the impact of different strategies in the parameter tuning process of ASBA (i.e. ASBA$_{IB}$, ASBA$_{CB}$, ASBA$_{FB}$ and ASBA$_{HIF}$) in our experiments is neutralised by considering them in different blocks. HGS parameters include $MaxIt$, $nPop$, P_c, P_m, LS_{iter}, Max_{ipt}, T_0 and T_f. Furthermore, HPV parameters include $MaxIt$, $nPop$ and LS_{iter}. Lastly, $Maxdc$, P_{AS} and P_R are the input parameters of ICA.

Since we intend to simultaneously analyse the effectiveness and efficiency of the algorithms by RSM, a multiple objective decision-making (MODM) problem is used to estimate the input variables by producing the desired response based on different objectives at the same time. Accordingly, in this paper, a utility function $d_i(Y_i)$ provided by Derringer and Suich (1980) is used to optimise the multiple responses. This technique estimates the appropriate parameters from the objective function and CPU time concurrently. $d_i(Y_i)$ is calculated as follows:

$$d_i(Y_i) = \left(\frac{H_i - Y_i}{H_i - L_i}\right)^S \quad L_i \le Y_i \le H_i \tag{90}$$

where $d_i(Y_i)$ represents the utility function of $Y(i)$, which is the response functions in the form of minimisation. L_i and H_i are the lower and upper bounds, respectively. S is the severity of $d_i(Y_i)$ by, which the decision maker can consider various weights for different goals. The severity value can be selected between 0 to10 (a higher severity means more emphasis on the goal). In this paper, the parameters of S for the objective function and CPU time are 3 and 1, respectively. The desirability of the obtained response can be calculated by the following equation:

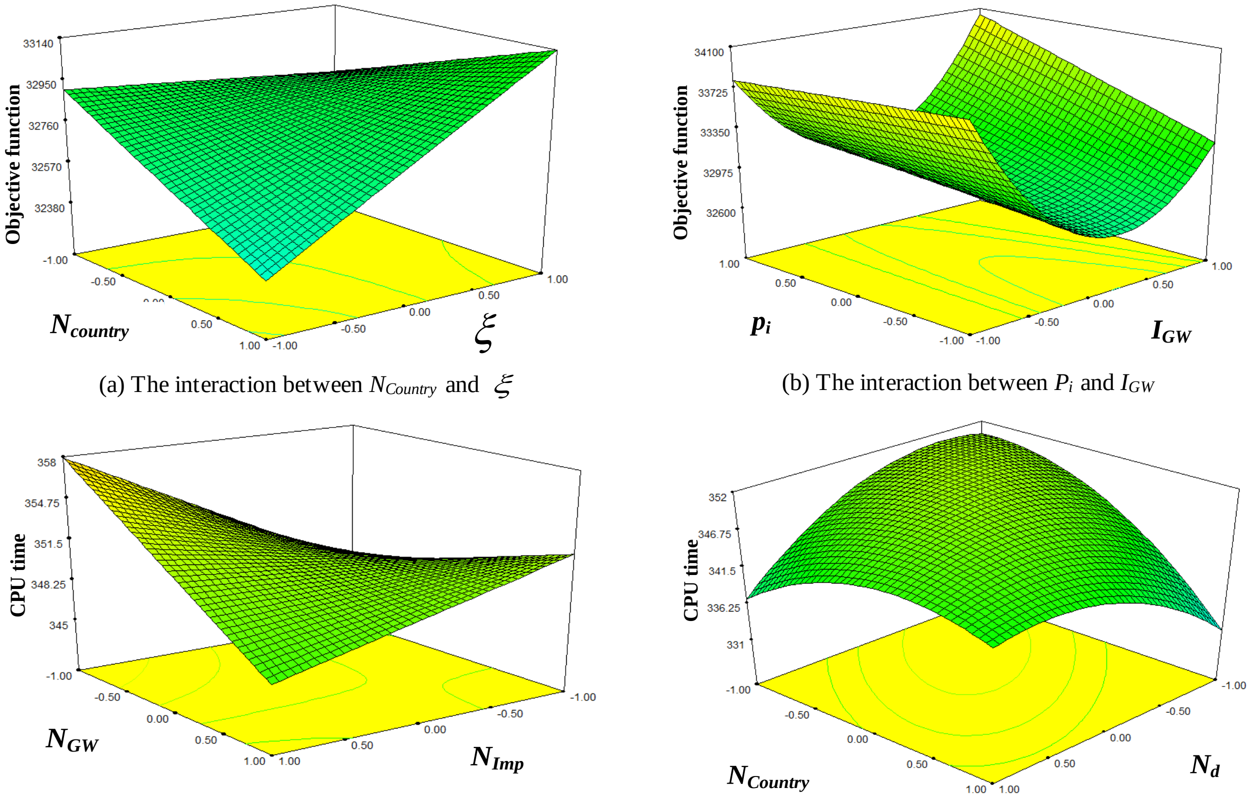
$$D = \sqrt[m]{d_1(y_1) \times d_2(y_2) \times \ldots \times d_m(y_m)} \tag{91}$$

where m is the number of objective functions. The tuned values for parameters, R-squared (R^2) and desirability (D) are estimated and reported in Table 5.

Table 5. Tuned parameters, R-squared (R^2) of objective function and CPU time and desirability (D).

Algorithm	Tuned parameters	R^2 Objective function	R^2 CPU time	D
ICA	$PopSize = 192,\ N_{Imp} = 7,\ Maxdc = 198, \xi = 0.038, P_{AS} = 0.34, P_R = 0.32, I_{GW} = 84,\ N_{GW} = 2$	76.2%	78.4%	0.938
SBA	$PopSize = 284,\ N_{Country} = 36, N_{Imp} = 8,\ \xi = 0.046, N_d = 192, P_i = 0.36, P_e = 0.4$	71.8%	74.5%	0.875
HGS	$MaxIt = 112, nPop = 92, P_c = 0.68, P_m = 34, LS_{iter} = 64, Max_{ipt} = 6, T_0 = 16.6, T_f = 0.006$	82.5%	79.4%	0.982
HPV	$MaxIt = 546,\ nPop = 96, LS_{iter} = 12$	62.6%	71.3%	0.842
ASBA	$PopSize = 192, N_{Country} = 32, N_{IMP} = 8, \xi = 0.042, N_d = 184, P_i = 0.36, P_e = 0.32, I_{GW} = 86, N_{GW} = 2$	68.2%	68.8%	0.864

As illustrated in Figure 14, the most important interactions (the highest partial R^2-square ($P_{r\text{-}s}$)) of ASBA are 4.37% between $N_{Country}$ and ξ and 3.2% between P_i and I_{GW} over the objective function and also 11.21% between N_{GW} and N_{Imp} and 1.26% between $N_{Country}$ and N_d over CPU time.



(a) The interaction between $N_{Country}$ and ξ

(b) The interaction between P_i and I_{GW}

(c) The interaction between N_{GW} and N_{Imp}

(d) The interaction between $N_{Country}$ and N_d

Figure 14. The most important interactions in parameters of ASBA over objective function (a,b) and CPU time (c,d).

6.3. Validation Approaches

The proposed mathematical model in Section 4 is complicated so that even solving it exactly for small- and medium-sized cases is impossible. Accordingly, we evaluate the efficiency of proposed algorithm by two approaches: (1) validation based on a lower bound and (2) validation based on previous solution benchmarks. For these aims, a comparative performance measure, namely relative percentage deviation (RPD), is defined and calculated as follows:

$$RPD = \frac{|Alg_{sol} - LB|}{LB} \times 100 \qquad (66)$$

Where Alg_{sol} is a value obtained by a given algorithm and LB is a lower bound of the given problem. The LB is used instead of the best value among the algorithms especially for the second validation.

6.3.1. Validation Based on Previous Solution Benchmarks

In the literature of IRP, four well-known solution algorithms are existed which are our benchmark methods. Our validation approach is done through a comparison process among the solutions of them and benchmark methods. We have considered four benchmark methods as follows. *Social-based algorithm (SBA)* as an effective population-based approach (Ramezani *et al.* 2015) and a basic approach of ASBA. *Imperialist competitive algorithm (ICA)* as a popular effective population-based approach, which has significant strength in solving many real-world permutation problems such as scheduling (Rabiee *et al.* 2012) and the VRPs (Wang *et al.* 2011). *Hybrid GA and SA (HGS)* are considered an effective method in routing problems integrated with inventory approaches (Karaoglan and Altiparmak 2010). The popular traditional approaches GA and PSO, two original methods of HGS, are considered the best evolutionary methods and swarm intelligent algorithms from population-based methods, respectively (Beheshti *et al.* 2014; Cheng *et al.* 2017). *HPV* developed by hybridisation of particle swarm optimisation (PSO) and variable neighbourhood search (VNS) is another effective approach in the IRP (Liu *et al.*

2016). SA and VNS, two basic methods of HPV, are considered as the best trajectory methods in permutation problems (Liu and Lin 2005; Popović *et al.* 2012).

In summary, overall we are dealing with eight algorithms: 1) four ASBA-based algorithms (i.e. ASBA with four scenarios); and 2) four benchmark algorithms, namely ICA, SBA, HGS and HPV. Note that the other basic and traditional metaheuristics such as GA, SA, PSO and VNS are not included in the comparison process due to the lower quality of their results compared to the other mentioned approaches. Since HGA, HPV, ICA and SCA are well-known metaheuristic algorithms, we do not explain them in details. However, the construction of the benchmark algorithms is described briefly in the following subsections. The Initialisation procedure and Encoding scheme in all methods are similar to the proposed solution algorithms. Interested readers may refer to Karaoglan and Altiparmak (2010), Liu *et al.* (2016), Rabiee *et al.* (2012) and Ramezani *et al.* (2015) for a more detailed review of these algorithms.

Social-based algorithm (SBA): As the SBA is a basic approach of ASBA, this algorithm is considered a benchmark method in this paper. As the construction of this algorithm has a lot of resemblances with ASBA, its original construction is not presented in this paper. Interested readers are referred to Ramezani and Lotfi (2013) and Ramezani *et al.* (2015) for more detail about SBA.

Imperialist competitive algorithm (ICA): ICA was firstly proposed by Atashpaz-Gargari and Lucas (2007) as a novel effective EA method based on humans' socio-political evolution. ICA has been widely applied to many non-permutation optimisation problems such as scheduling and the VRP (Bagher *et al.* 2011; Forouharfard and Zandieh 2010; Wang *et al.* 2011). Since this method is well-known, its construction is not presented in this paper. Interested readers may refer to Atashpaz-Gargari and Lucas (2007) and Rabiee *et al.* (2012) for more detail about ICA.

HGS algorithm: The HGS is developed by hybridising two well-known methods, namely GA and SA. The aim of designing this hybrid algorithm is to combine the strengths of GA and SA to improve the effectiveness of a single approach (Devika *et al.* 2014). In HGA, the initial population of individuals with *nPop* size is generated by VLA. Moreover, combining strong facets of various methods as a new integrated approach has been growing in this field of research because of their ability to increase computational power (Behnamian *et al.* 2009).

We consider GA (proposed by Holland 1992) because it has wide applications in solving permutation problems such as the VRP, the IRP, scheduling and other operations research problems (Abdelmaguid and Dessouky 2006; Ho *et al.* 2008; Mansouri 2005). In the GA, the RWS procedure has been applied to select the chromosome of crossover and mutation. Chromosomes among individuals are selected to mate and create offspring in crossover procedures. Additionally, the mutation procedure is performed to maintain diversity in the population and avoid local optimum solutions. The percentages of the population subject to crossover and mutation are shown by P_c and P_m, respectively. The crossover and swap, reversion and insertion mutations presented in ASBA are used in HGS as crossover and mutation operators. The GA as the main loop of HGS is repeated *MaxIt* times, and in each iteration, the SA algorithm is performed as the local search. We consider SA (introduced by Metropolis *et al.* 1953 and popularised by Kirkpatrick *et al.* 1983) because it has been known and widely applied to many permutation optimisation problems such as scheduling and the VRP (Elmi *et al.* 2011; Gaafar and Masoud 2005; Lin *et al.* 2009), because of its theoretical guarantee of convergence, good performance on many practical problems, and ease of implementation (Zhou and Chen 2010). SA boosts the search algorithm to escape from trapping in the local optimum solutions (Lin *et al.* 2009) through accepting worse solutions with some probability (Elmi *et al.* 2011).

First of all, the SA receives the best solution obtained by the main loop (i.e. GA as an initial solution, and then neighbours will be found around the initial solution). In the SA algorithm, the swap, reversion and insertion mutations are used to refine the neighbourhood structures. Each new solution in SA will be considered by the acceptance probability function i.e. $p = e^{(-\Delta C / T)}$ where T is the current temperature, $\Delta C = C(s') - C(s)$, ΔC is the difference of system energy the current solution (s) and new neighbour solution (s'). When ΔC is non-positive, the new state is always accepted (Elmi *et al.* 2011). SA is guided by the initial temperature T_0 and cooling rate T_f. These values essentially dictate the acceptability of solutions found during the search (McMullen and Frazier 2000). The temperature reduction form used in SA at each iteration is $T_{t+1} = \alpha * T_t$ where α is temperature discount rate. SA is repeated as local search LS_{iter} times and considers Max_{ipt} new neighbours at each temperature. The construction of HGS in the pseudo-code is presented in Figure A4 (Appendix).

HPV algorithm: HPV is constructed by hybridising PSO and VNS algorithms. In this algorithm, PSO forms the main loop and the VNS algorithm performs the local search. The hybrid approaches are constructed to achieve a trade-off between the global and local exploitation during the search process (Behnamian *et al.* 2009). As proposed by Kennedy and Eberhart (1995), PSO, a metaheuristic solution algorithm, has been used extensively in different fields including supply chain management, the VRP and scheduling (Panigrahi *et al.* 2011; Sedighizadeh and Masehian 2009).

The initial population called swarm with *nPop* size is generated by VLA in the initialisation process. Subsequently, this algorithm should iteratively move the population toward a direction, which improves a given fitness function. Each individual in the swarm is called a particle and is a candidate solution, which is characterised by a position vector, $\vec{x}_i$, and a velocity vector, $\vec{v}_i$. The velocity of particle i in iteration k is computed as follows:

$$\vec{v}_i(k+1) = w_i \times \vec{v}_i(k) + c_1 \times \vec{r}_1(\vec{x}_i^{pb}(k) - \vec{x}_i(k)) + c_2 \times \vec{r}_2(\vec{x}^{gb}(k) - \vec{x}_i(k)) \quad (65)$$

where $\vec{x}_i^{pb}$, the particle's personal best, is the best position of a particle in all the previous iterations; $\vec{x}^{gb}$, swarm global best, is the best position obtained so far; c_1 and c_2 are positive constants known as cognitive and social components, respectively; r_1 and r_2 are uniformly distributed samples in the interval [0,1]; and w_i is the inertia weight to control the impact of previous velocity on the current one. The parameters ω, c_1 and c_2 are assumed to be 0.9, 2 and 2, respectively (Moslemi and Zandieh 2011). A particle's new position is obtained by adding the velocity to its current position.

We take advantage of VNS because it is considered a very popular local search method (Sevkli and Aydin 2006) using a series of systematic changes of the neighbourhood structure (NS) within the search process in order to optimise problems. VNS receives the best solution achieved from PSO as the initialisation solution in each iteration. Then, a set of NSs symbolised by N_{ns} is used to find the near-optimal solution where $ns_{\max}$ represents the maximum number of NSs implemented with a random sequence. The swap, reversion and insertion mutations presented in ASBA construction are considered as NSs in the HPV algorithm. The VNS is applied NSs repeated LS_{iter} times at each iteration of the main loop, which is repeated *MaxIt* times. Interested readers may refer to Kennedy and Eberhart (1995), Hansen and Mladenović (2001) and Mladenovic and Hansen (1997) for more detailed literature on PSO and VNS. Figure A5 (Appendix) gives the HPV in pseudo-code.

6.3.2. Validation Based on a Suggested Lower Bound

In this section, we introduce a lower bound for the solutions of the master model. Comparing the growth rates of the gaps between the solutions of algorithms (proposed and benchmark algorithms) and the lower bound for different size of the problem can evaluate the effectiveness and efficiency of the designed algorithms. Accordingly, each test problem in all of the algorithms is run three times. Since each problem is solved three times, the average RPD ($\overline{RPD}$ or ARPD) is defined as the average of the three runs of each algorithm for a given test problem. To calculate the *LB*, used in RPD measure, some relaxations are applied. The restrictions are on (a) the time windows penalty function, (b) limitation of working time and (c) vehicle failure for all vehicles. Relaxing these terms results in a linear model in which all the vehicles will deliver products to customers without failing or being penalised due to violation of time windows. Therefore, constraints (14)-(33), (36)-(45) and (50) are eliminated and r_{1igp} and r_{2igp} are converted to parameters. The model with a revised objective function is solved subject to the other constraints to obtain the *LB*. The *LB* model is as follows:

$$z^{*} = \min \sum_{p\in P}\Big(\sum_{k\in K}\sum_{m_k\in M_k}\sum_{i,j\in A} c_k l_{ij} x_{ijm_kkp} + \sum_{g\in G}\sum_{i\in N} h_{igp}(s_{igp}+o_{igp}) + \sum_{g\in G}\sum_{i\in N} b_{igp}\pi_{igp} + \sum_{k\in K}\sum_{m_k\in M_k}\sum_{j\in N} f_{kp} x_{0jm_kkp}\Big) \tag{67}$$

Subject to

$$\sum_{k\in K}\sum_{m_k\in M_k}\sum_{i\in A, i\neq j} x_{ijm_kkp} = 1 \qquad \forall j\in N, p\in P \tag{68}$$

$$\sum_{j\in A} x_{ijm_kkp} + \sum_{j\in A} x_{jim_kkp} = 2z_{im_kkp} \qquad \forall i\in N,\ m_k\in M_k,\ k\in K,\ p\in P \tag{69}$$

$$\sum_{i\in S_{m_kk}}\sum_{j\in S_{m_kk}} x_{ijm_kkp} \le \sum_{i\in S_{m_kk}} z_{im_kkp} - z_{i'm_kkp} \qquad \forall S_{m_kk}\in N,\ m_k\in M_k,\ k\in K,\ p\in P,\ for\ some\ i'\in S_{m_kk} \tag{70}$$

$$\sum_{i\in N}\sum_{g\in G} y_{igm_kkp} \le q_k \qquad \forall m_k\in M_k,\ k\in K,\ p\in P \tag{71}$$

$$\sum_{g\in G} y_{igm_kkp} \le q_k z_{im_kkp} \qquad \forall i\in N,\ m_k\in M_k,\ k\in K,\ p\in P \tag{72}$$

$$o_{igp} = \hat{d}_{igp} + r_{1igp}(\varsigma_{ig} - s_{igp} + \pi_{igp-1}) + r_{2igp}(\omega_{ig} - o_{igp-1}) \qquad \forall i\in N, g\in G, p\in P \tag{73}$$

$$\hat{d}_{igp} = \alpha d_{igp-1} + (1-\alpha)\hat{d}_{igp-1} \qquad \forall i\in N, g\in G, p\in P \tag{74}$$

$$\sum_{k\in K}\sum_{m_k\in M_k} y_{igm_kkp} = o_{igp-1} \qquad \forall i\in N, g\in G, p\in P \tag{75}$$

$$o_{igp} \le \mathrm{O}_{igp} \qquad \forall i\in N, g\in G, p\in P \tag{76}$$

$$s_{igp} = s_{igp-1} - \pi_{igp-1} + \pi_{igp} + o_{igp-1} - d_{igp} \qquad \forall i\in N, g\in G, p\in P \tag{77}$$

$$s_{igp} + o_{igp-1} \le \varphi_{ig} \qquad \forall i\in N, g\in G, p\in P \tag{78}$$

$$z_{im_kkp} \in \{0,1\} \qquad \forall i\in A, m_k\in M_k, k\in K,\ p\in P \tag{79}$$

$$x_{ijm_kkp} \in \{0,1\} \qquad \forall (i,j)\in A, m_k\in M_k, k\in K,\ p\in P \tag{80}$$

$$s_{igp} \ge 0 \qquad \forall i\in N,\ g\in G,\ p\in P \tag{81}$$

$$\pi_{igp} \ge 0 \qquad \forall i\in N,\ g\in G,\ p\in P \tag{82}$$

$$y_{igm_kkp} \ge 0 \qquad \forall i\in N,\ g\in G,\ m_k\in M_k,\ k\in K,\ p\in P \tag{83}$$

$$o_{igp} \ge 0 \qquad \forall i\in N,\ g\in G,\ p\in P \tag{84}$$

$$\hat{d}_{igp} \ge 0 \qquad \forall i\in N,\ g\in G,\ p\in P \tag{85}$$

The LB is an inventory-routing problem (IRP) with proposed replenishment policy. Note that the LB is still NP-hard and solving it exactly for real-size problems is impossible. Accordingly, the LB model for real-sized problems is solved by using a general GA as a well-known method to find LB in all developed algorithms. However, the accuracy of GA in solving the LB model for small-size problems is investigated by comparing its results with exact results in the next subsection.

Sixteen types of small-size test problems are generated, which are different in terms of the numbers of retailers, vehicles and vehicle types. To measure the quality of the objective function as well as the run time of the coded GA, the mathematical formulation of the LB model (coded on GAMS solver) is utilised. Each test problem is run 10 times in MATLAB to find the solutions of GA and one time in GAMS to find its global solution. Therefore, overall 170 runs have been performed in MATLAB and GAMS. Table 6 shows the detailed specifications of the test problems (e.g. the numbers of retailers, periods and products), the GAMS run time (seconds), the GA average run time (seconds), GAMS objective value, maximum and average errors (%).

Table 6. The input features and output results of solving small test problems with GAMS and the GA.

Test ID	# Retailers	# Periods	# Vehicles	# Vehicle types	# Products	# Times that the problem solved with GA	GAMS run time (seconds)	GA average run time (seconds)	GAMS objective value	Maximum error (%)	Average error (%)
1	7	4	6	1	2	10	43	31.1	4831	0.0%	0.0%
2	7	4	6	2	2	10	51	31.7	4548	0.0%	0.0%
3	7	4	6	3	2	10	53	31.8	4351	0.8%	0.2%
4	7	4	6	4	2	10	105	32.4	4504	1.7%	0.3%
5	8	4	7	1	2	10	1415	34.3	10990	2.9%	1.6%
6	8	4	7	2	2	10	1064	34.7	10001	2.2%	1.2%
7	8	4	7	3	2	10	3260	35.5	9520	2.8%	1.5%
8	8	4	7	4	2	10	1072	35.6	9522	3.3%	2.4%
9	9	4	8	1	2	10	1955	37.3	13924	3.3%	1.1%
10	9	4	8	2	2	10	2104	38.2	11390	2.9%	2.4%
11	9	4	8	3	2	10	1341	38.1	11853	3.2%	2.1%
12	9	4	8	4	2	10	1869	38.7	10403	2.3%	1.8%
13	10	4	9	1	2	10	1860	40.6	13167	2.9%	1.9%
14	10	4	9	2	2	10	2110	41.2	15761	3.5%	2.9%
15	10	4	9	3	2	10	1845	41.6	15155	2.0%	1.9%
16	10	4	9	4	2	10	-	42.5	-	-	-

In terms of runtime, it is evident that the GA is much faster than the GAMS solver. The largest test problem (#16) was solved by the GA below 43 seconds while GAMS cannot solve it in exactly one hour. When it comes to the quality of the objective functions, for all test problems, the average and maximum error were less than 2.9% and 3.5%, respectively. Interestingly, for very small test problems, the GA obtains optimal solutions with no error.

6.4. Validation Results

In this section, we show the validity of proposed solution algorithms based on two presented validation approaches.

6.4.1. Based On Previous Solution Benchmarks

To evaluate the proposed algorithms, firstly the obtained data (objective function and CPU time) are transformed into RPD values. Then, the mean RPD results of the expected objective function and CPU time are calculated in Tables 7 and 8, respectively. The results show that from the objective function perspective, ASBA_{HIF} is better than other approaches while from the CPU time point of view, ICA yields better solutions.

Table 7. Average relative percentage deviation in terms of objective function.

Size	Problem	Algorithms							
		SBA	**ICA**	**HGS**	**HPV**	**ASBA$_{IB}$**	**ASBA$_{CB}$**	**ASBA$_{FB}$**	**ASBA$_{HIF}$**
Small	P.1	8.27	13.77	3.75	14.11	3.62	6.43	7.21	**2.53**
	P.2	9.27	14.20	4.07	15.35	3.92	7.33	9.09	**2.75**
	P.3	9.38	11.56	3.98	12.39	3.40	6.25	9.36	**2.32**
	P.4	9.95	10.14	4.05	10.12	3.55	5.80	7.21	**2.93**
	P.5	8.05	8.10	4.49	9.65	3.59	4.87	7.53	**2.36**
	P.6	6.24	8.00	2.88	10.82	2.86	4.09	5.79	**2.52**
	P.7	7.09	8.19	3.49	10.42	3.04	4.49	6.63	**3.01**
	P.8	6.40	8.09	3.74	10.49	2.71	4.36	5.76	**2.28**
Medium	P.9	7.95	8.29	3.63	10.93	3.04	5.29	6.25	**2.30**
	P.10	7.23	8.23	4.61	8.88	3.39	4.81	7.03	**2.45**
	P.11	6.67	8.05	3.75	10.27	2.62	4.17	6.02	**2.16**
	P.12	7.92	8.55	4.58	13.05	3.58	5.38	6.53	**2.32**
	P.13	8.01	8.70	4.04	11.66	3.22	5.07	7.78	**2.30**
	P.14	7.14	8.19	4.32	11.33	3.35	4.62	7.01	**2.25**
	P.15	7.03	8.17	4.66	10.23	2.88	5.61	6.47	**2.30**
	P.16	6.87	8.10	4.59	9.54	3.37	5.25	6.20	**2.26**
Large	P.17	8.16	9.13	5.60	10.80	4.41	5.60	7.64	**2.61**
	P.18	6.67	8.01	3.96	9.74	3.10	4.24	6.15	**2.22**
	P.19	7.96	8.28	3.81	9.08	3.32	4.71	6.29	**2.26**
	P.20	8.78	9.13	4.00	9.71	3.46	4.89	6.81	**2.28**
	P.21	8.79	9.06	4.76	10.68	3.80	5.67	6.77	**2.46**
	P.22	7.16	8.21	3.86	9.66	3.00	4.94	6.75	**2.22**
	P.23	6.78	8.08	4.13	9.06	2.88	4.71	6.63	**1.99**
	P.24	7.88	8.12	3.93	8.95	3.15	4.63	7.22	**2.01**
Average		7.74	9.01	4.11	10.71	3.30	5.13	6.92	**2.38**
standard deviation		0.995	1.734	0.536	1.630	0.412	0.780	0.898	**0.248**

Table 8. The average relative percentage deviation in terms of CPU time.

Size	Problem	Algorithms							
		SBA	**ICA**	**HGS**	**HPV**	**ASBA$_{IB}$**	**ASBA$_{CB}$**	**ASBA$_{FB}$**	**ASBA$_{HIF}$**
Small	P.1	0.06	**0.00**	4.85	6.56	0.52	0.31	0.89	2.38
	P.2	0.27	**0.00**	3.82	5.14	0.25	0.44	0.77	2.19
	P.3	0.10	**0.00**	3.11	3.91	0.10	0.39	0.55	1.86
	P.4	0.11	**0.00**	3.48	4.70	0.20	0.31	0.71	1.70
	P.5	0.06	**0.00**	3.17	4.46	0.14	0.29	0.64	1.72
	P.6	0.06	**0.00**	3.41	4.58	0.17	0.27	0.72	1.68
	P.7	0.04	**0.00**	3.53	4.42	0.27	0.25	0.68	1.72
	P.8	0.03	**0.00**	3.90	4.81	0.36	0.34	0.92	1.60
Medium	P.9	0.03	**0.00**	4.20	4.73	0.39	0.35	0.96	1.79
	P.10	0.03	**0.00**	4.42	4.96	0.46	0.37	0.98	1.83
	P.11	0.03	**0.00**	4.75	5.51	0.61	0.23	1.15	2.00
	P.12	0.03	**0.00**	5.45	5.65	0.64	0.25	1.44	2.60
	P.13	0.04	**0.00**	5.93	6.13	0.88	0.44	1.65	2.80
	P.14	0.05	**0.00**	6.11	6.68	0.77	0.37	1.78	3.08
	P.15	0.06	**0.00**	6.68	7.52	1.07	0.60	2.06	3.26
	P.16	0.17	**0.00**	7.84	9.07	1.34	0.80	2.39	3.87
Large	P.17	0.25	**0.00**	8.65	10.48	1.54	0.92	2.64	4.37
	P.18	0.40	**0.00**	10.45	12.86	1.90	1.20	3.27	5.42
	P.19	0.39	**0.00**	10.28	12.76	1.93	1.24	3.30	5.34
	P.20	0.42	**0.00**	11.60	14.50	2.05	1.34	3.88	6.46
	P.21	0.51	**0.00**	11.13	14.04	2.29	1.54	3.86	5.82
	P.22	0.71	**0.00**	13.25	16.82	2.79	1.94	4.57	7.05
	P.23	0.62	**0.00**	12.59	16.13	2.64	1.83	4.44	6.74
	P.24	0.73	**0.00**	13.86	17.87	2.97	2.11	4.95	7.65
Average		0.22	**0.00**	6.94	8.51	1.10	0.76	2.05	3.54
standard deviation		0.234	**0.000**	3.595	4.614	0.935	0.605	1.449	2.022

To analyse any significant difference between the effectiveness of the algorithms, a single-factor analysis of variance (ANOVA) is performed (see Table 9). Three main hypotheses have been checked before using ANOVA: normality, homogeneity of variance, and the independency of residuals. We studied these hypotheses before using ANOVA and found that there is no bias in the computational experiments. ANOVA results for the objective function confirm that there is at least one algorithm that is different in terms of performance measure.

Table 9. ANOVA results for the objective function and CPU time of all problems.

Aspect	Source	DF	SS	MS	F	P
Objective Function	Factor	7	4515.49	645.07	563.59	0.000
	Error	568	650.12	1.14		
	Total	575	5165.61			
CPU Time	Factor	7	5239.80	748.54	149.89	0.000
	Error	568	2836.62	4.99		
	Total	575	8076.42			

For pair-wise comparison of the proposed algorithms, Fisher's least significant difference method (FLSDM) is employed. The results obtained by FLSDM for the objective function are illustrated in Table 10. As shown in Figure 15, the means and interval plot of problems in terms of RPD confirm that there is a statistical significance in the solution quality from the objective function perspective. It can be clearly seen that ASBA$_{HIF}$ outperformed the other algorithms. According to Figure 16, the hybridisation of two policies – ICA and the frog-based method – has led to better results than other policies used in the ASCA construction. The priority of algorithms in terms of effectiveness is ASBA$_{HIF}$> ASBA$_{IB}$> HGS > ASBA$_{CB}$ > ASBA$_{FB}$ > SBA > ICA > HPV.

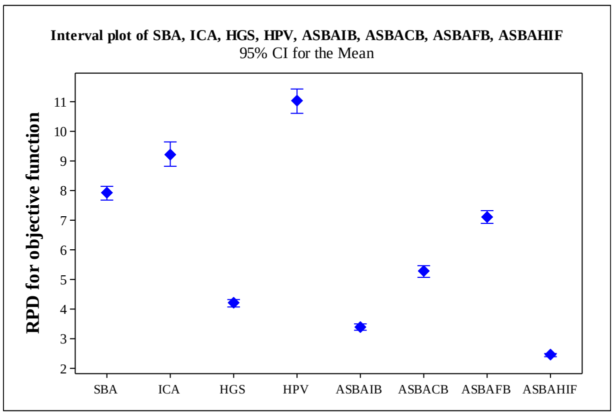


Figure 15. Means and interval plot for total problems' objective function.

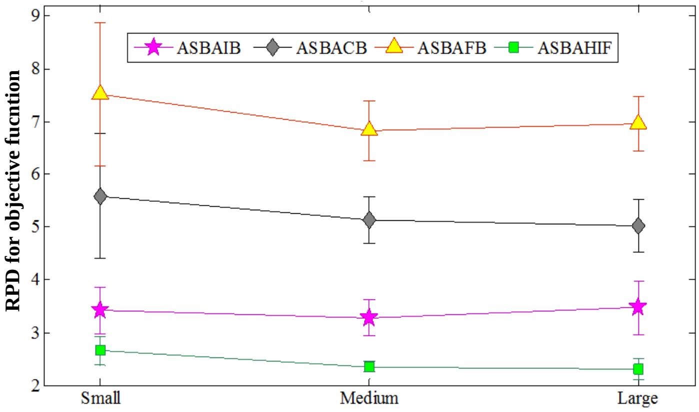


Figure 16. Plots of PRD of objective function for the interaction between different strategies of ASBA and size of problems.

Subsequently, to assess the efficiency of the proposed algorithms, a single factor ANOVA is performed on the CPU time obtained by running different problems three times as reported in Table 9. A pair-wise comparison is also conducted to evaluate the efficiency of the algorithms using FLSDM. The results in Table 10 confirm that the original ICA delivers a better CPU time than the other approaches. The significance of these results was evaluated through a statistical evaluation in terms of RPD with 95% confidence interval for CPU time as illustrated in Figure 17. As shown in this figure, the upper and lower bounds for most of the algorithms do not have any overlap. Therefore, the original ICA and SBA with the minimum lower and upper bounds are the best algorithms in terms of RPD for the CPU time metric. Figure 18 analyses the interaction of different strategies of ASBA and the size of problems. It indicates that according to the CPU time resulting from various ASBA policies, the ISA-based policy is significantly better than the other policies used in construction of ASBA. The priority of algorithms in terms of their effectiveness (i.e. objective function value) is therefore ASBAHIF> ASBAIB> HGS > ASBACB > ASBAFB > SBA > ICA > HPV. The priority of algorithms in terms of their efficiency (i.e. computational time) is ICA > SBA > ASBACB > ASBAIB> ASBAFB > ASBAHIF > HGS > HPV.

Table 10. Fisher's pair-wise comparisons for problems' objective function and CPU time.

Algorithms		RPD of Objective function			RPD of CPU time		
		Lower	Upper	Significant difference at 95% level	Lower	Upper	Significant difference at 95% level
SBA	ICA	0.775	1.857	Yes	-1.329	0.93	No
	HGS	-4.246	-3.165	Yes	5.611	7.871	Yes
	HPV	2.567	3.648	Yes	7.193	9.453	Yes
	ASBA$_{IB}$	-5.057	-3.975	Yes	-0.262	1.998	No
	ASBA$_{CB}$	-3.171	-2.089	Yes	-0.582	1.678	No
	ASBA$_{FB}$	-1.356	-0.274	Yes	0.706	2.966	Yes
	ASBA$_{HIF}$	-5.998	-4.916	Yes	2.201	4.461	Yes
ICA	HGS	-5.563	-4.481	Yes	5.811	8.071	Yes
	HPV	1.25	2.332	Yes	7.393	9.652	Yes
	ASBA$_{IB}$	-6.373	-5.292	Yes	-0.062	2.198	No
	ASBA$_{CB}$	-4.487	-3.405	Yes	-0.382	1.877	No
	ASBA$_{FB}$	-2.672	-1.59	Yes	0.906	3.165	Yes
	ASBA$_{HIF}$	-7.314	-6.232	Yes	2.4	4.66	Yes
HGS	HPV	6.272	7.354	Yes	0.452	2.712	Yes
	ASBA$_{IB}$	-1.352	-0.27	Yes	-7.003	-4.743	Yes
	ASBA$_{CB}$	0.535	1.617	Yes	-7.323	-5.063	Yes
	ASBA$_{FB}$	2.35	3.431	Yes	-6.035	-3.775	Yes
	ASBA$_{HIF}$	-2.293	-1.211	Yes	-4.54	-2.281	Yes
HPV	ASBA$_{IB}$	-8.165	-7.083	Yes	-8.585	-6.325	Yes
	ASBA$_{CB}$	-6.278	-5.196	Yes	-8.905	-6.645	Yes
	ASBA$_{FB}$	-4.463	-3.382	Yes	-7.617	-5.357	Yes
	ASBA$_{HIF}$	-9.106	-8.024	Yes	-6.122	-3.862	Yes
ASBA$_{IB}$	ASBA$_{CB}$	1.346	2.428	Yes	-1.45	0.809	No
	ASBA$_{FB}$	3.16	4.242	Yes	-0.162	2.098	No
	ASBA$_{HIF}$	-1.482	-0.4	Yes	1.333	3.592	Yes
ASBA$_{CB}$	ASBA$_{FB}$	1.274	2.356	Yes	0.158	2.418	Yes
	ASBA$_{HIF}$	-3.368	-2.287	Yes	1.653	3.913	Yes
ASBA$_{FB}$	ASBA$_{HIF}$	-5.183	-4.101	Yes	0.365	2.625	Yes

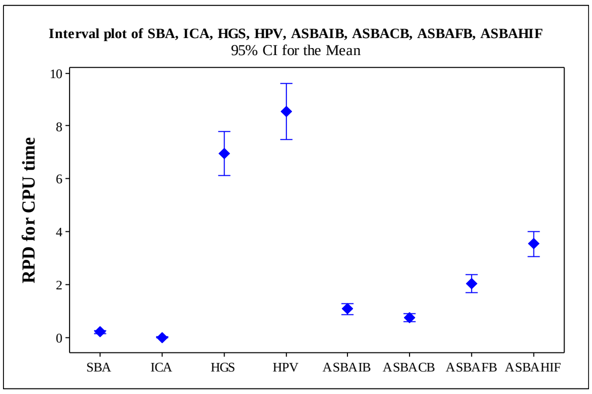


Figure 17. Means and interval plot for total problems CPU Time.

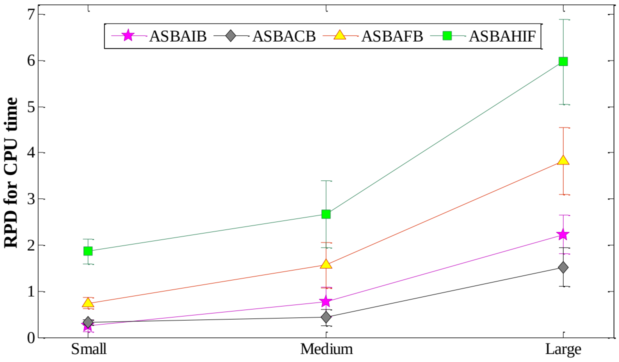


Figure 18. Plots of PRD of CPU time for the interaction between the different strategies of ASBA and size of problems.

6.4.2. Based on the Suggested Lower Bound

In this section, a sensitivity analysis is performed to analyse the role of the problem size in the different algorithms. First, the ARPD of the objective functions is calculated for each problem separately. As illustrated in Figure 19, the ARPD of different algorithms are plotted over different size problems. It can be observed that with the growth of the problem size, the efficiency of the proposed algorithm (i.e. ASBA) outperforms other approaches. Moreover, the growth rate of the gap between ARPD of the proposed algorithms and LB is smaller than other than previous algorithms showing the superiority of our proposed algorithms over the others especially for large-sized problems.

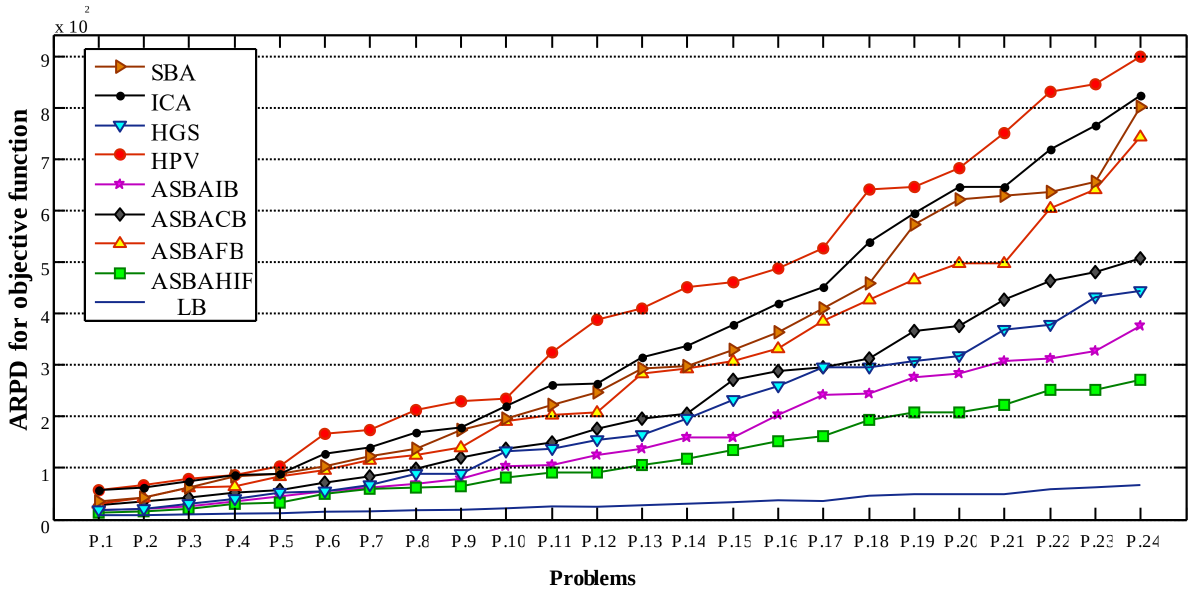


Figure 19. The interaction among different problems and performance of the algorithms.

As shown in Figure 20, it can be clearly seen that the algorithms have levelled for small test problems (i.e. 1st to 8th problems) in terms of CPU time but the difference takes an exponential trend after the 9th test problem. According to this figure, HPV results in a better CPU time than other approaches in all the test problems.

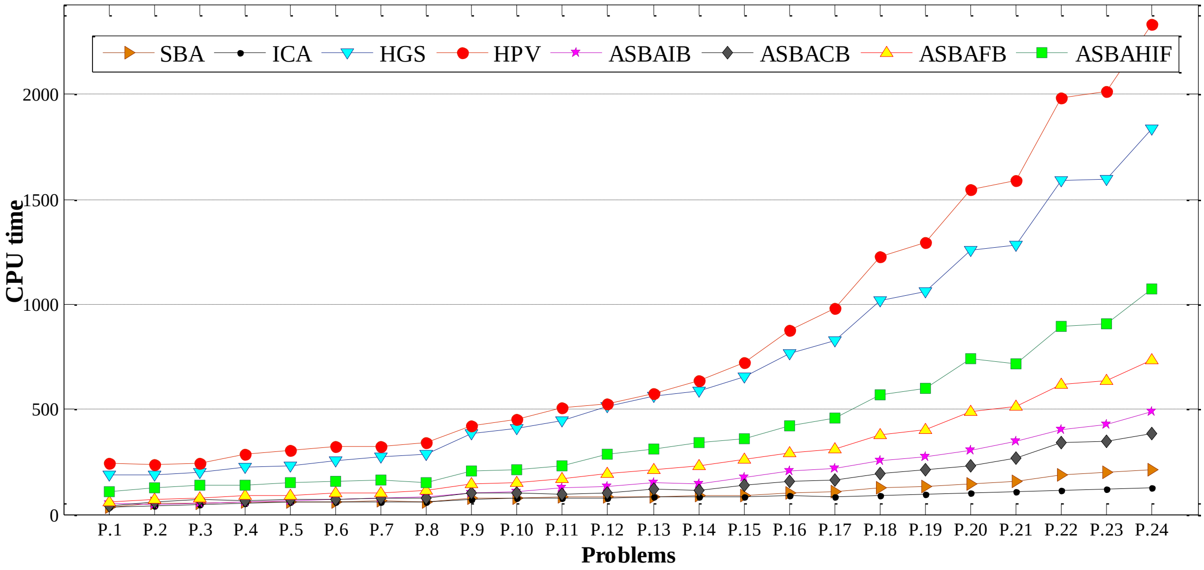


Figure 20. The interaction among different problems and CPU time of algorithms.

7. INSIGHTS

In this section, through sensitivity analyses on the model's parameters, some insights for practitioners are presented. To this aim, we perform the sensitivity analysis on 100 periods of the15th test problem. Five analyses are conducted.

7.1. Proportion of different cost components in total cost

As illustrated in Figure 21, the components of objective function (i.e. total transport cost) including vehicle transport cost and fixed vehicle cost, time window violation penalty cost, backlogging and inventory costs and also the cost of the failed vehicles (including the failed vehicle carrying cost and repair cost) are reduced during different iterations. This figure presents different terms in the total cost objective function optimised by ASBA. Although the proposed method reduces all costs, based on the input parameters, the rate of reducing in backlogging cost is more than the inventory cost. In other words, the cost of backlogging is more important than the inventory holding cost.

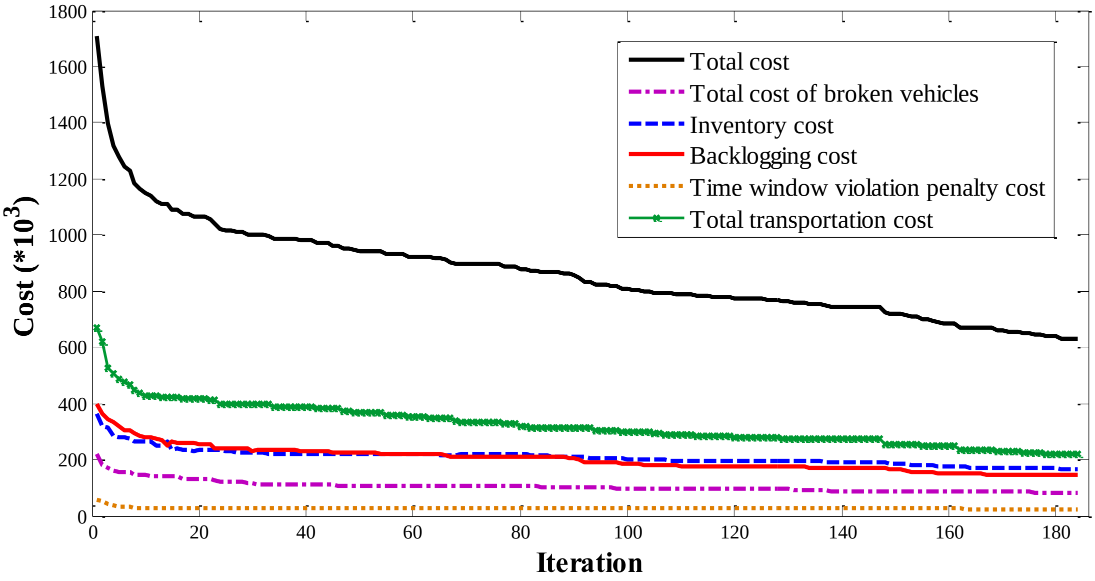


Figure 21. The trend of objective function components on different iterations resulted by ASBA.

Observation: Our observations show that in terms of the importance of cost components, practitioners should pay attention to backlogging cost, inventory holding cost, and the cost of the failed vehicles, respectively. Accordingly, managers should chiefly plan to reduce these cost terms rather than the others.

7.2. Impact of order-up-to-policy on total cost

The replenishment policy used in this paper tries to reduce the effect of disruptions such as vehicle failure by adjusting the demand order forecast by a single exponential smoothing method. In this method, the order value is adjusted based on previous period inventory level (net stock) and WIP. As illustrated in Figure 22, the effect of different values of the net stock and WIP correlation constants for all retailers (i.e. r_{1igp} and r_{2igp}) are analysed on the problems' objective function. In this regard, the sensitivity analysis was performed for the standard scope of these correlations (i.e. from 0 to 1). The values of r_{1igp} and r_{2igp} are considered the same for all retails, products and periods in this sensitivity analysis. According to the results, by increasing both factors r_{1igp} and r_{2igp}, the total cost is increased too.

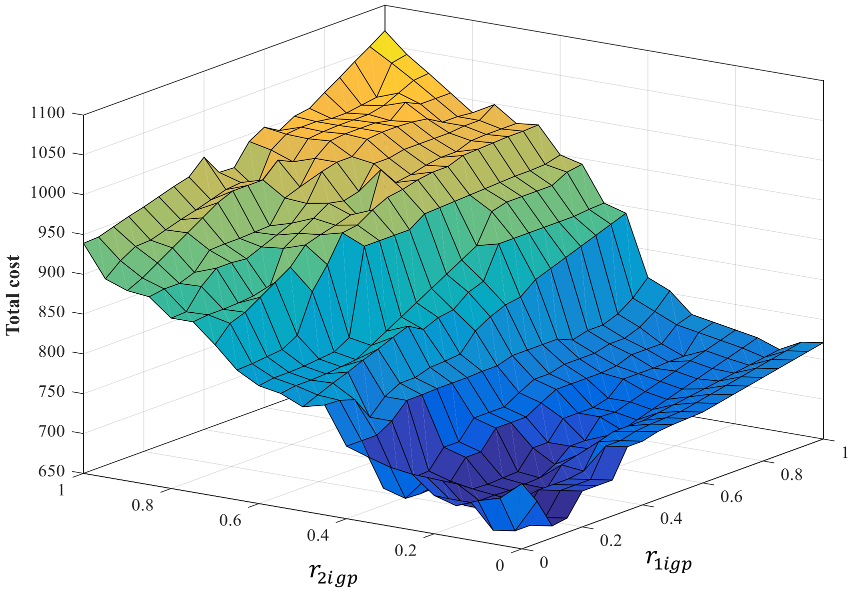
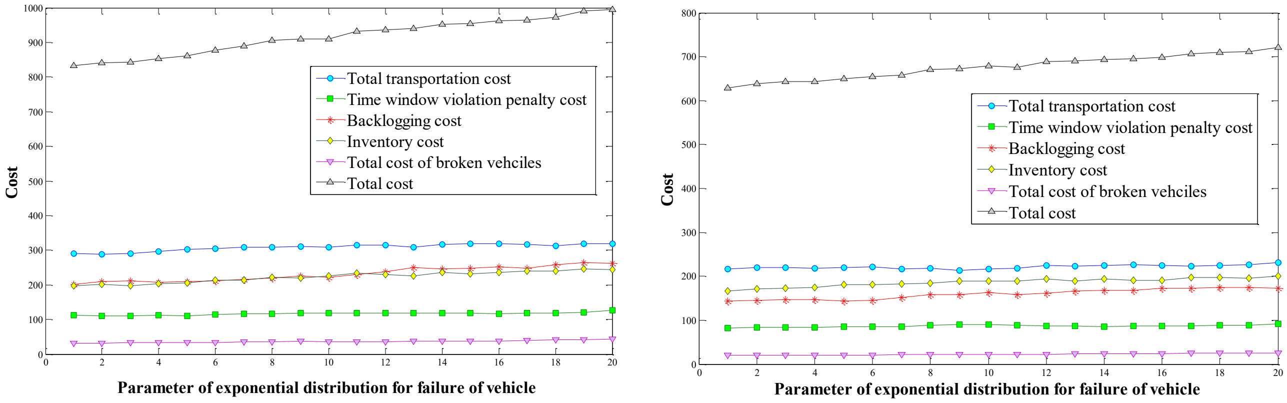


Figure 22. The effect of net stock and WIP correlation constants (r_{1igp} and r_{2igp}) on objective function.

Observation: Figure 22 shows that if r_{1igp} and r_{2igp} are ignored in the forecasting model or selected inappropriately, the total cost is increased. The appropriate value for both of these factors is between 0.05 and 0.1. Therefore, the inventory managers should consider a low effect of the net stock and WIP in the forecasting process of demand.

7.3. Impact of vehicle failure rate on cost components

The effect of different values of the vehicle failure parameter (based on exponential distribution) on the components of the objective function (i.e. total transport cost, time window violation penalty cost, backlogging cost, inventory cost, cost of the failed vehicles and total cost) is presented in Figure 23. Figures 23(a) and 23(b) are obtained from ASBA in iterations of 36th (as an example for a non-optimal solution) and 184th (as an optimum solution) respectively. According to Figure 23(a), by increasing the parameter by 50%, the total transport cost, time window violation penalty cost, backlogging cost, inventory cost, cost of the failed vehicles and total cost are increased 8%, 26%, 32%, 25%, 42% and 22.1%. The same amount of change in Figure 23(b) results in 4%, 14%, 19%, 21%, 26% and 13.8% increases in the total transportation cost, time window violation penalty cost, backlogging cost, inventory cost, cost of the failed vehicles and total cost.



(a) (b)

Figure 23. The effect of the parameter of vehicle failure time (exponential distribution) on objective function components.

Finally, it is concluded that

- by increasing the vehicle failure rate, all objective function components increase;
- the highest cost increase is related to the cost of the failed vehicles and then backlogging and inventory costs, respectively. The difference between backlogging and inventory costs is not significant;
- the proposed model reduces the shortage, the effect of increasing vehicle failure rate, the total cost and all its components because it (i) assigns the old vehicles to the paths located near the service centre and vice versa, (ii) forecasts the demand of the next period, and (iii) with appropriate sequences, assigns the retailers to the vehicles.

Observation: An increase in the vehicle failure rate highly affects backlogging cost and inventory cost, receptively, among the other cost components.

7.4. Correlation between vehicle failure probability and length of travelled paths

We solved the test problems and obtained optimal solutions for all 100 time periods. For each solution, we extracted the length of paths determined by the selected vehicles as well as λ. A large λ shows that a vehicle has a high chance of failure. Figure 24 shows the relationship between the length of paths and λ of vehicles assigned to them for period 23.

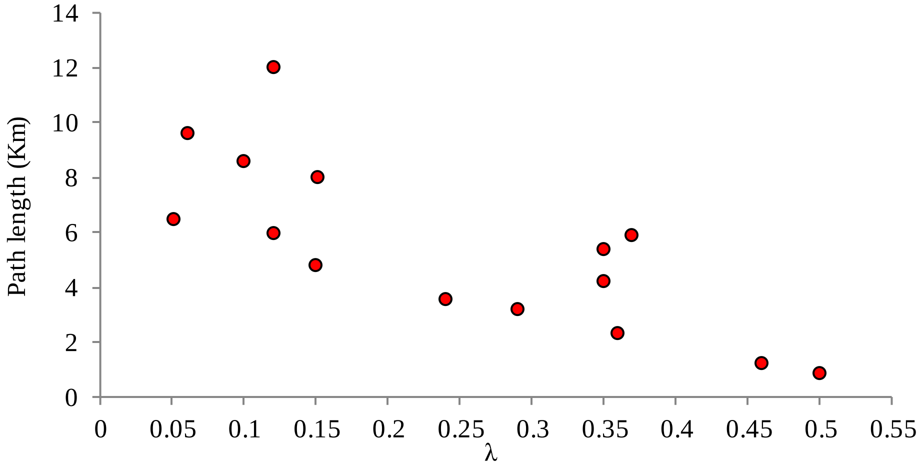


Figure 24. The relation between length of paths and the λ of vehicles assigned to them for period 23.

The figure has a negative trend direction and shows that the vehicles having a high probability of failure are assigned to short paths and, quite the opposite, the vehicles with a low probability of failure are allocated to long paths. In order to investigate the validity of this observation for the other 99 remained periods, we assess the slopes of all line trends fitted for such figures belonging to all time periods. Accordingly, Figure 25 demonstrates the percentage of different slope classes among the 100 observations.

Observation: It is clearly obvious that just over 80% of slopes were negative, meaning that vehicles with a high probability of failure are assigned to short paths and vice versa. Accordingly, it is highly recommended that practitioners assign archaic vehicles (i.e. vehicles with a high probability of failure) to short paths and new, sound vehicles are used to service remote customers.

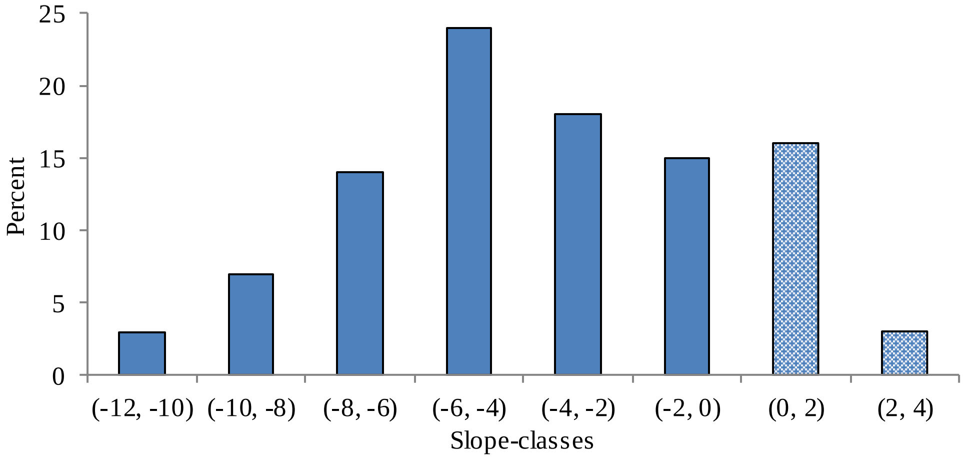


Figure 25. The relation between length of paths and the λ of vehicles assigned to them for period 23.

7.5. Relationship between vehicle failure probability and proximity to service centre

For each time period and each vehicle, we compute an average distance between the vehicle and the service centre along the vehicle's path (*DS*). *DS* for vehicle k on path p is calculated by averaging the distances of total failure points on path p, travelled by vehicle k, from the service centre. First, we observed that in each period, the average of λ for a vehicle having failure is more than the vehicles not having failure. Accordingly, Figure 26 demonstrates the average of λ for the vehicles failed in each time period as well as the average λ value for the other vehicles. Furthermore, according to Figure 27, we found that in each time period, the failed vehicles have a smaller *DS* than the others. In Figure 27, the vertical axis shows the discrepancy between *DS* of the failed and not failed vehicles in each time period.

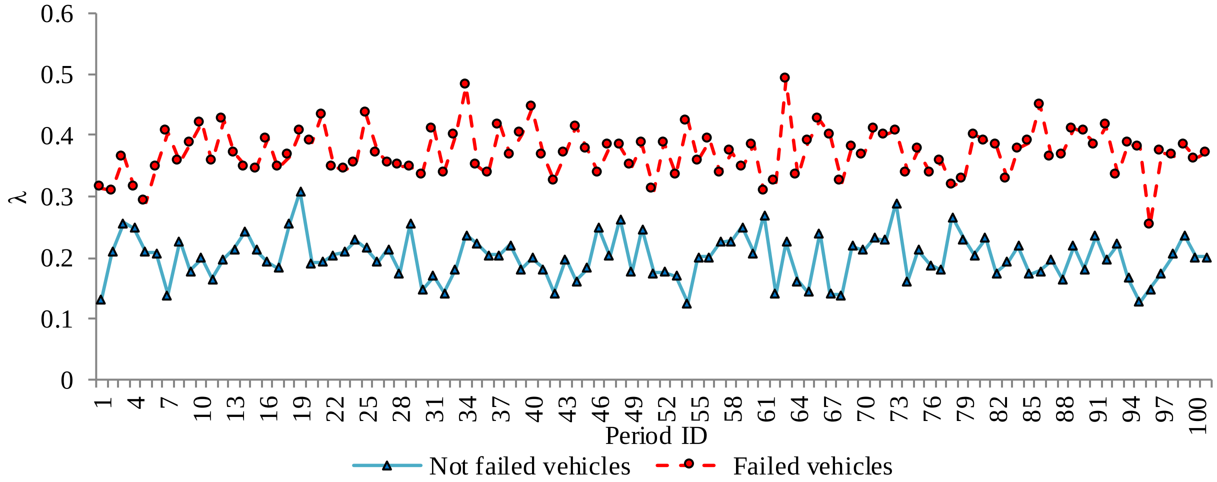


Figure 26. Average of λ for the failed vehicles and the others in each time period.

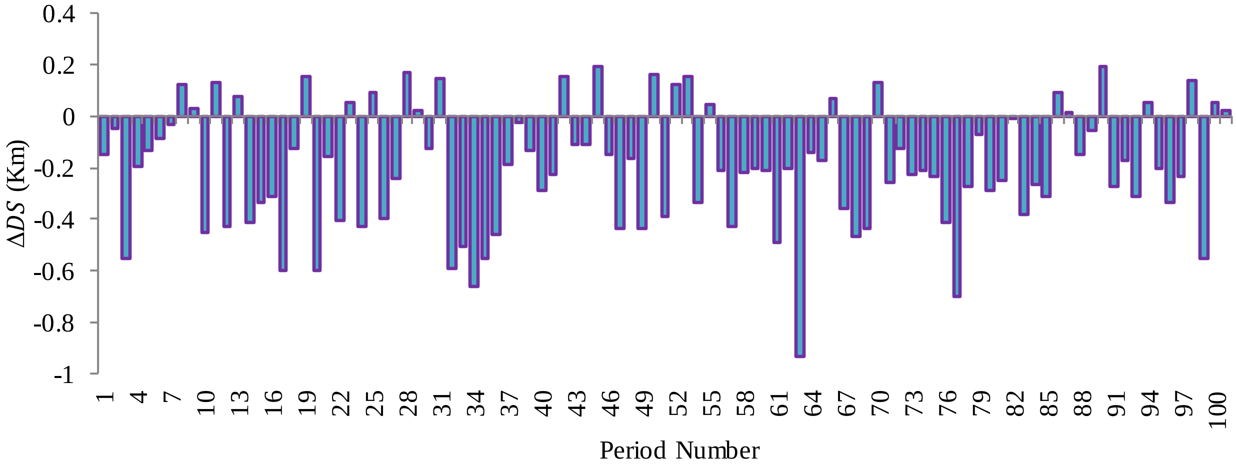


Figure 27. Discrepancies between DS of failed vehicles and the others in each time period.

Observation: Figures 26 and 27, in combination, indicate that in the optimal solution, the vehicles having a high failure probability are assigned to paths close to the service centre. Therefore, we highly recommend that decision makers assign newer vehicles in their fleet to the farther destinations.

8. CONCLUSION

In this paper, the IRP with possibility of vehicle failure is studied. The problem considers multiple periods, multiple products, holding cost, backlogging cost, soft time windows, multiple heterogeneous vehicles, deterministic-forecasted demand, variable vehicle speed (before and after failure), capacitated retailers and customer inventory policy (order-up-to policy). The vehicle failure strategy taken in this research is “fixed centre”. The problem is mathematically formulated and solved by applying a meta-heuristic algorithms (ASBA) with four scenarios. A “global war” procedure is embedded in the algorithm to avoid local optima. The comprehensive calibration methodologies RSM and MODM are employed to reach the best combination of the parameter values. In order to evaluate the effectiveness and efficiency of the proposed algorithms, some test problems are generated to help us to compare the proposed algorithms against benchmark methods. The benchmark methods are some popular algorithms, namely SBA, ICA, HGS and HPV, and a suggested lower bound. We successfully implemented the algorithm on problems with up to 320 retailers, 24 vehicles of different types and 10 products.

Computational experiments indicate that ASBA outperforms SBA, ICA, HGS and HPV in total cost performance measure and that HPV outperforms the others in terms of computational time. The hybrid-ICA-frog-based policy is more effective than other scenarios in ASBA construction. Our ASBA can solve large-sized IRPTWs subject to vehicle failure. We observe that the global war procedure can preserve premature convergence to improve the performance of this algorithm. Furthermore, ASBA with hybrid-ICA-frog-based initialisation policy can lead to better solutions compared to other initialisation scenarios. Figure 28 offers a decision-making support tool for planners based on this research.

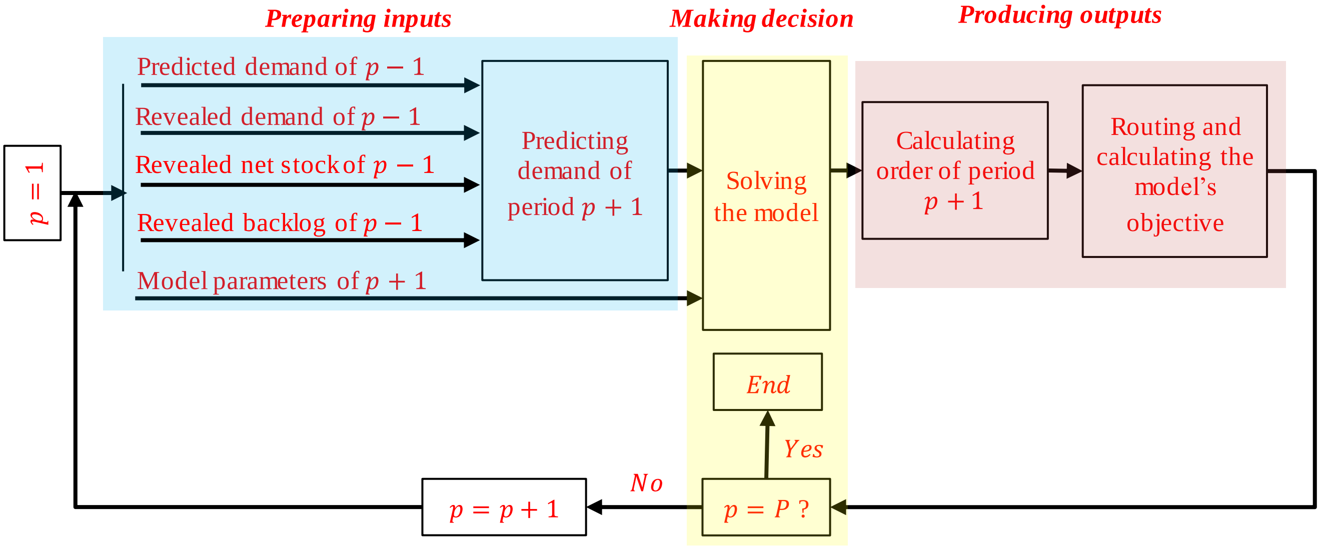

Preparing inputs
Making decision
Producing outputs
$p = 1$
Predicted demand of $p - 1$
Revealed demand of $p - 1$
Revealed net stock of $p - 1$
Revealed backlog of $p - 1$
Model parameters of $p + 1$
Predicting demand of period $p + 1$
Solving the model
Calculating order of period $p + 1$
Routing and calculating the model's objective
End
Yes
No
$p = p + 1$
$p = P$?

Figure 28. The process is taken into our model for predicting order of retailers and delivering them.

When it comes to insights for practitioners, we learned a number of lessons. First, practitioners should adjust their plans for reducing their total cost with a special focus on backlogging cost, inventory holding cost, and the cost of the failed vehicles, respectively. Second, ignoring or inappropriately applying the amount of inventory level (net stock) and WIP in forecasting inventory orders leads to increasing total costs. We observed if the inventory mangers consider a low effect of the net stock and WIP on the forecasting process of demand, the minimum value of total cost will be resulted. Third, increasing the probability of vehicle failure (through increasing the parameter of CDF) leads to increasing the terms of the objective function (total cost) differently. Among the different terms of cost, the highest cost increases are related to the cost of the failed vehicles and then backlogging and inventory costs, respectively. Fourth, practitioners intending to reduce total cost should assign vehicles with high probability of failure to short paths. Furthermore, for serving remote costumers, using vehicles with low failure probability is advised. Last but not least, assigning vehicles with high probability of failure to some paths that have less distance to a service centre is in line with the optimal solution.

There are some problems in the area that should be considered in future research. Considering the assumptions of this research (such as there only being one depot or supplier) can be a useful starting point. For example, in this research the location of the vehicle service centre is given while we may encounter some applications in which we need to identify the location of the service centre. Moreover, considering more than one fixed-centre service for repairing the failed vehicles, rerouting with repairing approaches (as depicted in Figure 1) or a mixture of them can be interesting. While the current solution techniques of this paper can be applied to other variations of the VRP (e.g., the initialisation process), other metaheuristic approaches can still be proposed and compared with our approaches. Finally, some areas are not well-covered in the literature of the IRP. For example, there is only one study in IRPs considering the time windows (Liu and Lee 2011). As it only considers soft time windows, considering hard time windows is a gap with many real-life applications.

REFERENCES

Abdelmaguid, T.F., Dessouky, M.M. 2006. A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research* 44(21) 4445-4464.

Abdelmaguid, T.F., Dessouky, M.M., Ordóñez, F. 2009. Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering* 56(4) 1519-1534.

Afzalan, E., Taghikhani, M., Sedighizadeh, M. 2012. Optimal Placement and Sizing of DG in Radial Distribution Networks Using SFLA. *International Journal of Energy Engineering* 2(3) 73-77.

Aghezzaf, E.-H., Raa, B., Van Landeghem, H. 2006. Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research* 169(3) 1048-1063.

Ahmadi-Javid, A., Seddighi, A.H. 2013. A location-routing problem with disruption risk. *Transportation Research Part E* 53(1) 63-82.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A. 2010. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research* 37(9) 1515-1536.

Anily, S., Federgruen, A., 1990. One warehouse multiple retailer systems with vehicle routing costs. *Management Science* 36(1) 92-114.

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G. 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science* 41(3) 382-391.

Arumugam, M.S., Rao, M. 2007. On a class of hybrid systems via a novel approach for real-coded genetic algorithm with hybrid selection. *International Journal of Information Technology & Decision Making* 6(02) 315-332.

Atashpaz-Gargari, E., Lucas, C. 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *Evolutionary computation*, IEEE Congress, 4661-4667.

Back, T., Hammel, U., Schwefel, H.P., 1997. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation* 1(1) 3-17.

Bagher, M., Zandieh, M., Farsijani, H. 2011. Balancing of stochastic U-type assembly lines: an imperialist competitive algorithm. *The International Journal of Advanced Manufacturing Technology* 54(1) 271-285.

Barnes-Schuster, D., Bassok, Y., 1997. Direct shipping and the dynamic single-depot/multi-retailer inventory system. *European Journal of Operational Research* 101(3) 509-518.

Beheshti, A.K., Hejazi, S., Mirmohammadi, S. 2014. A Hybrid Metaheuristic Algorithm for the Vehicle Routing Problem with Delivery Time Cost. *International Journal of Applied* 4(3) 49-63.

Behnamian, J., Fatemi Ghomi, S. 2011. Hybrid flowshop scheduling with machine and resource-dependent processing times. *Applied Mathematical Modelling* 35(3) 1107-1123.

Behnamian, J., Fatemi Ghomi, S., Zandieh, M. 2009. A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert Systems with Applications* 36(8) 11057-11069.

Berman, O., Larson, R.C. 2001. Deliveries in an inventory/routing problem using stochastic dynamic programming. *Transportation Science* 35(2) 192-213.

Bertazzi, L., Bosco, A., Guerriero, F., Laganà, D. 2013. A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies* 27(1) 89-107.

Bertazzi, L., Paletta, G., Speranza, M.G. 2002. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science* 36(1) 119-132.

Cannella, S., Ciancimino, E. 2010. On the bullwhip avoidance phase: supply chain collaboration and order smoothing. *International Journal of Production Research* 48(22) 6739-6776.

Carter, M.W., Farvolden, J.M., Laporte, G., Xu, J., 1995. Solving an integrated logistics problem arising in grocery distribution. *INFOR: Information Systems and Operational Research* 34(4) 290-306.

Çetinkaya, S., Lee, C.-Y. 2000. Stock replenishment and shipment scheduling for vendor-managed inventory systems. *Management Science* 46(2) 217-232.

Chandra, P., 1993. A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the operational research society* 44(7) 681-692.

Chang, P.C., Chen, S.H., Fan, C.Y. 2009. A hybrid electromagnetism-like algorithm for single machine scheduling problem. *Expert Systems with Applications* 36(2) 1259-1267.

Chen, F., Ryan, J.K., Simchi-Levi, D. 2000. The impact of exponential smoothing forecasts on the bullwhip effect. *Naval Research Logistics* 47(4) 269-286.

Chen, Y.F., Disney, S.M. 2007. The myopic order-up-to policy with a proportional feedback controller. *International Journal of Production Research* 45(2) 351-368.

Cheng, C., Yang, P., Qi, M., Rousseau, L.-M. 2017. Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review* 97(1) 97-112.

Chitsaz, M., Divsalar, A., Vansteenwegen, P. 2016. A two-phase algorithm for the cyclic inventory routing problem. *European Journal of Operational Research* 254(2) 410-426.

Chopra, S., Sodhi, M.S. 2004. Managing risk to avoid supply-chain breakdown. *MIT Sloan management review* 46(1) 53-61.

Coelho, L.C., Cordeau, J.-F., Laporte, G. 2012. The inventory-routing problem with transshipment. *Computers & Operations Research* 39(11) 2537-2548.

Coelho, L.C., Laporte, G. 2014. Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics* 155(1) 391-397.

Cordeau, J.F., Laporte, G. and Mercier, A. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52(8) 928-936.

Derringer, G., Suich, R., 1980. Simultaneous optimization of several response variables. *Journal of quality technology* 12(4) 214-219.

Devika, K., Jafarian, A., Hassanzadeh, A., Khodaverdi, R. 2016. Optimizing of bullwhip effect and net stock amplification in three-echelon supply chains using evolutionary multi-objective metaheuristics. *Annals of Operations Research* 242(2) 457-487.

Devika, K., Jafarian, A., Nourbakhsh, V. 2014. Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *European Journal of Operational Research* 235(3) 594-615.

Dondo, R.G., Cerdá, J. 2009. A hybrid local improvement algorithm for large-scale multi-depot vehicle routing problems with time windows. *Computers & Chemical Engineering* 33(2) 513-530.

Dror, M., Ball, M., Golden, B., 1985. A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research* 4(1) 1-23.

Elmi, A., Solimanpur, M., Topaloglu, S. 2011. A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. *Computers & Industrial Engineering* 61(1) 171-178.

Eusuff, M.M., Lansey, K.E. 2003. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management* 129(3) 210-225.

Fagerholt, K. 2004. A computer-based decision support system for vessel fleet scheduling—experience and future research. *Decision Support Systems* 37(1) 35-47.

Federgruen, A., Prastacos, G., Zipkin, P.H., 1986. An allocation and distribution model for perishable products. *Operations Research* 34(1) 75-82.

Federgruen, A., Zipkin, P., 1984. A combined vehicle routing and inventory allocation problem. *Operations Research* 32(5) 1019-1037.

Forouharfard, S., Zandieh, M. 2010. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. *The International Journal of Advanced Manufacturing Technology* 51(9) 1179-1193.

Fu, M. C. 2002. Optimization for simulation: Theory vs. practice. INFORMS Journal on Computing 14(3) 192–215.

Gaafar, L., Masoud, S. 2005. Genetic algorithms and simulated annealing for scheduling in agile manufacturing. *International Journal of Production Research* 43(14) 3069-3085.

Gaur, V., Fisher, M.L. 2004. A periodic inventory routing problem at a supermarket chain. *Operations Research* 52(6) 813-822.

Govindan, K., Jafarian, A., Khodaverdi, R. & Devika, K. 2014. Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics*, 152(1) 9-28.

Grosan, C., Abraham, A. 2007. Hybrid evolutionary algorithms: methodologies, architectures, and reviews. *Hybrid evolutionary algorithms*, Springer, Berlin, Heidelberg, 1-17.

Homberger, J. & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. INFOR: *Information Systems and Operational Research*, 37(3) 297-318.

Hansen, P., Mladenović, N. 2001. Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130(3) 449-467.

Harwood, D.W., Russell, E.R., Viner, J.G., 1989. Characteristics of accidents and incidents in highway transportation of hazardous materials. *Transportation Research Record* 1245(1) 23-33.

Hemmelmayr, V., Doerner, K.F., Hartl, R.F., Savelsbergh, M.W. 2009. Delivery strategies for blood products supplies. *OR spectrum* 31(4) 707-725.

Ho, W., Ho, G.T.S., Ji, P., Lau, H.C.W. 2008. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence* 21(4) 548-557.

Holland, J.H., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. *MIT Press*, 1-217.

Hu, X., Sun, L. 2012. Knowledge-based modeling for disruption management in urban distribution. *Journal of the Operational Research Society Expert Systems* 39(1) 906-916.

Huang, S.-H., Lin, P.-C. 2010. A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review* 46(5) 598-611.

Inuiguchi, M., J. Ramik. 2000. Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem, *Fuzzy Sets and Systems* 111(1) 3–28.

Jbili, S., Chelbi, A., Radhoui, M. and Kessentini, M. 2018. Integrated strategy of Vehicle Routing and Maintenance. *Reliability Engineering & System Safety* 170(1) 202-214.

Jetlund, A.S., Karimi, I. 2004. Improving the logistics of multi-compartment chemical tankers. *Computers & Chemical Engineering* 28(8) 1267-1283.

Jia, T., Li, X., Wang, N., Li, R. 2014. Integrated inventory routing problem with quality time windows and loading cost for deteriorating items under discrete time. *Mathematical Problems in Engineering*, 2014(1), 1-14.

Karaoglan, I., Altiparmak, F. 2010. A hybrid genetic algorithm for the location-routing problem with simultaneous pickup and delivery. *Computers and Industrial Engineering (CIE) 2010 40th International Conference on*, IEEE, 1-6.

Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks IV, Australia, 1942–1948.

Kennedy, J., Eberhart, R. (1995). "Particle Swarm Optimization". doi:10.1109/ICNN.1995.488968.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *science* 220(4598) 671-680.

Kleywegt, A.J., Nori, V.S., Savelsbergh, M.W. 2004. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science* 38(1) 42-70.

Klibi, W., Martel, A. 2012. Modeling approaches for the design of resilient supply networks under disruptions. *International Journal of Production Economics* 135(2) 882-898.

Le, T., Diabat, A., Richard, J.P., Yih, Y. 2013. A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. Optimization Letters 7(7), 1481-1502.

Leung, S.C.H., Tsang, S.O., Ng, W.L. & Wu, Y. 2007 A robust optimization model for multi-site production planning in an uncertain environment. *European Journal of Operational Research* 181(1), 224–238.

Li, J.Q., Mirchandani, P.B., Borenstein, D. 2007a. A decision support system for the single-depot vehicle rescheduling problem. *Computers & Operations Research* 34(1) 1008-1032.

Li, J.Q., Mirchandani, P.B., Borenstein, D. 2007b. Vehicle rescheduling problem: model and algorithms. *Networks* 50(1) 211-229.

Li, J.Q., Mirchandani, P.B., Borenstein, D. 2008a. Truck schedule recovery for solid waste collection in Porto Alegre, Brazil. *International Transactions in Operations Research* 15(1) 565-582.

Li, J.Q., Mirchandani, P.B. and Borenstein, D. 2009a. Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research* 194(3) 711-727.

Li, J.Q., Mirchandani, P.B. and Borenstein, D. 2009b. A Lagrangian heuristic for the real-time vehicle rescheduling problem. *Transportation Research Part E: Logistics and Transportation Review* 45(3) 419-433.

Li, J.-A., Wu, Y., Lai, K.K., Liu, K. 2008b. Replenishment routing problems between a single supplier and multiple retailers with direct delivery. *European Journal of Operational Research* 190(2) 412-420.

Li, M., Wang, Z., Chan, F.T. 2016. A robust inventory routing policy under inventory inaccuracy and replenishment lead-time. *Transportation Research Part E: Logistics and Transportation Review* 91(1) 290-305.

Lin, S.W., Gupta, J., Ying, K.C., Lee, Z.J. 2009. Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times. *International Journal of Production Research* 47(12) 3205-3217.

Liu, F.H. Shen, S.Y., 1999. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research society* 50(7) 721-732.

Liu, S.-C., Lee, W.-T. 2011. A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications* 38(10) 13223-13231.

Liu, S.-C., Lu, M.-C., Chung, C.-H. 2016. A hybrid heuristic method for the periodic inventory routing problem. *The International Journal of Advanced Manufacturing Technology* 85(9-12) 2345-2352.

Liu, S., Lin, C. 2005. A heuristic method for the combined location routing and inventory problem. *The International Journal of Advanced Manufacturing Technology* 26(4) 372-381.

Mamasis, K., Minis, I., Dikas, G. 2013. Managing vehicle breakdown incidents during urban distribution of a common product. *Journal of the Operational Research Society* 64(1) 925-937.

Mansouri, S.A. 2005. A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operational Research* 167(3) 696-716.

McMullen, P.R., Frazier, G.V. 2000. A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line. *IIE transactions* 32(8) 679-686.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics* 21(6) 1087.

Minis, I., Mamasis, K., Zeimpekis, V. 2012. Real-time management of vehicle breakdowns in urban freight distribution. *Journal of Heuristics* 18(1) 375-400.

Mirzaei, S., Seifi, A. 2015. Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering* 87(1) 213-227.

Mladenovic, N., Hansen, P., 1997. Variable neighborhood search. *Computers & Operations Research* 24(11) 1097-1100.

Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S. 2012. A variable neighborhood search for the multi-product inventory routing problem. *Electronic Notes in Discrete Mathematics* 39(1) 91-98.

Moin, N.H., Salhi, S. 2007. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society* 58(9) 1185-1194.

Moin, N.H., Salhi, S., Aziz, N. 2011. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics* 133(1) 334-343.

Montgomery, D.C., Myers, R.H. 2002. Response surface methodology: process and product optimization using designed experiments. *John Wiley & Sons, New York*.

Moslemi, H., Zandieh, M. 2011. Comparisons of some improving strategies on MOPSO for multi-objective (r, Q) inventory system. *Expert Systems with Applications*. 38(10) 12051–12057.

Mozafari, H., Ayob, A., Kamali, F. 2012. Optimization of functional graded plates for buckling load by using imperialist competitive algorithm. *Procedia Technology* 1(1) 144-152.

Mu, Q., Eglese, R.W. 2013. Disrupted capacitated vehicle routing problem with order release delay. *Annals of Operations Research* 207(1) 201-216.

Mu, Q., Fu, Z., Lysgaard, J., Eglese, R. 2011. Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society* 62(4) 742-749.

Mula, J., R. Poler, J.P. Garcia. 2006. MRP with flexible constraints: a fuzzy mathematical programming approach, *Fuzzy Sets and Systems* 157(1) 74–97.

Mulvey, J.M., Vanderbei, R.J., Zenios, S.A. 1995. Robust optimization of large-scale systems. *Operations Research* 43(2) 264-281.

Neter, J., Wasserman, W., Kutner, M.H., 1996. Applied linear regression models. *Irwin Homewood*.

Ngai, E.W.T., Leung, T.K.P., Wong, Y.H., Lee, M.C.M., Chai, P.Y.F., Choi, Y.S. 2012. Design and development of a context aware decision support system for real-time accident handling in logistics. *Decision Support Systems* 52(1) 816-827.

Nguyen, V.P., Prins, C., Prodhon, C. 2012. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *European Journal of Operational Research* 216(1) 113-126.

Niakan, F., Rahimi, M. 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. *Transportation Research Part E: Logistics and Transportation Review* 80(1) 74-94.

Panigrahi, B.K., Shi, Y., Lim, M.H. 2011. Handbook of swarm intelligence: Concepts, principles and applications. *Springer Science & Business Media*.

Paul, S.K., Sarker, R., Essam, D. 2015. Managing disruption in an imperfect production–inventory system. *Computers & Industrial Engineering* 84(1) 101-112.

Popović, D., Vidović, M., Radivojević, G. 2012. Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications* 39(18) 13390-13398.

Raa, B., Aghezzaf, E.-H. 2008. Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics* 112(1) 255-263.

Raa, B., Aghezzaf, E.-H. 2009. A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research* 192(2) 429-441.

Rabiee, M., Zandieh, M., Jafarian, A. 2012. Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics. *International Journal of Production Research* 50(24) 7428-7446.

Ramezani, F., Lotfi, S. 2013. Social-based algorithm (SBA). *Applied Soft Computing* 13(5) 2837-2856.

Ramezani, F., Nikoo, M., Nikoo, M. 2015. Artificial neural network weights optimization based on social-based algorithm to realize sediment over the river. *Soft Computing* 19(2) 375-387.

Santos, E., Ochi, L.S., Simonetti, L., González, P.H. 2016. A Hybrid Heuristic based on Iterated Local Search for Multivehicle Inventory Routing Problem. *Electronic Notes in Discrete Mathematics* 52(1) 197-204.

Schildbach, G., M. Morari. 2016. Scenario-based model predictive control for multi-echelon supply chain management, *European Journal of Operational Research* 252(2) 540–549.

Sedighizadeh, D., Masehian, E. 2009. Particle swarm optimization methods, taxonomy and applications. *International Journal of Computer Theory and Engineering* 1(5) 1793-8201.

Sevkli, M., Aydin, M. 2006. A variable neighbourhood search algorithm for job shop scheduling problems. *European Conference on Evolutionary Computation in Combinatorial Optimization, Springer, Berlin, Heidelberg*, 261-271.

Shen, Q., Chu, F., Chen, H. 2011. A Lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering* 35(10) 2113-2123.

Silver, E.A., Pyke, D.F., Peterson, R., 1998. Inventory management and production planning and scheduling. *Wiley New York*.

Singh, T., Arbogast, J.E., Neagu, N. 2015. An incremental approach using local-search heuristic for inventory routing problem in industrial gases. *Computers & Chemical Engineering* 80(1) 199-210.

Song, J.-H., Furman, K.C. 2013. A maritime inventory routing problem: Practical approach. *Computers & Operations Research* 40(3) 657-665.

Soysal, M. 2016. Closed-loop Inventory Routing Problem for returnable transport items. *Transportation Research Part D: Transport and Environment* 48(1) 31-45.

Soysal, M., Bloemhof-Ruwaard, J.M., Haijema, R., van der Vorst, J.G. 2015. Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics* 164(1), 118-133.

Soysal, M., Bloemhof-Ruwaard, J.M., Haijema, R. and van der Vorst, J.G. 2018. Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research* 89(1) 168-182.

Stålhane, M., Rakke, J.G., Moe, C.R., Andersson, H., Christiansen, M., Fagerholt, K. 2012. A construction and improvement heuristic for a liquefied natural gas inventory routing problem. *Computers & Industrial Engineering* 62(1) 245-255.

Tarantilis, C., Kiranoudis, C. 2001. A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of* *food engineering* 50(1) 1-9.

Tavakkoli-Moghaddam, R., Khalili, M., Naderi, B. 2009. A hybridization of simulated annealing and electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness. *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 13(10) 995-1006.

Yu, C.S. & Li, H. 2000. A robust optimization model for stochastic logistic problems. *International Journal of Production Economics* 64(1-3) 385-397.

Viswanathan, S., Mathur, K., 1997. Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Management Science* 43(3) 294-312.

Wan, X., J.F. Pekny, G.V. Reklaitis. 2005. Simulation-based optimization with surrogate models-Application to supply chain management. *Computers and Chemical Engineering* 29(6) 1317–1328.

Wang, G.-j., Zhang, Y.-B., Chen, J.-W. 2011. A novel algorithm to solve the vehicle routing problem with time windows: Imperialist competitive algorithm. *Advances in Information Sciences and Service Sciences* 3(5) 108-116.

Wang, X., Ruan, J., Shi, Y. 2012. A recovery model for combinational disruptions in logistics delivery: Considering the real-world participators. *International Journal of Production Economics* 140(1) 508-520.

Wang, X., Wang, Z., Wu, X., Hu, X. 2009. A Model and an Improved Genetic Algorithm for the Vehicle Routing Problem with Break-Down Vehicles. Fourth International Conference on *Innovative Computing, Information and Control (ICICIC)* 696-699.

Wang, X., Wu, X. and Hu, X. 2010. A study of urgency vehicle routing disruption management problem. In Information Engineering (ICIE) 2010 *WASE International Conference (IEEE)* 3(1) 452-455.

Yang, W., Wang, Zh. 2009. Rescue Method for Vehicle Routing Problem under Vehicle Disruptions. The 1st International Conference on *Information Science and Engineering (ICISE2009)*, 2345-2348.

Yu, Y., Chen, H., Chu, F. 2008. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research* 189(3) 1022-1040.

Yu, Y., Chu, C., Chen, H., Chu, F. 2012. Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research* 197(1) 135-158.

Zhalechian, M., Tavakkoli-Moghaddam, R., Zahiri, B., Mohammadi, M. 2016. Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. *Transportation Research Part E: Logistics and Transportation Review* 89(1) 182-214.

Zhang, X., Tang, L. 2007. Disruption management for the vehicle routing problem with time windows. In: Huang, D., Heutte, L., Loog, M. (eds.) Advanced Intelligent Computing Theories and Applications with Aspects of Contemporary Intelligent Computing Techniques. Proceedings of the Third International Conference on Intelligent Computing 2(1) 225-234.

Zhao, Q.-H., Wang, S.-Y., Lai, K.K. 2007. A partition approach to the inventory/routing problem. *European Journal of Operational Research* 177(2) 786-802.

Zhao, Q.H., Chen, S., Zang, C.X. 2008. Model and algorithm for inventory/routing decision in a three-echelon logistics system. *European Journal of Operational Research* 191(3) 623-635.

Zhou, E., Chen, X. 2010. A new population-based simulated annealing algorithm. *IEEE Transactions on Evolutionary Computation, Simulation Conference* (*WSC*), 1211-1222.

Zhu, S., Levinson, D.M. 2012. Disruptions to transportation networks: a review, *Network reliability in practice. Springer, New York*, 5-20.

Zipkin, P.H. 2000. Foundations of inventory management. *McGraw-Hill Boston, MA.*

APPENDIX

```
Vehicles failure - related calculation procedure()
Receive inputs { Parameters ($\lambda_{m_k kp}, \eta_{m_k ktp}, \Delta_{m_k ktp}, \rho_k, c_k, u_{m_k kp}, \gamma_{m_k kp}, \vartheta_{im_k kp}, \phi_{m_k kp}$);
               { $Variables\ (z_{im_k kp}, x_{ijm_k kp} : i, j \in A)$;
               { $l_{ij}, l'_{ij}, l''_{ij}, ls_j : i, j \in A$ //All distances can either calculate exactly or estimate with a suitable distance function;
Set { $p = 0, t = 0, \beta_{m_k ktp} = 0$;
    { $TCC = \{\}$ // Total moving or carrying costs, $TRC = \{\}$//Total repair costs;
for $p = 1 : |P|$
    for $k = 1 : |K|$
        for $m_k = 1 : |M_k|$
            for $i = 1 : |A|$
                If $z_{im_k kp} = 1$
                    for $t = 1 : |T|$
                        If $\Delta_{m_k ktp} > \eta_{m_k ktp}$ and $\sum_{t=\{1,\ldots,t-1\}} \beta_{m_k ktp} = 0$// The vehicle is broken done
                            $\beta_{m_k ktp} = 1$;
                        endIf
                    endfor
                        for $j = 1 : |A|$
                            If $x_{ijm_k kp} = 1$
                                find($x'_{ijm_k kp} | x'_{ijm_k kp} = 1$);
                            endIf
                        endfor
                endIf
            endfor
        endfor
    endfor
endfor
$TCC = \sum_{p \in P} \sum_{k \in K} \sum_{m_k \in M_k} \sum_{i \in A} \sum_{j \in A} x_{ijm_k kp} \left[ \left(1 - x'_{ijm_k kp}\right) c_k l_{ij} + x'_{ijm_k kp} \left( c_k l'_{ij} + \rho_k l''_{ij} + c_k ls_j \right) \right]$;
$TRC = \sum_{p \in P} \sum_{k \in K} \sum_{m_k \in M_k} \sum_{t \in T} \beta_{m_k ktp} \psi_{kp}$ ;
```

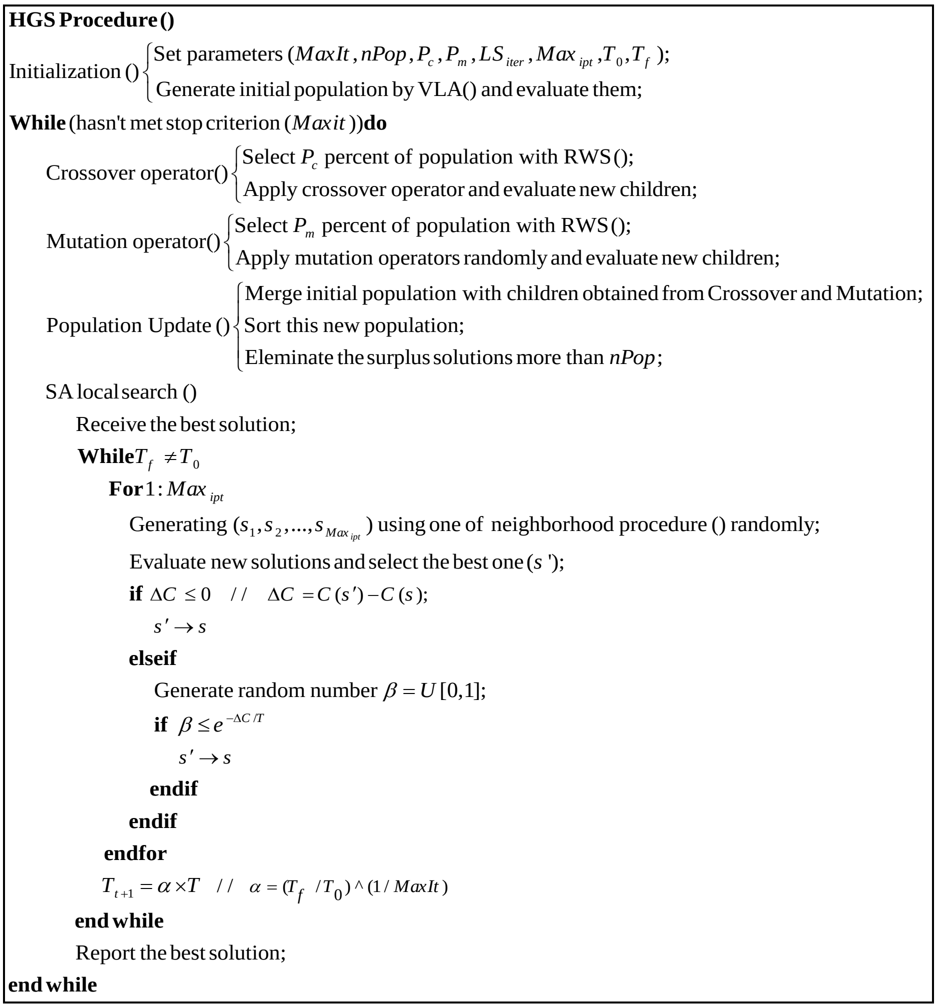
Figure A1. The pseudo-code of vehicles failure-related calculation procedure.

```
VLA procedure ()
Step 1: Calculate the total orders of all products $g \in G$ dispatched by all retailers in each
    period $p \in P$ ($TO_p$) as:
; $\hat{d}_{igp} = \alpha d_{igp-1} + (1-\alpha)\hat{d}_{igp-1}$   //$\alpha = 0.25$
    $TO_p = \sum_{i \in N} \sum_{g \in G} \min(\hat{d}_{igp}, O_{igp}),\ \forall p \in P$
Step 2: Put all the vehicles to TNV // $TVN = \{M_1, M_2 ..., M_k\}, M_k = \{1_k, 2_k, ...m_k\}$;
Step 3: Calculate the total capacity of required vehicle (TCV) for each period as:
    For r=1:| $TNV$ |
        $RV = \phi$ // Required vehicle (RV)
        Select a vehicle (v) randomly from $TNV$ and eliminate that from $TNV$
            $RV = RV \cup v$ and $TVN = TVN - v$
        Add the capacity of vehicle (v) to TCV
            $TCV_p = TCV_p + q_k$
        Check if selected vehicles capacity can satisfy the received orders, break the loop
            If $TCV_p \geq TO_p$; Break
    End for
Step 4: Calculate the expected load (EL) required for each vehicle in period $p \in P$ as:
    $EL_v = \left\lceil {TO_p}/{TCV_p} \right\rceil \times q_k\ ,\ \forall v \in RV$
Step 5: Assign the retailers to vehicles such that the following equation is minimised:
    $z = \sum_{p \in P} \sum_{v \in RV} \max(\sum_{i \in S_{m_k k}} \sum_{g \in G} d_{igp} - EL_v, 0)\quad \forall S_{m_k k} \subset V\ ,\ v \in RV$
```

Figure A2. The VLA procedure in the pseudo-code.

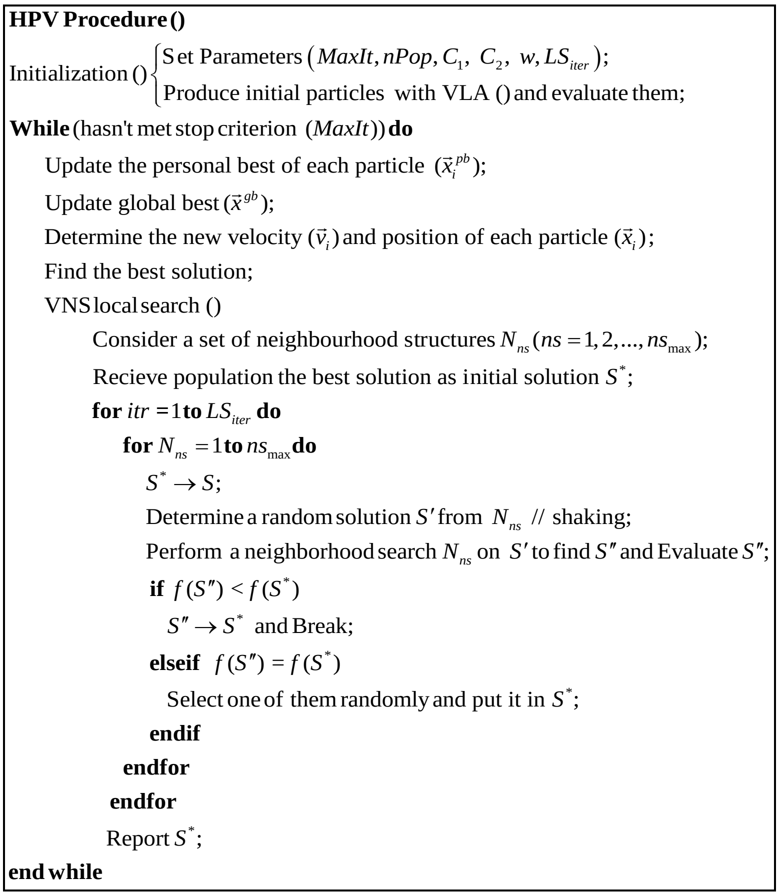
```
ASBA procedure ()
Initialization () {
    Set parameters ($nPop, N_{country}, N_{imp}, N_d, \xi, P_i, P_e, I_{GW}, N_{GW}$);
    Generating initial peoples by VLA() and evaluate them;
Form initial countries () {
    Select the most powerful $N_c$ people as leaders;
    Randomly allocate the remain people to countries ased on leaders power;
Form initial empires () {
    Choice most powerful $N_{imp}$ countries as the imperialists;
    Assign other countries to imperialists based on imperialists power;
for 1: $N_c$
  Do Crossover () and Mutation () on person(s) and replace the better solutions;
  for 1: $N_{imp}$
    Move people toward thier leader (People assimilaiton);
    Revolution among people;
    Move leaders / people toward imperialist / leaders (Countries assimilaiton);
    Revolution among countries;
    If the cost of colony is lower than own imperialist;
      Exchanging positions of the imperialist and a colony;
    endif
    Imperialistic competition () {
        Calculate total power of the empires;
        Select the weakest colony of weakest empire and assign it to a strange empire;
        Eliminate the powerless empires (the imperialist with no colony);
  endfor
  Global war () {
      Form new population as the same number as PopSize and evaluate them;
      Merge new and old population;
      Sort merged population and eleminate PopSize extra presons;
endfor
```

Figure A3. The adapted social-based algorithm (ASBA) in pseudo code.



```
HGS Procedure ()
Initialization () { Set parameters ($MaxIt, nPop, P_c, P_m, LS_{iter}, Max_{ipt}, T_0, T_f$);
                  { Generate initial population by VLA() and evaluate them;
While (hasn't met stop criterion ($Maxit$)) do
    Crossover operator() { Select $P_c$ percent of population with RWS();
                         { Apply crossover operator and evaluate new children;
    Mutation operator() { Select $P_m$ percent of population with RWS();
                        { Apply mutation operators randomly and evaluate new children;
    Population Update () { Merge initial population with children obtained from Crossover and Mutation;
                         { Sort this new population;
                         { Eleminate the surplus solutions more than $nPop$;
    SA local search ()
        Receive the best solution;
        While $T_f \neq T_0$
            For $1 : Max_{ipt}$
                Generating $(s_1, s_2, ..., s_{Max_{ipt}})$ using one of neighborhood procedure () randomly;
                Evaluate new solutions and select the best one ($s'$);
                if $\Delta C \leq 0$   //   $\Delta C = C(s') - C(s)$;
                    $s' \rightarrow s$
                elseif
                    Generate random number $\beta = U[0,1]$;
                    if $\beta \leq e^{-\Delta C / T}$
                        $s' \rightarrow s$
                    endif
                endif
            endfor
            $T_{t+1} = \alpha \times T$   //   $\alpha = (T_f / T_0) \wedge (1 / MaxIt)$
        end while
        Report the best solution;
end while
```

Figure A4. The pseudo-code of HGS.



```
HPV Procedure ()
Initialization () { Set Parameters $(MaxIt, nPop, C_1, C_2, w, LS_{iter})$;
                  { Produce initial particles with VLA () and evaluate them;
While (hasn't met stop criterion ($MaxIt$)) do
    Update the personal best of each particle ($\vec{x}_i^{pb}$);
    Update global best ($\vec{x}^{gb}$);
    Determine the new velocity ($\vec{v}_i$) and position of each particle ($\vec{x}_i$);
    Find the best solution;
    VNS local search ()
        Consider a set of neighbourhood structures $N_{ns} (ns = 1, 2, ..., ns_{\max})$;
        Recieve population the best solution as initial solution $S^*$;
        for $itr = 1$ to $LS_{iter}$ do
            for $N_{ns} = 1$ to $ns_{\max}$ do
                $S^* \rightarrow S$;
                Determine a random solution $S'$ from $N_{ns}$ // shaking;
                Perform a neighborhood search $N_{ns}$ on $S'$ to find $S''$ and Evaluate $S''$;
                if $f(S'') < f(S^*)$
                    $S'' \rightarrow S^*$ and Break;
                elseif $f(S'') = f(S^*)$
                    Select one of them randomly and put it in $S^*$;
                endif
            endfor
        endfor
        Report $S^*$;
end while
```

Figure A5. The pseudo-code of HPV.