Metaheuristics for time-dependent vehicle routing problem with time windows

Yun-C LIANG

Vanny MINANDA

Aldy GUNAWAN
Singapore Management University, aldygunawan@smu.edu.sg

Hsiang-L. CHEN

Metaheuristics for time-dependent vehicle routing problem with time windows

Yun-Chia Liang and Vanny Minanda

Department of Industrial Engineering and Management,
Yuan Ze University, Taiwan
Email: ycliang@saturn.yzu.edu.tw
Email: s1088906@mail.yzu.edu.tw

Aldy Gunawan

School of Computing and Information Systems,
Singapore Management University, Singapore
Email: aldygunawan@smu.edu.sg

Hsiang-Ling Chen*

Department of Industrial and Systems Engineering,
Chung Yuan Christian University, Taiwan
Email: achen@cycu.edu.tw
*Corresponding author

Abstract: Vehicle routing problem (VRP), a combinatorial problem, deals with the vehicle's capacity visiting a particular set of nodes while its variants attempt to fit real-world scenarios. Our study aims to minimise total travelling time, total distance, and the number of vehicles under time-dependent and time windows constraints (TDVRPTW). The harmony search algorithm (HSA) focuses on the harmony memory and pitch adjustment mechanism for new solution construction. Several local search operators and a roulette wheel for the performance improvement were verified via 56 Solomon's VRP instances by adding a speed matrix. The performance comparison with a genetic algorithm (GA) was completed with the same number of parameters and ran in the same computer specification to justify its performance. The results show that HSA can outperform the GA in some instances. The research outcomes suggest that HSA can solve TDVRPTW with comparable results to other commonly used metaheuristic approaches.

Keywords: vehicle routing problem; VRP; time window; harmony search algorithm; HSA; genetic algorithm; metaheuristic.

Reference to this paper should be made as follows: Liang, Y-C., Minanda, V., Gunawan, A. and Chen, H-L. (xxxx) 'Metaheuristics for time-dependent vehicle routing problem with time windows', *Int. J. Operational Research*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Yun-Chia Liang is a Professor in the Industrial Engineering and Management Department at Yuan Ze University, Taiwan. His researches and publications mainly cover the topics of logistics and production scheduling, intelligent computing and artificial neural network applications. In addition, he is serving as an editorial member/guest editor of several reputed journals, and also a member of several national and international committees and academic societies, e.g., CIIE, IEEE and ORSTW.

Vanny Minanda graduated with her Master's (MSc) in Industrial Engineering and Management (IEM) at the Yuan Ze University, Taiwan, in 2019. Currently, she is pursuing her PhD in the same major and university. Her research interest includes the applications of metaheuristics algorithms such as harmony search algorithm (HSA), virus optimisation algorithm (VOA), and genetic algorithm (GA) to solve vehicle routing problem (VRP) and its extensions, as well as the applications of internet of things (IoT) devices for transportation and optimisation problems.

Aldy Gunawan is an Assistant Professor of Computer Science (Practice) at the Singapore Management University. He received his PhD in Industrial and Systems Engineering. His main research interests include operations research, algorithm design, and data analytics related to metaheuristics, algorithm configuration, design of experiments, combinatorial optimisation and automated planning/scheduling.

Hsiang-Ling Chen is an Associate Professor of Operations and Logistics Management at the Department of Industrial and Systems Engineering, Chuang Yuan Christian University, Taiwan. Her research interests are project operations, omnichannel and logistics, and data analytics related to decision-making. She is also a member of several academic societies, e.g., CIIE, IIE and ORSTW.

This paper is a revised and expanded version of a paper entitled 'Harmony search algorithm for time-dependent vehicle routing problem with time windows' presented at APIEMS 2019: The 20th Asia Pacific Industrial Engineering and Management Systems, Kanazawa, Japan, 2–5 December 2019.

1 Introduction

Vehicle routing problem (VRP) extending the travelling salesman problem (TSP) is a combinatorial optimisation problem, also known as vehicle scheduling, truck dispatching, or delivery problem, where a route is defined from a subset of nodes (customers), and each node has a demand. Most research assumes that the travel cost and travel time between locations are time-independent. However, this assumption is often not practical in the real world since traffic conditions are dynamic, resulting in significant variations in speed and travel times for particular routes. Furthermore, the demand in a real-world case scenario represents each customer's requested amount of the delivery product, which needs to be fulfilled. Every node or customer can only be visited at most once. The most challenging part of VRP is that the number of vehicles to use must be calculated, while vehicle capacity and total travelling time (TT) cannot exceed the limit.

Time-dependent vehicle routing problem (TDVRP) is a variant of VRP which tries to define a sequence of nodes while considering time dependencies. Time dependency means that the TT from node i to node j depends on the starting time on node i. Like in a real-world scenario, a certain amount of time is required to travel from one node (place) to the next, and traffic congestion differs from one period to another.

For example, peak hours require longer TTs because the vehicle speed is slower due to heavy traffic.

Time-dependent vehicle routing problem with time windows (TDVRPTW) is an extension of TDVRP in which time windows are considered. The time windows can be defined as the time duration between each node's opening and closing times. It is also necessary to consider the service time of each node in the time window constraint. Since each customer (node) has different opening and closing times, assigning the route for each vehicle becomes a more challenging task. If a vehicle arrives at a node before the opening time, such vehicle must wait (this is defined as the waiting time) until the node opens, which translates into a lower chance to visit another node due to the time limit budget (the latest time the vehicle can return to the depot). Since TDVRPTW tries to simulate a real-world problem as closely as possible, this has been applied when solving routing and transportation problems such as e-commerce (Kumar and Pannerselvam, 2015), pickup and delivery problems, and logistics and transportation problems.

The remainder of the paper is organised as follows. Section 2 briefly reviews VRP, vehicle routing problem with time windows (VRPTW), TDVRP and TDVRPTW. In Section 3, the mathematical model of TDVRPTW is defined, and then the proposed harmony search algorithm (HSA) and genetic algorithm (GA) is introduced in detail. The test instances and the computational results are discussed in Section 4, and finally, the concluding remarks are summarised in Section 5.

2 Literature review

VRP was introduced by (Dantzig and Ramser, 1959) with the name truck dispatching problem (TDP). The idea from TDP is to determine the shortest route that visits each node once for delivering gasoline to a service station. VRP is an NP-hard problem which means that when one or more of the limitations increases (for example, the number of customers to be served increases), the problem cannot be solved in polynomial time (Lenstra and Kan, 1981).

The classical VRP aims to minimise the total route; however, different objectives are on maximising customer satisfaction (Yang et al., 2015), the vehicle utilisation rate (Zhang and Pavone, 2016) and delivery rate (Szczepański et al., 2017). More recent surveys include Koç et al. (2016), Masutti and de Castro (2017), Adewumi and Adeleke (2018), Gansterer and Hartl (2018, 2020), Gunawan et al. (2021) and Wang and Wasil (2021). Some surveyed from the heuristic and metaheuristic methodologies, from the benchmark datasets, while others reviewed various constraints and objectives of the VRP.

2.1 Vehicle routing problem with time windows

When a time window is imposed on each customer, a variant called VRPTW can be obtained and often encountered when working with transportation problems. The main difference emphasised on the time windows constraint means that each node (each node represents a customer to be served) has a specific time interval (defined as the time difference between the opening and closing time) in which it has to be served. The time interval includes an opening, closing, and service time. Alvarenga et al. (2007) proposed a GA and a set of partitioning formulations to solve VRPTW with minimising total travel distance as the primary objective, and the result showed that the algorithm outperforms all previously known methods. Qi et al. (2008) applied tabu search (TS) with three neighbourhood operators: all-exchange, all-two-opt, and all-cross exchange to minimise the number of routes and the total TT. The algorithm was tested on the 56 Solomon benchmark instances and performed well with an efficient computational time.

More recent research was developed with two types of time windows. With a soft time-window (VRPSTW), Iqbal et al. (2015) presented a swarm-based artificial bee colony (ABC) algorithm to solve for the multi-objective with penalty cost. On the other hand, with a hard time window (VRPHTW), Zhang et al. (2017) dealt with delivering consumer goods and agricultural products in the logistics industry and applied a hybrid algorithm, using TS and the ABC algorithm.

2.2 Time-dependent vehicle routing problem

The generated route can be applied to a real-world problem by adding time dependencies. Time dependencies mean that the travel time from i to j depends on the departure time from i. Due to all these external reasons, time dependencies are applied in the real world, and the TT from node i to node j becomes stochastic. Alinaghian and Naderipour (2016) have comprehensively studied factors influential in fuel consumption in time-dependent vehicle routing.

Malandraki and Daskin (1992) first formulated TDVRP to handle small problems (10–25 instances) in 2–3 different time zones with a GRASP metaheuristic where the TTs inside the network are given. Another important heuristic algorithm is ant colony optimisation (ACO), which Donati (2008) applied to solve a TDVRP with 100 customers and three time zones. In addition, local improvement operators such as two-opt and cross-exchange was applied to yield efficient computational results. Further research about TDVRP was done by Figliozzi (2008), who introduced the constructive method where a route is built by adding the new node that adds the least cost in each iteration. The result shows that the method built by this constructive method can handle both soft and hard time windows.

Finally, Norouzi et al. (2015) proposed a simulated annealing (SA) algorithm for solving TDVRP. The result shows that this approach successfully improved the results by 11% when tested using Solomon's benchmark instances. Franceschetti et al. (2017) dealt with the vehicles facing traffic congestion which, at peak periods, significantly restricts vehicle speeds and leads to increased emissions, and proposed an adaptive large neighbourhood search for the time-dependent pollution-routing problem (TDPRP). Huang et al. (2017) explicitly considered path selection in the road network and formulated the TDVRP-PF models under deterministic and stochastic traffic conditions. Lera-Romero and Miranda-Bront (2018) studied the time-dependent elementary shortest path problem with resource constraints (TDESPPRC), solved by two exact methods.

2.3 Time-dependent vehicle routing problem with time windows

Ichoua et al. (2003) assumed that the violation of the time windows constraints is acceptable and applied a TS algorithm to minimise TT and penalties. The authors constructed three different speed scenarios that guaranteed FIFO property under each scenario, the travel speeds in the morning and evening rush hours were obtained by dividing the travel speeds in the middle of the day by a factor α. Yildirim and Çatay

(2009) applied ACO and compared the performance of the methods for both the dependent and time-independent VRPTW. The objective of the time-independent case is to minimise the total distance, while the objective of the time-dependent case is to minimise the total travel time.

Figliozzi (2008) proposed an iterated route construction and improvement (IRCI), consisting of two different stages, route construction, and route improvement, to solve both soft and hard time windows associated with an auxiliary route building iterated during the construction heuristic. This proposed method obtained a comparable solution and required fast CPU times in problems with soft and hard time windows. Figliozzi (2012) continued his previous work in TDVRPTW. In this research, four different speed scenarios with five-time periods were constructed to deal with 56 Solomon's benchmark instances, in which each scenario had three different travelling speeds, which were 25%, 50% and 75% faster, respectively, compared to Solomon's original speeds.

A recent study by Ichoua et al. (2003) considered both time-dependent and time windows in their research. A GA was proposed to solve the benchmark instances. The proposed method was compared with the previous research (Kumar and Pannerselvam, 2015) that used the same speed matrix. The result shows better cumulative total distance travelled from 56 Solomon benchmark instances; however, the TT and NV show that the previous studies perform better. Liu et al. (2020) minimised the sum of the fixed costs of the vehicle used and the costs of drivers, fuel consumption, and carbon emission with an improved ant colony algorithm.

3 Research methodology

This section will first introduce the mathematical model of the TDVRPTW and then explain how HSA and GA solved the TDVRPTW in detail.

3.1 TDVRPTW mathematical model

The mathematical TDVRPTW model is adapted from previous research (Donati, 2008), and the notations used in the formulation are summarised as below:

$x_{i,j}^k$	equal to 1 if travel occurs from node *i* to node *j* in vehicle *k*, otherwise $x_{i,j}^k = 0$
K	a set of available vehicles
V	a set of vertexes
d_{ij}^k	the travel distance from node *i* to *j* by vehicle *k*
$q_{\max}$	the vehicle capacity
q_i	demand in node *i*
C	a set of vertices that serves *n* customers
C_d	cost per unit distance travelled
C_t	cost per unit route duration
$[e_i, l_i]$	service time window of node *i*

g_i service time in node i

$t_{i,j}$ TT from node i to node j

y_{ik} service start time node i by vehicle k.

TDVRPTW's primary objective (1) is to minimise the number of vehicles, while the secondary objective is to minimise the total TT or distance, as shown in equation (2). From equation (3), vehicle capacity cannot be exceeded, while equation (4) indicates that all nodes must be assigned to one of the vehicles. When a vehicle arrives at a node, the same vehicle has to leave from such node as formulated in equation (5). The starting and ending points for all routes must be the depot, as shown in equation (6), and each vehicle must leave from the depot and return to the depot exactly once, as shown in equations (7) and (8). The vehicle that visits a node must satisfy the time window starting time as shown in equation (9) and closing time as shown in equation (10) by considering the service time to allow travel time between customers indicated in equation (11), where the decision variables are indicated by constraint (12) and constraint (13).

- primary objective:

$$\text{Minimise} \sum_{k\in K}\sum_{i\in C} x_{0j}^k \tag{1}$$

- secondary objective:

$$\text{Min } C_d \sum_{k\in K}\sum_{(i,j)\in A} d_{ij}^k x_{ij}^k + C_t \sum_{k\in K}\sum_{j\in C}\left(y_{n+1}^k - y_0^k\right) x_{0j}^k \tag{2}$$

subject to:

$$\sum_{i\in C} q_i \sum_{j\in V} x_{ij}^k \le q_{\max}, \forall k \in K \tag{3}$$

$$\sum_{k\in K}\sum_{j\in V} x_{ij}^k = 1, \forall i \in C \tag{4}$$

$$\sum_{i\in V} x_{il}^k - \sum_{i\in V} x_{ij}^k = 0, \forall l \in C, \forall k \in K \tag{5}$$

$$x_{i0}^k = 0, x_{n+1,i}^k = 0, \forall i \in V, \forall k \in K \tag{6}$$

$$\sum_{j\in V} x_{0j}^k = 1, \forall k \in K \tag{7}$$

$$\sum_{j\in V} x_{j,n+1}^k = 1, \forall k \in K \tag{8}$$

$$e_i \sum_{j\in V} x_{ij}^k \le y_i^k \ \forall i \in V, \forall k \in K \tag{9}$$

$$l_i \sum\nolimits_{j\in V} x_{ij}^k \ge y_j^k, \forall (i, j) \in A, \forall \in K \tag{10}$$

$$x_{i,j}^k \left(y_i^k + g_i + t_{i,j}\left(y_i^k + g_i\right)\right) \le y_j^k, \forall (i, j) \in A, \forall k \in K \tag{11}$$

$$x_{ij}^{k} \in \{0, 1\}, \forall (i, j) \in A, \forall k \in K \tag{12}$$

$$y_{i}^{k} \in \Re, \forall (i, j) \in A, \forall k \in K. \tag{13}$$

3.2 Harmony search algorithm

HSA is one of the common algorithms used to solve optimisation problems such as VRP, TSP and orienteering problem (OP). The idea behind HSA was inspired by natural phenomena, just like physical annealing in SA and human memory in TS. HSA was developed based on music observation where finding a perfect combination of harmonies adjusts the pitch of an instrument which in turn yields a better harmony. HSA belongs to the population-based algorithms family, where each individual is generated using the elite population archive or generated randomly. This is because the population of elite individuals has a higher tendency to yield a better solution by competing among themselves in each iteration. The comparison between harmony improvisation and optimisation can be observed in Table 1.

Table 1 Comparison harmony improvement and optimisation

Factors	*Harmony improvisation*	*Optimisation*
Targets	Aesthetic standard	Objective function
Best states	Fantastic harmony	Global optimum
Components	Pitches of instruments	Values of variables
Process units	Each practice	Each iteration

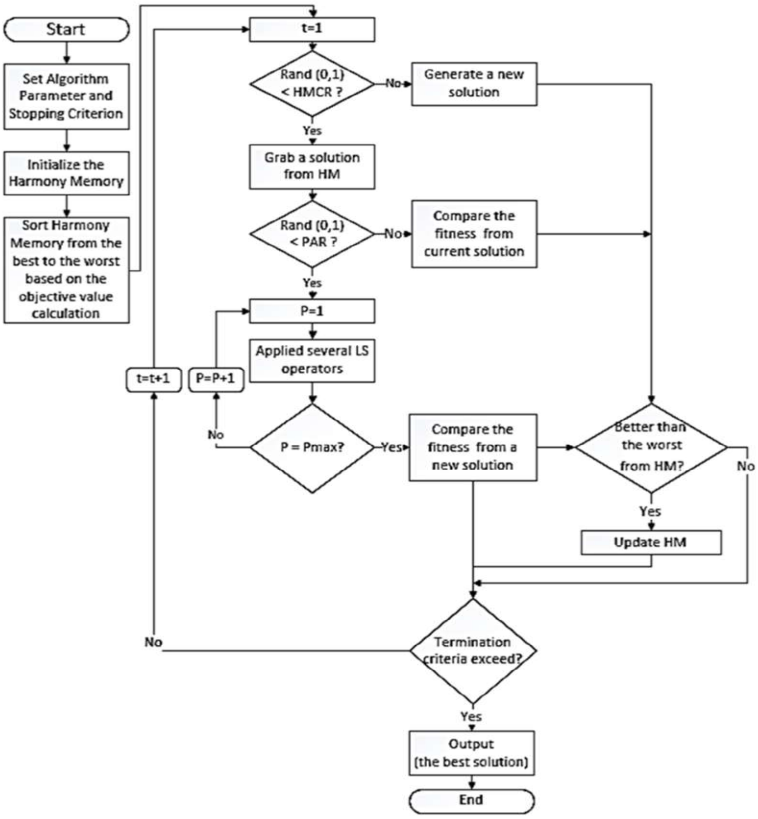
HSA implementation consists of three steps: initialisation, solution improvisation and update mechanism. In the initialisation step, the main parameters of HSA are initialised, and random initial solutions are generated and stored as the initial harmony memory (HM). In the solution improvisation step, the new solution is constructed by considering HM or randomly generating using the harmony memory considering ratio (HMCR). If the solution is generated using the HM, another parameter, pitch adjustment ratio (PAR), is employed to determine whether local search operators (LSOs) can improve the newly generated solution. Finally, if the improved solution does not violate any constraints in the update process, it will replace the worst solution in HM. A flowchart of this process in more detail is illustrated in Figure 1, and more details of the HSA will be described in the following sections.

3.2.1 Initial solution generation

Harmony memory size (HMS) was set to 100; thus, 100 initial solutions were generated at this phase. These initial solutions are generated without violating the constraints such as time windows, TT and vehicle capacity.

The initial solution generation procedure is as follows: the first node to be added in a route will be randomly selected, and the following nodes are chosen from the closest node from the previously added node. The node chosen has the least waiting time and is close to the distance to the previous node. If a node causes a route to violate a particular constraint, such node will not be added to that route (and it is returned to the unvisited nodes list), and such route is closed. Then, a new route is generated, and this process is repeated until all nodes are assigned to vehicles.

Figure 1 HSA flowchart



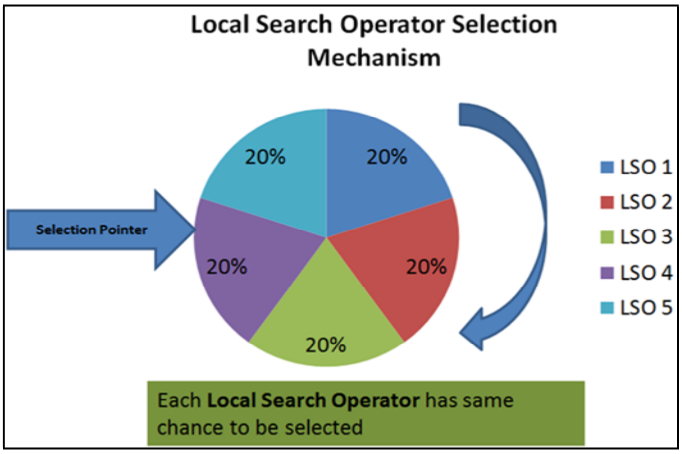
3.2.2 Solution improvisation

The solution improvisation is determined by two parameters: HM considering rate and pitch adjusting rate. HMCR (a random decimal between 0 and 1) is used as a decision criterion. A value of 0.95 for HMCR means that 5% of the time, the algorithm will generate a new random solution (that will substitute the worst possible solution only if it has a better fitness value in the update mechanism section). For the remaining 95% of the time, a route is randomly selected from HM as the input to pitch adjusting part. On the other hand, PAR determines the probability of modifying the selected solution through different LSOs. For example, if the value of PAR is equal to 0.3, i.e., 30% of the time, the algorithm will improve the selected HM solution by applying LSOs. For the remaining 70% of the time, the solution will be compared against the worst solution in HM and updated (in the update mechanism) only if it has a better fitness value.

There are five LSOs: swap, move, a group move, insert, and two-opt will be applied during the improvement stage to find the optimal solution. An unbiased selection of the

LSO operators is employed to choose which operator will be performed at each iteration. The selection probability of LSO operators is illustrated in Figure 2.

Figure 2 Roulette wheel for LSOs selection (see online version for colours)



3.2.3 Update mechanism

As the primary objective in TDVRPTW is to minimise the number of vehicles, and the secondary objective is to minimise the travelling distance or total TT, three different mechanisms for updating the solutions are considered and introduced as follows.

- Update mechanism 1:

  In the first update mechanism, TT as an update mechanism, a route is updated only if the number of vehicles (primary objective), as shown in equation (1), and the TT (secondary objective) shown in equation (14) is smaller than before (better fitness value).

  Secondary objective:

  $$\text{Min } C_t \sum_{k \in K} \sum_{j \in C} \left(y_{n+1}^k - y_0^k \right) x_{0j}^k \tag{14}$$

- Update mechanism 2:

  While in the TD + TT as an update mechanism scenario, a route is updated only if the total number of vehicles (primary objective) and the secondary objective fitness value (TD + TT), as shown in equations (1) and (2), respectively, are smaller than before.

- Update mechanism 3:

  Update mechanism 3 is an extension of update mechanism 2, where the second objective function is normalised as defined in equation (15). The normalisation function is applied due to the difference in range for the values of these two components.

Secondary objective:

$$\text{Min } \boldsymbol{normalise}\left(C_d \sum_{k\in K} \sum_{(i,j)\in A} d_{ij}^k x_{ij}^k + C_t \sum_{k\in K} \sum_{j\in C} \left(y_{n+1}^k - y_0^k \right) x_{0j}^k \right) \tag{15}$$

Normalisation was achieved through the standardisation formula shown below:

$$F_{new} = \frac{F_{current} - F_{\min}}{F_{\max} - F_{\min}} \tag{16}$$

From equation (16), $F_{current}$ represents the current objective value, $F_{\max}$ denotes the maximum objective value in each iteration in the current HM, and $F_{\min}$ expresses the minimum value in each iteration in the current HM. It is important to remember that the $F_{\max}$ and $F_{\min}$ values are adjusted dynamically for each iteration, and the maximum and minimum values for both TT and TD will be used to calculate the normalisation formula for this update mechanism scenario. This process will be repeated until the stopping criterion of a defined parameter is reached.

3.2.4 Travelling distance and time calculation

Euclidean distance formula was used to calculate the travelling distance between nodes. The latitude (x-coordinate) and longitude (y-coordinate) values of nodes calculate travelling distance.

$$d_{ij} = \sqrt{\left(x_i + x_j\right)^2 + \left(y_i + y_j\right)^2} \tag{17}$$

TT in a time-dependent problem is affected by the time horizon and departure time from the previous node. Time horizons represent traffic congestion during different periods of the day.

TT calculation between two vertices (i and j) can be seen from the pseudo-code below where the departure time (t_0) from node i and the distance between vertices i and j (d_{ij}) are needed. v_{cTk} represents the velocity when period k starts, while t_k and $(\overline{t_k})$ represent the moment when period k starts and ends respectively, t denotes the current time, and t' represents the arrival time at node j.

Set *t to t_0*

Set *d to d_{ij}*

Set t' to $t + (d_{ij} / V_k)$

While $(t' > \overline{t_k})$:

$d \leftarrow d - V_k(\overline{t_k} - t)$

$t \leftarrow \overline{t_k}$

$t' \leftarrow t + (d_{ij} / V_k)$

$k \leftarrow k + 1$

end while

return travel_time $(t' - t_0)$

3.3 Genetic algorithm

GA proposed to solve TDVRPTW with parameters such as the number of iterations, crossover probability (was set same as HMCR), and mutation rate (was set the same as PAR) defined the same HSA to make a fair comparison when compared its solution to HSA best-known solution. However, the proposed GA in this study and GA proposed by Kumar and Pannerselvam (2015) differ in one place. The updating mechanism in GA proposed by Kumar and Pannerselvam (2015) considered weight for an unconnected path when evaluating the solution. This implied that whenever an adjacent road is not connected, additional weight will be added to the total fitness; otherwise, such weight will be deducted from total fitness since the objective in this problem is minimising, which implied the smaller the fitness value, the better the solution is obtained. On the other hand, the GA proposed in this study follows one of three update mechanisms applied in HSA introduced in Section 3.2.3.

4 Computational experiment and analysis

The results of the computational experiment of the proposed algorithms to solve TDVRPTW are discussed in this section. Both algorithms were coded in Python 3.6 and were implemented on a computer with 16 GB RAM and an i7-4790 CPU @ 3.60 GHz.

4.1 Benchmark instances

Solomon's benchmark problems were used to test the performance of HSA and GA when solving the TDVRPTW. The 56 Solomon's problems compose six different problem types with 100 customers each. The six problem classes are R1, R2, C1, C2, RC1 and RC2. In problems R1 and R2, customers are randomly generated, while C1 and C2 are clustered problems, and RC1 and RC2 are a combination of randomly generated and clustered customers. In 56 Solomon's benchmark instances, each instance provides the customer's coordinates, opening and closing times, demand, vehicles capacities and service times. As a result, R1, C1, and RC1 have shorter scheduling horizons, tighter time windows, and also smaller vehicle capacities when compared to R2, C2 and RC2.

The speed matrix used for calculating TDVRPTW was adapted from previous research Figliozzi (2012). Such a speed matrix consists of three different speeds with a ratio of 2.5:1. In such a scenario, there are five different periods with equal time length, and all vehicles must depart in the early morning (period 1) where the traffic congestion is just starting. The speed matrix used to test HSA and GA performance when solving TDVRPTW in this research can be seen in Table 2.

Table 2 Speed matrix

Type of speed	*Time horizon*				
	I	*II*	*III*	*IV*	*V*
TD1	1.00	1.60	1.05	1.60	1.00
TD2	1.00	2.00	1.50	2.00	1.00
TD3	1.00	2.50	1.75	2.50	1.00

4.2 Computational results

The results of HSA are compared with the best-known solutions from the previous work (Figliozzi, 2012) that proposed the IRCI and Kumar and Pannerselvam (2015) that used GA to solve the same problem with the same benchmark instance. Table 3 shows the comparison between HSA performance in 56 benchmark instances and the two previous works of literature in speed matrix 1 (TD1) with update mechanism 1, while update mechanisms 2 and 3 are shown in Tables 4 and 5, respectively.

Table 3 Result comparison in TD1 and update mechanism 1

Instance		*Reference*		*Harmony search algorithm*		
		Figliozzi (2012)	*Kumar and Pannerselvam (2015)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*Gap against Kumar and Pannerselvam (%)*
R1	NV	11.67	13.58	12.00	2.82	–11.63
	TT	2,080.00	2,691.00	2,136.00	2.69	–20.62
	TD	1,295.00	1,624.00	1,388.00	7.35	9.81
R2	NV	2.882	3.64	2.81	–0.06	–22.57
	TT	1,990.00	3,077.00	1,824.00	–8.34	–40.72
	TD	1,216.00	939.00	1,114.00	–8.38	18.63
RC1	NV	11.38	13.63	12.00	5.44	–11.96
	TT	2,164.00	2,789.00	2,186.51	1.03	–21.60
	TD	1,405.00	1,431.00	1,474.37	4.93	3.03
RC2	NV	3.25	4.25	3.25	0.00	–23.52
	TT	2,177.00	3,366.00	2,004.00	–7.92	–40.45
	TD	1,444.00	1,128.00	1,421.00	–1.53	266.05
C1	NV	10.00	10.00	10.66	6.66	6.66
	TT	9,729.00	9,875.00	10,331.89	5.99	4.42
	TD	879.00	828.00	963.88	9.65	16.41
C2	NV	3.00	3.00	3.37	12.50	12.50
	TT	9,563.00	9,628.00	9,870.75	3.21	2.52
	TD	657.00	592.00	730.25	11.14	23.35

Table 4 Result comparison in TD1 and update mechanism 2

Instance		*Reference*		*Harmony search algorithm*		
		Figliozzi (2012)	*Kumar and Pannerselvam (2015)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*Gap against Kumar and Pannerselvam (%)*
R1	NV	11.67	13.58	12.33	5.68	–9.18
	TT	2,080.00	2,691.00	2,219.91	6.72	–17.50
	TD	1,295.00	1,624.00	1,351.58	4.47	6.92
R2	NV	2.882	3.64	2.81	–0.06	–22.57
	TT	1,990.00	3,077.00	1,988.27	–0.08	–35.28
	TD	1,216.00	939.00	1,083.00	–10.93	15.33

Table 4 Result comparison in TD1 and update mechanism 2 (continued)

Instance		*Reference*		*Harmony search algorithm*		
		Figliozzi (2012)	*Kumar and Pannerselvam (2015)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*Gap against Kumar and Pannerselvam (%)*
RC1	NV	11.38	13.63	12.62	10.94	–7.41
	TT	2,164.00	2,789.00	2,397.62	10.79	–14.03
	TD	1,405.00	1,431.00	1,479.62	5.31	3.36
RC2	NV	3.25	4.25	3.00	–7.69	–29.41
	TT	2,177.00	3,366.00	2,147.37	–1.36	–36.20
	TD	1,444.00	1,128.00	1,260.37	–12.71	11.73
C1	NV	10.00	10.00	10.78	7.77	7.77
	TT	9,729.00	9,875.00	10,445.33	7.36	5.77
	TD	879.00	828.00	941.22	7.07	13.67
C2	NV	3.00	3.00	3.37	12.50	12.50
	TT	9,563.00	9,628.00	9,864.37	3.15	2.45
	TD	657.00	592.00	679.00	3.44	14.80

Table 5 Result comparison in TD1 and update mechanism 3

Instance		*Reference*		*Harmony search algorithm*		
		Figliozzi (2012)	*Kumar and Pannerselvam (2015)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*Gap against Kumar and Pannerselvam (%)*
R1	NV	11.67	13.58	12.25	4.97	–9.79
	TT	2,080.00	2,691.00	2,213.41	6.41	–17.74
	TD	1,295.00	1,624.00	1,297.50	0.19	2.65
R2	NV	2.882	3.64	2.90	3.15	–20.07
	TT	1,990.00	3,077.00	2,088.90	4.97	–32.11
	TD	1,216.00	939.00	1,058.54	–12.94	12.73
RC1	NV	11.38	13.63	12.37	8.744	–9.20
	TT	2,164.00	2,789.00	2387.5	10.32	–14.39
	TD	1,405.00	1,431.00	1,438.37	2.37	0.51
RC2	NV	3.25	4.25	3.00	–7.69	–29.41
	TT	2,177.00	3,366.00	2,173.87	–0.14	–35.41
	TD	1,444.00	1,128.00	1,256.75	–12.96	11.41
C1	NV	10.00	10.00	10.66	6.66	6.66
	TT	9,729.00	9,875.00	10,503.67	7.96	6.36
	TD	879.00	828.00	884.44	0.61	6.81
C2	NV	3.00	3.00	3.25	8.33	8.33
	TT	9,563.00	9,628.00	9,792.37	2.39	1.70
	TD	657.00	592.00	5,588.75	–10.38	–0.54

Table 6 Result comparison in TD2 with all three update mechanisms

Instance		Figliozzi (2012)	Update mechanism 1		Update mechanism 2		Update mechanism 3	
			HSA	*Gap against Figliozzi (2012) (%)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*HSA*	*Gap against Figliozzi (2012) (%)*
R1	NV	10.75	11.50	3.97	11.00	2.32	10.83	0.77
	TT	1,897.00	2,080.08	9.65	2,076.75	9.447	2,003.91	5.63
	TD	1,258.00	1,331.01	5.80	1,228.66	–2.33	1,188.84	–5.53
R2	NV	2.55	3.00	17.64	2.54	–0.17	2.72	6.95
	TT	1,861.00	1,996.54	7.28	1,912.45	2.76	1,965.36	5.60
	TD	1,244.00	1,407.18	13.11	1,260.63	1.33	1,133.18	–8.90
RC1	NV	10.50	11.50	9.52	11.00	4.76	11.25	7.14
	TT	1,989.00	2,146.75	7.93	2,145.75	7.88	2,138.87	7.53
	TD	1,395.00	1,500.50	7.56	1,428.01	2.36	1,327.00	–4.87
RC2	NV	2.88	3.25	12.84	3.00	4.16	3.12	8.51
	TT	1,993.00	2,172.01	8.98	2,198.62	10.31	2,219.37	11.35
	TD	1,454.00	1,697.62	16.75	1,627.25	11.91	1,361.50	–6.36
C1	NV	10.00	11.33	13.33	11.33	13.33	10.55	5.55
	TT	9,644.00	10,660.22	10.53	10,734.33	11.300	10,383.89	7.67
	TD	864.00	1,102.88	27.64	1,049.00	21.41	921.00	6.59
C2	NV	3.00	3.37	12.50	3.25	8.33	3.12	4.16
	TT	9,495.00	9,809.12	3.31	9,824.50	3.47	9,702.25	2.18
	TD	654.00	664.25	1.56	776.37	18.71	596.37	–8.81

Table 7 Result comparison in TD3 with all three update mechanisms

Instance		*Figliozzi (2012)*	*Update mechanism 1*		*Update mechanism 2*		*Update mechanism 3*	
			HSA	*Gap against Figliozzi (2012) (%)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*HSA*	*Gap against Figliozzi (2012) (%)*
R1	NV	9.92	11.16	12.56	10.33	4.16	10.50	5.84
	TT	1,793.00	11,976.83	10.25	1,962.25	9.43	1,935.08	7.92
	TD	1,237.00	1,401.33	13.28	1,250.50	1.09	1,202.75	–2.76
R2	NV	2.27	2.90	28.15	2.63	16.13	2.63	16.13
	TT	1,774.00	1,900.18	7.11	1,852.54	4.42	1,871.72	5.50
	TD	1,269.00	1,500.09	18.21	1,292.81	1.87	1,100.00	–13.31
RC1	NV	10.00	11.50	15.00	10.62	6.25	10.62	6.25
	TT	1,860.00	2,069.87	11.28	2,028.37	9.05	2,021.62	8.68
	TD	1,362.00	1,568.12	15.13	1,449.50	6.42	1,531.00	–0.80
RC2	NV	2.75	3.50	27.27	2.87	4.54	3.12	13.63
	TT	1,867.00	2,278.00	22.01	2,022.00	8.30	2,065.50	10.63
	TD	1,434.00	2,148.62	49.83	1,683.75	17.41	1,280.00	–10.74
C1	NV	10.00	11.55	15.55	11.22	12.22	10.55	5.55
	TT	9,608.00	10,601.11	10.33	10,620.56	10.53	10,323.56	7.44
	TD	880.00	1,227.22	39.45	1,062.22	20.70	899.44	2.20
C2	NV	3.00	3.37	12.50	3.12	4.16	3.00	0.00
	TT	9,485.00	8,523.37	-10.13	9,670.12	1.95	9,578.50	0.98
	TD	697.00	847.25	21.55	728.00	4.44	590.12	–15.33

Table 8 Result comparison in TD1 and update mechanism 3

Instance		*Reference*		*Genetic algorithm*			*Harmony search algorithm*			
		Figliozzi (2012)	*Kumar and Pannerselvam (2015)*	*GA*	*Gap against Figliozzi (2012) (%)*	*Gap against Kumar and Pannerselvam (%)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*Gap against Kumar and Pannerselvam (%)*	*Gap against GA*
R1	NV	11.67	13.58	12.67	8.57	–6.70	12.52	4.97	–9.79	–3.31
	TD	1,295.00	1,264.00	1,031.17	–20.37	–18.42	1,297.50	0.19	2.65	25.83
	TT	2,080.00	2,691.00	2,113.33	1.60	–21.47	2,213.41	6.41	–17.75	4.74
R2	NV	2.82	3.64	4.09	45.04	12.36	2.90	2.84	–20.33	–29.10
	TD	1,216.00	939.00	1,036.73	–14.74	10.41	1,058.54	–12.95	12.73	2.10
	TT	1,990.00	3,077.00	2,562.55	28.77	–16.72	2,088.90	4.97	–32.11	–18.49
RC1	NV	11.38	13.63	13.12	15.29	–3.74	12.37	8.70	–9.24	–5.72
	TD	1,405.00	1,431.00	1,184.38	–15.70	–17.23	1,438.37	2.38	0.52	21.44
	TT	2,164.00	2,789.00	2,198.50	1.59	–21.17	2,387.50	10.33	–14.40	8.60
RC2	NV	3.25	4.25	4.00	23.08	–5.88	3.00	–7.69	–29.41	–25.00
	TD	1,444.00	1,128.00	1,266.00	–12.32	12.23	1,256.75	–12.97	11.41	–0.73
	TT	2,177.00	3,366.00	2,703.13	24.17	–19.69	2,173.87	–0.14	–35.42	–19.58
C1	NV	10.00	10.00	10.88	8.80	8.80	10.66	6.60	6.60	–2.02
	TD	879.00	828.00	903.11	2.74	9.07	884.44	0.62	6.82	–2.07
	TT	9,729.00	9,875.00	9,789.11	0.62	–0.87	10,503.67	7.96	6.37	7.30
C2	NV	3.00	3.00	3.50	16.67	16.67	3.25	8.33	8.33	–7.14
	TD	657.00	592.00	544.87	–17.07	–7.96	588.75	–10.39	–0.55	8.05
	TT	9,563.00	9,628.00	9,791.50	2.39	1.70	9,792.37	2.40	1.71	0.01

Table 6 summarises the comparison between HSA performance in 56 benchmark instances and previous literature in speed matrix 2 (TD2) from three different update mechanisms. Table 7 illustrates the comparison between HSA performance in 56 benchmark instances and previous literature in speed matrix 3 (TD3) in all different update mechanisms. Finally, Tables 8 to10 show the comparison between GA and HSA and GA, HSA to Figliozzi (2012) and Kumar and Pannerselvam (2015). It is important to mention that GA only tested on the update mechanism 3 since it performs the best when applied to HSA, as observed from Tables 3 to 7.

Table 9 Result comparison in TD2 and update mechanism 3

Instance		*Figliozzi (2012)*	*Genetic algorithm*		*Harmony search algorithm*		
			GA	*Gap against Figliozzi (2012) (%)*	*HSA*	*Gap against Figliozzi (2012) (%)*	*Gap against GA*
R1	NV	10.75	11.83	10.05	10.83	0.744	–8.45
	TD	1,258.00	1172.83	–6.77	1,188.41	–5.53	1.33
	TT	1,897.00	2,024.08	7.65	2,003.91	5.64	–1.87
R2	NV	2.55	2.81	10.20	2.72	6.67	–3.20
	TD	1,244.00	1,219.45	–1.97	1,133.81	–8.91	–7.07
	TT	1,861.00	1,836.90	–1.30	1,965.36	5.61	6.99
RC1	NV	10.50	12.00	14.29	11.25	7.14	–6.25
	TD	1,395.00	1,372.00	–1.65	1,327.00	–4.87	–3.28
	TT	1,989.00	2,112.669	6.22	2,138.87	7.53	1.24
RC2	NV	2.88	3.25	12.85	3.12	8.33	–4.00
	TD	1,454.00	1,293.00	–11.07	1,361.5	–6.36	5.30
	TT	1,993.00	2,243.98	12.59	2,219.37	11.36	–1.10
C1	NV	10.00	10.55	5.50	10.55	5.50	0.00
	TD	864.00	1,041.55	20.55	921.00	6.60	–11.57
	TT	9,644.00	9,872.88	2.37	10,383.89	7.67	5.18
C2	NV	3.00	3.25	8.33	3.12	4.00	–4.00
	TD	654.00	668.75	2.26	596.37	–8.81	–10.82
	TT	9,495.00	9,735.75	2.54	9,702.25	2.18	–0.34

From Tables 3 to 7, it can be observed that HSA can perform better in some of the benchmark instances. For example, when comparing against Figliozzi (2012), HSA yields near-optimal solutions in all NV, TD and TT. However, since the primary objective in this research is to minimise the NV and the secondary objective is to minimise the TT, the result from HSA is comparable with Figliozzi (2012) and performs better in NV and TT when compared to Kumar and Pannerselvam (2015). Furthermore, from Table 8, it can be observed that both HSA and GA were able to outperform Kumar and Pannerselvam (2015) in 9 out of 18 objectives and 11 out of 18 objectives, respectively, while compared to Figliozzi (2012), HSA outperformed 6 out of 18 objectives and GA obtained better solution in 5 out of 18 objectives speed matrix 1. Similarly, Table 9 and Table 10 present the result comparison of both GA and HSA when compared Figliozzi (2012) and Kumar and Pannerselvam (2015) in speed matrix 2 and speed matrix 3. Finally, it can be concluded that even though GA outperforms HSA in terms of the number of better solutions, it is important to point out that HSA still outperforms GA as it can be observed from the solution quality (average of all instances).

Table 10 Result comparison in TD3 and update mechanism 3

Instance		*Figliozzi (2012)*	*Genetic algorithm*		*Harmony search algorithm*		
			GA	*Gap against Figliozzi (2012) (%)*	*HSA*	*Gap against Figliozzi (%)*	*Gap against GA*
R1	NV	9.92	10.66	7.46	10.50	5.85	–1.50
	TD	1,237.00	1,231.41	–0.45	1,202.75	–2.77	–2.33
	TT	1,793.00	1,973.00	10.04	1,935.08	7.92	–1.92
R2	NV	2.27	2.72	19.82	2.63	15.86	–3.31
	TD	1,269.00	1,232.00	–2.92	1,100.00	–13.31	–10.71
	TT	1,774.00	1,840.18	3.73	1,871.72	5.51	1.71
RC1	NV	10.00	10.62	6.20	10.62	6.20	0.00
	TD	1,362.00	1,354.75	–0.53	1,351.00	–0.81	–0.28
	TT	1,860.00	1,991.37	7.06	2,021.62	8.69	1.52
RC2	NV	2.75	3.00	9.09	3.12	13.45	4.00
	TD	1,434.00	1,672.62	16.64	1,280.00	–10.74	–23.47
	TT	1,867.00	2,045.25	9.55	2,065.50	10.63	0.99
C1	NV	10.00	10.88	8.80	10.55	5.50	–3.03
	TD	880.00	875.77	–0.48	899.44	2.21	2.70
	TT	9,608.00	10,378.44	8.02	10,323.56	7.45	–0.53
C2	NV	3.00	3.13	4.17	3.00	0.00	–4.00
	TD	697.00	689.87	–1.02	590.122	–15.33	–14.46
	TT	9,485.00	9,644.5	1.68	9,578.50	0.99	–0.68

5 Conclusions

This study has successfully proposed a HSA and a GA for the TDVRPTW. When comparing with two existing algorithms in recent literature, the proposed algorithm shows competitive performance. Besides, this study may be further improved and validated by using different speed matrices, considering different LSOs such as the insertion heuristic, or trying to apply other metaheuristic algorithms such as virus optimisation algorithm proposed by Liang and Cuevas (2016).

References

Adewumi, A.O. and Adeleke, O.J. (2018) 'A survey of recent advances in vehicle routing problems', *International Journal of System Assurance Engineering and Management*, Vol. 9, No. 1, pp.155–172.

Alinaghian, M. and Naderipour, M. (2016) 'A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: a case study', *Computers and Industrial Engineering*, Vol. 100, No. 99, pp.210–222.

Alvarenga, G.B., Mateus, G.R. and de Tomi, G. (2007) 'A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows', *Computers and Operations Research*, Vol. 34, No. 6, pp.1561–1584.

Dantzig, G. and Ramser, J. (1959) 'The truck dispatching problem', *Management Science*, Vol. 6, No. 1, pp.80–91.

Donati, M.N. (2008) 'Time-dependent vehicle routing problem with a multi ant colony', *European Journal of Operational Research*, Vol. 185, No. 3, pp.1174–1191.

Figliozzi, M. (2008) *An Iterative Route Construction and Improvement Algorithm for the Vehicle Routing Problem with Soft and Hard Time Windows*, Working Paper, Portland State University.

Figliozzi, M. (2012) 'The time-dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 3, No. 48, pp.616–636.

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G. and Stobbe, M. (2017) 'Metaheuristic for the time-dependent pollution-routing problem', *European Journal of Operational Research*, Vol. 259, No. 3, pp.972–991.

Gansterer, M. and Hartl, R.F. (2018) 'Collaborative vehicle routing: a survey', *European Journal of Operational Research*, Vol. 268, No. 1, pp.1–12.

Gansterer, M. and Hartl, R.F. (2020) 'Shared resources in collaborative vehicle routing', *Top*, Vol. 28, No. 1, pp.1–20.

Gunawan, A., Kendall, G., McCollum, B., Seow, H.V. and Lee, L.S. (2021) 'Vehicle routing: review of benchmark datasets', *Journal of the Operational Research Society*, Vol. 72, No. 8, pp.1794–1807.

Huang, Y., Zhao, L., Van Woensel, T. and Gross, J.P. (2017) 'Time-dependent vehicle routing problem with path flexibility', *Transportation Research Part B: Methodological*, Vol. 100, No. 95, pp.169–195.

Ichoua, S., Gendreau, M. and Potvin, J-Y. (2003) 'Vehicle dispatching with time-dependent travel times', *European Journal of Operational Research*, Vol. 144, No. 2, pp.379–396.

Iqbal, S., Kaykobad, M. and Rahman, M.S. (2015) 'Solving the multi-objective vehicle routing problem with soft time windows with the help of bees', *Swarm and Evolutionary Computation*, Vol. 24, No. 5, pp.50–64 [online] https://doi.org/10.1016/j.swevo.2015.06.001.

Koç, Ç., Bektaş, T., Jabali, O. and Laporte, G. (2016) 'Thirty years of heterogeneous vehicle routing', *European Journal of Operational Research*, Vol. 249, No. 1, pp.1–21.

Kumar, S.N. and Pannerselvam, R. (2015) 'A time-dependent vehicle routing problem with time windows for e-commerce supplier site pickups using genetic algorithm', *Intelligent Information Management*, Vol. 7, No. 4, pp.181–194.

Lenstra, J.K. and Kan, A.H. (1981) 'Complexity of vehicle and scheduling problems', *Networks*, Vol. 11, No. 2, pp.221–227.

Lera-Romero, G. and Miranda-Bront, J.J. (2018) 'Integer programming formulations for the time-dependent elementary shortest path problem with resource constraints', *Electronic Notes in Discrete Mathematics*, Vol. 69, No. 6, pp.53–60 [online] https://doi.org/10.1016/j.endm.2018.07.008.

Liang, Y-C. and Cuevas, J.R. (2016) 'A novel metaheuristic for continuous problem: virus optimization algorithm', *Engineering Optimization*, Vol. 48, No. 1, pp.73–93.

Liu, C., Kou, G., Zhou, X., Peng, Y., Sheng, H. and Alsaadi, F.E. (2020) 'Time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach', *Knowledge-Based Systems*, Vol. 188, No. 2, p.104813 [online] https://doi.org/10.1016/j.knosys.2019.06.021.

Malandraki, C. and Daskin, M.S. (1992) 'Time-dependent vehicle routing problems-formulations, properties and heuristic algorithms', *Journal of Transportation Science*, Vol. 26, No. 3, pp.185–200.

Masutti, T.A. and de Castro, L.N. (2017) 'Bee-inspired algorithms applied to vehicle routing problems: a survey and a proposal', *Mathematical Problems in Engineering*, Vol. 2017, pp.1–20, ID: 3046830 [online] https://doi.org/10.1155/2017/3046830.

Norouzi, N., Amainick, M.S. and Moghaddam, R.T. (2015) 'A time-dependent vehicle routing problem solved by improved simulated annealing', *Proceedings of the Romanian Academy Series A: Mathematics, Physics, Technical Science, Information Science*, Vol. 16, pp.458–465.

Qi, M., Miao, L., Zhang, L. and Xu, H. (2008) 'A new tabu search heuristic algorithm for the vehicle routing problem with time windows', *15th Annual Conference Proceedings of the 2008 International Conference on Management Science and Engineering*, ICMSE, California, USA, 10–12 September, pp.1648–1653.

Szczepański, E., Żak, J., Jacyna-Gołda, I. and Murawski, J. (2017) 'Simulation support of freight delivery schedule in urban areas', *Procedia Engineering*, Vol. 100, No. 187, pp.520–525.

Wang, X. and Wasil, E. (2021) 'On the road to better routes: five decades of published research on the vehicle routing problem', *Networks*, Vol. 77, No. 1, pp.66–87.

Yang, B., Hu, Z.H., Wei, C., Li, S.Q., Zhao, L. and Jia, S. (2015) 'Routing with time-windows for multiple environmental vehicle types', *Computers and Industrial Engineering*, Vol. 100, No. 89, pp.150–161.

Yildirim, U.M. and Çatay, B. (2009) 'An ant colony algorithm for time-dependent vehicle routing problem with time windows', in Fleischmann, B., Borgwardt, K.H., Klein, R. and Tuma, A. (Eds.): *Operations Research Proceedings 2008*, Springer, Berlin, Heidelberg, pp.337–342.

Zhang, D., Cai, S., Ye, F., Si, Y.W. and Nguyen, T.T. (2017) 'A hybrid algorithm for a vehicle routing problem with realistic constraints', *Information Sciences*, Vol. 100, Nos. 394–395, pp.167–182.

Zhang, R. and Pavone, M. (2016) 'Control of robotic mobility-on-demand systems: a queueing-theoretical perspective', *The International Journal of Robotics Research*, Vol. 35, Nos. 1–3, pp.186–203.