Multi-trip vehicle routing problem with order release time

Wenli Li\textsuperscript{a}, Yong Wu\textsuperscript{b}, P N Ram Kumar\textsuperscript{c} and Kunpeng Li\textsuperscript{a}\textsuperscript{*}

\textsuperscript{a}School of Management, Huazhong University of Science & Technology, Wuhan, PR China;

\textsuperscript{b}Gold Coast Campus, Griffith University, Brisbane, Australia

\textsuperscript{c}Quantitative Methods & Operations Management Area, Indian Institute of Management Kozhikode, Kerala, India

Correspondence details:
Email: likp@mail.hust.edu.cn
Multi-trip vehicle routing problem with order release time

Abstract

This article studies a new variant of the vehicle routing problem called the multi-trip vehicle routing problem with order release time. This problem frequently arises in the context of last-mile delivery in e-commerce. The order release time represents the time at which the customers’ goods become available at the depot for final distribution. Vehicles attached to the depot are used to perform multiple trips owing to the relatively short delivery distance. In this work, firstly, a mixed-integer linear programming (MILP) model is formulated. As the problem is a proven NP-hard problem, for solving large sized instances quickly, an adaptive large neighborhood search algorithm combined with a labeling procedure (ALNS-L) is proposed. The performance of the algorithm is further augmented by incorporating an optimal serving sequence property. The effectiveness of both the mathematical model and ALNS-L framework is verified by conducting extensive computational experiments on existing benchmark problems and real-life data.

Keywords: multi-trip vehicle routing, last-mile delivery, e-commerce, order release time

1. Introduction

Nowadays, consumers are increasingly turning to e-commerce for their shopping needs. As a result, businesses across the globe have begun racing to develop new technologies and supply chain models for increasing parcel volume, expediting deliveries, and satisfying customers while trying to reduce the cost of logistics operations. Logistics operations represent a significant portion of the world economy. The logistics industry is worth USD 750 billion globally and USD 175 billion in Europe alone (GMI 2017). Of particular interest in a product's journey from a warehouse shelf to a customer's doorstep is the ‘last mile’ component of the delivery.
Last-mile logistics has become a popular area of interest for researchers owing to the growing demand for fully integrated omni-channel retailing. Furthermore, last-mile delivery costs account for nearly 53% of the total costs of shipping (Xiao et al. 2017). One of the most important problems in the context of logistics is the vehicle routing problem (VRP). Over the last four decades, many researchers have made significant contributions to developing efficient solution methodologies for solving the VRP and its variants. However, majority of the studies consider using each vehicle to serve only one trip in the given planning horizon. This assumption may be unrealistic when the vehicles are operated in metropolitan cities. For example, many central business districts forbid heavy/freight vehicles to operate during office hours. This constraint forces 3PL operators to restrict the fleet to smaller vehicles of lower capacity. Consequently, the vehicles’ delivery routes become shorter and the possibility of performing multiple trips within a given work shift thus arises. In the literature, as compared with the regular VRPs, few studies are focused on exploring the multi-trip VRP (MTVRP) (Olivera and Viera 2007; Mingozzi, Roberti, and Toth 2013; Cattaruzza et al. 2014a, 2014b; Cattaruzza, Absi, and Feillet 2016).

This research is motivated by a consulting project for a 3PL service provider that manages the last-mile delivery of electrical and electronic home appliances of a large retailing company that employs the ‘bricks-and-clicks’ multi-channel sales strategy. The retail company primarily caters to the demand requirements of customers who reside in metropolitan areas. When pre-paid orders are placed online, the depot is notified for delivery. However, not all the products are readily available at the depot. Therefore, the products are required to first be supplied to the depot before they can be prepared for the final leg of the delivery. This situation introduces the concept of ‘order release time’, which can be defined as the time by which the product is available for dispatch at the depot. This problem of considering multiple trips and the order release time is referred to as a MTVRP with order release time (MTVRPORT). This study extends the work done by Liu, Li, and Liu (2017) by considering the capacitated VRP with order release times and multiple trips. The
problem thus being studied has great practical implications in last-mile delivery and matches the operational characteristics of city logistics.

Cattaruzza, Absi, and Feillet (2016) studied a problem similar to the MTVRPORT. There are two important differences between their work and this study—the consideration of time windows and optimizing criterion. Their study focused on serving customers within the allocated time windows while minimizing the total distance traveled by the vehicles. In contrast, this work is deliberately focused on minimizing the sum of vehicle completion times while neglecting time window constraints. The reasoning for neglecting the time windows and optimizing ‘time’ rather than ‘distance’ is that the problem being studied is specifically applicable to the context of city logistics. That is, customers residing in large metropolitan cities generally tend to be busy with work throughout the day and may not be able to receive the deliveries personally. Hence, in the majority of cases, delivery items are either left with security personnel or in safe-lockers/mail boxes available at the residential complexes. Thus, this situation eliminates the need for the consideration of time windows. Secondly, in a city environment, traversing even short distances may generally require a long time owing to the heavy traffic and travel restrictions imposed. The minimization of completion times facilitates better driver utilization, thereby improving the overall customer service.

The MTVRPORT involves solving three optimization problems simultaneously. It comprises the allocation of orders to trips (order allocation), allocation of trips to vehicles (vehicle allocation), and deciding the visiting sequence of customers in each trip (trip route planning). To efficiently address this problem, firstly, the property of an optimal serving sequence for multiple trips of a single vehicle problem is proposed and proved. Subsequently, this property is extended to the scenario of multiple vehicles with multiple trips and used in the proposed ALNS-L algorithm.

The rest of this article is organized as follows: Section 2 presents a review of the relevant literature. The mathematical formulation of the MTVRPORT is presented in Section 3. Section 4 describes the property of the optimal serving sequence of trips
and the ALNS-L algorithm. Section 5 presents the details regarding the computational experiments and results obtained. Section 6 highlights the conclusions of this work and outlines the future research directions.

2. Literature review

Given the availability of the vast literature pertaining to VRPs, it is simply not practical to present a detailed review in this Section. Hence, the scope of this section is restricted to VRP studies that explicitly consider two important characteristics: a) multiple trips and b) order release times. The MTVRP is a proven \( NP \)-hard problem that started receiving significant attention in the last two decades (Fleischmann 1990; Taillard, Laporte, and Gendreau 1996; Olivera and Viera 2007; Petch and Salhi 2004; Salhi and Petch 2007). Mingozzi, Roberti, and Toth (2013) proposed an exact method for the MTVRP based on two set partitioning-like formulations. Cattaruzza et al. (2014a) developed a memetic algorithm to address the MTVRP with the objective of minimizing the total travel time. Cheikh et al. (2015) defined four neighborhood structures and applied them to a variable neighborhood search algorithm for the MTVRP.

The complexity of real-life logistics operations provides scope for extensions of the MTVRP. For example, Alonso, Alvarez, and Beasley (2008) studied the site-dependent periodic MTVRP in which each customer is assigned a delivery pattern to be served up to \( p \) times in a planning horizon of \( p \) periods. Brandão and Mercer (1997) considered the MTVRP with time windows (MTVRPTW) for biscuit distribution for a British company. Battarra, Monaci, and Vigo (2009) and Cattaruzza et al. (2014b) dealt with the MTVRPTW in the context of delivery of incompatible commodities. The majority of the studies are focused on solution methodologies based on heuristics and meta-heuristics. However, there are a few exceptions (Azi, Gendreau, and Potvin 2007, 2010; Macedo et al. 2011; Hernandez et al. 2014, 2016) that addressed the MTVRPTW using exact techniques.

VRPs with order release times have gained attention over the past few years. Arda et al. (2014) studied a real-world steel coil delivery problem in which steel coils
are delivered from different production sites to customers with the explicit consideration of order release times. Archetti, Feillet, and Speranza (2015) studied a routing problem wherein un-capacitated vehicles are loaded with goods that arrive at the depot over time. Reys, Erera, and Savelsbergh (2018) studied the complexity of VRP with release dates and deadlines based on the work of Archetti, Feillet, and Speranza (2015) and demonstrated that problems with special structure (i.e., half-line) can be solved in polynomial time. Liu, Li, and Liu (2017) developed Tabu search and Lagrangian relaxation procedure to deal with the capacitated VRP with order available time. Archetti et al. (2018) studied a traveling salesman problem with release dates and proposed an iterated local search (ILS) algorithm. Shelbourne, Battarra, and Potts (2018) proposed a path-relinking algorithm for the VRP with release and due dates. The efficacy of the PRA is validated by comparing it with an ILS algorithm.

In summary, it can be observed that the MTVRPORT, as a variant of the MTVRP, has received little attention. Owing to e-commerce and rapid global changes in customers’ shopping preferences, new operational constraints are evolving over time. Thus, it is necessary to further examine established problems by incorporating new realistic and practical considerations. This research gap forms the basis of this study.

3. Mathematical formulation

Let $G = (V, E)$ be a complete and undirected graph, where $V = \{0, 1, ..., n, n + 1\}$ is the set of vertices, and $E = \{(i, j), i, j \in V, i \neq j\}$ is the set of arcs. Vertices 0 and $n + 1$ represent the same depot, and others in $N = \{1, ..., n\}$ correspond to customers. Every arc $(i, j)$ has a non-negative distance and travel time $t_{ij}$ proportional to the distance between the vertices connected by the arc and therefore satisfying the triangular inequality. $K = \{1, 2, ..., m\}$ is a set of available vehicles at the depot. A vehicle can serve multiple trips in the planning horizon $H$. 
For every customer $i$, the corresponding order size is $q_i$ with an order release time $R_i$. The MTVRPORT aims to determine a set of trips and an assignment of each trip to a vehicle, such that the vehicle completion times is minimized and the following conditions are satisfied:

1. Every customer is served only once.
2. Every trip starts and ends at the depot.
3. Trips assigned to the same vehicle do not overlap with respect to time.
4. The completion time of every vehicle does not exceed $H$.
5. The starting time of any trip is not earlier than the release time of every onboard order in the trip.
6. The sum of the demands of the customers in any trip does not exceed the vehicle capacity $Q$.

Before presenting the mathematical model, the sets and indices, parameters, and decision variables are introduced as follows:

Sets and indices:
- $i, j$ indices for customers in the customer set $N = \{1,...,n\}$;
- $k$ index for vehicles in the vehicle set $K = \{1,2,...,m\}$;
- $r, r^*$ indices for trips;
- $T$ set of trips, $T = \{1,2,...,n\}$ (there will be maximum $n$ trips for $n$ customers).

Parameters:
- $q_i$ size of the order belonging to customer $i$;
- $R_i$ release time of the order belonging to customer $i$;
- $t_{ij}$ travel time between customers $i$ and $j$;
- $H$ the length of the planning horizon;
Q \quad \text{vehicle capacity.}

Decision variables:

\( s_i \) \quad \text{service starting time at customer } i;  

\( a_{ir} \) \quad \text{equals 1 if customer } i \text{ is served by trip } r \text{ and equals 0 otherwise;}

\( b_{ij} \) \quad \text{equals 1 if customer } i \text{ is served earlier than customer } j \text{ and equals 0 otherwise;}

\( v_{rk} \) \quad \text{equals 1 if trip } r \text{ is served by vehicle } k \text{ and equals 0 otherwise;}

\( S_r \) \quad \text{service starting time at the depot for trip } r;  

\( F_r \) \quad \text{service completion time after returning to the depot for trip } r;  

\( W_k \) \quad \text{completion time of vehicle } k \text{ for the working day;}

\( y_{rr^*} \) \quad \text{equals 1 if trip } r^* \text{ is traveled after trip } r \text{ and equals 0 otherwise.}

The mathematical model is formulated as follows:

Minimize \( Z = \sum_{k=1}^{m} W_k \) \hspace{1cm} (1)

subject to 

\[ F_r \leq H \sum_{i \in N} a_{ir}, \forall r \in T \] \hspace{1cm} (2)

\[ W_k \geq F_r - H(1 - v_{rk}), \forall r \in T, k \in K \] \hspace{1cm} (3)

\[ S_i \leq s_i - t_{0i} + (1 - a_{ir})H, \forall i \in N, r \in T \] \hspace{1cm} (4)

\[ F_r \geq s_i + t_{0i} - (1 - a_{ir})H, \forall i \in N, r \in T \] \hspace{1cm} (5)

\[ s_j \geq s_i + t_{ij} - (3 - a_{ir} - a_{jr} - b_{ij})H, \forall i, j \in N, i \neq j; r \in T \] \hspace{1cm} (6)

\[ \sum_{r \in T} a_{ir} = 1, \forall i \in N \] \hspace{1cm} (7)

\[ \sum_{r=1}^{n} r_i a_{ir} \leq Q, \forall r \in T \] \hspace{1cm} (8)

\[ S_r \geq a_{ir} R_r, \forall i \in N, r \in T \] \hspace{1cm} (9)
\[ b_j + b_{j'} = 1, \forall i, j \in N, i \neq j \]  
\[ y_{r,r'} + y_{r',r} = 1, \forall r, r^* \in T, r \neq r^* \]  
\[ S_{r'} \geq F_{r'} - (3 - v_{i_k} - v_{r^*_{i_k}} - y_{r_{i_k}, r^*_{i_k}})H, \forall r, r^* \in T, r \neq r^*, k \in K \]  
\[ \sum_{k \in K} v_{i_k} = 1, \forall r \in T \]  
\[ 0 \leq S_{r}, F_{r}, s_{r}, W_k \leq H \]  
\[ a_{r}, b_{r}, v_{i_k}, y_{r^*_{i_k}} \in \{0,1\}, \forall i, j \in N; r, r^* \in T; k \in K \]  

The objective function (1) is to minimize the sum of the vehicle completion times. The completion time of each vehicle includes the waiting time at the depot and the travel time. Constraints (2) ensure that all the trips are completed within the planning horizon \( H \). Constraints (3) imply that the completion time of any vehicle is greater than or equal to the completion time of all the assigned trips. Constraints (4) ensure that the service starting time of any customer is always greater than or equal to the service starting time of that trip at the depot plus the travel time from the depot to the customer. Similarly, Constraints (5) ensure that the service completion time of every trip is not less than the service starting time of any customer it serves plus the travel time from the customer to the depot. Constraints (6) account for the travel time between two customers (in the same trip). Constraints (7) guarantee that every customer is served by only one trip. Constraints (8) ensure that the total load does not exceed the vehicle capacity \( Q \) for all the trips. Constraints (9) take into consideration the order release time, i.e., every vehicle cannot depart before the latest order release time among all the loaded orders. Constraints (10) make the two events—customer \( i \) is served earlier than customer \( j \) and customer \( j \) is served earlier than customer \( i \)—mutually exclusive. The same is defined for trips in Constraints (11). Constraints (12) ensure that the trips assigned to the same vehicle do not overlap over time. Constraints (13) imply that every trip should necessarily be allocated to one of the vehicles. For trip \( r^* \) which does not visit customers can be
seen as \( F_r = 0 \) as per the inequality (2). Lastly, constraints (14) indicate the range of \( S_r, F_r, s_i \) and \( W_k \), and constraints (15) impose the binary nature of the variables.

4. Heuristic algorithm

Owing to NP-hardness, large problem instances of VRP and its variants tend to be computationally intractable. Therefore, a heuristic approach based on an ALNS-L algorithm is proposed for solving the MTVRPORT. The heuristic approach is designed to work in two stages. In the first stage, the ALNS algorithm is used to generate trips. In the second stage, the trips are sorted based on the property of the single vehicle multiple trips problem. Subsequently, this information is used in the labeling procedure for constructing MTVRPORT solutions by assigning trips to vehicles.

4.1. Optimal serving sequence of multiple trips assigned to the same vehicle

In this study, the trip available time represents the earliest time at which the vehicle can depart from the depot, which typically equals the latest order release time of all the orders allocated to the trip. If multiple trips are allocated to the same vehicle, the vehicle has to ensure that all the orders allocated to its next trip are ready before it can start the next trip, which implies that sometimes the vehicle will have to wait for the next trip available time even if it is idle at the depot. Obviously, changing the serving sequence of trips will affect a vehicle’s completion time. The following property describes an optimal trip serving sequence for a vehicle when a set of trips is generated.

Property 1: Given a vehicle and its allocated trip set, there always exists a serving sequence that is at least as good as other serving sequences. This serving sequence is obtained by serving the trips according to the non-decreasing order of the trip available times.

Proof: Assume that there are \( n \) trips, \( r_1, r_2, \ldots, r_n \), that are to be served by one vehicle. Suppose the corresponding trip available times are \( S_1, S_2, \ldots, S_n \), and the trip travel
times are $d_1,d_2,...,d_n$. An implicit assumption here is that the trip travel times are proportional to the trip travel distances. Without loss of generality, $S_1 < S_2 < ... < S_n$ is assumed. Now, it is supposed that there exists an optimum sequence of trips with a serving order that does not follow the non-decreasing order of trip available times. For example, such a trip sequence is $r_n,r_{n-1},...r_1$. Obviously, the relation $S_n > S_{n-1} > ... > S_1$ should hold and therefore, the vehicle completion time will be $S_n + d_n + d_{n-1} + d_{n-2} + ... + d_1 = S_n + d_1 + d_2 + ... + d_n$. On the contrary, if the service sequence follows the non-decreasing order of the trip available times of $r_1,r_2,...,r_{n-1},r_n$, the result will be one of the following $n$ possibilities:

\[
\begin{align*}
S_1 + d_1 + d_2 + ... + d_n \\
S_2 + d_2 + ... + d_n \\
... \\
S_n + d_n
\end{align*}
\]

This indicates that all the possibilities are better than $S_n + d_1 + d_2 + ... + d_n$, and hence, sorting the trip available times in the non-decreasing order will not result in inferior solutions as compared with other trip sequences. While generating the trips for MTVRPORT, this property is applied to the labeling procedure.

### 4.2. Labeling procedure

Drawing the idea proposed by Cattaruzza, Absi, and Feillet (2016), the labeling procedure is designed to find the optimal assignment of trips to vehicles. Initially, the trips are sorted in a non-decreasing order of trip available times in the trip graph as per Property 1 proposed in Section 4.1. These trips are treated as ‘nodes’, where node 0 represents an empty trip without any customer. Labels are associated with nodes in the graph. Each label associated with node $j$ represents a partial assignment of trips to vehicles that goes from node 0 to node $j$. Starting from node 0, every node is labeled until the last node. Every label has $m+4$ fields. The first $m$ fields store the completion times of the $m$ vehicles. The $(m+1)^{th}$ field stores the vehicle number
for the node at this label. The \((m+2)th\) and \((m+3)th\) fields store the immediate predecessor node and its label, respectively. The \((m+4)th\) field stores the cost of the label of that particular node. The cost for each label is calculated as follows:

\[
C = \sum_{k=1}^{N} W_k + p_r \sum_{r=1}^{N} \max\{0, W_k - H_r\} + p_l \sum_{k=1}^{N} \sum_{r=1}^{N} \max\{0, L_r - Q_k\},
\]

where \(W_k\) is the current travel completion time for vehicle \(k\), and \(L_r\) is the load for trip \(r\) (node \(r\)). \(A_k\) represents the set of trips that are assigned to vehicle \(k\), and \(r \in A_k\) indicates that trip \(r\) is allocated to vehicle \(k\). Penalty coefficients \(p_r\) and \(p_l\) are used for overtime and overload with values of 20 and 2, respectively. These values are partially based on empirical observations, which are further validated by conducting preliminary tests. While extending a label, \(m\) new labels are constructed for possible allocation of the new trip to each vehicle. When extending to the last node, i.e., the last trip, the label with the least cost is the desired result, and hence the related solution is obtained.

Label dominance is determined using the rule below. Let \(Lab_1\) and \(Lab_2\) be two labels for the same node \(j\). In expression (17), the overload penalty and overtime penalty from node 0 to node \(j\) of each label are included in the cost function. When a vehicle completes a trip, it returns to the depot to perform the next trip. Therefore, the loads of multiple trips are not accounted for in a vehicle. As \(\sum_{i=1}^{m} \theta_i(\text{Lab}_1, \text{Lab}_2) \geq 0\), expression (17) not only guarantees that the cost of \(Lab_1\) is less than \(Lab_2\), but also includes the maximal additional penalty that can be introduced in the partial assignment represented by \(Lab_1\) as compared to that represented by \(Lab_2\). Obviously, the coefficient \(p_r\) and \(\sum_{i=1}^{m} \theta_i(\text{Lab}_1, \text{Lab}_2)\) make the expression (17) an important source of identifying the dominant labels. Thus, \(Lab_1\) is better than \(Lab_2\) if and only if
\[
C(Lab_1) + p_r \sum_{k=1}^{T} \theta_k(Lab_1, Lab_2) \leq C(Lab_2)
\]  

(17)

where \( C(\cdot) \) is the cost function (16) for any label of node \( j \).

\[
\theta_k(Lab_1, Lab_2) = \max\{0, \min\{H, W_k(Lab_1)\} - \min\{H, W_k(Lab_2)\}\}
\]  

(18)

\( W_k(Lab) \) is the current travel completion time of vehicle \( k \) associated with \( Lab \).

This dominance rule considerably reduces the number of labels required in ALNS-L.

### 4.3. Adaptive large neighborhood search algorithm

Neighborhood search algorithms are a wide class of iterative improvement algorithms that find better solutions by searching the ‘neighborhood’ of the current solution (Quan and Wu 2017). Shaw (1998) introduced the framework of large neighborhood search that starts with an initial solution, and the quality of the solution is improved iteratively by applying one destroy and one repair operator. Though the concept is simple, it may result in blind search and thereby time-consuming. To overcome the drawbacks, Ropke and Pisinger (2006) proposed an ALNS in which a pair of operators is randomly selected at each iteration to make a move, and statistics are computed to favor the most promising pairs. The method is adaptive as the most frequent pairs may change during the search. Since then, the ALNS algorithm has been widely used for solving different variants of the VRP with promising results (Pisinger and Ropke 2007; Azi, Gendreau, and Potvin 2014).

Though the ALNS-L algorithm in this study draws inspiration from Ropke and Pisinger (2006), two special removal operators that are specific to MTVRPORT are introduced. The framework of the ALNS-L is presented in Figure 1. In a manner similar to Xiao et al. (2014), this work introduces a simulated annealing procedure instead of a simple descent mechanism to determine the acceptance of inferior quality solutions. This mechanism aids in diversifying the search and thereby seeks better solutions. The established notations in the simulated annealing are as follows: \( \Gamma \) is the temperature, \( t \) is the lowest temperature, \( r \) is the cooling rate, \( N \) is the number of searches at a certain temperature, \( S_b, S_c \) and \( S_v \) are the best, current,
and new solutions of the MTVRPORT, respectively.

(Insert Figure 1)

4.3.1 ALNS removal operators

This section describes eight removal operators employed in ALNS-L. Of the eight operators, two are designed specifically for MTVRPORT and are known as the release removal operator and distance-release-oriented removal operator. A virtual roulette wheel is used to select the removal operators. Given a solution and an integer \( q \), the removal operators will remove \( q \) customers from the given solution. The value of \( q \) is given by a random number in the interval \([0.1, 0.4]\) multiplied by the size of the instance.

**Release removal operator (OP1):** For a given client \( i \) and solution \( s \), let \( \Delta R(i,j) \) denote the absolute difference in order release times of clients \( i \) and \( j \). The purpose of this operator is to ensure that customers with near identical order release times are allocated to the same trip or to enable the swapping of the positions of the customers while maintaining the feasibility of the solution. Figure 2 describes this operation in detail, where the value of \( L[0] \) represents a customer \( j \) that is nearest to customer \( i \). The measurement of ‘near’ is based on the value of \( \Delta R(i,j) \).

(Insert Figure 2)

**Random removal operator (OP2):** As the name suggests, this removal operator randomly selects \( q \) customers for removal.

**Worst removal operator (OP3):** For a given customer \( j \) and solution \( s \), let \( C^*_j(s) \) denote the cost when \( j \) is removed from \( s \), i.e., \( \Delta C_{-j}(s) = C(s) - C^*_j(s) \). All the customers in solution \( s \) are stored in the decreasing order of \( \Delta C_{-j}(s) \) in an array \( L \). The operator removes the customer \( j \) with the maximum \( \Delta C_{-j}(s) \), until \( q \) customers are removed. Figure 3 shows the step-by-step operations performed with
this removal operator, where \( L(y^p*|L|) \) represents the \( y^p*|L| \)-th customer in the array \( L \). It is also the customer that makes \( \DeltaC_{j,s} \) the \( y^p*|L| \)-th largest. A random positive number that is less than 1.0 is selected to control the position of the customer to be removed in the array \( L \). A parameter \( p \) is introduced, which is a positive integer greater than 1 to control the randomness of the worst removal operator to increase the diversity of the solution.

(Insert Figure 3)

**Related removal operator (OP4):** This operator uses the distance between customers to define their relationship, where the distance is inversely proportional to the relationship. Figure 4 describes the steps of this operator.

(Insert Figure 4)

**Distance-release-oriented removal operator (OP5):** For a given client \( i \) and solution \( s \), let \( \DeltaR(i,j)+t_o \) be the sum of the travel time and absolute value of the difference in order release times of customers \( i \) and \( j \). This operator mirrors Figure 4 and uses \( \DeltaR(i,j)+t_o \) as the ordering criterion. The rationale of this operator is to ensure the feasibility of the solution while swapping the positions of customers with near identical order release times.

**Historical node-pair removal operator (OP6):** This operator uses the historical information of customers. For every pair of nodes \( (i,j) \in E \), there exists a corresponding weight \( C^*(i,j) \), which represents the current best value a solution can achieve using edge \( (i,j) \). \( C^*(i,j) \) is initialized with a large number and updated when a new solution is found. When calculating the cost for a customer, all the \( C^*(i,j) \) are summed along the way from the depot to the customer. The customer with the maximum cost is selected for removal, until \( q \) customers are removed.

**Historical customer-pair removal operator (OP7):** This operator uses the historical
information of customers. For every pair of customers \((i, j) \in E\), there exists a corresponding weight \(H^*(i, j)\) that records the number of services that both customers are served by the same vehicle. \(H^*(i, j)\) is initialized with 0, and every time a new solution is found, the weight for each edge is updated. By summing up all \(H^*(i, j)\) along the way from the depot to the customer, the customer \(i\) with the maximum \(H^*(i, j)\) is removed, until \(q\) customers are removed.

**Distance-oriented removal operator (OP8):** This operator randomly selects one customer from the current solution \(s\) for removal and then removes the \(q-1\) customers nearest to the selected customer.

### 4.3.2 ALNS repair operators

The proposed ALNS-L framework adopts repair operators to reinsert removed customers in the solution. Let \(C_g\) denote the set of customers that are removed by the removal operators.

**Basic greedy repair operator:** Let \(\Delta f(i, r)\) represent the lowest cost increase when customer \(i\) is inserted into trip \(r\). If customer \(i\) does not meet the capacity or work shift constraint, a large number is assigned to \(\Delta f(i, r)\). Using a greedy approach, \((i, r) = \arg\min_{i \in C_g, r \in T} \Delta f(i, r)\) is computed to identify customer \(i\) to be reinserted into the trip \(r\) with the least cost increase. This process is repeated until all the customers in \(C_g\) are reinserted or until there exists no more feasible insertions.

**Regret repair operator:** Let \(\Delta f^g_i\) represent the change in objective function value when customer \(i\) is inserted into the \(g\)-th lowest cost solution. At every iteration, customer \(i\), who satisfies \(i = \arg\max_{i \in C_g} \sum_{k=2}^{g} (\Delta f^k_i - \Delta f^1_i)\), is inserted into the best position in the selected trip. In this operator, \(g\) takes the value of 2, 3, 4,
4.3.3 Initial solution

The regret repair operator \( g = 2 \) is used to construct the initial solution. All the customers are placed in set \( C_R \) before the process of constructing the initial solution.

4.3.4 Adaptive mechanism

To avoid blind search, the proposed ALNS-L framework utilizes the roulette wheel selection mechanism based on their respective scores \( \pi_i \) to choose the removal and repair operators. At every iteration, the current solution is partially destroyed and then repaired using a pair of removal and repair operators. The operators are randomly selected using the roulette wheel selection mechanism to generate a new solution. The scores \( w^h_p \) are initialized to 0 and updated for every operator \( P \) at the end of the time interval \( h \) as follows:

\[
w^h_p = w^h_p + \begin{cases} 
\sigma_1, & \text{if the new solution is better than the global best solution} \\
\sigma_2, & \text{if the new solution is better than the current solution but inferior to the global best solution} \\
\sigma_3, & \text{if the new solution is inferior to the current solution, but is accepted}
\end{cases}
\]

A relatively more successful operator is normally assigned a higher score to ensure its frequent selection. The different values of \( \sigma \) satisfy the expression \( \sigma_1 > \sigma_2 > \sigma_3 \). The operators accumulate their scores within a time interval that roughly translates into 100 iterations.

Using scores \( w^h_p \), \( \pi^{h+1}_p = \varepsilon \frac{w^h_p}{a_p} + (1 - \varepsilon)\pi^h_p \) can be obtained, where \( a_p \) denotes the number of times operator \( P \) is used in the time interval \( h \), while \( \varepsilon \) is the reaction factor that controls the pace at which the weight adjustment algorithm reacts to the changes in the scores \( \pi_i = \pi^h_i / \sum \pi^h_P, P \in all \ removal / repair \ operators \). Based on pilot testing, in this
work, $\varepsilon$ is assigned a value of 0.1.

5. Computational Experiments

In this section, two versions of the proposed ALNS-L approach are evaluated. The first one, ALNS-L$_1$, uses six removal operators (i.e., OP2, OP3, OP4, OP6, OP7, and OP8). The second one, ALNS-L$_2$, uses two new removal operators and includes a total of seven removal operators (i.e., OP1, OP2, OP3, OP4, OP5, OP6, and OP7). The reason for not using all the eight removal operators in ALNS-L$_2$ is because that there is a certain similarity between the operators ‘OP5’ and ‘OP8’. Hence, with the purpose of providing operator ‘OP5’ with more opportunities to be selected as a removal operator, the operator ‘OP8’ is removed in ALNS-L$_2$.

All the algorithms were coded using C++. The computational experiments were conducted on a desktop computer that runs on an Intel Core i5 processor with a 2.5-GHz CPU and 4-GB RAM. Preliminary numerical experiments were conducted to select the values for different parameters. Based on the preliminary results, the parameter $p$ used in the removal operators is set to a value of 6. In simulated annealing, the cooling rate $\gamma$ is assigned a value of 0.95, and the maximum number of iterations at a specific temperature setting is selected as 100. The values of $\sigma_1$, $\sigma_2$, and $\sigma_3$ are set as 33, 9, and 13, respectively. The heuristic is executed five times on every problem instance, and the best solution is reported.

5.1 Capacitated vehicle routing problem with order release time

The instances presented by Liu, Li, and Liu (2017) correspond to the capacitated VRP with order release times. The authors generate Groups A, B, and P problem instances based on the well-known CVRP benchmarks, and the order release time for every customer is generated randomly in the range $[1/8, 1/4]*D/m$, where $D$ is the travel time of the giant route that visits all the customers from index 1 to $n$ in each instance.

While testing the instances provided by Liu, Li, and Liu (2017), for direct comparison, the labeling procedure in ALNS-L$_1$ and ALNS-L$_2$ is removed. The
number of trips is set to the number of vehicles provided in each problem instance, and the vehicle route is assumed to be constrained only by the vehicle capacity.

Table 1 presents the results pertaining to different groups of problem instances A, B, and P. ‘LB’ denotes the best lower bounds, and ‘GTS’ represents the results of Liu, Li, and Liu (2017). For brevity, this section presents only the average results pertaining to different groups. As can be observed from Table 1, both ALNS-L₁ and ALNS-L₂ outperform Liu, Li, and Liu (2017) in terms of computational quality as well as computational time. It should be noted that among the variants, ALNS-L₂ provides better results than ALNS-L₁. The results also demonstrate that the new operators in ALNS-L₂ contribute significantly to the improvement of solutions. Indeed, it can be observed from Table 2 that the release removal operator (OP1) is used most frequently (on an average 17.03%) among all the removal operators. The distance-release-oriented removal operator (OP5), although used relatively less frequently, is still comparable with other operators from the literature.

5.2 ALNS-L applied to real-life problem instances

In this section, the real-life data provided by the 3PL company is used to validate the proposed solution methodologies. It should be noted that some of the information available in the real-life data is masked for confidentiality reasons. For each problem instance, the number of vehicles \( m \) is varied. As \( m \) increases, the planning horizon \( H \) decreases. Problem instances with ‘ \( T_H = / \)’ imply that the planning horizon \( H \) is not controlled.

Among the two versions of ALNS-L, the better one, i.e., ALNS-L₂ is selected for further assessment. The ALNS-L₂’s performance is benchmarked with respect to CPLEX 12.8, and its results are presented in Table 3. The computational time limit for solving problem instances to optimality is set to 10800s. Firstly, across all the problem instances for which the optimal solution is found, the objective function
values provided by ALNS-L are always greater than or equal to the solutions provided
by CPLEX, i.e., $Z_2 \geq Z_1$. This fact validates the correctness of the mathematical model.
From the results, it can be observed that CPLEX provides optimal solutions (identical
LB and UB values) for problem instances only up to 19 customers. However, as the
size of problem instances increases, CPLEX does not converge to optimality within
the stipulated time. On the other hand, ALNS-L$_2$ requires a very short computational
time to converge to solutions that are near-optimal even for larger problem instances.

(Insert Table 3)

Table 4 extends the numerical tests to larger problem instances ($\geq 19$
customers). To demonstrate the impact of order release time, two approaches based on
the proposed ALNS-L$_2$ algorithm are compared: the first approach generates all the
trips without considering order release time (without integration) while the second
approach considers the optimization of trips and order release times simultaneously
(with integration). In the first approach, the change in objective value caused by the
order release times is initially not computed when the removal and repair operators
are used to generate new trips. Secondly, for the allocation of vehicles, the trip
available times for the newly generated trips are calculated. Lastly, Property 1 and the
labeling procedure are used to allocate trips to vehicles, and the solution for the
MTVRP is obtained. In both approaches, it can be observed that when the
number of used vehicles is increased and the planning horizon $H$ is reduced,
expectedly, the objective function value increases. For trips with a longer trip
available time, vehicles could be reused to serve them while ensuring that $H$ is not
violated. Therefore, consideration of order release time in MTVRP results in the
usage of fewer number of vehicles and thereby lower operational costs. As the
number of customer increases, $H$ may be violated if insufficient vehicles are available.
Thus, this situation presents the planner with a trade-off between the cost of
maintaining a higher fleet size and costs pertaining to the availability of skilled
manpower for a longer duration.

(Insert Table 4)

From Table 4, it can also be observed that the objective function values for
‘with integration’ clearly outperform those ‘without integration’. This obviously suggests that considering the order release time aids in reducing costs. On an average, the MTVRPORT effectively reduces the objective function values by 6.45%. The short solution times in these test instances (average of 10.17 s) indicate that ALNS-L₂ can be used by logistics service providers to quickly generate different operation plans while balancing various factors.

6. Conclusions and scope for future work

In this article, a new variant of the VRP christened as the MTVRPORT is studied. The problem has significant relevance particularly in the context of city logistics. With the objective of minimizing the sum of vehicle completion times, a MILP model is presented. This study proposes and proves an optimal property by which multiple trips are allocated to a single vehicle. Subsequently, this property is extended to develop a labeling procedure that is used to augment the efficacy of the ALNS algorithm. By conducting extensive computational experiments on instances available in the literature as well as real-life problems, both the MILP model and ALNS-L framework are validated. The results indicate that the two new removal operators proposed in the ALNS-L framework significantly contribute to obtaining good quality solutions. The results also highlight the importance of considering ‘order release time’ on the vehicle fleet size and overall logistics costs.

Future research can be focused on developing exact solution methodologies for solving problem instances to optimality. It is also worth studying the problem by incorporating uncertainty in the order release times and deadlines for customers.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China [grant number 71372133], [grant number 71620107002], [grant number 71831007].
References


Figure 1: Framework of ALNS-L
Given current solution \( s \), \( q \in N \);

Select a random customer \( i \) from \( s \);

\( L = \) all customers in \( s \), except customer \( i \), in decreasing order of \( \Delta R(i,j) \);

While \( q > 0 \) do

remove \( L[0] \);

\( q = q - 1 \);

Figure 2: Release removal operator

---

Given current solution \( s \), \( q \in N, p \in R \);

Store all customers in \( s \) in decreasing order of \( \Delta C_{-j}(s) \) in an array \( L \);

While \( q > 0 \) do

Choose a random number \( y \) from interval \( [0,1) \);

\( i = L[(\lceil y^p |L| \rceil)] \);

remove \( i \) from \( L \);

\( q = q - 1 \);

Figure 3: Worst removal operator
1. Given current solution $s$, $q \in N, p \in R$;

2. Select a random customer $i$ from $s$;

3. Set remove set $D = \{i\}$;

4. While $|D| < q$ do

   5. Select a random customer $j$ from $D$;

   6. Construct an array $L$ to include all customers in $s$, but not in $D$, in descending order of relationship to $j$;

   7. Choose a random number $y$ from interval $[0,1)$;

   8. $D = D \cup \{i\} \cap \{j\}$;

Figure 4: Related removal operator

<table>
<thead>
<tr>
<th>Group</th>
<th>LB</th>
<th>GTS</th>
<th>ALNS-L$_1$</th>
<th>ALNS-L$_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$Z_1$</td>
<td>$T_1(s)$</td>
</tr>
<tr>
<td>A</td>
<td>1551</td>
<td>1659</td>
<td>7</td>
<td>1648.11</td>
</tr>
<tr>
<td>B</td>
<td>1430</td>
<td>1561</td>
<td>8</td>
<td>1553.48</td>
</tr>
<tr>
<td>P</td>
<td>787</td>
<td>853</td>
<td>10</td>
<td>846.64</td>
</tr>
</tbody>
</table>

Table 2: Removal operator utilization rates (%) in ALNS-L$_2$

<table>
<thead>
<tr>
<th>Group</th>
<th>OP1</th>
<th>OP2</th>
<th>OP3</th>
<th>OP4</th>
<th>OP5</th>
<th>OP6</th>
<th>OP7</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>17.03</td>
<td>15.43</td>
<td>12.06</td>
<td>13.71</td>
<td>12.57</td>
<td>15.49</td>
<td>13.72</td>
</tr>
<tr>
<td>B</td>
<td>17.08</td>
<td>15.48</td>
<td>11.62</td>
<td>14.04</td>
<td>12.53</td>
<td>15.57</td>
<td>13.68</td>
</tr>
<tr>
<td>Avg.</td>
<td>17.03</td>
<td>15.67</td>
<td>12.25</td>
<td>13.75</td>
<td>12.46</td>
<td>15.26</td>
<td>13.58</td>
</tr>
</tbody>
</table>
Table 3: ALNS-L$_2$ compared with CPLEX 12.8

<table>
<thead>
<tr>
<th>No. of customers</th>
<th>No. of vehicles</th>
<th>Planning Horizon</th>
<th>ALNS-L$_2$</th>
<th>CPLEX</th>
<th>Gap (z2-z1)/z2*100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td>m</td>
<td>$T_H$</td>
<td>Obj (z2)</td>
<td>CPU (s)</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>30</td>
<td>20.35</td>
<td>0.13</td>
<td>20.35</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>/</td>
<td>33.11</td>
<td>0.16</td>
<td>33.11</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>50</td>
<td>31.46</td>
<td>0.15</td>
<td>31.46</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>/</td>
<td>46.92</td>
<td>0.19</td>
<td>46.92</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>50</td>
<td>28.97</td>
<td>0.16</td>
<td>28.97</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>/</td>
<td>39.32</td>
<td>0.18</td>
<td>39.32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>50</td>
<td>31.88</td>
<td>0.14</td>
<td>31.88</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>/</td>
<td>43.78</td>
<td>0.21</td>
<td>43.78</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>50</td>
<td>34.99</td>
<td>0.27</td>
<td>34.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>30</td>
<td>51.64</td>
<td>0.22</td>
<td>51.64</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>/</td>
<td>68.29</td>
<td>0.36</td>
<td>68.29</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>60</td>
<td>38.5</td>
<td>1.32</td>
<td>37.55</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>70</td>
<td>115.47</td>
<td>0.32</td>
<td>115.47</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>/</td>
<td>126.27</td>
<td>0.46</td>
<td>126.27</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>60</td>
<td>43.57</td>
<td>1.70</td>
<td>42.87</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>35</td>
<td>51.04</td>
<td>1.37</td>
<td>49.30</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>/</td>
<td>83.55</td>
<td>1.22</td>
<td>83.55</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>60</td>
<td>43.57</td>
<td>1.70</td>
<td>42.87</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>40</td>
<td>62.67</td>
<td>1.71</td>
<td>59.78</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>/</td>
<td>110.15</td>
<td>1.45</td>
<td>110.15</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>200</td>
<td>194.81</td>
<td>2.25</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>105</td>
<td>206.21</td>
<td>2.29</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>80</td>
<td>206.21</td>
<td>2.59</td>
<td>206.27</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>/</td>
<td>225.31</td>
<td>2.28</td>
<td>248.27</td>
</tr>
</tbody>
</table>

Table 4: ALNS-L applied to real world instances: with and without integration

<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>$T_H$</th>
<th>Without integration (z1)</th>
<th>CPU (s)</th>
<th>With integration (z2)</th>
<th>CPU (s)</th>
<th>Gap (z2-z1)/z2*100</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1</td>
<td>60</td>
<td>45.40</td>
<td>1.39</td>
<td>38.5</td>
<td>1.32</td>
<td>-17.9%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>35</td>
<td>65.15</td>
<td>1.37</td>
<td>51.04</td>
<td>1.37</td>
<td>-27.6%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>/</td>
<td>98.65</td>
<td>1.42</td>
<td>83.55</td>
<td>1.22</td>
<td>-18.1%</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>60</td>
<td>47.43</td>
<td>1.98</td>
<td>43.57</td>
<td>1.70</td>
<td>-8.9%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>40</td>
<td>66.03</td>
<td>1.96</td>
<td>62.67</td>
<td>1.71</td>
<td>-5.4%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>/</td>
<td>122.23</td>
<td>2.05</td>
<td>110.15</td>
<td>1.45</td>
<td>-11%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>200</td>
<td>188.07</td>
<td>2.64</td>
<td>194.81</td>
<td>2.25</td>
<td>3.5%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>105</td>
<td>198.67</td>
<td>2.48</td>
<td>206.21</td>
<td>2.29</td>
<td>3.7%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>80</td>
<td>209.67</td>
<td>2.64</td>
<td>206.21</td>
<td>2.59</td>
<td>-1.7%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>/</td>
<td>245.97</td>
<td>2.67</td>
<td>225.31</td>
<td>2.28</td>
<td>-9.2%</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>50</td>
<td>48.75</td>
<td>3.97</td>
<td>47.22</td>
<td>2.86</td>
<td>-3.2%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>40</td>
<td>66.45</td>
<td>3.0</td>
<td>65.52</td>
<td>2.95</td>
<td>-1.4%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>/</td>
<td>134.85</td>
<td>2.98</td>
<td>128.02</td>
<td>2.78</td>
<td>-5.3%</td>
</tr>
<tr>
<td>57</td>
<td>1</td>
<td>200</td>
<td>193.46</td>
<td>5.56</td>
<td>190.11</td>
<td>5.70</td>
<td>-1.8%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>110</td>
<td>202.56</td>
<td>5.64</td>
<td>193.46</td>
<td>5.87</td>
<td>-4.7%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>75</td>
<td>211.66</td>
<td>5.66</td>
<td>202.56</td>
<td>5.67</td>
<td>-4.5%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>/</td>
<td>238.96</td>
<td>5.58</td>
<td>226.51</td>
<td>5.46</td>
<td>-5.5%</td>
</tr>
<tr>
<td>75</td>
<td>1</td>
<td>200</td>
<td>193.27</td>
<td>12.12</td>
<td>197.79</td>
<td>10.63</td>
<td>2.3%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>105</td>
<td>202.67</td>
<td>12.15</td>
<td>197.79</td>
<td>10.52</td>
<td>-2.5%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>70</td>
<td>221.57</td>
<td>12.32</td>
<td>216.69</td>
<td>11.59</td>
<td>-2.3%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>/</td>
<td>256.27</td>
<td>12.48</td>
<td>251.29</td>
<td>11.22</td>
<td>-2.0%</td>
</tr>
<tr>
<td>88</td>
<td>1</td>
<td>260</td>
<td>247.97</td>
<td>21.61</td>
<td>251.45</td>
<td>15.67</td>
<td>1.4%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>130</td>
<td>257.57</td>
<td>19.08</td>
<td>251.45</td>
<td>15.64</td>
<td>-2.4%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>80</td>
<td>276.97</td>
<td>21.17</td>
<td>264.08</td>
<td>15.77</td>
<td>-4.9%</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>/</td>
<td>316.07</td>
<td>19.23</td>
<td>300.15</td>
<td>15.58</td>
<td>-5.3%</td>
</tr>
<tr>
<td>95</td>
<td>1</td>
<td>260</td>
<td>210.93</td>
<td>21.34</td>
<td>199.66</td>
<td>18.97</td>
<td>-5.6%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>130</td>
<td>220.23</td>
<td>21.06</td>
<td>199.66</td>
<td>18.92</td>
<td>-10.3%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>70</td>
<td>238.83</td>
<td>21.55</td>
<td>208.96</td>
<td>19.89</td>
<td>-14.3%</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>/</td>
<td>276.03</td>
<td>21.89</td>
<td>246.16</td>
<td>19.01</td>
<td>-12.1%</td>
</tr>
<tr>
<td>107</td>
<td>1</td>
<td>250</td>
<td>227.56</td>
<td>28.96</td>
<td>226.06</td>
<td>24.20</td>
<td>-0.7%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>90</td>
<td>247.96</td>
<td>28.50</td>
<td>226.06</td>
<td>25.74</td>
<td>-9.7%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>60</td>
<td>269.06</td>
<td>35.57</td>
<td>236.66</td>
<td>26.36</td>
<td>-13.7%</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>/</td>
<td>311.76</td>
<td>28.45</td>
<td>279.26</td>
<td>26.53</td>
<td>-11.6%</td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11.83</td>
</tr>
</tbody>
</table>