Iterated Local Search with Neighbourhood Reduction for the Pickups and Deliveries Problem Arising in Retail Industry*

Hanyu Gu$^{1[0000-0003-2035-2583]}$, Lucy MacMillan^{2}, Yefei Zhang^{1}, and Yakov Zinder$^{1[0000-0003-2024-8129]}$

^{1} School of Mathematical and Physical Sciences, University of Technology Sydney
15 Broadway Ultimo NSW 2007, Australia
`Hanyu.Gu@uts.edu.au, Ye.f.zhang@student.uts.edu.au, yakov.zinder@uts.edu.au`
^{2} Australian National Couriers
29 Huntingwood Drive Huntingwood NSW 2148, Australia
`Lucym@ancdelivers.com.au`

Abstract. The paper studies a vehicle routing problem with simultaneous pickups and deliveries that arises in the retail sector, which considers a heterogeneous fleet of vehicles, time windows of the demands, practical restrictions on the drivers and a roster specifying the order of vehicle loading at the depot. The high competition in this industry requires that a viable optimisation approach must achieve a good balance of solution time, quality and robustness. In this paper, a novel iterated local search algorithm is proposed which dynamically reduces the neighbourhood so that only the most promising moves are considered. The results of computational experiments on real-world data demonstrate the high efficiency of the presented optimisation procedure in terms of computation time, stability of the optimisation procedure and solution quality.

Keywords: Vehicle routing problem · Iterated local search · Neighbourhood reduction

1 Introduction

This paper considers a vehicle routing problem with simultaneous pickups and deliveries (VRPSPD) which arises in the retail sector. The features of this problem include: a heterogeneous fleet of vehicles, time window for pickups and deliveries, open routes, restriction on shift length and loading roster at the depot. In spite of the practical importance of these features, few applications in the literature considered all of them simultaneously [9], [6]. Furthermore, the objective of the considered problem is to maximise the number of allocations which is practically essential, but is rarely considered in the literature [9].

Since VRPSPD is NP-hard in the strong sense [3], the majority of the publications in this topic present various heuristics and metaheuristics [9], [6]. In practice, a scheduler expects to produce a good solution within a short time limit, typically no more than one minute. In contrast, most research in the literature focuses more on solution quality. In this paper, an iterated local search [7] based optimisation algorithm is presented to achieve a satisfactory balance between solution quality and computation time.

* Supported by an Australian Government Research Training Program Scholarship

The iterated local search algorithm (ILS) has been widely used to solve combinatorial optimisation problem [7]. It iteratively generates a sequence of local optimums. At each iteration a local search is performed on a problem-specific neighbourhood structure. A perturbation mechanism is employed to avoid local optimum and expand the search space. By allowing infeasible solutions in the designed neighborhood structure [12], [5], ILS has been demonstrated to be much faster than the state-of-the-art for solving the Workforce Scheduling and Routing Problem, which is, from a practical application viewpoint, similar to the studied problem in this paper.

The most time-consuming component in ILS is the evaluation of potential moves in the local search procedure due to the large size of the neighbourhood of the current solution. It is also critical to select proper moves to increase the probability of converging to the global optimum. This paper presents a mechanism to reduce the neighbourhood dynamically, which makes the move evaluation faster, and at the same time direct search in the most promising part of the neighbourhood.

Contributions of this paper include

- development of a MIP model for a VRPSPD problem with many features from the retail sector
- introduction of neighbourhood reduction to speedup the ILS algorithm
- computational studies on real-world data

The remainder of the paper is organised as follows. Section 2 presents the problem formulation. Section 3 describes the proposed iterated local search. Section 4 presents the results for the computational experiments. Section 5 concludes the paper.

2 Problem statement

The considered vehicle routing problem can be stated as follows. Let $G = \{L, A\}$ be a directed graph, where the set of vertices $L = \{0\} \cup C$ and $C = \{1, 2, ..., l\}$, the set of arcs $A = A_D \cup A_C$ and $A_D = \{(0, i) | i \in C\}$, $A_C = \{(i, j) | i \neq j, \forall i, j \in C\}$. Vertex 0 represents the depot and the remaining vertices represent the l customers. Each arc $(i, j) \in A$ has an associated travel time $t_{i,j}$.

The delivery to customer $i \in C$ is characterised by its weight w_i^d and volume v_i^d. The pickup from customer $j \in C$ is characterised by its weight w_i^p and volume v_i^p. For customer $i \in C$, the associated time window $[a_i, b_i]$ indicates the earliest and latest time when the driver can start the corresponding services, and let $p_i > 0$ be the service time required for the driver to complete the service.

Let T be the set of all vehicles. Each vehicle $i \in T$ is differed by its weight capacity W_i and volume capacity V_i. All vehicles $i \in T$ depart from the same single depot and are not required to return to depot after serving all allocated customers. The driver in each vehicle $i \in T$ finishes the shift after serving the last allocated customers. Due to the loading capacity of the depot, each vehicle $i \in T$ arrives at the depot at the specified starting time r_i with loading time δ_i. Furthermore, there exists an upper bound S_i on the shift time of the driver in vehicle $i \in T$, which is the length of time interval between the time when driver starts loading at the depot and the time when driver finishes the service of the last allocated customers.

Each customer $i \in C$ can be served only once, but not all vehicles are capable to serve certain customers. In this paper, two types of vehicles are considered, i.e., the one-man vehicle $T' \subset T$ and the two-men vehicle $T'' \subset T$. The customers are also classified as either one-man customer $C' \subset C$, or two-men customer $C'' \subset C$. The one-man customer can be served by all vehicles, while two-men customer can only be served by two-men vehicles.

The objective is to maximise the total number of allocated customer services while respecting all the constraints on drivers, vehicles and the depot.

Let x_{jk}^i be a binary variable indicating if customer j is the immediate predecessor of customer k in the route of vehicle i; η_j^i be a binary variable indicating if customer j is allocated to vehicle i; γ_j^i be a binary variable indicating if customer j is the first customer to visit after vehicle i departing from the depot; θ_j^i be a binary variable indicating if customer j is the last customer in the route of vehicle i. Denote the time when driver in vehicle i starts serving customer k by s_k^i; the weight of the vehicle when leaving customer j by y_j ; the volume of the vehicle when leaving customer j by z_j. The considered problem can be formulated as follows:

$$J = \max \sum_{i \in T} \sum_{j \in C} \eta_j^i \tag{1}$$

subject to

$$\sum_{i \in T} \eta_j^i \leq 1, \quad \forall j \in C \tag{2}$$

$$\sum_{j \in C} \gamma_j^i \leq 1, \quad \forall i \in T \tag{3}$$

$$\gamma_j^i + \sum_{k \in C} x_{k,j}^i = \eta_j^i, \quad \forall i \in T, j \in C \tag{4}$$

$$\theta_j^i + \sum_{k \in C} x_{j,k}^i = \eta_j^i, \quad \forall i \in T, j \in C \tag{5}$$

$$a_j \leq s_j^i \leq b_j, \quad \forall j \in C, i \in T \tag{6}$$

$$(r_i + \delta_i + t_{0,k})\gamma_k^i \leq s_k^i, \quad \forall i \in T, k \in C \tag{7}$$

$$s_j^i + (p_j + t_{j,k})x_{j,k}^i + (a_k - b_j)(1 - x_{j,k}^i) \leq s_k^i, \quad \forall i \in T, \forall (j,k) \in A_C \tag{8}$$

$$p_j + s_j^i - r_i - (p_j + b_j - r_i)(1 - \theta_j^i) \leq S_i, \quad \forall j \in C, i \in T \tag{9}$$

$$\sum_{k \in C} w_k^d \eta_k^i \leq W_i, \quad \forall i \in T \tag{10}$$

$$y_k \leq W_i + (\max_{e \in T} W_e - W_i)(1 - \eta_k^i), \quad \forall i \in T, k \in C \tag{11}$$

$$\sum_{j \in C} w_j^d \eta_j^i - w_k^d + w_k^p - (\max_{e \in T} W_e - w_k^d + w_k^p)(1 - \gamma_k^i) \leq y_k, \quad \forall i \in T, k \in C \tag{12}$$

$$y_j - w_k^d + w_k^p - (\max_{e \in T} W_e - w_k^d + w_k^p)(1 - x_{j,k}^i) \leq y_k, \quad \forall i \in T, \forall (j,k) \in A_C \tag{13}$$

$$\sum_{k \in C} v_k^d \eta_k^i \leq V_i, \quad \forall i \in T \tag{14}$$

$$z_k \leq V_i + (\max_{e \in T} V_e - V_i)(1 - \eta_k^i), \quad \forall i \in T, k \in C \tag{15}$$

$$\sum_{j \in C} v_j^d \eta_j^i - v_k^d + v_k^p - (\max_{e \in T} V_e - v_k^d + v_k^p)(1 - \gamma_k^i) \leq z_k, \quad \forall i \in T, k \in C \tag{16}$$

$$z_j - v_k^d + v_k^p - (\max_{e \in T} V_e - v_k^d + v_k^p)(1 - x_{j,k}^i) \leq z_k, \quad \forall i \in T, \forall (j,k) \in A_C \tag{17}$$

$$\sum_{i\in T'}\sum_{k\in C''}\eta_k^i = 0 \tag{18}$$

$$x_{j,k}^i \in \{0,1\}, \quad \forall\{j,k\}\in A_C, i\in T \tag{19}$$

$$\eta_j^i \in \{0,1\}, \quad \forall i\in T, j\in C \tag{20}$$

$$\theta_j^i \in \{0,1\}, \quad i\in T, j\in C \tag{21}$$

$$y_j \geq 0, \quad \forall j\in C \tag{22}$$

$$z_j \geq 0, \quad \forall j\in C \tag{23}$$

where (7) and (3) guarantee that a vehicle either stays at the depot or visits exactly one customer; (4) and (5) ensure that each customer must have an immediate successor from the same route except for the last customer; together with (2) ensure that a customer is visited by at most one vehicle; the arrival time, loading time at the depot, travelling time between vertices, the time window are taken into account by (7), (8) and (6)-(8) respectively; the shift length, weight capacity, volume capacity are enforced by (9), (10)-(13), and (14)-(17) respectively; (6) and (8) eliminate subtours by virtue of $p_i > 0$.

3 ILS with neighbourhood reduction

A critical component of ILS is the design of proper neighbourhood structures. It has been demonstrated by many publications that permitting infeasible solutions in local search together with the use of an augmented objective function can significantly boost the performance of the metaheuristics in the field of vehicle routing problem [1,2,8,12]. The neighbourhood structures considered in this paper are defined by the commonly used edge exchange operators, which allows the violation of the time window, shift length, weight and volume capacity constraints. However, the algorithm presented in this paper reduces the size of the neighbourhood by only allowing moves that lead to more allocations than the best known feasible solution. To be specific, let $J(s)$ be the number of allocated customers in a solution s which can be infeasible; $H(s,O)$ be the neighbourhood of a solution s induced by an edge exchange operator O permitting infeasible solution. The corresponding reduced neighbourhood is defined as

$$\widehat{H}(s,O) = \{s' \in H(s,O) | J(s') > J(s^*)\}$$

where s^* be the best known feasible solution. In the studied problem, it is permitted to have customers not allocated. Therefore, feasible solutions can be efficiently generated using simple heuristics (see Section 3.1 for more details). It should be noted that the reduced neighbourhood is dynamic since s^* can be updated in the iterative process of ILS. Since ILS can quickly find good solutions, the size of the reduced neighbourhood becomes significantly smaller after just a few iterations, which leads to faster convergence of the algorithm. Also, the solution process can be more stable because only solutions with more allocations are considered in the local search process.

The paper considers two edge exchange operators

– inter-route swap O_1: exchanges a sequence of up to two consecutive customers in a route with a sequence of up to two consecutive customers in another route; exchanges a sequence of up to two consecutive customers in a route with at most one unallocated customer;

– intra-route swap O_2: extract at most two consecutive customers from a route and insert it into a different position of the same route; reverse the order of a sequence of consecutive customers in the route.

It should be noted that O_2 cannot increase the number of allocated customers. Therefore, it is used mainly for repair infeasibility in the local search procedure.

In the local search procedure, the solution in the reduced neighbourhood is evaluated based on the augmented objective function

$$f(s) = J(s) - \alpha \times TW(s) - \beta \times WD(s) - \sigma \times Weight(s) - \psi \times Volume(s) \quad (24)$$

where $TW(s)$, $WD(s)$, $Weight(s)$, $Volume(s)$ are the total violation for constraints on time window, working duration, weight, volume corresponding to s and $\alpha, \beta, \sigma, \psi$ are non-negative weights for $TW(s)$, $WD(s)$, $Weight(s)$, $Volume(s)$. Furthermore, $TW(s)$, $WD(s)$, $Weight(s)$, $Volume(s)$ are computed by the technique used in [8,10,11].

The details of the local search procedure based on the reduced neighbourhood (NRS) are given in Algorithm 1.

Algorithm 1 NRS(s)

```
1: while TRUE do
2:    if $\widehat{H}(s, O_1) == \emptyset$ then return $s^*$ end if
3:    $s' = s$
4:    $s = argmax_x\{f(x) | x \in \widehat{H}(s, O_1)\}$
5:    if $f(s') < f(s)$ then
6:       $s' = s$
7:       if $s$ is feasible then $s^* = s$ end if
8:    else
9:       Break
10:   end if
11: end while
12: $s = s'$
13: if $\widehat{H}(s, O_2) \neq \emptyset$ then $s = argmax_x\{f(x) | x \in \widehat{H}(s, O_2)\}$ end if
14: if $f(s') < f(s)$ then
15:    if $s$ if feasible then $s^* = s$ end if
16: else
17:    $s = s'$
18: end if
19: return $s$
```

In this pseudocode, the input solution s is permitted to be infeasible. The edge exchange operator O_1 is applied until a local optimum is found under the reduced neighbourhood $\widehat{H}(s, O_1)$. Since the size of the reduced neighbourhood is related to the number of allocations in the current global optimum s^*, s^* is updated whenever a new global optimum is found (line 7 and 15). It should be noted that $\widehat{H}(s, O_1)$ is empty only if the current s^* has all customers allocated, which is also the global optimum. Following the strategy in [12], local search based on the edge exchange operator O_2 is performed for at most one iteration after the local optimum under O_1 is found (line 13). In line

13, $\widehat{H}(s, O_2)$ is empty only if s is a feasible solution. The output of NRS is either the input solution, or a solution with more allocated customers and higher augmented objective function value.

Algorithm 2 ILS with neighbourhood reduction (ILS-NR)

```
1: $s' = $ INITIAL()
2: $s^* = s'$
3: $t = J(s^*)$
4: $h = 1$
5: while $s^*$ has unallocated customers and $h \leq M$ do
6:    $\alpha = \beta = \sigma = \psi = 1$
7:    $e = 1$
8:    repeat
9:       $\bar{s} = s'$
10:      $s'$ =NRS($s'$)
11:      if $f(\bar{s}) \neq f(s')$ then Update $\alpha, \beta, \sigma, \psi$ end if
12:      $e++$
13:   until $f(\bar{s}) == f(s')$ or $s^*$ has unallocated customers or $e > E$
14:   if $J(s^*) > t$ then
15:      $t = J(s^*)$
16:      $h = 1$
17:   end if
18:   $s' = $ PERTURB($h$)
19:   $h++$
20: end while
21: return $s^*$
```

The ILS with neighbourhood reduction (ILS-NR) is now presented in Algorithm 2. It begins with the INITIAL procedure which generates a feasible solution for the problem (line 1). The details of INITIAL is given in Section 3.1. This solution is also the current best known solution s^* (line 2). It should be noted that the current best known s^* is a global variable and may be updated inside the NRS and PERTURB procedure.

After the call of the INITIAL procedure, the WHILE loop (line 5 - 20) is executed if the current best known solution s^* has at least one unallocated customer. The WHILE loop terminates if the current best known solution allocates all customers, or counter h exceeds M which is a parameter. Each iteration of the WHILE loop (line 5 - 20) attempts to find a solution with more allocations than the current best known solution s^* applying the local search procedure (line 8 - 13).

Each iteration of the local search (line 8 - 13) is an applications of NRS which aims to find a solution with a better value of the augmented objective function (24). The penalties for violation of corresponding constraints are updated to force the convergence to feasible solutions. Following [1,2,12], at the beginning of each iteration of the local search (line 8 - line 13), the initial value for weights $\alpha, \beta, \sigma, \psi$ in the augmented objective function (24) are set to one (line 6). If NRS returns an improving solution, a weight is multiplied by $1+\Delta$ if the corresponding constraint has a positive violation; otherwise the weight is divided by $1+\Delta$. Δ is a parameter that controls the strength of the adjustment. This weight updating mechanism is effective in producing feasible solutions, which explains why O_2 is only applied for one iteration in NRS (line 10). Local search terminates if either the NRS procedure fails to obtain a solution with better value of the augmented objective function (24), the current best known solution s^* allocates all customers or the count e exceeds E which is a parameter.

3.1 INITIAL procedure

The INITIAL procedure is a sweep heuristic [4] that constructs a feasible solution for the problem. First a list of customers is constructed based on the geographic coordinates of the customers. Then the customers are inserted to a route one by one until no customer can be inserted, in which case a new route is constructed. Since one-man vehicles can only serve one-man customers, whereas two-men vehicles can serve all-types of customers, the procedure constructs the routes for one-man vehicles first, then followed by the routes for two-men vehicles. When inserting a customer into the route, the procedure chooses the insertion position that respects all the constraints and gives the smallest increase in travel time. The procedure terminates until either no customers can be inserted into the vehicle's route, or all customers have been allocated.

3.2 PERTURB procedure

The PERTURB procedure expands the search space by randomly perturbing the current best solution s^*. An unallocated customer is randomly chosen, and then inserted into a position among the routes which gives the largest value of (24) when $\alpha = \beta = \sigma = \psi = 1$. Then, two randomly selected sequences of consecutive customers are swapped between two randomly selected routes. This random swap will be performed multiple times which depends on the counter h in the pseudocode for the ILS-NR (Algorithm 2). To be specific, the number of swaps starts from one and increases by one each time when counter h in Algorithm 2 increase. The current best solution s^* may also be updated in this process.

4 Computational study

This section presents the results of computational experiments aimed at the evaluation of the performance of ILS-NR. A total of 60 instances were provided by a transportation company working in the retail industry. Each instance represents the real-world situation on a particular day. The travel time from the location of the depot to each customer, and the travel time between the location of each customer are specified by a symmetric matrix. The time when driver arrives at the depot is specified by a roster and each driver can work for a maximum of 10 hours. ILS-NR is implemented in c++, and compiled with g++ O3. The following settings are used throughout the experiments [12]:

– The maximum permissible number of consecutive unsuccessful attempts to improve the current best known solution (the parameter M in Algorithm 2) is computed as $|C| + \lambda|T|)$, where C is the set of all customers, T is the set of all vehicles, $\lambda = 10$.
– The maximum number of exchange operations in the perturbation is five.
– The parameter Δ for adjusting the weights (Section 3) is 0.5.

In addition, the maximum permissible iterations for local search (the paramter E in Algorithm 2) is 100. All computational experiments are conducted on a computer with Intel Xeon CPU E5-2697 v3 2.60GHz and 8GB RAM.

Table 1: Comparison of performance between CPLEX, ILS-NR and CPLEX warm start

			CPLEX			**ILS-NR**				**CPLEX warm start**			
Instances	$\|C\|$	$\|T\|$	Obj	Gap(%)	Time(s)	Avg.	Max.	#Max	Time(s)	Input	Obj	Gap(%)	Time(s)
M-2017-07-23	30	3	27	10.88	9112	**28.00**	**28**	**30**	**0.13**	**28**	**28**	**0.00**	**3711**
M-2017-07-24	26	2	21	9.52	14257	21.93	22	28	0.10	22	22	9.09	21600
M-2017-07-25	14	2	**14**	**0.00**	**1**	**14.00**	**14**	**30**	**0.00**	**14**	**14**	**0.00**	**0**
M-2017-10-08	28	2	24	12.50	10288	24.63	26	1	0.17	26	26	3.85	21600
M-2017-10-09	22	2	21	4.76	21600	21.00	21	30	0.03	21	21	4.76	21600
M-2017-10-10	22	2	17	11.76	21600	17.00	17	30	0.07	17	17	11.76	11304
M-2017-10-16	34	2	26	19.99	21600	26.10	27	3	0.30	27	27	15.55	21600
M-2017-10-17	24	2	21	9.52	21600	21.30	22	10	0.10	22	22	4.55	21600
M-2017-10-21	34	2	24	29.17	21600	26.87	28	1	0.33	28	28	12.75	21600
M-2017-10-24	17	2	**17**	**0.00**	3	**16.90**	**17**	**27**	**0.00**	**17**	**17**	**0.00**	**0**
M-2017-10-30	37	2	27	29.63	21600	28.90	30	1	0.40	30	30	16.85	21600
M-2017-12-22	72	7	66	9.09	21600	69.43	70	13	3.47	70	70	2.86	21600
M-2017-12-23	70	5	59	18.64	21600	65.80	67	3	3.47	67	67	4.48	21600
M-2017-12-24	70	5	50	40.00	21600	57.47	59	1	3.20	59	59	18.64	21600
M-2017-12-25	70	5	52	25.00	21600	57.50	59	1	3.40	59	59	10.17	21600
R-2017-07-23	47	5	**47**	**0.00**	463	**47.00**	**47**	**30**	**0.00**	**47**	**47**	**0.00**	**1**
R-2017-07-24	65	3	48	14.58	21600	52.13	53	5	2.60	53	53	3.77	21600
R-2017-07-25	43	4	**42**	**0.00**	19472	**42.00**	**42**	**30**	**0.50**	**42**	**42**	**0.00**	**0**
R-2017-10-08	88	6	80	8.71	21600	**85.60**	**86**	**18**	**7.40**	**86**	**86**	**0.00**	**18582**
R-2017-10-09	63	4	54	5.56	21600	55.27	56	8	2.37	56	56	1.79	21600
R-2017-10-10	44	5	**44**	**0.00**	593	**44.00**	**44**	**30**	**0.00**	**44**	**44**	**0.00**	**0**
R-2017-10-16	72	5	64	9.37	21600	68.67	69	20	3.50	69	69	1.77	21600
R-2017-10-17	37	4	34	8.82	5084	35.93	36	28	0.37	36	36	2.78	21600
R-2017-10-21	60	5	55	5.45	21600	**58.00**	**58**	**30**	**1.80**	**58**	**58**	**0.00**	**1**
R-2017-10-24	53	6	**53**	**0.00**	**790**	**53.00**	**53**	**30**	**0.00**	**53**	**53**	**0.00**	**1**
R-2017-10-30	71	7	69	2.90	21600	**70.67**	**71**	**20**	**1.43**	**71**	**71**	**0.00**	**1**
R-2017-12-12	52	4	49	6.12	21600	**51.43**	**52**	**18**	**0.67**	**52**	**52**	**0.00**	**1**
R-2017-12-19	52	4	46	10.87	21600	**50.47**	**51**	**14**	**1.20**	**51**	**51**	**0.00**	**0**
R-2017-12-22	62	4	53	15.09	21600	57.03	58	3	2.47	58	58	5.17	21600
R-2017-12-23	70	5	63	9.52	21600	67.73	68	22	3.23	68	68	1.47	21600
R-2017-12-24	70	5	56	10.71	21600	**60.70**	**62**	**1**	**3.27**	**62**	**62**	**0.00**	**2**
R-2017-12-25	70	5	65	7.69	21600	**69.77**	**70**	**23**	**0.93**	**70**	**70**	**0.00**	**1**
T-2017-07-23	64	5	63	1.59	21600	**64.00**	**64**	**30**	**0.00**	**64**	**64**	**0.00**	**1**
T-2017-07-24	70	5	67	2.99	21600	**69.00**	**69**	**30**	**2.63**	**69**	**69**	**0.00**	**1**
T-2017-07-25	57	4	55	3.64	21600	**56.77**	**57**	**23**	**0.57**	**57**	**57**	**0.00**	**0**
T-2017-10-08	65	8	**65**	**0.00**	**3834**	**65.00**	**65**	**30**	**0.00**	**65**	**65**	**0.00**	**1**
T-2017-10-09	43	7	**43**	**0.00**	**31**	**43.00**	**43**	**30**	**0.00**	**43**	**43**	**0.00**	**1**
T-2017-10-10	46	5	**46**	**0.00**	**675**	**46.00**	**46**	**30**	**0.00**	**46**	**46**	**0.00**	**0**
T-2017-10-16	63	7	**63**	**0.00**	**6631**	**63.00**	**63**	**30**	**0.00**	**63**	**63**	**0.00**	**2**
T-2017-10-17	56	4	49	12.24	21600	52.53	53	16	1.43	53	53	3.77	13380
T-2017-10-21	76	4	58	8.62	21600	61.93	62	28	4.23	62	62	1.61	21600
T-2017-10-24	62	4	52	10.05	21600	55.33	56	10	2.30	56	56	1.79	21600
T-2017-10-30	36	5	**36**	**0.00**	**13**	**36.00**	**36**	**30**	**0.00**	**36**	**36**	**0.00**	**0**
T-2017-12-12	63	7	**63**	**0.00**	**1345**	**63.00**	**63**	**30**	**0.00**	**63**	**63**	**0.00**	**2**
T-2017-12-19	54	5	**54**	**0.00**	**923**	**54.00**	**54**	**30**	**0.00**	**54**	**54**	**0.00**	**1**
T-2017-12-22	91	7	75	18.67	21600	**88.73**	**89**	**22**	**7.47**	**89**	**89**	**0.00**	**7**
T-2017-12-23	70	5	63	11.11	21600	**69.93**	**70**	**28**	**0.67**	**70**	**70**	**0.00**	**1**
T-2017-12-24	70	5	63	9.52	21600	67.10	68	3	3.77	68	68	1.47	21600
T-2017-12-25	70	5	64	9.37	21600	68.53	69	16	3.23	69	69	1.45	21600
T-2017-12-26	70	5	65	4.62	21600	**67.97**	**68**	**29**	**3.07**	**68**	**68**	**0.00**	**1**
A-2017-10-16	100	4	53	30.19	21600	61.73	63	2	4.90	63	63	9.52	21600
A-2017-12-22	100	7	76	22.37	21600	82.70	84	2	9.20	84	84	10.72	21600
B-2017-10-08	100	6	72	15.16	21600	79.60	80	18	8.70	80	80	3.65	21600
B-2017-10-16	100	5	71	15.49	21600	78.93	80	6	8.03	80	80	2.50	21600
B-2017-10-30	100	7	81	13.58	21600	86.93	88	5	9.40	88	88	4.55	21600
B-2017-12-22	100	4	58	41.79	21600	67.97	70	1	7.90	70	70	17.49	21600
C-2017-07-24	100	5	86	9.30	21600	92.83	93	25	8.77	93	93	1.65	21600
C-2017-10-16	100	7	96	2.08	21600	**97.97**	**98**	**29**	**6.97**	**98**	**98**	**0.00**	**5**
C-2017-10-21	100	4	67	20.47	21600	75.40	76	12	9.67	76	76	6.06	21600
C-2017-12-22	100	7	89	11.24	21600	**97.87**	**99**	**1**	**10.20**	**99**	**99**	**0.00**	**5**

Table 2: Comparison of performance between ILS and ILS-NR

			ILS			**ILS-NR**					
Instances	$\lvert C\rvert$	$\lvert T\rvert$	Avg.	Max	Time(s)	Avg.	%	Max.	%	Time(s)	%
M-2017-07-23	30	3	28.00	28	0.17	28.00	0.00	28	0.00	**0.13**	**20.00**
M-2017-07-24	26	2	21.57	22	0.13	**21.93**	**-1.70**	22	0.00	**0.10**	**25.00**
M-2017-07-25	14	2	14.00	14	0.00	14.00	0.00	14	0.00	0.00	0.00
M-2017-10-08	28	2	24.13	25	0.17	**24.63**	**-2.07**	**26**	**-4.00**	0.17	0.00
M-2017-10-09	22	2	21.00	21	0.07	21.00	0.00	21	0.00	**0.03**	**50.00**
M-2017-10-10	22	2	16.97	17	0.03	**17.00**	**-0.20**	17	0.00	0.07	-100.00
M-2017-10-16	34	2	25.83	26	0.33	**26.10**	**-1.03**	**27**	**-3.85**	**0.30**	**10.00**
M-2017-10-17	24	2	21.03	22	0.10	**21.30**	**-1.27**	22	0.00	0.10	0.00
M-2017-10-21	34	2	26.23	28	0.27	**26.87**	**-2.41**	28	0.00	0.33	-25.00
M-2017-10-24	17	2	17.00	17	0.03	16.90	0.59	17	0.00	**0.00**	**100.00**
M-2017-10-30	37	2	28.47	29	0.40	**28.90**	**-1.52**	**30**	**-3.45**	0.40	0.00
M-2017-12-22	72	7	69.30	70	4.37	**69.43**	**-0.19**	70	0.00	**3.47**	**20.61**
M-2017-12-23	70	5	65.70	67	4.33	**65.80**	**-0.15**	67	0.00	**3.47**	**20.00**
M-2017-12-24	70	5	57.47	59	3.87	57.47	0.00	59	0.00	**3.20**	**17.24**
M-2017-12-25	70	5	57.37	58	4.00	**57.50**	**-0.23**	**59**	**-1.72**	**3.40**	**15.00**
R-2017-07-23	47	5	47.00	47	0.00	47.00	0.00	47	0.00	0.00	0.00
R-2017-07-24	65	3	51.80	52	2.93	**52.13**	**-0.64**	**53**	**-1.92**	**2.60**	**11.36**
R-2017-07-25	43	4	42.00	42	0.67	42.00	0.00	42	0.00	**0.50**	**25.00**
R-2017-10-08	88	6	85.50	86	16.10	**85.60**	**-0.12**	86	0.00	**7.40**	**54.04**
R-2017-10-09	63	4	55.07	56	2.90	**55.27**	**-0.36**	56	0.00	**2.37**	**18.39**
R-2017-10-10	44	5	44.00	44	0.00	44.00	0.00	44	0.00	0.00	0.00
R-2017-10-16	72	5	68.90	70	4.73	68.67	0.34	69	1.43	**3.50**	**26.06**
R-2017-10-17	37	4	36.00	36	0.43	35.93	0.19	36	0.00	**0.37**	**15.38**
R-2017-10-21	60	5	58.00	58	2.30	58.00	0.00	58	0.00	**1.80**	**21.74**
R-2017-10-24	53	6	53.00	53	0.00	53.00	0.00	53	0.00	0.00	0.00
R-2017-10-30	71	7	70.77	71	1.33	70.67	0.14	71	0.00	1.43	-7.50
R-2017-12-12	52	4	51.10	52	1.10	**51.43**	**-0.65**	52	0.00	**0.67**	**39.39**
R-2017-12-19	52	4	50.50	51	1.47	50.47	0.07	51	0.00	**1.20**	**18.18**
R-2017-12-22	62	4	56.67	58	9.67	**57.03**	**-0.65**	58	0.00	**2.47**	**74.48**
R-2017-12-23	70	5	67.77	68	4.07	67.73	0.05	68	0.00	**3.23**	**20.49**
R-2017-12-24	70	5	60.73	61	3.73	60.70	0.05	**62**	**-1.64**	**3.27**	**12.50**
R-2017-12-25	70	5	69.83	70	0.87	69.77	0.10	70	0.00	0.93	-7.69
T-2017-07-23	64	5	64.00	64	0.00	64.00	0.00	64	0.00	0.00	0.00
T-2017-07-24	70	5	69.00	69	3.20	69.00	0.00	69	0.00	**2.63**	**17.71**
T-2017-07-25	57	4	56.47	57	0.93	**56.77**	**-0.53**	57	0.00	**0.57**	**39.29**
T-2017-10-08	65	8	65.00	65	0.00	65.00	0.00	65	0.00	0.00	0.00
T-2017-10-09	43	7	43.00	43	0.00	43.00	0.00	43	0.00	0.00	0.00
T-2017-10-10	46	5	46.00	46	0.00	46.00	0.00	46	0.00	0.00	0.00
T-2017-10-16	63	7	63.00	63	0.00	63.00	0.00	63	0.00	0.00	0.00
T-2017-10-17	56	4	52.47	53	1.63	**52.53**	**-0.13**	53	0.00	**1.43**	**12.24**
T-2017-10-21	76	4	61.30	62	4.37	**61.93**	**-1.03**	62	0.00	**4.23**	**3.05**
T-2017-10-24	62	4	55.03	56	2.60	**55.33**	**-0.55**	56	0.00	**2.30**	**11.54**
T-2017-10-30	36	5	36.00	36	0.00	36.00	0.00	36	0.00	0.00	0.00
T-2017-12-12	63	7	63.00	63	0.00	63.00	0.00	63	0.00	0.00	0.00
T-2017-12-19	54	5	54.00	54	0.00	54.00	0.00	54	0.00	0.00	0.00
T-2017-12-22	91	7	88.97	89	15.00	88.73	0.26	89	0.00	**7.47**	**50.22**
T-2017-12-23	70	5	69.50	70	2.17	**69.93**	**-0.62**	70	0.00	**0.67**	**69.23**
T-2017-12-24	70	5	66.90	68	4.03	**67.10**	**-0.30**	68	0.00	**3.77**	**6.61**
T-2017-12-25	70	5	68.17	69	3.97	**68.53**	**-0.54**	69	0.00	**3.23**	**18.49**
T-2017-12-26	70	5	67.87	68	12.83	**67.97**	**-0.15**	68	0.00	**3.07**	**76.10**
A-2017-10-16	100	4	61.23	62	5.97	**61.73**	**-0.82**	**63**	**-1.61**	**4.90**	**17.88**
A-2017-12-22	100	7	82.63	84	10.90	**82.70**	**-0.08**	84	0.00	**9.20**	**15.60**
B-2017-10-08	100	6	79.33	81	16.80	**79.60**	**-0.34**	80	1.23	**8.70**	**48.21**
B-2017-10-16	100	5	78.17	80	9.20	**78.93**	**-0.98**	80	0.00	**8.03**	**12.68**
B-2017-10-30	100	7	86.47	88	17.07	**86.93**	**-0.54**	88	0.00	**9.40**	**44.92**
B-2017-12-22	100	4	67.20	68	8.90	**67.97**	**-1.14**	**70**	**-2.94**	**7.90**	**11.24**
C-2017-07-24	100	5	92.50	93	12.00	**92.83**	**-0.36**	93	0.00	**8.77**	**26.94**
C-2017-10-16	100	7	98.00	98	10.33	97.97	0.03	98	0.00	**6.97**	**32.58**
C-2017-10-21	100	4	75.20	76	16.60	**75.40**	**-0.27**	76	0.00	**9.67**	**41.77**
C-2017-12-22	100	7	98.20	99	13.73	97.87	0.34	99	0.00	**10.20**	**25.73**
Average			56.19	56.70	4.05	56.33	-0.32	56.82	-0.31	2.67	17.61

We first compare the performance of ILS-NR with CPLEX which solves the IP model in Section 2. Furthermore, we test the performance of CPLEX when the best solution from ILS-NR is used as a warm start. Both CPLEX and CPLEX with warm start have a time limit of 6 hours and memory limit of 7.5GB RAM. Version 12.10 of CPLEX is used for all the tests. In Table 1, the groups titled "CPLEX" and "CPLEX warm start" contain results obtained by CPLEX and CPLEX with warm start. In these groups, the objective value, optimality gap, computational time are displayed in columns titled "Obj", "Gap(%)" and "Time(s)". The column titled "Input" in group "CPLEX warm start" displays the objective value of the warm start solution. ILS-NR is run 30 times on each instance with the average objective value ("Avg."), best objective value ("Max."), number of runs the best objective value is obtained ("#Max") and computation time ("Time(s)") being reported under the group "ILS-NR".

According to Table 1, CPLEX can prove optimality for 13 instances. With warm start, CPLEX can prove optimality for another 16 instances with significantly reduced CPU time. Among these 29 instances proved optimality by CPLEX, ILS-NR can find optimal solutions with high frequency (#Max) within 10.2 seconds. For 45 out of 60 instances, the average objective values produced by ILS-NR are better than the objective values produced by CPLEX which has a time limit of 6 hours.

To demonstrate the effectiveness of the neighbourhood reduction, Table 2 presents the computational results for ILS-NR and ILS without neighbourhood reduction. The performance of ILS-NR was measured against ILS by the percentage difference

$$\frac{X_{ILS} - X_{ILS-NR}}{X_{ILS}} \times 100 \tag{25}$$

where X can either be the average objective value (column "Avg."), best found objective value ("Max") or CPU time ("Time"); X_{ILS-NR} is the value obtained by ILS-NR and X_{ILS} is the value obtained by ILS. Therefore, a negative percentage difference indicates that ILS-NR is better with respect to the average objective value and best found objective value, while a positive percentage difference indicates that ILS-NR is better with respect to CPU time. For readers' convenience, the superior results produced by ILS-NR are shown in bold.

In Table 2, ILS-NR is faster than ILS on 41 out of 60 instances with an average difference of 17.61%, which clearly demonstrates the improvement on computation time due to neighbourhood reduction. In terms of stability, the average objective value produced by the ILS-NR outperforms the average objective value produced by ILS on 49 instances.

5 Conclusion

This paper considers a practical vehicle routing problem with simultaneous pickups and deliveries which arises in the retail sector. A novel neighbourhood reduction technique is introduced to enhance the performance of the state-of-the-art iterated local search algorithm. Computational experiments carried out on a set of real-world instances demonstrate the superior performance of the proposed algorithm in terms of computational time, solution quality and stability. The advantage of the proposed algorithm is more conspicuous for time-critical applications given the longest computation time among the test instances is just 10.2 seconds.

References

1. Cordeau, J.F., Gendreau, M., Laporte, G.: A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks: An International Journal **30**(2), 105–119 (1997)

2. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society **52**(8), 928–936 (2001)
3. Garey, M.R., Johnson, D.S.: Computers and intractability, vol. 174. freeman San Francisco (1979)
4. Gillett, B.E., Miller, L.R.: A heuristic algorithm for the vehicle-dispatch problem. Operations research **22**(2), 340–349 (1974)
5. Gu, H., Zhang, Y., Zinder, Y.: Lagrangian relaxation in iterated local search for the workforce scheduling and routing problem. In: International Symposium on Experimental Algorithms. pp. 527–540. Springer (2019)
6. Koç, Ç., Laporte, G., Tükenmez, İ.: A review on vehicle routing with simultaneous pickup and delivery. Computers & Operations Research p. 104987 (2020)
7. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search: Framework and applications. In: Handbook of metaheuristics, pp. 129–168. Springer (2019)
8. Nagata, Y., Bräysy, O., Dullaert, W.: A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Computers & operations research **37**(4), 724–737 (2010)
9. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems. Journal für Betriebswirtschaft **58**(2), 81–117 (2008)
10. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & operations research **40**(1), 475–489 (2013)
11. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research **234**(3), 658–673 (2014)
12. Xie, F., Potts, C.N., Bektaş, T.: Iterated local search for workforce scheduling and routing problems. Journal of Heuristics **23**(6), 471–500 (2017)