Engineering Applications of Artificial Intelligence 95 (2020) 103901

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window

Yong Shi [a,b], Yanjie Zhou [c,*], Toufik Boudouh [d], Olivier Grunder [b]

[a] UMR SADAPT, INRAE, University of Paris-Saclay, 16 rue claude bertand, Paris, 75005, France
[b] Nanomedicine Lab, Univ. Bourgogne Franche-Comté, UTBM, rue mieg, Belfort, 90010, France
[c] Department of Industrial Engineering, Pusan National University, Busan, 46241, Republic of Korea
[d] ICB UMR 6303, CNRS, Univ. Bourgogne Franche-Comté, UTBM, rue de Leupe, Sevenans, 90400, France

ARTICLE INFO

Keywords:
Simultaneous pickup–delivery problem
Time window
Tabu search
Variable neighborhood search
Vehicle routing problem
Two-stage algorithm
Learning-based approach

ABSTRACT

Vehicle routing problem with simultaneous pickup–delivery and time window (VRPSPDTW) is computationally challenging as it generalizes the classical and NP-hard vehicle routing problem. According to the state-of-the-art, VRPSPDTW usually has two hierarchical optimization objectives: a primary objective of minimizing the number of vehicles (NV) and a secondary objective of reducing the transportation distance (TD). Given the existing research and our trial results, we find that the optimization of TD is not necessarily a promotion for reducing NV. In this paper, an effective learning-based two-stage algorithm, which has never been studied before, is proposed to solve the VRPSPDTW. In the first stage, a modified variable neighborhood search with a learning-based objective function is proposed to minimize the primary objective with retaining the potential structures. In the second stage, a bi-structure based tabu search (BSTS) is designed to optimize the primary and secondary objectives further. The experimental results on 93 benchmark instances demonstrate that the proposed algorithm performs remarkably well both in terms of computational efficiency and solution quality. In particular, the proposed two-stage algorithm improve several best known solutions (either a better NV or a better TD when NV are the same) from the state-of-the-art. To our knowledge, this is the first learning-based two-stage algorithm for solving VRPSPDTW reaching such a performance. Finally, we empirically analyze several critical components of the algorithm to highlight their impacts on the performance of the proposed algorithm.

1. Introduction

In recent years, with economic globalization, logistics companies are facing fierce competition. To improve the competitive power, logistics companies have to make a great effort to decrease their delivery costs, which include fixed vehicle cost, transportation cost, etc. Generally, customers send service requests, including pickup request or delivery request (individually or simultaneously) first. Logistics companies group customers based on their location. At the same time, logistics companies will also consider the limited number of employees and group customers with employees to assign tasks. Finally, the logistics companies make a route plan for each assigned vehicle, which starts from a warehouse and returns to the warehouse after servicing their customers. This problem is generally defined as pickup and delivery problem, which has been applied in many fields, such as the grocery delivery system, parcel delivery, and home health care services. If the delivery and pickup services are demanded in the logistics system, we call this pickup and delivery problem. When the customers have pickup requests and delivery requests simultaneously, the variant pickup and delivery problem becomes a vehicle routing problem with simultaneous pickup–delivery (VRPSPD).

In the same special conditions of VRPSPD, customers may have a service time constraint request, such as the customer only available from 10:00 AM to 10:30 AM, which is normally defined as time windows. When the customers have time windows constraints, the variant VRPSPD is defined as vehicle routing problem with simultaneous pickup–delivery with time window (VRPSPDTW).

There are two main objective functions for logistics companies to decrease their total cost. The primary one, the logistics companies can reduce the total number of assigned vehicles which involves the depreciation cost (or dispatching cost) of vehicles, driver's salary, and transportation costs. Secondly, the logistics companies can optimize the routes (routing cost) of the assigned vehicles. Lai and Cao (2010) minimized the total transportation cost for the VRPSPDTW by implementing an improved differential evolution algorithm (IDE). Wang

* Corresponding author.
E-mail addresses: shiyonghbwh@gmail.com (Y. Shi), ieyjzhou@pusan.ac.kr (Y. Zhou), toufik.boudouh@utbm.fr (T. Boudouh), olivier.grunder@utbm.fr (O. Grunder).

https://doi.org/10.1016/j.engappai.2020.103901
Received 16 January 2020; Received in revised form 24 July 2020; Accepted 16 August 2020
Available online 28 August 2020

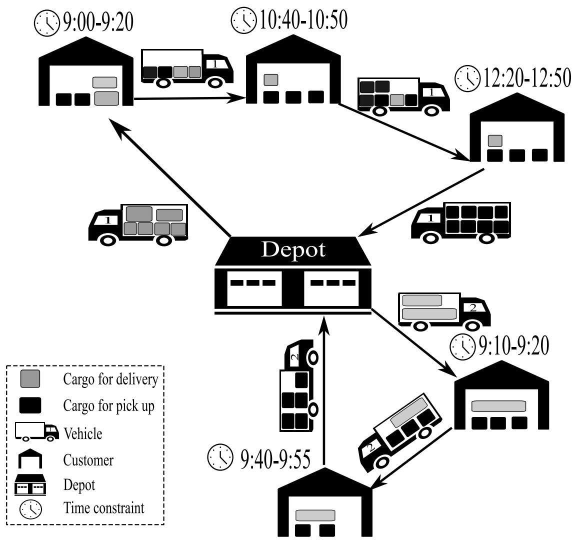

Fig. 1. An example of the VRPSPDTW.

and Chen (2012) studied VRPSPDTW by introducing a co-evolution genetic algorithm (CoGA). After that, Wang et al. (2015) introduced a parallelized simulated annealing (p-SA) to solve the instances proposed by Wang and Chen (2012) and improved the solution quality. Shi et al. (2018a) also considered this model by proposing a tabu based framework. The preliminary results in a conference paper showed that most of the solutions obtained are worse performance than the p-SA in Wang et al. (2015), despite the fact that several solutions are improved. Our research can be viewed as a further extension of the conference paper (Shi et al., 2018a), and the algorithm designed has been completely improved both from the structure and operators.

Addition, in the study of Wang and Chen (2012), Wang et al. (2015) and Shi et al. (2018a), they assume that the solution with less number of vehicles (NV) always dominates the solution with more NV, regardless of the Transportation Distance (TD). Given that we need to make a full comparison with the state-of-art to verify the efficiency of our proposed algorithm, we also assume that NV is the primary objective value, and TD is the secondary objective function. According to the experimental results from Wang and Chen (2012) and Wang et al. (2015), as well as our several trial results, we find the following features of the solution.

- In the benchmarks instances of Wang and Chen (2012), a lot of solutions have the feature that reducing TD leads to the increasing of NV. For example, in the instance Rdp 105, according to Wang and Chen (2012) and Wang et al. (2015), when NV = 15, the best found TD = 1375.31, while when NV = 14, the best found TD = 1399.81. This phenomenon reveals that decreasing of NV and TD are not in the same direction.
- Our trial experiments illustrate that reducing NV is much more difficult than decreasing TD when best NV is given.
- Even if we make a weighted objective function, such as obj = $100\,000 \times$ NV + TD, we find that variable neighborhood search (VNS) and tabu search (TS) are not able to improve NV better compared to the results from Wang and Chen (2012) and Wang et al. (2015). This motivates us to find a new way to decrease NV.

In this study, we propose a learning-based two-stage algorithm, also named as variable neighborhood search and bi-structure based tabu search algorithm (VNS-BSTS), to solve the VRPSPDTW with the following motivations and contributions.

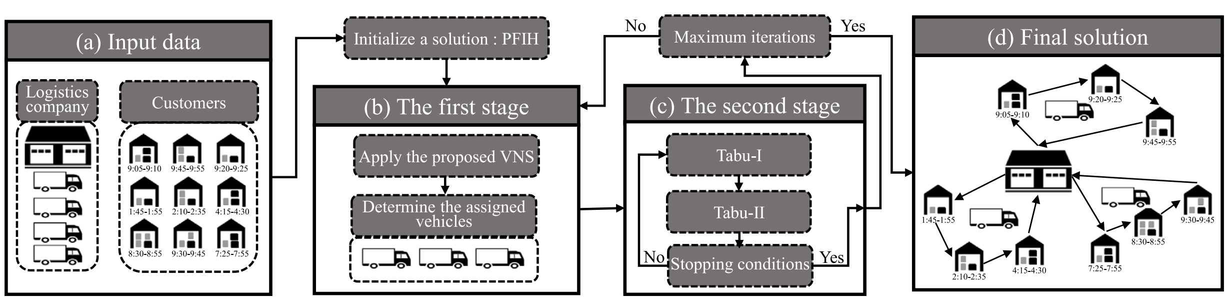

Fig. 2. An overview of the proposed two-stage algorithm: (a) input data, (b) the first stage: variable neighborhood search, (c) the second stage: Tabu search, (d) final solution.

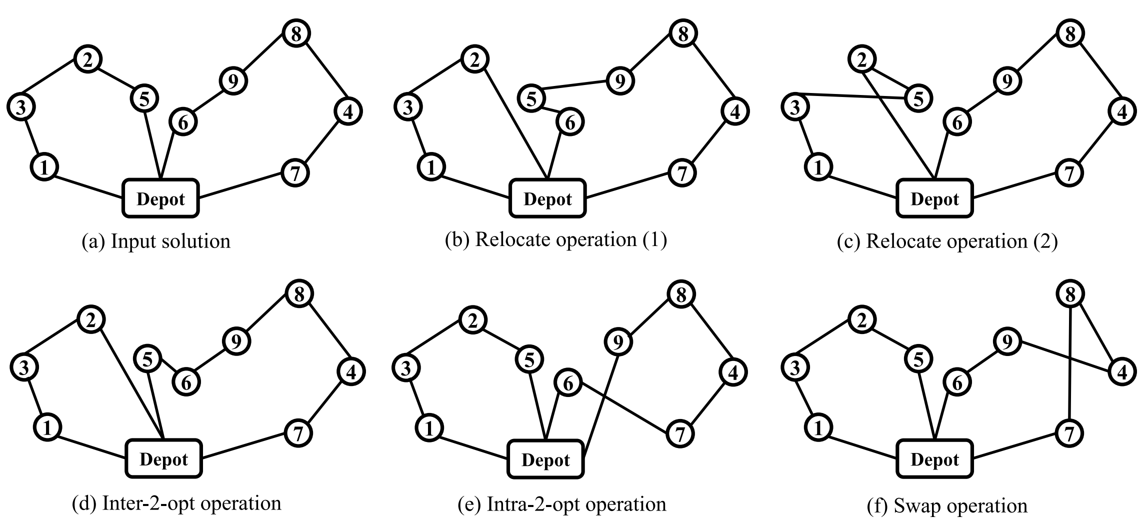

Fig. 3. An example of neighborhood of a given solution by using relocate operation, inter-2-opt operation, intra-2-opt operation and swap operation.

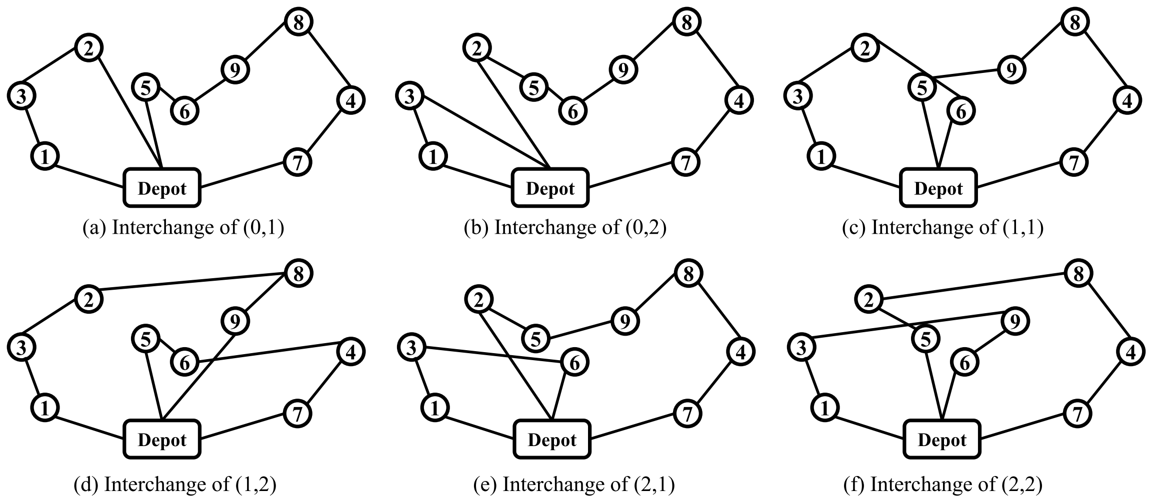

Fig. 4. An example of λ−interchange generation mechanism with $\lambda = 2$.

- According to the existing studies (Bent and Van Hentenryck, 2004; Nagata and Bräysy, 2009; Chen and Hao, 2018), an aggregated weighted objective function often drives the search for solutions with low travel costs, and this makes it difficult to reach a solution with fewer routes but the higher travel cost. To overcome this weakness, we divide the searching process into two stages, which are implemented by VNS and improved TS, respectively.
- Since the simple objective function f = NV is not able to help to optimize NV, we design the new objective function of the first stage by introducing a new learning-based evaluation framework, which is inspired from Graph Coloring Problem.
- Given that a pure TS is still easily trapped into local optima and a diversity neighborhood or tabu lists may further improve the global searching ability and overcome the local optima, we designed a bi-structure based tabu search (BSTS) for improving the searching ability in the second stage.

The remainder of this study is organized as follows. Section 2 reviews the related works of VRPSPDTW. Section 3 presents the mathematical formulations of VRPSPDTW. Section 4 introduces the proposed two-stage algorithm. Numerical experimental results are shown in Section 5. Analysis and discussions of the proposed two-stage algorithm are studied in Section 6. Finally, the conclusions and perspectives are presented in Section 7.

2. Literature review

In this section, three categories of the related works, including VRPSPD, VRPSPDTW, VNS and Tabu Search are presented. The detail shows in the followings.

Vehicle Routing Problem (VRP), regarded as a crucial application in transportation science and logistics, is computationally challenging due to its NP-hard nature. In recent years, VRP and its variants have attracted considerable attention (Prins, 2009; Peng et al., 2019). One of the most significant challenges of their work is developing efficient algorithms for solving their instances. The algorithms can be classified into two categories, including exact algorithms (Yu et al., 2019), heuristic algorithms (Solomon, 1987; Solomon and Desrosiers, 1988) and meta-heuristic algorithms (Luo et al., 2016; Shi et al., 2017).

2.1. VRPSPD

As an important extension of capacitated VRP, the vehicle routing problem with simultaneous pickup–delivery service was originally defined by Min (1989). After ten years, Min (1989), Dethloff (2001)

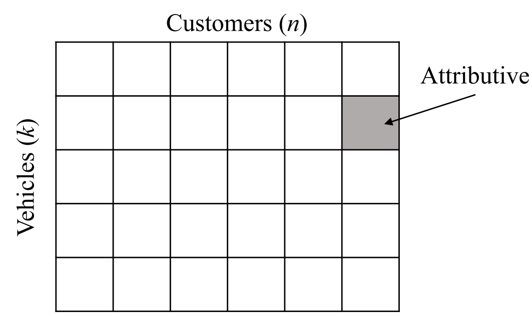

Fig. 5. The structure of tabu list of TL_1.

Table 1
The if-then rules for utilizing the lexicographic evaluation function (2)) in VNS.

If	Then
$e_1(s_1) < e_1(s_2)$	s_1 is better than s_2
$e_1(s_1) = e_2(s_2)$ and $e_2(s_1) < e_2(s_2)$	s_1 is better than s_2
$e_1(s_1) = e_2(s_2)$ and $e_2(s_1) = e_2(s_2)$ and $e_3(s_1) < e_3(s_2)$	s_1 is better than s_2
$e_1(s_1) = e_2(s_2)$ and $e_2(s_1) = e_2(s_2)$ and $e_3(s_1) = e_3(s_2)$ and $e_4(s_1) < e_4(s_2)$	s_1 is better than s_2

and Montané and Galvão (2002) are among the earliest to study the VRPSPDTW. Montané and Galvao (2006) proposed a TS to solve the vehicle routing problem with simultaneous pickup–delivery service. Fan (2011) initialized the solution of VRPSPD by proposing the cheapest insertion method and then improved by a TS. Additionally, VRPSPD was also solved by various techniques including hybrid meta-heuristic algorithm (Zachariadis et al., 2009), adaptive memory methodology (Zachariadis et al., 2010), saving-based ant algorithm (Çatay, 2010), perturbation based algorithm (Jun and Kim, 2012), hybrid discrete particle swarm algorithm (Goksal et al., 2013). Polat et al. (2015) introduced a perturbation-based variable neighborhood search heuristic for VRPSPD with time limit. Avci and Topaloglu (2015) designed an adaptive local search algorithm for the vehicle routing problem with simultaneous and mixed pickups and deliveries. Zachariadis et al. (2016) studied the VRPSPD with two-dimensional loading constraints. Kartal et al. (2017) addressed the simultaneous pickup and delivery considering a single allocation p-hub median location and routing problem.

Xu et al. (2017) studied an unpaired pickup and delivery vehicle routing problem with multi-commodity and multi-visit. Belgin et al. (2018) investigated a two-echelon VRPSPD and proposed a variable neighborhood descent with local search to solve it. Wang et al. (2018)

Table 2
An example of regency-based memory for TL_1.

Vehicles	Customers								
	1	2	3	4	5	6	7	8	9
1	−1	0	2	3	−2	−3	2	1	0
2	2	1	−3	2	0	1	−1	2	3

Table 3
An example of frequency-based memory for TL_1.

Vehicles	Customers								
	1	2	3	4	5	6	7	8	9
1	2	0	2	3	1	2	2	1	2
2	3	1	4	2	0	1	0	2	1

Table 4
Parameter settings for algorithms used in this study.

Algorithms	Parameters	Values
Two-stage	The maximum iteration	20
VNS	k_{max} : the number of neighborhood structures	6
	The max iteration for non-improvement (terminal conditions)	100 × size
TS	α : the parameter for adjusting the diversification operator	0.1
	$tabu\_length$: the length for the tabu list	0.4 × size
	The max iteration for non-improvement (terminal conditions) for each tabu component	100

studied the vehicle routing optimization with delivery and pickup by considering the collaboration and transportation resource sharing. Shi et al. (2018b) investigated the stochastic home health care routing problem by considering the simultaneous delivery and pickup of the drugs, and a hybrid simulated annealing is designed to obtain the near-optimal solutions. Wang et al. (2018) proposed an effective heuristic by combining K-means clustering and NSGA-II to solve the vehicle routing optimization with delivery and pickup.

Exact algorithm including branch-and-price and branch-cut-and-price have been applied to solve VRPSPD (Subramanian et al., 2013; Dell'Amico et al., 2006)

2.2. VRPSPDTW

Despite the broad applications of VRPSPDTW, however, compared with VRPSPD, VRPSPDTW is somehow less investigated. Angelelli and Mansini (2002) are the first to study VRPSPDTW by introducing an exact algorithm (Branch & Price). One of the limitations of exact algorithms is that it is very time consuming and cannot obtain an acceptable solution in a reasonable time, especially for NP-hard problems. As the first attempt to solve VRPSPDTW by implementing heuristic, Cao and Lai (2007) proposed an improved genetic algorithm by designing a new decimal coding according to a prior sequence among all customers. And their experimental results showed that the improved algorithm could find the optimal or near-optimum solution effectively when the maximum number of customer equals to 8. However, eight customers are far inadequate in a practical situation. Lai and Cao (2010) introduced an IDE to solve the VRPSPDTW and given an example with eight customers and two vehicles. Lai et al. (2010) investigated VRPSPDTW by designing a two-stage hybrid meta-heuristic, in which the simulated annealing and tabu search are integrated. Abundant experimental results have been done to validate the proposed two-stage framework. Boubahri et al. (2011) proposed a multi-agent colonies system to address the VRPSPDTW. Despite the novelty of the solving approach, the authors did not provide any numerical experimental results. Lai and Tong (2012) addressed the VRPSPDTW by considering the hybrid tabu search and ant colony optimization. Experimental results show the good performance of the proposed hybrid meta-heuristic algorithm.

Wang and Chen (2012) adopted a co-evolution genetic algorithm to investigate the VRPSPDTW and designed benchmark instances which have been generated based on the well-known Solomon's (Solomon, 1987) benchmark instances to evaluate the performance of the proposed co-evolution genetic algorithm. Their numerical experimental result showed that the proposed co-evolution genetic algorithm could provide better solutions within a comparatively shorter period compared with the commercial solver of CPLEX. Wang et al. (2015) proposed a parallel simulated annealing algorithm to solve the VRPSPDTW and tested on the benchmark instances proposed by Wang and Chen (2012). Their results showed that the proposed parallel simulated annealing algorithm could obtain the same results of Wang and Chen (2012) for small-scale benchmark instances with the number of customers varying from 10 to 50. For the medium-scale benchmark instances with the number of customers equaling to 100, 12 instances have better performance and 44 instances have the same result compared with the study of Wang and Chen (2012). Recently, a tabu search based framework for solving the benchmark instances is developed by Shi et al. (2018a) in a conference paper. Accordingly, they did not conduct a complete experiments and only reported the preliminary results for some of the instances. Even though Shi et al. (2018a) obtained several better solutions than Wang et al. (2015), most of them are performed worse than Wang et al. (2015). Hof and Schneider (2019) recently designed an adaptive large neighborhood search with path relinking (ALNS-PR) for solving the class of VRPSPD, including the VRPSPDTW. Their published results demonstrate that the ALNS-PR can effectively obtain better solutions than the previous benchmark results (see Hof (2019)).

Table 5
Comparison between CPLEX, GA, p-SA and proposed two-stage algorithm of small-scale instances to the VRPSPDTW.

Instance/size	CPLEX (Wang and Chen, 2012)			CoGA		p-SA		VNS-BSTS			Gap (VS. CoGA)***	
	NV	TD	CT(s)	NV	TD	NV	TD	NV	TD	CT(s)	NV	TD(%)
RCdp1001/10	3	348.98	1	3	348.98	3	348.98	3	348.98	1.6	0	0.00
RCdp1004/10	2	216.69	1503	2	216.69	2	216.69	2	216.69	0.40	0	0.00
RCdp1007/10	2	310.81	25	2	310.81	2	310.81	2	310.81	0.82	0	0.00
RCdp2501/25	5	551.05	16	5	551.05	5	552.21	5	551.05	4.81	0	0.00
RCdp2504/25	7*	738.32*	485,660**	4	473.46	4	473.46	4	473.46	4.17	0	0.00
RCdp2507/25	7*	634.20*	439,321**	5	540.87	5	540.87	5	540.87	4.40	0	0.00
RCdp5001/50	9	994.18	327,404	9	994.18	9	994.7	9	994.18	15.90	0	0.00
RCdp5007/50	13*	1814.33*	1,546,429**	7	809.72	7	810.04	7	809.72	17.99	0	0.00
Minimum												0.00
Mean												0.00
Maximum												0.00

* The "out of memory" values (Wang and Chen, 2012).
**CPU times in seconds executed on an Intel Core 2 Quad 2.4 GHz with 1G memory (Wang and Chen, 2012).
*** $\text{Gap}_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{CoGA}}$, $\text{Gap}_{TD} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{CoGA}}}{TD_{\text{CoGA}}} * 100\%$.

Table 6
Comparison with the published best results.

Instance ID	BKS		VNS-BSTS			gap*			
	NV	TD	NV	TD	CT	NV	TD(%)	Is solution improved?	Is solution non-worse?
rdp101	19	1650.8	19	1650.8	34.91	0	0.00	0	1
rdp102	17	1486.12	17	1486.12	31.60	0	0.00	0	1
rdp103	13	1297.01	13	1294.75	38.26	0	−0.17	1	1
rdp104	10	984.81	10	984.81	65.48	0	0.00	0	1
rdp105	14	1377.11	14	1377.11	33.95	0	0.00	0	1
rdp106	12	1252.03	12	1261.4	43.65	0	0.75	0	0
rdp107	10	1121.86	10	1144.02	44.13	0	1.98	0	0
rdp108	9	965.54	9	968.32	67.67	0	0.29	0	0
rdp109	11	1194.73	11	1224.86	40.18	0	2.52	0	0
rdp110	10	1148.2	11	1101.33	50.91	1	−4.08	0	1
rdp111	10	1098.84	10	1117.76	46.72	0	1.72	0	0
rdp112	9	1010.42	10	961.29	65.87	1	−4.86	0	1
cdp101	11	976.04	11	976.04	60.29	0	0.00	0	1
cdp102	10	941.49	10	942.45	68.87	0	0.10	0	0
cdp103	10	892.98	10	896.28	95.40	0	0.37	0	0
cdp104	10	871.4	10	872.39	70.62	0	0.11	0	0
cdp105	10	1053.12	10	1080.63	41.46	0	2.61	0	0
cdp106	10	967.71	10	963.45	57.30	0	−0.44	1	1
cdp107	10	987.64	10	987.64	79.04	0	0.00	0	1
cdp108	10	932.88	10	934.41	79.24	0	0.16	0	0
cdp109	10	910.95	10	909.27	68.59	0	−0.18	1	1
rcdp101	14	1776.58	14	1708.21	32.04	0	−3.85	1	1
rcdp102	12	1583.62	13	1526.36	43.85	1	−3.62	0	1
rcdp103	11	1283.52	11	1336.05	46.00	0	4.09	0	0
rcdp104	10	1171.65	10	1177.21	43.81	0	0.47	0	0
rcdp105	14	1548.96	14	1548.38	42.41	0	−0.04	1	1
rcdp106	12	1392.47	12	1408.19	47.52	0	1.13	0	0
rcdp107	11	1255.06	11	1295.43	54.70	0	3.22	0	0
rcdp108	10	1198.36	10	1207.6	51.23	0	0.77	0	0
Rdp201	4	1253.23	4	1254.57	46.59	0	0.11	0	0
Rdp202	3	1191.7	3	1202.27	119.56	0	0.89	0	0
Rdp203	3	946.28	3	949.42	107.10	0	0.33	0	0
Rdp204	2	833.09	2	837.13	129.22	0	0.48	0	0
Rdp205	3	994.43	3	1027.49	79.63	0	3.32	0	0
Rdp206	3	913.68	3	938.63	96.08	0	2.73	0	0
Rdp207	2	890.61	2	912.26	101.46	0	2.43	0	0
Rdp208	2	726.82	2	737.26	173.85	0	1.44	0	0
Rdp209	3	909.16	3	940.29	77.00	0	3.42	0	0
Rdp210	3	939.37	3	945.97	87.14	0	0.70	0	0
Rdp211	2	904.44	3	805.22	82.25	1	−10.97	0	1
Cdp201	3	591.56	3	591.56	53.42	0	0.00	0	1
Cdp202	3	591.56	3	591.56	70.56	0	0.00	0	1
Cdp203	3	591.17	3	591.17	56.26	0	0.00	0	1
Cdp204	3	590.6	3	599.33	50.34	0	1.48	0	0
Cdp205	3	588.88	3	588.88	50.98	0	0.00	0	1
Cdp206	3	588.49	3	588.49	49.15	0	0.00	0	1
Cdp207	3	588.29	3	588.29	50.58	0	0.00	0	1
Cdp208	3	588.32	3	588.32	49.61	0	0.00	0	1
RCdp201	4	1406.94	4	1437.48	32.00	0	2.17	0	0
RCdp202	3	1414.55	3	1412.52	52.00	0	−0.14	1	1
RCdp203	3	1050.64	3	1064.95	74.56	0	1.36	0	0
RCdp204	3	798.46	3	813.74	85.46	0	1.91	0	0
RCdp205	4	1297.65	4	1316.06	33.28	0	1.42	0	0
RCdp206	3	1146.32	3	1154.36	52.15	0	0.70	0	0
RCdp207	3	1061.84	3	1098.64	60.77	0	3.47	0	0
RCdp208	3	828.14	3	843.3	73.71	0	1.83	0	0

* $\text{Gap}_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{BKS}}$, $\text{Gap}_{TD} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{BKS}}}{TD_{\text{BKS}}} * 100\%$.
† "1" = True, "0" = False.

2.3. Variable neighborhood search

VNS, as a classic heuristic algorithm, its fundamental idea is to use multiple different neighborhoods for system search. First, the smallest neighborhood is used to explore a new solution. While the solution cannot be improved, it switches to a slightly larger neighborhood. If it can continue to improve the solution, then fall back to the smallest neighborhood, otherwise continue to switch to a larger neighborhood. VNS has the characteristics of easy coding, simple operation, and high efficiency. Therefore, this algorithm is broadly used in seeking near-optimal solutions for NP-hard problems. Jarboui et al. (2013) designed an adaptive VNS for solving the classical location-routing problem. In their work, five different types of neighborhood structures are employed in the sharking and local search phases. Finally, the results in the classical benchmark instances showed the efficiency of the proposed VNS. Stenger et al. (2013) employed an adaptive VNS for solving a VRP arising in small package shipping. On the basis of cyclic exchange neighborhoods, they have proposed an effective VNS algorithm that integrates an adaptive mechanism to bias random sharking steps. This method has been successfully applied to solve the proposed problem and its extensions. The comparison performed with the state-of-art highlights the effectiveness of the VNS. Salhi et al. (2014) formulated a new mixed linear programming model to address the multi-depot VRP with a heterogeneous vehicle fleet. The problem is solved by CPLEX

Table 7
Experimental results compare with Wang et al. (2015) and Hof and Schneider (2019) for the large-scale customer instances.

Instance	Size	**P-SA** (Wang et al., 2015)			ALNS-PR (Hof and Schneider, 2019)			VNS-BSTS			Gap 1(VNS-BSTS VS Basic p-SA)*					Gap 2(VNS-BSTS VSBasic ALNS-PR)**				
		NV	**TD**	**CT**	NV	TD	CT	NV	TD	CT	NV	TD (%)	Is solution improved?	Is solution non-worse?	Is NV improved?	NV	TD (%)	Is solution improved?	Is solution non-worse?	Is NV improved?
C1_2_1	200	21	3169.52	62	20	2846.2	55.54	20	2846.2	55.54	−1	−10.20	1	1	1	0	0.00	0	1	0
R1_2_1	200	22	5083.39	89	20	4849.8	113.46	20	4856.09	61.89	−2	−4.47	1	1	1	0	0.13	0	0	0
RC1_2_1	200	20	3865.18	81	19	3652.18	77.05	19	3668.98	50.92	−1	−5.08	1	1	1	0	0.46	0	0	0
C2_2_1	200	6	1972.97	112	6	1931.44	194.73	6	1931.44	148.44	0	−2.10	1	1	0	0	0.00	0	1	0
R2_2_1	200	5	4372.17	295	5	4042.67	186.02	6	3890.34	146.13	1	−11.02	0	0	0	1	−3.77	0	0	0
RC2_2_1	200	4	2662.75	216	4	2021.49	338.41	6	2165.45	146.57	2	−18.68	0	0	0	2	7.12	0	0	0
C1_4_1	400	42	8135.35	147	40	7533.03	220.13	40	7533.14	179.87	−2	−7.40	1	1	1	0	0.00	0	0	0
R1_4_1	400	42	12202.62	237	40	10671.7	537.22	40	10591.54	197.44	−2	−13.20	1	1	1	0	−0.75	1	1	0
RC1_4_1	400	40	10036.82	193	38	9772.56	401.1	39	9020.26	192.26	−1	−10.13	1	1	1	1	−7.70	0	0	0
C2_4_1	400	14	5085.08	279	12	4144.84	776.51	12	4144.84	635.93	−2	−18.49	1	1	1	0	0.00	0	1	0
R2_4_1	400	9	14119.64	735	9	8952.24	506.62	11	8745.02	568.83	2	−38.06	0	0	0	2	−2.31	0	0	0
RC2_4_1	400	13	7229.22	791	12	6621.94	681.44	12	6901.4	504.03	−1	−4.53	1	1	1	0	4.22	0	0	0
C1_6_1	600	69	19720.65	257	63	15594.21	765.79	62	15916.06	577.5	−7	−19.29	1	1	1	−1	2.06	1	1	1
R1_6_1	600	62	25729.28	581	59	22306.17	1211.12	59	22811.41	549.07	−3	−11.34	1	1	1	0	2.27	0	0	0
RC1_6_1	600	60	20535.26	733	57	19679.75	946.06	59	18338.05	573.54	−1	−10.70	1	1	1	2	−6.82	0	0	0
C2_6_1	600	20	9509.15	926	18	7830.16	1652.55	18	7830.16	839.5	−2	−17.66	1	1	1	0	0.00	0	1	0
R2_6_1	600	13	27294.11	2439	13	17459.41	1558.88	16	18870.81	1209.28	3	−30.86	0	0	0	3	8.08	0	0	0
RC2_6_1	600	20	22837.36	2860	16	12693.19	2607.91	17	13264.62	1050.03	−3	−41.92	1	1	1	1	4.50	0	0	0
C1_8_1	800	88	32801.92	1054	82	27035.71	1495.17	82	27344.05	2948.4	−6	−16.64	1	1	1	0	1.14	0	0	0
R1_8_1	800	93	51949.49	1869	80	39348.17	3082.84	80	39664.86	2671.93	−13	−23.65	1	1	1	0	0.80	0	0	0
RC1_8_1	800	88	32801.92	1620	75	38431.09	2234.79	78	32692.26	2932.16	−10	−0.33	1	1	1	3	−14.93	0	0	0
C2_8_1	800	27	14573.93	3636	24	11759.05	3022.47	24	11957.11	1341.19	−3	−17.96	1	1	1	0	1.68	0	0	0
R2_8_1	800	19	48611.6	7663	18	27270.04	3429.95	22	29294.91	929.74	3	−39.74	0	0	0	4	7.43	0	0	0
RC2_8_1	800	–	–	–	60	26652.1	2827.44	62	29640.17	1547.51	40	−24.72	–	–	–	2	11.21	0	0	0
C1_10_1	1000	110	52328.78	2418	102	44764.64	2253.4	104	45303.96	3131.22	−6	−13.42	1	1	1	2	1.20	0	0	0
R1_10_1	1000	115	77993.35	4539	100	58912.62	4651.74	101	58208.03	2529.47	−14	−25.37	1	1	1	1	−1.20	0	0	0
RC1_10_1	1000	102	66883.49	3483	93	63953.66	4325.54	96	51345.98	780.83	−6	−23.23	1	1	1	3	−19.71	0	0	0
C2_10_1	1000	33	23981.11	6529	30	17088.5	5009.45	32	17869.93	915.41	−1	−25.48	1	1	1	2	4.57	0	0	0
R2_10_1	1000	22	67441.51	21379	22	42117.48	5604.7	28	45533.11	934.84	6	−32.49	0	0	0	6	8.11	0	0	0
RC2_10_1	1000	–	–	–	75	40643.56	6460.91	78	46073.03	3093.92	–	–	–	–	–	3	13.36	0	0	0
Total													22	22	21			2	6	1

* $Gap1_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{p-SA}}$, $Gap1_{TD} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{GA}}}{TD_{\text{p-SA}}} * 100\%$.

** $Gap2_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{ALNS-PR}}$, $Gap2_{TD} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{ALNS-PR}}}{TD_{\text{ALN-PR}}} * 100\%$.

† "1" = True, "0" = False.

Solver and a proposed VNS, respectively. They reported that the VNS could get near-optimal solutions in a much shorter time than CPLEX Solver. Schneider et al. (2015) studied a newly vehicle routing problem with intermediate stops (VRPIS), and the stopping requirements at intermediate facilities are considered in the model. They proposed an adaptive variable neighborhood search (AVNS) to solve the VRPIS. The experimental results show that the proposed ALNS is quite efficient in solving the given instances. Hof et al. (2017) investigated a generalized VRPIS, and again an Adaptive Variable Neighborhood Search (AVNS) algorithm is employed to solve the battery swap station location-routing problem with capacitated electric vehicles. The experimental results reported by the authors show that AVNS is efficient for reducing battery swap stations. Recently, inspired by a home health care scenario, Frifita and Masmoudi (2020) extended the classical vehicle routing problem with time window by considering the constraints of temporal dependencies, multi-structures, and multi specialties problem (VRPTW-TD-2MS). The proposed VRPTW-TD-2MS is finally solved by the VNS, and experimental results demonstrate that VNS is efficient for solving the designed instances.

2.4. Tabu search

TS, regarded as a classical meta-heuristic, has been initially introduced by Glover (1986). TS is efficient for solving mathematical optimizations including linear programming, integer programming, non-linear programming (Youssef et al., 2001; Tan et al., 2001; Schneider and Krohling, 2014) and combinatorial optimization (Carrasco et al., 2015). In this subsection, the previous works related to vehicle routing problem by using TS are presented. Garcia et al. (1994) introduced a TS heuristic with exchanging the near arcs of the neighborhood to solve vehicle routing problem with time windows. Badeau et al. (1997) proposed a parallel TS for the vehicle routing problem with time windows. Taillard et al. (1997) proposed a TS for vehicle routing problem with soft time windows by adding a penalty value to the objective function value when the soft time window constraints are violated. Belhaiza et al. (2014) investigated a vehicle routing problem with multiple time windows and introduced a new hybrid variable neighborhood-TS heuristic and a minimum backward time slack algorithm applicable to solve it by recording the minimum waiting time and the minimum delay during the route generation and adjusts the arrival and departure times backward. Martínez-Puras and Pacheco (2016) investigated a TS based strategy for solving multi-objective period VRP, and the experimental results demonstrated that the proposed approach performed better than the classical NSGA-II (Non-dominated Sorting Genetic Algorithm). Silvestrin and Ritt (2017) proposed a TS embedded into an iterated local search to solve the multi-compartment vehicle routing problem and found that it consistently produces better solutions than existing heuristic algorithms. Lai et al. (2019a) proposed an intensification-driven TS which can solve the minimum differential dispersion problem. Shi et al. (2019) studied the home health care routing and scheduling problem with considering uncertainties. In their model, the robust optimization technique is employed to model the problem, and a simple tabu search is proposed to solve the model. According to Shi et al. (2019), the proposed tabu search showed a better performance than the simple simulated annealing and genetic algorithm.

The core characteristics of TS is the structure of the Tabu List (TL), which depends on the structure of the neighborhood of the solution. In Tabu search, neighborhood structure affects the convergence speed and the solution quality. Hence, in this study, different types of neighborhood structures are adopted, which includes $\lambda-$interchange generation mechanism, $k-$opt operation, relocate operation, insert operation, and swap operation. Based on the neighborhood structures, two types of TL are designed, which is the first time to be adopted in the TS algorithm to solve the problem considered in this study.

Two-stage algorithms have been applied to solve many combinatorial optimization problems, such as graph coloring, vehicle routing problems, bin packing problem (Rodriguez-Tello et al., 2008; Fu and Hao, 2015; Lai et al., 2018, 2019b; Wei and Hao, 2019), etc.

To sum up, we discover that, despite the critical applications of VRPSPDTW, there are not sufficient heuristic algorithms that have been developed for solving the classical benchmark instances of Wang and Chen (2012). Especially, two-stage based algorithms have never been performed to improve the solution quality. Additionally, even though TS has been popularly used for solving VRP and its variants, it still has the drawback of trapping into local optima. This work is the first attempt to integrate two different structures of tabu lists for further improving the searching ability of TS. What is more, we notice that the objective function in Wang and Chen (2012) and Wang et al. (2015) directly employed the objective function by joining the weighted TD and NV. However, we observe that the searching direction of reducing TD might not benefit from decreasing NV. Therefore, our work divides the optimization of NV as an independent procedure, and the objective function is a comparative feature of helping decreasing NV.

Our work contributes to the novelty of a learning-based two-stage procedure for solving VRPSPDTW in the following aspects.

(1) We are the first to consider designing a learning-based two-stage (VNS-BSTS) framework for solving the challenge instances of Wang and Chen (2012).

(2) The VNS is evaluated by a learning-based objective function rather than a commonly used single criterion, such as Wang and Chen (2012) and Wang et al. (2015).

(3) Considering that the diversity tabu lists can help to overcome local optima, we make the first effort to integrate multi-structure of TL for improving the solution quality of searching.

3. Mathematical formulation

To help readers understand the problem considered in this study, Fig. 1 provides an intuitive example. In this example, there are five customers and two vehicles. Vehicle 1 starts from the depot with full loaded cargoes and after servicing three customers it returns to the depot with full loaded cargoes. Vehicle 2 starts from the depot with un-full loaded cargoes and after servicing two customers it returns to the depot with un-full loaded cargoes.

Before introducing the mathematical formulation of the considered problem in this study, a list of assumptions are provided and the notations, which are used in the proposed formulation, are listed.

Problem assumptions:

- The vehicles considered in this study are homogeneous; i.e., each vehicle has an identical maximum speed, maximum operating time, loading capacity, and empty vehicle weight.
- The total customer demand on any vehicle's route cannot exceed the vehicle capacity.
- There is only one depot.
- All the vehicles start from the depot and return to the depot.
- Each customer has a non-negative demand that must be within the maximum vehicle capacity.
- Each customer can be visited only once by a single vehicle within a given time window.
- Each customer has delivery and pickup demand simultaneously.
- The service time of each customer depends on the amount of loading and unloading cargo to that customer.

The mathematical model is at Appendix B. We summarize two remarks about this model.

Remark 1. Let n be the number of customers, and V be the cardinality of vehicles. We can obtain that this model contains $(n+1)|V|+2n+3|V|$ variables and $2n^2+5n|V|+2n+6|V|$ constraints (Wang and Chen, 2012).

Table 8
Comparison between the solutions obtained by the VNS-BSTS and BSTS for all the instances.

Instance	BSTS		VNS-BSTS			Gap3 (VNS-BSTS VS p-SA)				
	NV	TD	NV	TD	CT	NV	TD(%)	Is solution improved?	Is solution non-worse?	Is NV improved?
Rdp101	19	1652.71	19	1650.8	34.91	0	−0.12	1	1	0
Rdp102	18	1474.86	17	1486.12	31.6	−1	0.76	1	1	1
Rdp103	14	1213.62	13	1294.75	38.26	−1	6.68	1	1	1
Rdp104	10	990.97	10	984.81	65.48	0	−0.62	1	1	0
Rdp105	15	1364.18	14	1377.11	33.95	−1	0.95	1	1	1
Rdp106	12	1262.81	12	1261.4	43.65	0	−0.11	1	1	0
Rdp107	11	1083.4	10	1144.02	44.13	−1	5.60	1	1	1
Rdp108	10	957.57	9	968.32	67.67	−1	1.12	1	1	1
Rdp109	12	1158.58	12	1160.08	40.18	0	0.13	0	0	0
Rdp110	11	1108.57	11	1101.33	50.91	0	−0.65	1	1	0
Rdp111	11	1073.03	10	1117.76	46.72	−1	4.17	1	1	1
Rdp112	10	971.23	10	961.29	65.87	0	−1.02	1	1	0
Cdp101	11	980.63	11	976.04	60.29	0	−0.47	1	1	0
Cdp102	10	952.69	10	942.45	68.87	0	−1.07	1	1	0
Cdp103	11	887.78	10	896.28	95.4	−1	0.96	1	1	1
Cdp104	10	878.59	10	872.39	70.62	0	−0.71	1	1	0
Cdp105	11	983.1	10	1080.63	41.46	−1	9.92	1	1	1
Cdp106	11	878.29	10	963.45	57.3	−1	9.70	1	1	1
Cdp107	11	909.64	10	987.64	79.04	−1	8.57	1	1	1
Cdp108	10	942.74	10	934.41	79.24	0	−0.88	1	1	0
Cdp109	10	915.64	10	909.27	68.59	0	−0.70	1	1	0
RCdp101	15	1640.27	14	1708.21	32.04	−1	4.14	1	1	1
RCdp102	14	1484.11	13	1526.36	43.85	−1	2.85	1	1	1
RCdp103	12	1322.41	11	1336.05	46	−1	1.03	1	1	1
RCdp104	11	1181.06	10	1177.21	43.81	−1	−0.33	1	1	1
RCdp105	14	1590.13	14	1548.38	42.41	0	−2.63	1	1	0
RCdp106	13	1398.47	12	1408.19	47.52	−1	0.70	1	1	1
RCdp107	12	1278.19	11	1295.43	54.7	−1	1.35	1	1	1
RCdp108	11	1155.46	10	1207.6	51.23	−1	4.51	1	1	1
Rdp201	5	1189.51	4	1254.57	46.59	−1	5.47	1	1	1
Rdp202	4	1084.69	3	1202.27	119.56	−1	10.84	1	1	1
Rdp203	4	898.95	3	949.42	107.1	−1	5.61	1	1	1
Rdp204	3	755.2	2	837.13	129.22	−1	10.85	1	1	1
Rdp205	4	982.02	3	1027.49	79.63	−1	4.63	1	1	1
Rdp206	3	915.46	3	938.63	96.08	0	2.53	0	0	0
Rdp207	3	820.11	2	912.26	101.46	−1	11.24	1	1	1
Rdp208	3	712.9	2	737.26	173.85	−1	3.42	1	1	1
Rdp209	3	925.63	3	940.29	77	0	1.58	0	0	0
Rdp210	4	931.03	3	945.97	87.14	−1	1.60	1	1	1
Rdp211	3	771.25	3	805.22	82.25	0	4.40	0	0	0
Cdp201	3	591.56	3	591.56	53.42	0	0.00	0	1	0
Cdp202	3	591.56	3	591.56	70.56	0	0.00	0	1	0
Cdp203	3	591.17	3	591.17	56.26	0	0.00	0	1	0
Cdp204	3	593.93	3	599.33	50.34	0	0.91	0	0	0
Cdp205	3	588.88	3	588.88	50.98	0	0.00	0	1	0
Cdp206	3	588.49	3	588.49	49.15	0	0.00	0	1	0
Cdp207	3	588.29	3	588.29	50.58	0	0.00	0	1	0
Cdp208	3	588.32	3	588.32	49.61	0	0.00	0	1	0
RCdp201	5	1314.18	4	1437.48	32	−1	9.38	1	1	1
RCdp202	5	1122.1	3	1412.52	52	−2	25.88	1	1	1
RCdp203	3	1089.79	3	1064.95	74.56	0	−2.28	1	1	0
RCdp204	3	818.97	3	813.74	85.46	0	−0.64	1	1	0
RCdp205	5	1230.46	4	1316.06	33.28	−1	6.96	1	1	1
RCdp206	4	1082.28	3	1154.36	52.15	−1	6.66	1	1	1
RCdp207	4	1020.85	3	1117.04	60.77	−1	9.42	1	1	1
RCdp208	3	850.56	3	843.3	73.71	0	−0.85	1	1	0
Total better								44	51	30

* $\text{Gap}_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{VNS-TSI}}$, $\text{Gap}_{\text{TD}} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{VNS-TSI}}}{TD_{\text{VNS-TSI}}} * 100\%$.
† "1" = True, "0" = False.

Remark 2. VRPSPDTW can be viewed as a combination of VRPTW and pickup and delivery problem. As been proved by the previous studies, VRPTW is NP-hard problem (Solomon and Desrosiers, 1988). Hence, the VRPSPDTW considered in this study is an NP-hard problem.

In order to make the results of our article comparable to the results published by Wang and Chen (2012) and Wang et al. (2015), we define the following relationship.

Definition 1. Let s_1 and s_2 be two solutions. Meanwhile, let NV_{s_i}(i = 1 or 2) be the number of vehicles of s_i, and TD_{s_i} be the total distance of s_i. Now the lexicographic-based objective can express the relationship of the solutions as follows.

(1) If $NV_{s_1} < NV_{s_2}$, then s_1 dominates s_2 regardless of the values of total distances.

(2) If NV_{s_1} equals to NV_{s_2}, and $TD_{s_1} < TD_{s_2}$, then s_1 dominates s_2.

(3) If NV_{s_1} equals to NV_{s_2}, and TD_{s_1} equals to TD_{s_2}, then s_1 equals to s_2.

4. A two-stage algorithm

Due to the hardness of the problem considered in this study itself, exact algorithms including the branch and bound, branch and price

Table 9
Comparison of the solutions obtained from the VNS-TS-I and VNS-BSTS for all the instances.

Instance	VNS-TSI		VNS-BSTS			Gap3 (VNS-TSI VS VNSBSTS)				
	NV	TD	NV	TD		NV	TD(%)	Is solution improved?	Is solution non-worse?	Is NV improved?
Rdp101	19	1657.73	19	1650.8	34.91	0	−0.42	1	1	0
Rdp102	17	1491.8	17	1486.12	31.6	0	−0.38	1	1	0
Rdp103	14	1223.18	13	1294.75	38.26	−1	5.85	1	1	1
Rdp104	10	1007.63	10	984.81	65.48	0	−2.26	1	1	0
Rdp105	14	1403.69	14	1377.11	33.95	0	−1.89	1	1	0
Rdp106	12	1272.07	12	1261.4	43.65	0	−0.84	1	1	0
Rdp107	11	1106.08	10	1144.02	44.13	−1	3.43	1	1	1
Rdp108	10	969.26	9	968.32	67.67	−1	−0.10	1	1	1
Rdp109	11	1215.83	11	1224.86	40.18	0	0.74	0	0	0
Rdp110	11	1114.46	11	1101.33	50.91	0	−1.18	1	1	0
Rdp111	11	1080.51	10	1117.76	46.72	−1	3.45	1	1	1
Rdp112	10	988.73	10	961.29	65.87	0	−2.78	1	1	0
Cdp101	11	992.27	11	976.04	60.29	0	−1.64	1	1	0
Cdp102	10	960.86	10	942.45	68.87	0	−1.92	1	1	0
Cdp103	10	921.11	10	896.28	95.4	0	−2.70	1	1	0
Cdp104	10	878.19	10	872.39	70.62	0	−0.66	1	1	0
Cdp105	10	1109.83	10	1080.63	41.46	0	−2.63	1	1	0
Cdp106	10	977.66	10	963.45	57.3	0	−1.45	1	1	0
Cdp107	11	914.67	10	987.64	79.04	−1	7.98	1	1	1
Cdp108	10	956.53	10	934.41	79.24	0	−2.31	1	1	0
Cdp109	10	960.86	10	909.27	68.59	0	−5.37	1	1	0
RCdp101	15	1652.11	14	1708.21	32.04	−1	3.40	1	1	1
RCdp102	13	1520.68	13	1526.36	43.85	0	0.37	0	0	0
RCdp103	11	1310.11	11	1336.05	46	0	1.98	0	0	0
RCdp104	10	1242.34	10	1177.21	43.81	0	−5.24	1	1	0
RCdp105	14	1560.85	14	1548.38	42.41	0	−0.80	1	1	0
RCdp106	12	1420.79	12	1408.19	47.52	0	−0.89	1	1	0
RCdp107	11	1285.23	11	1295.43	54.7	0	0.79	0	0	0
RCdp108	11	1167.6	10	1207.6	51.23	−1	3.43	1	1	1
Rdp201	4	1303.47	4	1254.57	46.59	0	−3.75	1	1	0
Rdp202	3	1235.34	3	1202.27	119.56	0	−2.68	1	1	0
Rdp203	3	977.55	3	949.42	107.1	0	−2.88	1	1	0
Rdp204	3	776.1	2	837.13	129.22	−1	7.86	1	1	1
Rdp205	3	1064.67	3	1027.49	79.63	0	−3.49	1	1	0
Rdp206	3	951.98	3	938.63	96.08	0	−1.40	1	1	0
Rdp207	3	842.81	2	912.26	101.46	−1	8.24	1	1	1
Rdp208	2	746.03	2	737.26	173.85	0	−1.18	1	1	0
Rdp209	3	945.72	3	940.29	77	0	−0.57	1	1	0
Rdp210	3	974.55	3	945.97	87.14	0	−2.93	1	1	0
Rdp211	3	817.18	3	805.22	82.25	0	−1.46	1	1	0
Cdp201	3	591.56	3	591.56	53.42	0	0.00	0	1	0
Cdp202	3	591.56	3	591.56	70.56	0	0.00	0	1	0
Cdp203	3	591.17	3	591.17	56.26	0	0.00	0	1	0
Cdp204	3	599.63	3	599.33	50.34	0	−0.05	1	1	0
Cdp205	3	588.88	3	588.88	50.98	0	0.00	0	1	0
Cdp206	3	588.49	3	588.49	49.15	0	0.00	0	1	0
Cdp207	3	588.29	3	588.29	50.58	0	0.00	0	1	0
Cdp208	3	588.32	3	588.32	49.61	0	0.00	0	1	0
RCdp201	4	1470.91	4	1437.48	32	0	−2.27	1	1	0
RCdp202	4	1190.86	3	1412.52	52	−1	18.61	1	1	1
RCdp203	3	1111.63	3	1064.95	74.56	0	−4.20	1	1	0
RCdp204	3	820.76	3	813.74	85.46	0	−0.86	1	1	0
RCdp205	4	1386.51	4	1316.06	33.28	0	−5.08	1	1	0
RCdp206	3	1201	3	1154.36	52.15	0	−3.88	1	1	0
RCdp207	3	1098.31	3	1098.64	60.77	0	0.03	0	0	0
RCdp208	3	871.23	3	843.3	73.71	0	−3.21	1	1	0
Total better								44	51	10

[*] $\text{Gap}_{NV} = \text{NV}_{\text{VNS-BSTS}} - NV_{\text{VNS-TSI}}$, $\text{Gap}_{TD} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{VNS-TSI}}}{TD_{\text{VNS-TSI}}} * 100\%$.

[†] "1" = True, "0" = False.

(Subramanian et al., 2013) and cutting planes cannot obtain the optimal solution in an acceptable time. In practical situations, the decision makers need to make decisions as soon as possible when they receive the order information from the customers. Hence, heuristic algorithms need to be designed to solve VRPSPDTW. This section mainly presents the detailed procedures of the two-stage heuristic algorithm, which is also named as VNS-BSTS for convenience. First, the main scheme of the VNS-BSTS is presented, then VNS and BSTS are introduced, respectively.

4.1. Main scheme of the two-stage algorithm

Variable neighborhood search (VNS) (de Armas et al., 2015), initially proposed by Mladenović and Hansen (1997), is very easy to implement and can be combined with any local search algorithm as a subroutine easily. TS is a meta-heuristic search method employing local search methods with the particular memory mechanism for mathematical optimization. It avoids the roundabout search through the local neighborhood search mechanism and the corresponding taboo criteria,

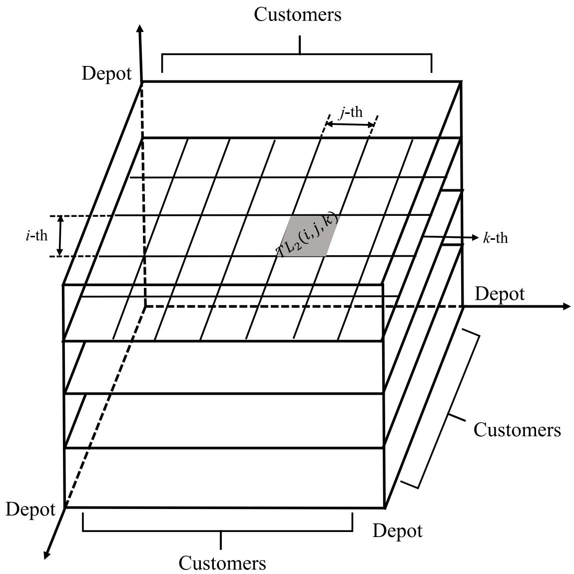

Fig. 6. The structure of tabu list of TL_2.

as well as aspiring some of the taboos' excellent states through the aspiration strategy. These strategies can improve the effective exploration of diversity, and finally achieving global optimization.

Inspired by the previous study which shows that Tabu search algorithm is very effective with large scale and real-time vehicle dispatching problems (Lou and Shi, 2005), in this study, we introduce an efficient learning-based two-stage algorithm, composed by VNS and TS, for solving the VRPSPDTW. The details of the proposed heuristics show as follows.

As mentioned that NV is the primary objective function, however, we find that TS has an excellent ability for decreasing TD but weakness at optimizing NV. Focusing on the minimization of TD, sometimes make it extremely difficult to find solutions with fewer routes, as this may require considerable degradation of the NV component of the objective function.

Therefore, in this study, we divide the solving processing into two stages. In the first stage, a VNS is proposed to make an effort to minimize the number of vehicles. In the second stage, a bi-structure based TS is designed to minimize the TD with the given number of assigned vehicles which was obtained in the first stage.

The overview of the proposed two-stage algorithm is shown in Fig. 2. A logistics company usually has limited resources including capacity of warehouse and number of vehicles, which is the input data shown in the left of Fig. 2(a). After the logistics received the service requirement from the customers, the decision makers of the logistics need to make decisions including the number of assigned vehicles and the routes for each assigned vehicle. Then the decision maker can execute the two-stage algorithm to obtain a solution. First, the two-stage algorithm uses the push forward insertion heuristic (PFIH) (Solomon, 1987) to generate an initial solution. Next go to the first stage and apply the proposed VNS to determine the number of assigned vehicles. Then go to the second stage and execute the proposed TS-I and TS-II. After finishing the second stage, check whether the maximum iteration is satisfied or not. If the maximum iteration is not satisfied, go to the first stage until it is satisfied. Otherwise, it returns the final best solution.

The detail of each component of the two-stage algorithm is described in the remaining part of this section.

4.1.1. Solution initialization

Solution initialization plays a significant role in heuristic algorithm. In this study, we adopt the Push-Forward Insertion Heuristic (PFIH), which is a very efficient method to generate the initial solution for VRPTW and was proposed by Solomon (1987). PFIH is a popular and efficient method to generate a initial solution and it has been used in different applications, such as home health care services (Shi et al., 2019) Due to the PFIH without considering the simultaneous pickup and delivery constraints, the insertion-based procedure proposed by Chen and Wu (2006) is integrated into the PFIH in this study.

4.2. The first stage

The foremost idea of the VNS is to employ multiple diverse neighborhoods for the process of seeking better solutions (Davari et al., 2013; Pacheco et al., 2018). First, the smallest neighborhood is employed to gain a better solution. When this solution cannot be improved, switch to a slightly larger neighborhood. If this neighborhood can continue to enhance the solution, return to the smallest neighborhood, otherwise continue to switch to a larger neighborhood. VNS has the characteristic of easy coding, flexible structure, and effective performance. VNS adopted in this study is motivated by its success in solving such problems as coloring graphs (Johnson et al., 1991), where the goal is to minimize the size of a game (for example, the number of colors), and the overall simplicity of its implementation.

4.2.1. Learning-based evaluation function

The evaluation function is another fundamental aspect of the VNS. As mentioned earlier, the objective function (1) is not always appropriate for minimizing the NV, as it may lead to a search for solutions with a lower TD and the inability to reduce routes. This reveals that minimizing the NV may not be on the same direction of reducing the TD. Hence, to overcome these limitations, we need to explore the properties of the solution including the number of routes, the number of customers in a route and violated constraints (including capacity constraints and time-window constraints) if we consider to insert the customers in smallest route into other non-empty routes. These properties have the potential to reduce the NV. Inspired by this, we designed a more sophisticated dictionary ordering which orders the properties to minimize the number of routes.

$$\min \quad \sigma \cdot \sum_{k\in V}\sum_{j\in C} x_{0jk} \tag{1}$$

Then, we designed a learning-based lexicographic evaluation function $e(\cdot)$ (see Eq. (2)), which is one of the most significant contributions of this subsection. Generally, $e(\cdot)$ minimizes the number of routes (primary criterion in (2)), minimizes the minimum value on a used route (secondary criterion in (2)), maximizes the squared sum of the squares of the route sizes (third criterion in (2)), and minimizes violated constraints if we consider inserting the customers in smallest route into other non-empty routes (fourth criterion in (2)). The second criterion was also used successfully in Wang et al. (2015), and the third criterion was partly adopted in Bent and Van Hentenryck (2004).

Before describing the lexicographic evaluation function $e(\cdot)$, the notations used in the lexicographic evaluation function are listed as follows to help reader easily understand the $e(\cdot)$.

- s: a given feasible solution.
- $|s|$: the total number of routes in the solution s.
- $r_i(s) = \{v_{i,1}, v_{i,2}, \dots, v_{i,end}\}$: the scheduling plan in a solution s, depot $\rightarrow v_{i,1} \rightarrow v_{i,2} \rightarrow \dots \rightarrow v_{i,end} \rightarrow$ depot.
- $|r_i(s)|$: the number of customers on route $r_i(s)$.
- $min\_r(s)$: a route set with the minimum customers in all non-empty routes of s.
- $\min_{1\le i\le |s|} |r_i(s)|$: the minimum number of customers for arbitrary routes in the solution s.
- $mvc(s)$: the minimized violated constraint if we consider to insert the customers in smallest route into other non-empty routes. (see Eq. (3))

Instance naming rules

Customer location type + dp + time window type + instance index

Example 1: Rdp201

R: Customer locations generated uniformly randomly over a square
2: Large time windows and large vehicle capacity
01: Instance index is 1

Example 2: Cdp103

C: Clustered customers whose time windows were generated based on a known solution
1: Narrow time windows and small vehicle capacity
03: Instance index is 3

Example 3: RCdp204

RC: A combination of randomly placed and clustered customers
2: Large time windows and large vehicle capacity
04: Instance index is 4

Fig. C.7. Instance naming rules.

- $mdl(i, j, min\_r(s), s)$: the penalty value generated while $i \in min\_r(s)$ is inserted to the position j of the solution s. (see Eq. (4))
- S'_{jk}: the new arrival time of customer j visited by vehicle k, if $i \in min\_r(s)$ is inserted to the position j of the solution s.
- $load'_k$: the total load of vehicle k, if $i \in min\_r(s)$ is inserted to the position j of the solution s.

As shown in Eq. (2), we evaluate the solutions s by considering four sub-indicators: $|s|$, $\min_{1\le i\le|s|}|r_i(s)|$, $-\sqrt{\sum_{1\le i\le|s|}|r_i(s)|^2}$ and $mvc(s)$. The importance of the features are in descending order.

$$e(s) = \left(|s|, \min_{1\le i\le|s|}|r_i(s)|, -\sqrt{\sum_{1\le i\le|s|}|r_i(s)|^2}, mvc(s)\right), \tag{2}$$

where:

$$mvc(s) = \sum_{i\in min\_r(s)} mdl(i, min\_r(s), s) \tag{3}$$

and

$$mdl(i, min\_r(s), s) = \begin{cases} 0 & \text{If node } i \text{ can be inserted to another route by satisfying all the constraints;} \\ +\infty & \text{If all the insertions of } i \text{ violate the capacity constraint;} \\ \min_{j\in\mathfrak{C}\setminus min\_r(s)} mdl(i, j, min\_r(s), s) & \text{Otherwise.} \end{cases} \tag{4}$$

in which,

$$mdl(i, j, r, s) = \sum_{h\in s}\sum_{k\in K}\max(S'_{jk} - b_j, 0) + \sum_{k\in K} max(load'_k - Q, 0) \tag{5}$$

This indicates that the algorithm prefers the solutions with many customer routes and routes with fewer customers than solutions where customers are more evenly distributed across the route. The intuition is that the guiding algorithm removes customers from some small routes and adds them to a larger route. This type of component is applied in many algorithms, a typical example being graph coloring (Johnson et al., 1991). The third component minimizes the minimum delay Routing plan. This concept is made up by Homberger and Gehring (1999) in the context of evolutionary algorithms. It facilitates the solution where customers on the smallest route can be repositioned on other routes without violating constraints or time window violations as small as possible. Minimizing the minimum latency is therefore beneficial to the solution, and in solutions that are difficult to reposition, customers can relocate more easily. More precisely, the minimum delay is defined as Eq. (3).

As one may notice that the lexicographic evaluation function (2) is quite complicated, and is not formulated by the commonly used objective function, like Wang and Chen (2012) and Wang et al. (2015). Now, first we present each sub-indicators in Eqs. (6)–(9), then explain how objective function (2) works by introducing if-then rules in Table 1.

$$e_1(s) = |s| \tag{6}$$

$$e_2(s) = \min_{1\le i\le|s|}|r_i(s)|, \tag{7}$$

$$e_3(s) = -\sqrt{\sum_{1\le i\le|s|}|r_i(s)|^2}, \tag{8}$$

$$e_4(s) = mvc(s). \tag{9}$$

4.2.2. Neighborhood structure

To explore all the possible neighborhoods of a given solution, in this study, different types of neighborhood generation methods are adopted, and the detail description of neighborhood generation methods are presented as follows.

The neighborhood search operators include the relocate, insert, swap, 2-opt operators. These operators can be applied inter-route or intra-route, which is defined as to move operations in this study. Relocate operation: relocate a customer from the current position to another position in the current route or different route. k−opt operation: k−opt

Table D.10
Experimental results compare with Wang and Chen (2012) for the instances with category Rdp1XX, Cdp1XX and RCdp1XX.

Instance ID	**Basic GA** (Wang and Chen, 2012)		CoGA (Wang and Chen, 2012)		VNS-BSTS			Gap 1(VNS-BSTS VS Basic GA)**					Gap 2(VNS-BSTS VSBasic CoGA)**				
	NV	**TD**	NV	TD	NV	TD	CT	NV	TD (%)	Is solution improved?	is solution non-worse?	is NV improved?	NV	TD (%)	Is solution improved?	Is solution non-worse?	Is NV improved?
Rdp101	19	1656.68	19	1653.53	19	1650.8	34.91	0	−0.35	1	1	0	0	−0.17	1	1	0
Rdp102	18	1474.93	17	1488.04	17	1486.12	31.6	−1	0.76	1	1	1	0	−0.13	1	1	0
Rdp103	14	1217.6	14	1216.16	13	1294.75	38.26	−1	6.34	1	1	1	−1	6.46	1	1	1
Rdp104	10	1029.38	10	1015.41	10	984.81	65.48	0	−4.33	1	1	0	0	−3.01	1	1	0
Rdp105	15	1391.29	15	1375.31	14	1377.11	33.95	−1	−1.02	1	1	1	−1	0.13	1	1	1
Rdp106	13	1258.82	13	1255.48	12	1261.4	43.65	−1	0.20	1	1	1	−1	0.47	1	1	1
Rdp107	11	1091.89	11	1087.95	10	1144.02	44.13	−1	4.77	1	1	1	−1	5.15	1	1	1
Rdp108	10	988.91	10	967.49	9	968.32	67.67	−1	−2.08	1	1	1	−1	0.09	1	1	1
Rdp109	12	1248.35	12	1160.00	11	1224.86	40.18	−1	−1.88	1	1	1	−1	5.59	1	1	1
Rdp110	12	1157.78	12	1116.99	11	1101.33	50.91	−1	−4.88	1	1	1	−1	−1.40	1	1	1
Rdp111	11	1084.27	11	1065.27	10	1117.76	46.72	−1	3.09	1	1	1	−1	4.93	1	1	1
Rdp112	10	998.08	10	974.03	10	961.29	65.87	0	−3.69	1	1	0	0	−1.31	1	1	0
Cdp101	11	1001.97	11	1001.97	11	976.04	60.29	0	−2.59	1	1	0	0	−2.59	1	1	0
Cdp102	10	1030.68	10	961.38	10	942.45	68.87	0	−8.56	1	1	0	0	−1.97	1	1	0
Cdp103	10	905.71	10	897.65	10	896.28	95.4	0	−1.04	1	1	0	0	−0.15	1	1	0
Cdp104	10	889.3	10	878.93	10	872.39	70.62	0	−1.90	1	1	0	0	−0.74	1	1	0
Cdp105	11	983.1	11	983.10	10	1080.63	41.46	−1	9.92	1	1	1	−1	9.92	1	1	1
Cdp106	11	878.29	11	878.29	10	963.45	57.3	−1	9.70	1	1	1	−1	9.70	1	1	1
Cdp107	11	913.81	11	913.81	10	987.64	79.04	−1	8.08	1	1	1	−1	8.08	1	1	1
Cdp108	11	922.59	10	951.24	10	934.41	79.24	−1	1.28	1	1	1	0	−1.77	1	1	0
Cdp109	10	938.1	10	940.49	10	909.27	68.59	0	−3.07	1	1	0	0	−3.32	1	1	0
RCdp101	15	1665.2	15	1652.90	14	1708.21	32.04	−1	2.58	1	1	1	−1	3.35	1	1	1
RCdp102	14	1516.34	14	1497.05	13	1526.36	43.85	−1	0.66	1	1	1	−1	1.96	1	1	1
RCdp103	13	1370.7	12	1338.76	11	1336.05	46	−2	−2.53	1	1	1	−1	−0.20	1	1	1
RCdp104	11	1198.99	11	1188.49	10	1177.21	43.81	−1	−1.82	1	1	1	−1	−0.95	1	1	1
RCdp105	15	1561.01	14	1581.26	14	1548.38	42.41	−1	−0.81	1	1	1	0	−2.08	1	1	0
RCdp106	14	1453.34	13	1422.87	12	1408.19	47.52	−2	−3.11	1	1	1	−1	−1.03	1	1	1
RCdp107	12	1301.58	12	1282.10	11	1295.43	54.7	−1	−0.47	1	1	1	−1	1.04	1	1	1
RCdp108	11	1195.39	11	1175.04	10	1207.6	51.23	−1	1.02	1	1	1	−1	2.77	1	1	1
Total										29	29	21			29	29	18

* $\text{Gap1}_{NV} = \text{NV}_{\text{VNS-BSTS}} - \text{NV}_{\text{GA}}$, $\text{Gap1}_{\text{TD}} = \frac{\text{T}D_{\text{VNS-BSTS}} - \text{T}D_{\text{GA}}}{\text{T}D_{\text{GA}}} * 100\%$.

** $\text{Gap2}_{NV} = \text{NV}_{\text{VNS-BSTS}} - \text{NV}_{\text{CoGA}}$, $\text{Gap2}_{\text{TD}} = \frac{\text{T}D_{\text{VNS-BSTS}} - \text{T}D_{\text{CoGA}}}{\text{T}D_{\text{CoGA}}} * 100\%$.

† “1” = True, “0” = False.

Table D.11
Experimental results compare with Wang and Chen (2012) for the instances with category Rdp2XX, Cdp2XX and RCdp2XX.

Instance ID	**Basic GA** (Wang and Chen, 2012)		CoGA (Wang and Chen, 2012)		VNS-BSTS			Gap 1(VNS-BSTS VS Basic GA)*					Gap 2(VNS-BSTS VSBasic CoGA)**				
	NV	**TD**	NV	TD	NV	TD	CT(s)	NV	TD (%)	Is solution improved?	is solution non-worse?	is NV improved?	NV	TD (%)	Is solution improved?	Is solution non-worse?	Is NV improved?
Rdp201	4	1299.17	4	1280.44	4	1254.57	46.59	0	−3.43	1	1	0	0	−2.02	1	1	0
Rdp202	4	1121.88	4	1100.92	3	1202.27	119.56	−1	7.17	1	1	1	−1	9.21	1	1	1
Rdp203	3	1031.02	3	950.79	3	949.42	107.1	0	−7.91	1	1	0	0	−0.14	1	1	0
Rdp204	3	774.75	3	775.23	2	837.13	129.22	−1	8.05	1	1	1	−1	7.98	1	1	1
Rdp205	4	998.8	3	1064.43	3	1027.49	79.63	−1	2.87	1	1	1	0	−3.47	1	1	0
Rdp206	3	992.14	3	961.32	3	938.63	96.08	0	−5.39	1	1	0	0	−2.36	1	1	0
Rdp207	3	868.59	3	835.01	2	912.26	101.46	−1	5.03	1	1	1	−1	9.25	1	1	1
Rdp208	3	732	3	718.51	2	737.26	173.85	−1	0.72	1	1	1	−1	2.61	1	1	1
Rdp209	4	894.45	3	930.26	3	940.29	77	−1	5.12	1	1	1	0	1.08	0	0	0
Rdp210	3	1037.2	3	983.75	3	945.97	87.14	0	−8.80	1	1	0	0	−3.84	1	1	0
Rdp211	3	827.17	3	839.61	3	805.22	82.25	0	−2.65	1	1	0	0	−4.10	1	1	0
Cdp201	3	591.56	3	591.56	3	591.56	53.42	0	0.00	0	1	0	0	0.00	0	1	0
Cdp202	3	591.56	3	591.56	3	591.56	70.56	0	0.00	0	1	0	0	0.00	0	1	0
Cdp203	3	591.17	3	591.17	3	591.17	56.26	0	0.00	0	1	0	0	0.00	0	1	0
Cdp204	3	590.6	3	590.60	3	599.33	50.34	0	1.48	0	0	0	0	1.48	0	0	0
Cdp205	3	588.88	3	588.88	3	588.88	50.98	0	0.00	0	1	0	0	0.00	0	1	0
Cdp206	3	588.49	3	588.49	3	588.49	49.15	0	0.00	0	1	0	0	0.00	0	1	0
Cdp207	3	588.29	3	588.29	3	588.29	50.58	0	0.00	0	1	0	0	0.00	0	1	0
Cdp208	3	588.32	3	588.32	3	588.32	49.61	0	0.00	0	1	0	0	0.00	0	1	0
RCdp201	5	1376.98	4	1587.92	4	1437.48	32	−1	4.39	1	1	1	0	−9.47	1	1	0
RCdp202	4	1210.75	4	1211.12	3	1412.52	52	−1	16.66	1	1	1	−1	16.63	1	1	1
RCdp203	4	971.11	4	964.65	3	1064.95	74.56	−1	9.66	1	1	1	−1	10.40	1	1	1
RCdp204	3	857.23	3	822.02	3	813.74	85.46	0	−5.07	1	1	0	0	−1.01	1	1	0
RCdp205	5	1302.51	4	1410.18	4	1316.06	33.28	−1	1.04	1	1	1	0	−6.67	1	1	0
RCdp206	4	1167.52	3	1176.85	3	1154.36	52.15	−1	−1.13	1	1	1	0	−1.91	1	1	0
RCdp207	4	1058.86	4	1036.59	3	1098.64	60.77	−1	3.76	1	1	1	−1	5.99	1	1	1
RCdp208	3	908.08	3	878.57	3	843.3	73.71	0	−7.13	1	1	0	0	−4.01	1	1	0
Total										19	26	12			18	25	7

* $\mathrm{Gap1}_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{GA}}$, $\mathrm{Gap1}_{\text{TD}} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{GA}}}{TD_{\text{GA}}} * 100\%$.

** $\mathrm{Gap2}_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{CoGA}}$, $\mathrm{Gap2}_{\text{TD}} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{CoGA}}}{TD_{\text{CoGA}}} * 100\%$.

† "1" = True, "0" = False.

is a local search algorithm which is widely used in traveling salesman problem and vehicle routing problem. In this study, k set equal to 2. Swap operation: swap the position of two customers in the same route or different routes. Fig. 3 shows an example of the neighborhood of a given solution by using relocate operation, 2−opt operation, and swap operation. In this example, there are nine customers and two vehicles.

4.2.3. The pseudo code of VNS

The detailed procedures of VNS are given in Algorithm 1. The VNS also starts from an initial solution s_0. In line 2, we initialize the best solution as s_0, and set the parameters for the algorithm. Lines 3–17 give detailed procedures for improving the best solution s^b. The procedure in line 6 illustrates the switching of neighborhood structure. Line 7 presents the strategy of generating new solutions. VNS terminates when it reaches the stopping criterion: the best solution has non-improvement for specific iterations. Finally, the algorithm returns the best solution.

Algorithm 1: The pseudo code of VNS

1 **Input**: initial solution s_0 and neighborhood $N_k, k = 1, 2, \ldots, k_{max}$
2 $s^b \leftarrow s_0$; ⊳ s^b denote the current best solution
3 **while** stopping conditions are not met **do**
4 $\quad k \leftarrow 1$;
5 $\quad$ **while** $k \le k_{\max}$ **do**
6 $\quad\quad s' \leftarrow Shaking(N_k(s^b))$; ⊳ Randomly select a solution from k^{th} neighborhood of s^b
7 $\quad\quad s'' \leftarrow LocalSearch(s')$; ⊳ s'' is a local optima obtained by local search
8 $\quad\quad$ ⊳ Use the if-then rules in Table 1
9 $\quad\quad$ **if** $e(s'') < e(s^b)$ **then**
10 $\quad\quad\quad$ ⊳ $e(\cdot)$ is computed by Eq. (2)
11 $\quad\quad\quad s^b \leftarrow s''$; ⊳ update the current best solution
12 $\quad\quad\quad k \leftarrow 1$; ⊳ Reset neighborhood size
13 $\quad\quad$ **else**
14 $\quad\quad\quad k \leftarrow k + 1$; ⊳ Enlarge the neighborhood size
15 $\quad\quad$ **end**
16 $\quad$ **end**
17 **end**
18 **Return**: s^b.

4.3. The second stage

TS begins from an initialized solution and moves to a new solution from the neighborhoods of the current solution. The move operation is added to the tabu list (TL), and the current solution moves iteratively until some stopping criterion has been satisfied. The move operations recorded in the TL cannot be revisited until it reaches the expiration rules. Hence, in TS, the key characteristic is the TL, which is designed based on the neighborhood structure. The objective function of TS shows in Eq. (10).

$$\min \sum_{k\in V} \sum_{i\in N} \sum_{j\in N} c_{ij} x_{ijk} \tag{10}$$

4.3.1. Neighborhood structures

λ−interchange generation mechanism which was introduced by Osman (1993), is a method to generate neighborhood of a given solution S. A λ−interchange between a pair of route (r_p, r_q) is an exchange of a subset of visited customers from the route r_p with a subset of visited customers for the route r_q and the number of exchanged customers should be less than λ. Let $v_p \subseteq V_p$ denotes a subset, where V_p is the set of all the visited customers in route r_p and $|v_p| \le \lambda$. Let $v_q \subseteq V_q$ denotes a subset, where V_q is the set of all the visited customers in route r_q and $|v_q| \le \lambda$. The new generated routes can be denoted $r_p^* = (V_p - v_p) \cup v_q$ and $r_q^* = (V_q - v_q) \cup v_p$ by using the λ−interchange generation mechanism, respectively.

In this study, let couples (λ_1, λ_2) indicate the λ in route 1 and 2 with $\lambda = 2$ and $\lambda_1 \le \lambda, \lambda_2 \le \lambda$, respectively. Then all the possible combination of (λ_1, λ_2) are listed: (0,1), (0,2), (1,1), (1,2), (2,1), (2,2). Fig. 4 shows an example of λ−interchange generation mechanism with $\lambda = 2$ by using the solution in Fig. 3(a) as an initial solution.

4.3.2. Route refinement

When executing neighborhood search for a given solution $s^{current}$, the following rules are adopted. (1) All the above-defined neighborhood structures are adopted for $s^{current}$. (2) All the possible combinations of each type of the above-defined neighborhood structures are explored. (3) Only feasible neighborhood solutions are considered. To further minimize the TD after finishing the neighborhood search, in this study, 2-opt and 3-opt operators are applied to each route.

4.3.3. The structure of tabu list

TL records the move operations which generate neighborhood solutions for the Tabu search to avoid cycling. Since different types of move operations have different structures, hence different TL data structures need to be designed to save the move operations. In this study, the move operations are summarized as two different categories: the moved customers are located on different routes; the moved customers are located on the same route. The structure of the two different TL is defined as TL_1 and TL_2, respectively. TL_1 represents the moved customers located on different routes.

Let us assume that the total number of customers is n and the total number of vehicles is k. Then the Tabu list TL_1 can be represented by a $n \times k$ matrix. The element $TL_1(i, j)$ of matrix denotes the forbidden times of movement for inserting customer i to vehicle j. Fig. 5 shows the structure of Tabu list of TL_1. The Tabu List TL_2 is defined for the moved customers located in same route, which can be represented by a three-dimensional array with the size of $(n+2) \times (n+2) \times (n+2)$, in which n is the number of the customers. For each dimension, 2 extra elements are need to represent the depot. The TL_2 mainly gives the information on some forbidden moves, which considers the right-side and the left-side of the customer. Fig. 6 shows the structure of the tabu list of TL_2.

In this study, two different types of attributive memories are adopted for TL_1 and TL_2, which includes the regency-based memory and frequency-based memory. Here, we give two examples for the regency-based memory and frequency-based memory. Let assume that $n = 9$ and $k = 2$. Table 2 shows an example of the regency-based memory for TL_1. The value of an element for the regency-based memory can be a positive, negative integer or zero.

Table 3 shows an example of the frequency-based memory for TL_1. The value of an element for the frequency-based memory must be a positive integer or zero.

Analogously, TL_2 also adapts regency-based memory and frequency-based memory. Here gives an example for the regency-based memory. $TL_2(3, 2, 4) = 5$ means that the movement of resulting $3 \rightarrow 2 \rightarrow 4$ will not be allowed in the following five iterations unless it encounters the criterion of aspirations. Once a movement is performed, the Tabu values $TL_2 = TL_2 - 1$, and the new movement information will be added to the Tabu list.

4.3.4. Intensification and diversification

As mentioned in the above description of designing TL, the frequency-based memory is adopted in this study, which corresponds to the long term memory. In this subsection, intensification and diversification are studied for the frequency-based memory, which are two important criteria which affect the convergence speed and solution quality of Tabu search. As we have pointed out, the NV is a very difficult objective value to be optimized. So, in the intensification stage, we will try to decrease the NV without considering the tabu list and change TD, because the decreasing of the NV could not lead to the circling of the solutions. The intensification will be conducted in every 10 iterations. To escape from poor local minima, a frequency-based diversification technique was proposed by penalizing the high frequency of used moves. The frequency list is established for recording the frequency of the used moves.

Let s be a solution candidate generated from $s^{current}$. Set P represents the available moves excluding the forbidden ones. p' indicates the moves from $s^{current}$ to s. $g(s^{current}, p')$ is used to calculate the frequency of the movement from FL_1. λ is the value of current iterations. $\alpha \in [0,1]$ is a random number. Consider that the current solution is a local optimum whose objective value is f. We choose the best move according to the modified objective function f' from Eq. (11).

$$f'(s) = f(s)(1 + \alpha \sum_{p' \in P} g(s^{current}, p')/\lambda), \tag{11}$$

4.3.5. Aspiration criterion

If the current move operation is recorded in the Tabu list, and the move operation yields a better solution than the current best solution, then an aspiration criterion is triggered to accept the current move operation. We should note that the decreasing of NV is also in the situation of taking aspiration criterion since this means a better solution is obtained than the current best solution.

4.4. Post-optimization

A post-optimization is performed after finishing the tabu search. Due to the tabu list, the final solution, especially intra-route, may not be optimal if we ignore the post-optimization stage. In our two-stage algorithm, we have employed the Generalized Insertion Procedure (GENI), which was originally proposed by Gendreau et al. (1992). GENI can be seen as an iterative insertion heuristic method. Each new vertex u is inserted between two vertices i and j that are not necessarily consecutive in the current tour. However, after insertion, both i and j will be adjacent. GENI can be implemented by two different types. The difference between them is the method of generating a path after insertion. Type 1 is to call 3-opt to generate the final path, while type 2 is to execute 4-opt to generate the final routes. In our work, we used the type 1.

4.5. The pseudo code of TS

The stopping criterion for tabu search is that it stops after the best solutions $s^\star$ has failed to improve after μ time iterations. Due to the two types of tabu lists, in this study, two tabu search named as TS-I and TS-II are proposed. The parameters for TS-I and TS-II are defined as $TL = TL_1$, $FL = FL_1$, $L = L_1$, $TL = TL_2$, $FL = FL_2$, and $L = L_2$. The detailed description of TS is depicted in Algorithm 2.

Algorithm 2: The pseudo code of the proposed TS

Input: current best solution s^*, current solution $s^{current}$, tabu list TL, frequency list FL, tabu length L, and the parameter of termination criterion.
while Stopping criterion is unsatisfied **do**
  $N(s^{current}) \leftarrow \left(N(s^{current}) \backslash forbidden\_solutions\right) \cup aspirated\_solutions$;
    ▷ $N(s^{current})$ denotes the neighborhood solutions of $s^{current}$.
      ▷ remove the forbidden solutions and add the aspirated solutions.
  Find the best solution $s^{new} \in N(s^{current})$;
  if $f(s^{new}) < f(s^\star)$ **then**
    $s^* \leftarrow s^{new}$; ▷ s^{new} is accepted by aspiration criterion.
  end
  $s^{current} \leftarrow s^{new}$; ▷ Replace the current solution with a neighborhood solution.
  Update the tabu list TL;
  Update the frequency list FL.
end
$s^\star \leftarrow postOptimization(s^\star)$. ▷ The best solution is further improved by the post-optimization operators introduced at 4.4.
Return: s^*.

5. Experimental results

In this section, the proposed two-stage algorithm (VNS-BSTS) is evaluated by conducting extensive computational experiments based on benchmark instances and the state-of-the-art results in the literature. The proposed two-stage algorithm is implemented by JAVA programming language.

This section first describes the benchmark instances, then reports the testing results on the small-scale instances, which owns 10, and 25 and 50 customers. After that, the comparison between the solutions obtained by the two-stage algorithm, GA proposed by Wang and Chen (2012), and p-SA introduced by Wang et al. (2015) are carried out to show their performance. Last, but not least, we empirically analyze the main component of the two-stage algorithm to check their role played in the optimization process.

5.1. Benchmark instances

The benchmark instances, originally introduced by Wang and Chen (2012), are popular and widely used in the previous research related to simultaneous delivery and pickup problem with time window. Consequently, we also test our proposed two-stage algorithm by solving the benchmark instances of Wang and Chen (2012). These instances can be categorized as 3 types, namely " Rdp", "Cdp", "RCdp". The detail of the instance naming rules shows in Fig. C.7. All the instances have 100 customers. Wang and Chen (2012) provide geographical positions for every customer and the number of products to be picked up and delivered for each customer, as well as their time windows.

5.2. Parameter settings

Prior to performing the experiments, several trials were undertaken to adjust the algorithm parameters to ensure convergence and speed and to achieve high-quality solutions with reasonable computing time. The parameter settings can be viewed in Table 4.

5.3. Computational results and comparison

To evaluate the efficiency of the two-stage algorithm, we perform a comparison between the solutions obtained by a basic genetic algorithm (BGA) (Wang and Chen, 2012), a co-evolution genetic algorithm (CoGA) (Wang and Chen, 2012), a parallel simulated annealing (p-SA) (Wang et al., 2015) , an Efficient Tabu Search based Procedure (ESTP) (Shi et al., 2018a), an adaptive large neighborhood search (ALNS) (Hof and Schneider, 2019; Hof, 2019) and our proposed two-stage algorithm.

5.3.1. Experimental results for instances with small size

The exact solution can be used to evaluate the accuracy of the proposed two-stage algorithm. Wang and Chen (2012) employed the powerful commercial mathematical programming software CPLEX Solver to find the optimal solutions for the small-scale instances. Table 5 reports the comparison among the best results obtained by CPLEX (Wang and Chen, 2012), CoGA, p-SA and the proposed two-stage algorithm.

As we can see in Table 5, the first column indicates the ID and size of the instances."RCdp" refers to the fact that the customer locations are a mix of uniformly random (R) and clustered (C) positions. The

Table D.12
Comparison with the experimental results with Wang et al. (2015) for all the instances.

Instance ID	P-SA (Wang et al., 2015)		VNS-BSTS			Gap3 (VNS-BSTS VS p-SA)*				
	NV	TD	NV	TD	CT	NV	TD(%)	Is solution improved?	Is solution non-worse?	Is NV improved?
Rdp101	19	1660.98	19	1650.8	34.91	0	−0.61	1	1	0
Rdp102	17	1491.75	17	1486.12	31.6	0	−0.38	1	1	0
Rdp103	14	1226.77	13	1294.75	38.26	−1	5.54	1	1	1
Rdp104	10	1000.65	10	984.81	65.48	0	−1.58	1	1	0
Rdp105	14	1399.81	14	1377.11	33.95	0	−1.62	1	1	0
Rdp106	12	1275.69	12	1261.4	43.65	0	−1.12	1	1	0
Rdp107	11	1082.92	10	1144.02	44.13	−1	5.64	1	1	1
Rdp108	10	962.48	9	968.32	67.67	−1	0.61	1	1	1
Rdp109	12	1181.92	11	1224.86	40.18	−1	3.63	1	1	1
Rdp110	11	1106.52	11	1101.33	50.91	0	−0.47	1	1	0
Rdp111	11	1073.62	10	1117.76	46.72	−1	4.11	1	1	1
Rdp112	10	966.06	10	961.29	65.87	0	−0.49	1	1	0
Cdp101	11	992.88	11	976.04	60.29	0	−1.70	1	1	0
Cdp102	10	955.31	10	942.45	68.87	0	−1.35	1	1	0
Cdp103	10	958.66	10	896.28	95.4	0	−6.51	1	1	0
Cdp104	10	944.73	10	872.39	70.62	0	−7.66	1	1	0
Cdp105	11	989.86	10	1080.63	41.46	−1	9.17	1	1	1
Cdp106	11	878.29	10	963.45	57.3	−1	9.70	1	1	1
Cdp107	11	911.90	10	987.64	79.04	−1	8.31	1	1	1
Cdp108	10	1063.73	10	934.41	79.24	0	−12.16	1	1	0
Cdp109	10	947.90	10	909.27	68.59	0	−4.08	1	1	0
RCdp101	15	1659.59	14	1708.21	32.04	−1	2.93	1	1	1
RCdp102	13	1522.76	13	1526.36	43.85	0	0.24	0	0	0
RCdp103	11	1344.62	11	1336.05	46	0	−0.64	1	1	0
RCdp104	10	1268.43	10	1177.21	43.81	0	−7.19	1	1	0
RCdp105	14	1581.54	14	1548.38	42.41	0	−2.10	1	1	0
RCdp106	13	1418.16	12	1408.19	47.52	−1	−0.70	1	1	1
RCdp107	11	1360.17	11	1295.43	54.7	0	−4.76	1	1	0
RCdp108	11	1169.57	10	1207.6	51.23	−1	3.25	1	1	1
Rdp201	4	1286.55	4	1254.57	46.59	0	−2.49	1	1	0
Rdp202	4	1150.31	3	1202.27	119.56	−1	4.52	1	1	1
Rdp203	3	997.84	3	949.42	107.1	0	−4.85	1	1	0
Rdp204	2	848.01	2	837.13	129.22	0	−1.28	1	1	0
Rdp205	3	1046.06	3	1027.49	79.63	0	−1.78	1	1	0
Rdp206	3	959.94	3	938.63	96.08	0	−2.22	1	1	0
Rdp207	2	899.82	2	912.26	101.46	0	1.38	0	0	0
Rdp208	2	739.06	2	737.26	173.85	0	−0.24	1	1	0
Rdp209	3	947.80	3	940.29	77	0	−0.79	1	1	0
Rdp210	3	1005.11	3	945.97	87.14	0	−5.88	1	1	0
Rdp211	3	812.44	3	805.22	82.25	0	−0.89	1	1	0
Cdp201	3	591.56	3	591.56	53.42	0	0.00	0	1	0
Cdp202	3	591.56	3	591.56	70.56	0	0.00	0	1	0
Cdp203	3	591.17	3	591.17	56.26	0	0.00	0	1	0
Cdp204	3	594.07	3	599.33	50.34	0	0.89	0	0	0
Cdp205	3	588.88	3	588.88	50.98	0	0.00	0	1	0
Cdp206	3	588.49	3	588.49	49.15	0	0.00	0	1	0
Cdp207	3	588.29	3	588.29	50.58	0	0.00	0	1	0
Cdp208	3	599.32	3	588.32	49.61	0	−1.84	1	1	0
RCdp201	4	1513.72	4	1437.48	32	0	−5.04	1	1	0
RCdp202	4	1273.26	3	1412.52	52	−1	10.94	1	1	1
RCdp203	3	1123.58	3	1064.95	74.56	0	−5.22	1	1	0
RCdp204	3	897.14	3	813.74	85.46	0	−9.30	1	1	0
RCdp205	4	1371.08	4	1316.06	33.28	0	−4.01	1	1	0
RCdp206	3	1166.88	3	1154.36	52.15	0	−1.07	1	1	0
RCdp207	3	1089.85	3	1098.64	60.77	0	0.81	0	0	0
RCdp208	3	862.89	3	843.3	73.71	0	−2.27	1	1	0
Total better								46	52	13

* $\text{Gap}_{NV} = NV_{\text{VNS-BSTS}} - NV_{\text{p-SA}}$, $\text{Gap}_{\text{TD}} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{p-SA}}}{TD_{\text{p-SA}}} * 100\%$.

† "1" = True, "0" = False.

four digit numbers that follow "RCdp" is the data set number, and the number that follows the slash denotes the number of customers. The instances used in Table 5 contain 10, 25 and 50 customers. For each instance in the solving approach, we have given the results of NV, TD and computational time. The last column of this table shows the gaps between our algorithm and the proposed CoGA. We find that VNS-BSTS can reach an excellent performance when solving small-size instances because all the gap values are 0.

5.3.2. Experimental results for instances with medium size

Wang and Chen (2012) originally proposed the instances with medium size, including 100 customers in each instance. In this work, they designed the basic GA and CoGA to solve the instances and provided the best solutions they obtained. After that, Wang et al. (2015) proposed a parallel Simulated Annealing for solving the instances of Wang and Chen (2012). Compared with CoGA, the given p-SA algorithm achieved better NV solutions for 12 instances and the same NV solutions for the remaining 44 instances. For the 44 instances with the same NV solutions, a secondary objective, TD, the p-SA gave better solutions than the GA for 16 instances and equal solutions for 7 instances. Additionally, Shi et al. (2018a) proposed the tabu search based heuristics in a conference paper; however, they only tested several instances and reported the results.

Table D.13
Comparison with the experimental results with Shi et al. (2018a) for part of the instances.

Instance ID	ESTP (Shi et al., 2018a)		VNS-BSTS			Gap3 (VNS-TSII VS VNS-BSTS)*				
	NV	TD	NV	TD		NV	TD(%)	Is solution improved?	Is solution non-worse?	Is NV improved?
Rdp201	4	1268.5219	4	1254.57	46.59	0	−1.10	1	1	0
Rdp202	4	1099.6076	3	1202.27	119.56	−1	9.34	1	1	1
Rdp203	3	981.4787	3	949.42	107.1	0	−3.27	1	1	0
Rdp204	3	775.9343	2	837.13	129.22	−1	7.89	1	1	1
Rdp205	3	1045.1245	3	1027.49	79.63	0	−1.69	1	1	0
Rdp206	3	973.4375	3	938.63	96.08	0	−3.58	1	1	0
Rdp207	3	841.2407	2	912.26	101.46	−1	8.44	1	1	1
Rdp208	2	740.8216	2	737.26	173.85	0	−0.48	1	1	0
Rdp209	3	999.063	3	940.29	77	0	−5.88	1	1	0
Rdp210	3	964.59	3	945.97	87.14	0	−1.93	1	1	0
Rdp211	3	805.53	3	805.22	82.25	0	−0.04	1	1	0
Cdp201	3	591.56	3	591.56	53.42	0	0.00	0	1	0
Cdp202	3	591.56	3	591.56	70.56	0	0.00	0	1	0
Cdp203	3	591.17	3	591.17	56.26	0	0.00	0	1	0
Cdp204	3	599.87	3	599.33	50.34	0	−0.09	1	1	0
Cdp205	3	588.88	3	588.88	50.98	0	0.00	0	1	0
Cdp206	3	588.49	3	588.49	49.15	0	0.00	0	1	0
Cdp207	3	588.29	3	588.29	50.58	0	0.00	0	1	0
Cdp208	3	588.32	3	588.32	49.61	0	0.00	0	1	0
RCdp201	4	1441.52	4	1437.48	32	0	−0.28	1	1	0
RCdp202	4	1216.59	3	1412.52	52	−1	16.10	1	1	1
RCdp203	3	1106.72	3	1064.95	74.56	0	−3.77	1	1	0
RCdp204	3	900.65	3	813.74	85.46	0	−9.65	1	1	0
RCdp205	5	1253.44	4	1316.06	33.28	−1	5.00	1	1	1
RCdp206	4	1161.89	3	1154.36	52.15	−1	−0.65	1	1	1
RCdp207	3	1125.5	3	1098.64	60.77	0	−2.39	1	1	0
RCdp208	3	873.28	3	843.3	73.71	0	−3.43	1	1	0
Total better								20	27	6

* $Gap_{NV} = NV_{VNS\text{-}BSTS} - NV_{ETSP}$, $Gap_{TD} = \frac{TD_{VNS\text{-}BSTS} - TD_{ETSP}}{TD_{ETSP}} * 100\%$.
† "1" = True, "0" = False.

Hof and Schneider (2019) recently proposed an adaptive large neighborhood search with path re-linking (LANS-PR) to solve the same problem with us. There is no doubt that ALNS-PR is an efficient algorithm. In fact, according to Hof and Schneider (2019), ALNS-PR obtained an amazing improvement to the results obtained by Wang et al. (2015) and Wang and Chen (2012). Table 6 provides a comparison between the solution obtained by VNS-BSTS and ALNS-PR.

The detailed comparisons among our solutions and the previous studies could be found in Appendix. In this section, we combine all the experimental results obtained by Wang and Chen (2012), Wang et al. (2015), Shi et al. (2018a), Hof and Schneider (2019) and Hof (2019), so that we can get the Best Known Solutions (BKS). Therefore, the comparison is performed between our solutions and the BKS.

For all 56 instances, the proposed two-stage algorithm obtained the 23 (41.07%) solutions the same or better as compared with BKS. More precisely, 6 (10.71%) solutions are improved on the basis of BKS.

The proposed VNS-BSTS can improve several solutions than the published best results, as well as obtain one-third of the same results. This demonstrates that the proposed two-stage approach is efficient for solving these instances. But we also notice that there is still the potential to improve our solutions.

5.4. Experimental results for instances with large size

In order to further check the efficiency of the proposed algorithm, we also test the instances with large-scale customers (more than 200). The large size instances are originally designed by Wang et al. (2015). Wang et al. (2015) provided the near-optimal solutions without making any comparison, while Hof and Schneider (2019) recently solved the same benchmark instances with amazing improvement. We can find that we improve 22 out of 28 (78.58%) the solutions of instances, and we decrease 21 instances (75.00%). When compared with Hof and Schneider (2019), we find that we have improved solutions of two instances and decrease NV for one instance. We also notice that there is a certain gap between our results and the results in Hof and Schneider (2019) when solving large-scale instances (see Table 7).

6. Analysis and discussions

In this section, we undertake additional analysis to investigate the contribution of each underlying component of the proposed two-stage algorithm. Specifically, we explore the role of the first stage in enhancing the primary objective NV, and the second stage in optimizing the second objective TD.

6.1. Role of the first stage

As discussed earlier, the leading role of VNS is to optimize the primary objective function NV. However, we additionally notice that the second-stage TS algorithm helps to reduce NV while optimizing TD. So naturally, one may ask: is the first stage algorithm really needed? And if the answer is affirmative, we will further consider which of the two stages obtains more improvements of NV? This part will answer this question using an empirical experiment, which makes a comparison between the solution obtained by the two-stage algorithm and the only second stage, respectively.

BSTS, as a VNS-BSTS variant without the first stage, starts with the initial solution generated by PFIH (Solomon, 1987) and then improves it with BSTS. This experiment allows us to recognize how much the first stage further contributes to the reducing of the primary objective NV.

Based on the results presented in Table 8, the gaps between the two algorithms may be summarized as follows:

For all 56 instances, the VNS-BSTS algorithm achieved 51 (91.07%) the same or better solutions as compared with the BSTS. More specifically, in 44 (78.57%) instances, the VNS-BSTS algorithm obtained solutions that had at least one vehicle less than the solutions obtained by the BSTS. For the remaining 30 (53.57%) instances , the NV is

Table D.14
Comparison with the recent results obtained by Hof and Schneider (2019).

Instance ID	ALNS-PR (Hof and Schneider, 2019)		VNS-BSTS			gap*			
	NV	TD	NV	TD	CT	NV	TD(%)	Is solution improved?	Is solution non-worse?
rdp101	19	1650.8	19	1650.8	34.91	0	0.00	0	1
rdp102	17	1486.12	17	1486.12	31.60	0	0.00	0	1
rdp103	13	1297.01	13	1294.75	38.26	0	−0.17	1	1
rdp104	10	984.81	10	984.81	65.48	0	0.00	0	1
rdp105	14	1377.11	14	1377.11	33.95	0	0.00	0	1
rdp106	12	1252.03	12	1261.4	43.65	0	0.75	0	0
rdp107	10	1121.86	10	1144.02	44.13	0	1.98	0	0
rdp108	9	965.54	9	968.32	67.67	0	0.29	0	0
rdp109	11	1194.73	11	1224.86	40.18	0	2.52	0	0
rdp110	10	1148.2	11	1101.33	50.91	1	−4.08	0	1
rdp111	10	1098.84	10	1117.76	46.72	0	1.72	0	0
rdp112	9	1010.42	10	961.29	65.87	1	−4.86	0	1
cdp101	11	976.04	11	976.04	60.29	0	0.00	0	1
cdp102	10	941.49	10	942.45	68.87	0	0.10	0	0
cdp103	10	892.98	10	896.28	95.40	0	0.37	0	0
cdp104	10	871.4	10	872.39	70.62	0	0.11	0	0
cdp105	10	1053.12	10	1080.63	41.46	0	2.61	0	0
cdp106	10	967.71	10	963.45	57.30	0	−0.44	1	1
cdp107	10	987.64	10	987.64	79.04	0	0.00	0	1
cdp108	10	932.88	10	934.41	79.24	0	0.16	0	0
cdp109	10	910.95	10	909.27	68.59	0	−0.18	1	1
rcdp101	14	1776.58	14	1708.21	32.04	0	−3.85	1	1
rcdp102	12	1583.62	13	1526.36	43.85	1	−3.62	0	1
rcdp103	11	1283.52	11	1336.05	46.00	0	4.09	0	0
rcdp104	10	1171.65	10	1177.21	43.81	0	0.47	0	0
rcdp105	14	1548.96	14	1548.38	42.41	0	−0.04	1	1
rcdp106	12	1392.47	12	1408.19	47.52	0	1.13	0	0
rcdp107	11	1255.06	11	1295.43	54.70	0	3.22	0	0
rcdp108	10	1198.36	10	1207.6	51.23	0	0.77	0	0
Rdp201	4	1253.23	4	1254.57	46.59	0	0.11	0	0
Rdp202	3	1191.7	3	1202.27	119.56	0	0.89	0	0
Rdp203	3	946.28	3	949.42	107.10	0	0.33	0	0
Rdp204	2	833.09	2	837.13	129.22	0	0.48	0	0
Rdp205	3	994.43	3	1027.49	79.63	0	3.32	0	0
Rdp206	3	913.68	3	938.63	96.08	0	2.73	0	0
Rdp207	2	890.61	2	912.26	101.46	0	2.43	0	0
Rdp208	2	726.82	2	737.26	173.85	0	1.44	0	0
Rdp209	3	909.16	3	940.29	77.00	0	3.42	0	0
Rdp210	3	939.37	3	945.97	87.14	0	0.70	0	0
Rdp211	2	904.44	3	805.22	82.25	1	−10.97	0	1
Cdp201	3	591.56	3	591.56	53.42	0	0.00	0	1
Cdp202	3	591.56	3	591.56	70.56	0	0.00	0	1
Cdp203	3	591.17	3	591.17	56.26	0	0.00	0	1
Cdp204	3	590.6	3	599.33	50.34	0	1.48	0	0
Cdp205	3	588.88	3	588.88	50.98	0	0.00	0	1
Cdp206	3	588.49	3	588.49	49.15	0	0.00	0	1
Cdp207	3	588.29	3	588.29	50.58	0	0.00	0	1
Cdp208	3	588.32	3	588.32	49.61	0	0.00	0	1
RCdp201	4	1406.94	4	1437.48	32.00	0	2.17	0	0
RCdp202	3	1414.55	3	1412.52	52.00	0	−0.14	1	1
RCdp203	3	1050.64	3	1064.95	74.56	0	1.36	0	0
RCdp204	3	798.46	3	813.74	85.46	0	1.91	0	0
RCdp205	4	1297.65	4	1316.06	33.28	0	1.42	0	0
RCdp206	3	1146.32	3	1154.36	52.15	0	0.70	0	0
RCdp207	3	1061.84	3	1098.64	60.77	0	3.47	0	0
RCdp208	3	828.14	3	843.3	73.71	0	1.83	0	0

* $\mathrm{Gap}_{NV} = \mathrm{NV}_{\text{VNS-BSTS}} - NV_{\text{ALNS-PR}}$, $\mathrm{Gap}_{\mathrm{TD}} = \frac{TD_{\text{VNS-BSTS}} - TD_{\text{ALNS-PR}}}{TD_{\text{ALNS-PR}}} * 100\%$.

† "1" = True, "0" = False.

decreased. The empirical results demonstrate that the learning-based VNS has played a crucial role for decreasing NV.

6.2. Contribution of the second stage

The improvement of the second objective TD mainly relies on BSTS, which composes by TS-I and TS-II. This section mainly analyzes the comparison between the solutions VNS-TS-I, VNS-TS-II and VNS-BSTS.

The comparison of the solutions obtained from the VNS-TS-I and VNS-BSTS for all the instances is detailed in Table 9. For all 56 instances, the VNS-BSTS algorithm achieved 51 (91.07%) the same or better solutions as compared with the BSTS. More specifically, in 44 (78.67%) instances, the VNS-BSTS algorithm obtained better solutions than the VNS-TS-I. The NV for both methods are almost the same. This indicates that the TS-II has played an crucial role in improving TD.

7. Conclusions

Vehicle routing problem (VRP), as a classical NP-hard problem with board applications, has attracted numerous researchers. Vehicle routing problem with simultaneous pickup–delivery and time windows (VRP-SPDTW), regarded as an extension of VRP with adding simultaneous pickup–delivery and time windows constraints, is more computationally challenging and has been widely applied in the logistics industry.

Solving VRPSPDTW will help the logistics companies to reduce cost and provide better customer service, which will increase the competitiveness of logistics companies who are facing fierce competition. In this problem, we are given a set of homogeneous vehicles (the same capacity, engine and etc.), a depot (warehouse), and a set of customers with deterministic pickup–delivery demands and time windows.

VRPSPDTW has two hierarchical optimization objectives. A primary objective of minimizing the number of vehicles (NV) which has significant effect on logistics, because vehicles involve a series of costs including charge of procurement of a vehicle, depreciation cost, cost of repair and maintenance, driver cost, fuel cost, carbon tax and etc. And a secondary objective of minimizing the transportation distance (TD). Due to the complexity to evaluate and calculate the cost generated by the primary objective of NV, usually, previous studies tended to use the f = NV as a objective function to evaluate the cost of NV. We also noticed that the existing methods for solving VRPSPDTW rely on an aggregated weight function to optimize the two objectives, which has the clear drawback of examining many irrelevant candidate solutions. According to the observation of the best solution obtained by previous studies, we found that the optimization of TD is not necessarily a promotion for reducing NV. Therefore, in this paper, an effective learning-based two-stage algorithm, also named as VNS-BSTS, is designed to solve the VRPSPDTW.

In the first stage, we integrated a learning-based objective function with variable neighborhood search (VNS) to minimize the NV while retains the potential structures of a solution which can be used in the second stage to minimize the TD. In the second stage, we optimize TD and NV by developing bi-structure based tabu search. Experimental results on 93 benchmark instances , which are compared with the state-of-the-art results obtained by previous studies, demonstrate that our two-stage algorithm performs remarkably well both in terms of computational efficiency and solution quality. According to the statistical results of the solutions obtained by this study, we found that the two-stage algorithm proposed by this study discovers several improved best-known values. To our knowledge, this is the first two-stage algorithm reaching such a performance. To explore the contributions of the VNS and Tabu search on the proposed two-stage algorithm, finally, we analyzed several key components of the algorithm to highlight their impacts on the performance of the two-stage algorithm. Another contribution of this study is that the two-stage algorithm may give a new research direction for academics to solve different variants of VRP, such as green vehicle routing problem and electric vehicle routing problem, by using two-stage approach. We should also notice that some solutions still do not reach the best know solutions, and it indicates that there is still the potential to improve the searching ability of the proposed algorithm.

The model and meta-heuristic algorithm of this article are of great significance. On the one hand, the objective function, which is based on the learning mechanism, aims to reduce the number of vehicles. This strategy provides a new way of thinking for future research. On the other hand, the research in this paper has significant practical value. In most courier companies, the cost of workers, including monthly salary, insurance, etc., is getting higher and higher. While, in the actual logistics work, a courier corresponds to a vehicle. Therefore, it is of great significance to optimize the number of vehicles (number of workers) first and then optimize the distance of vehicles in order to reduce the operating costs of courier companies and determine the number of workers.

Many exciting directions in this paper deserve further research. (1) The algorithm could be further improved to solve the very large-scale instances (Li et al., 2005; Hof and Schneider, 2019; Hof, 2019). (2) Meta-heuristics (Ezugwu, 2019; Fathollahi-Fard et al., 2018), integrated with learning techniques (Bi and Zhang, 2018) or multi-agent systems, can also be adapted to make an effort to improve the quality of the solutions further. (3) The variants of the proposed two-stage algorithm could apply to other new applications, such as delivery and pickup services with drone (Karak and Abdelghany, 2019), online integrated order picking, and delivery for the O2O community supermarket (Zhang et al., 2019).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to express our great thanks to three anonymous reviewers and the editor for providing us constructive comments and suggestions to improve our work. We acknowledge Prof. Hsiao-Fan Wang and Ying-Yen Chen for kindly providing us the benchmark instances with middle size. We thank Prof. Chao Wang, located at Beijing, for offering us the benchmark instances with large size, as well as encouraging us publish our work as soon as possible. The first author would like to gratefully thank the financial support from European Union's Horizon 2020 research program (Grant Agreement No. 727520), French National Research Agency (Grant Agreement No. ANR-16-CONV-0003), and China Scholarship Council (Grant Agreement No. 201504490047). The current paper is a further extension of the work which had been presented by the first author at the *16th IFAC Symposium on Information Control Problems in Manufacturing*. Thanks for the helpful comments and suggestions from the unknown-name researchers attended in this conference.

Appendix A. The detailed routes of the obtained near-optimal solutions

The detailed routes for the middle size instances can be found at: https://drive.google.com/file/d/18WePZKZCiaG2O-nbgBtMNme06aFqbHD_/view?usp=sharing

The routes for the large size instances can be found at https://drive.google.com/file/d/1BLKIbnizIyWnBkyaMqhT41mpGIUMkKk9/view?usp=sharing

Appendix B. Mathematical model

Input parameters:

V:	Set of all vehicles.
$\|V\|$:	The cardinality of V.
K:	The total number of vehicles in set V.
C:	Set of all customers.
N:	Set of all customers, namely $N = C \cup \{0\} \cup \{n+1\}$.
Q:	The capacity of each vehicle.
i:	The index of customers $(i = 0, 1, 2, \dots, n, n+1)$. Especially, $i = 0$ or $n+1$ represents the index of unique depot.
k:	The index of vehicles $(i = 0, \dots, \|V\|)$.
$[a_i, b_i]$:	The time window for customer i. Especially, when $i = 0$ and $i = n+1$, a_i is the opening time of the depot, while b_i is the closing time of the depot.
d_i:	The delivery demand of customer i.
p_i:	The pickup demand of customer i.
c_{ij}:	The transportation cost between customer i and customer j.
t_{ij}:	The travel time between customers i and j for a single trip.
t_i:	The service time for customer i.
σ:	A sufficient large number, which represents the dispatching cost of a vehicle.

Decision variables:

x_{ijk}: If vehicle k travels from customer i to customer j, in which $i \neq j$, $x_{ijk} = 1$. Otherwise, $x_{ijk} = 0$

$y_{ij} \in \mathcal{R}_+$: The demand picked up from customers up to customer i, and transported in arc(i, j);

$z_{ij} \in \mathcal{R}_+$: The demand to be delivered up to customer i and transported in arc (i, j).

S_{ik}: The beginning service time of customer i.

The mixed-integer programming model can be formulated as:

$$\min \sigma \cdot \underbrace{\sum_{k\in V}\sum_{j\in C} x_{0jk}}_{\text{Number of Vehicles}} + \underbrace{\sum_{k\in V}\sum_{i\in N}\sum_{j\in N, j\neq i} c_{ij}x_{ijk}}_{\text{Transportation Distance}} \tag{B.1}$$

s.t.

$$\sum_{k\in V}\sum_{j\in N} x_{ijk} = 1, \forall i \in C, \tag{B.2}$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0, \forall h \in C; k \in V, \tag{B.3}$$

$$\sum_{j\in N} x_{0jk} \leq 1, \forall k \in V, \tag{B.4}$$

$$\sum_{j\in N} x_{j(n+1)k} \leq 1, \forall k \in V, \tag{B.5}$$

$$\sum_{i\in N} y_{ji} - \sum_{i\in N} y_{ij} = p_j, \forall j \in C, \tag{B.6}$$

$$\sum_{i\in N} z_{ij} - \sum_{i\in N} z_{ji} = d_j, \forall j \in C, \tag{B.7}$$

$$y_{ij} + z_{ij} \leq Q \sum_{k\in V} x_{ijk}, \forall i, j \in N, \tag{B.8}$$

$$S_{ik} + t_i + t_{ij} - M(1 - x_{ijk}) \leq S_{jk}, i, j \in N; k \in V, \tag{B.9}$$

$$a_i \leq S_{ik} \leq b_i, i \in N; k \in V, \tag{B.10}$$

$$x_{ijk} \in \{0, 1\}, y_{ij} \geq 0, z_{ij} \geq 0, i, j \in N, k \in V, \tag{B.11}$$

Objective function (B.1) aims to minimize the Number of Vehicles (NV) and Transportation Distance (TD). Constraint (B.2) ensures that each customer is serviced exactly once. Constraint (B.3) indicates the flow conservation. Constraints (B.4)–(B.5) indicate that each assigned vehicle departs from the depot and must return to the depot. Constraints (B.6)–(B.7) are flow equations for pickup and delivery demand. Constraint (B.8) is the capacity constraint. Constraints (B.9)–(B.10) ensure that each customer must be visited with the time window constraints. Finally, constraint (B.11) defines the nature of the decision variables.

Appendix C. The description of the benchmark instances

See Fig. C.7.

Appendix D. The comparison with the previous studies in the same instances

See Tables D.10–D.14.

References

Angelelli, E., Mansini, R., 2002. The vehicle routing problem with time windows and simultaneous pick-up and delivery. In: Quantitative Approaches to Distribution Logistics and Supply Chain Management. Springer, pp. 249–267.

de Armas, J., Melián-Batista, B., Moreno-Pérez, J.A., Brito, J., 2015. Gvns for a real-world rich vehicle routing problem with time windows. Eng. Appl. Artif. Intell. 42, 45–56.

Avci, M., Topaloglu, S., 2015. An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. Comput. Ind. Eng. 83, 15–29.

Badeau, P., Guertin, F., Gendreau, M., Potvin, J.-Y., Taillard, E., 1997. A parallel tabu search heuristic for the vehicle routing problem with time windows. Transp. Res. C 5 (2), 109–122.

Belgin, O., Karaoglan, I., Altiparmak, F., 2018. Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. Comput. Ind. Eng. 115, 1–16.

Belhaiza, S., Hansen, P., Laporte, G., 2014. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput. Oper. Res. 52, 269–281.

Bent, R., Van Hentenryck, P., 2004. A two-stage hybrid local search for the vehicle routing problem with time windows. Transp. Sci. 38 (4), 515–530.

Bi, J., Zhang, C., 2018. An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowl.-based Syst. 158, 81–93.

Boubahri, L., Addouche, S.-A., El Mhamedi, A., 2011. Multi-ant colonies algorithms for the vrpspdtw. In: Communications, Computing and Control Applications (CCCA), 2011 International Conference on. IEEE, pp. 1–6.

Cao, E.-b., Lai, M.-y., 2007. Vehicle routing problem with simultaneous delivery and pick-up with time windows. In: International Conference on Transportation Engineering 2007. pp. 160–166.

Carrasco, R., Pham, A., Gallego, M., Gortázar, F., Martí, R., Duarte, A., 2015. Tabu search for the max–mean dispersion problem. Knowl.-Based Syst. 85, 256–264.

Çatay, B., 2010. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. Expert Syst. Appl. 37 (10), 6809–6817.

Chen, Y., Hao, J.-K., 2018. Two phased hybrid local search for the periodic capacitated arc routing problem. European J. Oper. Res. 264 (1), 55–65.

Chen, J.-F., Wu, T.-H., 2006. Vehicle routing problem with simultaneous deliveries and pickups. J. Oper. Res. Soc. 57 (5), 579–587.

Davari, S., Zarandi, M.H.F., Turksen, I.B., 2013. A greedy variable neighborhood search heuristic for the maximal covering location problem with fuzzy coverage radii. Knowl.-Based Syst. 41, 68–76.

Dell'Amico, M., Righini, G., Salani, M., 2006. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transp. Sci. 40 (2), 235–247.

Dethloff, J., 2001. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. OR-Spectrum 23 (1), 79–96.

Ezugwu, A.E., 2019. Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times. Knowl.-Based Syst. 172, 15–32.

Fan, J., 2011. The vehicle routing problem with simultaneous pickup and delivery based on customer satisfaction. Procedia Eng. 15, 5284–5289.

Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R., 2018. The social engineering optimizer (seo). Eng. Appl. Artif. Intell. 72, 267–293.

Frifita, S., Masmoudi, M., 2020. Vns methods for home care routing and scheduling problem with temporal dependencies, and multiple structures and specialties. Int. Trans. Oper. Res. 27 (1), 291–313.

Fu, Z.-H., Hao, J.-K., 2015. A three-phase search approach for the quadratic minimum spanning tree problem. Eng. Appl. Artif. Intell. 46, 113–130.

Garcia, B.-L., Potvin, J.-Y., Rousseau, J.-M., 1994. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. Comput. Oper. Res. 21 (9), 1025–1033.

Gendreau, M., Hertz, A., Laporte, G., 1992. New insertion and postoptimization procedures for the traveling salesman problem. Oper. Res. 40 (6), 1086–1094.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13 (5), 533–549.

Goksal, F.P., Karaoglan, I., Altiparmak, F., 2013. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Comput. Ind. Eng. 65 (1), 39–53.

Hof, J., 2019. Metaheuristics for Vehicle-Routing Problems Arising in Sustainable Logistics (Ph.D. thesis). Technische Universität Kaiserslautern.

Hof, J., Schneider, M., 2019. An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery. Networks 74 (3), 207–250.

Hof, J., Schneider, M., Goeke, D., 2017. Solving the battery swap station location-routing problem with capacitated electric vehicles using an avns algorithm for vehicle-routing problems with intermediate stops. Transp. Res. B 97, 102–112.

Homberger, J., Gehring, H., 1999. Two evolutionary netaheuristics for the vehicle routing problem with time windows. INFOR: Inf. Syst. Oper. Res. 37 (3), 297–318.

Jarboui, B., Derbel, H., Hanafi, S., Mladenović, N., 2013. Variable neighborhood search for location routing. Comput. Oper. Res. 40 (1), 47–57.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C., 1991. Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning. Oper. Res. 39 (3), 378–406.

Jun, Y., Kim, B.-I., 2012. New best solutions to vrpspd benchmark problems by a perturbation based algorithm. Expert Syst. Appl. 39 (5), 5641–5648.

Karak, A., Abdelghany, K., 2019. The hybrid vehicle-drone routing problem for pick-up and delivery services. Transp. Res. C 102, 427–449.

Kartal, Z., Hasgul, S., Ernst, A.T., 2017. Single allocation p-hub median location and routing problem with simultaneous pick-up and delivery. Transp. Res. E: Logist. Transp. Rev. 108, 141–159.

Lai, M., Cao, E., 2010. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. Eng. Appl. Artif. Intell. 23 (2), 188–195.

Lai, X., Hao, J.-K., Glover, F., Lü, Z., 2018. A two-phase tabu-evolutionary algorithm for the 0–1 multidimensional knapsack problem. Inform. Sci. 436, 282–301.

Lai, X., Hao, J.-K., Glover, F., Yue, D., 2019a. Intensification-driven tabu search for the minimum differential dispersion problem. Knowl.-Based Syst. 167, 68–86.

Lai, X., Hao, J.-K., Yue, D., 2019b. Two-stage solution-based tabu search for the multidemand multidimensional knapsack problem. European J. Oper. Res. 274 (1), 35–48.

Lai, M.-Y., Liu, C.-S., Tong, X.-J., 2010. A two-stage hybrid meta-heuristic for pickup and delivery vehicle routing problem with time windows. J. Ind. Manag. Optim. 6 (2), 435.

Lai, M., Tong, X., 2012. A metaheurisitic method for vehicle routing problem based on improved ant colony optimization and tabu search. J. Ind. Manag. Optim. 8 (2), 469.

Li, F., Golden, B., Wasil, E., 2005. Very large-scale vehicle routing: new test problems, algorithms, and results. Comput. Oper. Res. 32 (5), 1165–1179.

Lou, S.-Z., Shi, Z.-K., 2005. An effective tabu search algorithm for large-scale and real-time vehicle dispatching problems. In: 2005 International Conference on Machine Learning and Cybernetics, Vol. 6. IEEE, pp. 3579–3584.

Luo, Z., Qin, H., Zhang, D., Lim, A., 2016. Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost. Transp. Res. E: Logist. Transp. Rev. 85, 69–89.

Martínez-Puras, A., Pacheco, J., 2016. Moamp-tabu search and nsga-ii for a real bi-objective scheduling-routing problem. Knowl.-Based Syst. 112, 92–104.

Min, H., 1989. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transp. Res. A 23 (5), 377–386.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24 (11), 1097–1100.

Montané, F.A.T., Galvão, R.D., 2002. Vehicle routing problems with simultaneous pick-up and delivery service. Opsearch 39 (1), 19–33.

Montané, F.A.T., Galvao, R.D., 2006. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Comput. Oper. Res. 33 (3), 595–619.

Nagata, Y., Bräysy, O., 2009. A powerful route minimization heuristic for the vehicle routing problem with time windows. Oper. Res. Lett. 37 (5), 333–338.

Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann. Oper. Res. 41 (4), 421–451.

Pacheco, J., Porras, S., Casado, S., Baruque, B., 2018. Variable neighborhood search with memory for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. Knowl.-Based Syst. 145, 236–249.

Peng, Z., Al Chami, Z., Manier, H., Manier, M.-A., 2019. A hybrid particle swarm optimization for the selective pickup and delivery problem with transfers. Eng. Appl. Artif. Intell. 85, 99–111.

Polat, O., Kalayci, C.B., Kulak, O., Günther, H.-O., 2015. A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. European J. Oper. Res. 242 (2), 369–382.

Prins, C., 2009. Two memetic algorithms for heterogeneous fleet vehicle routing problems. Eng. Appl. Artif. Intell. 22 (6), 916–928.

Rodriguez-Tello, E., Hao, J.-K., Torres-Jimenez, J., 2008. An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. Comput. Oper. Res. 35 (10), 3331–3346.

Salhi, S., Imran, A., Wassan, N.A., 2014. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. Comput. Oper. Res. 52, 315–325.

Schneider, E.R., Krohling, R.A., 2014. A hybrid approach using topsis, differential evolution, and tabu search to find multiple solutions of constrained non-linear integer optimization problems. Knowl.-Based Syst. 62, 47–56.

Schneider, M., Stenger, A., Hof, J., 2015. An adaptive vns algorithm for vehicle routing problems with intermediate stops. Or Spectr. 37 (2), 353–387.

Shi, Y., Boudouh, T., Grunder, O., 2017. A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand. Expert Syst. Appl. 72, 160–176.

Shi, Y., Boudouh, T., Grunder, O., 2018a. An efficient tabu search based procedure for simultaneous delivery and pick-up problem with time window. IFAC-PapersOnLine 51 (11), 241–246.

Shi, Y., Boudouh, T., Grunder, O., 2019. A robust optimization for a home health care routing and scheduling problem with consideration of uncertain travel and service times. Transp. Res. E: Logist. Transp. Rev. 128, 52–95.

Shi, Y., Boudouh, T., Grunder, O., Wang, D., 2018b. Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care. Expert Syst. Appl. 102, 218–233.

Silvestrin, P.V., Ritt, M., 2017. An iterated tabu search for the multi-compartment vehicle routing problem. Comput. Oper. Res. 81, 192–202.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.

Solomon, M.M., Desrosiers, J., 1988. Survey paper—time window constrained routing and scheduling problems. Transp. Sci. 22 (1), 1–13.

Stenger, A., Vigo, D., Enz, S., Schwind, M., 2013. An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. Transp. Sci. 47 (1), 64–80.

Subramanian, A., Uchoa, E., Pessoa, A.A., Ochi, L.S., 2013. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. Optim. Lett. 7 (7), 1569–1581.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.-Y., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. Transp. Sci. 31 (2), 170–186.

Tan, K., Lee, L., Ou, K., 2001. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. Eng. Appl. Artif. Intell. 14 (6), 825–837.

Wang, H.-F., Chen, Y.-Y., 2012. A genetic algorithm for the simultaneous delivery and pickup problems with time window. Comput. Ind. Eng. 62 (1), 84–95.

Wang, C., Mu, D., Zhao, F., Sutherland, J.W., 2015. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. Comput. Ind. Eng. 83, 111–122.

Wang, Y., Zhang, J., Assogba, K., Liu, Y., Xu, M., Wang, Y., 2018. Collaboration and transportation resource sharing in multiple centers vehicle routing optimization with delivery and pickup. Knowl.-Based Syst. 160, 296–310.

Wei, Z., Hao, J.-K., 2019. Iterated two-phase local search for the set-union knapsack problem. Future Gener. Comput. Syst. 101, 1005–1017.

Xu, D., Li, K., Zou, X., Liu, L., 2017. An unpaired pickup and delivery vehicle routing problem with multi-visit. Transp. Res. E: Logist. Transp. Rev. 103, 218–247.

Youssef, H., Sait, S.M., Adiche, H., 2001. Evolutionary algorithms, simulated annealing and tabu search: a comparative study. Eng. Appl. Artif. Intell. 14 (2), 167–181.

Yu, Y., Wang, S., Wang, J., Huang, M., 2019. A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. Transp. Res. B 122, 511–527.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2009. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. Expert Syst. Appl. 36 (2), 1070–1081.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2010. An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. European J. Oper. Res. 202 (2), 401–411.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2016. The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints. European J. Oper. Res. 251 (2), 369–386.

Zhang, J., Liu, F., Tang, J., Li, Y., 2019. The online integrated order picking and delivery considering pickers' learning effects for an o2o community supermarket. Transp. Res. E: Logist. Transp. Rev. 123, 180–199.