



A Coordination-Based Algorithm for Dedicated Destination Vehicle Routing in B2B E-Commerce

Tsung-Yin Ou¹, Chen-Yang Cheng², Chun Hsiung Lai³ and Hsin-Pin Fu^{1,*}

¹Department of Marketing and Distribution Management, National Kaohsiung University of Science and Technology, Taiwan

²Department of Industrial Engineering and Management, Taipei University of Technology, Taiwan

³Department of Enterprise Information and Industrial Engineering, Tunghai University, Taiwan

*Corresponding Author: Hsin-Pin Fu. Email: hpfu@nkust.edu.tw Received: 09 March 2021; Accepted: 09 May 2021

Abstract: This paper proposes a solution to the open vehicle routing problem with time windows (OVRPTW) considering third-party logistics (3PL). For the typical OVRPTW problem, most researchers consider time windows, capacity, routing limitations, vehicle destination, etc. Most researchers who previously investigated this problem assumed the vehicle would not return to the depot, but did not consider its final destination. However, by considering 3PL in the B2B e-commerce, the vehicle is required back to the nearest 3PL location with available space. This paper formulates the problem as a mixed integer linear programming (MILP) model with the objective of minimizing the total travel distance. A coordinate representation particle swarm optimization (CRPSO) algorithm is developed to obtain the best delivery sequencing and the capacity of each vehicle. Results of the computational study show that the proposed method provides solution within a reasonable amount of time. Finally, the result compared to PSO also indicates that the CRPSO is effective.

Keywords: Third-party logistics; open vehicle routing problem with time windows; dedicated destination

Notations

<i>t</i> :	Iteration index, $t = 1T$
<i>i</i> :	Particle index, $i = 1I$
<i>d</i> :	Dimension index, $d = 1D$
<i>u</i> :	Uniform random number in the interval [0,1]
w(t):	Inertia weight in the t^{th} iteration
$v_{id}(t)$:	Velocity of the i^{th} particle at the d^{th} dimension in the t^{th} iteration
$x_{id}(t)$:	Position of the i^{th} particle at the d^{th} dimension in the t^{th} iteration
P_{id} :	Personal best position (<i>p</i> best) of the i^{th} particle in the d^{th} dimension
P_{gd} :	Global best position (gbest) in the d^{th} dimension
C_p :	Personal best position acceleration constant
C_g :	Global best position acceleration constant
X_{max} :	Maximum position value



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

X_{min} :	Minimum position value
X_i :	Vector position of the <i>i</i> th particle, $[x_{i1}, x_{i2}, \dots, x_{iD}]$
V_i :	Vector velocity of the i^{th} particle, $[v_{i1}, v_{i2}, \dots v_{iD}]$
P_i :	Vector personal best position of the i^{th} particle, $[P_{i1}, P_{i2}, \dots P_{iD}]$
P_g :	Vector global best position, $[P_{g1}, P_{g2}, \dots P_{gD}]$
R_i :	Set of vehicle routes corresponding to the i^{th} particle
D_i :	Set of distances between vehicles and destinations corresponding to <i>i</i> th particle
$\Psi(X_i)$:	Fitness value of X_i

1 Introduction

In B2B e-commerce, logistics is viewed as an increasingly important activity. Regardless of industry, most e-businesses rely on logistics management to enhance their competitiveness [1]. Despite the importance of logistics management, e-businesses tend to focus primarily on developing their core abilities (e.g., R&D, product, marketing). Logistics and other non-core activities often are outsourced to other companies [2]. Independent third-party logistics (3PL) companies provide professional, integrated distribution services and information technology to help decrease fixed and variable costs of logistics [3]. With the rapid development of electronic commerce, e-businesses are facing new challenges in the logistics supply chain and are partnering extensively with 3PL firms to deliver products on time, increase customer satisfaction, decrease logistics costs and increase profits.

Logistics professionals must take many constraints into account in order to plan optimal distribution routes that consider customer demand, vehicle routing, vehicle capacity, etc. Together, these constraints associated with delivery logistics are called the vehicle routing problem (VRP) [4]. Ref. [5] solved online pick-to-sort order batching problem for managing frequent arrivals in B2B e-commerce. Ref. [6] applied the meta-heuristic method of ant colony optimization (ACO) to an established set of vehicle routing problems (VRP). The VRP can be divided into two types: the Hamiltonian cycle (traditional VRP, where the vehicle returns to the depot), and the Hamiltonian path (the open vehicle routing problem, or OVRP, where the vehicle does not need to return to the depot). Thus, routing destination is the biggest difference between the VRP and the OVRP.

In this paper, we propose a solution to the open vehicle routing problem with time windows (OVRPTW) considering 3PL. In the problem, a 3PL company leaves its vehicles at its client's depot until they are loaded. Post-delivery, vehicles do not return to the depot, but to the nearest 3PL company location with available space. In most previous OVRPTW literature that considered 3PL, a vehicle's final destination was not considered. However, in this research, we not only consider common constraints in OVRPTW such as vehicle capacity, but also a 3PL constraint in which the vehicles must return to a 3PL company location - a limitation of destination. We propose a mixed integer linear programming (MILP) model that considers these practical characteristics. In this research, we use a classical OVRPTW formulation with 3PL considerations to solve an experimental problem set. Due to the computational complexity of the model, we designed a coordinate representation particle swarm optimization (CRPSO) algorithm to obtain the near-optimal vehicle routing plan with the objective of minimizing total travel distance.

The rest of this paper is organized as follows. In Section 2, we review the literature on related VRPs considering 3PL and OVRPTW. In Section 3, we present the proposed MILP model for the problem and the CRPSO algorithm used to obtain the solutions. In Section 4, we present a computational study that demonstrates the excellent performance of the CRPSO algorithm. Finally, in Section 5 we summarize the results of this research and suggest a possible direction for future research.

2 Literature Review

Many enterprises outsource non-core functions such as distribution logistics to promote competitiveness; in response to this trend, the 3PL sector is growing rapidly [2]. Increasingly, VRP researchers also are considering 3PL. [7] presented a web-based decision support system (DSS) for waste lube oils collection and recycling operations considering cooperation with a 3PL company. Because the logistics function is outsourced to a 3PL company, vehicle routing begins from the depot and ends at a 3PL location. This feature of delivery is similar to our research. In the study, the DSS system enables schedulers to tackle reverse supply chain management problems interactively and can be applied to realistic reverse logistical planning problems [8].

Baykasoglu and Kaplanoglu proposed a multi-agent based load consolidation decision-making approach considering many kinds of logistics mechanisms, including in-house logistics systems and 3PL, so as to improve the logistics efficiency of enterprises [9]. Amorim et al. formulated models for a case of perishable goods with a mix of fixed and loose shelf lives. When the shelf life of product did not match the distribution plan, it would be outsourced to a 3PL company. The results show that the economic benefits derived from using an integrated approach depend greatly on the freshness level of delivered products [10]. Moon et al. extended the VRPTW to the VRPTW with overtime and outsourcing vehicles (VRPTWOV), which allows overtime for drivers and the possibility of using outsourced vehicles. They developed a mixed integer programming model, a genetic algorithm (GA) and a hybrid algorithm based on simulated annealing to demonstrate the efficiency of their solution [11]. In this paper, we extend research in this important area by proposing a solution to an experimental problem set that incorporates 3PL considerations into a classical OVRPTW formulation.

Routing destination is the biggest difference between VRP and OVRP. VRP is called the Hamiltonian cycle, and OVRP is called the Hamiltonian path [12]. All other constraints, such as vehicle capacity, time windows, etc. are the same. OVRP was not as important as VRP in the early 1980s. Schrage was the first to classify routing types and define OVRP with the objective of minimizing the number of routes (i.e., vehicles) and total cost [13].

OVRP is a very common problem in daily life, especially in logistics, transportation and other similar industries. Several scholars have proposed solutions to problems in these contexts. Sariklis and Powell proposed a two-stage model based on a minimum spanning tree to solve a vehicle routing decision problem [14]. Li et al. developed a hybrid ant colony algorithm (ACO) combined with taboo search (TA) to solve OVRP [15]. Fleszar et al. proposed variable neighborhood search (VNS) to determine the customer service sequence [16]. OVRPTW (the focus of this research) extends OVRP by considering the concept of time windows. Recently, researchers have proposed solutions to such problems. Repoussis et al. proposed a comprehensive mathematical model to capture all aspects of OVRPTW, which they solved using a greedy look-ahead route construction heuristic algorithm [7].

In most of the extant literature, researchers solved VRP, VRPTW, OVRP and OVRPTW as individual problems. Unlike previous studies, however, this paper addresses a new topic: OVRPTW considering 3PL. Since OVRPTW is NP-hard, most researchers have solved such problems using heuristic algorithms [16]. However, due to the computational complexity of the model, it was necessary to develop an algorithm based on particle swarm optimization (PSO) to solve the proposed problem. Since a standard PSO algorithm cannot be applied to discrete problems directly, the encoding and decoding methods are critical. Ai and Kachitvichyanukul proposed a PSO algorithm for solving a vehicle routing problem with simultaneous pickup and delivery (VRPSPD) as well as a capacitated vehicle routing problem (CVRP) [17,18]. The solution representation for VRPSPD with n customers and m vehicles is a (n + 2m) dimensional particle. The decoding method starts by transforming the particle into a priority list of customers and a priority matrix of vehicles to serve each customer. The vehicle routes are constructed

based on the customer priority list and the vehicle priority matrix. Applying this encoding method, we propose a coordinate representation particle swarm optimization (CRPSO) algorithm to obtain the optimal solution. The designed algorithm with *n* customers and *m* vehicles yields (n + 2m + m) dimensions for each particle. The encoding and decoding methods are described in detail in Section 4.

3 OVRPTW Considering 3PL

3.1 Problem Description

The logistics department at the depot is determining routes for delivery vehicles, and the forwarder will load goods into the vehicles based on customer demand and deliver them following the assigned routes. As shown in Fig. 1, the problem considering 3PL can be described as follows. There is one depot, and the number of vehicles and demand for each customer are known. Many vehicles owned by a 3PL company are parked at the depot and ready to be loaded. All vehicles have the same capacity, and depart the depot to deliver goods to customers with specific demand. Each customer must be served only once by one vehicle within the delivery time window, which is bounded by an earliest start time and latest start time. Since this is an OVRPTW problem considering 3PL, the vehicles do not return to the depot, but to the nearest 3PL company location with available space. Based on these constraints, the objective is to minimize the total travel distance.



Figure 1: Illustration of delivery logistics

3.2 Mathematical Model

In this section, we formulate the mixed integer linear programming model for the addressed OVRPTW problem considering 3PL. Specifically, we modify Repoussis et al.'s MILP model [7] to incorporate 3PL considerations. Notations are defined as follows:

- A. Notations
- Indexes

i, *j*: the index of the node being used as the demand point, including the depot

k: the service vehicle index

r: the index of 3PL company locations

• Sets

N: the set of customers including node of the depot, where *i* or j = 1.

V: the set of vehicles

PL: the set of 3PL locations

• Parameters

C: capacity of each vehicle

- q_i : demand of customer i
- c_{ij} : the cost from node *i* to node *j*, $i \neq j$
- t_{ij} : travel time from node *i* to node *j*, $i \neq j$
- d_{ij} : the distance from node *i* to node *j*, $i \neq j$
- w_k : fixed cost for the acquisition of vehicle k
- $[e_i, l_i]$: time window, $i \in N$, where

 e_i : the earliest service start time for customer i

 l_i : the latest service start time for customer i

- s_i: service time of customer i
- p_i : departure time from customer i
- a_i : arrival time to customer i

• Variables

$$x_{ij}^{k} = \begin{cases} 1 \text{ if customer } j \text{ is visited after customer } i \text{ by vehicle } k \\ 0 & \text{otherwise} \end{cases}$$
(1)

$$z_k = \begin{cases} 1 \text{ If i venicie } k \text{ is active} \\ 0 & \text{otherwise} \end{cases}$$
(2)

$$Y_{ir}^{k} = \begin{cases} 1 \text{ if } 3PL \ r \text{ is visited after customer } i \text{ by vehicle } k \\ 0 & \text{otherwise} \end{cases}$$
(3)

C. Mixed integer linear programming model

We formulated a mixed integer linear programming model for the addressed OVRPTW problem considering 3PL. We describe the objective function and constraints below.

Objective function

$$Min\sum_{k=1}^{|V|}\sum_{i=1}^{N}\sum_{j=1}^{N}d_{ij}x_{ij}^{k}$$
(4)

The objective function is to minimize the total travel distance.

Subject to

$$\sum_{k=1}^{|V|} \sum_{i=1}^{N} x_{ij}^{k} = 1, \forall j = 2, 3, \dots N$$
(5)

$$\sum_{k=1}^{|V|} \sum_{j=1}^{N} x_{ij}^{k} = 1, \forall i = 2, 3, \dots N$$
(6)

Constraints (5) and (6) ensure that exactly one vehicle arrives at and departs from each customer and the depot.

$$x_{ij}^k \le z_k, \forall i, j = 2, 3, \dots N, \forall k = 1, 2, \dots, |V|$$
(7)

Constraint (7) is relative to x and z variables, ensuring that all customers are visited by active vehicles.

$$\sum_{i=1}^{N} x_{iu}^{k} - \sum_{j=1}^{N} x_{uj}^{k} = 0, \forall k = 1, 2, ..., |V|, \forall u = 1, 2, ..., N$$
(8)

Constraint (8) guarantees the flow continuance for each vehicle route.

$$\sum_{i=1}^{N} q_i \left(\sum_{j=1}^{N} x_{ij}^k \right) \le C, \forall k = 1, 2, \dots, |V|$$
(9)

Constraint (9) ensures that the total service quantity of each vehicle does not exceed its capacity.

$$\sum_{(i,j)\in S\times S} x_{ij}^k \le |S| - 1, \forall S \subseteq N : 2 \le |S| \le N, \forall k = 1, 2, ..., |V|$$
(10)

Constraint (10) eliminates sub-tour routes.

$$a_j \ge (p_i + t_{ij}) - (1 - x_{ij}^k)M, \forall k = 1, 2, \dots, |V|, \forall i, j = 1, 2, \dots, N$$
(11)

$$a_j \le (p_i + t_{ij}) + (1 - x_{ij}^k)M, \forall k = 1, 2, \dots, |V|, \forall i, j = 1, 2, \dots, N$$
(12)

Constraints (11) and (12) are related to time windows and ensuring feasible schedules for vehicles. If customers i and j are scheduled consecutively on the route of vehicle k, the arrival time of customer j is equal to the departure time of customer i plus the travel time between these two customers, where M is a large number.

$$a_i \le p_i - S_i, \forall i = 2, 3, \dots N$$
 (13)

$$e_i \le p_i \le l_i, \forall i = 2, 3, \dots N \tag{14}$$

Constraints (13) and (14) insure that the relationships between arrival time, departure time and service time are compatible with customer i's time window.

$$p_i = 0 \tag{15}$$

Constraint (15) sets the departure time of all vehicles from the depot to be zero.

$$x_{ij}^k \in \{0, 1\}, \forall i, j = 1, 2, \dots, \forall k = 1, 2, \dots, |V|$$
(16)

$$z_k \in \{0,1\}, \forall i, j = 1, 2, \dots, N, \forall k = 1, 2, \dots, |V|$$

$$(17)$$

Lastly, constraints (16) and (17) define variables x and z for each vehicle k. So far, Eqs. (1) to (17) describe the standard VRPTW formulation without considering the "open" routing concept.

900

$$\sum_{j=2}^{N} x_{1j}^{k} \le 1, \forall k = 1, 2, \dots |V|$$
(18)

$$\sum_{j=2}^{N} x_{i1}^{k} = 0, \forall k = 1, 2, \dots |V|$$
(19)

Constraints (18) and (19) specify such "open" characteristics. Constraint (18) guarantees that every vehicle will depart from the depot to service a sequence of customers, and constraint (19) ensures that no vehicles will return to the depot. So far, Eqs. (1) to (19) comprise the classical OVRPTW model. In this case, we need the following constraint:

$$\sum_{r=N+1}^{N+PL} Y_{ir}^k = 1$$
(20)

Constraint (20) ensures that the final destination for each vehicle is a 3PL company location. It is worth mentioning the difference between our problem and the original problem solved by [7]. Equation (20) limits the end point of each vehicle's route to a 3PL company location. That is, when a vehicle finishes making its deliveries, it returns to a specific destination.

3.3 CRPSO Algorithm

We developed a coordinate representation particle swarm optimization algorithm to search for nearoptimal solutions of the appropriate customer sequence and determine the feasible capacity of each vehicle based on coordinate dimensions and evolutionary processes. To evaluate the fitness values of the coordinate-coded dimensions obtained from the particle swarm optimization algorithm, we first developed a customer sequencing assignment procedure to determine the customer delivery priorities. Then, we used coordinate representation to generate a vehicle priority matrix and a destination priority matrix. In order to construct vehicle routes, vehicle capacity must be limited. Following the procedures described in the previous section with the associated constraints, the travel distance for each vehicle route can be calculated as the fitness value of each particle dimension. The CRPSO is repeated until the termination condition is satisfied.

Particle swarm optimization was proposed by Eberhart and Kennedy [19]. PSO was first intended to simulate social behavior as a stylized representation of the movement of a group of organisms (e.g., a flock of birds or a school of fish). [20] propose that PSO achieves better specific work output across a range of algorithm control parameters and converges to optimum solution with lower computation cost. [21] implement Particle swarm optimization (PSO) and artificial bee colony (ABC) optimization methods to the histogram stretching technique in parameter selection process. [22] also integrate Particle swarm optimization (PSO) to obtain the optimal parameter combination of the regularization parameter c and the kernel function width coefficient in least squares support vector machine (LSSVM). The newly combined methodology provides better generalization ability, and higher prediction accuracy for highway cost prediction in complex environments. In PSO, a swarm of P particles serves as a searching agent for a specific problem solution. The searching strategy of PSO requires updating the new position and velocity for the next iteration based on the current velocity of each particle (v_i) , the personal best experience of each particle $(x_{p(i)})$, and the global best experience of all particles (x_{g}) . The procedure of calculating the new velocity and the position of every particle in the next iteration could be shown in the mathematical model. Equation (21) shows that the new velocity of the particle is updated using the current position and the current velocity. Each particle moves the new position in the next iteration according to the Equation (22).

$$v_{id}(t+1) = w * v_{id}(t) + c_1 r_1 \left(x_{p(id)}(t) - x_{id}(t) \right) + c_2 r_2 \left(x_{gd}(t) - x_{id}(t) \right)$$
(21)

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$
(22)

where $v_{id}(t)$ represents the velocity of the d^{th} dimension of the i^{th} particle in the t^{th} iteration. The variable $x_{id}(t)$ represents the position of the d^{th} dimension of the i^{th} particle in the t^{th} iteration. The variable w represents the inertia weight, c_1 is the self-cognition acceleration coefficient, and c_2 is the social cognition acceleration coefficient, r_1 and r_2 are two separately generated, uniformly distributed random numbers in the range [0,1].

The CRPSO framework for solving OVRPTW considering 3PL in this paper is based on the Object Library for Evolutionary Techniques [23]. The notations and a description of the algorithm are provided below.

CRPSO Framework

1) Set iteration t = 1. Initialize I particles as a population, generate the i^{th} particle with random position X_i in the range $[X^{max}, X^{min}]$. Velocity $V_i = 0$ and personal best $P_i = X_i$ for i = 1...I.

2) For i = 1...I, decode X_i to a set of vehicle routes R_i and vehicle destination D_i (see decoding method in Section 4.3).

3) For i = 1...I, compute the performance measurement of R_i and D_i , i.e., the total travel distance for all routes, and set this as the fitness value of X_i , represented by $\Psi(X_i)$.

- 4) Update *p*best, for i = 1...I, update $P_i = X_i$, if $\Psi(X_i) < \Psi(P_i)$.
- 5) Update gbest, for i = 1...I, update $P_g = P_i$, if $\Psi(P_i) < \Psi(P_g)$.

6) Update the velocity and the position of each i^{th} particle

$$w(t) = w(T) + (t - T)/(1 - T)[w(1) - w(T)]$$
(23)

$$v_{id}(\tau + 1) = w(t)v_{id}(t) + C_p u(P_{id} - x_{id}(t)) + C_g u(P_{gd} - x_{id}(t))$$
(24)

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$
(25)

If x_{id} $(t + 1) > X_{max}$, then

$$x_{id}(t+1) = X_{max} \tag{26}$$

$$v_{id}(t+1) = 0$$
 (27)

If x_{id} $(t + 1) < X_{min}$, then

$$x_{id}(t+1) = X_{min} \tag{28}$$

$$v_{id}(t+1) = 0$$
 (29)

7) If the stopping criterion is met, i.e., t = T, stop. Otherwise, t = t + 1 and return to step 2.

8) Decode Pg as the best set of vehicle routes found, $R^* + D^*$ with its corresponding performance measurement $\Psi(Pg)$.

3.4 Solution Representation of CRPSO

The solution representation of vehicle routes is one of the key elements for an effective implementation of CRPSO to solve OVRPTW considering 3PL. The solution representation in CRPSO of OVRPTW considering 3PL with n customers and m vehicles consists of (n + 2m + m) dimensional particles, as shown in Fig. 2. Each dimension of a particle is encoded as a real number. Hence, the solution representation is divided into four parts: the customer priority list, the vehicle priority matrix, the 3PL

destination priority matrix, and the vehicle capacity matrix. Fig. 2 illustrates an example for eight customers and two vehicles.



m+*m* dimensions of vehicle

Figure 2: Particle dimensions in CRPSO

The first eight dimensions are related to customers, and each dimension represents a single customer. These dimensions are required to create a priority list of customers to be added to the routes. The priority list is determined by sorting the first eight dimensional values. Smaller values indicate higher priority customers. The second and third parts of CRPSO extract the reference points for vehicles. These reference points determine the vehicle priority matrix for routes, which is constructed based on the relative distance between these points and a customer's location. In other words, a vehicle is defined as a reference point on a Google map. Customers are priority matrix based on the distance between these points. In the second and third parts, the four dimensions consist of longitude and latitude for each vehicle. Therefore, the representation is called a coordinate representation. The last part, comprised of two dimensions, is associated with the capacity of each vehicle. The value of each dimension is the service limitation for each vehicle. The purpose of this representation is to avoid problems such as exceeding delivery capacity.

In summary, the proposed solution representation consists of three types of dimensional designs, including customer sequencing, vehicle coordinates, and vehicle capacity. The problem with n customers and m vehicles requires (n + 2m + m) dimensions for every particle. Each particle dimension is encoded as a real number. The first n dimensions represent customer priorities, and the values of each dimension are converted into a customer priority list in the decoding procedure. The second 2m dimensions represent the reference points for vehicles. These values are turned into the vehicle priority matrix and the 3PL destination priority matrix. The last m dimensions represent the capacity of each vehicle. An example of the CRPSO solution representation is displayed in Fig. 3.

3.5 Decoding Method

The decoding method is modified from Ai and Kachitvichyanukul's decoding method in our CRPSO solution [17]. The notations and decoding procedure are presented below.

Notations

- x_{id} Position of the *i*th particle in the *d*th dimension
- R_{ii} Route of the *j*th vehicle corresponding to the *i*th particle
- D_{ii} Distance to destination of the j^{th} vehicle corresponding to the i^{th} particle

Decoding procedure

1) Construct the priority list of customers (U) and the list of vehicles (L).

- i) Build set $S = \{1, 2, ..., n\}$ and $U = \theta$.
- ii) Select *c* from set *S* where $x_{ic} = \min_{i=1}^{n} x_{id}$.
- iii) Add c to the last position in set \tilde{U} .

- iv) Remove c from set S.
- v) Repeat step 1, parts b–d until $S = \theta$.
- 2) Construct the vehicle priority matrix (V) and the 3PL destination priority matrix (D).
 - a. Set the vehicle reference position. For j = 1...m, set $xref_i = x_{i,n+j}$ and $yref_i = x_{i,n+m+j}$.
 - b. For each customer k, k = 1...n.
 - i) For each vehicle j = 1...m, set δ_j as the Euclidean distance between customer k and the reference point of vehicle j.
 - ii) Build set $S = \{1, 2, ..., m\}$ and $V_k = \theta$.
 - iii) Select c from set S where $\delta_c = \min_{d \in S} \delta_d$.
 - iv) Add c to the last position in set V_k .
 - v) Remove *c* from set *S*.
 - vi) Repeat step 2b, parts iii–v until $S = \theta$.
 - c. For each vehicle j, j = 1...m
 - i) For each 3PL z = 1...Z, set μ_j as the Euclidean distance between 3PL z and the reference point of vehicle j.
 - ii) Build set $S = \{1, 2, ..., Z\}$ and $D_k = \theta$.
 - iii) Select c from set S where $\mu_c = \min_{d \in S} \mu_d$.
 - iv) Add *c* to the last position in set D_k .
 - v) Remove *c* from set *S*.
 - vi) Repeat step 2c, parts iii–v until $S = \theta$.
- 3) Determine vehicle routes and vehicle 3PL destinations.

a. Set k = 1.

b. Add customers one by one to the route.

- i) Set $l = U_k$ and p = 1.
- ii) Set $j = V_{l,q}$.
- iii) Create a new candidate route by inserting customer l into the best sequence in route R_{ij} , which has the smallest additional cost.
- iv) Check the capacity and route time constraints of the candidate route.
- v) If a feasible solution is reached, update route R_{ij} with the candidate route.
- vi) If p = m, go to step 3c. Otherwise, set p = p + 1 and go to step 3b, part ii.

c. If k = n, stop. Otherwise, set k = k + 1 and repeat step 3b.

d. Set j = 1

- e. Assign vehicles one by one to a 3PL destination.
 - i) Set $l = L_k$ and q = 1
 - ii) Set $z = D_{l.q}$.
 - iii) Create a new candidate destination by assigning vehicle l to the best destination in the 3PL D_{ij} .
 - iv) If a feasible solution is reached, update the route D_{ii} with the candidate destination.
 - v) If q = Z, stop. Otherwise, set q = q + 1 and go to step 3e, part ii.



Figure 3: CRPSO solution representation

4 Computational Results

This section compares the performance of the developed CRPSO algorithm to PSO using problems of the same scale, and evaluates the quality of the CRPSO solution by analyzing the computational results. This section consists of three parts: benchmark instances, parameter settings and PSO dimensions, and a comparison table. We tested our research experiments using Solomon's 56 VRPTW benchmark instances [24] on a computer equipped with an Intel(R) Core(TM) i5-3210M 2.50GHz CPU and 4 GB RAM running the Microsoft Windows 7 operating system.

4.1 Benchmark Instances

We tested the proposed heuristic on three different data sets [24]. Solomon's 56 VRPTW benchmark problems consist of six sets (C1, C2, R1, R2, RC1, RC2), each of which contains between 8 and 12 problems; each data set has 100 nodes. C, R, and RC represent three different types of customer sets. C represents Clustered customers, R indicates randomly (uniformly) distributed customers, and RC represents Semi-clustered customers; that is, a combination of clustered and randomly (uniformly) distributed customers.

Moreover, C, R and RC problems can be further classified into two types: type 1 (C1, R1, RC1) problems have short time windows and small vehicle capacities, and type 2 (C2, R2, RC2) problems have long time windows and large vehicle capacities. However, the proposed problem in this paper is OVRPTW considering 3PL, so the problem set differs from Solomon's data [24]. Therefore, we divided the original 100 nodes in the experimental problem set into two groups; we assigned the first 90 nodes in each problem to customers, and the remaining 10 nodes to 3PL companies. After a vehicle makes its final delivery, it must return to the nearest 3PL location with available space. Vehicle destinations are limited to the 10 3PL nodes, but the capacity of each 3PL location is limited to three (i.e., only three vehicles can return to each 3PL location). Hence, each vehicle must be assigned to a 3PL location according to the 3PL destination priority matrix. If the nearest 3PL is full, the vehicle will return to the nearest location with available space.

4.2 Parameter Settings and PSO Dimensions

The parameter settings in PSO and CRPSO include population size as 100, iteration as 200, C_p as 2 and C_g as 2. A PSO problem with *n* customers and *m* vehicles consists of (n + m) dimensional particles. Each dimension in each particle is encoded as a real number, and the solution representation is divided into two parts. The first part is used to create a customer priority list by sorting the dimensional values. Smaller values indicate higher priority customers. The second part is same as the capacity dimension in CRPSO.

4.3 Comparison Table

The proposed CRPSO algorithm and PSO was implemented using the Visual Studio C# programming language. Some criteria can be used to evaluate the effectiveness of the developed CRPSO algorithm. One common criterion is the solution gap between the optimal solution of PSO and the best solution found by the CRPSO algorithm. The experiments verify the solution to determine the improvement rate for travel distance. The solution gap is defined as below [25].

Solution
$$Gap(\%) = \frac{S-B}{B} \times 100\%$$
 (30)

where B is the optimal solution obtained from the PSO result, and S is the optimal solution of the CRPSO algorithm.

All 56 data sets from Solomon's problems [24] are tested and the results are shown in Tab. 1. In the table, TD is travel distance, NV is the number of vehicles used, and CPU is the computational time in seconds. The objective in this research is to minimize the total travel distance, that is, TD is viewed as an indicator of solution quality that enables the PSO and CRPSO solutions to be compared. Overall, the proposed CRPSO algorithm is effective at finding the shortest path to service all customers. Compared to the PSO result, the average travel distances for all three problem types are shorter, as indicated by the solution gap. Beyond the solution gap, NV is another important index to discuss.

Figs. 4–6 indicate solution quality based on TD and NV. In problem set C, CRPSO is more efficient than PSO for most problems. Although the NV of PSO is less than the NV of CRPSO in problem 5 of subset C2, the total distance is also longer. In this case, routes with a lower NV are not the most appropriate solution. The same situation can be also observed in problem 7 of subset R1. Moreover, PSO and CRPSO have approximately equivalent solving abilities in problem set R compared with the other two problem types.

The experiment in this paper reveals two factors that can be used as comparison criteria to analyze solution quality. As shown in the comparison table, the proposed CRPSO is more feasible for solving OVRPTW considering 3PL. Whether by total travel distance or number of vehicles, CRPSO consistently outperforms PSO with respect to solution quality. Nonetheless, the number of vehicles used is slightly

higher in a few problems, as mentioned above. In the tradeoff between travel distance and number of routes, this is a reasonable result. Our approach seems to be a very practical tool that can help 3PL companies effectively schedule their daily routes.

Set	NO.	PSO		CRPSO			Gap(%)	
		TD	NV	CPU(s)	TD	NV	CPU(s)	
C1	1	5,839.49	25.00	0.09	5,829.17	23.00	0.65	-0.18
	2	5,886.57	23.00	0.08	5,811.75	23.00	0.66	-1.27
	3	5,996.79	25.00	0.08	5,833.07	25.00	0.92	-2.73
	4	6,285.30	23.00	0.09	5,599.02	20.00	0.69	-10.92
	5	6,040.21	25.00	0.10	5,887.57	23.00	0.67	-2.53
	6	6,437.06	23.00	0.09	5,820.83	21.00	0,686	-9.57
	7	6,012.17	23.00	0.09	5,666.44	23.00	0.68	-5.75
	8	6,307.69	24.00	0.08	5,772.41	22.00	0.74	-8.49
	9	6,341.36	22.00	0.10	5,872.71	20.00	0.68	-7.39
	Average	6,127.40	23.67	0.09	5,788.11	22.22	0.63	-5.42
C2	1	6,376.06	24.00	0.12	5,762.44	24.00	0.69	-9.62
	2	6,110.67	22.00	0.09	5,971.94	22.00	0.66	-2.27
	3	6,208.32	24.00	0.09	5,988.48	23.00	0.75	-3.54
	4	6,470.55	23.00	0.07	5,948.77	22.00	0.67	-8.06
	5	6,636.96	21.00	0.09	5,967.21	22.00	0.70	-10.09
	6	6,171.45	22.00	0.09	5,983.65	22.00	0.73	-3.04
	7	6,098.70	24.00	0.09	5,895.62	23.00	0.96	-3.33
	8	6,110.15	24.00	0.11	5,889.13	24.00	0.86	-3.62
	Average	6,272.86	23.00	0.09	5,925.91	22.75	0.75	-5.45
R1	1	6,027.03	21.00	0.08	5,901.64	20.00	0.65	-2.08
	2	6,001.92	22.00	0.13	5,942.11	21.00	0.84	-1.00
	3	6,134.95	22.00	0.16	5,687.43	20.00	0.80	-7.29
	4	6,378.26	21.00	0.17	5,993.17	21.00	0.87	-6.04
	5	6,816.80	19.00	0.11	6,169.61	18.00	0.89	-9.49
	6	6,137.93	22.00	0.11	5,769.13	21.00	0.97	-6.01
	7	6,648.73	21.00	0.22	5,660.74	22.00	1.80	-14.86
	8	6,359.76	22.00	0.67	6,214.38	21.00	0.97	-2.29
	9	6,244.31	21.00	0.36	6,021.46	20.00	0.93	-3.57
	10	5,982.76	23.00	0.13	5,938.51	23.00	0.76	-0.74
	11	6,217.68	22.00	0.15	5,907.13	20.00	0.92	-4.99
	12	6,151.80	21.00	0.09	6,078.12	20.00	0.70	-1.20
	Average	6,258.49	21.42	0.20	5,940.29	20.58	0.93	-4.96

Table 1: Experimental result of PSO and CRPSO

(Continued)

Table 1 (continued).								
Set	NO.	PSO				CRPSO		Gap(%)
		TD	NV	CPU(s)	TD	NV	CPU(s)	
R2	1	6,062.57	22.00	0.12	5,984.24	20.00	0.74	-1.29
	2	6,490.04	21.00	0.11	6,017.24	20.00	0.81	-7.29
	3	6,271.16	23.00	0.15	5,912.37	23.00	0.71	-5.72
	4	6,330.67	21.00	0.10	6,178.23	20.00	0.84	-2.41
	5	5,888.25	22.00	0.10	5,716.92	21.00	0.96	-2.91
	6	5,975.79	22.00	0.10	5,864.48	21.00	0.79	-1.86
	7	6,285.56	20.00	0.09	5,873.82	20.00	1.03	-6.55
	8	6,009.84	23.00	0.10	5,921.78	22.00	0.72	-1.47
	9	6,158.43	22.00	0.09	5,843.92	21.00	0.74	-5.11
	10	6,102.55	22.00	0.10	6,048.70	21.00	0.73	-0.88
	11	6,555.39	21.00	0.11	6,324.54	21.00	0.84	-3.52
	Average	6,193.66	21.73	0.11	5,971.48	20.91	0.81	-3.55
RC1	1	7,668.69	23.00	0.09	7,591.64	23.00	0.70	-1.00
	2	7,846.58	23.00	0.28	7,647.89	22.00	0.77	-2.53
	3	7,393.69	25.00	0.08	7,228.71	24.00	0.64	-2.23
	4	7,362.31	25.00	0.13	7,312.96	23.00	0.67	-0.67
	5	7,689.42	23.00	0.28	7,384.49	22.00	0.66	-3.97
	6	7,785.82	23.00	0.12	7,672.44	23.00	0.70	-1.46
	7	7,509.93	25.00	0.22	7,501.02	24.00	0.75	-0.12
	8	7,501.88	24.00	0.17	7,476.89	24.00	0.70	-0.33
	Average	7,594.79	23.88	0.17	7,477.01	23.13	0.70	-1.54
RC2	1	7,373.82	25.00	0.10	7,326.41	23.00	0.68	-0.64
	2	7,884.37	23.00	0.18	7,231.79	22.00	0.84	-8.28
	3	7,696.52	24.00	0.08	7,375.69	24.00	0.76	-4.17
	4	7,364.79	24.00	0.11	6,885.50	23.00	0.94	-6.51
	5	7,739.94	23.00	0.13	7,381.47	25.00	0.86	-4.63
	6	7,571.78	23.00	0.09	7,269.71	19.00	0.98	-3.99
	7	7,763.32	24.00	0.20	7,282.05	24.00	0.80	-6.20
	8	7,363.41	24.00	0.09	7,349.42	24.00	0.89	-0.19
	Average	7,594.74	23.75	0.12	7,262.75	23.00	0.84	-4.33





5 Conclusion

This paper presented a solution to an open vehicle routing problem with time windows considering 3PL. The delivery vehicles are operated by a 3PL company, and return to the nearest 3PL company location with available space once deliveries are complete. This paper modified the mixed integer linear programming model used by Repoussis et al. (2007) considering standard constraints of OVRPTW and 3PL [7]. Due to computational complexity, a CRPSO algorithm to obtain the near-optimal solution is developed. Results of the computational study show that the proposed CRPSO algorithm provides solutions within a reasonable amount of time. The encoding method yields the optimal distribution so that the delivery quantity for each route does not exceed vehicle capacity. Furthermore, the proposed algorithm reduces the number of vehicles used to make deliveries to customers. Finally, the PSO mechanism can generate multiple solutions and continues to iteratively search for the best solution. In terms of future research directions, this research can be expanded to add more locations for each 3PL company so that the vehicle has more destination options. The utility of such an expansion should be investigated.

Funding Statement: All the authors received no specific funding for this study.

Conflicts of Interest: All the authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Hertz and M. Alfredsson, "Strategic development of third-party logistics providers," *Industrial Marketing Management*, vol. 32, no. 2, pp. 139–149, 2003.
- [2] J. B. Quinn, "Strategic outsourcing: Leveraging knowledge capabilities," *Sloan Management Review*, vol. 40, no. 4, pp. 9–21, 1999.

- [3] M. Christopher, Logistics & supply chain management. New York: Pearson Publishing, 2016.
- [4] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," Management Science, vol. 6, no. 1, pp. 80–91, 1959.
- [5] K. H. Leung, C. K. Lee and K. L. Choy, "An integrated online pick-to-sort order batching approach for managing frequent arrivals of B2B e-commerce orders under both fixed and variable time-window batching," *Advanced Engineering Informatics*, vol. 45, pp. 101–125, 2020.
- [6] J. E. Bell and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Advanced Engineering Informatics*, vol. 18, no. 1, pp. 41–48, 2004.
- [7] P. P. Repoussis, C. D. Tarantilis and G. Ioannou, "The open vehicle routing problem with time windows," *Journal of the Operational Research Society*, vol. 58, no. 3, pp. 355–367, 2007.
- [8] P. P. Repoussis, D. C. Paraskevopoulos, G. Zobolas, C. D. Tarantilis and G. Ioannou, "A web-based decision support system for waste lube oils collection and recycling," *European Journal of Operational Research*, vol. 195, no. 3, pp. 676–700, 2009.
- [9] A. Baykasoglu and V. Kaplanoglu, "A multi-agent approach to load consolidation in transportation," *Advances in Engineering Software*, vol. 42, no. 7, pp. 477–490, 2011.
- [10] P. Amorim, H.-O. Günther and B. Almada-Lobo, "Multi-objective integrated production and distribution planning of perishable products," *International Journal of Production Economics*, vol. 138, no. 1, pp. 89–101, 2012.
- [11] I. Moon, J.-H. Lee and J. Seong, "Vehicle routing problem with time windows considering overtime and outsourcing vehicles," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13202–13213, 2012.
- [12] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [13] L. Schrage, "Formulation and structure of more complex/realistic routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 229–232, 1981.
- [14] D. Sariklis and S. Powell, "A heuristic method for the open vehicle routing problem," *Journal of the Operational Research Society*, vol. 51, no. 5, pp. 564–573, 2000.
- [15] X. Li, P. Tian and S. C. Leung, "An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems," *Journal of the Operational Research Society*, vol. 60, no. 7, pp. 1012–1025, 2009.
- [16] K. Fleszar, I. H. Osman and K. S. Hindi, "A variable neighbourhood search algorithm for the open vehicle routing problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 803–809, 2009.
- [17] T. J. Ai and V. Kachitvichyanukul, "Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 380–387, 2009a.
- [18] T. J. Ai and V. Kachitvichyanukul, "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery," *Computers & Operations Research*, vol. 36, no. 5, pp. 1693–1702, 2009b.
- [19] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory. Paper presented at the MHS'95," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.
- [20] J. Clarke, L. McLay and J. T. McLeskey Jr, "Comparison of genetic algorithm to particle swarm for constrained simulation-based optimization of a geothermal power plant," *Advanced Engineering Informatics*, vol. 28, no. 1, pp. 81–90, 2014.
- [21] A. Elbir, H. Ilhan and N. Aydın, "The implementation of optimization methods for contrast enhancement," *Computer Systems Science and Engineering*, vol. 34, no. 2, pp. 101–107, 2019.
- [22] X. Wang, S. Liu and L. Zhang, "Highway cost prediction based on LSSVM optimized by initial parameters," *Computer Systems Science and Engineering*, vol. 36, no. 1, pp. 259–269, 2021.
- [23] S. Nguyen, T. J. Ai and V. Kachitvichyanukul, *Object library for evolutionary techniques ETLib: User's guide.* Thailand: High Performance Computing Group, Asian Institute of Technology, 2010.
- [24] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," Operations Research, vol. 35, no. 2, pp. 254–265, 1987.
- [25] T.-L. Chen, J.-T. Lin and S.-C. Fang, "A shadow-price based heuristic for capacity planning of TFT-LCD manufacturing," *Journal of Industrial and Management Optimization*, vol. 6, no. 1, pp. 209–241, 2010.