IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 9, 2021, accepted January 19, 2022, date of publication February 15, 2022, date of current version March 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152062

A Simulated Annealing Algorithm for the Vehicle Routing Problem With Parcel Lockers

VINCENT F. YU[1,2], **HADI SUSANTO**[1], **PANCA JODIAWAN**[1], **TSAI-WEI HO**[3], **SHIH-WEI LIN**[4,5,6], **AND YU-TSUNG HUANG**[7]

[1]Department of Industrial Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[2]Center for Cyber-Physical Innovation, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[3]Universal Global Scientific Industrial Company Ltd., Nantou 542, Taiwan
[4]Department of Information Management, Chang Gung University, Taoyuan 333, Taiwan
[5]Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 243, Taiwan
[6]Department of Neurology, Linkou Chang Gung Memorial Hospital, Taoyuan 333, Taiwan
[7]Kerry TJ Logistics Company Ltd., Taipei 100, Taiwan

Corresponding author: Shih-Wei Lin (swlin@mail.cgu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST106-2410-H-011-002-MY3; and in part by the Center for Cyber-Physical System Innovation from the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan. The work of Shih-Wei Lin was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST109-2410-H-182-009MY3; and in part by Chang Gung Memorial Hospital under Grant BMRPA19.

ABSTRACT Due to the growth in e-commerce, city logistics needs to cope with the rise of customers' demand. Consequently, it adds to the complexity of the last-mile delivery process. Moreover, this process often contributes significantly to the distribution cost of companies. An alternative to alleviate the problem is the utilization of parcel lockers. It allows the delivery to be extended not only for home delivery but also for locker delivery, which brings several advantages in terms of cost-saving. However, considering multiple delivery options means adding new complexity to the delivery system. In the previous works, vehicle routing problem with time windows (VRPTW) has been widely dealt by only considering home delivery. This research proposes a new VRPTW variant by adding locker delivery as one of the delivery options. The developed problem is called the vehicle routing problem with parcel lockers (VRPPL). The goal of VRPPL is to minimize the total traveling cost. In this research, we formulate a new mathematical programming model and develop a simulated annealing (SA) algorithm to deal with VRPPL. A newly generated set of instances was developed from the well-known Solomon's VRPTW instances. The performance of the proposed algorithm is also presented.

INDEX TERMS Vehicle routing problem, simulated annealing, time window, parcel locker, delivery options.

I. INTRODUCTION

Technology advancement has driven global e-commerce to experience an unexpected fast growth rate. Global e-commerce sales revenue is predicted to grow from \$1.86 trillion in 2016 to \$4.48 trillion in 2021 [1]. This phenomenon brings advantages, e.g., higher revenue, to e-commerce industries. Nevertheless, the industries also face new challenges to perform the last-mile delivery to their customer's destination, which possibly causes several negative impacts, i.e., increasing traffic densities, air pollution, and inefficient deliveries due to spatially distributed customers' locations and failed delivery. Moreover, the delivery cost increases to 28% of the total expenses if the customers have not received the goods successfully [2]. Consequently, to respond to the challenges, the e-commerce industries need to plan delivery strategies that reduce the negative impacts and maintain customer satisfaction.

Various emerging technologies are adopted to increase delivery efficiencies, such as electric vehicles, drop boxes, cargo bikes, product distribution by drone, crowdshipping, autonomous vehicles, and parcel lockers [3]. Parcel lockers are a promising alternative to overcome the issue since it offers flexibility in collection hours, security, convenience, and savings. Instead of delivering the packages directly to the customer, which is the primary cause of the aforementioned negative impacts, parcel lockers serve as facilities from where customers can pick up their packages. By utilizing parcel

The associate editor coordinating the review of this manuscript and approving it for publication was Baozhen Yao.

lockers, delivery efficiency can be improved since it reduces the transportation cost by simultaneously placing several packages in a selected parcel locker [4].

The parcel lockers system has been implemented in more than twenty countries, including the UK, Asia, the US, Europe, and Canada. In US, Amazon has developed a new system, namely, Amazon lockers and Amazon the Hub [5]. In Japan, a similar system is called Lotte BOX. Likewise, the world's most widely distributed and largest parcel locker system is InPost parcel lockers. It has approximately 4,000 stations in 20 countries worldwide.

The utilization of parcel lockers has also gained the interest of the research community. Deutsch and Golany [4] addressed designing a parcel locker network problem, aiming to optimally choose the number, locations, and sizes of parcel locker facilities. The primary objective of this issue is to maximize the total profit from the revenue obtained from customers who use the service minus the fixed and operational costs of the selected facilities. The problem is modeled by binary integer linear programming and was solved by translating it into a well-known uncapacitated facility location problem. Bengtsson and Vikingson [6] conducted an experimental study to examine the effect of parcel locker delivery based on the perspective of customers.

In the perspective of logistics planning, harnessing the parcel lockers system brings a new complexity. Previously, the couriers only needed to serve each customer within their time windows, and hence the sequence of customer visits used to be the only consideration. This problem is formally described as the vehicle routing problem with time windows (VRPTW). By integrating parcel lockers, several new considerations must be included, i.e., which customers need to be assigned to parcel lockers and how to distribute packages into the parcel lockers. Therefore, in this study, we evaluate a new logistics problem called the vehicle routing problem with parcel lockers (VRPPL). The VRPPL considers three types of customers. First, customers who require the delivery at their home location. Second, the customers who select parcel lockers. Third, customers who are willing to receive the packages either at their homes or at particular parcel lockers. VRPPL also considers the capacity of the parcel locker, time windows of each customer, and vehicle's capacity. The objective of VRPPL is to minimize the total travelled distance.

During the pandemic, when the interaction between humans is limited due to social distancing, addressing the VRPPL will make a significant contribution because the delivery system allows consumers to place order(s) and receive goods without interacting with sellers and couriers. This condition can be achieved when parcel lockers are integrated into the delivery system.

Since no previous work has ever dealt with this problem, a new mathematical programming model was developed in this work, inspired by the mathematical model of VRPTW. The model is built as a three-index vehicle flow formulation, wherein, SA is selected for solving the VRPPL. SA is a frequently used algorithm to solve combinatorial problems, particularly VRP classes. In many cases, it shows a competitive performance compared to other metaheuristics [7], [8].

The contribution of the work is then two-fold. First, a new mixed-integer non-linear programming is proposed to describe the problem. AMPL with Gurobi solver was utilized to solve the developed mathematical model. Second, due to the complex nature of the problem, a metaheuristic - SA algorithm - was developed to solve the problem efficiently.

This paper consists of six sections. The following section includes a review of the relevant literature. In section 3, we introduce the mathematical model of VRPPL. The detail of the SA algorithm is described in section 4. Next, section 5 presents the explanation of benchmark instances, experimental setup, results, and analysis. It includes the comparison between Gurobi output and SA output to show the efficiency of the proposed algorithm. Finally, notable findings and insight from the research are summarized in section 6.

II. LITERATURE REVIEW

Due to the fast expansion of the e-commerce market, research on the utilization of parcel lockers has emerged as a trend. From the company perspective, distributing packages to parcel lockers is more convenient than other methods. It could save 55-66% of transportation costs compared to the conventional system [4]. Although the business-to-consumer model is proliferating, especially for home deliveries, their operational cost to handle mass orders becomes quite expensive. Parcel lockers provide more advantages in flexibility, security, and efficiency than regular home delivery [4].

Recently, the integration of parcel lockers in the vehicle routing problem (VRP) was investigated Zhou, *et al.* [9] addressed a multi-depot two-echelon vehicle routing problem with delivery options. The problem involves routing decisions in a two-level distribution network, where the first level considers the routing between the depots and the satellites, and the second level considers the routing between the satellites to the customers. The customers, in this case, maybe the ones who require delivery at their home, or at a selected pickup location where they can pick up the item, or those who are willing to receive the delivery either at home or at a particular pickup location. The main limitation of the model by Zhou, *et al.* [9] is that each customer is available all the time for the delivery. However, in the real world, each customer may have his/ her preferred time window.

Sitek *et al.* [10] introduced the capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD) to minimize the distances traveled by vehicles and the penalty for delivering items to alternative pick-up points. Tilk *et al.* [11] investigated the vehicle routing problem with delivery options (VRPDO) and presented a branch-and-price-and-cut algorithm to solve the problem. In the problem, customers can individually prioritize their delivery options beforehand. Each delivery option defined by the customer has its own time window. The goal of VRPDO is to minimize the total cost of serving all customers by one of their delivery options.

Enthoven *et al.* [12] introduced the two-echelon vehicle routing problem with covering options (2E-VRP-CO) to minimize operational costs. In this research, goods are shipped from a central depot to intermediate locations in the first echelon. There are two ways of serving customers. First, the goods are shipped to covering locations and customers need to pick up the goods by themselves. Second, city freighters originating from satellites deliver the goods to customers' locations.

The VRP is an evergreen research field, which was first mentioned by Dantzig [13]. In VRP, there are several nodes representing customers that have a non-negative demand and therefore need to be served by a vehicle. Several vehicles are available at a depot, and each vehicle has a capacity, which determines the maximum demand that can be loaded in the vehicle. The objective of VRP is to minimize the total travel distance of utilized vehicles. Solomon extended VRP by considering the time-windows constraint, known as the VRPTW [14]. He generated benchmark instances for VRPTW, known as "Solomon's Instances." The VRPTW as a constrained problem is included in the category of NP-hard problems. Intensive research has focused on developing different approaches for solving the problem and its variants over the past years, such as exact approaches [15], heuristic approaches [16], and metaheuristics [17] to solve this type of problem.

The exact approaches, such as mathematical programming, may not be able to deal with large-scale instances because the required runtime rises exponentially as the nodes increase [18]. This issue can be solved by the means of heuristics. The heuristic approaches applied to VRP are categorized into route-constructing and route-improving heuristics [19]. A route-constructing heuristic mainly aims to produce a feasible solution by iteratively inserting a customer into a route based on several cost minimization criteria while considering the problem constraints [20]. A route-improving heuristic seeks to improve the solution [15], [17], [21]. To overcome the tendency of being trapped in local optimum solutions, metaheuristic methods are widely adopted.

The metaheuristic approach proposed for solving the VRP includes single-solution-based, population-based, and hybrid metaheuristics. Single-solution metaheuristics focus on modifying a single candidate solution such as SA [22], tabu search [23], and variable neighborhood search [24]. Population-based metaheuristics deal with multiple candidate solutions. Some variants are listed but not limited to particle swarm optimization (PSO) [18], genetic algorithm (GA) [25], ant colony optimization [26], memetic algorithm [27], and harmony search [28]. The hybrid metaheuristic is a result of the combination of different metaheuristics, integrating a local search procedure, or exact algorithms [29]. Several researchers have developed the hybridization of metaheuristics such as a GA-hybrid [30], Iterated Local Search-hybrid [31], GA-PSO [32], and hybrid-PSO [33].

This research proposes a SA heuristic for VRPPL. SA is a metaheuristic aiming to approximate global optimization in an extensive search area for an optimization problem [34]. The concept of SA was primarily present by Metropolis in 1953, when it was first used to solve the combinatorial optimization problem. Until now, several studies have applied SA in various types of routing problems and scheduling problems, such as traveling salesman problem [35], logistic resource planning [34], allocation problem [36], job rotation scheduling problem [37], pickup and delivery problem [38], and many more. So, it has been successfully applied to various highly complicated combinatorial optimization problems and a wide variety of real case problems.

III. MATHEMATICAL PROGRAMMING MODEL

VRPPL is an extension of VRPTW. The main characteristic of VRPPL that differentiates it from VRPTW is that it considers three types of customers to be served. A type-1 customer, i.e., home delivery customer (HC), is defined as a customer whose package must be delivered to the customer's home within the predetermined time window. This type of delivery commonly occurs in VRPTW. A type-2 customer, i.e., locker delivery customer (LC), is defined as a customer whose package needs to be delivered to a parcel locker station. Each type-2 customer has determined a parcel locker based on his/ her preference. The vehicle which carries the package of a customer of this type will deliver the package to the designated parcel locker, and the customer picks up the package from there. A type-3 customer, i.e., home or locker delivery customer (HLC), is one whose package could be delivered either to the customer's home within the time window or a parcel locker station determined by the customer beforehand.

In this research, each customer whose package is delivered to his/ her home is to be visited exactly once in the planning period. Each available parcel locker in the problem could be visited by more than one vehicle and possibly more than once for each vehicle. All utilized vehicles begin to serve customers from a depot and end the journey at the depot.

The VRPPL is represented as an undirected network. Let $G = (N, A)$ be an undirected network, where $N = N_0 \cup N_{HC} \cup N_{LC} \cup N_{HLC} \cup N_{PL}$ is the set of nodes and A is the set of edges connecting each pair of nodes in N. Each edge $(i, j) \in A$ as a traveling cost c_{ij} and a traveling time t_{ij}. $N_0 = \{0\}$ represents the depot. The set of all customers is $N_C = N_{HC} \cup N_{LC} \cup N_{HLC}$, where N_{HC} represents the set of type-1 customers, N_{LC} represents the set of type-2 customers, and N_{HLC} represents the set of type-3 customers. Each customer i has a non-negative demand d_i, $\forall i \in N_C$. N_{PL} is the set of parcel locker stations, $\xi_j (j \in N_{PL})$ represents the capacity of a parcel locker station, i.e. the number of customers that can be served by the parcel locker station. $N_{CPL} = N_C \cup N_{PL}$ is the set containing all customers and parcel locker stations. Each node i in N_{CPL} has a time window $[a_i, b_i]$ and a service time s_i. The selection of parcel locker by a customer is done beforehand. Thus, ℓ_{ij} $(i \in N_C, j \in N_{PL})$ is defined as a binary parameter for modeling the selection of parcel locker. If customer i chooses to pick up the package at parcel locker j, then ℓ_{ij} $(i \in N_C, j \in N_{PL})$ is 1, otherwise, it equals 0. K is the set of vehicles with a homogeneous capacity Q.

This model has several decision variables. Some of them are binary variables. $x_{ijk} = 1$ if vehicle $k(k \in K)$ travels from node $i(i \in N)$ to $j(j \in N)$. $h_i = 1$ if the demand of customer $i(i \in N_C)$ is delivered to his/her home. $l_i = 1$ if the demand of customer $i(i \in N_C)$ is delivered to a parcel locker. $y_{ij} = 1$ if the demand of customer $i(i \in N_C)$ is assigned to parcel locker station $j(j \in N_{PL})$. The remaining decision variables are defined as continuous variables. μ_{ik} is the time vehicle $k(k \in K)$ starts to service customer $i(i \in N_{CPL})$. ∂_k is the time vehicle $k(k \in K)$ leaves the depot. ν_k is the time vehicle $k(k \in K)$ visits the last customer. ψ_{jk} is the number of packages delivered to parcel locker station $j(j \in N_{PL})$ by vehicle $k(k \in K)$. M is an arbitrary big number.

The problem formulation is as follows.

Minimize

$$Z = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \tag{1}$$

Subject to:

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in N_{HC} \tag{2}$$

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} \leq h_i, \quad \forall i \in N_{HLC} \tag{3}$$

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 0, \quad \forall i \in N_{LC} \tag{4}$$

$$\sum_{j \in N_C} x_{0jk} \leq 1, \quad \forall k \in K \tag{5}$$

$$\sum_{\substack{i \in N \\ i \neq j}} x_{ijk} - \sum_{\substack{i \in N \\ j \neq i}} x_{jik} = 0, \quad \forall k \in K,\ j \in N \tag{6}$$

$$\sum_{i \in N_C} x_{i0k} \leq 1, \quad \forall k \in K \tag{7}$$

$$\sum_{i \in N} \sum_{\substack{j \in N_C \\ i \neq j}} d_j x_{ijk} + \sum_{p \in N} \sum_{\substack{m \in N_{PL} \\ p \neq m}} \psi_{mk} x_{pmk} \leq Q, \quad \forall k \in K \tag{8}$$

$$\partial_k + t_{0i} - \mu_{ik} \leq M(1 - x_{0ik}), \quad \forall k \in K,\ i \in N_{CPL} \tag{9}$$

$$\mu_{ik} + s_i + t_{ij} - \mu_{jk} \leq M(1 - x_{ijk}), \quad \forall k \in K, \quad i \in N_{CPL}, \quad j \in N_{CPL} \tag{10}$$

$$\mu_{ik} - \nu_k \leq M(1 - x_{i0k}), \quad \forall k \in K,\ i \in N_{CPL} \tag{11}$$

$$a_i \times h_i \leq \sum_{k \in K} \mu_{ik} \times h_i \leq b_i \times h_i, \quad \forall k \in K,\ i \in N_C \tag{12}$$

$$h_i + l_i = 1, \quad \forall i \in N_C \tag{13}$$

$$h_i = 1, \quad \forall i \in N_{HC} \tag{14}$$

$$l_i = 1, \quad \forall i \in N_{LC} \tag{15}$$

$$\sum_{j \in N_{PL}} y_{ij} = l_i, \quad \forall i \in N_C \tag{16}$$

$$y_{ij} \leq \ell_{ij}, \quad \forall i \in N_C,\ j \in N_{PL} \tag{17}$$

$$\sum_{k \in K} \psi_{jk} = \sum_{i \in N_{LC} \cup N_{HLC}} y_{ij} \times d_i, \quad \forall j \in N_{PL} \tag{18}$$

$$\sum_{i \in N_C} y_{ij} \leq \xi_j, \quad \forall j \in N_{PL} \tag{19}$$

$$\sum_{i \in N} x_{ijk} \times M \geq \psi_{jk}, \quad \forall j \in N_{PL},\ k \in K \tag{20}$$

$$\sum_{k \in K} \sum_{i \in N} x_{ijk} \geq h_j, \quad \forall j \in N_C \tag{21}$$

$$x_{iik} \geq 0, \quad \forall i \in N,\ k \in K \tag{22}$$

$$y_{ik} \geq 0,\ \forall i \in N,\ k \in K \tag{23}$$

$$h_i \in \{0, 1\}, \quad \forall i \in N_C \tag{24}$$

$$l_i \in \{0, 1\}, \quad \forall i \in N_C \tag{25}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i \in N_C,\ j \in N_{PL} \tag{26}$$

$$\psi_{jk} \geq 0, \quad \forall j \in N_{PL},\ k \in K \tag{27}$$

$$\mu_{ik} > 0, \quad \forall i \in N,\ k \in K \tag{28}$$

$$\partial_k \geq 0, \quad \forall k \in K \tag{29}$$

$$\nu_k \geq 0, \quad \forall k \in K \tag{30}$$

The objective function of VRPPL (1) is to minimize the total distance traveled by all vehicles. Constraints (2) to (4) define customer deliveries based on their delivery options. If a customer belongs to home delivery, a vehicle needs to visit the customer node. If a customer belongs to locker delivery, a vehicle must not visit the customer node. If a customer belongs to home and locker delivery, a vehicle could either visit the customer or not. Constraints (5) to (7) ensure the flow balance through the network.

Constraints (8) guarantee that a vehicle's total load does not exceed the vehicle's capacity. The total load is defined by the packages delivered to the customers' home location and the selected parcel lockers. Constraints (9) to (11) track the arrival time at the visited nodes. Constraints (12) ensure that it has to be delivered within the predefined time windows if a delivery belongs to home delivery. Constraints (13) – (15) guarantee that each customer's delivery is either home delivery or locker delivery. Constraints (16) ensure that if the locker delivery option is assigned to a particular customer, a delivery to any available parcel locker stations should be made by any vehicle to fulfill the customer's demand. Constraints (17) ensure that the selection of parcel locker stations for a locker delivery is based on the customer's preferences. Constraints (18) guarantee that the total load assigned to a parcel locker equals the total demand of each customer assigned to the parcel locker. Constraints (19) define the capacity of a parcel locker. Constraints (20) guarantee that a parcel locker station assigned to customers' demands should be visited by a vehicle. Constraints (21) ensure that a customer's location needs to be visited if the chosen delivery option is home delivery. Constraints (22) to (30) define the range of variables.

0	20	8	6	25	3	23	22	24	10	4	1	5	21	7	15	11	13	19	16	14	17	18	9	2	12	0
0	20	27	27	27	27	27	27	27	27	27	27	27	21	7	15	11	13	19	16	26	26	26	26	26	12	0

FIGURE 1. An example of solution representation.

IV. SIMULATED ANNEALING FOR VRPPL

Section 4 explains the solution representation of the problem, the initial solution, the neighborhood moves, and the developed SA algorithm.

VRPPL is a special case of VRP, and VRP itself is an NP-hard problem [39], therefore, VRPPL is also an NP-hard problem. We considered using SA because the algorithm is one of the promising alternatives to deal with this class of problems and has proven to furnish good results for various types of VRPs, including capacitated vehicle routing problem [40], VRPTW[41], hybrid vehicle routing problem (hybrid-VRP) [8], time-dependent vehicle routing problem [42], green vehicle routing problem [43], and VRP with pickup delivery, providing evidence that SA can provide comparable results.

SA algorithm operates by following a physics process, of cooling or annealing. Annealing is the process of producing better-aligned metal with low energy-state crystallization by slowing the cooling down temperature process [34]. The optimization procedure of SA searches for a (near) global minimum, imitating the slow cooling process in the physical annealing process. The procedure starts with an initial solution. At each iteration, a new solution is taken from the predefined neighborhoods of the current solution. This objective function value of the new solution is compared with the current best solution to evaluate whether an improvement has been achieved. If the new solution's objective function value is better (smaller) in minimization, the new solution becomes the current solution. The search continues by processing with a new iteration referred to the new solution. A new solution that is worse with a degraded (larger) objective function value may also be accepted as the new current solution. This computational purpose is not to restrict the search to those solutions that decrease the objective function value and allow moves that increase the objective function value. This mechanism will avoid the solution of getting stuck in the local optimum [34].

A. SOLUTION REPRESENTATION

A two-dimensional array describes a solution of VRPPL. Specifically, the array has two rows where a permutation of n customers $(1, 2, \ldots n)$ is placed in the first row and the chosen delivery option of each customer is placed in the second row. At the beginning and the end of the first row, 0 value is placed, representing a depot instance. The second row consists of customer nodes and parcel locker nodes. If a customer node located in a particular index of the first row is assigned to home delivery, the customer node itself is placed in the related cell of the second row. Otherwise, a selected parcel locker is placed in the second row. Figure 1 represents a solution from a C101 instance with 25 customers.

The objective value is calculated based on the second row of the solution representation representing the vehicle's actual visited nodes. The second row of a solution representation is built by using the following rule. If a customer currently being evaluated is a home-delivery type, the vehicle needs to visit the customer's location. If the customer belongs to the LC, the vehicle selects the nearest feasible parcel locker station to visit. If the customer belongs to home or locker-delivery type, a parameter p is utilized to determine whether the customer is served at his/ her home or assigned to a parcel locker. The p is a constant parameter and ranges from 0 to 1. In this case, a random number between 0 and 1 is generated. If the generated value is between 0 and p, the customer is served at his/her home. Otherwise, the customer is assigned to the selected parcel locker and the way of selecting the parcel locker station is the same with the type-2 customers. In this way, the algorithm explores a wider solution space by allowing random selection of a delivery option for type-3 customers. This mechanism provides a higher probability of achieving good results for VRPPL.

An infeasible solution is not permitted in this algorithm. In the case of time windows and/or capacity violation while visiting a node, the current vehicle needs to return to the depot without visiting the node. A new vehicle is assigned to visit the node and the succeeding nodes based on the provided sequence in the first row of solution representation. If the total number of utilized vehicles exceeds the maximum number of vehicles, the solution is deemed as an infeasible solution, and its objective value is set to a considerably significant number.

Figure 2 describes solution representation. As depicted in Figure 2, three routes are needed to fulfill the demand of all customers. The first route starts from customer 20 and then parcel locker with an index of 27 and finally visits customer 21 before returning to the depot. The real first route is obtained from the second row of the solution representation example; Figure 1. 27 presents a selected parcel locker station where the demand of several customers (8, 6, 25, 3, 23, 22, 24, 10, 4, 1, and 5) is assigned. The second route contains customers 7, 15, and 11. All three nodes belong to customer nodes, indicating that all deliveries made by the second vehicle are home deliveries. The third route consists of customers 13, 19, 16, and 12, sequentially. Finally, a parcel

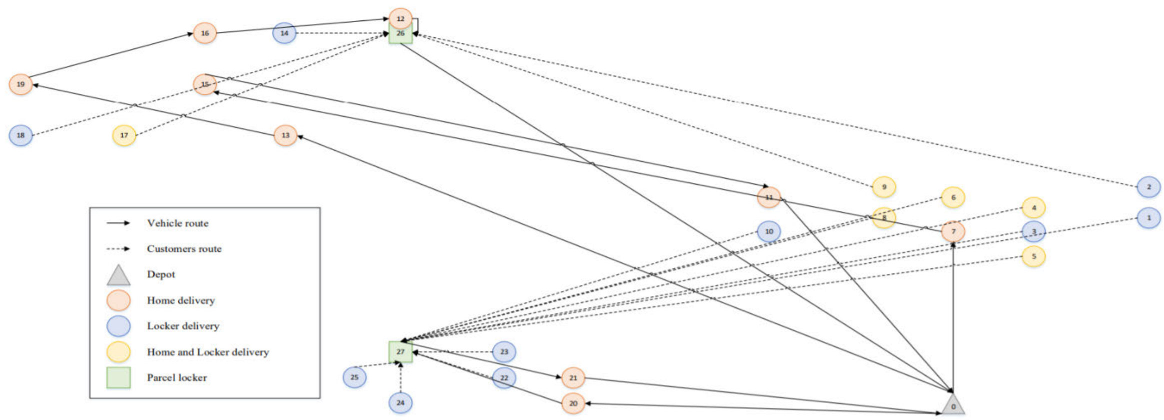

FIGURE 2. An illustration of solution representation.

Original Solution	0	20	8	6	25	3	23	22	24	10	4	1	5	21	7	15	11	13	19	16	14	17	18	9	2	12	0
By Swap	0	20	8	6	25	3	23	22	11	10	4	1	5	21	7	15	24	13	19	16	14	17	18	9	2	12	0
By Insertion	0	20	8	6	25	3	23	22	11	24	10	4	1	5	21	7	15	13	19	16	14	17	18	9	2	12	0
By inversion	0	20	8	6	25	3	23	22	11	15	7	21	5	1	4	10	24	13	19	16	14	17	18	9	2	12	0

FIGURE 3. An example of neighborhood solutions.

locker with an index of 26 is visited before returning to the depot. The assigned customers to the parcel locker are customers 14, 17, 18, 9, and 2.

B. THE INITIAL SOLUTION

The procedure for building an initial solution consists of two steps. First, all the customers are listed in an ascending order based on their indexes. Each customer in the list is assigned to a particular delivery based on the type of customer. If a customer belongs to the home-delivery type, the customer is assigned to home delivery. If a customer belongs to the locker-delivery type, the customer is assigned to the selected parcel locker station. If a customer belongs to the home and locker-delivery type, the customer is also assigned to the selected parcel locker station.

After the first step is done, the nearest neighborhood algorithm is then performed. The nearest algorithm then selects the closest node according to the currently visited node among the unassigned nodes. This algorithm is terminated if all customer nodes have been assigned.

C. NEIGHBORHOOD MOVES

To find a solution with a better objective value, we employ three neighborhood moves and apply them to the first row of the solution representation: (1) swap, (2) insertion, and (3) inversion. The currently evaluated solution is designated as X and $N(X)$ is the set of its neighborhood solutions. A new solution Y is chosen from $N(X)$ at each iteration by using one of the three moves, as mentioned above. The swap move exchanges two randomly chosen nodes in X. The insertion move selects two nodes randomly and then removes one of the nodes and inserts it to another selected node position. The inversion move randomly selects two nodes and reverses the sequence between them (including the chosen two nodes). Figure 3 illustrates the results after applying each of the moves.

D. SIMULATED ANNEALING PROCEDURE

The SA is performed after an initial solution is created. In the beginning, the current temperature T is set to the initial temperature, T_0. The two types of iterations in SA are inner and outer iterations. The inner iteration focuses on finding the new solution based on the current solution utilizing neighborhood moves. In contrast, the outer iteration exists to control the temperature of SA. Two types of solutions are kept during the iterations of SA, including $\sigma_{current}$ which represents the current solution and σ_{best} which represents the best-found solution. Initially, $\sigma_{current}$ is obtained from the initial solution and σ_{best} is set to $\sigma_{current}$. A new solution σ_{new}, is generated from one of the three neighborhood moves explained in Section IV.C. The selection of a move is based on the randomly generated value, r_1, ranging from 0 to 1. If r_1 is less than 1/3, the swap is selected. If r_1 is between

SA procedure (I_{iter}, T_0, $N_{non\text{-}improving}$, and α)

Step 1:
Generating the initial solution, σ_{init}

Step 2:
Let $T = T_0$; $N = 0$; $\sigma_{best} = \sigma_{current} = \sigma_{init}$;
$R = 0$; FoundBestSol = false;

Step 3:
$N = N + 1$;

Step 3.1:
Generating σ_{new} based on $\sigma_{current}$;
Generate $r_1 \sim U(0,1)$
If $r_1 \leq 1/3$
Generate σ_{new} by swap;
Else If $(1/3 < r_1 \leq 2/3)$
Generate σ_{new} by insertion;
Else
Generate σ_{new} by inversion;

Step 3.2:
If $\Delta = f(\sigma_{new}) - f(\sigma_{current}) \leq 0$ $\{\sigma_{current} = \sigma_{new};\}$
Else
Generate $r_2 \sim U(0,1)$;
If $r_2 < e^{-(\Delta/\beta T)}$ $\{\sigma_{current} = \sigma_{new};\}$

Step 3.3:
If $f(\sigma_{new}) < f(\sigma_{best})$ {
$\sigma_{best} = \sigma_{new}$; $R = 0$; FoundBestSol = true ;
}

Step 4:
If $N = I_{iter}$ {
$T = T \times \alpha$; $N = 0$;
If (FoundBestSol is false) {$R = R+1$; FoundBestSol = true;}
}
Else {Go to Step 3;}

Step 5:
If $R = N_{non\text{-}improving}$ or $T \leq T_f$ {Terminate the SA procedure}
Else {Go to Step 3;}

FIGURE 4. The pseudocode of the proposed SA.

$1/3$ and $2/3$, the insertion is selected, otherwise, the inversion is chosen.

After σ_{new} is generated, the objective value of σ_{new} is calculated. Let $f(\sigma)$ represent the objective value of a solution σ. To determine whether the σ_{new} is accepted as the new $\sigma_{current}$, a comparison between $f(\sigma_{new})$ and $f(\sigma_{current})$ is necessary. Consequently, Δ is defined as $f(\sigma_{new}) - f(\sigma_{current})$. Since the objective of VRPPL is to minimize the traveled distance, σ_{new} is indeed accepted when it has a lower objective value which means $\Delta < 0$. Moreover, if $f(\sigma_{new}) < f(\sigma_{best})$, σ_{new} is accepted as new σ_{best}. If σ_{new} results in a higher objective value, it is not directly be rejected. A random value, r_2, with a range from 0 to 1 is generated. The new neighborhood solution is accepted with a probability $e^{-(\Delta/\beta T)}$, where β is Boltzmann constant. If $r_2 < e^{-(\Delta/\beta T)}$, σ_{new} is accepted as a new $\sigma_{current}$. Otherwise, σ_{new} is rejected.

After an inner iteration cycle is completed, T is reduced by multiplying with a constant α. Before another inner iteration cycle starts, it is necessary to check whether the termination condition has been met. The termination of the SA procedure depends on the defined final temperature, T_f, and the defined number of non-improving iteration, N. If either of these two criteria is reached, the SA is terminated. Otherwise, a new inner iteration cycle is performed. The pseudocode of the proposed SA is shown in Figure 4.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed SA algorithm was implemented in C++. The experiments were conducted on a computer with an Intel Core 3.40 GHz CPU. The mathematical programming model for VRPPL was solved using a Gurobi solver on the same machine.

TABLE 1. Tested parameters value for SA using OFAT experiment.

I_{iter}	T_0	T_f	$N_{non\text{-}improving}$	α	β
600L	10	**0.1**	**50**	0.97	**1**
800L	**20**	**0.05**	**100**	**0.98**	**1/3**
1000L	**30**	0.01	150	**0.99**	1/5

Bold values denote the selected upper and lower bound values of each parameter for $2^{£}$ factorial design.

A. TEST INSTANCES

To assess the performance of the proposed SA algorithm, we randomly generated a set of VRPPL instances based on Solomon's benchmark problems [14]. We generated small, medium, and large datasets. Each dataset contains 56 testing instances. The original problem contains various types of information, i.e., number of available vehicles, number of depots and customers, the coordinates, the time window, and service time of each node.

Several modifications are applied to the original instance. First, each customer is assigned as the type-1, type-2, or type-3 customer. We randomly assign each customer to one of these three options with 1/3 probability of each. The number of parcel lockers is the number of customers divided by 20 and the obtained value is rounded up to the nearest integer. The locations of parcel lockers are randomly generated integer numbers derived from a range between the smallest and the largest value of x- and y-coordinates of all customer nodes. The time window of each parcel locker station is set the same as that of the depot. The service time is set to half of the customer's service time. The type-2 and type-3 customers are assumed to choose the nearest parcel locker station. The test instances for VRPPL can be downloaded from http://web.ntust.edu.tw/∼vincent/vrppl.

B. TUNING PARAMETER

We utilize $2^{£}$ factorial design to select the best combination of SA parameters. The selected testing instances include C101, C201, R101, R201, RC101, and RC201. To execute the $2^{£}$ factorial design, we determine each SA parameter's upper and lower bound values using a pilot experiment. One Factor at A Time (OFAT) was then chosen as the experiment method for performing the pilot experiment. Five replications are utilized for each combination of OFAT. Table 1 presents the tested values of each SA parameter in the pilot experiment. The bold-type values are the selected upper and lower bound values of each SA parameter.

After conducting the pilot experiment, the $2^{£}$ factorial design, where £= 6 factors, is performed using the selected upper and lower-bound values. In total, there are 64 tested combinations. The best parameter combination selection is based on the lowest average objective value obtained from solving selected testing instances. By using this selection criterion, we obtained $I_{iter} = 1000L$, $T_0 = 30$, $T_f = 0.05$, $\alpha = 0.99$, $N_{non-improving} = 100$, and $\beta = 1$.

We also perform an experiment to determine the value of parameter p. The candidate values of parameter p are 0.1, 0.3, 0.5, 0.7, and 0.9. We choose the value for parameter p which has the smallest average gap to the minimum objective values obtained in the experiments. Based on Table 2, for instances with 25 customers, the chosen p parameter value is 0.5. For instances with 50 customers, the p parameter value is 0.3, and lastly for instances with 100 customers, the p parameter value is 0.1.

C. COMPUTATIONAL RESULTS

In this section, we present the performance evaluation of our proposed SA in solving both VRPTW and VRPPL instances. In particular, we compare the result obtained by SA with those by Zhang *et al.* [44] and Moradi *et al.* [19] in solving the VRPTW instances. Since VRPPL is a new problem that has never been dealt with in any previous work, the results obtained by SA are compared with the results obtained by the Gurobi solver on the same machine, which is used to solve the mathematical model of VRPPL. These types of comparisons have been widely adopted in the VRP domain [8], [34], [40].

1) FIRST DATASET – VRPTW DATASET

The proposed SA is tested on the original VRPTW benchmark instances. Two datasets, small and large, are mainly tested, consisting of 25 customer nodes and 100 customer nodes. The parameters employed by the SA were obtained from the parameter tuning process explained in section 5.B and five replications were performed for evaluation. The results are summarized in Tables 3 and 4.

We compared the results obtained by our developed SA with the optimal solutions provided in http://web.cba.neu.edu/∼msolomon/problems.htm for small datasets. In Table 3, the best-known solution (BKS) represents optimal solutions for C1-25, C2-25, R1-25, R2-25, RC1-25, and RC2-25 datasets. Two types of measurement are collected from the results of the SA, i.e., the averaged mean solutions (avg. mean) and the averaged best solutions (Avg. Best). The avg. mean solution shows the grand average of total distance values over all replications of all instances for a particular dataset while avg. best only takes the average of the best-obtained total distance value. Based on Table 3, the SA obtains all optimal solutions for the C2-25 dataset. For the other datasets, i.e., C1-25, R1-25, R2-25, R2-25, and RC1-25, the average gap values are less than 0.26%. The largest gap is 0.76%, resulted from solving instances in the RC2-25 dataset.

TABLE 2. Experimental results of determining parameter p.

p	Average						Gap						Avg. Gap
	C101	C201	R101	R201	RC101	RC201	C101	C201	R101	R201	RC101	RC201	
25 customers													
0.1	199.8	195.5	178.6	311.5	332.6	270.2	0.00%	0.00%	0.00%	2.23%	0.00%	0.00%	0.37%
0.3	199.8	195.5	178.6	308.1	332.6	271.48	0.00%	0.00%	0.00%	1.12%	0.00%	0.47%	0.26%
0.5	**199.86**	**195.5**	**178.6**	**304.7**	**332.6**	**272.12**	**0.03%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.71%**	**0.12%**
0.7	199.8	195.5	178.6	304.7	332.6	273.74	0.00%	0.00%	0.00%	0.00%	0.00%	1.31%	0.22%
0.9	199.8	195.5	178.6	308.1	332.6	271.46	0.00%	0.00%	0.00%	1.12%	0.00%	0.47%	0.26%
Minimum	199.8	195.5	178.6	304.7	332.6	270.2							0.12%
50 customers													
0.1	374.2	289.76	454.14	409.2	627.7	538.5	0.80%	2.12%	0.00%	0.00%	0.00%	0.00%	0.49%
0.3	**371.24**	**283.74**	**455.36**	**413.14**	**628.68**	**539.14**	**0.00%**	**0.00%**	**0.27%**	**0.96%**	**0.16%**	**0.12%**	**0.25%**
0.5	374.02	287.24	456.18	411.06	628.32	539.14	0.75%	1.23%	0.45%	0.45%	0.10%	0.12%	0.52%
0.7	373.7	286.54	455.42	413.14	629.82	538.5	0.66%	0.99%	0.28%	0.96%	0.34%	0.00%	0.54%
0.9	375.2	285	458.6	414.78	628.32	538.5	1.07%	0.44%	0.98%	1.36%	0.10%	0.00%	0.66%
Minimum	371.24	283.74	454.14	409.2	627.7	538.5							0.25%
100 customers													
0.1	**849.26**	**571.82**	**776.9**	**707.64**	**1029.2**	**729.58**	**0.30%**	**0.00%**	**0.16%**	**1.16%**	**0.16%**	**0.75%**	**0.42%**
0.3	855.42	599.74	776.28	705.26	1037.24	731.04	1.02%	4.88%	0.08%	0.82%	0.94%	0.95%	1.45%
0.5	857.88	574.08	789.44	699.54	1029.14	724.16	1.31%	0.40%	1.77%	0.00%	0.15%	0.00%	0.61%
0.7	854.34	592.32	777.58	708.82	1032.66	730.04	0.90%	3.59%	0.24%	1.33%	0.50%	0.81%	1.23%
0.9	846.76	585.86	775.68	705.14	1027.56	734.68	0.00%	2.46%	0.00%	0.80%	0.00%	1.45%	0.78%
Minimum	846.76	571.82	775.68	699.54	1027.56	724.16							0.42%

Bold values denote the selected parameter value for p

$$\text{Avg. Gap} = \frac{(Objective\ value - Minimum\ Objective\ value)}{Minimum\ Objective\ value} \times 100\%$$

For large datasets of VRPTW, the comparison of the results is presented in Table 4. In this table, the BKS column represents the best solutions among solutions provided in http://web.cba.neu.edu/∼msolomon/heuristi.htm, by Zhang *et al.* [44], and Moradi *et al.* [19]. Columns representing Zhang et al[44] and Moradi *et al.* [19] contain the average of the best solutions. The performance of SA is measured in terms of gap percentage which is defined by the following formula:

$$Gap^a(\%) = \frac{\text{SA's Avg.Best-BKS}}{\text{BKS}} \times 100\%$$

$$Gap^b(\%) = \frac{\text{SA's Avg.Best-Avg.Best in Zhang et.al. [41]}}{\text{Avg.Best in Zhang et.al. [41]}} \times 100\%$$

$$Gap^c(\%) = \frac{\text{SA's Avg.Best-Avg.Best in Moradi et.al. [16]}}{\text{Avg.Best in Moradi et.al. [16]}} \times 100\%$$

Based on Table 4, the smallest gap between the SA's average values of best solutions and the BKS is obtained from solving the C2-100 dataset, i.e., 0.11%, while the largest gap reaches 4.29% from solving R1-100 dataset. The SA can improve the results of state-of-the-art algorithms in R2-100 with an average gap of 0.34%. In more detail, the proposed SA could outperform that by Zhang *et al.* [44] in three instance sets, i.e., C2-100, R2-100, and RC1-100 with the largest gap of -0.78%. However, the algorithm by Zhang *et al.* [44] outperformed the proposed SA for the remaining three instance sets with the largest gap of 0.66%. On average, the SA outperformed the solution from Zhang *et al.* [89] by 0.07%. The proposed SA outperforms the solution from Moradi et al [19] in three instance sets, i.e., R2-100, RC1-100, and RC2-100 with the largest gap of 7.87%, while the algorithm by Moradi *et al.* [19] outperforms SA with the largest gap of 4.29% for solving the R1-100 dataset. The reported results in Table 4 show that the proposed

TABLE 3. Averaged results of small datasets of VRPTW.

Dataset		BKS	SA		
			Avg. Mean	Avg. Best	Gap[a] (%)
C1-25	NV	3.00	3.07	3.00	
	TD	190.59	192.92	191.09	0.26
C2-25	NV	2.00	2.00	2.00	
	TD	214.45	215.29	214.45	0.00
R1-25	NV	4.92	5.08	5.08	
	TD	463.37	464.44	464.44	0.23
R2-25	NV	2.73	2.49	2.73	
	TD	382.15	384.60	383.15	0.26
RC1-25	NV	3.25	3.28	3.25	
	TD	350.24	351.10	351.10	0.25
RC2-25	NV	2.88	2.60	2.75	
	TD	319.28	324.82	321.70	0.76

$\text{Gap}^{a} = \frac{(Avg.Best - BKS)}{BKS} \times 100\%$

TABLE 4. Averaged results of large datasets of VRPTW.

Dataset		BKS	Zhang et al. [41]	Moradi et al. [16]	SA				
					Avg. Mean	Avg. Best	Gap[a](%)	Gap[b] (%)	Gap[c] (%)
C1-100	NV	10.00	10.00	10.00	10.00	10.00			
	TD	826.70	828.38	826.70	828.38	828.38	0.20	0.00	0.20
C2-100	NV	3.00	3.00	3.00	3.03	3.00			
	TD	589.21	590.40	589.21	590.80	589.86	0.11	-0.09	0.11
R1-100	NV	11.76	13.50	11.76	13.67	13.50			
	TD	1140.44	1187.90	1140.44	1197.28	1189.33	4.29	0.12	4.29
R2-100	NV	4.73	4.73	4.76	4.31	4.36			
	TD	891.24	891.24	912.45	896.72	888.19	-0.34	-0.34	-2.66
RC1-100	NV	12.38	13.25	12.47	13.28	13.25			
	TD	1336.45	1361.08	1386.78	1361.83	1350.40	1.04	-0.78	-2.62
RC2-100	NV	5.50	5.50	4.90	5.18	5.13			
	TD	1017.47	1017.47	1029.78	1035.51	1024.16	0.66	0.66	-0.55
Average							**0.99**	**-0.07**	**-0.20**

$\text{Gap}^{b} = \frac{(SA'sAvg.Best - Zhang\ et\ al.[41])}{Zhang\ et\ al.[41]} \times 100\%$

$\text{Gap}^{c} = \frac{(SA'sAvg.Best - Moradi\ et\ al.[16])}{Moradi\ et\ al.[16]} \times 100\%$

SA results in comparable solutions with the state-of-the-art algorithms.

2) SECOND DATASET – VRPPL DATASET

VRPPL has never been studied in earlier. Therefore, the Gurobi solver is utilized to obtain solutions from solving the testing instances and the obtained results are compared with results obtained by the proposed SA. The performance of SA is measured in terms of gap percentage which is defined by the following formula:

$$Gap(\%) = \frac{\text{SA's best objective value-Gurobi's objective value}}{\text{Gurobi's objective value}} \times 100\%$$

TABLE 5. Result of small datasets of VRPPL.

Problem	Gurobi			SA				Gap[d]	Gap[e]
	NV	Distance	Time (s)	NV	Distance (Best)	Distance (Avg)	Time (s)		
C101	3	199.8	1310.7	3	199.8	199.8	7.6	0.00%	0.00%
C102	3	143.5	18000	3	143.5	155.08	19.1	0.00%	-7.47%
C103	3	224.6	18000	3	203.9	203.90	10.4	-9.22%	0.00%
C104	3	182	18000	3	181.8	181.80	11.6	-0.11%	0.00%
C105	3	210.6	18000	3	206.9	206.90	9.1	-1.76%	0.00%
C106	3	212.4	18000	3	208.8	208.80	7.5	-1.69%	0.00%
C107	3	210.6	18000	3	206.9	206.90	8.9	-1.76%	0.00%
C108	3	158.7	18000	3	155.3	159.38	8.6	-2.14%	-2.56%
C109	3	160.5	18000	3	141.5	143.46	11.4	-11.84%	-1.37%
C201	2	195.5	188.6	2	195.5	195.50	5.4	0.00%	0.00%
C202	1	119.6	195.2	1	119.6	119.60	8.8	0.00%	0.00%
C203	1	146.4	18000	1	146.4	146.40	9.5	0.00%	0.00%
C204	1	150.6	18000	1	145.9	145.90	6.5	-3.12%	0.00%
C205	2	168.2	18000	2	168.2	168.20	8.2	0.00%	0.00%
C206	2	171.3	18000	2	171.3	171.30	7.4	0.00%	0.00%
C207	1	121.3	18000	1	121.3	121.30	11.9	0.00%	0.00%
C208	1	162.2	18000	1	162.2	162.20	7.4	0.00%	0.00%
R101	2	178.6	108.9	2	178.6	178.600	6.4	0.00%	0.00%
R102	2	187.8	18000	2	187.8	187.80	8.1	0.00%	0.00%
R103	2	197.6	18000	2	197.6	197.60	4.8	0.00%	0.00%
R104	3	238.9	18000	3	238.9	238.94	9.0	0.00%	-0.02%
R105	2	194.1	164.9	2	194.1	194.10	6.7	0.00%	0.00%
R106	2	177.0	18000	2	177.0	177.00	5.8	0.00%	0.00%
R107	2	238.9	18000	3	238.9	238.90	8.7	0.00%	0.00%
R108	3	257.9	18000	2	250.8	251.28	10.5	-2.75%	-0.19%
R109	3	260.7	18000	3	260.7	260.70	7.9	0.00%	0.00%
R110	3	239.2	18000	3	239.2	239.20	9.4	0.00%	0.00%
R111	2	245.3	18000	2	245.3	245.30	5.8	0.00%	0.00%
R112	2	185.6	18000	2	185.6	185.60	6.6	0.00%	0.00%
R201	3	304.7	653.3	3	304.7	304.70	5.6	0.00%	0.00%
R202	2	257.8	18000	2	257.8	265.32	8.6	0.00%	-2.83%
R203	2	185.6	18000	2	185.6	185.60	7.0	0.00%	0.00%
R204	1	154.2	922.3	1	154.2	154.20	6.3	0.00%	0.00%
R205	1	219.1	18000	1	219.1	225.52	10.0	0.00%	-2.85%
R206	1	195.8	7773.5	1	195.8	195.80	7.9	0.00%	0.00%
R207	1	197.9	18000	1	197.9	197.96	7.8	0.00%	-0.03%
R208	1	163.8	18000	1	163.8	163.80	5.9	0.00%	0.00%
R209	1	232.6	2126.8	1	232.6	232.60	8.1	0.00%	0.00%
R210	2	279.1	18000	2	279.1	279.18	6.3	0.00%	-0.03%
R211	1	131.5	514.0	1	131.5	131.50	6.4	0.00%	0.00%
RC101	4	367.7	18000	3	332.6	332.60	5.4	-9.55%	0.00%

TABLE 5. ***(Continued.)*** **Result of small datasets of VRPPL.**

RC102	4	313.8	18000	3	297.9	297.90	9.1	-5.07%	0.00%
RC103	4	306.4	18000	3	243.8	243.80	6.2	-20.43%	0.00%
RC104	3	329.1	18000	3	309.4	309.40	12.6	-5.99%	0.00%
RC105	4	387.7	18000	3	309.7	309.70	4.7	-20.12%	0.00%
RC106	4	279.1	18000	4	279.1	279.10	7.8	0.00%	0.00%
RC107	3	310.9	18000	3	299.9	300.56	7.9	-3.54%	-0.22%
RC108	4	298.7	18000	3	275.7	275.70	6.4	-7.70%	0.00%
RC201	2	270.2	18000	2	270.2	270.56	8.1	0.00%	-0.13%
RC202	3	293.9	18000	3	293.9	294.62	13.1	0.00%	-0.24%
RC203	1	208.0	18000	1	208.0	208.00	5.8	0.00%	0.00%
RC204	1	175.7	18000	1	175.7	175.70	6.0	0.00%	0.00%
RC205	3	283.2	18000	3	283.2	284.48	5.6	0.00%	-0.45%
RC206	2	246.7	18000	2	246.7	247.40	11.4	0.00%	-0.28%
RC207	2	250.5	18000	2	250.5	250.50	8.1	0.00%	0.00%
RC208	1	182.3	18000	1	182.3	182.30	6.1	0.00%	0.00%
Average			15034.967				8.1	-1.91%	-0.33%

$$\text{Gap}^{d} = \frac{(SA\ Distance(Best) - Gurobi\ Distance)}{Gurobi\ Distance} \times 100\%$$

$$\text{Gap}^{e} = \frac{(SA\ Distance(Best) - SA\ \ Distance(Avg))}{SA\ \ Distance(Avg)} \times 100\%$$

The solver was terminated after 5 hours or when it has found an optimal solution. Tables 5, 6, and 7 present the results for small, medium, and large datasets of VRPPL comprising of 25, 50, and 100 customer nodes, consecutively. Among the instances of a small dataset, Gurobi could only provide 10 optimal solutions successfully obtained by the developed SA. The proposed SA improves the results of Gurobi for instances in small datasets with the largest improvement of 20.43% for RC103. SA on average improved the Gurobi's results by 1.90%.

For the medium dataset, the proposed SA improves Gurobi's results with the largest improvement of 39.22% for C204. Based on Table 6, the SA improves most of the results of Gurobi while there are some instances in which Gurobi performs better in terms of solution quality. On average, SA improves the Gurobi's results by 4.82%.

For the large dataset, the proposed SA improves Gurobi's results with the largest improvement of 66.09% for R208. The average gap of -27.68% means that SA on average improves Gurobi's results. Based on Table 5, VI, and VII, we conclude that on average, SA provides better solutions in terms of objective value and computational time.

Furthermore, we investigated the gap between the best solution and the average solution obtained by SA for each instance. The average gaps are 0.33%, 0.92%, and 2.21% for VRPPL small-, medium-, and large-instances, respectively. Therefore, we conclude that the proposed SA is reasonably robust for solving the VRPPL.

D. ANALYSIS AND DISCUSSION

In this section, we have discussed the proposed methods to deal with the VRPPL.

In this paper, we proposed a mathematical programming model for the VRPPL. Besides the VRPPL, the mathematical model can also solve another VRP variant, i.e., VRPTW. To implement the mathematical model for solving the VRPTW, we need some adjustments because VRPTW only considers one type of customer, i.e., type-1 customers. Thus, we need to omit type-2 and type-3 customer-related sets, parameters, decision variables, and constraints so that we may convert the mathematical model to solve VRPPTW.

The mathematical model is solved using an exact solver, Gurobi. The exact solver mainly aims to obtain optimal solutions with one primary issue, i.e., the significant computational time. As the size of instances grows larger, the complexity also grows and limits the exact solver to obtain even feasible solutions for several instances, as shown in Table 7. This situation becomes an important issue the approach is to be implemented into a real application. Therefore, an exact approach will not be enough but the results can be a baseline for evaluating the solutions obtained from applying the proposed heuristic, i.e., SA.

The limitation of the exact solver leads to the development of a heuristic, i.e., the SA. According to the experiments presented in Tables 5 - 7, it can provide high-quality solutions in a reasonable computational time when being compared to the results of the exact solver. However, we are aware

TABLE 6. Result of medium datasets of VRPPL.

Problem	Gurobi			SA				Gap[d]	Gap[e]
	NV	Distance	Time (s)	NV	Distance (Best)	Distance (Avg)	Time (s)		
C101	5	367.9	18000	5	368.5	371.24	206.3	0.16%	-0.74%
C102	5	353.2	18000	5	357.7	359.08	184.1	1.27%	-0.38%
C103	5	308.5	18000	5	317.9	323.92	228.9	3.05%	-1.86%
C104	5	404.8	18000	5	344.9	348.24	233.7	-14.80%	-0.96%
C105	5	350.7	18000	5	374.9	376.74	205.1	6.90%	-0.49%
C106	5	321.9	18000	5	321.9	337.06	234.5	0.00%	-4.50%
C107	5	329.3	18000	5	327.3	330.56	181.5	-0.61%	-0.99%
C108	5	338.9	18000	5	338.8	339.76	222.8	-0.03%	-0.28%
C109	5	370.1	18000	5	335.4	341.38	252.9	-9.38%	-1.75%
C201	2	282.2	18000	2	282.2	283.74	173.8	0.00%	-0.54%
C202	2	227.6	18000	2	233.5	242.02	188.1	2.59%	-3.52%
C203	2	257.8	18000	2	264.5	270.70	123.9	2.60%	-2.29%
C204	3	451.5	18000	2	274.4	277.14	150.1	-39.22%	-0.99%
C205	2	266.1	13771.7	2	267.0	268.00	148.0	0.34%	-0.37%
C206	3	266.7	18000	2	235.0	236.14	159.6	-11.89%	-0.48%
C207	3	321.8	18000	2	288.7	288.70	145.9	-10.29%	0.00%
C208	3	288.1	18000	3	288.1	288.10	148.0	0.00%	0.00%
R101	5	449.4	2991.5	5	453.5	455.36	206.4	0.91%	-0.41%
R102	6	442.1	18000	5	423.2	423.32	161.1	-4.28%	-0.03%
R103	4	316.6	18000	4	320.5	326.32	144.8	1.23%	-1.78%
R104	4	416.5	18000	4	377.8	377.84	171.7	-9.29%	-0.01%
R105	5	413.5	7903.3	5	413.5	414.46	171.5	0.00%	-0.23%
R106	5	452.3	18000	5	448.1	450.70	206.4	-0.93%	-0.58%
R107	5	430.6	18000	4	389.0	391.32	189.1	-9.66%	-0.59%
R108	5	576.4	18000	4	429.6	443.96	184.7	-25.47%	-3.23%
R109	6	504.1	18000	5	479.7	486.56	163.7	-4.84%	-1.41%
R110	5	409.4	18000	4	396.1	396.80	152.8	-3.25%	-0.18%
R111	5	435.7	18000	4	364.1	370.94	230.3	-16.43%	-1.84%
R112	4	387.3	18000	4	367.8	367.80	134.0	-5.03%	0.00%
R201	4	409.2	18000	4	409.2	413.14	156.2	0.00%	-0.95%
R202	2	320.1	18000	2	320.2	328.46	156.8	0.03%	-2.51%
R203	2	333.5	18000	3	312.7	315.82	196.2	-6.24%	-0.99%
R204	2	297.1	18000	1	302.6	305.72	128.2	1.85%	-1.02%
R205	4	352.7	18000	4	352.7	356.64	158.8	0.00%	-1.10%
R206	3	437.9	18000	2	384.8	386.76	182.8	-12.13%	-0.51%
R207	2	317.0	18000	2	312.7	322.04	177.8	-1.36%	-2.90%
R208	2	261.1	18000	1	247.2	251.20	144.0	-5.32%	-1.59%
R209	2	365.4	18000	2	358.6	370.44	137.9	-1.86%	-3.20%
R210	3	339.1	18000	2	320.2	322.34	168.5	-5.57%	-0.66%
R211	2	421.7	18000	2	329.4	331.28	140.0	-21.89%	-0.57%

TABLE 6. ***(Continued.)*** **Result of medium datasets of VRPPL.**

RC101	5	627.7	18000	5	627.7	628.68	124.6	0.00%	-0.16%
RC102	5	546.8	18000	5	538.3	552.48	149.1	-1.55%	-2.57%
RC103	5	468.3	18000	5	468.3	468.30	108.8	0.00%	0.00%
RC104	5	594.5	18000	5	546.1	546.10	143.1	-8.14%	0.00%
RC105	6	581.0	18000	6	579.8	581.68	180.8	-0.21%	-0.32%
RC106	5	556.7	18000	5	555.6	555.60	115.2	-0.20%	0.00%
RC107	6	646.6	18000	5	578.8	578.80	143.1	-10.49%	0.00%
RC108	5	538.6	18000	5	532.8	532.82	105.9	-1.08%	0.00%
RC201	4	538.5	18000	4	538.5	539.14	145.6	0.00%	-0.12%
RC202	3	419.2	18000	3	412.6	415.24	126.8	-1.57%	-0.64%
RC203	4	481.3	18000	3	415.0	415.46	110.7	-13.78%	-0.11%
RC204	2	362.2	18000	1	332.3	332.30	99.4	-8.26%	0.00%
RC205	4	448.7	18000	4	440.8	441.04	195.0	-1.76%	-0.05%
RC206	3	471.0	18000	3	470.2	470.84	200.9	-0.17%	-0.14%
RC207	3	407.5	18000	1	352.5	355.76	120.7	-13.50%	-0.92%
RC208	2	271.2	18000	1	243.7	243.70	160.4	-10.14%	0.00%
Average			17476.188				165.7	-4.82%	-0.92%

$$\text{Gap}^{d} = \frac{(SA\ Distance(Best) - Gurobi\ Distance)}{Gurobi\ Distance} \times 100\%$$

$$\text{Gap}^{e} = \frac{(SA\ Distance(Best) - SA\ \ Distance(Avg))}{SA\ \ Distance(Avg)} \times 100\%$$

of the NFL theorem which states that "there is no absolute efficient heuristic" [45], meaning that each algorithm has advantages and limitations. Therefore, our proposed SA may also perform well in solving VRPPL and VRPTW, but its performance cannot be generalized to all problems before further experiments are conducted.

The critical factors in implementing the SA algorithm to provide a good solution for VRPPL include the utilization of a set of neighborhood moves, the acceptance criterion, and tuning of the parameters. By utilizing various neighborhood moves, the SA could explore a wide range of solution spaces so that it may find a promising solution. The acceptance criterion utilized in the SA enables the algorithm to escape from the local optimal solutions. Lastly, the SA has many parameters defined in Section IV.D. Previous studies [40], [46] have shown that parameter tuning is required so that SA can achieve a good performance. Therefore, we conducted parameter tuning for finding an appropriate parameter configuration.

The computational time required by the SA can be defined by analyzing the operations defined in the SA in terms of complexity. The complexity of the proposed SA is in (1) the objective calculation procedure, (2) the procedure of copying a solution, and (3) the neighborhood moves. The other remaining parts can be executed in a constant time.

The procedures of both calculating the objective function and copying a solution have a complexity of $O(|N_C|)$ because they sequentially evaluate all nodes in the solution representation. There are three neighborhood moves with different complexities. The swap requires a constant time while the other two, insertion, and inversion move, depending on the locations of selected nodes. For the insertion, the complexity is $O(n_i)$, where n_i represents the number of nodes that need to be shifted due to the insertion move. For the inversion, the complexity is $O(\lfloor n_v/2 \rfloor)$ where n_v is the number of nodes located between the selected two nodes (including the two nodes) in the inversion move. Based on the aforementioned complexities, they become the primary reason that the larger the dataset, the longer is the computational time of the SA.

The proposed mathematical model and SA can be extended to solve other potential problems, especially the VRP domain. To adopt the mathematical model, current constraints need to be modified or some new constraints are required to be added. For the SA, the neighborhood moves and the acceptance criterion can be adopted to deal with other VRP variants with some modifications in the proposed solution representation and objective value calculation.

Statistical tests were conducted to statistically evaluate the performance of our proposed SA. We used Wilcoxon signed-rank tests to measure the significance of the difference between two groups of paired data. The Wilcoxon signed-rank is a non-parametric statistical test to analyze two related samples [47] and was used for a similar problem [48]. Tables 8 and 9 show the result for Wilcoxon signed-rank for

TABLE 7. Result of large datasets of VRPPL.

Problem	Gurobi			SA				Gap[d]	Gap[e]
	NV	Distance	Time (s)	NV	Distance (Best)	Distance (Avg)	Time (s)		
C101	10	838.5	18000	10	827.6	849.26	514.4	-1.30%	-2.55%
C102	N/A	N/A	18000	11	1026.7	1046.96	622.8	N/A	-1.94%
C103	14	2079.6	18000	10	879.0	890.74	737.6	-57.73%	-1.32%
C104	11	791.0	18000	10	700.4	710.06	444.7	-11.45%	-1.36%
C105	10	996.2	18000	10	950.5	989.52	618.9	-4.59%	-3.94%
C106	N/A	N/A	18000	12	921.8	932.50	593.0	N/A	-1.15%
C107	N/A	N/A	18000	11	1030.5	1081.22	414.2	N/A	-4.69%
C108	13	1835.7	18000	10	786.7	789.74	675.5	-57.14%	-0.38%
C109	10	1324.2	18000	10	819.5	832.28	471.1	-38.11%	-1.54%
C201	3	511.8	18000	3	511.8	564.30	533.4	0.00%	-9.30%
C202	3	516.9	18000	4	513.0	513.86	362.9	-0.75%	-0.17%
C203	5	699.3	18000	3	497.1	528.28	558.4	-28.91%	-5.90%
C204	6	1113.6	18000	3	512.2	527.66	418.6	-54.01%	-2.93%
C205	4	548.6	18000	3	453.8	474.56	449.8	-17.28%	-4.37%
C206	4	736.9	18000	4	636.2	652.20	729.8	-13.67%	-2.45%
C207	5	616.0	18000	3	461.5	476.68	338.2	-25.08%	-3.18%
C208	4	759.7	18000	4	550.9	561.44	477.3	-27.48%	-1.88%
R101	11	760.7	18000	11	760.1	776.90	701.4	-0.08%	-2.16%
R102	8	774.6	18000	9	718.0	735.78	692.0	-7.31%	-2.42%
R103	N/A	N/A	18000	11	738.4	750.48	912.9	N/A	-1.61%
R104	14	1695.0	18000	8	676.3	697.10	609.3	-60.10%	-2.98%
R105	8	773.4	18000	9	725.8	733.92	551.7	-6.15%	-1.11%
R106	10	979.2	18000	8	651.8	677.00	545.8	-33.44%	-3.72%
R107	9	939.0	18000	8	719.1	728.00	543.9	-23.42%	-1.22%
R108	9	875.7	18000	8	533.0	544.06	590.7	-39.13%	-2.03%
R109	10	870.8	18000	8	673.8	687.10	598.7	-22.62%	-1.94%
R110	10	957.5	18000	8	652.6	666.64	554.0	-31.84%	-2.11%
R111	8	943.5	18000	9	739.4	756.64	440.4	-21.63%	-2.28%
R112	10	1200.8	18000	8	687.9	698.12	507.0	-42.71%	-1.46%
R201	5	764.2	18000	6	700.1	707.64	599.5	-8.39%	-1.07%
R202	6	817.3	18000	4	603.2	612.32	356.9	-26.20%	-1.49%
R203	5	674.8	18000	4	585.6	592.30	615.0	-13.22%	-1.13%
R204	5	618.4	18000	2	387.2	405.32	514.1	-37.39%	-4.47%
R205	4	651.3	18000	3	491.4	496.40	483.3	-24.55%	-1.01%
R206	5	763.9	18000	3	544.5	549.38	513.0	-28.72%	-0.89%
R207	6	678.2	18000	2	492.1	503.92	343.4	-27.44%	-2.35%
R208	6	1046.3	18000	2	354.8	359.30	533.5	-66.09%	-1.25%
R209	4	534.2	18000	3	442.3	454.66	569.1	-17.20%	-2.72%
R210	9	1059.8	18000	4	551.7	555.90	455.9	-47.94%	-0.76%
R211	7	771.8	18000	3	430.7	444.02	466.0	-44.20%	-3.00%

TABLE 7. ***(Continued.)*** **Result of large datasets of VRPPL.**

RC101	N/A	N/A	18000	11	1023.5	1029.20	540.9	N/A	-0.55%
RC102	9	1680.0	18000	10	945.3	976.58	708.6	-43.73%	-3.20%
RC103	N/A	N/A	18000	11	1082.1	1090.58	527.8	N/A	-0.78%
RC104	N/A	N/A	18000	10	889.4	914.04	522.5	N/A	-2.70%
RC105	N/A	N/A	18000	11	1082.9	1104.82	564.6	N/A	-1.98%
RC106	11	1200.6	18000	10	817.0	825.08	755.8	-31.95%	-0.98%
RC107	11	1214.9	18000	10	817.1	824.92	513.0	-32.74%	-0.95%
RC108	N/A	N/A	18000	9	964.1	971.22	758.6	N/A	-0.73%
RC201	6	744.0	18000	6	720.0	729.58	569.7	-3.23%	-1.31%
RC202	5	789.9	18000	5	599.2	627.94	505.7	-24.14%	-4.58%
RC203	N/A	N/A	18000	4	684.3	704.36	496.3	N/A	-2.85%
RC204	6	975.2	18000	2	491.4	503.34	491.8	-49.61%	-2.37%
RC205	4	879.5	18000	5	762.9	770.64	531.7	-13.26%	-1.00%
RC206	5	720.2	18000	5	573.7	588.40	584.0	-20.34%	-2.50%
RC207	4	1197.2	18000	4	599.5	611.14	468.3	-49.92%	-1.90%
RC208	6	797.0	18000	3	499.8	507.26	437.7	-37.29%	-1.47%
Average			18000				547.1	-27.68%	-2.21%

$$\mathrm{Gap}^{d} = \frac{(SA\ Distance(Best) - Gurobi\ Distance)}{Gurobi\ Distance} \times 100\%$$

$$\mathrm{Gap}^{e} = \frac{(SA\ Distance(Best) - SA\ \ Distance(Avg))}{SA\ \ Distance(Avg)} \times 100\%$$

TABLE 8. **Result of wilcoxon signed-rank tests on average solution value and average best solution value for VRPTW datasets.**

	SA vs.	BKS	Zhang et al. [41]	Moradi et al. [16]
Small datasets	Test on average best solution value			
	W	-2.023		
	Asymp. Sig. (2-tailed)	.043*		
	Test on average mean solution value			
	W	-2.201		
	Asymp. Sig. (2-tailed)	.028*		
Large datasets	Test on average best solution value			
	W	-1.572	-.405	-.314
	Asymp. Sig. (2-tailed)	.116	.686	.753
	Test on average mean solution value			
	W	-2.201	-2.023	-.314
	Asymp. Sig. (2-tailed)	.028*	.043*	.753

*Denotes that significant difference exists

VRPTW and VRPPL, respectively. We used the confidence level of alpha = 0.05 to test the hypothesis. If the asymptotic significance (*Asymp. Sig*) (2 tailed) is less than alpha, then we conclude that the two paired samples are significantly different.

Based on Table 8, the *Asymp. Sig.* of average best solution value for VRPTW large datasets are all larger than alpha implying that the performance of the proposed SA algorithm is similar compared to state-of-the-art algorithms in terms of solution quality. According to Table 9, the *Asymp. Sig.* for best solution value and running time for all VRPPL datasets were all less than alpha. This implies that the solution values and running times of the two proposed methods are significantly different. Finally, based on Tables 5, 6, 7, and 9, we can conclude that the SA significantly outperforms the Gurobi in terms of solution quality and run time.

When implementing the model for real-world cases, the model considers three types of consumers. Therefore,

TABLE 9. Result of wilcoxon signed-rank tests on best solution value and running time for vrppl datasets.

	SA vs.	Gurobi solver
Small datasets	Test on best solution value	
	W	-3.527
	Asymp. Sig. (2-tailed)	.000*
	Test on running time	
	W	-6.509
	Asymp. Sig. (2-tailed)	.000*
Medium datasets	Test on best solution value	
	W	-4.482
	Asymp. Sig. (2-tailed)	.000*
	Test on running time	
	W	-6.509
	Asymp. Sig. (2-tailed)	.000*
Large datasets	Test on best solution value	
	W	-5.841
	Asymp. Sig. (2-tailed)	.000*
	Test on running time	
	W	-6.509
	Asymp. Sig. (2-tailed)	.000*

*Denotes that significant difference exists

we need to group consumers by customer type to facilitate product delivery. Data acquisition is also critical for the implementation. The information on the capacity of each parcel locker is needed in advance because, in general, parcel locker management involves other companies.

VI. CONCLUSION

The research proposes a new variant of VRPTW, namely VRPPL, by considering three types of customers that have different delivery choices. A mathematical model and a SA algorithm are proposed. Moreover, we generated a set of VRPPL instances since this problem is a new variant that has never been dealt with in the previous works.

We analyze the performance of the proposed SA in solving both VRPTW and VRPPL benchmark instances. For the small datasets of VRPTW, the SA can lead to near-optimal results, with the highest gap of 0.76%. For the large datasets of VRPTW, the highest average best value gap for SA is -7.87% compared with the best-known solution from Moradi *et al.* [19]. For small and large instances of VRPPL, the SA can outperform the Gurobi with an average gap of 1.90% and 4.82%, respectively. The results indicate that the proposed SA provides reasonable results to the available VRPTW benchmark instances and high-quality results to VRPPL instances.

Several future directions that could be considered are listed as follows. This is a new problem, and we are the first to propose these methods, i.e., a mathematical model and a SA, to deal with the VRPPL. Future works may focus on improving the solutions of VRPPL by proposing a new heuristic-based algorithm. Another avenue to address is to consider another extension on the function of parcel lockers. One of the assumptions considered in this work is that a parcel locker is used to deliver what the customers demand. In future work, the parcel locker may also serve as a place for collecting returned goods. Such extension leads to another real problem arising in the context of reverse logistics.

REFERENCES

[1] Statistica. (2014). *Global Retail E-Commerce Market Size 2014–2021*. Statista. [Online]. Available: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

[2] Y. Wang, D. Zhang, Q. Liu, F. Shen, and L. H. Lee, "Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions," *Transp. Res. E, Logistics Transp. Rev.*, vol. 93, pp. 279–293, Sep. 2016.

[3] M. Behnke, "Recent trends in last mile delivery: Impacts of fast fulfillment, parcel lockers, electric or autonomous vehicles, and more," in *Logistics Management*. Cham, Switzerland: Springer, 2019, pp. 141–156.

[4] Y. Deutsch and B. Golany, "A parcel locker network as a solution to the logistics last mile problem," *Int. J. Prod. Res.*, vol. 56, nos. 1–2, pp. 251–261, Nov. 2018, doi: 10.1080/00207543.2017.1395490.

[5] J. Fang, *The Spatial Dynamics of Amazon Lockers*. Los Angeles, CA, USA: USC Sol Price School of Public Policy, 2019.

[6] C. Bengtsson and A. Vikingson, "Exploring and evaluating the parcel locker—A Swedish consumer perspective," M.S. thesis, Dept. Des. Sci., Lund Univ., Lund, Sweden, 2015.

[7] S.-W. Lin and V. F. Yu, "Solving the team orienteering problem with time windows and mandatory visits by multi-start simulated annealing," *Comput. Ind. Eng.*, vol. 114, pp. 195–205, Dec. 2017.

[8] V. F. Yu, A. A. N. P. Redi, Y. A. Hidayat, and O. J. Wibowo, "A simulated annealing heuristic for the hybrid vehicle routing problem," *Appl. Soft Comput.*, vol. 53, pp. 119–132, Apr. 2017, doi: 10.1016/j.asoc.2016.12.027.

[9] L. Zhou, R. Baldacci, D. Vigo, and X. Wang, "A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution," *Eur. J. Oper. Res.*, vol. 265, no. 2, pp. 765–778, Mar. 2018.

[10] P. Sitek and J. Wikarek, "Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): Model and implementation using hybrid approach," *Ann. Oper. Res.*, vol. 273, nos. 1–2, pp. 257–277, 2019.
[11] C. Tilk, K. Olkis, and S. Irnich, "The last-mile vehicle routing problem with delivery options," *OR Spectr.*, vol. 43, no. 4, pp. 877–904, Dec. 2021, doi: 10.1007/s00291-021-00633-0.
[12] D. L. J. U. Enthoven, B. Jargalsaikhan, K. J. Roodbergen, M. A. J. uit het Broek, and A. H. Schrotenboer, "The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers," *Comput. Oper. Res.*, vol. 118, Jun. 2020, Art. no. 104919, doi: 10.1016/j.cor.2020.104919.
[13] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, Oct. 1959, doi: 10.1287/mnsc.6.1.80.
[14] M. M. Solomon, "Vehicle routing and scheduling with time windows constraints: Models and algorithms," Ph.D. dissertation, Dept. Decis. Sci., Univ. Pennsylvania, Philadelphia, PA, USA, 1983.
[15] D. Pecin, C. Contardo, G. Desaulniers, and E. Uchoa, "New enhancements for the exact solution of the vehicle routing problem with time windows," *INFORMS J. Comput.*, vol. 29, no. 3, pp. 489–502, Aug. 2017.
[16] Y. Nagata and O. Bräysy, "A powerful route minimization heuristic for the vehicle routing problem with time Windows," *Oper. Res. Lett.*, vol. 37, no. 5, pp. 333–338, Sep. 2009.
[17] A. Lim and X. Zhang, "A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows," *INFORMS J. Comput.*, vol. 19, no. 3, pp. 443–457, 2007.
[18] Y.-J. Gong, J. Zhang, O. Liu, R.-Z. Huang, H. S.-H. Chung, and Y.-H. Shi, "Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 254–267, Mar. 2012.
[19] B. Moradi, "The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution model," *Soft Comput.*, vol. 24, pp. 6741–6769, Sep. 2019.
[20] K.-W. Pang, "An adaptive parallel route construction heuristic for the vehicle routing problem with time Windows constraints," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11939–11946, Sep. 2011.
[21] Y. Nagata, O. Bräysy, and W. Dullaert, "A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time Windows," *Comput. Oper. Res.*, vol. 37, no. 4, pp. 724–737, Apr. 2010.
[22] R. Baños, J. Ortega, C. Gil, A. Fernández, and F. de Toro, "A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time Windows," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1696–1707, Apr. 2013.
[23] J.-F. Cordeau and M. Maischberger, "A parallel iterated Tabu search heuristic for vehicle routing problems," *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2033–2050, Sep. 2012.
[24] Z. Xu and Y. Cai, "Variable neighborhood search for consistent vehicle routing problem," *Expert Syst. Appl.*, vol. 113, pp. 66–76, Dec. 2018.
[25] H.-F. Wang and Y.-Y. Chen, "A genetic algorithm for the simultaneous delivery and pickup problems with time window," *Comput. Ind. Eng.*, vol. 62, no. 1, pp. 84–95, Feb. 2012.
[26] B. Barán and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows," *Appl. Informat.*, pp. 97–102, Feb. 2003.
[27] Y. Qi, Z. Hou, H. Li, J. Huang, and X. Li, "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 62, pp. 61–77, Oct. 2015.
[28] E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "Meta-harmony search algorithm for the vehicle routing problem with time windows," *Inf. Sci.*, vol. 325, pp. 140–158, Dec. 2015.
[29] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *J. Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, 2011.
[30] G. Jeon, H. R. Leep, and J. Y. Shim, "A vehicle routing problem solved by using a hybrid genetic algorithm," *Comput. Ind. Eng.*, vol. 53, no. 4, pp. 680–692, Nov. 2007.
[31] A. Subramanian, P. H. V. Penna, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for the heterogeneous fleet vehicle routing problem," *Eur. J. Oper. Res.*, vol. 221, no. 2, pp. 285–295, Sep. 2012.
[32] Y. Marinakis and M. Marinaki, "A hybrid genetic—Particle swarm optimization algorithm for the vehicle routing problem," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1446–1455, Mar. 2010.
[33] Y. Marinakis, M. Marinaki, and G. Dounias, "A hybrid particle swarm optimization algorithm for the vehicle routing problem," *Eng. Appl. Artif. Intell.*, vol. 23, no. 4, pp. 463–472, 2010.
[34] V. F. Yu, S.-W. Lin, W. Lee, and C.-J. Ting, "A simulated annealing heuristic for the capacitated location routing problem," *Comput. Ind. Eng.*, vol. 58, no. 2, pp. 288–299, Mar. 2010, doi: 10.1016/j.cie.2009.10.007.
[35] C.-S. Jeong and M.-H. Kim, "Fast parallel simulated annealing for traveling salesman problem on SIMD machines with linear interconnections," *Parallel Comput.*, vol. 17, nos. 2–3, pp. 221–228, Jun. 1991.
[36] S. Sofianopoulou, "Simulated annealing applied to the process allocation problem," *Eur. J. Oper. Res.*, vol. 60, no. 3, pp. 327–334, Aug. 1992.
[37] S. U. Seçkiner and M. Kurt, "A simulated annealing approach to the solution of job rotation scheduling problems," *Appl. Math. Comput.*, vol. 188, no. 1, pp. 31–45, May 2007.
[38] C. Wang, D. Mu, F. Zhao, and J. W. Sutherland, "A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time Windows," *Comput. Ind. Eng.*, vol. 83, pp. 111–122, May 2015.
[39] G. Laporte and Y. Nobert, "Exact algorithms for the vehicle routing problem," *North-Holland Math. Stud.*, vol. 132, no. 1, pp. 147–184, 1987.
[40] N. M. E. Normasari, V. F. Yu, C. Bachtiyar, and Sukoyo, "A simulated annealing heuristic for the capacitated green vehicle routing problem," *Math. Problems Eng.*, vol. 2019, pp. 1–18, Jan. 2019.
[41] W.-C. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time Windows," *Ann. Oper. Res.*, vol. 63, no. 1, pp. 3–27, Feb. 1996.
[42] Y. Kuo, "Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem," *Comput. Ind. Eng.*, vol. 59, no. 1, pp. 157–165, 2010.
[43] K. Karagul, Y. Sahin, E. Aydemir, and A. Oral, "A simulated annealing algorithm based solution method for a green vehicle routing problem with fuel consumption," in *Lean and Green Supply Chain Management* (International Series in Operations Research & Management Science), vol. 273, T. Paksoy, G. W. Weber, and S. Huber, Eds. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-319-97511-5_6.
[44] D. Zhang, S. Cai, F. Ye, Y.-W. Si, and T. T. Nguyen, "A hybrid algorithm for a vehicle routing problem with realistic constraints," *Inf. Sci.*, vols. 394–395, pp. 167–182, Jul. 2017.
[45] D. H. Wolpert, *What the no Free Lunch Theorems Really Mean; How to Improve Search Algorithms*, vol. 4. Santa Fe, NM, USA: Santa Fe Institute, 2012.
[46] B. Rabbouch, F. Saâdaoui, and R. Mraihi, "Empirical-type simulated annealing for solving the capacitated vehicle routing problem," *J. Experim. Theor. Artif. Intell.*, vol. 32, no. 3, pp. 437–452, May 2020.
[47] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, 9th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2011.
[48] V. F. Yu, Winarno, A. Maulidin, A. A. N. P. Redi, S.-W. Lin, and C.-L. Yang, "Simulated annealing with restart strategy for the path cover problem with time Windows," *Mathematics*, vol. 9, no. 14, p. 1625, Jul. 2021.

VINCENT F. YU received the Ph.D. degree in industrial & operations engineering from the University of Michigan, Ann Arbor. He is currently a Professor of Industrial Management with the National Taiwan University of Science and Technology. He has published articles in *Applied Mathematical Modelling*, *Applied Soft Computing*, *Computers & Industrial Engineering*, *Computers & Operations Research*, *European Journal of Operational Research*, *Industrial Marketing Management*, *International Journal of Production Research*, *Journal of Cleaner Production*, *Journal of Intelligent Manufacturing*, *Omega*, and *Transportation Research Part E: Logistics and Transportation Review*. His current research interests include operations research, logistics management, soft computing, and artificial intelligence.

HADI SUSANTO received the B.Eng. degree in industrial engineering from the Institut Teknologi Bandung (ITB), Indonesia, the M.B.A. degree in industrial management from the National Taiwan University of Science and Technology, Taipei, Taiwan, and the M.T. (Magister Teknik) degree from ITB. He is currently pursuing the Ph.D. degree in industrial management with the National Taiwan University of Science and Technology. From 2012 to 2014, he was a Lecturer in industrial engineering with the Institut Teknologi Nasional (ITENAS), Bandung, Indonesia. His research interests include supply chain management, optimization, shelf-space allocation, routing problem, and waste management. He has already published his previous works in several conferences, e.g., Asia Pacific Industrial Engineering and Management Society (APIEMS) and Operations Research and Maritime Logistics (ORML).

PANCA JODIAWAN received the B.Eng. degree in industrial engineering from Bunda Mulia University, Jakarta, Indonesia, in 2016, and the M.B.A. degree in industrial management from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2019, where he is currently pursuing the Ph.D. degree in industrial management. From 2016 to 2017, he was a Lecturer Assistant with the Industrial Engineering Department, Bunda Mulia University. He has already published his previous works in several conferences, e.g., International Congress of Logistics and SCM Systems 2019 and International Conference on Industrial Engineering & Engineering Management 2019. His research interests include combinatorial optimization in logistics and metaheuristics design.

TSAI-WEI HO received the master's degree in industrial management from the National Taiwan University of Science and Technology. She has expertise in supply chain management, especially in logistic network design and planning. She is currently working at Universal Global Scientific Industrial Company Ltd., Nantou, Taiwan. She plans to build up her on-the-job experiences and continue her involvement in the supply chain and logistics industry. Her research interest includes vehicle routing problem combining with the parcel lockers.

SHIH-WEI LIN received the bachelor's, master's, and Ph.D. degrees in industrial management from the National Taiwan University of Science and Technology, Taiwan, in 1996, 1998, and 2000, respectively. He is currently a Professor with the Department of Information Management, Chang Gung University, Taiwan. He is also with the Department of Neurology, Linkuo Chang Gung Memorial Hospital, Taoyuan, Taiwan, and the Department of Industrial Engineering and Management, Ming Chi University of Science and Technology, Taipei, Taiwan. His papers have appeared in *Computers & Operations Research*, *European Journal of Operational Research*, *Journal of the Operational Research Society*, *European Journal of Industrial Engineering*, *International Journal of Production Research*, *International Journal of Advanced Manufacturing Technology*, *Knowledge and Information Systems*, *Applied Soft Computing*, *Applied Intelligence*, *Expert Systems with Applications*, and IEEE Access. His current research interests include meta-heuristics and data mining.

YU-TSUNG HUANG received the B.Sc. degree in finance, and the M.B.A. and Ph.D. degrees in business and management from the National Central University, Taiwan. He is currently the Director of Kerry TJ Logistics Company Ltd., Taiwan. He has published articles in *Journal of Medical Systems*, *International Journal of Industrial Engineering*, *International Journal of Strategic Decision Science*, *Lecture Notes in Computer Science*, and *International Journal of Logistics and SCM Systems*. His research interests include the areas of global logistics management, transportation, warehousing, and business process management.