Available online at: http://e-jurnal.lppmunsera.org/index.php/JSMI

Jurnal Sistem dan Manajemen Industri

ISSN (Print) 2580-2887 ISSN (Online) 2580-2895

Tabu search heuristic for inventory routing problem with stochastic demand and time windows

Meilinda Fitriani Nur Maghfiroh[1*], Anak Agung Ngurah Perwira Redi[2]

[1] Department of Industrial Engineering, Universitas Islam Indonesia, Jl. Kaliurang KM 14.5 Sleman Yogyakarta, 55584, Indonesia

[2] Department of Industrial Engineering, Sampoerna University, Jl. Raya Pasar Minggu No.Kav. 16, Jakarta 12780, Indonesia

ARTICLE INFORMATION

Article history:

Received: June 1, 2022
Revised: October 10, 2022
Accepted: November 19, 2022

Keywords:

Inventory routing problem
Stochastic demand
Time windows
Tabu search
Variable neighborhood descent

*Corresponding Author

Meilinda Fitriani Nur Maghfiroh
E-mail:
meilinda.maghfiroh@uii.ac.id

ABSTRACT

This study proposes the hybridization of tabu search (TS) and variable neighbourhood descent (VND) for solving the Inventory Routing Problems with Stochastic Demand and Time Windows (IRPSDTW). Vendor Managed Inventory (VMI) is among the most used approaches for managing supply chains comprising multiple stakeholders, and implementing VMI require addressing the Inventory Routing Problem (IRP). Considering practical constraints related to demand uncertainty and time constraint, the proposed model combines multi-item replenishment schedules with unknown demand to arrange delivery paths, where the actual demand amount is only known upon arrival at a customer location with a time limit. The proposed method starts from the initial solution that considers the time windows and uses the TS method to solve the problem. As an extension, the VND is conducted to jump the solution from its local optimal. The results show that the proposed method can solve the IRPSDTW, especially for uniformly distributed customer locations.

1. INTRODUCTION

Integration and coordination of different supply chain management (SCM) components have become crucial to connect organizational units and coordinate material, information, and financial flows to increase the supply chain's competitiveness [1]. Vendor-managed inventory (VMI) is being implemented to support SCM in handling the complete component. VMI refers to a supplier's capability to monitor its retailers' inventory levels. VMI is also free to select when and how much stock to restock at each location/shop, allowing for more efficient supply chain operations. Under VMI, the supplier is accountable for organizing the replenishment of inventory and the selection of vehicles. The VMI concept can be modelled as an inventory routing problem (IRP) [2], allowing the optimum coordination of inventory replenishment and vehicle routing problems. IRP is an extension vehicle routing problem (VRP) in which the supplier decides when, how much, and where to deliver the goods [3].

The application of IRP has been found in different industries, including the perishable and food retail industry [4]–[8], the biogas industry [9], the petrochemical industry [10], the spare part manufacturing industry [11], and furniture industry [12]. Several extensions of IRP are also studied, including time-dependent IRP [13], multi-period IRP [14], IRP with time-varying demand [15], multi-vehicle cyclic IRP [16], IRP with split up and deliveries [17], and maritime IRP [18]. As an NPhard problem, inventory and routing problems are solved using many approaches, including exact methods and metaheuristics. Coelho et al. [19] utilized Brand and Cut to solve multi-attribute IRP and embedded it with variable neighbourhood search (VNS) as the local search. Skålnes et al. [15] also incorporated Branch and Cut to solve IRP with time-varying demand. The approach will depend on the factors involved in the problem, such as inventory policies, time horizon, demand distribution, and many others [20]. Some metaheuristics approaches include simulated annealing [16], genetic algorithm [14], insertion heuristics [21], and tabu search [22]. Matheuristics has also proven to be an efficient and robust method to solve variants of IRP [13], [18].

Among the previous IRP variants studies, one notably studied quite excessively is stochastic IRP. Stochasticity in the problem might occur due to unknown actual demand, uncertain travel times, or uncertain supply levels. Some modelled the uncertainty using random variables or different values throughout the planning horizon. In some studies, scenarios are generated to capture the uncertainty of the problem.

Huang and Lin [6] investigated a multi-product single-period IRP with unpredictable demand and stockouts. The goal is to minimize the overall cost, supplied by the sum of planned routes, recourse costs, and predicted stockout costs. The authors proposed a modified ant colony optimization to solve the problem with the modified version shows a more efficient result. An IRP with stochastic requests with specific probability and split delivery is proposed by Yu et al. [23]. The authors tried to minimize total cost and maximise customer satisfaction by proposing the split delivery strategy. The authors used a hybrid solution technique that linearized the sub-model to solve the problem. Further, Bertazzi et al. [24] investigated an IRP with stochastic demand that always needs to be satisfied. The aim is to minimize the total cost, which is calculated by adding the estimated inventory cost, penalty cost for the out-stock situation, and routing cost.

Coelho et al. [25] studied IRP with dynamic and stochastic demand. The authors provided heuristic policies where only single-vehicle transshipments are allowed. The objectives are to minimize inventory, shortage, routing, and transhipment costs. The authors proposed several strategies using a heuristics approach in which it is revealed that additional forecasting capability might help find the stochastic demand information. Furthermore, the decision on the distribution route should not be in a long planning period since it will not improve the solution.

Gruler et al. [26] studied a single-period IRP with stochastic demand and stockouts to reduce inventory and routing expenses. The authors proposed a variable neighbourhood search with simheuristic to solve the problem. Nikzad et al. [27] studied stochastic IRP in drug distribution with uncertain demand to reduce inventory, shipping, and stockout costs. The uncertainty is modelled as different demand scenarios. Two chance-constrained stochastic formulations and two-stage stochastic programming were suggested. Qu et al. propose a heuristic decomposition strategy for tackling the stochastic joint replenishment issue with several items [28]. In this study, the inventory and transportation policies were jointly examined. Another investigation was undertaken by Yang et al. [29]. They devised heuristic methods and an optimum restocking strategy using a single product for a stochastic vehicle-routing issue. In research by Archetti et al. [30], the supplier monitors each store's inventory and develops a collaborative replenishment plan in a stochastic environment where stock-outs are forbidden. In addition, Christiansen & Lysgaard [31] researched capacitated VRP with stochastic replenishment demand for a single item. The most current research formulates a fixed-fleet VRP with stochastic demands and multiple items (VRPSDMI) to choose a route with the lowest estimated total costs. In their study, a central supplier must resupply a subset of customers with uncertain demand, identify which customers to restock using viable daily route plans, and minimize the overall cost.

The variant of IPR, which consider both stochastic demand and time windows, is still hardly found, although almost similar study with lead time has been proposed by Roldan [32]. In this study, we extend the vehicle routing problem with stochastic demand and multi-item (VRPSDMI) problem proposed by Huang & Lin [6] and add the time window constraint for each retailer into the inventory routing problem with stochastic demand and time windows (IRPSDTW). Sometimes, when the vendor tries to deliver goods, the actual demand can only be known upon arrival. A stock-out will occur if the vendor fulfills their vehicle based on the maximum vehicle capacity and the actual demand on a route exceeds the prepared inventory. Consequently, the vehicle will need to return to the

http://dx.doi.org/10.30656/jsmi.v6i2.4813

depot, replenish the vehicle, and serve the rest of the retailers until their working hours end. Since specific locations are visited without previous knowledge of their inventory levels and demand, the distribution method may result in high operating costs and various issues. In this situation, there is a significant likelihood of a stock-out, and daily trips to each retailer are impractical owing to vehicle limits. Also, the likelihood of being out of stock will go up when the demand is random, and the vehicle capacity is small compared to how much is needed at the retailers.

With additional constraints and time windows, this study tries to understand the trade-off between stock-out costs and transportation costs that occurred because of the uncertain demand within its time windows and find the optimal number of vehicle configurations to minimize total cost. In real conditions, vendors also need to be able to serve the retailers according to the schedule prepared beforehand, thus indicating the need for time windows constraint.

1.1. Problem Description

IRPSDTW can be described as an undirected graph, $G = (V, A)$, where $V = \{0, 1, \dots, N\}$ is the group of locations that needs replenishment and holding given items and with a given capacity. Node 0 denotes the depot with $A = \{(i, j): i, j \in V, i \neq j\}$ is the set of a route consisting of k number of customer locations, and d_{ij} represents the distance between location i and location j. The mathematical model for the problem is modified from the models from Huang & Lin [6]. There is M number of vehicles, and each vehicle has Q capacity. Each location i holds C number of items, where maximum inventory for item c in location i is denoted by I_i^c.

In IRPSDTW, each location i is assumed to have ξ_i^c demand based on a normal distribution with $E[\cdot]$ is the expected value of the random variable, while the actual demand required is implied as ζ_i^c . Furthermore, this study also considers the failure cost if vehicle m failed to fulfil the demand in location i for item c with a probability of $\theta_{i,m}^c$. Additionally, the probability of stock out for item c in location i during the replenishment cycle is $P_i^c(\cdot)$ with S_c per-unit cost of item c being understocked. Further, we denoted T as the unit cost (per km), W as regular working day, $s_{i,j}^m$ notated the speed of vehicle, where each location i needed δ_i^m replenishment time, and u_i^m served as the flow variable for vehicle m once it served location i. Decision variable $x_{i,j}^m = \{0,1\}$, where $x_{i,j}^m = 1$ implored that there is path between location i and j using vehicle m, whereas decision variable y_i^m will be equal to 1 if vehicle m is assigned to replenish location i.

The objective function cost (z) consists of stock out cost SOC, transportation cost TR, failure cost F and penalty cost PEN due to time constraint violation and is presented in equation (1). The transportation cost is calculated by multiplying the planned travel distance $D^m = \sum_i \sum_j d_{i,j} \cdot x_{i,j}^m, \forall i, j \in \{0\} \cup V_1, i \neq j$, with unit cost T. The transportation cost per replenishment cycle is $\sum_{m=1}^{M} (D^m \times T)$. Additional transport cost might occur if the vehicle fails to fulfil demand in location I, which require the vehicle m to restock by returning to depot and have a recourse path equal with $2d_{i,0}$. The probability of failure can be calculated from expected demand and actual demand data equivalent with $\theta_{i,m}^c = \max_{c \in C}\{Pr(\sum_{i=1}^{f} \zeta_i^c > \sum_{i=1}^{k} E[\xi_i^c])\}$. The failure cost is $\sum_{m=1}^{M} \sum_{i=1}^{k} (2d_{i,0} \times \theta_{i,m}^c \times T)$.

$$\begin{aligned} z &= \sum_{m=1}^{M} (D^m \times T) \\ &+ \sum_{m=1}^{M} \sum_{i=1}^{k} (2d_{i,0} \times \theta_{i,m}^c \times T) \\ &+ \int_{x=1}^{x=\infty} \sum_{i \in V_2} \sum_{c=1}^{C} S_c \times P_i^c(x)\, dx \\ &+ a_j^m \\ &- \sum_{m \in M} \sum_{i \in V} x_{i,j}^m (a_i^m + t_{i,j} + \delta_i + w_i) \end{aligned} \tag{1}$$

A stockout happens when a consumer's demand for a certain product is not promptly met. Given that M trucks cannot serve location during a replenishment cycle, the nodes are divided into two sets, V1 and V2, with V1 representing locations designated for service and V2 representing those not. Hence the SOC is equal to $SOC = \int_{x=1}^{x=\infty} \sum_{i \in V_2} \sum_{c=1}^{C} S_c \times P_i^c(x)\, d$

The time window of a location i, is specified by an interval [e_i, l_i] representing the earliest and the latest arrival time. All vehicles must arrive at a site before the end of the time windows l_i.

http://dx.doi.org/10.30656/jsmi.v6i2.4813

Accordingly [e_0, l_0] represents the time windows for the depot. The travel time between path $(i, j) \in E$ is denoted by $t_{i,j}$ where $t_{i,j} = d_{i,j} \times T$. a_i^m is the arrival time of vehicle m at customer i and p_i^m is the departure time at location i.

The time windows are viewed as a hard time constraint that cannot be breached. However, it is permissible to break the limitations during the computing operation. Each time a solution exceeds or violates the time limit, it incurs a penalty. The penalty can be calculated following as $a_j^m - \sum_{m\in M}\sum_{i\in V} x_{i,j}^m(a_i^m + t_{i,j} + \delta_i + w_i)$.

Subject to

$$\sum_{i\in V_1}\sum_{c=1}^{C} E[\xi_i^c] \times y_i^m \le Q \tag{2}$$

$$\sum_{j\in V_1}\sum_{m=1}^{M} x_{i,j}^m = 1, \forall i \in V_1, i \neq j, j \neq 0, i \neq 0 \tag{3}$$

$$\sum_{i\in V_1}\sum_{m=1}^{M} x_{i,j}^m = 1, \forall i \in V_1, i \neq j, j \neq 0, i \neq 0 \tag{4}$$

$$\sum_{\substack{j\in V_2\\ j\neq i}}\sum_{m=1}^{M} x_{i,j}^m = 0, \forall i \neq j \tag{5}$$

$$\sum_{\substack{i\in V_2\\ i\neq j}}\sum_{m=1}^{M} x_{i,j}^m = 0, \forall j \neq i \tag{6}$$

$$\sum_{i\in V_1}\sum_{m=1}^{M} x_{i,0}^m = \sum_{j\in V_1}\sum_{m=1}^{M} x_{0,j}^m \tag{7}$$

$$x_{i,j}^m \le y_i^m, \forall i \in V_1, \forall m \in M \tag{8}$$

$$u_i^m - u_j^m + Q.x_{i,j}^m \le Q - \sum_{c=1}^{C} E[\xi_j^c], \forall m \in M \tag{9}$$

$$\sum_{c=1}^{C} E[\xi_i^c] \le u_i^m \le Q, \forall i \in V_1, \forall m \in M \tag{10}$$

$$\sum_{i=1}^{N} y_i^m w_i^m + D^m/s_{i,j}^m \le W, \forall m \in M \tag{11}$$

$$a_0^m = w_0^m = \delta_0^m \tag{12}$$

$$e_i \le a_i^m \le l_i, i \in V, m \in M \tag{13}$$

$$x_{i,j}^m \in \{0,1\}, \forall i \in V_1, m \in M \tag{14}$$

$$y_i^m \in \{0,1\}, \forall i \in V_1 \tag{15}$$

Constraint (2) states that the expected total demand for vehicle m allocated to restock location i should be less or equal with vehicle's capacity. Constraint (3) guarantees that any vehicle leaving location is counted only once in set V1 and (4) that any vehicle that visits a location is only counted once in set V_1. Constraints (5) and (6) guarantee that no vehicle will visit the vending machines belonging to set V_2. In addition, constraint (7) denotes vehicle flow in and out of the depot. Constraints (8) guarantee that route (i,j) is served by vehicle m if vehicle m is tasked with restocking location i. The sub-tour elimination limitations (9) and (10) impose the capacity and vehicle flow on the routes. For the time windows, constraint (11) indicates that the intended delivery time allocated to vehicle m should be within working hour W. Constraints (12-13) ensure that each vehicle must arrive in the retailers i within the time interval. Decision variables are presented in (14-15).

2. RESEARCH METHODS

This paper combines a tabu search algorithm and variable neighbourhood descent (VND) to solve this proposed inventory routing problem. Tabu search has been proven to be able to solve variants of vehicle routing problems, including inventory routing problems, with robust results [22], [33], [34]. Furthermore, the algorithm is also able to solve a stochastic problem [35], and additional time windows constraint [36], [37]. Hybridization of metaheuristics algorithm is not new, including using VND as part of Tabu Search for routing problems [38]. VND has also been proven to be a robust local search algorithm [39] [40]. This solving algorithm is developed as a two-phase method. The first phase tries to find the initial solution by considering the time windows that will affect the stock out happen. A time-heuristic-based method is applied to find the initial solution will low time violation. After constructing the initial solution, the intra-route swap method is conducted to improve the initial solution. Furthermore, two-movement, insert and swap are implemented in the second phase. In the end part of the algorithm, we perform VND to find the optimal solution.

2.1. Initial Solution

To construct the initial solution, we try not to violate the time windows of each customer. Hence, we first order the customers based on the centre of their time window ½ (e_i + l_i). Until all

http://dx.doi.org/10.30656/jsmi.v6i2.4813

customers are assigned to a route, we sequentially select a pre-ordered customer and insert it in the route *k*. The number of *k* is based on the number of vehicles available (*M*). The detailed procedure for Phase 1 is as follows:

Step1. Order ascending all customers based on the center of their time window ½ ($e_i + l_i$).
Step2. Set up $k \leftarrow 1$ as the first route. The number of vehicles *m* to be the maximum route constructed.
Step3. Select customer *i* from the list of ordered customers.
Step4. Insert customer *i* at the start of route *k*.
Step5. Set $k \leftarrow k + 1$.
Step6. If *k* exceeds the number of available vehicles *m*, set $k \leftarrow 1$.
Step7. Repeat until all customers are assigned to a route.

2.2. Tabu Search Algorithm

According to Glover [41], the Tabu Search algorithm explores the solution space by going from a solution *x* discovered at iteration *t* to the best solution x_{t+1} in a subset of the neighbourhoods *N(x)*.

After acquiring the initial solution in Phase 1, we implemented the insert move Tabu search strategy to enhance the solution. Next, a swap move is utilized to enhance the existing optimal solution. We execute 2-opt till the maximum iteration count Max_{Ite} after the tabu algorithm. The following is the phase 2 technique in detail (Fig. 1):

Step1. Set up *Ite* as the iteration number and *Max_Ite* as an iteration threshold in the improvement phase. X_0 is an initial solution, and the optimal solution $X^* = X_0$.
Step2. Perform insert to produce a new solution (from X_0 to X_1) by randomly selecting two routes, V_i and V_j, and choose one of the nodes in V_i and insert randomly in V_j.
Step3. Select the move with the lowest objective function (1) from the candidate list.
Step4. Perform Step5 if the move is included in the tabu list; otherwise, perform Step6.
Step5. If this candidate has $total\_cost\ (X_1) \leq total\_cost(X^*)$, go to Step 6. Otherwise, go to Step3, and select another move.
Step6. Perform the insert move. Update $X_0 = X_1$, $total\_cost(X_0) = total\_cost(X_1)$, and replace the Tabu list.
If $total\_cost\ (X_1) \leq total\_cost(X^*)$, $X^* = X_1$, $total\_cost\ (X^*) = total\_cost(X_1)$.
Step7. Repeat Step2-Step7 for the swap move.
Step8. Perform 2-opt move. If $total\_cost(X_1) \leq total\_cost(X^*)$, $X^* = X_1$, $total\_cost(X^*) = total\_cost(X_1)$.
Step9. If $total\_cost\ (X_1) \leq total\_cost(X^*)$, $X^* = X_1$, $total\_cost\ (X^*) = total\_cost(X_1)$.
Step10. Stopping criterion. If $Ite \geq Max\_Ite$, record the optimal solution X^*, calculate $total\_cost(X^*)$, and stop procedure. Otherwise, go back to Step 2.

```
Algorithm2. VND

Input: The set of neighborhood structures $N_k$, for
       $k = 1, 2, . . . , k_{max}$ (swap, insert, 2-opt)
Initialization: Find an initial solution $x$;
Repeat
   improve=false
   $k \leftarrow 1$
   while $k \leq k_{max}$ do
      $x' \leftarrow$ Local Search (x)
if $f(x') < f(x)$ then
   improve=true
   $x \leftarrow x'$
end if
$k \leftarrow k+1$
until improve=false
return $x$
```

Fig. 1. Variable neighborhood descent algorithm

2.3. Variable Neighborhood Descent (VND)

We proposed a tabu search algorithm incorporating two neighbourhood structures to explore different possibilities of a solution. Here, we also applied the VND in the local search phase to explore another possibility of the solution. In this VND, we applied three different neighborhoods to avoid local optimal.

3. RESULTS AND DISCUSSION

3.1. Test Instances

The test instances for the IRPSDTW were performed on the existing test instances for the VRP. They were generated from Solomon's data set [42], with 100-node instances containing data similar to Huang & Lin [6]. The penalty for violating the time windows is small at 0.1 per minute. Solomon's benchmark data for time windows is standardized from 0 to 480.

3.2. Algorithm Verification

For verification, Tabu VND (TVND) is first applied to solve datasets listed in Table 1. In this dataset, the time windows are ignored, and the time windows constraint is related to working hours, as proposed by Huang & Lin [6]. The results are being compared with those obtained

http://dx.doi.org/10.30656/jsmi.v6i2.4813

from Ant Colony Optimization (ACO), and Modified ACO (MACO) proposed previously. The conventional ACO minimises travel distance and has a higher stock-out cost than MACO. The gap is obtained from the difference between TVND with MACO.

As shown in Table 1, the proposed algorithm can solve the IRPSDMI instances and get the solution nearly the same as the result of MACO. Our algorithm performs well for clustered instances, uniformly distributed and mixed instances, with the average gap with ACO being -9.0% and MACO 0.1%. Although the proposed algorithm could not outperform MACO in some instances, the current results are considered efficient and robust for solving the IRPSDMI. However, the computational time could not be compared directly due to different specifications. The uncertainty might affect the overall result depending on the value of demand when running the instances. A t-test is conducted to compare the results between ACO and MACO with TSVND (Tabel 2). Based on the paired sample t-test, the result between ACO and TSVND is significantly different at level 0.01. although slightly worse than MACO, the t-test result shows that the difference is insignificant, indicating that the proposed algorithm still could perform well.

3.3. Result

We run our algorithm in Solomon bench-marks which are divided into three categories: C-type (clustered customers), R-type (uniformly distributed customers) and RC-type (a mix of R and C types). Parameter selection may influence the quality of the computational results. Thus, for this problem, we set the parameter with the details below:

a.	tabu list	=	7
b.	tabu tenure	=	3
c.	Max_Item	=	2500
d.	aspiration criterion	=	1%

Table 1. Comparison of ACO [6], MACO [6], and Tabu VND

Data Set	ACO (1)	MACO (2)	TS (3)	Gap ((1)-(3))/(1) (%)	Gap ((2)-(3))/(2) (%)
c101	22197.0	21240.0	21663.0	**-2.4%**	2.0%
c102	18590.8	13598.9	13583.9	**-26.9%**	**-0.1%**
c103	19861.8	12960.0	13665.8	**-31.2%**	5.4%
c104	21346.2	21140.3	19623.5	**-8.1%**	**-7.2%**
c105	19840.8	17886.7	16895.0	**-14.8%**	**-5.5%**
r101	20607.8	19976.7	19756.1	**-4.1%**	**-1.1%**
r102	15121.5	15083.4	15083.4	**-0.3%**	0.0%
r103	13854.4	13884.5	13854.4	**0.0%**	**-0.2%**
r104	14348.1	14131.5	14211.8	**-0.9%**	0.6%
r105	14893.5	13414.0	13414.0	**-9.9%**	**0.0%**
rc101	20294.3	19923.2	20167.9	**-0.6%**	1.2%
rc102	22486.0	20082.9	20797.0	**-7.5%**	3.6%
rc103	20881.8	19726.7	20167.4	**-3.4%**	2.2%
rc104	21356.9	18244.7	19342.3	**-9.4%**	6.0%
rc105	22179.8	19815.9	18701.4	**-15.7%**	**-5.6%**
Average				**-9.0%**	0.1%

Table 2. Paired Sample t-test for ACO-TSVND and MACO-TSVND

	ACO	TSVND	MACO	TSVND
Mean	19190.71	17395.14	17407.29	17395.14
Variance	9460012	9553435	9971260.40	9553435
Observations	15	15	15	15
Pearson Correlation	0.81		0.97	
Hypothesized Mean Difference	0		0	
df	14		14	
P(T<=t) one-tail	0.001**		0.474	

** significant at level 0.01

http://dx.doi.org/10.30656/jsmi.v6i2.4813

Table 3. IRPSDTW with tabu VND result

Instances	Best results					Standard deviation (%)	Time (second)
	Stock out cost	Travel cost	Failure cost	Total cost	Average		
c101	7453.7	15586.3	6833.8	29873.7	31158.3	4.30%	497
c102	5969.7	12264.1	5931.9	24165.8	24649.1	2.00%	503
c103	3503.4	15780.1	4340.0	23623.5	23624.2	0.00%	606
c104	6278.8	15048.7	5567.5	26895.0	26948.8	0.20%	576
c105	5757.1	15430.7	4547.9	25735.8	26541.3	3.13%	515
r101	3598.0	17161.1	8053.3	28812.4	29014.0	0.70%	564
r102	8456.6	10953.7	*3384.5*	22794.8	23038.7	1.07%	564
r103	6822.3	9469.7	*2678.2*	18970.2	18971.1	0.01%	561
r104	6504.2	13003.4	*2572.6*	22080.2	22228.1	0.67%	527
r105	6658.1	11778.8	*2355.6*	20792.5	21000.5	1.00%	502
rc101	4753.3	20756.7	5919.8	31429.8	31524.0	0.30%	563
rc102	10759.1	14986.3	4551.6	30297.0	30660.6	1.20%	514
rc103	3416.7	20888.3	5362.4	29667.4	30919.3	4.22%	569
rc104	3919.9	20706.2	5216.3	29842.3	30101.9	0.87%	584
rc105	6827.3	17820.1	5554.0	30201.4	30382.6	0.60%	604
Average						1.35%	550

This parameter is chosen after several trials and seems to give the best result. Each dataset is run 50 times to understand the algorithm's robustness. In this computational experiment, two types of sets, 1 and 2, are investigated due to their different time windows nature. Table 3 shows the best result, average result, and standard deviation based on the results of 50 times running.

Table 3 shows that by adding the time windows constraint, the total cost incurred is higher compared with the results of IRPSDMI. As additional constraints are applied, the solution space is narrowed down, and the probability of being stocked out increases. It might happen due to time limitations resulting in different routes between IRPSDMI and IRPSDTW, although with the exact location coordinate and parameter. The stock-out cost increases because the vehicle fails to fulfil the customers demand in the interval. In this study, the challenge is to improve the solution for the time windows constraint. The narrower the time window of a certain customer, the more difficult it is to insert this customer into a route and find a feasible solution.

3.4. Sensitivity Analysis and Managerial Implications

Furthermore, this research examined the impact of adding car cost per unit vehicle usage and vehicle count on the solutions. The depot and customer coordinates were retrieved from Solomon's 56 benchmark issue RC101. The cost of transportation will grow as the number of cars increases, whereas the cost of stocking out will drop. Fig. 2 demonstrates that using ten trucks will reduce the overall cost. Although increasing the number of vehicles reduces the likelihood of stock-outs, the reduction in stock-out costs may not be sufficient to offset the rise in transportation expenses, increasing the overall cost.

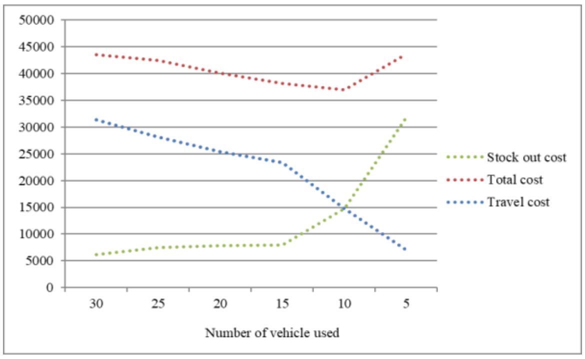

Fig. 2. The trade-off between transportation cost and stock-out cost

As the vendors are liable to fulfil the retailers' demand, providing a recourse strategy will be beneficial as it can reduce the total cost, including travel costs, stock-out costs, and failure costs. Based on the results of our study, setting the delivery schedule using a wide time window can

http://dx.doi.org/10.30656/jsmi.v6i2.4813

help create a more flexible route. Sometimes, a penalty cost should not be incurred although the vendor delivers after or before the time windows constraint. Furthermore, using the recourse strategy where vehicles return to the depot to fulfil retailers' orders can also minimize the total cost if the number of vehicles used is adequate. On the other hand, replenishment strategies may differ and should be assessed considering various retailer/customer demand conditions.

4. CONCLUSION

This study has formulated a model for the multi-item replenishment problem in uncertain demand with time windows. A hybrid tabu search and variable neighbourhood descent are being introduced to minimise total costs from transportation, vehicle failure, stock out cost and penalty cost. Suppliers are liable to meet the demands so that a recourse plan may cut overall costs, including travel, stock out, and failure costs. According to our analysis, adopting large time frames may assist develop a more flexible delivery path. In certain circumstances, a vendor shouldn't be penalized for delivering after or before the deadline. Using the recourse method, where vehicle return to the depot to fulfil store orders, may also save costs if the quantity of vehicles utilized is sufficient.

Several scenarios for modelled demand uncertainty can be developed for future research, and two-phase stochastic programming should be used to solve the problem. Furthermore, as the number of vehicles directly impacts the total cost, different vehicle sizes can also be considered to minimize stock-out costs and travel costs.

REFERENCES

[1] H. Stadtler and C. Kilger, *Supply Chain Management and Advanced Planning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-74512-9

[2] A. Rusdiansyah and D. B. Tsao, ‘An integrated model of the periodic delivery problems for vending-machine supply chains’, *J. Food Eng.*, vol. 70, no. 3, pp. 421–434, 2005, doi: 10.1016/j.jfoodeng.2004.05.073.

[3] A. M. Campbell, M. Gendreau, and B. W. Thomas, ‘The orienteering problem with stochastic travel and service times’, *Ann. Oper. Res.*, vol. 186, no. 1, pp. 61–81, Jun. 2011, doi: 10.1007/s10479-011-0895-2.

[4] F. Neves-Moreira, B. Almada-Lobo, J.-F. Cordeau, L. Guimarães, and R. Jans, ‘Solving a large multi-product production-routing problem with delivery time windows’, *Omega*, vol. 86, pp. 154–172, Jul. 2019, doi: 10.1016/j.omega.2018.07.006.

[5] Y. Qiu, J. Qiao, and P. M. Pardalos, ‘Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory’, *Omega (United Kingdom)*, vol. 82, pp. 193–204, 2019, doi: 10.1016/j.omega.2018.01.006.

[6] S.-H. Huang and P.-C. Lin, ‘A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty’, *Transp. Res. Part E Logist. Transp. Rev.*, vol. 46, no. 5, pp. 598–611, Sep. 2010, doi: 10.1016/j.tre.2010.01.006.

[7] A. Al Shamsi, A. Al Raisi, and M. Aftab, ‘Pollution-Inventory Routing Problem with Perishable Goods’, P. Golinska, Ed. Cham: Springer International Publishing, 2014, pp. 585–596, doi: 10.1007/978-3-319-07287-6_42.

[8] M. Soysal, J. M. Bloemhof-Ruwaard, R. Haijema, and J. G. A. J. van der Vorst, ‘Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty’, *Int. J. Prod. Econ.*, vol. 164, pp. 118–133, Jun. 2015, doi: 10.1016/j.ijpe.2015.03.008.

[9] J. E. Fokkema, M. J. Land, L. C. Coelho, H. Wortmann, and G. B. Huitema, ‘A continuous-time supply-driven inventory-constrained routing problem’, *Omega*, vol. 92, p. 102151, Apr. 2020, doi: 10.1016/j.omega.2019.102151.

[10] S. Treitl, P. C. Nolz, and W. Jammernegg, ‘Incorporating environmental aspects in an inventory routing problem. A case study from the petrochemical industry’, *Flex. Serv. Manuf. J.*, vol. 26, no. 1–2, pp. 143–169, Jun. 2014, doi: 10.1007/s10696-012-9158-z.

[11] F. E. Achamrah, F. Riane, and S.

Limbourg, ‘Spare parts inventory routing problem with transshipment and substitutions under stochastic demands’, *Appl. Math. Model.*, vol. 101, pp. 309–331, Jan. 2022, doi: 10.1016/j.apm.2021.08.029.

[12] P. L. Miranda, R. Morabito, and D. Ferreira, ‘Optimization model for a production, inventory, distribution and routing problem in small furniture companies’, *TOP*, vol. 26, no. 1, pp. 30–67, May 2017, doi: 10.1007/s11750-017-0448-1.

[13] F. A. Touzout, A.-L. Ladier, and K. Hadj-Hamou, ‘An assign-and-route matheuristic for the time-dependent inventory routing problem’, *Eur. J. Oper. Res.*, vol. 300, no. 3, pp. 1081–1097, Aug. 2022, doi: 10.1016/j.ejor.2021.09.025.

[14] M. Mahjoob, S. S. Fazeli, S. Milanlouei, L. S. Tavassoli, and M. Mirmozaffari, ‘A modified adaptive genetic algorithm for multi-product multi-period inventory routing problem’, *Sustain. Oper. Comput.*, vol. 3, pp. 1–9, 2022, doi: 10.1016/j.susoc.2021.08.002.

[15] J. Skålnes, H. Andersson, G. Desaulniers, and M. Stålhane, ‘An improved formulation for the inventory routing problem with time-varying demands’, *Eur. J. Oper. Res.*, vol. 302, no. 3, pp. 1189–1201, 2022, doi: 10.1016/j.ejor.2022.02.011.

[16] V. F. Yu, A. T. Widjaja, A. Gunawan, and P. Vansteenwegen, ‘The Multi-Vehicle Cyclic Inventory Routing Problem: Formulation and a Metaheuristic Approach’, *Comput. Ind. Eng.*, vol. 157, p. 107320, Jul. 2021, doi: 10.1016/j.cie.2021.107320.

[17] M. Mahjoob, S. S. Fazeli, L. S. Tavassoli, M. Mirmozaffari, and S. Milanlouei, ‘A green multi-period inventory routing problem with pickup and split delivery: A case study in flour industry’, *Sustain. Oper. Comput.*, vol. 2, pp. 64–70, 2021, doi: 10.1016/j.susoc.2021.04.002.

[18] M. W. Friske, L. S. Buriol, and E. Camponogara, ‘A relax-and-fix and fix-and-optimize algorithm for a Maritime Inventory Routing Problem’, *Comput. Oper. Res.*, vol. 137, p. 105520, Jan. 2022, doi: 10.1016/j.cor.2021.105520.

[19] L. C. Coelho, A. De Maio, and D. Laganà, ‘A variable MIP neighborhood descent for the multi-attribute inventory routing problem’, *Transp. Res. Part E Logist. Transp. Rev.*, vol. 144, p. 102137, Dec. 2020, doi: 10.1016/j.tre.2020.102137.

[20] S. T. Vadseth, H. Andersson, and M. Stålhane, ‘An iterative matheuristic for the inventory routing problem’, *Comput. Oper. Res.*, vol. 131, p. 105262, Jul. 2021, doi: 10.1016/j.cor.2021.105262.

[21] A. Karoonsoontawong, O. Kobkiattawin, and C. Xie, ‘Efficient Insertion Heuristic Algorithms for Multi-Trip Inventory Routing Problem with Time Windows, Shift Time Limits and Variable Delivery Time’, *Networks Spat. Econ.*, vol. 19, no. 2, pp. 331–379, Jun. 2019, doi: 10.1007/s11067-017-9369-7.

[22] M. Seifbarghy and Z. Samadi, ‘A tabu search-based heuristic for a new capacitated cyclic inventory routing problem’, *Int. J. Math. Oper. Res.*, vol. 6, no. 4, pp. 491–504, Jan. 2014, doi: 10.1504/IJMOR.2014.063159.

[23] Y. Yu, C. Chu, H. Chen, and F. Chu, ‘Large scale stochastic inventory routing problems with split delivery and service level constraints’, *Ann. Oper. Res.*, vol. 197, no. 1, pp. 135–158, Aug. 2012, doi: 10.1007/s10479-010-0772-4.

[24] L. Bertazzi, A. Bosco, F. Guerriero, and D. Laganà, ‘A stochastic inventory routing problem with stock-out’, *Transp. Res. Part C Emerg. Technol.*, vol. 27, pp. 89–107, Feb. 2013, doi: 10.1016/j.trc.2011.06.003.

[25] L. C. Coelho, J.-F. Cordeau, and G. Laporte, ‘Heuristics for dynamic and stochastic inventory-routing’, *Comput. Oper. Res.*, vol. 52, pp. 55–67, Dec. 2014, doi: 10.1016/j.cor.2014.07.001.

[26] A. Gruler, J. Panadero, J. de Armas, J. A. Moreno Pérez, and A. A. Juan, ‘Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs’, *Comput. Ind. Eng.*, vol. 123, pp. 278–288, Sep. 2018, doi: 10.1016/j.cie.2018.06.036.

[27] E. Nikzad, M. Bashiri, and F. Oliveira,

'Two-stage stochastic programming approach for the medical drug inventory routing problem under uncertainty', *Comput. Ind. Eng.*, vol. 128, pp. 358–370, Feb. 2019, doi: 10.1016/j.cie.2018.12.055.

[28] W. W. Qu, J. H. Bookbinder, and P. Iyogun, 'Integrated inventory-transportation system with modified periodic policy for multiple products', *Eur. J. Oper. Res.*, vol. 115, no. 2, pp. 254–269, 1999, doi: 10.1016/S0377-2217(98)00301-4.

[29] W.-H. Yang, K. Mathur, and R. H. Ballou, 'Stochastic Vehicle Routing Problem with Restocking', *Transp. Sci.*, vol. 34, no. 1, pp. 99–112, Feb. 2000, doi: 10.1287/trsc.34.1.99.12278.

[30] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza, 'A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem', *Transp. Sci.*, vol. 41, no. 3, pp. 382–391, Aug. 2007, doi: 10.1287/trsc.1060.0188.

[31] C. H. Christiansen and J. Lysgaard, 'A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands', *Oper. Res. Lett.*, vol. 35, no. 6, pp. 773–781, Nov. 2007, doi: 10.1016/j.orl.2006.12.009.

[32] R. Roldán, R. Basagoiti, and E. Onieva, 'Inventory routing problem with stochastic demand and lead time: State of the art', in *Advances in Intelligent Systems and Computing*, 2014, vol. 299, pp. 73–82, doi: 10.1007/978-3-319-07995-0_8.

[33] M. Alinaghian, E. B. Tirkolaee, Z. K. Dezaki, S. R. Hejazi, and W. Ding, 'An augmented Tabu search algorithm for the green inventory-routing problem with time windows', *Swarm Evol. Comput.*, vol. 60, p. 100802, Feb. 2021, doi: 10.1016/j.swevo.2020.100802.

[34] H. Jia, Y. Li, B. Dong, and H. Ya, 'An Improved Tabu Search Approach to Vehicle Routing Problem', *Procedia - Soc. Behav. Sci.*, vol. 96, pp. 1208–1217, Nov. 2013, doi: 10.1016/j.sbspro.2013.08.138.

[35] G. Li and J. Li, 'An Improved Tabu Search Algorithm for the Stochastic Vehicle Routing Problem With Soft Time Windows', *IEEE Access*, vol. 8, pp. 158115–158124, 2020, doi: 10.1109/ACCESS.2020.3020093.

[36] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin, 'Tabu search for the time-dependent vehicle routing problem with time windows on a road network', *Eur. J. Oper. Res.*, vol. 288, no. 1, pp. 129–140, Jan. 2021, doi: 10.1016/j.ejor.2020.05.041.

[37] J. Ochelska-Mierzejewska, 'Tabu search algorithm for vehicle routing problem with time windows', in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 40, A. Poniszewska-Marańda, N. Kryvinska, S. Jarząbek, and L. Madeyski, Eds. Cham: Springer International Publishing, 2020, pp. 117–136, doi: 10.1007/978-3-030-34706-2_7.

[38] Z. Zeng, X. Yu, K. He, and Z. Fu, 'Adaptive Tabu search and variable neighborhood descent for packing unequal circles into a square', *Appl. Soft Comput.*, vol. 65, pp. 196–213, Apr. 2018, doi: 10.1016/j.asoc.2017.11.051.

[39] L. Barrero, F. Robledo, P. Romero, and R. Viera, 'A GRASP/VND Heuristic for the Heterogeneous Fleet Vehicle Routing Problem with Time Windows', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, vol. 12559 LNCS, pp. 152–165, doi: 10.1007/978-3-030-69625-2_12.

[40] A. Khmelev and Y. Kochetov, 'A hybrid VND method for the split delivery vehicle routing problem', *Electron. Notes Discret. Math.*, vol. 47, pp. 5–12, Feb. 2015, doi: 10.1016/j.endm.2014.11.002.

[41] F. Glover, 'Future paths for integer programming and links to artificial intelligence', *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986, doi: 10.1016/0305-0548(86)90048-1.

[42] M. M. Solomon, 'Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints', *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987, doi: 10.1287/opre.35.2.254.