Engineering Journal issn: 0125-8281

Article

Hybrid Genetic Algorithm for Multi-Period Vehicle Routing Problem with Mixed Pickup and Delivery with Time Window, Heterogeneous Fleet, Duration Time and Rest Area

Kittitach Kamsopa[1,a], Kanchana Sethanan[1,b], Thitipong Jamrus[1,c,*], and Liliana Czwajda[2,d]

1 Research Unit on System Modeling for Industry, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen, 40002, Thailand
2 Centre for Industrial Applications of Mathematics and Systems Engineering, Polish Academy of Sciences, ul. Śniadeckich 8, Warsaw, Poland
E-mail: [a]kittitachkamsopa@gmail.com, [b]skanch@kku.ac.th, [c,*]thitja@kku.ac.th (Corresponding author), [d]liliana.czwajda@gmail.com

Abstract. Most logistics industries are improving their technology and innovation in competitive markets in order to serve the various needs of customers more efficiently. However, logistics management costs are one of the factors that entrepreneurs inevitably need to reduce, so that goods and services are distributed to a number of customers in different locations effectively and efficiently. In this research, we consider the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, duration time and rest area (MVRPMPDDR). In the special case that occurs in this research, it is the rest area for resting the vehicle after working long hours of the day during transportation over multiple periods, for which with confidence no research has studied previously. We present a mixed integer linear programming model to give an optimal solution, and a meta-heuristic approach using a hybrid genetic algorithm with variable neighborhood search algorithm (GAVNS) has been developed to solve large-sized problems. The objective is to maximize profits obtained from revenue after deducting fuel cost, the cost of using a vehicle, driver wage cost, penalty cost and overtime cost. We prepared two algorithms, including a genetic algorithm (GA) and variable neighborhood search algorithm (VNS), to compare the performance of our proposed algorithm. The VNS is specially applied instead of the mutation operator in GA, because it can reduce duplicate solutions of the algorithms that increase the difficulty and are time-consuming. The numerical results show the hybrid genetic algorithm with variable neighborhood search algorithm outperforms all other proposed algorithms. This demonstrates that the proposed meta-heuristic is efficient, with reasonable computational time, and is useful not only for increasing profits, but also for efficient management of the outbound transportation logistics system.

Keywords: Mixed pickup and delivery with time windows, multi-period, rest area, hybrid genetic algorithm, variable neighborhood search algorithm, maximize profits.

ENGINEERING JOURNAL Volume 25 Issue 10
Received 5 May 2021
Accepted 10 October 2021
Published 31 October 2021
Online at https://engj.org/
DOI:10.4186/ej.2021.25.10.71

This article is based on the presentation at The 7th KKU International Engineering Conference 2021 (KKU-IENC 2021) at Khon Kaen University, Khon Kaen, Thailand, 12th-14th May 2021.

DOI:10.4186/ej.2021.25.10.71

1. Introduction

Nowadays, most industries are rapidly developing their technology and innovating so that they can achieve industry 4.0. Hence, many companies are focusing on reforming the inbound and outbound systems for industries, including the production processes and services that lead to new products and new service structures, using advanced knowledge and high technology. Moreover, they are trying to enhance the value involving both products and services through the development of the logistics system. The logistics system is important to many industries in terms of continuous product flow, from inbound into outbound systems, which consist of transportation and distribution of materials and mass products by different modes: land, rail, water and air. Due to the variety of modern technologies, especially e-commerce businesses, effective and fast transport and distribution of products on time to consumers is the key to business competitiveness nowadays. Also, with a variety of products and choices to access the ordering process so that consumers can access quickly, such as online shopping, online tracking, etc., there is a challenge that entrepreneurs face. Entrepreneurs who are ready for logistics will have an advantage over their competitors in the same business. The inevitable solution for entrepreneurs is the management of transport services to distribute products to consumers in many areas accurately, on time and effectively. One choice for transport management is to hire a third-party logistics provider (3PL), who is responsible for supply chain management to handle the transportation system efficiency. In the current situation, most entrepreneurs are tending to hire third-party logistics providers to handle increased transportation and reduce transportation costs.

For the framework of this research, we considered problems encountered in vehicle routing with mixed pickup and delivery (VRPPD). The characteristic of the VRPPD problem is to determine the routing of product transfer, with pickups from one node for delivery at the second node. The objective is to find the optimal routes for maximizing profits, using a heterogeneous fleet from a central depot to serve multi-consumers to one (1-many-1) with known constant demands. To increase such profits, in real practice, the vehicle may be allowed to have flexible mixed pickup and delivery, while customers have different time windows for each period. Each vehicle can accept pickups at the depot, according to product demand by customers, and the vehicle can service nodes for pickups or deliveries at any level of product in the vehicle, so a vehicle does not necessarily only pickup or deliver. When the vehicle returns to the depot, it must finish all shipping, leaving an empty vehicle, because this is efficient transportation management and is a policy. Moreover, the vehicle may be allowed to travel for several periods to support many customers based on the maximum duration of the existing route. Therefore, we must have a rest area for the vehicle to rest after working long hours during the transportation period. On the other hand, if transportation cannot meet the requirements of the customer within his time window, the customers may allow the vehicle to visit and service a customer outside his time window. In this case, the transport company will be responsible for penalties for late arrival, as a vehicle routing problem with semi-soft time windows constraint (VRPSSTW) and overtime for the drivers in case of exceeding normal working hours, which is a reason to consider both penalty costs and overtime costs. Therefore, this research proposes the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, duration time and rest area (MVRPMPDDR) as shown in Fig. 1.

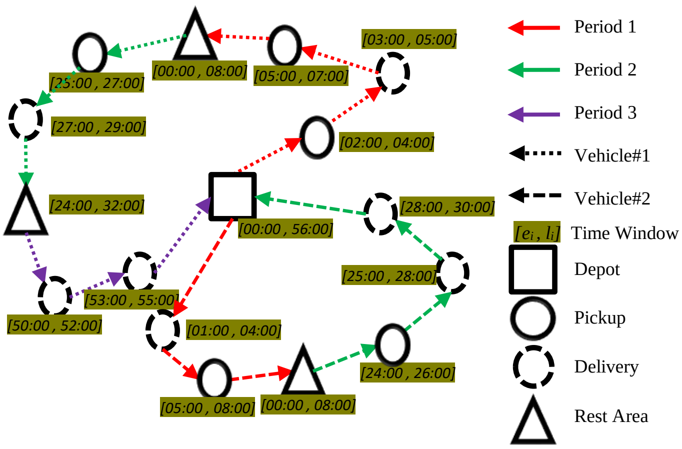

Fig. 1. Illustration of the MVRPMPDDR problem.

The MVRPMPDDR is a well-known NP-hard optimization problem and is highly complex for large-sized problems, for which mathematical models are not available. Thus, we have generated the hybrid meta-heuristic approach, using a hybrid genetic algorithm with variable neighborhood search algorithm (GAVNS) for solving MVRPMPDDR. The nature of a genetic algorithm (GA) is to search for answers based on the genetic selection mechanism from nature. The advantages of GA are its rapid computation, and that the process is not complicated. Therefore, integer encoding is applied to this problem because it is a simple process for the genetic algorithm, but its weakness is local search is not strong enough to find the optimal result. Consequently, a variable neighborhood search algorithm (VNS) was applied for a hybrid with the genetic algorithm. The VNS is a local search method to search the solution space systematically, while the conceptual characteristics of the neighborhood change to descend to local minima and escape from the valleys which contain them.

Vehicle routing management has been attracting attention over the past years, as can be observed in the academic literature. It has expanded in many ways to deal with vehicle routing problems (VRP) in various industries, including third-party logistics providers (3PL) for real-life transportation situations, such as capacitated vehicle routing problems [1-3] and inventory open vehicle routing problems [4]. In particular, the vehicle routing problem with pickup and delivery (VRPPD) is a well-known problem in the fields of operations research, logistics and transportation optimization. this problem is necessary for managing pickup and delivery activities within the capacity restriction of one vehicle's route, such as in the research study of Avci & Topaloglu [5], which proposed the heterogeneous vehicle routing problem with simultaneous pickup and delivery (VRPSPD). For its solution, a hybrid local search algorithm was developed in which a non-monotone threshold adjusting strategy is integrated with tabu search. The objective minimizes the total costs. Dechampai et al. [6] developed a differential evolution (DE) that used self-organizing maps to cluster customers before finding the optimal solution of the capacitated vehicle routing problem with flexibility of mixing pickup and delivery services, and the maximum duration of a route, to determine routing in the poultry industry. The objective is to minimize the total cost. Ting et al. [7] presented three metaheuristic algorithms, tabu search (TS), genetic algorithm (GA) and scatter search (SS), for solving the multi-vehicle selective pickup and delivery problem (MVSPDP), in which the objective minimizes the total cost. There is also a vehicle routing problem with delivery and pickup with time window (VRPPDTW), which adds to the consideration of different visit times for customers. For example, Bhusiri et al. [8] introduces the vehicle routing problem with soft time windows (VRPSTW) by using the mathematical method and the Branch-and-Price method in a set partitioning master problem and its new subproblem. The objective function minimizes total costs. The proposed solutions have further been compared to the state-of-the-art literature of VRPSTW, VRPHTW and VRPSSTW, signifying the superiority of this issue. Wang et al. [9] proposed a general mixed-integer programming model and parallel simulated annealing (p-SA) by using the residual capacity and radial surcharge (RCRS) insertion heuristic for solving VRP with simultaneous delivery and pickup with time window (VRPSPDTW). The objective minimizes the routing cost. Ahkamiraad & Wang [10] proposed a mixed-integer linear programming model and a hybrid of the genetic algorithm and particle swarm optimization (HGP) for solving multiple cross-docked VRP with pickup, delivery and time windows. The purpose is to minimize transportation and fixed costs.

In terms of a multi-period vehicle routing problem with delivery and pickup (MPVRPPD), it is considered to be a customer service that has different time windows in multiple periods. The MPVRPPD is not only a combinatorial problem, but it is also more complex if it considers both delivery and pickup, and a time window (MPVRPPDTW). There is research about multi-period VRPPDTW. For example, Mancini [11] studied the multi-depot multi-period vehicle routing problem with a heterogeneous fleet (MDMPVRPHF). The goal of the problem is to minimize the total delivery cost. The vehicles have a limit on the maximum route duration. A mixed integer programming (MIP) formulation was presented and an Adaptive large neighborhood search (ALNS) based metaheuristic approach was applied. Chami et al. [12] conducted a study on the multi-period pickup and delivery problem with time windows and paired demands (Mu-PDPTWPD). The aim was to minimize the total traveled distance and the penalty cost simultaneously. The proposed solution is to create a mathematical model and develop a greedy randomized adaptive search procedure with a hybrid genetic algorithm (GRASP-HGA), which results in efficiency of the calculation time compared to the Standard case (Benchmark Instances). Furthermore, Larrain et al. [13] presented the multi-period vehicle routing problem with due dates. The objective is to minimize the distribution costs and the costs related to delayed deliveries. For this method, the new branch-and-bound algorithm was applied and a variable MIP neighborhood descent (VMND) algorithm developed to increase performance for problem resolution. In the special case that occurs in this research, it is the rest area for resting the vehicle after working long hours of the day during transportation over multiple periods, for which with confidence no research has studied previously.

The MVRPMPDDR problem is an NP-hard problem (Non-deterministic polynomial-time); it is extremely difficult to find the optimal solution. Therefore, the heuristic and/or meta-heuristic are available choices to guide searches more efficiently. Recently, a genetic algorithm has still been popular, which is a population-based search technique. An important characteristic of the genetic algorithm is finding methods that combine directed and stochastic search that can produce a balance between the exploration and exploitation of the search space, its straightforward nature, and quick convergence in search of the optimal solution.

The Genetic algorithm (GA) is one of the influential techniques that has been applied in various research fields such as vehicle routing problems [14-16], scheduling problems [17-18], Location–allocation problem [19] and supply chain [20]. Although the GA has been used in a variety of research areas, it has limitations when applied for solving a problem, as the solution process may not be complex enough to find the optimal solution. So, in order to improve the diversity of GA, the hybridization of the GA with other methods has been used to enhance the performance of the traditional GA. For example, Nguyen et al. [21] developed a hybrid generational genetic algorithm with a local search (the unified tabu search (UTS) and the random variable neighborhood Search (RVNS)) and improved the two crossover operators to seek diversification of the exploration in the solution space. For solving the periodic vehicle routing problem with time windows, the objective is to minimize total costs. Vidal et al. [22] proposed two alternative hybrid metaheuristic algorithms. The first algorithm is based on an iterated local search algorithm and the second algorithm is a hybrid genetic search (The Unified Hybrid Genetic Search). This problem is the clustered vehicle routing problem and the objective is to minimize total traveling costs. Xia et al. [23] introduced a hybrid genetic algorithm with variable neighborhood search (GAVNS) to solve dynamic integrated process planning and scheduling

DOI:10.4186/ej.2021.25.10.71

Table 1. A comparison of the literature on restrictions and approaches.

Researchers	Restrictions						Approaches			
	Visit type	Capacity	Multi-period	Time windows	Duration	Rest area for vehicle	Exact method	Heuristic	Meta-heuristic	Local search
Joshi and Kaur (2015)	D	√						√		
Cinar et al. (2016)	D	√			√			√		
Amous et al. (2017)	D	√						√		√
Supithak (2018)	D	√							√	
Avci and Topaloglu (2016)	PD	√							√*	√*
Dechampai et al. (2017)	P/D	√			√			√	√	
Ting et al. (2017)	P/D	√		√					√	
Bhusiri et al. (2014)	D	√		√			√			
Wang et al. (2015)	PD	√		√			√		√	
Ahkamiraad and Wang (2018)	Cross-docked	√		√			√		√*	
Mancini (2016)	D	√	√		√		√			√
Al Chami et al. (2018)	P/D	√	√	√	√		√		√*	√*
Larrain et al. (2019)	D	√	√		√		√	√		
Wang and Chen (2012)	PD	√		√			√		√	
Karakatic and Podgorelec (2015)	D	√							√	
Mohammed et al. (2017)	D	√							√	
Nguyen et al. (2014)	D	√		√					√*	√*
Vidal et al. (2015)	D	√							√*	√*
Arakaki et al. (2018)	D	√							√*	√*
Baniamerian et al. (2019)	Cross-docked	√					√		√*	√*
Our study	P/D	√	√	√	√	√	√		√*	√*

(DPPS). The objective is to maximize the makespan and mean flow time. The results show that the proposed method has achieved significant improvement in solving the DIPPS. Arakaki et al. [24] introduced a hybrid genetic algorithm with feasibility and local search procedures for solving the open capacitated arc routing problem (OCARP). The objective is to find the lowest cost set of routes that services all edges with positive demand (required edges). Baniamerian et al. [25] proposed a mixed-integer linear programming (MILP) model and developed a new hybrid metaheuristic algorithm based on the modified variable neighborhood search (MVNS) with four shaking and two neighborhood structures and a genetic algorithm (GA) to solve a profitable heterogeneous vehicle routing problem with cross-docking (PHVRPCD). The objective is to maximize the total profit of the system including the terms of the total revenue, purchasing and traveling costs. Furthermore, in Hnin et al. [26] the genetic algorithm (GA) is applied to adapt the hyperparameter of support vector regression, including particle swarm optimization (PSO) and Bayesian optimization (BO), in short-term load forecasting (STLF) in Thailand. The purpose of this paper is to improve forecasting accuracy by optimizing the hyperparameters of SVR.

Therefore, this research proposes a mathematical model for solving small-sized problems and develops the meta-heuristic approach using a hybrid genetic algorithm with variable neighborhood search algorithm (GAVNS) for solving large-sized problems of the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, duration time and rest area (MVRPMPDDR). The objective is to maximize total profit. In addition, the model of this research can be an alternative for various logistic industries. There are published reports which explain various solutions as summarized in Table 1.

2. Model Formulation

2.1. Assumptions and Constraints

The proposed formulation is based on the following assumptions:

(1) The logistic depot and rest area are known, and the customer points are also known.

(2) Each customer can receive pickup or delivery only and every customer receives service only once by one vehicle, but one vehicle can serve multiple customers on only a single path and there are no split shipments.

(3) Customers have different time windows for each period. Therefore, the vehicle should provide service within his time window, otherwise there will be penalties for late arrival.

(4) The demands for products are known and the products at the depot are unlimited.

(5) The product types are defined as a single commodity that is calculated from the weight of the products.

(6) The customer requirement must make sure that the number of products in the depot (Buffer stock) combined with pickup demands must be equal to delivery demands. The delivery demands must always be more than or equal to the pickup demands.

(7) Each vehicle takes the depot as its starting point, travels along the distribution route to the customer designated locations, then returns to the rest area or the depot within the time duration. Otherwise, there will be overtime costs.

(8) The vehicles can service multiple periods and are limited to the maximum duration time constraint of a route.

(9) The capacity of the vehicle is heterogeneous and the cumulative demands of a customer must not exceed the maximum capacity of the transport vehicle.

(10) The numbers of vehicles are unlimited and the speed of a vehicle is determined by a constant.

(11) Each vehicle can accept pickup demands at the depot, according to the customer requirements.

(12) The vehicle can service pickups or deliveries at any level of product in the vehicle, so a vehicle does not necessarily only pickup or deliver.

(13) Before the vehicle returns to the depot, it must finish all shipping with an empty vehicle.

(14) Rest areas do not allow activities for pickup or delivery to customers.

2.2. Mathematical Model

Indices

i, j	indices of depots, customers and rest areas; $i, j = 1,2,\dots,N,N+1,\dots,M$
k	index of vehicles; $k = 1,2,\dots,G$
t	index of the period for vehicles; $t = 1,2,\dots,MDT$

Input parameters

N	Maximum number of customers
M	Maximum number of rest areas
G	Maximum number of vehicles
MDT	Maximum duration time of a route (periods)
V	Number of depots, customers and rest areas
pp	Product price for customers (cost/kg)
f_k	Fuel cost of vehicle k (cost/km)
ov_k	Cost of using a vehicle k (cost/unit)
dw	Driver wage cost (cost/period)
pc	Penalty cost for late arrival after the time window by the customer (cost/hour)
ot_rate_t	Overtime late cost (cost/hour)
$dis_{i,j}$	Distance from customer i to customer j (km)
$tra_{i,j}$	Traveling time from customer i to customer j (hours)
s_i	Service time of customer i (hours)
p_i	Demand of customer for pickup i (unit)
d_i	Demand of customers for delivery i (unit)
$Cmax_k$	Maximum capacity of vehicle k (kg)
Tv_t	Maximum duration time that a vehicle that can give service in each period t (hours)
$E_{i,t}$	Open times of customers i (hours) during period t
$l_{i,t}$	Closed times of customers i (hours) during period t
M	Very big number
$EC_{i,t}$	Delay time edge of closed time for customers i (hours) that they allowed to occur during period t
$l'_{i,t}$	Delay time edge of closed time compared between the time of the customer i allowed and the vehicle can be serviced during period t

Set

Vw	Set of depots; $Vw = \{v_1\}$
Vc	Set of all customers; $Vc = \{v_2, v_3, v_4, \dots, v_N\}$
Vr	Set of rest areas; $Vr = \{v_{N+1}, v_{N+2}, v_{N+3}, \dots, v_M\}$
Vwc	Set of depot and all customers; $Vwc = Vw \cup Vc = \{v_1, v_2, v_3, \dots, v_N\}$
Vwr	Set of depot and rest areas; $Vwr = Vw \cup Vr = \{v_1, v_{N+1}, v_{N+2}, \dots, v_M\}$
Vcr	Set of all customers and rest areas; $Vcr = Vc \cup Vr = \{v_2, v_3, v_4, \dots, v_M\}$
$Vwcr$	Set of depot, all customers and rest areas; $Vwcr = Vw \cup Vc \cup Vr = \{v_1, v_2, v_3, \dots, v_M\}$
K	Set of available vehicles; $K = \{k_1, k_2, k_3, \dots, k_G\}$
T	Set of periods that vehicle can be serviced; $T = \{t_1, t_2, t_3, \dots, t_{MDT}\}$

DOI:10.4186/ej.2021.25.10.71

Decision variables

$x_{k,i,j,t}$	A binary variable that takes the value 1 if the route of vehicle k is between customer *i* and *j* during period t; 0 otherwise
y_k	A binary variable that takes the value 1 if the vehicle *k* is active; 0 otherwise
$z_{k,t}$	A binary variable that takes the value 1 if the vehicle *k* travels during period *t*; 0 otherwise
$b_{k,i,t}$	A real variable representing the starting service time of vehicle *k* at the customer *i* during period *t* (hours)
$b\_rest_{k,i,t}$	A real variable representing the arrival time of vehicle k at the rest area *i* during period *t* (hours)
$b\_de_{k,1,t}$	A real variable representing the arrival time of vehicle *k* at the depot during period *t* (hours)
$il_{k,1,1}$	An integer variable representing the quantity of products that are transferred to the vehicle *k* at the depot on the first period (unit)
$vl_{k,i,t}$	An integer variable representing the quantity of the products on vehicle *k* when service customer *i* during period t is completed (unit)
$rl\_s_{k,i,t+1}$	An integer variable representing the quantity of the products on vehicle k at the rest area i when starting to service the customer in the next period (unit)
$rl\_e_{k,i,t}$	An integer variable representing the quantity of products on vehicle *k* at the rest area *i* at the end of the service for that period (unit)
$w_{i,t}$	A real variable representing the late time when the vehicle to service the customer *i* is delayed, during period *t* (hours)
$ww_{k,i,t}$	A real variable representing the overtime for the vehicle *k* when the vehicle returned to depot or rest area *i* during period *t* (hours)
$u_{k,i,t}$	Decision variables for sub-tour

Objective function

$$Maximize\ profit$$

$$= \sum_{i \in Vwcr} \sum_{j \in Vwcr} \sum_{k} \sum_{t} \left(\left((p_i + d_i) pp \right) - \left(f_k dis_{i,j} \right) \right) x_{k,i,j,t} - \sum_{k} y_k ov_k - \sum_{k} \sum_{t} z_{k,t} dw - \sum_{i \in Vc} \sum_{t} pc\, w_{i,t} - \sum_{k} \sum_{i \in Vwr} \sum_{t} ot\_rate_t\, ww_{k,i,t} \quad (1)$$

Constraints

$$\sum_{k} \sum_{i \in Vwcr} \sum_{t} x_{k,i,j,t} = 1 \qquad \forall\, j \in Vc \quad (2)$$

$$\sum_{j \in Vc} \sum_{t} x_{k,1,j,t} \le 1 \qquad \forall\, k \quad (3)$$

$$\sum_{i \in Vcr} \sum_{t} x_{k,i,1,t} \le 1 \qquad \forall\, k \quad (4)$$

$$\sum_{i \in Vwcr} x_{k,i,h,t} - \sum_{j \in Vwcr} x_{k,h,j,t} = 0 \quad \forall\, h \in Vc, k, t \quad (5)$$

$$\sum_{i \in Vwcr} x_{k,i,h,t} - \sum_{j \in Vwcr} x_{k,h,j,t+1} = 0 \quad \forall\, h \in Vr, k, t < T \quad (6)$$

$$\sum_{i \in Vwcr} \sum_{k} x_{k,i,j,t} \ge 0 \qquad \forall\, j \in Vr, t < T \quad (7)$$

$$\sum_{i \in Vr} \sum_{j \in Vwc} x_{k,i,j,1} = 0 \qquad \forall\, k \quad (8)$$

$$\sum_{i \in Vr} \sum_{j \in Vr} \sum_{t} x_{k,i,j,t} = 0 \qquad \forall\, k \quad (9)$$

$$\sum_{i \in Vwcr} \sum_{j \in Vr} x_{k,i,j,T} = 0 \qquad \forall\, k \quad (10)$$

$$z_{k,t} \ge x_{k,i,j,t} \qquad \forall\, i \in Vwcr, j \in Vwcr, k, t \quad (11)$$

$$y_k \ge z_{k,t} \qquad \forall\, k, t \quad (12)$$

$$b_{k,i,t} + s_i + tra_{i,j} - M\left(1 - x_{k,i,j,t}\right) \le b_{k,j,t} \quad \forall\, i \in Vwc, j \in Vc, k, t \quad (13)$$

$$b_{k,i,t} + s_i + tra_{i,j} - M\left(1 - x_{k,i,j,t}\right) \le b_{rest_{k,j,t}} \quad \forall\, i \in Vwc, j \in Vr, k, t < T \quad (14)$$

$$e_{i,t} + s_i + tra_{i,j} - M\left(1 - x_{k,i,j,t}\right) \le b_{k,j,t} \quad \forall\, i \in Vr, j \in Vwc, k, t > 1 \quad (15)$$

$$b_{k,i,t} + s_i + tra_{i,1} - M\left(1 - x_{k,i,1,t}\right) \le b_{de_{k,1,t}} \quad \forall\, i \in Vcr, k, t \quad (16)$$

$$e_{i,t} \le b_{k,i,t} \le l'_{i,t} \qquad \forall\, i \in Vcr, k, t \quad (17)$$

$$l'_{i,t} \le min \left\{ \left(\left[\begin{matrix} (NEGO_{i,t}), \\ min \\ \left(l_{h,t} - s_i - tra_{i,h}\right) \forall\, h \in Vwr, \\ \left(l_{i,t} + s_i + \left(\frac{f_k (dis_{h,i} + dis_{i,h})}{pc}\right) \forall\, h \in Vwcr \right. \end{matrix} \right] \right) \right\} \quad \forall\, i \in Vc, t \in T \quad (18)$$

$$b_{k,i,t} - l_{i,d} = w_{i,t} \qquad \forall\, i \in Vc, k, t \quad (19)$$

$$b\_rest_{k,i,t} - Tv_t = ww_{k,i,t} \quad \forall\, i \in Vr, k, t < T \quad (20)$$

$$b\_de_{k,1,t} - Tv_t = ww_{k,1,t} \quad \forall\, i \in Vw, k, t \quad (21)$$

$$il_{k,1,1} \le Cmax_k \qquad \forall\, k \quad (22)$$

$$vl_{k,j,t} \leq Cmax_k + M\left(1 - \sum_{i \in Vwcr} x_{k,i,j,t}\right) \quad \forall j \in Vcr, k, t < T \tag{23}$$

$$il_{k,1,1} + \sum_{i \in Vwcr} \sum_{j \in Vcr} \sum_{t} p_j x_{k,i,j,t} = \sum_{i \in Vwcr} \sum_{j \in Vcr} \sum_{t} d_j x_{k,i,j,t} \quad \forall k \tag{24}$$

$$vl_{k,j,t} \geq il_{k,1,1} - d_j + p_j - M(1 - x_{k,1,j,t}) \quad \forall j \in Vc, k, t \tag{25}$$

$$vl_{k,j,t} \geq vl_{k,i,t} - d_j + p_j - M(1 - x_{k,i,j,t}) \quad \forall i \in Vc, j \in Vc, k, t \tag{26}$$

$$rl_{e_{k,j,t}} \geq vl_{k,i,t} - d_j + p_j - M(1 - x_{k,i,j,t}) \quad \forall i \in Vc, j \in Vr, k, t < T \tag{27}$$

$$vl_{k,j,t} \geq rl_{s_{k,i,t}} - d_j + p_j - M(1 - x_{k,i,j,t}) \quad \forall i \in Vr, j \in Vwc, k, t > 1 \tag{28}$$

$$vl_{k,j,t} - vl_{k,i,t} + d_j - p_j + M(1 - x_{k,i,1,t}) \geq 0 \quad \forall i \in Vc, j \in Vwcr, k, t \tag{29}$$

$$vl_{k,j,t} - rl_{s_{k,i,t}} + d_j - p_j + M(1 - x_{k,i,1,t}) \geq 0 \quad \forall i \in Vr, j \in Vwc, k, t > 1 \tag{30}$$

$$rl_{s_{k,i,t+1}} - rl_{e_{k,i,t}} = 0 \quad \forall j \in Vr, k, t < T \tag{31}$$

$$u_{k,i,t} - u_{k,j,t} + (N-1)(x_{k,i,j,t}) \leq N - 2 \quad \forall i \in Vc, j \in Vc, k, t \tag{32}$$

$$b_{k,i,t}, b_{rest_{k,i,t}}, b_{de_{k,1,t}}, il_{k,1,1}, vl_{k,i,t}, rl_{s_{k,i,t+1}}, rl\_e_{k,i,t}, w_{i,t}, ww_{k,i,t}, u_{k,i,t} \geq 0 \quad \forall i \in Vcr, k, t \tag{33}$$

$$x_{k,i,j,t}, y_k, z_{k,t} \in \{0,1\} \tag{34}$$

The objective function Eq. (1) maximize profits of revenue after deducting fuel costs, the cost of using a vehicle, driver wage cost, penalty cost and overtime cost. Equation (2) ensures that any customer *j* is visited only once by one vehicle *k*. Equations (3)-(4) guarantee that each route of the vehicle *k* must start and finish at a depot during period *t*. Equations (5)-(6) ensure that the routing continuity for each vehicle visits the customer and rest area in each period. Equation (7) ensures the rest areas allow more than one vehicle to visit. Equation (8) guarantees that vehicle *k* cannot begin traveling from the rest area to customer *i* in the first period. Equation (9) ensures that vehicle *k* cannot travel between the rest area *i* and rest area *j* during the same period *t*. Equation (10) guarantees that vehicle *k* cannot travel from depot or customer *i* to rest area *j* on the last period. Equations (11)-(12) are used to update the numbers of vehicle *k* during period *t*. Equations (13)-(16) are used to update starting service and arrival times. Equation (17) explains the time window constraints which are acceptable times for the customer. Equation (18) explains the calculation of the time window that ensures that the service is never delayed beyond the acceptable time limit specified by the customer. Equation (19) is the time of vehicle *k* arriving at the customer late, which is outside the time window condition. Equations (20)-(21) are the times for overtime. Equation (22) ensures that the initial load at the depot must not exceed the vehicle capacity. Equation (23) ensures that the loads when traveling between routes must be less than or equal to the vehicle capacity. Equation (24) ensure that the initial loads combined with the total pickup loads must be equal to the total delivery loads on each vehicle route. Equation (25) is used to update the load on the vehicle *k* leaving the depot to visit the first customer during period *t*. Equation (26) is used to update the load on the vehicle *k* visiting between customer *i* and customer *j* during period *t*. Equations (27)-(28) are used to update the load on the vehicle *k* visiting the rest area. Equations (29)-(30) guarantee that the vehicle must finish all shipping with the empty vehicle before completion at the depot. Equation (31) ensures the adjustment of the balanced load at rest area *i* before transfer to the next period. Equation (32) ensures sub-tour conditions. Equation (33) is for continuous decision variables. Equation (34) is for binary decision variables.

3. Hybrid Genetic Algorithm with Variable Neighborhood Search Algorithm (GAVNS)

3.1. Genetic Algorithm

For problems that are too large-sized and too complicated to be solved by exact solution methods, in this research, the meta-heuristic using a Genetic Algorithm (GA) is applied. A Genetic Algorithm is a class of models that mimics natural evolution aiming to find methods that combine directed and stochastic search that can produce a balance between exploration and exploitation of the search space. The GA was pioneered in 1975 [27]. Although there are newer algorithms currently available, GA is still accepted, especially when applying the hybrid method with another algorithm [17]. The general structure of a genetic algorithm has operators that consist of encoding, decoding, crossover, mutation and selection. This algorithm can be described as shown in Fig. 2.

procedure: Traditional Genetic Algorithm
input: MVRPMPDDR data, GA parameters (*popSize, maxGen*, p_c, p_m)
output: the best route
begin
  $t \leftarrow 0$; // *t*: generation number
  - initialize *P*(*t*) by chromosomes operation-based encoding procedure; // *P(t)*: population
  - evaluate *P*(*t*) by chromosomes operation-based

decoding procedure and keep the best solution;
while $t \leq maxGen$ // *maxGen*: maximize generation
- generate $F(t)$ from $P(t)$ by Weight mapping crossover; // $F(t)$: offspring
- generate $F(t)$ from $P(t)$ by Swap mutation procedure;
- evaluate $F(t)$ by chromosomes operation-based decoding procedure and keep the best solution;
- select $P(t+1)$ from $P(t)$ and $F(t)$ by roulette wheel selection;

$t \leftarrow t+1$;
endwhile
end

Fig. 2. Illustration of Traditional Genetic Algorithm.

3.1.1. Initial population

In the encoding procedure, the number of the population is the number of chromosomes that will be used in a genetic algorithm iteration. The initial chromosome population is randomly generated as an integer number according to the customer number in each chromosome, because it is easily encoded for the conventional operators in the genetic algorithm. For the encoding procedure in this research, the nodes are arranged alternately between a customer's requirement for pickup and delivery, by arranging the pickup node first, followed by the delivery node, until achieving the needs of all customers, as shown in Fig. 3.

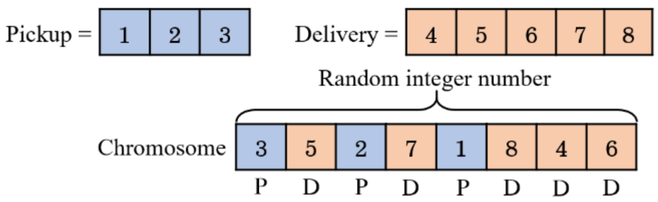

Fig. 3. Example of encoding procedure.

In the decoding procedure, chromosomes are decoded to evaluate the fitness, with the constraints of the multi-period vehicle routing problem with mixed pickup and delivery and considering constraints consisting of the time windows for customers, capacity of fleet, multi-periods, rest area for fleet and penalty costs as shown in Fig. 4. From this Fig., we can describe the operation as follows.

procedure: Decoding Procedure
input: MVRPMPDDR data
output: the best route
begin
Step 1: $\{U\} \leftarrow$ randomly generated in the integer number for the customer i unvisited
// U: set of the undefined sequence of customers i
$i \leftarrow 1$; // i: the customers i
$k \leftarrow 1$; // k: the vehicle number
$t_k \leftarrow 1$; // t_k: period of service by vehicle k
Step 2: **while** $\{U\} \neq \emptyset$ **do**
for i = 1: U
if check time window constraints of customer j
- update cumulative time of vehicle k ($ct_{i,k,t}$)

if check capacity constraints of the vehicle k for pickup j or delivery j
- update cumulative products on vehicle k ($cv_{i,k,t}$)

endif
endif
if check duration time constraints of the vehicle k should be returned to rest area or depot
- update cumulative time of vehicle k returned to rest area ($ct_{rest,k,t}$) or depot ($ct_{depot,k,t}$)

if the vehicle k returned to rest area ($t_k < MDT$) // *MDT*: maximum duration time
$t_k \leftarrow t_k+1$;
else the vehicle k returned to depot ($t_k = MDT$)
$t_k \leftarrow 1$;
$k \leftarrow k+1$;
endif
- update customers i to $\{S\}$ and delete customers i in $\{U\}$

// S: set of the defined sequence of customers i
endif
$i \leftarrow i+1$;
endfor
endwhile
Step 3: **if** check cumulative products on vehicle k returned to depot ($cv_{depot,k,t}$) > 0 **do**
- repair set $\{S\} \rightarrow$ set $\{U\}$ then return to step 2

endif
Step 4: update size of set of the vehicles k
Step 5: fitness evaluation by objective function
Step 6: trade-off between allowed penalty costs or vehicle costs for using the new vehicle
Step 7: select the best solution
End
return route $\{S\}$;

Fig. 4. Illustration of Decoding Procedure.

Step 1: Generate an initial solution; it is randomly constructed in the integer number according to the customer number in the chromosome and adds to undefined sequences in set U.

Step 2: The customer constraints in set U are considered. in order to be chosen for the route of the vehicle k with the following steps:
- Check time window constraints of existing customers.

DOI:10.4186/ej.2021.25.10.71

- Check current capacity constraints of the vehicle (loading or unloading of vehicle).
- Check maximum duration time constraints of the vehicle route (check period).
- Update customers *i* to the defined sequence in set *S* and delete customers *i* in set *U*.
- Repeat until the route is complete (set $U=\emptyset$).

Step 3: Check cumulative products on vehicle that must be empty at the end of the trip.
If loading vehicles are not empty, the defined sequence must be repaired in set *S*, then return to step2.

Step 4: Update fleet size.

Step 5: Fitness evaluation by objective function.

Step 6: Trade-off objective function between allowing penalty cost or vehicle cost for using the new vehicle.

Step 7: Select the best solution.

3.1.2. Crossover procedure

For the crossover procedure, a weight-mapping crossover (WMX) was developed in the genetic algorithm because is easier than a one cut-point crossover, since the chromosome repair process is created mechanically in the weight-mapping crossover. By pairing all parent chromosomes, a cut-point is picked randomly and designated as a crossover point. The decision for crossover is randomly generated as a uniform number within [0,1). If the random value is within the range of the crossover rate, it will create the offspring by using the segment of own parent to the left of the cut-point, then sort the nodes from ascending and remapping for the right segment of other parents [28] as shown in Fig. 5.

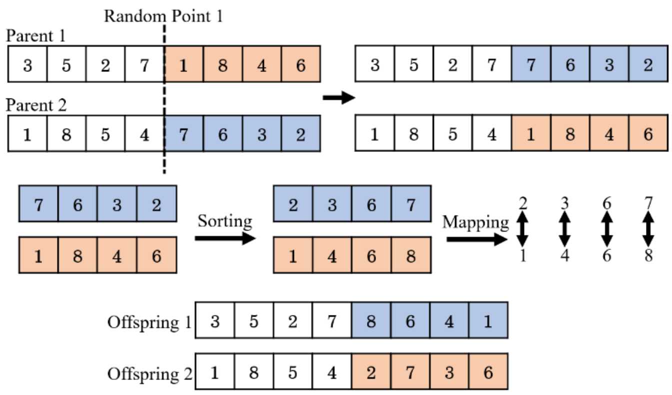

Fig. 5. Example of Weight Mapping Crossover (WMX).

3.1.3. Mutation procedure

The mutation procedure helps avoiding being trapped in local minima. In this research, a swap mutation is used. Similar to the crossover procedure, a cut-point is picked randomly and designated a mutation point for two allele values in a string. If the random value from a uniform number is within the range of the mutation rate, there will be an exchange of two allele values within the chromosome. Furthermore, determining the number of mutant points must be considered according to the chromosome length in proportion. Therefore, the proportion of mutation points should be determined according to the chromosome length.

3.1.4. Selection procedure

One of the major genetic operator components is the selection procedure, because it significantly affects the best solution. the roulette wheel selection was used in this research. It will generate the wheel according to the cumulative probability of each chromosome, then the survival chromosome is chosen by randomly selecting the points in the zone of the probability wheel, which survives for the next evolution procedure.

3.2. Variable Neighborhood Search Algorithm

The Variable neighborhood search algorithm (VNS) is a meta-heuristic that uses a local search principle to systematically search solution space, with the concept of the neighborhood change which has more than one type of neighborhood structure. The change of neighborhood handles descent to a local minimum and escaping from the valley which contains it. The VNS method begins with the initial solution and improves it by applying operations in two nested levels, consisting of shake and local search. The shaking level is used for diversifying the search in the solution space, while the local search level is used for intensifying the search within the current local neighborhood [25]. The basic VNS pseudo-code is explained in Fig. 6.

procedure: Variable Neighborhood Search
input: VNS parameters ($maxGen$, $kmax$, N_k);
output: the best route
begin
    $t \leftarrow 0$; // t: generation number
    - initialize $F(t)$ by chromosomes operation-based encoding procedure to find the initial solution.
    While $t \le maxGen$
        for $i \leftarrow 1 : NP$ chromosomes
            $k \leftarrow 1$
            While $k \le k_{max}$
                - generate $F'(t)$ from $F(t)$ by shaking: Generate a random point F' in neighborhood structures ($F'(t) \in N_k(F(t))$);
                - evaluate $F'(t)$ by a chromosome operation-based decoding procedure and keep the best solution;
                While $F''(t) \ge F'(t)$
                    - generate $F''(t)$ from $F'(t)$ randomly in the local search structures;
                    - evaluate $F''(t)$ by chromosomes operation-based decoding procedure and keep the best solution;

DOI:10.4186/ej.2021.25.10.71

```
                if F"(t) > F'(t)
                    -update F'(t) from F"(t)
                endif
            endwhile
            if F'(t) > F(t)
                -update F(t) from F'(t) and k ← 1;
            else
                k ← k+1;
            endif
        endwhile
        i ← i+1
    endfor
    t ← t+1
  endwhile
end
```

Fig. 6. Illustration of Variable Neighborhood Search.

3.2.1. Shake and local search procedure

The shake procedure is used to select neighborhood structures that are used to find the solution. In this paper, we designed neighborhood structures for the proposed VNS with four neighborhood structures. The step for neighborhood selection is chosen sequentially in the neighborhood structure (N_k). When an improved solution obtained from the shaking procedure is entered into the step of the local search, the result is compared with the result of the initial solution. If the result is better, it will be updated instead of the result of the initial solution, in order to be the result for comparison in the next iteration. The stopping criteria are considered from the result of a current neighborhood that is not improved when compared to the result of the initial solution. Then neighborhoods are moved to the search space of the next neighborhood (N_{k+1}), which iterates until all the neighborhoods are complete.

After the shaking procedure, the local search is applied to intensify the search within the current local neighborhood. In this research, four types of local searches were used. The steps to solve the problem are the same procedure as the four shaking in the neighborhood structure, but one of these operators is randomly selected for application. After that, the result is compared with the results of the current neighborhood. If the result is better, it will be updated instead of the result of the current neighborhood in order to be the result for comparison in the next iteration local search. The stopping criteria are considered from the result of local search that is not improved when compared to the result of current neighborhoods. Therefore, the result has been made more effective before searching for results in the search space of the next neighborhood structure.

Neighborhood Structures N_1 (Swap)

The swap principle is the exchange of the node positions by randomly selecting a cut-point and choosing two different nodes i_1 and i_2, then exchanging between i_1 and i_2, in which the iteration of the swap depends on 15% of the number of customers as shown in Fig. 7.

Neighborhood Structures N_2 (Insert)

The insert principle is to move a node to another node by randomly selecting a cut-point and choosing two different nodes i_1 and i_2, then removing node i_1 to the position of i_2, in which the iteration of the insert depends on 15% of the number of customers as shown in Fig. 8.

Neighborhood Structures N_3 (K cyclic move)

The K cyclic move principle is a systematic motion by selecting a cyclic point randomly. The nodes are systematically removed to replace other positions nodes, in which the number of nodes of the move depends on 30% of the number of customers (it equals 15% for the iterations of the swap, insert and transpose) as shown in Fig. 9.

Neighborhood Structures N_4 (Transpose)

The transpose principle exchanges adjacent positions by selecting a cut-point randomly and choosing two adjacent position nodes i_1 and i_2, then adjacent positions are exchanged with node i_1 to the position of i_2, in which the iteration of the transpose depends on 15% of the number of customers as shown in Fig. 10.

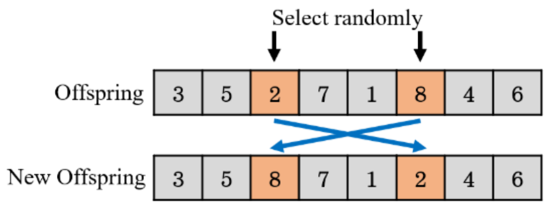

Fig. 7. Illustration of Neighborhood Structures N_1 (Swap).

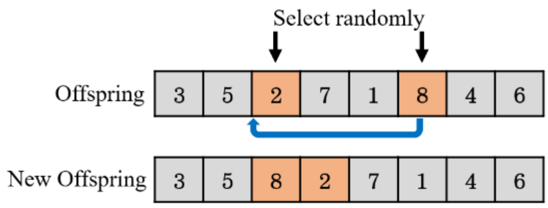

Fig. 8. Illustration of Neighborhood Structures N_2 (Insert).

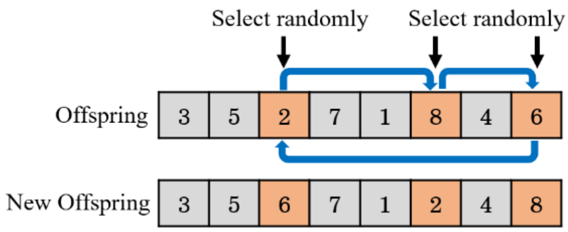

Fig. 9. Illustration of Neighborhood Structures N_3 (K cyclic move).

DOI:10.4186/ej.2021.25.10.71

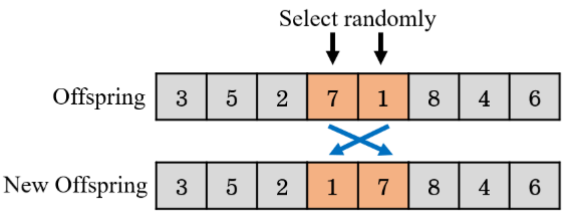

Fig. 10. Illustration of Neighborhood Structures N_4 (Transpose).

3.3. Hybrid GA-VNS Algorithm

The structure of the hybrid genetic algorithm with variable neighborhood search algorithm consists of an initial solution, crossover operator, mutation operator and selection with the roulette wheel. The variable neighborhood search algorithm is specially applied instead of the mutation operator, because we want to reduce duplicate solutions of the algorithms that increase the difficulty and are time-consuming. This approach significantly decreases computational time and leads to better results. The initial solution of the genetic algorithm is used in the first step of the variable neighborhood search. This algorithm can be described as shown in Fig. 11.

procedure: Hybrid Genetic Algorithm
input: MVRPMPDDR data, GA parameters (*popSize, maxGen*, p_c, p_M), VNS parameters (*maxGen*, *kmax*, N_k) ;
output: the best route
begin
  $t \leftarrow 0$; // t: generation number
  - initialize $P(t)$ by chromosomes operation-based encoding procedure; // $P(t)$: population
  - evaluate $P(t)$ by chromosomes operation-based decoding procedure and keep the best solution;
  while $t \leq maxGen$ // *maxGen*: maximize generation
    - generate $F(t)$ from $P(t)$ by Weight mapping crossover; // $F(t)$: offspring
    - evaluate $F(t)$ by chromosomes operation-based decoding procedure and keep the best solution;
    - Variable Neighborhood Search Algorithm
    - select $P(t+1)$ from $P(t)$ and $F(t)$ by roulette wheel selection;
    $t \leftarrow t+1$;
  endwhile
end

Fig. 11. Illustration of Hybrid GA-VNS algorithm.

4. Experiment and Results

4.1. Computational Framework

In order to test the model of the hybrid genetic algorithm with the variable neighborhood search algorithm we solve the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, duration time and rest area to maximize profits to the entrepreneur. In this section the algorithm is tested with 3 sub-algorithms which are VNS, GA, and GAVNS; it was also validated by comparing the solutions with the optimal solution obtained by Lingo v.17 for Windows software and the proposed algorithms were run with MATLAB R2017a on a 2.5 GHz PC, with 12.00 GB of RAM and Windows 10 Pro operating system. The performance of the proposed methods was tested using 2 sizes of problem instances. Details of generated data are shown in Table 2.

Table 2. Size of tested problems.

Problem	Data set	Problem size	No. of nodes		
			Pickup	Delivery	Rest area
1	A	S	3	4	1
	B	S	3	4	1
	C	S	3	4	1
2	A	S	4	6	2
	B	S	4	6	2
	C	S	4	6	2
3	A	S	5	7	2
	B	S	5	7	2
	C	S	5	7	2
4	A	S	7	8	2
	B	S	7	8	2
	C	S	7	8	2
5	A	L	10	10	3
	B	L	10	10	3
	C	L	10	10	3
6	A	L	13	17	5
	B	L	13	17	5
	C	L	13	17	5
7	A	L	17	23	6
	B	L	17	23	6
	C	L	17	23	6
8	A	L	24	26	8
	B	L	24	26	8
	C	L	24	26	8
9	A	L	27	33	9
	B	L	27	33	9
	C	L	27	33	9
10	A	L	35	45	12
	B	L	35	45	12
	C	L	35	45	12

Table 2 shows that 10 problem instances were used to test the algorithm. Each problem is separated into 3 sub-

DOI:10.4186/ej.2021.25.10.71

problems (A, B and C problems). So, in total there were 30 problems that consist of small and large instances divided into 12 and 18 respectively. The numbers of pickup, delivery, and rest area nodes are assigned according to the problem size. The time windows with starting and ending times are defined between 2 to 4 hours for each customer. If the customer is not served in his time window, the entrepreneur will be responsible for penalty costs according to the delay time rate. And the duration time for each vehicle to give service is 5 periods, with a period of giving service of 8 hours. If the service time of any vehicle exceeds 8 hours, it will affect overtime costs. For the assigned parameters of the GA experiment, the maximum generation is 500, population size (*NP*) is 10, the crossover rate (P_c) is 0.8, and the mutation rate (P_m) is 0.2, which is the value used mostly in the literature. Furthermore, the assigned parameters of the VNS experiments, the maximum number of neighborhoods (*kmax*) and neighborhood structures (N_k) are set to 4. All proposed algorithms are derived from the calculation of 5 replications.

4.2. Computational Results

In this section, computational results are presented by the profit values obtained from the optimal solutions of the mathematical model using mixed-integer linear programming and the profit values obtained from the best solutions of the proposed algorithms and computational time, as shown in Table 3 for all-sized problems. A comparison of the percentage efficiency and relative improvement of the solution generated by the proposed algorithms is shown in Table 4. The final result is the performance analysis of each algorithm, which was tested by statistical tests for all-sized problems as shown in Table 5.

Table 3. Computational results for all-sized problems.

Problem	Data set	Problem size	Profits (unit)				Computational time (sec.)			
			Optimal	VNS	GA	GAVNS	Optimal	VNS	GA	GAVNS
1	A	S	4,204	4,204	4,204	4,204	3.41	4.50	5.30	43.06
	B	S	6,279	6,279	6,279	6,279	2.90	9.43	10.46	85.40
	C	S	7,264	6,274	6,274	6,274	8.07	42.93	7.71	59.61
2	A	S	8,933	8,888	8,888	8,888	52.44	11.72	11.53	107.26
	B	S	13,392	13,338	13,338	13,338	93.70	11.91	13.76	100.75
	C	S	13,969	13,900	13,900	13,900	1,055.41	99.65	14.78	129.32
3	A	S	14,281	14,232	14,232	14,232	303.34	12.68	23.91	121.93
	B	S	13,892	13,816	13,816	13,816	781.45	21.88	24.19	301.20
	C	S	16,862	15,701	15,707	15,707	5,255.46	32.22	40.34	354.79
4	A	S	18,446	18,446	18,446	18,446	31,900.63	19.17	36.79	166.42
	B	S	17,502	17,290	17,277	17,290	22,735.13	34.94	33.33	330.19
	C	S	16,287	16,211	16,211	16,211	2,123.24	22.06	43.61	263.53
5	A	L	-	19,894	17,465	19,894	-	49.74	44.09	213.58
	B	L	-	19,001	18,283	19,001	-	62.84	71.93	892.09
	C	L	-	17,824	17,786	17,824	-	64.51	68.46	784.75
6	A	L	-	24,514	24,359	24,677	-	161.49	129.08	1,064.20
	B	L	-	26,901	26,702	26,917	-	111.95	148.77	2,191.60
	C	L	-	27,563	26,690	26,892	-	167.55	188.91	2,416.80
7	A	L	-	24,491	22,943	24,647	-	121.71	184.89	2,354.20
	B	L	-	25,080	24,860	26,493	-	195.52	301.06	4,431.30
	C	L	-	25,861	23,393	26,157	-	286.73	246.24	4,182.20
8	A	L	-	38,234	36,852	38,838	-	494.86	404.02	8,116.30
	B	L	-	40,142	39,687	40,930	-	635.59	615.88	7,031.00
	C	L	-	37,856	35,601	37,958	-	812.11	782.56	11,219.00
9	A	L	-	57,279	55,626	61,060	-	424.81	380.54	4,947.90
	B	L	-	64,351	60,411	65,422	-	990.86	740.15	15,265.00
	C	L	-	63,084	62,269	64,383	-	871.30	780.95	12,339.00
10	A	L	-	66,082	62,793	67,547	-	742.87	526.76	9,141.10
	B	L	-	72,870	70,075	75,671	-	908.67	1,068.10	12,413.00
	C	L	-	70,970	67,877	71,126	-	1,526.80	1,533.90	20,321.00

In Table 3, all proposed algorithms were able to find comparable results to the optimal solution for the small-sized problems. In with some problem instances, proposed algorithms achieved the optimal solution in a better computational time than the calculation of the mathematical model. For the large-sized problems, the mathematical model could not find the optimal solution. We can see the best solution for the GAVNS algorithm showed better profit than the VNS and GA algorithms, but the computational times are longer due to the complex search logic. Table 4 shows a comparison of percentage efficiencies of the profits from the best solution of the proposed algorithms and the optimal solution using the mathematical model as determined by Eq. (35). In problem instances 1 to 4 for the small-sized problems, the comparison of the average efficiency between the best solution of the VNS algorithms with the optimal solution is 97.959%, the comparison result of the GA algorithms with the optimal solution is 97.956% and the comparison result of GAVNS algorithms with the optimal solution is 97.962%. The results show that GAVNS algorithms can find the results closest to the optimal solution. In problem instances 5 to 10 for the large-sized problems, an optimal solution cannot be obtained by the mathematical model, due to the multi-period vehicle routing problem with mixed pickup and delivery being NP-hard, and the solution being highly complex. Therefore, we used the proposed algorithms, which gave an acceptable solution compared to the optimal solution for solving large-sized problems. The results of the average relative improvement of the solution generated by the proposed algorithms were determined by Eq. (36). It shows that in a comparison between the GAVNS and the VNS algorithms, the relative improvement in the profits averaged 0.79% and in a comparison between the GAVNS and GA algorithms, the relative improvement in the profits averaged 3.18%. So, this shows that the GAVNS algorithm is better than the VNS and GA algorithms.

Table 4. Percentage efficiency and relative improvement of the solution generated by proposed algorithms.

Problem	Data set	Efficiency (%)			RI (%)	
		VNS	GA	GAVNS	GAVNS & VNS	GAVNS & GA
1	A	100.00	100.00	100.00	0.00	0.00
	B	100.00	100.00	100.00	0.00	0.00
	C	86.36	86.36	86.36	0.00	0.00
2	A	99.50	99.50	99.50	0.00	0.00
	B	99.60	99.60	99.60	0.00	0.00
	C	99.51	99.51	99.51	0.00	0.00
3	A	99.66	99.66	99.66	0.00	0.00
	B	99.45	99.45	99.45	0.00	0.00
	C	93.12	93.15	93.15	0.04	0.00
4	A	100.00	100.00	100.00	0.00	0.00
	B	98.79	98.72	98.79	0.00	0.08
	C	99.53	99.53	99.53	0.00	0.00
5	A	-	-	-	0.00	13.91
	B	-	-	-	0.00	3.93
	C	-	-	-	0.00	0.21
6	A	-	-	-	0.66	1.31
	B	-	-	-	0.06	0.81
	C	-	-	-	0.22	3.50
7	A	-	-	-	0.64	7.43
	B	-	-	-	5.63	6.57
	C	-	-	-	1.14	11.82
8	A	-	-	-	1.58	5.39
	B	-	-	-	1.96	3.13
	C	-	-	-	0.27	6.62
9	A	-	-	-	6.60	9.77
	B	-	-	-	1.66	8.29
	C	-	-	-	2.06	3.39
10	A	-	-	-	2.22	7.57
	B	-	-	-	3.84	7.99
	C	-	-	-	0.22	4.79
Average (%)		97.959	97.956	97.962	0.79	3.18

$$Proposed\ Solution_{EF(\%)} = \frac{R_{Proposed\ Solution}}{R_{OPT}} \times 100 \quad (35)$$

where

$Proposed\ Solution_{EF(\%)}$ = Efficiency of the VNS, GA, and GAVNS (%)

R_{OPT} = The result of the mixed integer linear programming

$R_{Proposed\ Solution}$ = The result of the VNS, GA, and GAVNS

$$RI(\%) = \frac{(R_{first} - R_{second})}{R_{second}} \times 100 \quad (36)$$

where

$RI(\%)$ = Relative improvement compared between R_{first} and R_{second}

R_{first}, R_{second} = The result of the VNS, GA, and GAVNS

The statistical test was performed using IBM SPSS Software V.26 using the paired samples t-test at a statistical significance of 0.05. The result is shown in Table 5. The results of the statistical tests show that the solutions obtained from the proposed algorithms compared with the optimal solution of the mathematical model for small-sized problems were not significantly different at $\alpha = 0.05$. This means that we can conclude that our proposed algorithms are efficient. In large-sized problems, we compare the solutions of the proposed algorithm consisting of the VNS, GA and GAVNS. The results of the tests showed the solutions obtained from the three algorithms were significantly different at $\alpha = 0.05$, with the GAVNS algorithm showing a better result than the VNS and GA algorithms. This has demonstrated the performance of the search results for GAVNS algorithm.

Table 5. Statistical test for all-sized problems.

	P-value					
	Small-sized problems			Large-sized problems		
Method	VNS	GA	GAVNS	VNS	GA	GAVNS
Optimal	0.075	0.074	0.075	-	-	-
VNS	-	0.646	0.339	-	0.000	0.011
GA	-	-	0.339	-	-	0.000

5. Discussion

This research focuses on the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, service duration time of vehicles and rest area for vehicles (MVRPMPDDR). The objective is to sequence the pickup and delivery nodes of vehicles to maximize profits, which are comprised of revenue after deducting fuel cost, the cost of using a vehicle, driver wage cost, penalty cost and overtime cost. We present a mixed-integer linear programming model that can handle small-sized problems (instances 1 to 4). For large-sized problems (instances 5 to 10), a solution could not be found using LINGO, since the vehicle routing problem with pickup and delivery is NP-hard, so the solution is highly complex, and the numbers of variables were excessive. So, a genetic algorithm (GA) was applied to solve the problems. In addition, we developed a hybrid algorithm that is a hybrid genetic algorithm with a variable neighborhood search algorithm (GAVNS) to increase the efficiency in finding the solution of the algorithm.

In our research, the development of the GAVNS algorithm is due to inserting the VNS algorithm for solution search for the GA algorithm, which is applied instead of the mutation operator to increase the diversity of the local search space systematically. For GAVNS operators, including the initial population, the crossover used the weight mapping crossover, while the mutation instead used the VNS algorithm, and the selection procedure used roulette wheel selection. The VNS method begins with the initial solution provided by the GA algorithm. and improves it by applying operations in two nested levels consisting of shake and local search by shake first, then local search. The shake procedure was designed for four neighborhood structures (N_k) consisting of Swap, Insert, K cyclic move and Transpose. The local search procedure was designed in the same way as the four neighborhood structures of the shake procedure. The results for the proposed algorithm show that the GAVNS algorithm outperforms the VNS and GA algorithms. In this paper, mathematical and heuristics models have been developed to solve the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, duration time and rest area, and this method should prove to be beneficial to other transportation industries in Thailand and around the world by increasing profits for transportation entrepreneurs from an efficient transportation management system.

6. Conclusions

This research is the multi-period vehicle routing problem with mixed pickup and delivery with time windows, heterogeneous fleet, service duration time of vehicles and rest area for vehicles (MVRPMPDDR). Mixed-integer linear programming gave the optimal solution for small sized problems (instances 1 to 4), but for large-sized problems (instances 5 to 10), the solution could not be found using LINGO because there were many variables and limitations. Accordingly, we developed a hybrid algorithm that is a hybrid genetic algorithm with a variable neighborhood search algorithm (GAVNS) to increase the solution efficiency to be comparable to that of mixed-integer linear programming for solving the problems. The results for the GAVNS algorithm were better than both the VNS and GA algorithms by 0.79% and 3.18% respectively. The GAVNS algorithm could find the best solutions efficiently and was the nearest to the optimal solution. However, we believe that this issue can be extended to other real-world problem models and will be valuable in the future. There are many limitations to this study that can be improved to develop approaches with vehicle routing in real-world problems. Future development will expand product categories to multiple products and the total demand of customers receiving pickups and deliveries from more than one vehicle (split demand). Furthermore, we will consider pickup and delivery activities occurring at the rest area. Moreover, we may consider traffic density for vehicle route restrictions. The proposed method modifications and development approaches using the GAVNS algorithm and other meta-heuristics or hybrid methods may improve the solution efficiency.

Acknowledgements

This work was supported by the Research Unit on System Modeling for Industry (Grant No. SMI.KKU MS 601007), Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand. The authors also thank Mr. Ian Thomas for linguistic corrections.

References

[1] S. Joshi and S. Kaur, "Nearest neighbor insertion algorithm for solving capacitated vehicle routing problem," in *Proceedings of the 2nd International Conference on Computing for Sustainable Global Development*, India, IEEE, 2015, pp. 86-88.

[2] D. Cinar, K. Gakis, and P. M. Pardalos, "A 2-phase constructive algorithm for cumulative vehicle routing problems with limited duration," *Expert Systems with Applications*, vol. 56, pp. 48-58, 2016.

[3] M. Amous, S. Toumi, B. Jarboui, and M. Eddaly, "A variable neighborhood search algorithm for the capacitated vehicle routing problem," *Electronic Notes in Discrete Mathematics*, vol. 58, pp. 231-238, 2017.

[4] A. Supithak and W. Supithak, "Determination of inventory replenishment policy with the open vehicle routing concept in a multi-depot and multi-retailer distribution system," *Engineering and Applied Science Research*, vol. 45, no. 1, pp. 23-31, 2018.

[5] M. Avci and S. Topaloglu, "A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery," *Expert Systems with Applications*, vol. 53, pp. 160-171, 2016.

[6] D. Dechampai, L. Tanwanichkul, K. Sethanan, and R. Pitakaso, "A differential evolution algorithm for the capacitated VRP with flexibility of mixing pickup and delivery services and the maximum duration of a route in the poultry industry," *Journal of Intelligent Manufacturing*, vol. 28, no. 6, pp. 1357-1376, 2017.

[7] C. K. Ting, X. L. Liao, Y. H. Huang, and R. T. Liaw, "Multi-vehicle selective pickup and delivery using metaheuristic algorithms," *Information Sciences*, vol. 406, pp. 146-169, 2017.

[8] N. Bhusiri, A. G. Qureshi, and E. Taniguchi, "The trade-off between fixed vehicle costs and time-dependent arrival penalties in a routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 62, pp. 1-22, 2014.

[9] C. Wang, D. Mu, F. Zhao, and J. W. Sutherland, "A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows," *Computers & Industrial Engineering*, vol. 83, pp. 111-122, 2015.

[10] A. Ahkamiraad and Y. Wang, "Capacitated and multiple cross-docked vehicle routing problem with pickup, delivery, and time windows," *Computers & Industrial Engineering*, vol. 119, pp. 76-84, 2018.

[11] S. Mancini, "A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based metaheuristic," *Transportation Research Part C: Emerging Technologies*, vol. 70, pp. 100-112, 2016.

[12] Z. Al Chami, H. Manier, M. A. Manier, and E. Chebib, "An advanced grasp-hga combination to solve a multi-period pickup and delivery problem," *Expert Systems with Applications*, vol. 105, pp. 262-272, 2018.

[13] H. Larrain, L. C. Coelho, C. Archetti, and M. G. Speranza, "Exact solution methods for the multi-period vehicle routing problem with due dates," *Computers & Operations Research*, vol. 110, pp. 148-158, 2019.

[14] K. Sethanan and T. Jamrus, "Hybrid differential evolution algorithm and genetic operator for multi-trip vehicle routing problem with backhauls and heterogeneous fleet in the beverage logistics industry," *Computers & Industrial Engineering*, vol. 146, p. 106571, 2020.

[15] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving the multi depot vehicle routing problem," *Applied Soft Computing*, vol. 27, pp. 519-532, 2015.

[16] M. A. Mohammed, M. K. A. Ghani, R. I. Hamed, S. A. Mostafa, M. S. Ahmad, and D. A. Ibrahim, "Solving vehicle routing problem by using improved genetic algorithm for optimal solution," *Journal of Computational Science*, vol. 21, pp. 255-262, 2017.

[17] C. Chamnanlor, K. Sethanan, C. F. Chien, and M. Gen, "Hybrid genetic algorithms for solving reentrant flow-shop scheduling with time windows," *Industrial Engineering and Management Systems*, vol. 12, no. 4, pp. 306-316, 2013.

[18] C. Chamnanlor, K. Sethanan, M. Gen, and C. F. Chien, "Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints," *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1915-1931, 2017.

[19] W. Neungmatcha, K. Sethanan, M. Gen, and S. Theerakulpisut, "Adaptive genetic algorithm for solving sugarcane loading stations with multi-facility services problem," *Computers and Electronics in Agriculture*, vol. 98, pp. 85-99, 2013.

[20] T. Jamrus, C. F. Chien, M. Gen, and K. Sethanan, "Multistage production distribution under uncertain demands with integrated discrete particle swarm optimization and extended priority-based hybrid genetic algorithm," *Fuzzy Optimization and Decision Making*, vol. 14, no. 3, pp. 265-287, 2015.

[21] P. K. Nguyen, T. G. Crainic, and M. Toulouse, "A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows," *Journal of Heuristics*, vol. 20, no. 4, pp. 383-416, 2014.

[22] T. Vidal, M. Battarra, A. Subramanian, and G. Erdogan, "Hybrid metaheuristics for the clustered vehicle routing problem," *Computers & Operations Research*, vol. 58, pp. 87-99, 2015.

[23] H. Xia, X. Li, and L. Gao, "A hybrid genetic algorithm with variable neighborhood search for dynamic integrated process planning and scheduling," *Computers & Industrial Engineering*, vol. 102, pp. 99-112, 2016.

[24] R. K. Arakaki and F. L. Usberti, "Hybrid genetic algorithm for the open capacitated arc routing problem," *Computers & Operations Research*, vol. 90, pp. 221-231, 2018.

[25] A. Baniamerian, M. Bashiri, and R. Tavakkoli-Moghaddam, "Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking," *Applied Soft Computing*, vol. 75, pp. 441-460, 2019.

[26] S. W. Hnin and C. Jeenanunta, "Bayesian optimization in a support vector regression model for short-term electricity load forecasting," *Engineering and Applied Science Research*, vol. 46, no. 3, pp. 267-275, 2019.

[27] J. H. Holland, *Adaption in Natural and Artificial System.* Ann Arbor, MI: University of Michigan Press, 1975.

[28] J. E. Lee, M. Gen, and K. G. Rhee, "Network model and optimization of reverse logistics by hybrid genetic algorithm," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 951-964, 2009.

Kittitach Kamsopa received the B.S in degrees of science program in statistics from Khon Kaen University, Thailand, in 2014 and M.S. degree of engineering program in industrial and logistics engineering management from Khon Kaen University, Thailand, in 2019. Currently, he is working in warehouse of company managed by Warehouse Management Systems and Local Optimize Programs.

Kanchana Sethanan was born in Roi-Ed, Thailand, in 1969. She received the M.S. degree in industrial engineering from Oklahoma State University and the Ph.D. degree in industrial engineering from West Virginia University, USA, in 2001. She is a professor with the Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand. Her research interests include optimization problem, scheduling, simulation model, supply chain and logistics, resource allocation, and transportation problem.

Thitipong Jamrus received the B.S. and M.S. degrees in industrial engineering from Khon Kaen University, Thailand, in 2009 and 2011, respectively, and the Ph.D. degree from the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan, He is an asisstance professor with the Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University. His research interests include production planning and scheduling, supply chain management, decision analysis, and operations research.

Liliana Czwajda is Centre for Industrial Applications of Mathematics and Systems Engineering, Polish Academy of Sciences, ul. Śniadeckich 8, Warsaw, Poland.