EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Green vehicle routing problem with mixed and simultaneous pickup and delivery, time windows and road types using self-adaptive learning particle swarm optimization

Nitikorn Srijaroon[1)], Kanchana Sethanan*[1)], Thitipong Jamrus[1)] and Chen-Fu Chien[2)]

[1)]Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand
[2)]Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan

Received 16 November 2020
Revised 14 February 2021
Accepted 22 March 2021

Abstract

This research focuses on the third-party logistics (3PL) management in sustainable reverse logistics industry that involves fuel consumption and emission concerns based on the comprehensive modal emission model (CMEM) in transportation operations on either deliver finished products to customers or pick-up malfunctioned/expired products or perform both operations for recycling or waste management at the depot. We formulated a novel mixed-integer linear programming (MILP) model for an extension of the green vehicle routing problem with mixed and simultaneous pickup and delivery problem, time windows, and road types (G-VRPMSPDTW-RT) that yields optimal solutions and proposed a self-adaptive learning particle swarm optimization (SAL-PSO) to improve the quality of solutions in large problems. Our work aims to minimize total transportation costs, including fuel consumption costs and driver costs. The validation of SAL-PSO was conducted by the comparison of the optimal solutions obtained from CPLEX and the best solutions obtained from the standard and proposed meta-heuristics. The relative improvement (RI) between the standard PSO and the SAL-PSO in the G-VRPMSPDTW-RT was 0.15-7.31%. The SAL-PSO outperformed the standard PSO by the average of 3.25%.

Keywords: Mixed and simultaneous pickup and delivery, Sustainable reverse logistics, Particle swarm optimization, Self-adaptive learning

1. Introduction

In 2018, Business-to-Consumer (B2C) e-commerce in reverse logistics was worth around 3 billion Baht and was growing, because of the free shipping and flexible return policies that facilitate the return of products in case of failed deliveries and customer returns. Thus, there are opportunities for this kind of business to expand in the future [1].

In the period of the recent COVID-19 pandemic, which has had an impact on global supply chains, there is now considerable concern about how businesses can support customers by providing goods and services via a logistics network in such a crisis. In Thailand, the third-party logistics (3PL) industry has been growing in terms of parcel delivery, since the increase of e-commerce markets as a result of consumer behavior changes during the lockdowns in some specific areas at high risk of an outbreak and the work-from-home policy in order to maintain social distancing [2].

In sustainable reverse logistics, besides integrating customer demand operations into effective transportation, environmental impacts are also considered. Road transportation is the main transportation mode used in the 3PL services in Thailand [3]. Although the lockdown relaxation started in the middle of May 2020 and some businesses have reopened, diesel fuel still has the highest market share at 58% of the fuel consumption of road transportation [4]. It seems that pollution emissions are bouncing back gradually, which affects sustainable transportation in the future.

Moreover, in urban areas traffic congestion plays an essential role in aggressive driving behavior that affects the high potential fuel consumption of transportation [5]. Sometimes, drivers need to increase their drive speed to serve customers and meet their time windows. Although this can guarantee the quality of service to customers, it can cause high pollution to the environment. Therefore, speed limitation for road type is defined for healthier driving behavior.

The vehicle routing problem with simultaneous pickup and delivery (VRPSPD) was first introduced by Min [6]. It is an extension of the vehicle routing problem with pickup and delivery (VRPPD) where customers can receive and send goods simultaneously. Later, Nagy and Salhi [7] introduced the vehicle routing problem with mixed pickup and delivery (VRPMPD) which is similar to VRPSPD, but customers can either send or receive goods at once. Both VRPSPD and VRPMPD reflect customer demands in the operations of reverse logistics by distributing products to customers and picking up customer return items at lower costs. Several authors have attempted to solve these problems in many fields. For example, Lin et al. [8] proposed a genetic algorithm-based optimization model (GOM) for VRPSPD in the green transportation of filled water carboys. Osaba et al. [9] designed a discrete firefly algorithm to solve a rich vehicle routing problem in a real-world newspaper distribution system with recycling policy.
According to the survey of Green Vehicle Routing Problem by Lin et al. [10], green logistics has been much taken into account recently in different aspects: Green-VRP minimizes the energy consumption of transportation; Pollution Routing Problem (PRP) reduces fuel

*Corresponding author. Tel.: +668 1553 6429
Email address: skanch@kku.ac.th; ksethanan@gmail.com
doi: 10.14456/easr.2021.66

consumption, greenhouse gas (GHG) emissions, and driver costs in road transportation; and VRP in Reverse Logistics (VRP-RL) deals with forward and backward product flow for recycling end of life products or waste management, respectively in order to make the supply chain management more sustainable. The previous literatures that relate to all three aspects are as follows. Bloemhof-Ruwaard, van Beek, Hordijk, and Van Wassenhove [11] revealed that there was the environmental awareness of the routing in reverse logistics in their work. Sbihi and Eglese [12] introduced a time-dependent VRP for green logistics of waste management that minimizes emissions. Kuo [13] proposed total fuel consumption for the time-dependent vehicle routing problem (TDVRP) where speed depends on begin time. Bektas and Laporte [14] introduced PRP, which is an extension of the classical VRP with constant travel speed as a decision variable. This model considers the amount of greenhouse emission, fuel consumption, travel time and driver costs. Next, Erdogan and Miller Hooks [15] proposed G-VRP that finds refueling stops at alternative fuel stations (AFSs) with minimal travel cost. Then, Xiao et al. [16] developed the fuel consumption rate considered capacitated VRP (FCVRP) to minimize fuel consumption. Later, Franceschetti et al. [17] proposed the time-dependent PRP, which aims to minimize the emission and driver costs regarding traffic congestion and vehicle speed. Moreover, Demir and Woensel [18] implemented the multi-vehicle, multi-depot one-to-one pickup and delivery PRP (PDPRP). It is an extension of the classical VRPPD, and time windows, which minimizes the total fuel and driver costs. Xiao et al. [19] not only introduced green vehicle routing and scheduling problem (GVRSP) under time-varying traffic conditions to minimize vehicle emissions, but they [20] also developed a genetic algorithm with the exact dynamic programming procedure (GA-DP) for the time-dependent vehicle routing & scheduling problem with CO2 emissions optimization (TD-VRSP-CO2). Also, Poonthalir and Nadarajan [21] developed a bi-objective Fuel efficient Green Vehicle Routing Problem (F-GVRP) with varying speed solved using Particle Swarm Optimization with Greedy Mutation Operator and Time varying acceleration coefficient (TVa-PSOGMO). For pickup and delivery problem (PDP), Soysal et al. [22] introduced the green one-to-one PDP with road segmentation to improve the PDP with vehicle speed and road category in urban and non-urban areas to minimize the total costs. For reverse logistics problem, Tuntitippawan and Asawarungsaengkul [23] developed a vehicle routing problem with backhauls and time windows (VRPBTW) to minimize the total of route distance by using artificial bee colony (ABC) algorithm with local search. Sethanan and Jamrus [24] proposed a hybrid differential evolution algorithm and genetic operator for multi-trip vehicle routing problem with backhauls and heterogeneous fleet in the beverage logistics industry. In terms of VRPSPD/VRPMPD, not much has been implemented in this area. Majidi et al. [25] presented a non-linear mixed integer programming model and the adaptive large neighborhood search heuristic for the pollution routing problem with simultaneous pickup and delivery (PRPSPD) to minimize fuel consumption and emissions.

According to the qualitative comparison of GA, PSO, and DE by Kachitvichyanukul [26], the drawbacks of GA are its high influence of population size on solution time and it fails to evaluate fitness function due to very complex high dimensional problem and high scale of iterations [27]. Also, the disadvantages of DE are its less influence of best solution on population and its limited effect on solution quality regarding sub-grouping with homogeneous population. Although PSO has tendency of premature convergence, many studies have been published on VRP with Simultaneous Pickup and Delivery (VRPSPD) and VRP with Mixed Pickup and Delivery (VRPMPD) solved by modified PSO. The following authors have successfully implemented modified PSO in their research with significant improvement. Ai and Kachitvichyanukul [28] proposed a modified PSO with multiple social learning structures: global best, local best and near neighbor best (GLNPSO) for VRPSPD. Goksal et al. [29] developed a hybrid discrete PSO for the VRPSPD. Kachitvichyanukul et al. [30] introduced the generalized multi-depot VRP with multiple pickup and delivery requests (GVRP-MDMPDR) which was applied to the GLNPSO to solve the problem. However, only a few applied Green VRP to VRPSPD/VRPMPD. Norouzi et al. [31] implemented the modified PSO in a time-dependent VRP to minimize fuel consumption. Also, Li et al. [32] developed a G-VRP model based on modified PSO for cold chain logistics to minimize total costs. Interestingly, Zhan et al. [33] introduced Adaptive PSO (APSO) approach to automatically control PSO parameters e.g., inertia weight, acceleration coefficients, and other parameters to improve exploration ability and convergence speed so that the global best particle can avoid local optima. Wang et al. [34] proposed self-adaptive learning PSO (SLPSO) with the probability of selecting four PSO based search strategies in the economic load dispatch problem of power systems (ELD). Xu [35] developed an adaptive parameter tuning of particle swarm optimization based on velocity information (APSO-VI) by adjusting the inertia weight according to average absolute velocity and non-linear ideal velocity to avoid the local optima and improve the convergence speed. Pornsing et al. [36] presented two novel adaptive PSO approaches: survival sub-swarms adaptive PSO (SSS-APSO) and survival sub-swarms adaptive PSO with velocity-line bouncing (SSS-APSO-vb) which obtain best solution and converge to the optima more quickly. A few studies of adaptive PSO relating to VRP. Marinakis et al. [37] proposed three adaptive strategies: Greedy Randomized Adaptive Search Procedure (GRASP) for initial solutions, Adaptive Combinatorial Neighborhood Topology for particle movement, and all adaptive parameters used in Multi-Adaptive Particle Swarm Optimization (MAPSO) to solve VRP with TW.

From the literature review of the previous studies, many authors conducted different PSO approaches to solve VRPSPD and VRPMPD without the adaptive parameters. However, there is still room for improvement in the extension of green VRP with mixed and simultaneous pickup and delivery, time windows, and road types (G-VRPMSPDTW-RT) by using this adaptive PSO approach, because it reflects the characteristics of real-world reverse logistics with dynamic customer demands for over the horizon planning.

2. The green vehicle routing problem with mixed and simultaneous pickup and delivery problem, time windows and road types

2.1 Assumptions and constraints

We propose a mixed integer linear programming (MILP) model for the G-VRPMSPDTW-RT, and a solution to the problem is to minimize the total transportation costs including fuel consumption costs and driver operation costs. It consists of a set of routes such that:

- All vehicle routes include mixed and simultaneous pickup and delivery nodes.
- A fleet of vehicles is composed of single unit vehicles and the number is limited.
- All pickup and delivery demands of a single commodity, units of time (including customer service time, customer time windows and maximum time duration), and travel distance of all nodes are non-negative deterministic values.
- The depot is allowed to have all delivery demands which are less than or equal to the vehicle capacity, as the initial loads of each vehicle leave to the first customer.
- Each vehicle returns to the depot with the total of pickup demands of en route customers.
- Each customer can have mixed or simultaneous pickup and delivery requests.

- All pickup and delivery demands must be served before returning to the depot.
- All fuel consumption characteristics used must refer to the comprehensive modal emission model (CMEM) in Section 2.2.
- The time dependent travel time used for this problem depends on vehicle speed level and road types announced by the Ministry of Transport, Thailand.

2.2 CMEM Model

In this research, we applied a fuel consumption model of Demir et al. [38] and Demir and Woensel [18] based on the comprehensive modal emissions model (CMEM) of Barth et al. [39] and Barth and Boriboonsomsin [40] to the green vehicle routing problem with mixed and simultaneous pickup and delivery, time windows and road types, because this model represents a fuel consumption rate in relation to vehicle load fluctuation and varying travel time along the entire route. The fuel consumption can be calculated by the CMEM model as follows:

$$F(v) = \lambda(e_k N_k V_k + w_k \gamma \alpha v + \gamma \alpha f v + \beta_k \gamma v^3)\frac{d}{v} \tag{1}$$

where $\lambda = \frac{\xi}{\hbar\psi}$ and $\gamma = \frac{1}{1000\eta_{tf}\eta}$ are constants.
$\alpha = gsin\theta + gC_r cos\theta$ is a vehicle-arc specific constant.
$\beta_k = 0.5C_d \rho A_k$ is a vehicle-arc specific constant.
The format of parameter definitions is as follows Table 1.

Table 1 Parameters for a comprehensive modal emission model (CMEM)

Parameters	Definition	Value(s)
w_k*	Curb-weight (kg)	6,350-11,793
ξ	Fuel-to-air mass ratio	1
e_k	Engine friction factor $(kJ/rev/liter)$	0.2-0.25
N_k	Engine speed (rev/s)	33-51
V_k	Engine displacement $(liters)$	5-7
g	Gravitational constant (m/s^2)	9.81
C_d	Coefficient of aerodynamic drag	0.7
ρ	Air density (kg/m^3)	1.2041
A_k	Frontal surface area (m^2)	3.912-5.88
C_r	Coefficient of rolling resistance	0.01
θ	Slope of the road (radians)	0
η_{tf}	Vehicle drive train efficiency	0.4
η	Efficiency for diesel engines	0.9
f_c*	Cost of fuel and carbon dioxide equivalents (CO_2e) emission per liter $(baht)$	19.04
f_d*	Driver wage $(baht/s)$	0.02315
$\hbar$	Heating value of a typical diesel fuel (kJ/g)	44.32
ψ	Conversion factor $(g/s\ to\ L/s)$	737
$v_{i,j}^{min}$*	Lower speed limit (km/h)	20
$v_{i,j}^{max}$*	Upper speed limit (km/h)	120

Note: refers to parameters adjusted to the transportation environment in Thailand.
k subscript refers to each vehicle type

Regarding the optimal speed levels of CMEM model, Franceschetti et al. [17] suggested two optimal speed levels that affect fuel emissions: the upper optimal speed level $(\overline{v_{i,j}})$ that minimizes both fuel consumption and driver costs and the lower optimal speed level $(\underline{v_{i,j}})$ that only minimizes fuel consumption.

$$\overline{v_{i,j}} = ((f_c \lambda e_k N_k V_k + d_c)/2f_c \lambda \beta \gamma)^{1/3} \tag{2}$$

$$\underline{v_{i,j}} = (e_k N_k V_k / 2\beta\gamma)^{1/3} \tag{3}$$

2.3 Mathematical formulation of the G-VRPMSPDTW-RT

The mathematical formulation is presented below following the preceding definitions of parameters, indices, and decision variables.
Indices:

- i, j The index of vertices, pickup and delivery operations; $i = 1,2,3, \dots, N$.
- k The index of vehicles; $k = 1,2,3, \dots, M$.
- r The speed level allowed for road type; $r = 1,2,3, \dots, S$.

Sets:

- V The set of the vertices; $V = V_0 \cup P \cup D = \{v_0, v_1, v_2, \dots, v_N\}$
- V_0 The set of the beginning and returning of the same depot; $V_0 = \{v_0, v_{N+1}\}$
- V_c The set of the customers; $V_c = P \cup D = \{v_1, v_2, v_3, \dots, v_N\}$
- K The set of the vehicles; $K = \{k_{1,} k_{2,} k_{3,} \dots k_{M,}\}$
- Q The set of the vehicle capacities; $Q = \{q_{1,} q_{2,} q_{3,} \dots q_{M,}\}$

P The set of the pickup requests; $P = \{p_{1,}, p_{2,}, p_{3,} \dots, p_{N,}\}$
D The set of the delivery requests; $D = \{d_{1,} d_{2,} d_{3,} \dots d_{N,}\}$
R The set of the speed levels; $R = \{r_{1,} r_{2,} r_{3,} \dots v_{S,}\}$

Parameters:

N Maximum number of vertices
K Maximum number of vehicles
R Maximum number of segment lines
q_k The capacity of vehicle k in kilograms (kg)
$D_{i,j}$ The distance in meters (m) from customer i to customer j
p_i The amount of goods in kilograms (kg) to be picked up from customer i
d_i The amount of goods in kilograms (kg) to be delivered to customer i
o_i The open time of the customer operation i in seconds (s)
c_i The close time of the customer operation i in seconds (s)
s_i The service time for customer operation i in seconds (s)
$v_{i,j}^{min}$ The minimum speed limit for arc i to j in seconds (s)
$v_{i,j}^{max}$ The maximum speed limit for arc i to j in seconds (s)
v_r A non-decreasing average drive speed level in meters per second (m/s).
f_c The cost of fuel consumption $(Baht/litre)$
d_c The driver wage $(Baht/second)$

Decision variables:

$X_{i,j,k}$ A binary variable that takes the value 1 if the route of vehicle k is between customer i and j; 0 otherwise.
$Z_{i,j,k,r}$ A binary variable that takes the value 1 if the arc (i,j) is traversed via a vehicle k at a speed level r; 0 otherwise.
U_i The sub-tour variable after serving customer i.
$l_{v_0,k}$ The initial loads of vehicle k leaving the depot in kilograms (kg).
l_i The loads of vehicle k after serving customer i in kilograms (kg).
b_i The begin time at which vehicle k starts to service customer i in seconds (s).
w_i The wait time at which vehicle k starts to delay at customer i before travelling to the next customer in seconds (s).

Objective function:

$$Minimize \quad \sum_{k\in K} \sum_{i\in V} \sum_{j\in V} e_k N_k V_k \lambda f_c \left(D_{i,j} \sum_{r\in R} \frac{Z_{i,j,k,r}}{v_r} \right) \tag{4.1}$$

$$+ \sum_{k\in K} \sum_{i\in V} \sum_{j\in V} w_k \gamma \lambda f_c \alpha_{i,j} D_{i,j} X_{i,j,k} \tag{4.2}$$

$$+ \sum_{k\in K} \sum_{i\in V} \sum_{j\in V} \gamma \lambda f_c \alpha_{i,j} D_{i,j} X_{i,j,k} \left(L_{v_0,k} + L_{i,k} \right) \tag{4.3}$$

$$+ \sum_{k\in K} \sum_{i\in V} \sum_{j\in V} \beta_k \gamma \lambda f_c \left(D_{i,j} \sum_{r\in R} Z_{i,j,k,r} v_r^2 \right) \tag{4.4}$$

$$+ \sum_{k\in K} \sum_{i\in V} \sum_{j\in V} \left(D_{i,j} \sum_{r\in R} \frac{Z_{i,j,k,r}}{v_r} \right) d_c + \sum_{i\in V} (s_i + w_i) d_c \tag{4.5}$$

Subject to:

$$\sum_{i\in V} \sum_{k\in K} X_{i,j,k} = 1 \qquad ; \forall j \in V_c \tag{5}$$

$$\sum_{i\in V} X_{i,h,k} - \sum_{j\in V} X_{h,j,k} = 0 \qquad ; \forall k \in K, \forall h \in V_c \tag{6}$$

$$\sum_{j\in V_c} X_{v_0,j,k} \leq 1 \qquad ; \forall k \in K \tag{7}$$

$$\sum_{j\in V_c} X_{i,v_0,k} \leq 1 \qquad ; \forall k \in K \tag{8}$$

$$l_{v_0,k} = \sum_{i\in V} \sum_{j\in V_c} d_j X_{i,j,k} \qquad ; \forall k \in K \tag{9}$$

$$l_j \geq l_{v_0,k} - d_j + p_j - M\left(1 - X_{v_0,j,k}\right) \quad ; \forall k \in K, \forall j \in V_c \tag{10}$$

$$l_j \geq l_i - d_j + p_j - M\left(1 - \sum_{k \in K} X_{i,j,k}\right) \quad ; \forall k \in K, \forall i,j \in V_c, i \neq j \tag{11}$$

$$l_{v_{N+1}} = \sum_{i \in V_c} \sum_{j \in V} p_i X_{i,j,k} \quad ; \forall k \in K \tag{12}$$

$$l_{v_0,k} \leq q_k \quad ; \forall k \in K \tag{13}$$

$$l_j \leq q_k + M\left(1 - \sum_{v \in V} X_{i,j,k}\right) \quad ; \forall k \in K, \forall j \in V, i \neq j \tag{14}$$

$$\sum_{r \in R} Z_{i,j,k,r} = X_{i,j,k} \quad ; \forall k \in K, \forall i,j \in V_c, i \neq j, \forall r \in R \tag{15}$$

$$v_{i,j}^{min} \leq \sum_{r \in R} Z_{i,j,k,r} \cdot v_r \leq v_{i,j}^{max} \quad ; \forall k \in K, \forall i,j \in V_c, i \neq j, \forall r \in R \tag{16}$$

$$b_i + s_i + w_i + D_{i,j} \sum_{r \in R} \frac{Z_{i,j,k,r}}{v_r} - M\left(1 - X_{i,j,k}\right) \leq b_j \quad ; \forall k \in K, \forall i,j \in V, i \neq j \tag{17}$$

$$b_j + s_j + w_i + D_{i,v_0} \sum_{r \in R} \frac{Z_{i,v_0,k,r}}{v_r} - M\left(1 - X_{j,v_0,k}\right) \leq c_0 \quad ; \forall k \in K, \forall j \in V \tag{18}$$

$$o_i \leq b_i \leq c_i \quad ; \forall i \in V \tag{19}$$

$$U_j \geq U_i + 1 - N\left(1 - \sum_{k \in K} X_{i,j,k}\right) \quad ; \forall i,j \in V_c \tag{20}$$

$$X_{i,j,k} \in \{0,1\} \quad ; \forall k \in K, \forall i,j \in V \tag{21}$$

$$Z_{i,j,k,r} \in \{0,1\} \quad ; \forall k \in K, \forall i,j \in V, r \in R \tag{22}$$

$$l_{v_0,k} \geq 0; \quad ; \forall k \in K, \forall i \in V \tag{23}$$

$$l_i \geq 0;\ U_i \geq 0 \quad ; \forall i \in V \qquad (24)\text{ - }(25)$$

$$b_i \geq 0; w_i \geq 0 \quad ; \forall i \in V\ , \forall k \in K \qquad (26)\text{ - }(27)$$

The objective function (4) aims to minimize the total transportation costs including fuel consumption costs and driver operation costs which includes the engine-dependent fixed consumption (4.1), the curb weight related consumption (4.2), the payload related consumption (4.3), the aerodynamic drag related consumption (4.4) and the driver costs (4.5). Constraint (5) ensures that each customer is visited only once. Constraint (6) controls vehicle flow balance. Constraint (7) minimizes the number of vehicles starting from the depot. Constraint (8) minimizes the number of vehicles returning to the depot. Constraint (9) allows the initial loads equal to all *en route* delivery loads. Constraint (10) controls the flow of goods loads of the first customer. Constraint (11) controls the flow of goods loads of *en route* customers. Constraint (12) allows the return loads equal to all *en route* pickup loads. Constraint (13) states the initial loads must not exceed the vehicle capacity. Constraint (14) states each *en route* load must not exceed the vehicle capacity. Constraint (15) ensures a vehicle traverses over an arc with only one drive speed level and Constraint (16) ensures that the drive speed level must not exceed its minimum and maximum speed limit. Constraint (17) states the start time of the current customer i adding the travel time, the service time, and the wait time must be less than or equal to the start time of the next customer$j$, while Constraint (18) states the start time of the last customer j adding the travel time, the service time, and the wait time to the depot must be less than or equal to the closing time of the depot. Constraint (19) ensures the arrival time must satisfy the customer time window. Constraint (20) is sub-tour elimination constraint. Constraints (21)-(22) are binary decision variables constraint while Constraints (23)-(27) are continuous decision variables constraints.

2.4 The constructive heuristic of the G-VRPMSPDTW-RT

All customers must be served by sorting their opening service time in ascending order with regards to delivery demands in descending order, and pickup demands in ascending order. Then, choose the available minimum vehicle capacity first to start a route serving both mixed and simultaneous pickup and delivery demands as shown in Algorithm 1. The travel speed is calculated for each arc with regards to both the minimum and maximum speed limitations in order to minimize the degree of energy consumption used by CMEM parameters Eq. (1). Although all vehicle capacities, depot and customer time windows constraints are met in all feasible routes, the initial solutions obtained from this method are ineffective because there are many factors affecting the total transportation costs in terms of fuel consumption at different road types as well as driver operations, including drive time, wait time and service time which make this problem more complex.

Algorithm 1: The route feasibility check.

$\boldsymbol{input}$: $an\ assigned\ vehicle\ (k), a\ feasible\ route (r), a\ current\ customer (x)$
$\boldsymbol{output}$: $feasibility: TRUE\ if\ feasible; otherwise\ FALSE, a\ feasible\ route\ (r)$
$\boldsymbol{procedure}$
    $customers \leftarrow r + x$
    $b_i \leftarrow o_{depot}$
    $feasibility\ \leftarrow FALSE$
    $\boldsymbol{for}\ i, j\ in\ customers\ \boldsymbol{do}$
        $D \leftarrow \sum d_j$
        $P \leftarrow \sum p_j$
        $\boldsymbol{if}\ (D < q_k) AND\ (P < q_k)\ \boldsymbol{then}$
            $l \leftarrow D$
            $\boldsymbol{if} \left(l_i - d_j + p_j < q_k\right) \boldsymbol{then}$
                $l_j \leftarrow l_i - d_j + p_j$
                $v_{i,j} \leftarrow \overline{v_{i,j}}$ Eq. (2)
                $\boldsymbol{if}\ \left(v_{i,j} \geq v_{i,j}^{max}\right) \boldsymbol{then}$
                    $v_{i,j} \leftarrow v_{i,j}^{max}$
                $\boldsymbol{else}$
                    $v_{i,j} \leftarrow \underline{v_{i,j}}$ Eq. (3)
                $t_{i,j} \leftarrow euclidean\_distance(i,j)/v_{i,j}$
                $\boldsymbol{if}\ \left(\max_{c \in r}\{o_j, b_i + t_{i,j}\} < c_j\right)\ \boldsymbol{then}$
                    $w_i \leftarrow \max_{c \in r}\{o_j - b_i - t_{i,j}, 0\}$
                    $\boldsymbol{if}\ (b_i + w_i < c_i)\ \boldsymbol{then}$
                        $b_j \leftarrow b_i + w_i + s_j + t_{i,j}$
                        $feasibility\ \leftarrow TRUE$
                        $r\ \leftarrow j$
                        $TC \leftarrow \boldsymbol{transport\_costs}(r)$
        $\boldsymbol{if}\ \left(b_j \geq c_{depot}\right) \boldsymbol{then}$
            $feasibility\ \leftarrow FALSE$
    $\boldsymbol{end\ for}$
    $return\ feasibility, r$
$\boldsymbol{end\ procedure}$

3. The SAL-PSO for G-VRPMSPDTW-RT

The PSO was first introduced by Kennedy and Eberhart [41]. The standard PSO comprises three methods including initial solutions, velocity updates and position updates. In this research, the optimal solutions of large-scale problem instances could not be obtained by the MILP optimizer. Therefore, a self-adaptive learning particle swarm optimization algorithm is proposed for solving the formulation of G-VRPMSPDTW-RT that is described in Section 3.1.

3.1 Initial solutions

There are two methods involved in the initial solutions of SAL-PSO.

3.1.1 The particle representation method

The position of each particle is generated by a composition of two vectors with the number of customers as dimensions: an integer random vector as a decimal part representing a sequence of vehicle types (small and large) combined with a uniform random vector as a fractional part representing a sequence of customers. This encoding improves the better exploration of the search space in terms of vehicle assignment for a swarm. In addition, the velocity of each particle is initialized with a zero vector as depicted in Figure 1.

3.1.2 The particle decoding method

The particle position will be grouped by the decimal values as vehicle types and the rank of order value rearranged by considering the fractional values in ascending order as a customer service sequence. Next, a route feasibility check will be performed by verifying mixed and simultaneous pickup and delivery loads, vehicle capacity, time windows as well as speed limits for road type constraints. After that, total transportation costs will be calculated by using the CMEM model in Eq. (1). The swarm will select the minimum transportation costs among the personal best values of all particles to be a global best value and proceed to the next iteration of the optimization process until the stopping criteria are met.

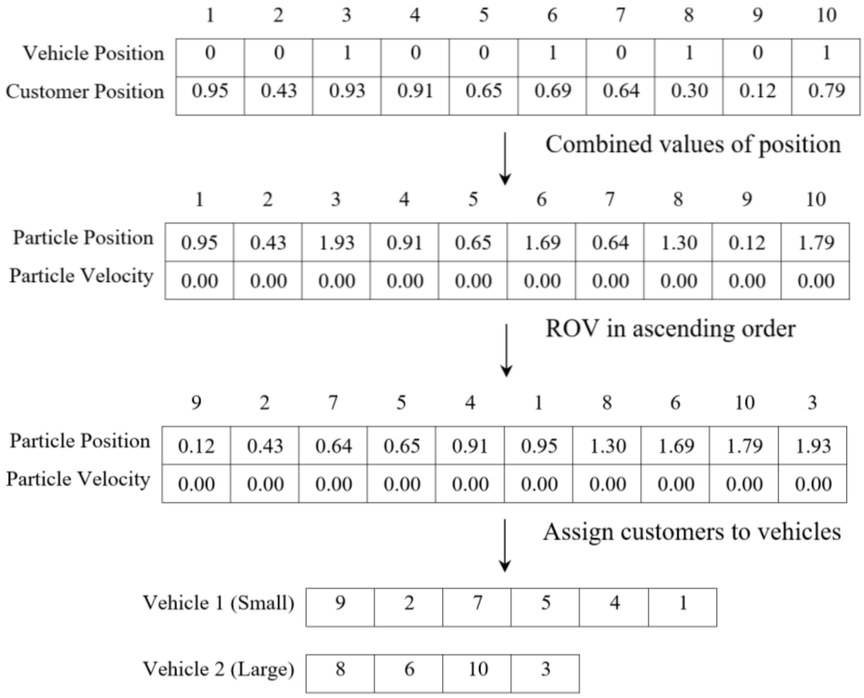

Figure 1 The particle representation

The pseudo code of the SAL-PSO is presented in Algorithm 2.

Algorithm 2: The self-adaptive learning particle swarm optimization (SAL-PSO)

```
input: problem instance data (I), particle numbers (N), particle bounds (B), and max iteration (iteration_max)
output: swarm (S), global best position (X_gbest), global best value (TC_gbest)
procedure
    I ← problem instance data
    P ∈ S
    for (P: 1 to N) do
        P ← particle_initialization(I)
    end if
    i ← 0
    X_pbest, X_gbest ← ∅
    while (i < iteration_max) do
        for (P: 1 to N) do
            if (fitness_evaluation(P) < fitness_evaluation(P_pbest)) then
                X_pbest ← X
                if (fitness_evaluation(P) < fitness_evaluation(P_gbest)) then
                    X_gbest ← X
                end if
            else
                particle_reinitialization(P)
            end if
            r ← uniform_random()
            if (r ≤ 0.5) do
                adaptive_inertia_weight_velocity_info(P)            Eq. (29-31)
                time_varying_acceleration_coefficients(P)           Eq. (32-33)
            else
                sigmoid_inertia_weight(P)                           Eq. (34)
            end if
            particle_velocity_update(P)                             Eq. (28)
            particle_position_update(P)                             Eq. (35)
        end for
        i ← i + 1
    end while
end procedure
```

3.2 Self-adaptive learning mechanisms

In this study, we introduce the PSO parameter adjustment with a combination of adaptive inertia weight and acceleration coefficients mechanisms. In addition, the wavelet mutation is applied to a local search method as well. These approaches enhance the ability to perform an effective search in diversification and intensification manners.

3.2.1 The velocity update methods

Each particle of the swarm moves within the search space with the velocity based on the individual and companion experience introduced by Shi and Eberhart [42] in order to improve the performance of the optimization.

$$v_i(t) = wv_i + c_p r_1 (x_{pbest} - x_i) + c_g r_2 (x_{gbest} - x_i) \tag{28}$$

where w is the inertia weight of the velocity, v_i is the velocity of particle i, c_p and c_g are the coefficient individual and social learning experiences respectively, r_1 and r_2 are random values, x_i is the current position of particle i, x_{pbest} is the best individual learning experience of the current particle, and x_{gbest} is the best social learning experience of the whole swarm.

In this study, the first velocity update mechanism adopts the work of Xu [35] using the adaptive inertia weight based on velocity information. The average velocity of the swarm is defined as follows.

$$\overline{v(t)} = \frac{1}{n_d n_s} \sum_{i=1}^{n_s} \sum_{i=1}^{n_d} |v_{ij}(t)| \tag{29}$$

where $v_{ij}(t)$ is an absolute value of the current velocity at a dimension j of a particle i at each iteration t, and the n_d and n_s are the total numbers of dimensions and particles respectively. The ideal velocity of the swarm is calculated as follows.

$$v_{ideal}(t) = v_s \left(\frac{1 + \cos\left(\pi\left(\frac{t}{T_{0.95}}\right)\right)}{2} \right) \tag{30}$$

where v_s represents an initial ideal velocity, which is an average of the difference between the maximum and minimum of particle positions at each iteration t with the 95% of the maximum iteration T.

The adaptive inertia weight based on the relationship between the average velocity and the ideal velocity is determined by the following equation.

$$w(t+1) = \begin{cases} \max\{\omega(t) - \Delta\omega, \omega_{min}\}; & if\ \overline{v(t)} \geq v_{ideal}(t) \\ \min\{\omega(t) + \Delta\omega, \omega_{max}\}; & if\ \overline{v(t)} < v_{ideal}(t) \end{cases} \tag{31}$$

where $\omega_{min} = 0.4$ and $\omega_{max} = 0.9$ are the minimum and maximum inertia weights, and $\Delta\omega = 0.1$ is the movement of the inertia weight.

In addition, the time-varying acceleration coefficients taken from Ratnaweera et al. [43] to lessen social learning ability and improve the cognitive learning ability throughout the entire iterations are as shown in the equations below.

$$c_p(t) = c_{ps} + (c_{pf} - c_{ps})\frac{t}{T} \tag{32}$$

$$c_g(t) = c_{gs} + (c_{gf} - c_{gs})\frac{t}{T} \tag{33}$$

where $c_{ps} = 2.5$ and $c_{pf} = 0.5$ are the minimum and maximum values of cognitive learning and $c_{gs} = 0.5$ and $c_{gf} = 2.5$ are the minimum and maximum values of social learning respectively.

Moreover, the second velocity update mechanism applies the sigmoid inertia weight from Tian and Shi [44] to balance the adjustment during the PSO iterations.

$$w(t+1) = \begin{cases} 0.9; & t \geq \alpha T \\ \frac{1}{1+e^{(10t-2T)/T}} + 0.4; & otherwise \end{cases} \tag{34}$$

where t and T are the current iteration and maximum iteration respectively, as well as $\alpha = 0.2$ is the proportion of the maximum inertia weight. Also, the $c_p, c_g = 2.0$ are used as defaults.

3.2.2 The position update method

The movement of each particle in the swarm is updated by the following equation.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{35}$$

4. Numerical experiments

In this section, we instantiated different characteristics of 24 test cases: eight small-scale problem instances (S01-S08), eight medium-scale problem instances (M01-M08), and eight large-scale problem instances (L01-L08). The factors attributed to the objective function are varied in terms of the number of mixed and simultaneous pickup and delivery operations, and the number of vehicles as shown in Table 2.

Each pickup and delivery demand from a customer was generated in a range of values between 0-1,000 kg. The mixed pickup and delivery demands were randomly distributed at 30-60% of all demands. Two types of vehicle, capacity 3,650 and 7,000 kg, in Table 1 were used in these experiments. All customers have normal time windows. The speed limits of all road types are randomly selected based on Thailand Highway standard 45, 60 and 80 km/hr. The proposed SAL-PSO algorithm was executed and compared with the optimal solutions, or lower bound solutions, of the G-VRPMSPDTW-RT model obtained by the MILP optimizer. For small-scale and medium-scale problems, the time limit was set to 1,440 minutes; for large-scale problems, the time limit was 2,880 minutes. The experiments are 24 sets with 25 particles and a maximum of 5,000 iterations. The experiments were implemented and performed by IBM ILOG CPLEX 12.10 and Python 3.7 on an HP Pavilion workstation with a processor Intel Core i7-10750H at 2.60 GHz with 16GB of RAM running on Windows 10.

The optimal solutions of the eight small-scale problem instances (S01-S08) displaying the total transportation costs and computational time are shown in Table 3; as the problems get more complex, the computational time burdens the resources. Since the 3PL accepts the computational time less than 24 hours, instances M01 to L08, which are medium-scale and large-scale respectively, are not accepted by the business, but we acquired the lower bounds from IBM ILOG CPLEX Optimization Studio. Thus, we propose the SAL-PSO algorithm for solving both small and large problems.

Table 2 Problem instances.

Group of data	Instance name	No. of customers	No. of deliveries	No. of pickups	No. of SPD	No. of MPD	No. of vehicles
Small	S1	10	8	8	6	4	2
	S2	10	8	8	6	4	3
	S3	10	8	7	5	5	3
	S4	12	9	9	4	6	2
	S5	12	9	9	4	6	3
	S6	12	10	8	6	6	3
	S7	15	9	8	5	7	3
	S8	15	12	12	7	6	3
Medium	M1	20	16	16	12	8	3
	M2	20	14	14	8	12	3
	M3	22	18	18	14	8	4
	M4	22	16	15	9	13	4
	M5	25	20	20	15	10	4
	M5	25	18	17	10	15	4
	M7	27	22	22	17	10	5
	M8	30	24	24	18	12	5
Large	L1	40	32	32	24	16	6
	L2	40	28	28	16	24	6
	L3	42	34	34	26	16	8
	L4	42	30	29	17	25	7
	L5	45	35	35	27	18	7
	L6	45	32	31	18	27	8
	L7	48	41	41	34	14	7
	L8	50	42	42	34	16	8

Table 3 The optimal transportation costs obtained by MILP

Instance	CPLEX 12.10	
	Transportation costs (Baht)	Computational time (Minutes)
S1	1,833.21	3.58
S2	1,744.12	17.53
S3	1,804.65	17.61
S4	2,075.06	72.47
S5	1,604.97	377.70
S6	1,935.24	409.70
S7	1,916.95	1,404.33
S8	1,874.29	1,441.06

The G-VRPMSPDTW-RT model, standard PSO and SAL-PSO algorithms were tested for their performance with 24 problem instances. The best solutions of these methods are presented in Table 4. Although the constructive heuristic provided ineffective solutions for small-scale instances S01 to S08, the standard PSO and SAL-PSO yielded the best solutions for the same instances. Moreover, both PSO and SAL-PSO gave good solutions for medium-scale and large-scale instances M01 to L08 compared with the lower bounds. The computational times of SAL-PSO were rather higher than those of PSO for small-scale instances, but they gradually decreased for most of the medium-scale and large-scale instances.

$$HP\%_{SAL-PSO} = \frac{Solution_{OPT}}{Solution_{SAL-PSO}} \times 100 \quad (36)$$

where $HP(\%)_{SAL-PSO}$ = Heuristic performance of SAL-PSO (%)
$Solution_{OPT}$ = Optimal solution obtained from MILP
$Solution_{SAL-PSO}$ = The best solution obtained from SAL-PSO

$$RI(\%)_{SAL-PSO} = \frac{(Solution_{PSO} - Solution_{SAL-PSO})}{Solution_{PSO}} \times 100 \quad (37)$$

where $RI(\%)_{SAL-PSO}$ = Relative improvement of SAL-PSO (%)
$Solution_{PSO}$ = The best solution obtained from the standard PSO
$Solution_{SAL-PSO}$ = The best solution obtained from SAL-PSO

Table 5 illustrates a comparison of best solution performance obtained from MILP, the standard PSO and SAL-PSO among small-scale, medium-scale, and large-scale instances. The percentage of heuristic performance and relative improvement were calculated by Eq. (36) and Eq. (37) respectively. The tests reveal that both standard PSO and SAL-PSO have an average percentage efficiency at 98.76% and 99.61%, respectively, for small-scale instance S01 to S08. The average percentage of relative improvement for this group was 0.86%. For medium-scale and large-scale instances, on the other hand, the SAL-PSO has an average percentage efficiency of 97.01% compared with 92.90 % for the standard PSO. Interestingly, for higher problem complexity, the average percentage of relative improvement for these groups was 4.45%. The computational time used by the SAL-PSO is slightly higher than the values of the standard PSO among all three groups. In Table 6, the statistical analysis is conducted to verify if the proposed method is significantly different from the mathematical model by using the paired t-tests at the 95% reliability level, and the p-value of SAL-PSO was greater than 0.05 which means it is not significantly different from the MILP. As can be seen from this table, the SAL-PSO can be used as a representative of the mathematical model to solve the problem.

Table 4 A comparison of the best solutions and average transportation costs obtained from each instance

Problem instance	CPLEX 12.10		Constructive heuristic		PSO		SAL-PSO	
	Best cost (Baht)	CPU time (Minutes)	Best cost (Baht)	CPU time (Minutes)	Best cost (Baht)	CPU time (Minutes)	Best cost (Baht)	CPU time (Minutes)
S01	1,833.21	3.58	2,487.18	0.0011	1,833.21	4.23	1,833.21	5.16
S02	1,744.12	17.53	2,199.97	0.0023	1,744.12	4.57	1,744.12	5.43
S03	1,804.65	17.61	2,257.00	0.0012	1,804.65	4.36	1,804.65	5.26
S04	2,075.06	72.47	3,279.46	0.0023	2,075.06	6.47	2,075.06	4.97
S05	1,604.97	377.70	2,236.04	0.0024	1,604.97	6.67	1,604.97	7.18
S06	1,935.24	409.70	2,235.24	0.0024	1,935.24	6.52	1,935.24	7.12
S07	1,916.95	1,404.33	2,395.46	0.0022	1,916.95	19.57	1,916.95	26.80
S08	1,874.29	1,441.06	3,202.94	0.0020	1,874.29	20.23	1,874.29	29.50
M01	3,456.81	1,440.00	5,321.34	0.0011	3,613.60	24.45	3,470.96	35.65
M02	3,385.45	1,440.00	5,218.46	0.0012	3,521.65	25.40	3,437.61	34.62
M03	3,737.21	1,440.00	5,741.06	0.0016	3,908.87	54.00	3,788.72	44.62
M04	3,862.19	1,440.00	5,682.79	0.0014	3,985.87	52.79	3,925.25	51.45
M05	3,597.79	1,440.00	5,775.57	0.0030	3,782.00	67.31	3,622.85	70.15
M06	3,672.84	1,440.00	5,564.17	0.0030	3,794.12	69.32	3,690.87	71.78
M07	3,986.40	1,440.00	6,083.06	0.0021	4,241.99	65.42	4,025.65	72.20
M08	4,240.16	1,440.00	7,026.64	0.0031	4,480.75	77.95	4,293.03	80.28
L01	7,597.05	2,880.00	12,344.36	0.0079	8,599.38	106.13	7,818.00	144.59
L02	7,625.46	2,880.00	11,157.96	0.0083	8,397.18	108.64	7,798.26	131.16
L03	7,690.06	2,880.00	15,692.61	0.0043	8,434.37	131.77	7,803.00	150.55
L04	7,794.52	2,880.00	15,384.42	0.0047	8,357.94	134.03	7,934.79	142.79
L05	8,706.79	2,880.00	13,126.55	0.0055	9,095.77	136.43	8,872.94	157.16
L06	8,549.61	2,880.00	13,749.85	0.0060	9,106.27	154.28	8,642.95	162.14
L07	9,712.17	2,880.00	14,966.70	0.0049	10,378.48	181.79	9,967.54	167.58
L08	9,846.59	2,880.00	17,361.56	0.0052	10,408.28	196.54	9,999.01	197.73

Table 5 The percentage of the proposed algorithm performance acquired from Table 4

Problem instance	Transportation costs (Baht)			Computational time (Minutes)			HP %		RI %
	MILP	PSO	SAL-PSO	MILP	PSO	SAL-PSO	PSO	SAL-PSO	
S01	1,833.21	1,833.21	1,833.21	3.58	4.23	5.16	99.28	99.43	0.15
S02	1,744.12	1,744.12	1,744.12	17.53	4.57	5.43	99.42	99.62	0.20
S03	1,804.65	1,804.65	1,804.65	17.61	4.36	5.26	99.53	99.68	0.16
S04	2,075.06	2,075.06	2,075.06	72.47	6.47	4.97	99.15	99.85	0.70
S05	1,604.97	1,604.97	1,604.97	377.70	6.67	7.18	98.86	99.63	0.78
S06	1,935.24	1,935.24	1,935.24	409.70	6.52	7.12	98.40	99.32	0.94
S07	1,916.95	1,916.95	1,916.95	1,404.33	19.57	26.80	98.06	99.62	1.59
S08	1,874.29	1,874.29	1,874.29	1,441.06	20.23	29.50	97.42	99.74	2.39
					(S)	Average	98.76	99.61	0.86
M01	3,456.81	3,613.60	3,470.96	1,440.00	24.45	35.65	95.44	99.13	3.87
M02	3,385.45	3,521.65	3,437.61	1,440.00	25.40	34.62	96.94	99.51	2.65
M03	3,737.21	3,908.87	3,788.72	1,440.00	54.00	44.62	92.47	97.81	5.77
M04	3,862.19	3,985.87	3,925.25	1,440.00	52.79	51.45	96.19	98.31	2.21
M05	3,597.79	3,782.00	3,622.85	1,440.00	67.31	70.15	94.48	99.26	5.05
M06	3,672.84	3,794.12	3,690.87	1,440.00	69.32	71.78	96.70	98.96	2.34
M07	3,986.40	4,241.99	4,025.65	1,440.00	65.42	72.20	93.32	97.62	4.61
M08	4,240.16	4,480.75	4,293.03	1,440.00	77.95	80.28	94.40	99.15	5.03
L01	7,597.05	8,599.38	7,818.00	2,880.00	106.13	144.59	87.71	88.99	5.16
L02	7,625.46	8,397.18	7,798.26	2,880.00	108.64	131.16	87.74	93.45	6.51
L03	7,690.06	8,434.37	7,803.00	2,880.00	131.77	150.55	89.95	92.39	2.71
L04	7,794.52	8,357.94	7,934.79	2,880.00	134.03	142.79	91.78	94.97	3.48
L05	8,706.79	9,095.77	8,872.94	2,880.00	136.43	157.16	94.16	98.14	4.23
L06	8,549.61	9,106.27	8,642.95	2,880.00	154.28	162.14	91.37	98.04	7.31
L07	9,712.17	10,378.48	9,967.54	2,880.00	181.79	167.58	91.49	96.86	5.88
L08	9,846.59	10,408.28	9,999.01	2,880.00	196.54	197.73	92.24	96.28	4.38
					(M&L)	Average	92.90	97.01	4.45
					(Total)	Average	94.85	97.88	3.25

Table 6 The statistical analysis of results obtained in Table 5

Method	Constructive heuristic	PSO	SAL-PSO
MILP	0.000	0.020	0.073
Constructive heuristic		0.000	0.000
PSO			0.020

5. Discussion

This article presents economic and environmental costs of a reverse logistics framework of mixed and simultaneous pickup and delivery with time windows and road types solved by mathematical programming, the standard particle swarm optimization (PSO), and the self-adaptive learning particle swarm optimization (SAL-PSO). The aim is to decrease the total transportation and driver costs for the 3PL business that provides mixed and simultaneous pickup and delivery operations. The CPLEX found the optimal solutions in the small-scale instances S01 to S08. Similarly, both standard PSO and SAL-PSO algorithms have the same ability to reach the optimal solution for small-scale instances. As expected, the CPLEX required a tremendous time to find the lower bounds for the medium-scale instances M01 to M08 and large-scale instances L01 to L08, because the complexity of mixed integer linear programming becomes NP-hard. Therefore, the computational time is unacceptable for large-scale problems. In contrast, the SAL-PSO brought forth better solutions for both medium-scale and large-scale instances with competitively low computational time compared with CPLEX. The PSO with self-adaptive learning provides a variety of solutions when the particles move with adaptive inertia weight by using velocity information from the average velocity and the ideal velocity as well as the sigmoid-like function throughout the search space within time varying iterations to find the best solutions. Even though the SAL-PSO requires relatively higher computational time compared with the standard PSO since it performs additional calculation of parameter adaptiveness in order to enhance its intensification and diversification mechanisms for every iteration, it still returns the best solutions in all 24 test cases. As can be seen in Table 6, the statistical analysis is conducted to verify if the SAL-PSO is significantly different from the standard PSO by using the paired t-tests at the 95% reliability level, and the p-value was less than 0.05 which means it is significantly different from the standard PSO. Moreover, the percentage of relative improvement between the solutions obtained from the original PSO and the SAL-PSO in medium-scale and large-scale problem instances range between 2.21% and 7.31%. In our view the results emphasize the validity of the proposed SAL-PSO algorithm. It shows a better performance over the standard PSO for medium-scale and large-scale instances.

6. Conclusions

This research has investigated the transportation activities of sustainable reverse logistics which is strongly challenging when we determine various factors attributed to the economic and environmental costs. We propose the mixed integer linear programming model and the SAL-PSO algorithm for a vehicle routing to serve customers with mixed and simultaneous demands of the 3PL company under the time and speed restrictions. In general, it suggests that the SAL-PSO outperforms the standard PSO by 3.25% of the average percentage of relative improvement. The present findings might have important implications for solving this problem.

However, the current study was limited by the constant speed levels used by vehicles. The average speed levels are derived from the minimum and maximum speed limit discretization. This relates to the approximation of the travel time interval that a vehicle travels from one customer to another. It can vary the fuel consumption calculated by the CMEM model from the details of the travel time. To further our research, we intend to reformulate the mathematical model with a continuous speed variable in future study. It will allow the travel time of vehicles to represent real-world problems. Moreover, we need to adopt the local search mechanism to improve the solution quality and the computational time of the proposed SAL-PSO.

7. Acknowledgements

This work is supported by the Research Unit on System Modeling for Industry (Grant No. SMI.KKU MS611008), Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand and by Graduate Studies, Khon Kaen University, Thailand in collaboration with the Decision Analysis Lab, National Tsing Hua University, Taiwan. The authors would also like to thank Mr. Ian Thomas for critical review of the manuscript.

8. References

[1] Komolsuradej N. Reverse logistics: the new opportunities for Thai logistics players in e-commerce trend [Internet]. 2018 [cited 2020 Oct 7]. Available from: https://www.scbeic.com/en/detail/file/product/4610/f1vip3u53j/Note_EN_Reverse_Logistics_20180404.pdf.

[2] Saowanee R, Pornsawan R, Yuwawan J. The effects of COVID-19 towards a new trend of transportation under the next normal [Internet]. Bangok, Thailand; 2020 [cited 2020 Oct 7]. Available from: https://www.bot.or.th/Thai/ResearchAndPublications/DocLib_/Article_14Apr2020.pdf.

[3] Council O of the NE and SD. Thailand’s logistics report 2018 [Internet]. 2019 [cited 2020 Oct 7]. Available from: https://www.nesdb.go.th/ewt_dl_link.php?nid=9359.

[4] Energy forecast and information technology center. Energy situation January-June 2020 [Internet]. Bangkok, Thailand; 2020. [cited 2020 Oct 7]. Available from: http://www.eppo.go.th/index.php/th/component/k2/item/download/20313_2995209d28edea089f8d992494177bb3.

[5] Faria MV, Duarte GO, Varella RA, Farias TL, Baptista PC. How do road grade, road type and driving aggressiveness impact vehicle fuel consumption? Assessing potential fuel savings in Lisbon, Portugal. Transp Res Transp Environ. 2019;72:148-61.

[6] Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transp Res A Gen. 1989;23(5): 377-86.

[7] Nagy G, Salhi S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. Eur J Oper Res. 2005;162(1):126-41.

[8] Lin C, Choy KL, Ho GTS, Ng TW. A genetic algorithm-based optimization model for supporting green transportation operations. Expert Syst Appl. 2014;41(7):3284-96.

[9] Osaba E, Yang XS, Diaz F, Onieva E, Masegosa AD, Perallos A. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. Soft Comput. 2017;21(18):5295-308.

[10] Lin C, Choy KL, Ho GTS, Chung SH, Lam HY. Survey of green vehicle routing problem: past and future trends. Expert Syst Appl. 2014;41(4):1118-38.

[11] Bloemhof-Ruwaard JM, Van Beek P, Hordijk L, Van Wassenhove LN. Interactions between operational research and environmental management. Eur J Oper Res. 1995;85(2):229-43.

[12] Sbihi A, Eglese RW. Combinatorial optimization and green logistics. 4OR-Q J Oper Res. 2007;5(2):99-116.

[13] Kuo Y. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Comput Ind Eng. 2010;59(1):157-65.

[14] Bektaş T, Laporte G. The pollution-routing problem. Transp Res B Meth. 2011;45(8):1232-50.

[15] Erdogan S, Miller-Hooks E. A green vehicle routing problem. Transp Res E Logist Transp Rev. 2012;48(1):100-14.

[16] Xiao Y, Zhao Q, Kaku I, Xu Y. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Comput Oper Res. 2012;39(7):1419-31.

[17] Franceschetti A, Honhon D, Van Woensel T, Bektaş T, Laporte G. The time-dependent pollution-routing problem. Transp Res B Meth. 2013;56:265-93.

[18] Demir E, Van Woensel T. Mathematical modeling of CO_2e emissions in one-to-one pickup and delivery problems. IEEE International conference on industrial engineering and engineering management; 2013 Dec 10-13; Bangkok, Thailand. New York: IEEE; 2013. p. 63-7.

[19] Xiao Y, Konak A. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. Transp Res E Logist Transp Rev. 2016;88:146-66.

[20] Xiao Y, Konak A. A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem. J Clean Prod. 2017;167:1450-63.
[21] Poonthalir G, Nadarajan R. A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP). Expert Syst Appl. 2018;100:131-44.
[22] Soysal M, Çimen M, Demir E. On the mathematical modeling of green one-to-one pickup and delivery problem with road segmentation. J Clean Prod. 2018;174:1664-78.
[23] Tuntitippawan N, Asawarungsaengkul K. An artificial bee colony algorithm with local search for vehicle routing problem with backhauls and time windows. KKU Eng J. 2016;43(S3):404-8.
[24] Sethanan K, Jamrus T. Hybrid differential evolution algorithm and genetic operator for multi-trip vehicle routing problem with backhauls and heterogeneous fleet in the beverage logistics industry. Comput Ind Eng. 2020;146:106571.
[25] Majidi S, Hosseini-Motlagh SM, Ignatius J. Adaptive large neighborhood search heuristic for pollution-routing problem with simultaneous pickup and delivery. Soft Comput. 2018;22(9):2851-65.
[26] Kachitvichyanukul V. Comparison of three evolutionary algorithms: GA, PSO, and DE. Ind Eng Manag Syst. 2012;11(3): 215-23.
[27] Kar AK. Bio inspired computing-a review of algorithms and scope of applications. Expert Syst Appl. 2016;59:20-32.
[28] Ai TJ, Kachitvichyanukul V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. Comput Oper Res. 2009;36(5):1693-702.
[29] Goksal FP, Karaoglan I, Altiparmak F. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Comput Ind Eng. 2013;65(1):39-53.
[30] Kachitvichyanukul V, Sombuntham P, Kunnapapdeelert S. Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via PSO. Comput Ind Eng. 2015;89:125-36.
[31] Norouzi N, Sadegh-Amalnick M, Tavakkoli-Moghaddam R. Modified particle swarm optimization in a time-dependent vehicle routing problem: minimizing fuel consumption. Optim Lett. 2017;11:121-34.
[32] Li Y, Lim MK, Tseng ML. A green vehicle routing model based on modified particle swarm optimization for cold chain logistics. Ind Manag Data Syst. 2019;119(3):473-94.
[33] Zhan Z, Zhang J. Adaptive particle swarm optimization. International conference on ant colony optimization and swarm intelligence; 2008 Sep 22-24; Brussels, Belgium. Berlin: Springer; 2008. p. 227-34.
[34] Wang Y, Li B, Weise T, Wang J, Yuan B, Tian Q. Self-adaptive learning based particle swarm optimization. Inform Sci. 2011;181(20):4515-38.
[35] Xu G. An adaptive parameter tuning of particle swarm optimization algorithm. Appl Math Comput. 2013;219(9):4560-9.
[36] Pornsing C, Sodhi MS, Lamond BF. Novel self-adaptive particle swarm optimization methods. Soft Comput. 2016;20(9): 3579-93.
[37] Marinakis Y, Marinaki M, Migdalas A. A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. Inform Sci. 2019;481:311-29.
[38] Demir E, Bektaş T, Laporte G. An adaptive large neighborhood search heuristic for the pollution-routing problem. Eur J Oper Res. 2012;223(2):346-59.
[39] Barth M, Younglove T, Scora G. Development of a heavy-duty diesel modal emissions and fuel consumption model. California PATH. 2005:1-124.
[40] Barth M, Boriboonsomsin K. Energy and emissions impacts of a freeway-based dynamic eco-driving system. Transp Res D Transp Environ. 2009;14(6):400-10.
[41] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of ICNN'95 - international conference on neural networks; 1995 Nov 27-Dec 1; Perth, Australia. New York: IEEE; 1995. p. 1942-8.
[42] Shi Y, Eberhart RC. Parameter selection in particle swarm optimization. International conference on evolutionary programming; 1998 Mar 25-27; San Diego, USA. Berlin: Springer; 1998. p. 591-600.
[43] Ratnaweera A, Halgamuge SK, Watson HC. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evol Comput. 2004;8(3):240-55.
[44] Tian D, Shi Z. MPSO: modified particle swarm optimization and its applications. Swarm Evol Comput. 2018;41:49-68.