Highlights

An Agent-Based Optimisation Approach for Vehicle Routing Problem with Unique Vehicle Location and Depot

Anees Abu-Monshar,Ammar Al-Bazi,Vasile Palade

- A VRP where each vehicle has a unique location for starting and ending its route
- A hybrid agent interaction messaging protocol to construct feasible routes
- Higher quality solutions than popular for benchmarked MDVRPTW instances
- Results on missed customers are generated by the Modified MDVRPTW instances

An Agent-Based Optimisation Approach for Vehicle Routing Problem with Unique Vehicle Location and Depot

Anees Abu-Monshar[a,*], Ammar Al-Bazi[b] and Vasile Palade[c]

[a] Institute for Future Transport and Cities, Coventry University, Coventry, UK
[b] Institute for Advanced Manufacturing and Engineering, Coventry University, Coventry, UK
[c] Faculty Research Centre for Data Science, Coventry University, Coventry, UK

ARTICLE INFO

Keywords:
Vehicle Routing Problem
Unique Vehicle Location and Depot
Agent-Based Modelling
Optimisation
Hybrid Messaging Protocol

ABSTRACT

The Vehicle Routing Problem (VRP) is a well studied logistical problem along with its various variants such as VRP with customer Time-Window (VRPTW). However, all the previously studied variants assume that vehicles are mostly the same in terms of their capacity, location and home location (depot). This study uses the agent-based approach for solving VRPTW with vehicle's unique location and depot. This is to minimise the number of used vehicles as the main target. Other targets including total distance travelled, waiting time and time are also considered as criteria to evaluate the quality of the generated vehicle routes. This is achieved by proposing a Messaging Protocol-based Heuristics Optimisation (MPHO) model that balances between centrally-distributed agents' interactions and accommodates certain priority rules specifically developed for the problem. Furthermore, modifications to certain constraints checking techniques are introduced by implementing time Push Forward (PF) checking recursively tailored to the route's unique start/ending locations as well as calculating the reduced waiting time to find and check the limit of the total route duration. In order to justify the superiority of the proposed MPHO model, numerical tests have been conducted on benchmark problems including single and multiple depot instances as well as modified instances tailored to the problem. This is made possible by randomising vehicles' capacities and their unique locations and depots. Key results reveal that, in multiple depot instances, higher quality solutions compared with previous benchmark outcomes are obtained in terms of minimising the total number of vehicles along with fastest solution time (CPU) at the expense of total time and distance travelled.

1. Introduction

Vehicle Routing Problem (VRP) is one of the well-known logistical problems that was extended from the Travelling Salesman Problem (TSP) to accommodate additional constraints. The problem was first introduced by Dantzig and Ramser (1959) to provide routing plans for vehicles to visit customers' locations starting and ending at the same depot. VRP is proven as an NP-hard problem (Lenstra and Kan, 1981) and most of the previously used approaches are mainly (meta)heuristics, that provide near-optimal solutions (Laporte, 2009). An insertion heuristic was proposed by Solomon (1987) to solve a Time-Window variant (VRPTW) and Schneider (2016) adopted Tabu-Search to solve the same problem.

Although VRP has been well-explored along with its variants, predominantly those deal with the assumption of all vehicles or groups of vehicles start their routes from a specific depot and end at by the same depot. However, in reality, each vehicle could have a different start/end location. Savelsbergh and Sol (1995) highlighted the challenge in a similar problem but with pickup and delivery operations. Le et al. (2019) addressed the applicability of this paper's problem settings to the trending crowd-shipping applications where matching and routing supply (vehicles) with demand (customers), especially when the supply are individuals having different attributes. Therefore, in VRP, considering every vehicle to be unique in terms of its route start and end locations (independent depot) and heterogeneous capacities of the fleet as well as Time-Windows of the customer is a real challenge to most of the 3PL companies, and hence it is the focus of this study.

Therefore, in this study, we propose an innovative Agent-Based optimisation model for solving Vehicle Routing Problems with vehicles of unique start and end locations. Technical variants such as time-window and heterogeneous capacity of each vehicle is considered when optimising the routing plans of vehicles. This model is developed to assist transportation planners and logistics operators working in 3PL companies to manage effectively and efficiently their collection and delivery operations, and to achieve the best utilisation of their available vehicles for best customer satisfaction practice.

The novelty of this paper is that it introduces a new agent messaging protocol-based heuristics optimisation model, following the hybrid cooperation approach, to tackle problems of routing of vehicles for best customer service including collection and delivery operations given that each vehicle has a unique start and end location.

The remainder of the paper is structured as follow: An up-to-date review of relevant literature is provided in Section 2. The statement of the problem under study is presented in Section 3. The agent-based conceptual model, architecture and the proposed messaging protocol-based

*Corresponding author
monshara@coventry.ac.uk (A. Abu-Monshar); aa8535@coventry.ac.uk (A. Al-Bazi); ab5839@coventry.ac.uk (V. Palade)
ORCID(s): 0000-0001-9541-5062 (A. Abu-Monshar); 0000-0002-5057-4171 (A. Al-Bazi); 0000-0002-6768-8394 (V. Palade)

heuristics optimisation are presented in Section 4. Section 5 shows the model validation and test of accuracy of results on benchmark instances as well as other modified ones. Finally, conclusion and future recommendations are stated in Section 6.

2. Literature Review

This section reviews literature of VRPTW cases that considers a form of vehicle uniqueness, specifically in start and end locations. The closest variants encountered are the Multiple Depot (MDVRPTW) and open variants. The first is where routes initiate from different start locations and end there, however, certain vehicles are grouped in one of the locations making some vehicle share their starting and ending locations. The problem was first introduced by Cordeau et al. (2001). Mancini (2016) modelled multiple depot problem that allows the vehicles to return to any depot location, however, time windows variant was not considered. For broad review on Multiple Depot VRP without time-windows, please refer to Montoya-Torres et al. (2015). On the other hand, the open variant does not dictate the vehicle to return to a depot and its route ends at the last customer served, however, as this implies, this variant does not take into consideration a unique home location for every vehicle to return to contrary to what this paper is focused at. This variant was introduced by Repoussis et al. (2007).

In multiple depot problems, Goel and Gruhn (2008) and Goel (2010) introduced the generalised VRP combining real-life constraints which are customer/vehicle compatibility, customers with multiple locations of pickup and delivery, and vehicle fleets, each with different cost, travel time, multi-dimensional capacity, operating time and start and end locations. Bettinelli et al. (2011) analysed a multiple depot case where vehicles can freely associate with any depot with the objective of minimising the total travelled cost which includes the vehicles fixed cost and routing costs, distance and time using the branch-cut-price algorithm. Zarandi et al. (2011) considered fuzzy travel times for a location routing problem with Time-Window and multiple depots, where the number of depots and their locations has to be determined in parallel with the routes, to optimise the opening depots and routing costs. Simulated Annealing algorithm was used to deliver this objective. Xu et al. (2012) considered a multiple depot case with heterogeneous capacity fleet aiming to minimise the travel costs as well as time-window and vehicle working time violations, using Variable Neighbourhood Search (VNS) approach and they later introduced simulated annealing when accepting solutions within VNS (Yang et al., 2013). However, they only tested their method on single depot instances while it was later applied to case study of two depots (Xu and Jiang, 2014). Adelzadeh et al. (2014) considered a multiple depot case with fuzzy time-windows and heterogeneous vehicles where each vehicle has different capacity, speed and operating costs aiming to minimise the travelled distance and maximise the customer service level, using the simulated annealing approach. Dayarian et al. (2015) formulated a set partitioning for a case of multiple depot with heterogeneous capacity vehicles aimed to minimise the routes' fixed and variable costs then solved by a proposed branch-and-price algorithm with improved solution exploration to reduce the computational expense. Afshar-Nadjafi and Afshar-Nadjafi (2016) investigated the multiple depot and heterogeneous capacity fleet with time-window problem considering time-dependent travel times, where they depended on the time of departure, as it affects the travel variable cost which is aimed to be minimised along with the fixed cost associated with each vehicle. Additionally constraint of limiting the number of vehicles in depots is introduced in a later work of the same authors (Afshar-Nadjafi and Afshar-Nadjafi, 2017). The Simulated Annealing approach was used to solve the problem. Li et al. (2016) considered a variant of multiple depot where vehicles may end their routes at any depot with the objective of minimising the total travel costs by adopting a hybrid evolutionary and local search approach. Kramer et al. (2019) studied a case with multiple depots, heterogeneous capacity vehicles, periodic demand, relaxed customer time-window, certain vehicle-customer compatibility and maximum route duration as well as customers per route with the aim of minimising costs and travelled distance using an Iterated Local Search (ILS) algorithm. Alcaraz et al. (2019) investigated a case with multiple depot, heterogeneous capacity fleet, customer time-windows, vehicle-customer compatibility and mandatory route breaks and maximum duration with the possibility of outsourcing the last mile demand to minimise the total costs involved in distance travelled and outsourcing using a developed heuristic algorithm. Zhen et al. (2020) investigated the multiple depot with time-window problem considering multiple trips per vehicle aiming to minimise the total time by formulating a mathematical model then solved using a hybrid PSO and Genetic Algorithm (GA).

In open VRP problems, Brito et al. (2015) studied both close and open variant, meaning that some vehicles require return to the depot while other outsourced ones are not, with fuzzy capacities and time-windows. They used a hybrid Ant Colony Optimisation with metaheuristics with the aim of minimising total travelled distance. Schopka and Kopfer (2016) introduced the reverse open VRP, where all vehicles start their routes at their current position and end at a central depot, considering vehicles with different capacities (heterogeneous). They formulated a Mixed Integer Program (MIP) model to minimise vehicles fixed cost and travel time using an Adaptive Large Neighbourhood Search (ALNS) algorithm. Shen et al. (2018) studied the open variant with multiple depot, where vehicles start their routes from more than one depot. This problem was solved using a Particle Swarm Optimisation (PSO) for routes construction then improving the quality of solution further using a Tabu Search (TS) to minimise the driver's cost, time-window penalties, fuel and emission costs. Babagolzadeh et al. (2019) formulated an MIP for the Open variant with Two-Echelon, where goods are delivered first to intermediate satellites (smaller depots) then delivered to end customer. Additionally, goods become available for delivery at certain release time in satellites and time-windows are relaxed with penalties. The model aims at minimising fuel emission costs, penalties and total distance.

2.1. Research Gap

From the literature reviewed above, two technical variants can be seen as closely related to the problem under study, the multiple depot open variant introduced by Shen et al. (2018) and the reverse open variant introduced by Schopka and Kopfer (2016). However, the first variant is still considered as multiple depot, where certain vehicles can share starting location, and routing stops at the last customer while the second variant limits all vehicles to return to one centralised depot.

It can be concluded from the reviewed literature above all VRPs dictate that vehicles are associated with depots either at the start or end of their routes, apart from Goel and Gruhn (2008) and Goel (2010) who considered the generalised VRP that combines a wide range of constraints as stated above. However, this paper differs from Goel's variant in terms of focusing more on the modelling aspect of the unique vehicle location and home location along with other relevant constraints such as vehicle capacity and time window of customer to minimise number of vehicles as a main objective. This is a contrary to previous research in problem in the generalised VRP which broadly maximises profits of highly constrained problems. In addition, this study introduces a new agent-based optimisation model that is specifically developed to solve VRP problems with such additional features of location and capacity. This model aims at maximising customer coverage while minimising of the number of vehicles and total travelled distance.

3. Problem Statement

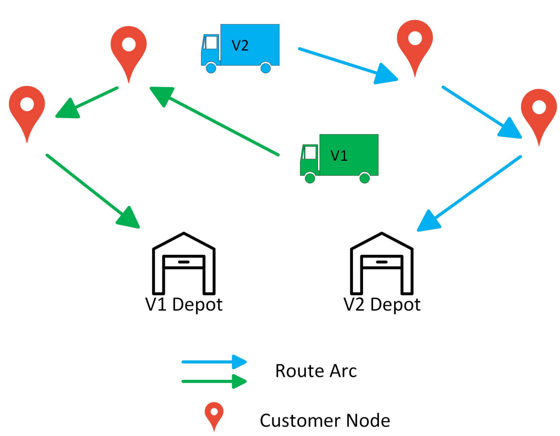

Figure 1: Problem Map

In the traditional Vehicle Routing Problem (VRP) vehicles are assumed to have the same location for their routes to start/end at their representative depots. However, this might not be a practical case since the location availability of the vehicles differs from one vehicle to another. Previously, VRP dealt with different vehicles in terms of their capacity but this problem adds to it the unique starting and ending locations of each vehicle route and hence the complexity of the problem appears. As a result, a vehicle route should start from its current location and end at its home location/depot while serving certain customers in between.

Figure 1 illustrates an example of such routes on a hypothetical map. However, it does not reflect each customer time-window constraint, which indicates the customer availability to receive the service, and the vehicles' time-shifts that indicate when the vehicle can move from its location and the deadline by when it should arrive at its depot. Additional timings are also considered in this problem which are the travel time of each route arc, servicing time at each customer node and the duration limit of the route. This paper is concerned about finding feasible vehicle routes aiming at maximising the covered customers while minimising the vehicles used and the total distance travelled. The study also considers the computational efficiency when solving such NP-hard problems.

4. The Agent-Based Approach

The applications of agent-based optimisation approach specifically in VRP is not new. Mes et al. (2007) used a competitive agent-based approach for a Dial-A-Ride (DARP) problem by considering an auction mechanism to govern the order-vehicles' communication. Barbucha and Jędrzejowicz (2009) adopted a centralised agent-based architecture that utilises a manager agent which governs parts of order-vehicle interactions for better solutions that utilise intra/inter routes improvements. In later works by the same authors (2012; 2013), a hybrid agent-based approach with meta-heuristics has been developed: Guided Local Search (GLS) and population based optimisation mechanisms. Vokřínek et al. (2010) utilised certain ordering rules and strategies to achieve solutions with a minimum number of vehicles. Rules are further extended in a time-window problem (Kalina and Vokřínek, 2012) by adapting Solomon's insertion (Solomon, 1987). Martin et al. (2016) utilised the agent-based approach to improve the searching process for promising solutions in a vast solution space, by running different metaheuristics agents while cooperating by exchanging best moves. However, there is still a pressing need to develop more sophisticated models including agent-based to deal with the increasing number and changing themes of complex real-life constraints.

The motivation of selecting the ABM approach in this research is due to the fact that various logistics applications have characteristics that can be modelled using this approach; however, the evaluation studies of using it in such application is still immature and limited (Davidsson et al., 2005). In addition, the role of agents' interaction design in optimisation applications are highlighted by Barbati et al. (2012), which can be classified into either competitive or cooperative agents. The cooperative approach is believed to suit this VRP problem settings in terms of selection of customers and allocation of vehicles. It is worth mention-

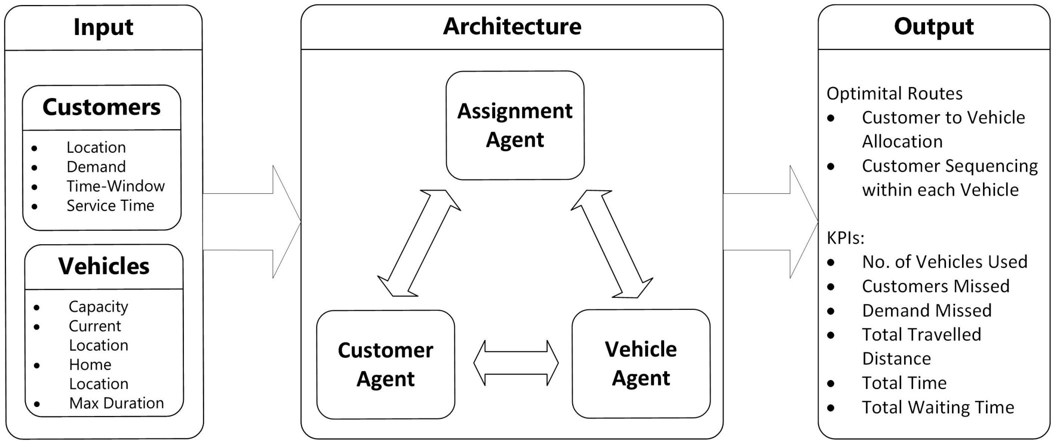

Figure 2: Model Inputs/Outputs and Agent-Based Architecture

ing that the agent-based approach in general is decentralised by nature, so it is able to solve large-sized problems in a reduced computational time. However, the drawback of this approach is that it does not guarantee the optimality of the solutions compared to traditional centralised optimisation approaches (Davidsson et al., 2007); therefore, a hybrid approach is considered. Monostori et al. (2006) highlighted the hybrid cooperative approach and stated its benefits by considering both centralised and distributed interactions by defining certain responsibilities for a manager agent to guide/select sub-agents in their task execution for a better balance in time and solution quality.

4.1. The Agent-Based Conceptual Model

It is vital to define the given inputs and the preferred outcomes of the agent-based approach in order to define the scope of the developed model and the agent-based architecture including assignment, customer and vehicle agents. Figure 2 shows all the given inputs categorised in terms of customers and vehicles, outputs which are the generated routes along with a number of Key Performance Indicators (KPI). The data given for each customer consist of location, quantity demanded, time-window at which customer is available and service time required per visit. Vehicles, on the other hand, have given resource attributes which are capacity, the location where it starts its route, home location or depot where it should end it and the maximum duration allowed for it to operate.

In this architecture, its core consists of the assignment agent, customer agent and vehicle agent. The assignment agent is designed to control certain priority rules that organise the communication amongst agents, customers and vehicles. The customer agent mainly initiates requests then evaluates their responses from the vehicle agents and selects a vehicle. Finally, the vehicle agent performs certain optimisation tasks by evaluating customer requests (section 4.3). Tasks and communication for all agent types are described in detail under subsection 4.2.

The main outcome of the developed agent-based model is feasible routes of vehicles that determine the allocation of customers to vehicles and their sequence. In order to test the overall performance of the routes, certain KPIs are selected which are as follows:

- The total number of vehicles used
- Total customers missed (coverage)
- The total demand missed which indicates the total demanded quantity from every customer missed by the utilised capacity from all vehicles. The model only considers meeting customer demand in full otherwise all its demand is considered missed.
- Total travelled distance which is the summation of all the distances travelled by each vehicle following their travelled routes.
- Total time which is the summation of all allocated time for every vehicle if they followed their routes.
- Total waiting time is the summation of all the waiting times of every vehicle given if in their resulted routes, a vehicle arrives at a location before the opening time-window of the customer begins.

4.2. The Messaging Protocol-Based Heuristics Optimisation (MPHO) Model

Adopting the hybrid cooperative approach as an agent-based interaction base requires a degree of both distribution and centralisation. Such hybrid cooperative approaches provide a level of tracking of global objectives through certain negotiation protocols with mediator or assignment agent while still maintaining individual agent autonomy. A good example of such cooperation protocols was presented by Mes et al. (2007) and Martin et al. (2016), which first modelled the agents as requester (customer) agent and resource (vehicle) agent coordinated according an auction approach while the latest modelled the agents as metaheuristics, which ran with different parameters while exchanging the best moves.

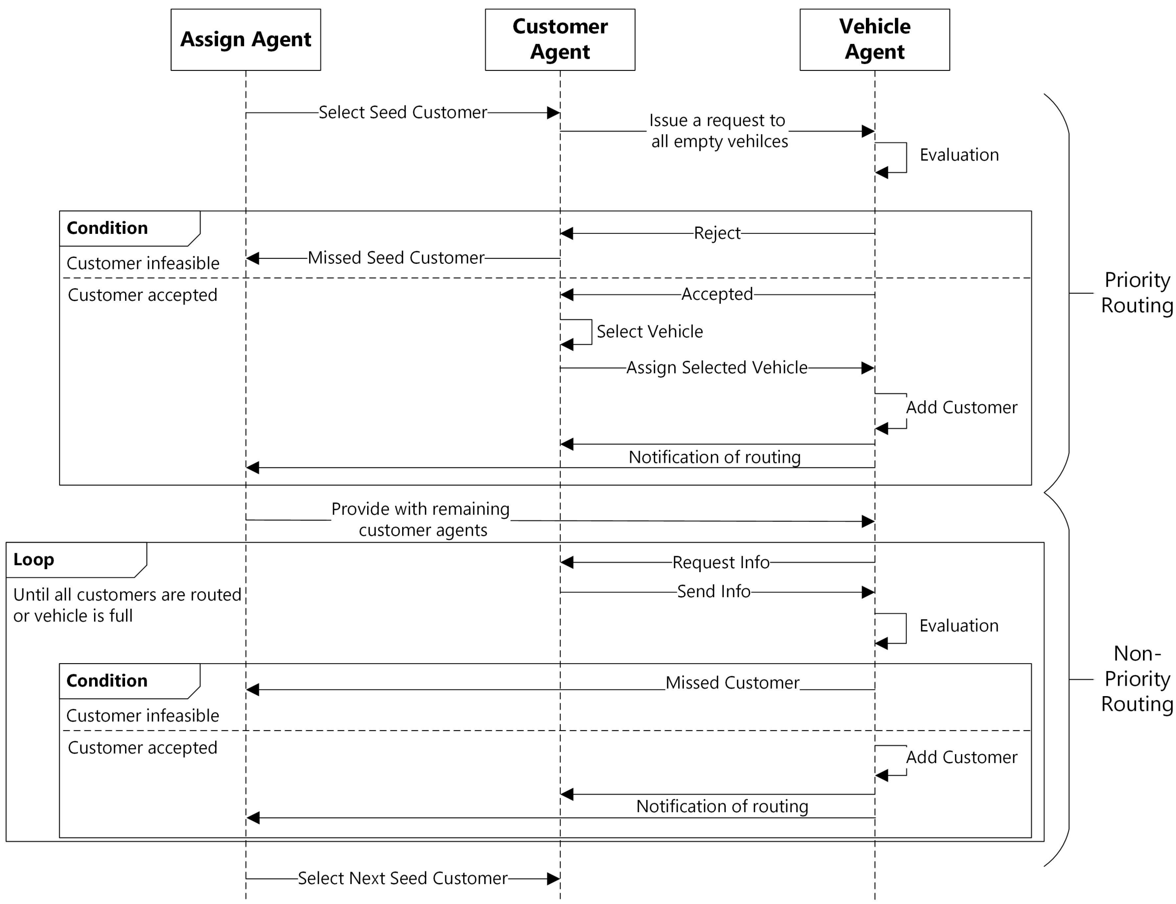

Figure 3: Messaging Protocol-Based Heuristics Optimisation (MPHO) Model

This paper focuses on the first example approach. The representation of the proposed cooperative messaging protocol follows the standard Agent Communication Language (ACL) (FIPA, 2000).

The proposed Messaging Protocol-based Heuristics Optimisation (MPHO) model aims to solve VRPTW problems and is inspired from Solomon's Time-Windows Push Forward feasibility as well as the insertion method (Solomon, 1987). However, we have further improved Solomon's in the form of agents messaging-based optimisation with the possibility to accommodate more priority rules that are not limited to only prioritising customers based on their distance or earliest deadline, especially in the case of scattered vehicles as there is no central depot for customer distances to be compared with, rather with all vehicles as a whole. In the proposed agent-based messaging approach, optimisation would be possible for different and various attributes of each agent, especially vehicles.

The MPHO is presented in Figure 3, the protocol starts with the **Priority Routing phase** where the assignment agent initiates the routes construction process by firstly prioritising customers by one of the rules proposed for prioritising customers shown in Table 1 then selects the top customer agent to be a seed customer to be firstly considered. The method of selecting a seed customer is adopted from Solomon; however, various different rules are applied in order to tailor it to the situation where vehicles have different locations as well as different home locations. A customer noted as a seed means that it has been given the priority to initiate a vehicle route; therefore, at this stage of routing, the seed is dubbed as "Priority Routing". The seed customer agent issues a request to all non-utilised vehicle agents where they all perform feasibility evaluation (more on how vehicle agent evaluates a customer agent request in subsection 4.3), in which it mainly checks if the customer agent will not violate any vehicle constraints, such as capacity, time or duration constraints. Then every vehicle agents return their responses to the seed customer agent and the latter selects a vehicle based on a specific priority rule, shown in Table 1 for prioritising vehicles. If there is at least one vehicle it is feasible; otherwise, it will be marked as missed. If a vehicle agent has been selected by the seed agent then it accommodates it in its route and notifies back to both assignment and customer agents.

However, if all customers are prioritised as seed, then the top prioritised customers will be allocated to all available vehicles on a one-to-one basis. As a consequence, the optimisation process will not achieve a good value for minimising the number of vehicles. Therefore, the **Non-Priority Routing phase** is followed after the vehicle agent has routed a seed customer to further construct its route with other possible non-prioritised customers. This phase starts with the assignment agent providing the vehicle agent with all unserved customers to consider routing them as much as possible without any violations of constraints. The vehicle agent requests individual customer information from all unserved customer agents, then performs evaluation as described in subsection 4.3 and adds customers to route one by one sequentially by choosing customers as per the evaluation described. If a customer is added, then its agent is notified. On the other hand, if the vehicle has been exhausted and no more customers can be considered without constraints' violations, then it refers to the assignment agent by returning the remaining unserved customers, then the assignment agent will repeatedly prioritise the unserved customers and select a seed to repeat the cycle again.

With respect to priority rules, which govern the selection of either a seed customer by the assigned agent or selecting a vehicle by the seed customer agent, the rules are listed in Table 1. In Table 1, customer selection is based on three cases: earliest deadline, furthest minimum distance and furthest average distance.

Table 1
Priority Rules

Rule	Variations
Customer Priority	Earliest Deadline Farthest Min Distance Farthest Avg Distance
Vehicle Priority	Nearest Vehicle

A customer selection based on its deadline of visit means that customers will be prioritised based on their late time-window, for instance, if a customer i has a time-window of $(e_i,\ l_i)$ then a priority customer u is selected as l_u , its late time-window, is minimum. For distance priority, two measures have been implemented since vehicles are not centralised in one depot. One of the measures is to evaluate each customer distances from all vehicles and choose the minimum for each customer, therefore, customer with the highest minimum distance value is prioritised first. Similarly, with the furthest average distance measure; however, instead of selecting the highest minimum, it averages all distances from all vehicles for a particular customer then selects a customer with the highest average. Vehicle selection, on the other hand, is only implemented through a customer agent, selected as seed, since it has the priority to initiate a vehicle route. The seed customer agent only prioritises vehicles that are closest to it.

By describing both phases of the MPHO model, priority and non-priority routing, along with the adopted rules, it can be realised that the developed MPHO model as a whole has improved Solomon's insertion heuristic in the way that it sequentially constructs vehicle routes, therefore, the computational efficiency will tend to behave as insertion heuristics.

4.3. Vehicle Agent Evaluation

When a vehicle agent receives a customer request/information, it tries to evaluate the possibility of accommodating it in its route by firstly checking its constraint when a customer is added then performs assessments on every customer provided then chooses the best assessed, the process is then repeated with the updated list of customers. The vehicle agent will make sure that the customer demand will not exceed its current available capacity and if the customer's time-window falls within its operating shifts as indicated in the following two conditions:

$$q_i + Q_{v\ cur} <= Q_v \tag{1}$$

$$e_v <= e_i \qquad l_i <= l_v \tag{2}$$

where q_i is customer i demanded quantity, $Q_{v\ cur}$ is the current occupied capacity of vehicle v, Q_v is the total capacity of the vehicle while $(e_i,\ l_i)$ and $(e_v,\ l_v)$ are customer i time-window and the vehicle operating shift, respectively. Furthermore, each vehicle will perform further checks on the best position within its route to insert the customer. Solomon's insertion techniques have been proven to be highly effective in constructing routes with time-window problems (Solomon, 1987). Therefore, in this paper, searches of insertion's best position in the vehicle routes have been adapted from Solomon by utilising his Push Forward (PF) technique for time-window feasibility check of later customers in the route as well as the objective functions of distance saving and urgency, where distance saving is the change in distances due to the insertion while the urgency is how much the next customer is delayed due to the same insertion. However, the mentioned PF technique has been tailored to be implemented within our vehicle agent as part of the feasibility evaluation to take into consideration the unique vehicle attributes, especially the different ending from start locations of the routes. The PF is a time concept of how much delay it would take for a vehicle to arrive at the next customers in the route if a particular customer is inserted before. It is calculated iteratively for every customer next in route through equation 3 while making sure that the new arrival time at customer i is within its late time-window in equation 4.

$$PF = b_i^{new} - b_i \tag{3}$$

$$b_i^{new} <= l_i \tag{4}$$

where b_i^{new}, b_i are the new and original arrival time of customer i next in route, respectively. Other customers next in route also depend on the push forward of the previous one, therefore, the implementation of PF was done in a recursive manner within each vehicle agent and is represented in Algorithm 1.

where w_i is the waiting time for customer i. Algorithm 1 takes the previously calculated PF and the next customer in route to its PF while checking any possible time-window

Algorithm 1: Push Forward Recursive

```
Data: customer i, Previous PF_{i-1}
Result: feasible or not
PF_i = max(0, PF_{i-1} - w_i);
if PF_i = 0 then
    stop, feasible;
else
    if b_i + PF_i > l_i then
        stop, violation;
    else
        if next i is Home Location then
            stop, feasible;
        else
            Push Forward Recursive (next customer
              i, PF_i);
        end
    end
end
```

violation in that customer. If the next customer in route happens to be the last visit location (vehicle home depot), then the recursion stops. At the end, such recursive function will indicate if the inserted customer could result in a violation or not.

The previous equations 1 and 2 as well as the implemented PF in Algorithm 1 all perform feasibility checks that ensures no constraints violations, however, they do not evaluate the cost or saving of customer potential insertion in a specific position to compare it with other position within the vehicle route. Therefore, each vehicle agent will assess the inserted customer position through equation 5 and selects the position with the maximum value.

$$\frac{\lambda}{2}(d_{vu} + d_{uh}) - \alpha_1(d_{vu} + d_{uh} - \mu d_{vh}) - \alpha_2(b_{j_u} - b_j) \tag{5}$$

$$\lambda, \mu >= 0 \tag{6}$$

$$\alpha_1, \alpha_2 >= 0 \qquad \alpha_1 + \alpha_2 = 1 \tag{7}$$

where λ, μ, α_1 and α_2 are parameters while d_{vu}, d_{uh} and d_{vh} are distances between vehicle to customer, customer to vehicle's home location and vehicle and its home location, respectively. b_{j_u} is the the new arrival time at customer j when customer u is inserted in position (i, j). The first term in equation 5 has been improved to suit the modelling of the research problem where vehicle's start location is different than its depot. Therefore, the parameter λ was divided in half and the distance is added with the extra distance from customer to the particular vehicle's home location, contrary to what Solomon previously used where the term was equivalent to λd_{vu} as it only considers the distance from the vehicle to customer, based on the assumption that the vehicle location is the same as its home depot. The second term is the distance saving function from Clarke and Wright (1964) weighted along with the cost of pushing forward of customer j in the third term.

One of the drawbacks of Solomon's insertion method is that it assumes that all vehicle should initiate their routes as early as possible. As a consequence, this would lead to unnecessary waiting times especially at the first customer to visit in a vehicle's route (Chiu et al., 2006). This will affect the calculation of the total time of each vehicle if the maximum duration constraint is applied. Solomon designed his benchmark problems without such constraint; however, in multiple depot benchmarks with time-window by Cordeau et al. (2001) the constraint was introduced. As a result, there is a need to eliminate such unnecessary waiting time, in order to check this constraint of maximum duration for every possible insertion. The proposed MPHO model overcomes this issue by calculating the waiting times sequentially for every node in route starting from the vehicle to its node in route which is its home depot. The first step is to calculate the departure time from the vehicle's current location to the first customer in route, shown in equation 8, t_{v_i} where $i = 1$ is travel time needed from the vehicle location to the first customer in route. It takes the maximum between zero and the difference between the first customer early time-window e_i and the travel time for the purpose of eliminating the waiting time for the first customer.

$$dep_v = max(0, e_i - t_{v_i}) \qquad i = 1 \tag{8}$$

Secondly, the arrival time at every customer is found. It is governed by the departure time, service time and travel time from the previous node, either the vehicle's initial location or previous customer in route. It is worth noting that in case the previous node is the vehicle initial location, then its serving time is set to zero. Arrival time calculation is represented in equation 9, where dep_{i-1}, s_{i-1} and t_{i-1i} are the previous node departure time, servicing time and the travel time, respectively. Finally, the waiting time at customer i can now be easily calculated by equation 10 by taking the maximum between 0 and its difference between the arrival time and its early time-window e_i.

$$b_i = dep_{i-1} + s_{i-1} + t_{i-1i} \tag{9}$$

$$w_i = max(0, e_i - b_i) \tag{10}$$

In order to calculate the total waiting time for a route, calculating the waiting time for every customer in a route is necessary. Therefore, Algorithm 2 is developed to calculate the total waiting time W_{v_u} for a vehicle v if customer u is inserted in its route.

Upon the calculation of route total waiting time, it is possible to check for the maximum duration constraint for each vehicle route. The duration is defined by the travel, service and waiting times. When customer u is inserted between a position (i, j), additional travel and servicing time should be added while waiting time has to be recalculated as shown

Algorithm 2: Calculating Total Waiting Time

Data: customer u, position (i, j)
Result: route total waiting time W_{v_u}
Add customer u in the position (i, j);
Calculate departure time dep_v from the vehicle;
while *remaining customers in route* **do**
  Calculate arrival time b_i for customer i;
  Calculate waiting time w_i for customer i;
  Add w_i to the total W_{v_u};
end

in Algorithm 2. The additional travelling time can be calculated in a similar way as the distance saving from Clarke and Wright (1964) but with times instead of distances and is represented in equation 11.

$$\Delta t_{u\ ij} = t_{iu} + t_{uj} - t_{ij} \tag{11}$$

where t_{iu}, t_{uj} and t_{ij} are the travel times between customers (i, u), (u, j) and (i, j), respectively. The time saving equation adds the extra times due to the inserted node while eliminating the previously defined time between (i, j). Accordingly, the new route duration after the insertion will be easily calculated from the equation 12 given the previous route total travel T_v and serving times S_v.

$$dur_{v_u} = (T_v + \Delta t_u) + (S_v + s_u) + W_{v_u} \tag{12}$$

The first term in the equation relates to the new total travel time given the change in it with Δt. The second term updates the change of the servicing time by adding the servicing time s_u of the inserted customer. The last term is the calculated total waiting time from Algorithm 2.

$$dur_{v_u} <= dur_{v\ max} \tag{13}$$

In order to check the new duration dur_{v_u} with the inserted customer u against the duration constraint, it can be simply compared to the maximum limited duration of a particular vehicle v as seen in equation 13.

5. Experimental Analysis

5.1. Implementation and Hardware

In this section, the proposed MPHO model is tested on benchmark problems, VRPTW instances from Solomon (1987) and MDVRPTW instances from Cordeau et al. (2001). These benchmarks, however, are for homogenous vehicles located at depots, as there are no instances where vehicles are scattered and heterogeneous. It will still be beneficial to test and validate the approach on these well-known instances. In order to test the proposed model on scattered vehicles and heterogeneous problems, certain modifications are done to MDVRPTW benchmarks to make it applicable to the problem settings.

The output criteria studied are: the number of vehicles used (V), the Total Travelled Distance (TD), Total Waiting Time (WT), Total Time (TT) and the CPU core time in seconds. With respect to the parametric settings, extensive experiments are done while only the best outputs are reported along with their parametric setting. The range and values of each parameter are: μ is either 1 or 2, α_1 and α_2 are between 0.0 and 1.0 while λ is always set to 1. On the other hand, the rules settings for customers prioritisation were either by latest deadline (LTW) or farthest distance, which is also split in to two rules either average (Far_Avg) or minimum (Far_Min) distance of all vehicles. The experiments are conducted by considering all the combinations of these parameters and rules, except experiments on single depot benchmarks as the distance rule is only considered to be farthest distance (Far) as all vehicles are stationed in the same location/depot. In this section, each table shows the best solutions selected based on prioritising the minimum number of vehicles used objective then the total distance, as minimising the total distance is not the primary focus of this paper contrary to what most of MDVRPTW papers have considered. It is worth mentioning, that the current proposed approach is only for route construction, compared to what extensively studied in previous literature that also used route improvements strategies such as local search or metaheuristics, this may explain the costly distance deviations in the following results.

The agent-based model is flexibility programmed on Python to accommodate the different types of problem, the single depot, multiple depot, scattered vehicles with each different depot and the heterogeneous vehicles. Each problem run is conducted on a single core of an Intel(R) Xeon(R) Broadwell CPUs E5-2683 v4 @ 2.10GHz (32 CPU-cores/node) with the availability of 128GB of RAM, the multiple cores of the CPU are utilised to conduct multiple problem instance with different parametric setting all at once.

5.2. Results on Single Depot Benchmarks

Although the study is focused on multiple depots/scattered vehicles variants, a better judgement about the solution quality can be concluded when testing the method on the well-known Solomon's benchmark 100 customer instances. Solomon designed six sets of benchmark instances, generated based on two criteria: geographical location and length of routing horizon. The geographical data generated could be Clustered (C), uniformly Randomised (R) or hybrid (RC) while the horizon length can be either short (1) or long (2). For example, an instance C202 indicates that it is clustered with long time horizon while the last two digits show the instance number. The output on these benchmark instances are only measured in terms the number of Vehicles (V) and the total Travelled Distance (TD) as these are the only criteria known to be compared with best previous solutions. Best known solutions to Solomon instances are taken from SINTEF (2017) research foundation website. Results are reported in the Appendix A in Table A.1.

Regarding clustered instances, optimal solutions for

Table 2
Optimisation Results on MDVRPTW Instances Compared to Best Known Solutions

	V%		TD%		WT%		TT%	
Instance	Cordeau	Chiu	Cordeau	Chiu	Cordeau	Chiu	Cordeau	Chiu
pr01	-25.00%	0.00%	37.82%	-3.04%	-64.93%	1837.43%	0.85%	7.29%
pr02	-25.00%	0.00%	40.14%	7.02%	-90.79%	289.51%	-7.59%	6.61%
pr03	-18.75%	-13.33%	40.20%	4.79%	-92.21%	89.11%	-17.03%	4.62%
pr04	-15.00%	-5.56%	38.48%	7.42%	-75.87%	139.87%	-3.50%	8.33%
pr05	-4.35%	-4.35%	39.86%	-1.61%	-81.00%	771.74%	-6.45%	3.95%
pr06	-3.57%	-3.57%	37.35%	6.21%	-75.02%	811.22%	-4.12%	9.73%
pr07	-30.00%	-12.50%	46.93%	7.91%	-97.27%	320.79%	-17.56%	6.15%
pr08	-23.53%	-7.14%	54.07%	13.04%	-86.24%	175.06%	-14.33%	12.01%
pr09	-21.74%	-10.00%	55.08%	9.50%	-80.70%	797.79%	-7.59%	12.55%
pr10	-10.34%	-7.14%	46.06%	4.66%	-75.63%	1192.61%	-3.62%	9.78%
Avg	-17.73%	-6.36%	43.60%	5.59%	-81.97%	642.51%	-8.09%	8.10%
pr11	0.00%	0.00%	18.53%	-3.41%	-58.67%	388.48%	7.20%	-0.25%
pr12	0.00%	0.00%	31.60%	11.73%	-30.35%	∞	12.49%	13.50%
pr13	-8.33%	-8.33%	34.80%	6.61%	-54.35%	∞	12.91%	7.14%
pr14	-6.25%	-6.25%	35.51%	11.11%	-74.92%	∞	5.12%	9.19%
pr15	-5.00%	-5.00%	39.99%	11.09%	-68.69%	∞	8.09%	9.06%
pr16	-4.17%	-4.17%	36.83%	10.61%	-63.50%	∞	6.62%	9.60%
pr17	0.00%	0.00%	30.13%	1.54%	42.83%	1214.85%	17.38%	4.10%
pr18	0.00%	0.00%	39.96%	6.80%	-41.34%	1445.00%	15.69%	6.94%
pr19	-11.11%	-11.11%	42.07%	14.91%	-73.23%	∞	7.11%	11.48%
pr20	0.00%	0.00%	33.60%	2.61%	-45.60%	3341.44%	9.54%	6.57%
Avg	-3.49%	-3.49%	34.30%	7.36%	-46.78%	∞	10.22%	7.73%

short horizon instances are produced with an average of 1.2 seconds of CPU time along with 13.6% increase in distance cost while solutions on long horizon instances has seen slight increase of 3.3% on the number of vehicles coupled with 16.5% increase in the total distance, given that the average time required is 11.5 seconds. The parameters $\mu = 1$, $\alpha_1 = 0.9$ and $\alpha_2 = 0.1$ can be noticed as the most frequent in producing best solutions in addition to the late time-window priority rule applied in most of the instances to produce the best solution found by the method.

In the uniformly randomised locations instances, the method resulted, with respect to the short routing horizon instances, in around 18% increase in both the number of vehicles and distance travelled given the average optimisation time is 1.3 seconds. More increased deviations are seen on the long horizon instances where it increased the vehicles by around 23% and distances by 36% with an average 35.5 seconds of optimisation time. Mainly, the far distance rule has resulted in the best solutions with exception of only three long time horizon instances where the late time-window rule resulted in the best results. With respect to the parameters, μ is seen to be mostly set to 1 while α_1 is mostly above 0.7 with exception in four instances.

Similar behaviour to the randomised instance can also be seen in the hybrid or semi-clustered instances. A costly increase in both short and long time horizon instances with around 15% deviations in both criteria for short horizon problems while around 17% and 47% deviation in the vehicles used and distance travelled, respectively, in the long horizon problem. The average optimisation time required for each is 1.1 and 22.7 seconds for short and long horizon problem, respectively. Similarly to the randomised instance, the far distance rules have mainly resulted in most of the best results. μ is mostly seen to be set to one with exception of four instance while α_1 is mainly set above 0.7 with exceptions on five instances.

5.3. Results on Multiple Depot Benchmarks

In this subsection, the approach is tested on MDVRPTW benchmark instances from Cordeau et al. (2001), generated instances with different problem sizes and number of depots, then compared with the best known solutions. Table A.2 shows the MPHO model's results on the 20 benchmark instances. As a general overview of the results it can be seen that the approach was able to produce optimal solutions on every instance and takes 20 seconds at most. As our primary objective is to minimise the number of vehicles, comparison to other known solutions that have considered this objective is made. MDVRPTW was mainly focused on minimising the total travelled distance. However, Chiu et al. (2006) have considered minimising the number of vehicles by considering the minimising of the total waiting time and compared their approach to Cordeau et al. (2001) by adapting their Tabu search with the new objective. Table 2 compares the solutions produced to Chiu's and Cordeau's results in deviation percentages. It is worth mentioning that Chiu et al. managed to reduce waiting time to zero which explain the infinite deviation percentages in some of the instances.

From the comparison table, results in most instances have shown improvement on minimising the number of vehicle. For instances with tight time-windows (pr01-10), 17% reduction was achieved compared to Cordeau's solutions in

Table 3
Modification Scenarios to Cordeau's Instances

Scenario	Change	Vehicles' locations	Depots' Locations	Capacities
1	Locations	Uniform Dist $[-100, 100]^2$	Uniform Dist $[-50, 50]^2$	Not Changed
2	Capacities	Not Changed	Not Changed	$\sim N(Q, 0.1Q)$
3	Both	Uniform Dist $[-100, 100]^2$	Uniform Dist $[-50, 50]^2$	$\sim N(Q, 0.1Q)$

terms of the vehicles used and 6% reduction to the best know solution. However, this has resulted at the expense of the total distance which has increased 43% and 5% compared to Cordeau's and Chiu's solutions, respectively. Waiting times, on the other hand, have shown significant reduction, around 82%, compared to Cordeau's results while worsened with respect to Chiu's. The significant deviation from Chiu is attributed to their ability to achieve near zero total waiting time which inflated the deviation percentages. A better representation of time can be seen in the total time, which shows that it has improved 8% on Cordeau's while resulting in an 8% increase compared to Chiu's.

Similarly for wide time-window instances (pr11-20), the number of vehicles have been also reduced with around 3.5% reduction to both approaches with the expense on the total distance of 34% and 7.4% increase from both Cordeau's and Chiu's, respectively. Similarly to the previous tight time-window instances, significant reduction in waiting times has resulted compared to Cordeau's with around 47% decrease while very significant deviation compared to Chiu's because of their model's ability to achieve zero waiting time in instances 12, 13, 14, 15, 16 and 19. Total time deviations have shown increase compared to both Cordeau's and Chiu's approaches with 10.22% and 7.73% increase, respectively, contrary to the previous tight time-window instances where at least it has improved compared to Cordeau's. In general, the results of the proposed approach have produced better results in terms of the number of vehicles used at the expense of the total distance.

5.4. Results on Modified Multiple Depot Benchmarks

The previous benchmark instances were selected, however, differently than the main research problem settings. They are based on the assumption that vehicles start and end their routes at a single depot, in addition to them having similar capacity. Therefore, slight changes to the benchmark instances are needed to suit the faced problem settings and to make the proposed method applicable in term of solution. Cordeau's 20 benchmark instances are chosen to be modified by randomising each vehicle's location, their depots' location as well as their capacities. Statistical distributions are assumed for each of the randomised attributes. Previously when these benchmarks are introduced, location coordinates are assumed to follow continuous uniform distribution where depots' coordinates should be within the $[-50, 50]^2$ square while customers are within the $[-100, 100]^2$ square (Cordeau et al., 1997). Therefore, location coordinates modification will aim to randomise uniformly each vehicle location within the $[-100, 100]^2$ square and their depots to be within the $[-50, 50]^2$ square. Furthermore, the capacity of each vehicle is also considered to be random as it is assumed to follow a discrete approximation from normal distribution with a mean of the original vehicle capacity Q and standard deviation of 10% from the mean.

In order to apply the location and capacity modifications for a numerical analysis, they have been applied individually first then combined. First, each vehicle location and a new special depot for it have been randomised for every instance. Second, the capacity of each vehicle is randomised. Finally, combining the first and second modifications. Table 3 summarises these modification scenarios.

With such modifications, however, serving all customers is not guaranteed as the availability of the resources (vehicle) is changed due to changes to their capacity and locations, with the latter affecting travelling time. In addition, Cordeau et al. (1997) experimented when generating the problem parameters in order to ensure the feasibility of each instance. This research approach, on the other hand, is more pragmatic when serving the customers as it considers missing customers when the vehicles' constraints are exhausted. As a result, 2 new output parameters are introduced to measure the missing customers (C Missed) as well as their total missed demanded quantities (D Missed).

The results of these three scenarios are compared with the method's output on the original instances from Table A.2 by calculating the deviation on the five output criteria: V, TD, WT, TT and CPU time. All customers have been served by the method on the original instance and comparing with percent deviation will be inflated, therefore, the missed customer and demand in the modified scenarios are reported as it is.

Table 4 presents the results of scenario 1 where vehicles locations are modified as well as their depots. Instances with tight time-window (pr01-10) showed, on average, increase in the solution criteria and very slight decrease of 2.6% in the CPU time compared when applied to the original instances. Number of vehicles, travelled distance, waiting time and total time all have increased around 10%, 17%, 46% and 8.9%, respectively. The waiting time showed reductions in 7 instances; however, two instances have resulted in significant increase, therefore, skewing the average. Only in one instance (pr06), the method could not serve all the customers where it missed 7 with total demand of 84. On the other hand, wide time-window instances showed similar results in terms of increasing the vehicles used, total distance and total time, however, with lesser extent with around 2%, 14% and 7% deviations, respectively. Waiting time, on the contrary,

Table 4
Optimisation Results on Scenario 1

	Parameters				Output						
Inst.	C Rule	μ	α_1	α_2	V%	TD%	WT%	TT%	C Missed	D Missed	CPU(s)
pr01	Far_Min	1	0.6	0.4	0.0%	14.7%	-66.1%	3.5%	0	0	14.7%
pr02	Far_Avg	2	0.9	0.1	11.1%	14.5%	127.6%	12.7%	0	0	-8.7%
pr03	Far_Avg	1	1.0	0.0	23.1%	24.2%	17.9%	15.8%	0	0	-11.4%
pr04	Far_Avg	1	0.8	0.2	11.8%	16.2%	-42.9%	6.7%	0	0	4.2%
pr05	Far_Avg	2	1.0	0.0	9.1%	26.0%	-26.5%	13.0%	0	0	-3.3%
pr06	LTW	2	1.0	0.0	3.7%	23.5%	-54.0%	8.1%	7	84	-6.3%
pr07	LTW	1	0.7	0.3	14.3%	2.1%	589.7%	8.4%	0	0	4.0%
pr08	LTW	1	0.8	0.2	7.7%	7.5%	-6.3%	4.1%	0	0	-3.5%
pr09	Far_Avg	1	1.0	0.0	16.7%	20.6%	-48.9%	8.7%	0	0	-10.7%
pr10	Far_Avg	1	1.0	0.0	7.7%	18.9%	-35.4%	8.0%	0	0	-5.4%
Avg					10.5%	16.8%	45.5%	8.9%	0.7	8.4	-2.6%
pr11	Far_Min	1	0.7	0.3	0.0%	8.2%	-91.5%	2.4%	1	3	65.7%
pr12	Far_Avg	1	0.9	0.1	0.0%	12.8%	-58.6%	3.8%	0	0	19.4%
pr13	Far_Avg	2	0.5	0.5	9.1%	17.8%	-30.5%	9.2%	1	3	-13.0%
pr14	Far_Avg	2	1.0	0.0	0.0%	7.6%	25.4%	4.8%	0	0	-23.9%
pr15	Far_Avg	1	0.9	0.1	0.0%	9.2%	-43.8%	3.3%	0	0	-6.5%
pr16	Far_Avg	1	0.9	0.1	4.3%	16.3%	-15.8%	7.4%	0	0	-1.6%
pr17	LTW	1	0.7	0.3	0.0%	7.3%	30.9%	4.0%	2	32	56.6%
pr18	Far_Avg	1	0.9	0.1	0.0%	14.4%	-2.2%	7.9%	0	0	0.3%
pr19	Far_Avg	2	1.0	0.0	6.3%	22.8%	-6.2%	12.1%	0	0	-10.1%
pr20	Far_Avg	1	0.9	0.1	0.0%	28.0%	0.1%	14.1%	0	0	1.2%
Avg					2.0%	14.4%	-19.2%	6.9%	0.4	3.8	8.8%

Table 5
Optimisation Results on Scenario 2

	Parameters				Output						
Inst.	C Rule	μ	α_1	α_2	V%	TD%	WT%	TT%	C Missed	D Missed	CPU(s)
pr01	LTW	2	0.9	0.1	0.0%	0.0%	0.0%	0.0%	0	0	2.5%
pr02	Far_Min	1	0.9	0.1	0.0%	0.0%	0.0%	0.0%	0	0	2.4%
pr03	Far_Avg	1	0.8	0.2	0.0%	2.3%	-25.9%	0.6%	0	0	-0.6%
pr04	Far_Avg	2	0.6	0.4	0.0%	6.8%	-16.9%	2.9%	0	0	1.8%
pr05	Far_Avg	1	0.9	0.1	0.0%	1.4%	31.1%	2.4%	0	0	-2.9%
pr06	Far_Avg	2	1.0	0.0	-3.7%	-0.9%	-34.6%	-2.7%	0	0	-0.6%
pr07	Far_Avg	2	0.9	0.1	0.0%	-0.1%	0.0%	-0.1%	0	0	-2.6%
pr08	Far_Avg	1	1.0	0.0	0.0%	-4.7%	15.2%	-1.9%	0	0	-0.7%
pr09	Far_Avg	1	0.9	0.1	5.6%	2.3%	-3.6%	1.1%	0	0	-8.1%
pr10	Far_Avg	1	0.8	0.2	-3.8%	3.5%	-54.0%	-1.8%	0	0	0.9%
Avg					-0.2%	1.1%	-8.9%	0.1%	0	0	-0.8%
pr11	Far_Min	1	0.9	0.1	0.0%	1.3%	56.1%	2.4%	0	0	6.1%
pr12	Far_Min	1	0.7	0.3	0.0%	-1.0%	-37.1%	-2.8%	0	0	5.4%
pr13	Far_Avg	1	1.0	0.0	0.0%	-0.1%	-51.0%	-1.6%	0	0	-16.0%
pr14	Far_Min	2	1.0	0.0	-6.7%	0.7%	-54.7%	-1.3%	0	0	-4.7%
pr15	Far_Min	2	1.0	0.0	0.0%	0.3%	73.7%	2.5%	0	0	-2.6%
pr16	Far_Avg	1	0.9	0.1	0.0%	1.4%	10.3%	1.1%	0	0	-3.5%
pr17	Far_Avg	2	0.9	0.1	0.0%	2.4%	50.7%	3.1%	0	0	6.3%
pr18	Far_Avg	1	1.0	0.0	0.0%	1.0%	-14.2%	0.1%	0	0	3.8%
pr19	Far_Min	1	1.0	0.0	0.0%	-2.4%	26.0%	-0.4%	0	0	15.2%
pr20	Far_Avg	1	1.0	0.0	0.0%	0.0%	1.9%	0.1%	0	0	4.8%
Avg					-0.7%	0.4%	6.2%	0.3%	0	0	1.5%

showed a reduction of 19.2%. These instances resulted also in missed customers and demand, on average 0.4 and 3.8, respectively, where three instances (pr11-13-17), could not achieve the maximum satisfaction of all customers. Contrary to tight time-window instance, solution times have increased 8.8%. With respect to the solution parameters, the most frequent rule that resulted in best solution is the average distance priority while the minimum and late time-window where considered in 8 instances. μ is seen to be frequently set to 1 while α_1 is set to be above 0.6 except in two instances.

Table 6
Optimisation Results on Scenario 3

	Parameters				Output						
Inst.	C Rule	μ	α_1	α_2	V%	TD%	WT%	TT%	C Missed	D Missed	CPU(s)
pr01	Far_Min	1	0.6	0.4	0.0%	14.7%	-66.1%	3.5%	0	0	13.3%
pr02	Far_Avg	1	0.9	0.1	11.1%	18.0%	-34.1%	10.6%	0	0	26.3%
pr03	Far_Avg	2	0.6	0.4	23.1%	29.9%	-34.4%	17.8%	0	0	-3.1%
pr04	Far_Avg	1	1.0	0.0	11.8%	9.2%	-60.7%	1.5%	0	0	-4.5%
pr05	Far_Avg	1	1.0	0.0	9.1%	23.6%	-10.0%	12.5%	0	0	-5.4%
pr06	Far_Avg	1	0.9	0.1	3.7%	20.2%	-71.4%	5.5%	5	65	-6.1%
pr07	LTW	1	0.9	0.1	0.0%	2.2%	108.4%	2.8%	0	0	6.9%
pr08	Far_Avg	1	0.8	0.2	15.4%	18.2%	-40.0%	8.5%	0	0	-24.5%
pr09	Far_Avg	1	0.6	0.4	16.7%	21.4%	-59.7%	8.4%	0	0	-9.4%
pr10	LTW	1	0.9	0.1	7.7%	7.3%	-16.4%	2.9%	0	0	-1.7%
Avg					9.8%	16.5%	-28.4%	7.4%	0.5	6.5	-0.8%
pr11	Far_Min	1	0.8	0.2	0.0%	-0.2%	-25.8%	-2.5%	2	22	41.7%
pr12	Far_Avg	2	0.6	0.4	0.0%	11.8%	-74.1%	2.3%	0	0	34.8%
pr13	Far_Avg	2	1.0	0.0	9.1%	27.0%	-72.0%	13.4%	1	12	-26.6%
pr14	Far_Avg	1	1.0	0.0	0.0%	7.6%	16.3%	4.5%	0	0	-21.5%
pr15	Far_Avg	1	1.0	0.0	0.0%	11.9%	-19.2%	5.5%	0	0	-21.2%
pr16	Far_Avg	2	1.0	0.0	4.3%	23.9%	0.2%	11.6%	1	12	-14.5%
pr17	LTW	2	0.9	0.1	0.0%	4.2%	-46.3%	-0.6%	2	32	20.7%
pr18	Far_Avg	1	0.9	0.1	0.0%	14.6%	18.1%	8.6%	0	0	3.2%
pr19	Far_Min	2	0.8	0.2	6.3%	31.3%	-26.5%	16.0%	0	0	27.9%
pr20	Far_Avg	2	0.5	0.5	0.0%	32.5%	-57.0%	13.6%	0	0	-0.4%
Avg					2.0%	16.4%	-28.6%	7.3%	0.6	7.8	4.4%

The results of the second scenario where only capacities are randomised are shown in Table 5. Overall, there have been very slight deviations on the outputs compared to the original instances and all customers have been satisfied. For tight time-window instances, vehicles have been marginally reduced while waiting times have shown fair reduction of 8.9%. Total distance and time criteria have shown a very slight increase while the CPU time has reduced with less than 1%. Similarly, wide time-window instances have shown very slight deviations, however, waiting times have shown a fair increase of around 6%. μ has been mostly set to 1, however, fair share of instances has resulted in best solution where this parameter is set to 2. α_1 has been set to value above 0.6 except for only one instance. It can be also deduced that the distance sorting rule is used in all instances expect for the first one and the average distance priority is the most frequent.

Table 6 represents the results of scenario 3 when both locations and capacities are randomised. Tight time-window instances have seen increase in all criteria except for the waiting times and, similarly to scenario 1, one instance (pr06) has faced missed customers. Vehicles, total distance and time have increased around 10%, 17% and 7%, respectively, while waiting time decreased by 28.4%. The total customers missed and their demands are 5 and 65, respectively. CPU time has marginally reduced with less than 1%. Similar output resulted for wide time-window instances, however, with only slight increase of 2% for the vehicles used, with much distributed missed customer across 4 instances with an average of missed customer and demand of 0.6 and 7.8, respectively, in addition to around 4% increase in the solution time. μ in these instances can be mostly seen to be set to 1, however, when looking at wide time-window instances it was mostly set to 2. α_1 is above 0.6 with exceptions in five instances. Similarly, to the previous scenarios, the distance rule is adopted mostly with the average distance rule being the most commonly used.

6. Conclusion

The VRPTW studied in this paper is different from the most previous variants, given that vehicles could have different attributes in terms of their capacities, locations and home locations. The agent-based approach has been adopted in order to flexibly solve such problems where each entity of either demand or resource; (in this paper: customer or vehicle), can have unique attributes, which is more applicable for practical scenarios. This paper proposed Messaging a Protocol-based Heuristics Optimisation (MPHO) model which adopted a hybrid approach between centralised and distributed agents' interactions for the purpose of constructing routes sequentially. In order to evaluate the performance of MPHO, it has been tested on VRPTW and MDVRPTW benchmark instances as well as modified MDVRPTW where locations and capacites are randomised. In VRPTW instances, the MPHO model can generate routes quickly with up to 1.5 minute of solution time, however, at the expense of increasing the number of vehicle and total distance especially when the instances are randomised (R or RC) with a long scheduling horizon. Comparing MPHO's output to best solutions in MDVRPTW, it has been concluded that the proposed method is very fast in generating routes with minimum number of vehicles at the expense of increasing the total time and distance travelled. Results on modified instances where locations are randomised have mainly increased solution costs in all criteria and some customers were not covered, however, with minimum effect in the solution time. Capacity modified instances have resulted in very minor changes in solution quality. On the other hand, when capacity and locations are both randomised, tight time-window instances have shown a slight reduction in missed customers compared to location-only modified instances while in wide time-window instances they have been increased. With respect to the parametric settings, most of the best solutions were achieved when μ is 1, α_1 is above 0.7 while the priority rule is set to far average distance, only distance in the case of VRPTW instances.

For future research, other vehicle attributes can be considered such as different vehicles' shifts and operating times. In addition, the dynamic problem where updates are required to routes during their execution will be investigated. In terms of the solution approach, the introduction of route improvement and Multiple Objectives Optimisation methods is necessary to improve the solution quality. More tests can be conducted and experimenting with different randomisation parameters as well as designing a systematic parametric testing in order to better conclude the effect of these parameter on the solution quality.

A. Optimisation Results for Benchmark Instances

Results for VRPTW are shown in Table A.1 while results for MDVRPTW are represented in Table A.2.

Table A.1
Optimisation Results on VRPTW Instances

	Parameters				Output				
Inst.	C Rule	μ	α_1	α_2	V	V%	TD	TD%	CPU(s)
C101	LTW	1	1.0	0.0	10	0.0%	853.0	2.9%	0.9
C102	LTW	1	0.9	0.1	10	0.0%	980.6	18.3%	1.0
C103	LTW	1	1.0	0.0	10	0.0%	1061.2	28.2%	1.2
C104	LTW	1	1.0	0.0	10	0.0%	1118.1	35.6%	1.7
C105	LTW	1	0.9	0.1	10	0.0%	860.8	3.8%	0.9
C106	LTW	1	0.9	0.1	10	0.0%	896.7	8.2%	0.9
C107	Far	1	1.0	0.0	10	0.0%	914.6	10.3%	1.1
C108	Far	1	0.9	0.1	10	0.0%	854.8	3.1%	1.4
C109	Far	2	0.6	0.4	10	0.0%	925.3	11.6%	2.0
Avg						0.0%		13.6%	1.2
C201	LTW	1	0.9	0.1	3	0.0%	591.6	0.0%	5.8
C202	LTW	1	0.9	0.1	3	0.0%	713.8	20.7%	9.3
C203	LTW	1	0.9	0.1	3	0.0%	795.5	34.6%	10.8
C204	LTW	1	1.0	0.0	4	33.3%	868.6	47.1%	20.3
C205	Far	1	0.9	0.1	3	0.0%	611.6	3.9%	13.2
C206	Far	2	0.9	0.1	3	0.0%	651.0	10.6%	14.8
C207	Far	1	1.0	0.0	3	0.0%	683.1	16.1%	13.2
C208	Far	2	0.6	0.4	3	0.0%	627.9	6.7%	24.3
Avg						3.3%		16.5%	11.5
R101	Far	1	0.9	0.1	20	5.3%	1847.8	11.9%	0.6
R102	Far	2	0.8	0.2	19	11.8%	1723.0	15.9%	0.9
R103	Far	1	0.9	0.1	15	15.4%	1488.6	15.2%	1.1
R104	Far	1	1.0	0.0	12	33.3%	1275.2	26.6%	1.7
R105	Far	1	0.8	0.2	14	0.0%	1523.9	10.7%	0.8
R106	Far	1	0.4	0.6	13	8.3%	1468.1	17.3%	1.0
R107	Far	1	1.0	0.0	12	20.0%	1339.4	21.3%	1.3
R108	Far	1	1.0	0.0	11	22.2%	1185.1	23.3%	1.8
R109	Far	1	0.5	0.5	13	18.2%	1412.3	18.2%	1.0
R110	Far	2	0.8	0.2	12	20.0%	1270.9	13.6%	1.5
R111	Far	1	0.9	0.1	12	20.0%	1283.4	17.0%	1.3
R112	Far	2	0.6	0.4	11	22.2%	1136.3	15.7%	1.9
Avg						18.0%		17.9%	1.3
R201	Far	2	0.8	0.2	4	0.0%	1656.8	32.3%	6.9
R202	LTW	1	0.9	0.1	4	33.3%	1577.0	32.3%	16.9
R203	Far	2	1.0	0.0	3	0.0%	1313.0	39.7%	22.4
R204	Far	1	0.7	0.3	3	50.0%	1027.9	24.5%	58.1
R205	LTW	1	0.6	0.4	3	0.0%	1470.3	47.9%	15.4
R206	Far	2	0.7	0.3	3	0.0%	1217.8	34.4%	21.4
R207	Far	1	1.0	0.0	3	50.0%	1200.3	34.8%	24.1
R208	Far	1	1.0	0.0	3	50.0%	930.4	28.0%	88.8
R209	Far	2	0.7	0.3	3	0.0%	1310.9	44.2%	22.8
R210	LTW	1	0.8	0.2	3	0.0%	1440.2	53.3%	22.2
R211	Far	2	0.8	0.2	3	50.0%	1080.2	22.0%	62.5
Avg						23.3%		36.1%	35.5
RC101	Far	1	0.8	0.2	16	14.3%	1808.2	6.6%	0.8
RC102	Far	1	0.1	0.9	14	16.7%	1771.4	13.9%	1.0
RC103	Far	1	1.0	0.0	13	18.2%	1549.7	22.8%	1.1
RC104	Far	1	0.0	1.0	11	10.0%	1350.2	18.9%	1.5
RC105	Far	1	0.2	0.8	16	23.1%	1838.3	12.8%	0.9
RC106	Far	1	0.8	0.2	13	18.2%	1522.5	6.9%	0.9
RC107	Far	1	0.8	0.2	12	9.1%	1538.7	25.0%	1.2
RC108	Far	2	0.7	0.3	11	10.0%	1326.6	16.4%	1.4
Avg						14.9%		15.4%	1.1
RC201	Far	2	0.6	0.4	4	0.0%	2132.6	51.6%	5.8
RC202	Far	1	0.9	0.1	4	33.3%	1764.2	29.2%	11.8
RC203	LTW	2	1.0	0.0	4	33.3%	1586.8	51.2%	20.6
RC204	LTW	2	0.7	0.3	3	0.0%	1206.3	51.1%	67.0
RC205	Far	1	1.0	0.0	5	25.0%	1908.8	47.1%	8.1
RC206	Far	1	0.9	0.1	4	33.3%	1734.8	51.3%	9.2
RC207	Far	1	1.0	0.0	4	33.3%	1497.2	41.1%	16.4
RC208	Far	1	1.0	0.0	3	0.0%	1244.5	50.3%	42.9
Avg						17.2%		46.6%	22.7

Table A.2
Optimisation Results on MDVRPTW Instances

	Parameters				Output				
Inst.	C Rule	μ	α_1	α_2	V	TD	WT	TT	CPU(s)
pr01	LTW	2	0.9	0.1	6	1494.0	211.2	2258.2	0.3
pr02	Far_Min	1	0.9	0.1	9	2470.8	104.0	3860.8	1.3
pr03	Far_Avg	1	0.8	0.2	13	3376.5	174.6	5354.1	2.9
pr04	Far_Avg	2	1.0	0.0	17	4096.6	443.0	7037.7	5.4
pr05	Far_Avg	1	0.9	0.1	22	4383.2	421.1	7927.3	7.7
pr06	Far_Avg	2	1.0	0.0	27	5362.3	626.0	9821.3	11.7
pr07	Far_Avg	2	0.9	0.1	7	2091.4	37.5	3136.9	0.7
pr08	Far_Avg	1	1.0	0.0	13	3312.8	333.7	5546.5	3.0
pr09	Far_Avg	2	1.0	0.0	18	4394.7	518.9	7489.6	7.0
pr10	Far_Avg	2	1.0	0.0	26	5429.5	673.5	9800.0	10.4
pr11	Far_Min	2	0.7	0.3	4	1222.7	48.4	1824.0	0.5
pr12	Far_Min	1	0.7	0.3	8	1974.6	204.8	3465.4	2.3
pr13	Far_Min	2	0.9	0.1	11	2723.8	142.3	4669.1	6.1
pr14	Far_Avg	2	0.9	0.1	15	3045.3	173.9	5717.2	12.1
pr15	Far_Min	2	1.0	0.0	19	3513.5	223.4	6859.9	13.6
pr16	Far_Avg	1	0.9	0.1	23	4028.2	330.9	8192.1	20.2
pr17	Far_Avg	2	0.9	0.1	6	1626.7	89.4	2724.1	1.1
pr18	Far_Avg	1	1.0	0.0	12	2532.4	145.2	4577.6	4.7
pr19	Far_Avg	1	1.0	0.0	16	3283.1	205.5	6064.6	11.7
pr20	Far_Avg	1	1.0	0.0	24	4184.3	406.1	8287.4	17.4

CRediT authorship contribution statement

Anees Abu-Monshar: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization. **Ammar Al-Bazi:** Conceptualization, Writing - Review & Editing, Supervision. **Vasile Palade:** Supervision.

References

Adelzadeh, M., Mahdavi Asl, V., Koosha, M., 2014. A mathematical model and a solving procedure for multi-depot vehicle routing problem with fuzzy time window and heterogeneous vehicle. The International Journal of Advanced Manufacturing Technology 75, 793–802. doi:10.1007/s00170-014-6141-8.

Afshar-Nadjafi, B., Afshar-Nadjafi, A., 2016. Multi-depot time dependent vehicle routing problem with heterogeneous fleet and time windows. International Journal of Operational Research 26, 88. doi:10.1504/IJOR.2016.075651.

Afshar-Nadjafi, B., Afshar-Nadjafi, A., 2017. A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. Journal of King Saud University - Engineering Sciences 29, 29–34. doi:10.1016/j.jksues.2014.04.007.

Alcaraz, J.J., Caballero-Arnaldos, L., Vales-Alonso, J., 2019. Rich vehicle routing problem with last-mile outsourcing decisions. Transportation Research Part E: Logistics and Transportation Review 129, 263–286. doi:10.1016/j.tre.2019.08.004.

Babagolzadeh, M., Shrestha, A., Abbasi, B., Zhang, S., Atefi, R., Woodhead, A., 2019. Sustainable Open Vehicle Routing with Release-Time and Time-Window: A Two-Echelon Distribution System. IFAC-PapersOnLine 52, 571–576. doi:10.1016/j.ifacol.2019.11.219.

Barbati, M., Bruno, G., Genovese, A., 2012. Applications of agent-based models for optimization problems: A literature review. Expert Systems with Applications 39, 6020–6028. doi:10.1016/j.eswa.2011.12.015.

Barbucha, D., 2012. Agent-based guided local search. Expert Systems with Applications 39, 12032–12045. doi:10.1016/j.eswa.2012.03.074.

Barbucha, D., 2013. Experimental Study of the Population Parameters Settings in Cooperative Multi-agent System Solving Instances of the VRP, in: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Nguyen, N.T. (Eds.), Transactions on Computational Collective Intelligence IX. Springer Berlin Heidelberg, Berlin, Heidelberg. volume 7770, pp. 1–28. doi:10.1007/978-3-642-36815-8_1.

Barbucha, D., Jędrzejowicz, P., 2009. Agent-Based Approach to the Dynamic Vehicle Routing Problem, in: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Springer, Berlin, Heidelberg. pp. 169–178. doi:10.1007/978-3-642-00487-2_18.

Bettinelli, A., Ceselli, A., Righini, G., 2011. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies 19, 723–740. doi:10.1016/j.trc.2010.07.008.

Brito, J., Martínez, F., Moreno, J., Verdegay, J., 2015. An ACO hybrid metaheuristic for close–open vehicle routing problems with time windows and fuzzy constraints. Applied Soft Computing 32, 154–163. doi:10.1016/j.asoc.2015.03.026.

Chiu, H.N., Lee, Y.S., Chang, J.H., 2006. Two approaches to solving the multi-depot vehicle routing problem with time windows in a time-based logistics environment. Production Planning & Control 17, 480–493. doi:10.1080/09537280600765292.

Clarke, G., Wright, J., 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research 12, 568–581.

Cordeau, J.F., Gendreau, M., Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30, 105–119. doi:10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G.

Cordeau, J.F., Laporte, G., Mercier, A., 2001. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society 52, 928–936.

Dantzig, G.B., Ramser, J.H., 1959. The Truck Dispatching Problem. Management Science 6, 80–91. doi:10.1287/mnsc.6.1.80.

Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., Wernstedt, F., 2005. An analysis of agent-based approaches to transport logistics. Transportation Research Part C: Emerging Technologies 13, 255–271. doi:10.1016/j.trc.2005.07.002.

Davidsson, P., Persson, J.A., Holmgren, J., 2007. On the Integration of Agent-Based and Mathematical Optimization Techniques, in: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (Eds.), Agent and Multi-Agent Systems: Technologies and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg. volume 4496, pp. 1–10. doi:10.1007/978-3-540-72830-6_1.

Dayarian, I., Crainic, T.G., Gendreau, M., Rei, W., 2015. A column generation approach for a multi-attribute vehicle routing problem. European Journal of Operational Research 241, 888–906. doi:10.1016/j.ejor.2014.09.015.

FIPA, 2000. Specifications. http://www.fipa.org/specifications/index.html.

Goel, A., 2010. A Column Generation Heuristic for the General Vehicle Routing Problem, in: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Blum, C., Battiti, R. (Eds.), Learning and Intelligent Optimization. Springer Berlin Heidelberg, Berlin, Heidelberg. volume 6073, pp. 1–9. doi:10.1007/978-3-642-13800-3_1.

Goel, A., Gruhn, V., 2008. A General Vehicle Routing Problem. European Journal of Operational Research 191, 650–660. doi:10.1016/j.ejor.2006.12.065.

Kalina, P., Vokřínek, J., 2012. Parallel Solver for Vehicle Routing and Pickup and Delivery Problems with Time Windows Based on Agent Negotiation. 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC) , 1558–1563.

Kramer, R., Cordeau, J.F., Iori, M., 2019. Rich vehicle routing with auxiliary depots and anticipated deliveries: An application to pharmaceutical distribution. Transportation Research Part E: Logistics and Transportation Review 129, 162–174. doi:10.1016/j.tre.2019.07.012.

Laporte, G., 2009. Fifty Years of Vehicle Routing. Transportation Science 43, 408–416. doi:10.1287/trsc.1090.0301.

Le, T.V., Stathopoulos, A., Van Woensel, T., Ukkusuri, S.V., 2019. Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. Transportation Research Part C: Emerging Technologies 103, 83–103. doi:10.1016/j.trc.2019.03.023.

Lenstra, J.K., Kan, A.H.G.R., 1981. Complexity of vehicle routing and scheduling problems. Networks 11, 221–227. doi:10.1002/net.3230110211.

Li, J., Li, Y., Pardalos, P.M., 2016. Multi-depot vehicle routing problem with time windows under shared depot resources. Journal of Combinatorial Optimization 31, 515–532. doi:10.1007/s10878-014-9767-4.

Mancini, S., 2016. A real-life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Matheuristic. Transportation Research Part C: Emerging Technologies 70, 100–112. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X15002314, doi:10.1016/j.trc.2015.06.016.

Martin, S., Ouelhadj, D., Beullens, P., Ozcan, E., Juan, A.A., Burke, E.K., 2016. A multi-agent based cooperative approach to scheduling and routing. European Journal of Operational Research 254, 169–178. doi:10.1016/j.ejor.2016.02.045.

Mes, M., van der Heijden, M., van Harten, A., 2007. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. European Journal of Operational Research 181, 59–75. doi:10.1016/j.ejor.2006.02.051.

Monostori, L., Váncza, J., Kumara, S., 2006. Agent-Based Systems for Manufacturing. CIRP Annals 55, 697–720. doi:10.1016/j.cirp.2006.10.004.

Montoya-Torres, J.R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., Herazo-Padilla, N., 2015. A literature review on the vehicle routing problem with multiple depots. Computers & Industrial Engineering 79, 115–129. doi:10.1016/j.cie.2014.10.029.

Repoussis, P.P., Tarantilis, C.D., Ioannou, G., 2007. The open vehicle routing problem with time windows. Journal of the Operational Research Society 58, 355–367. doi:10.1057/palgrave.jors.2602143.

Savelsbergh, M.W.P., Sol, M., 1995. The General Pickup and Delivery Problem. Transportation Science 29, 17–29. doi:10.1287/trsc.29.1.17.

Schneider, M., 2016. The vehicle-routing problem with time windows and driver-specific times. European Journal of Operational Research 250, 101–119. doi:10.1016/j.ejor.2015.09.015.

Schopka, K., Kopfer, H., 2016. An Adaptive Large Neighborhood Search for the Reverse Open Vehicle Routing Problem with Time Windows, in: Mattfeld, D., Spengler, T., Brinkmann, J., Grunewald, M. (Eds.), Logistics Management. Springer International Publishing, Cham, pp. 243–257. doi:10.1007/978-3-319-20863-3_18.

Shen, L., Tao, F., Wang, S., 2018. Multi-Depot Open Vehicle Routing Problem with Time Windows Based on Carbon Trading. International Journal of Environmental Research and Public Health 15, 2025. doi:10.3390/ijerph15092025.

SINTEF, 2017. Transportation Optimization Portal - TOP. https://www.sintef.no/projectweb/top/. Last Modified: Tue, 05 Dec 2017 22:28:38 GMT.

Solomon, M.M., 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research 35, 254–265.

Vokřínek, J., Komenda, A., Echouček, M., 2010. Agents Towards Vehicle Routing Problems, in: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems. pp. 773–780.

Xu, Y., Jiang, W., 2014. An improved variable neighborhood search algorithm for multi depot heterogeneous vehicle routing problem based on hybrid operators. International Journal of Control and Automation 7, 299–316.

Xu, Y., Wang, L., Yang, Y., 2012. A New Variable Neighborhood Search Algorithm for the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows. Electronic Notes in Discrete Mathematics 39, 289–296. doi:10.1016/j.endm.2012.10.038.

Yang, Y., Xu, Y., Wang, L., 2013. Study on multi depot heterogeneous vehicle routing problem with an improved variable neighborhood search algorithm. Information Technology Journal 12, 7137–7142.

Zarandi, M.H.F., Hemmati, A., Davari, S., 2011. The multi-depot capacitated location-routing problem with fuzzy travel times. Expert Systems with Applications 38, 10075–10084. doi:10.1016/j.eswa.2011.02.006.

Zhen, L., Ma, C., Wang, K., Xiao, L., Zhang, W., 2020. Multi-depot multi-trip vehicle routing problem with time windows and release dates. Transportation Research Part E: Logistics and Transportation Review 135, 101866. doi:10.1016/j.tre.2020.101866.

Anees Abu-Monshar is currently a Ph.D candidate in simulation and optimisation of logistics system at Coventry University, UK. He received his M.Sc. in Engineering Business Management from the same University while his B.Sc. in Industrial Engineering from the German Jordanian University, Amman, Jordan. He was engaged in an industrial experience by modelling a sub-assembly line at Jaguar Land Rover. His research interests include agent-based modelling, logistics, transportation and optimisation.

Ammar Al-Bazi is a Senior Lecturer in Business Information Systems in the Faculty of Engineering, Environment and Computing and an associate member of the Manufacturing and Materials Engineering Research Centre at Coventry University, UK. He holds a Ph.D. in computer simulation and optimisation from Teesside University, UK. His research interests include hybrid simulation modelling, manufacturing and logistics simulation, metaheuristic optimisation algorithms and hybrid intelligent systems.

Vasile Palade is a Professor in Artificial Intelligence and Data Science in the Faculty of Engineering and Computing and an associate member of the faculty research centre for data science at Coventry University. He previously held academic and research positions both in the UK at the University of Oxford and University of Hull, and in Romania at the University of Galati. His research interests include machine learning/computational intelligence, and encompass mainly neuro-fuzzy systems, ensemble of classifiers, and class imbalance learning.