Applied Soft Computing 125 (2022) 109187

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A multi-objective centralised agent-based optimisation approach for vehicle routing problem with unique vehicles

Anees Abu-Monshar [a,*], Ammar Al-Bazi [b]

[a] Institute for Future Transport and Cities, Coventry University, Coventry, UK
[b] Institute for Advanced Manufacturing and Engineering, Coventry University, Coventry, UK

ARTICLE INFO

Article history:
Received 2 November 2021
Received in revised form 22 April 2022
Accepted 13 June 2022
Available online 18 June 2022

Keywords:
Vehicle routing problem
Unique vehicles
Agent-based modelling
Centralised agent cooperation
Metaheuristics
Multi-objective optimisation

ABSTRACT

Motivated by heterogeneous service suppliers in crowd shipping routing problems, vehicles' similarity assumption is questioned in the well-known logistical Vehicle Routing Problems (VRP) by considering different start/end locations, capacities, as well as shifts in the Time Window variant (VRPTW). In order to tackle this problem, a new agent-based metaheuristic architecture is proposed to capture the uniqueness of vehicles by modelling them as agents while governing the search with centralised agent cooperation. This cooperation aims to generate near optimum routes by minimising the number of vehicles used, total travelled distance, and total waiting times. The innovative architecture encapsulates three individual core modules in a flexible metaheuristic implementation. First, the problem is modelled by an agent-based module that includes its components in representing, evaluating, and altering solutions. A second metaheuristic module is then designed and integrated, followed by a multi-objective module introduced to sort solutions generated by the metaheuristic module based on Pareto dominance. Tests on benchmark instances were run, resulting in better waiting times, with an average reduction of 2.21-time units, at the expense of the other objectives. Benchmark instances are modified to tackle the unique vehicle's problem by randomising locations, capacities, and operating shifts and tested to justify the proposed model's applicability.

1. Introduction

Since their introduction by Dantzig and Ramser [1], Vehicle Routing Problems (VRPs) have been evolving into various variants motivated by cases from real-life applications, which resulted in incorporating additional problem attributes such as Multiple Depot (MDVRP), customer Time Window (VRPTW), Dynamic customer demand arrival (DVRP) and other variants [2]. However, the complexity of the problem increases with such additional real-life constraints. As a result, it becomes challenging to find an optimal solution since the fundamental VRP is considered NP-hard [3]. Consequently, solution approaches to these variants became pragmatic, aiming at finding near-optimal solutions in an efficient computational time, and the most commonly used methods are (meta)heuristics [4].

With the trend of crowd shipping, where the supply of shipping services is offered by a large mass of heterogeneous actors, matching supply with demand becomes a challenge that also requires optimum routing of the problem [5]. Therefore, this paper takes into consideration the heterogeneity in the supply of shipping services in a routing problem by challenging the vehicles similarity assumptions in VRPs, not only in the capacity as seen in variants of heterogeneous VRPs but also the start and ending routes' locations as well as different operating shifts. However, such a problem with unique vehicle attributes was not previously investigated. Schopka and Kopfer [6] considered vehicles can start from anywhere but end at one depot, while Goel and Gruhn [7] and Goel [8] considered unique vehicle locations, starting and ending; however, the other attributes are not considered. As a result, a unique way of modelling is required to capture the uniqueness of the vehicles' attributes.

Therefore, this paper aims to propose an innovative agent-based optimisation model to capture the uniqueness of vehicles in this routing problem, along with the customer Time Windows (VRPTW) variant.

The main contribution of this study can be summarised as follows:

- To propose a new agent-based representation of this unique routing problem that adopts the centralised agent cooperation approach.
- To introduce a new customisable metaheuristic framework integrated with the proposed agent-based model, including multi-objective components.

* Corresponding author.
E-mail addresses: monshara@coventry.ac.uk (A. Abu-Monshar), aa8535@coventry.ac.uk (A. Al-Bazi).

https://doi.org/10.1016/j.asoc.2022.109187
1568-4946/

- To provide a generalised way of evaluating and altering VRPTW solutions to make it applicable to apply any appropriate metaheuristic framework.

This innovative agent-based optimisation model will support logistics planners in managing their collection or delivery operations for better cost-effective and resource-utilised solutions.

This paper is structured as follows: recent related work is presented in Section 2. In Section 3, the centralised agent-based optimisation approach is explained along with the proposed model with its problem, metaheuristics, and multiple objectives components. Experimental results on benchmark instances and modified instances are shown in Section 4. Finally, Section 5 concludes the paper and suggests directions for future research.

2. Related work

This work focuses on modelling and solving VRPTW with the consideration of the uniqueness of every vehicle. The Multiple Depot VRPTW (MDVRPTW) and Open VRPTW (OVRPTW) are closely related to this problem in terms of locations, while other related studies may include different capacities. MDVRPTW, first tackled by Cordeau et al. [9], does allow a group of homogeneous vehicles to be grouped having the same start and end location, while OVRPTW, introduced by Repoussis et al. [10], relaxes the end of routes constraints which do not require the vehicle to return to depot and routes, therefore, remain open. The following Sections, 2.1 and 2.2, review the recent literature on these two variants. A thorough review of these two types of variants is conducted in this section. This review provides a clear understanding of these variants, including their settings, objectives, constraints and methodologies.

2.1. Previous work on MDVRPTW

The MDVRPTW problem has been investigated by many researchers, including but not limited to Chiu et al. [11], who adopted a two-phase (route construction and improvement) heuristic approach to minimise vehicle waiting times leading to reduce total vehicles used. Ting and Chen [12] tackled the MDVRPTW problem by proposing a Multiple Ant Colony System (MACS) to construct routes. The routes construction starts by firstly allocating orders to depots and then routing within depots, while routes improvements are implemented within each ant using Simulated Annealing (SA). Weise et al. [13] developed a multi-objective evolutionary algorithm to minimise missed customers and total distance while maximising vehicle utilisation. Bettinelli et al. [14] adopted an exact approach for a mixed fleet problem where any vehicle can associate with any depot and proposed a branch-and-cut-and price algorithm to minimise the costs endured by vehicles' fixed costs and distances. Luo and Chen [15] proposed a multi-stage Shuffled Frog Leaping Algorithm (SFLA) to the problem that firstly clusters customers to depots as centroids, then performs an enhanced neighbourhood local search within/across every cluster and finally selects the elitist individual for another new clustering round unless termination conditions are met. Dayarian et al. [16] proposed a branch-and-price algorithm to solve heterogeneous vehicle problems to minimise fixed and variable routes' costs. Cantu-Funes et al. [17] formulated a mathematical model for multi-depot and periodic variants, where customers can be served and adopted a two-phase heuristic approach to construct and improve routes using local search.

Contrary to the previous MDVRPTW studies, the following softened the customer time window constraint. Maischberger and Cordeau [18] performed a parallel implementation of Cordeau et al. [9] using Tabu Search (TS) and achieved slightly improved results in the total distances. Xu and Jiang [19] proposed a Variable Neighbourhood Search (VNS) for a case with heterogeneous vehicles under relaxed time window and vehicle's working time constraints to minimise travel costs. The same authors extended their study by adopting diversification strategies in accepting solutions per iteration using SA. Wang et al. [20] proposed a multi-objective evolutionary algorithm for a case with soft time windows to minimise travel time, distances, waiting time, and time window violations. Sadati et al. [21] tackled the MDVRPTW problem with relaxed constraints and proposed a Variable Tabu Neighbourhood Search (VTNS) with granular local search and tabu shaking for its intensification and diversification mechanisms, respectively. Firstly, an initial solution is provided by assigning customers to the nearest depot, and then a modified saving heuristic is used to construct routes.

Further studies have incorporated additional problem constraints. Dharmapriya et al. [22] considered a Split Delivery (MDVRPTWSD) problem with additional objectives from a previous study [13] that balanced workload across all vehicles and solved it using a single solution metaheuristic, TS and SA. Adelzadeh et al. [23] proposed an SA algorithm for a problem with fuzzy customer time windows and heterogeneous vehicles that differ in travel times, capacity, and cost. Cornillier et al. [24] considered a routing problem of Multiple products for Petrol Station Replenishment (MPSRPTW), where vehicles are heterogeneous. Heuristic optimisation was used to reduce large numbers of feasible trips if obtained. A mathematical model was formulated to select the required trips if the objective is to maximise the revenues. Afshar-Nadjafi and Afshar-Nadjafi [25,26] adopted the SA approach for a problem with time-dependent travel times determined based on the departure time from the depot and a limited number of heterogeneous fleets at every depot to minimise fixed and variable travel costs. Li et al. [27] adopted a hybrid evolutionary and local search approach for a variant where vehicles could end their routes at any depot to minimise travel costs. Zhen et al. [28] formulated a mathematical model for a variant with possible multiple trips per vehicle solved with a hybrid swarm and evolutionary algorithm to minimise the total time. Zarandi et al. [29] considered a Multiple Depot Location Routing Problem (MDLRPTW) with fuzzy travel times, where each depot location has to be determined along with its routes, by proposing an SA algorithm to optimise the travelled distances.

Pickup and delivery cases were also considered within multi-depot problems. Flisberg et al. [30] proposed a concept of flow demand nodes (locations) to represent the Pickup and Delivery (PDVRP) variant with multiple depots extending a TS developed by Cordeau et al. [9] with a relaxation of serving all customers' constraints. Sombuntham et al. [31] proposed Particle Swarm Optimisation (PSO) algorithm for a case with both pickup and delivery, where customers may require either collection or delivery services. The aim was to optimise the number of vehicles and the total distance.

More generalised and rich formulations can be seen in Goel and Gruhn [7]. They introduced the General VRP (GVRP), where real-life constraints are combined into one problem, including heterogeneous fleets with different locations (start/end), travel times and operating times, order-vehicle compatibility, and capacity dimensions constraints. A local search with an adaptive neighbourhood structure maximised the total profit. A later study proposed column generation heuristic was used to solve the same GVRP settings [8]. Kramer et al. [32] studied a highly constrained variant, dubbed rich VRP, where they considered a heterogeneous fleet, periodic demand, soft time windows, customer-vehicle compatibility as maximum duration, and customer constraints for each vehicle solved using an Iterated Local Search (ILS) heuristic in order to minimise travel distance and operating costs. Alcaraz et al. [33] also considered a similar rich VRP with an additional constraint of required route breaks for drivers while considering the possibility of outsourcing last-mile demand solved using a proposed heuristic algorithm to minimise the outsourcing costs and travelled distance.

2.2. Previous work on OVRPTW

The OVRPTW problem was previously investigated, including the targets, faced constraints, and the used methodology. The problem has seen variations in how the customer time window constraint was modelled. Norouzi et al. [34] proposed a multi-objective PSO algorithm for an Open VRP that considers serving customers as earlier as possible in their time window, given that the sooner customers are served, the more the sales volume, aiming to maximise the sales while minimising the total distance and achieving vehicles load balance. Brito et al. [35] adopted the Ant Colony Optimisation (ACO) approach for both Closed and Open VRP (COVRPTW), where some vehicles may require to return to a depot with fuzzy customers' time windows and vehicles capacities in order to minimise the total distance. Xia and Fu [36] proposed a TS algorithm for a Soft Time Window and customer Satisfaction Rate variant (OVRPSTWSR), where customer satisfaction is measured based on the degree of its soft time window violation. The aim was to minimise the used vehicles, distance, and time window penalties.

Further studies have incorporated additional problem constraints. Brandão [37] proposed an ILS algorithm for a variant where drivers' shift constrains routes' durations to minimise the number of used vehicles. Hashemi et al. [38] formulated a mathematical model solved using a heuristic algorithm for a multi-trip variant of a real-life case of an OVRPTW. Yu et al. [39] considered an OVRPTW with a two-dimensional capacity loading problem. The original routing problem is solved using a population-based metaheuristic, dubbed Learning Whale Optimisation Algorithm (LWOA), while the best individual in a generation undergoes further loading decisions using a proposed filling algorithm.

Other major modifications to OVRPTW can be seen in Schopka and Kopfer [6]. They introduced the Reverse Open VRP (ROVRPTW), where vehicles can start at any location and end at a central depot. An Adaptive Large Neighbourhood Search (ALNS) was proposed for a Mixed Integer Programming (MIP) formulation of the problem to optimise vehicle costs and travel times. Niu et al. [40] proposed a TS algorithm for a Green Open variant (GOVRPTW) to minimise emission costs and drivers' wages based on the vehicles used and distance travelled. Shen et al. [41] proposed both PSO and TS algorithms for routes construction and improvement, respectively, for a multiple depot variant (MDOVRPTW) to minimise drivers' costs, time window violations, fuel, and emission. Babagolzadeh et al. [42] formulated a MIP model for a variant called Two-Echelon Open (2E-OVRPTW), where stocks are transported to intermediate depots before final customer delivery can be released at a specific release time. The aim was to minimise emission costs, time window violations, and total distance. Rahmani [43] combined variants of a Close–Open VRPTW (COVRPTW) and Multiple Cross-Docking (MCVRPTW). The latter is concerned about vehicle compatibility at (un)loading at specific docks that occurs multiple times during the supply chain. Considering additional delivery and installation times uncertainty, they formulated a mathematical model that was solved using a robust hybrid metaheuristic of Firefly Algorithm (FA) improved using genetic operators to minimise the total costs.

Based on the above review, some studied variants are highly similar to the problem being investigated in this paper in terms of the vehicles' start/end locations. However, each vehicle's uniqueness, including its different operating shifts, was not previously investigated. For instance, Schopka and Kopfer [6] considered every vehicle to start at a unique location, but the end of the routes is the same at a single depot, while Shen et al. [41] considered a multi depot case where the vehicles in a depot are still sharing the same starting location. Goel and Gruhn [7] and Goel [8] captured only the uniqueness in these locations, while in our previous study [44], we considered both unique locations and capacities. However, this paper further extends our previous study and investigates the uniqueness of the vehicles' capacities and locations as well as assigning different shifts to vehicles, a problem that has not been tackled before. This investigation includes modelling each vehicle as an agent to capture individualised attributes of the agents for a more generalised heterogeneous variant.

3. The proposed centralised agent-based metaheuristic optimisation architecture

The concept of applying agent-based modelling in optimisation has been addressed in earlier studies. Barbati et al. [45] discussed the fundamentals of designing agents for optimisation problems in scheduling and routing. They differentiated between two different types of agent-based optimisation approaches: competitive, where agents greedily seek to benefit their objective, and cooperative, where the agents collaborate in optimising their collective (global) objectives. Cooperative approaches can be classified into distributed, where decentralised agents measure objects from their local perspective and act upon them, and centralised, where a super/governing agent performs global measures of objectives and directs the sub-agents to act to improve such globally measured objectives. Monostori et al. [46] further questioned the degrees in the cooperative approaches by further classifying a hybrid approach that divides the optimisation tasks and responsibilities between either being distributed or centralised.

Adopting the agent-based optimisation approach in VRP is not new. Mes et al. [47] considered a Dial-A-Ride Problem (DARP) solved using an auction algorithm to coordinate customer vehicle communications using a competitive agent-based approach. Barbucha [48] provided a centralised approach embedded with a Guided Local Search (GLS), and in a later study [49] used GLS to DVRP problem, and in [50], a population-based evolutionary algorithm was also used for the same problem. Vokřínek et al. [51] considered certain agent sorting rules and strategies aimed to minimise vehicles. Such rules are further extended by adapting Solomon's insertion [52] to incorporate time windows [53]. Martin et al. [54] adopted the agent-based approach to enhance the metaheuristic search procedure for large solution spaces that runs different metaheuristics agents in parallel and exchange the best-found moves.

Davidsson et al. [55] highlighted the benefits of adopting the decentralised approach in optimisation compared to the classical optimisation techniques for large-sized problems in reducing solution time and computational complexity. However, such a decentralised or hybrid way of optimisation does not guarantee optimality or perform the necessary, feasible regions searches.

Therefore, we based our methodology on the centralised cooperative approach, contrary to our previous study that adopted the hybrid cooperative approach [44]. This centralised approach applies a proposed metaheuristic framework compatible with the routing problem under study. This problem is generally modelled and evaluated using the agent-based approach.

The proposed methodology overcomes the localised greedy solutions generated by using the decentralised and hybrid agent-based approaches by centralising the solution search with the help of a metaheuristic algorithm. Such a shift to centralised search is similar to how heuristics evolved to metaheuristics due to their inflexibility in tackling combinatorial optimisation problems such as VRP [56]. However, the main challenge is how

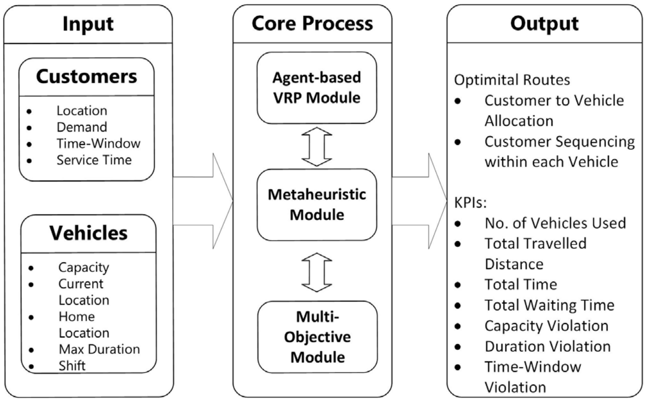

Fig. 1. The proposed agent-based metaheuristic architecture.

to apply a metaheuristic algorithmic framework to the problem under study, given the vehicles' heterogeneity. The issue can be addressed by designing proper solutions Representation, Evaluation and Variation of the problem before applying any metaheuristic framework components.

This study's scope lies in the modelling and optimisation of a VRP, which has a set of inputs for the model to be initiated and generate optimal outputs (routes). The input data is the information about all the customers and vehicles where they can be initialised as agents. This data will then set the initial attributes for these agents. As a result, customer agents will have these attributes: location, demanded quantity, time window, and service time. In contrast, a vehicle agent has the capacity, current location, home location, shift, and maximum duration attributes.

On the other hand, the desired model outputs should be vehicle routes measured against specific objectives: number of vehicles used, total travelled distance, total travel time, total waiting time and any constraints violations found at the end of the search. The relaxed constraints are the late time window for each customer and duration and capacity limits for every vehicle. These inputs and outputs are presented in Fig. 1.

Fig. 1 also shows process core modules that help the search for optimal routes. The first is the agent-based model of the problem consisting of the assignment, customer, and vehicle agents, which represent the solutions. It performs centralised evolutions and variations, the second being the metaheuristic framework that performs a more centralised and global search, and finally, a multi-objective module that aids in sorting the found solutions considering the multiple objectives of the problem.

In metaheuristic design and implementation, it is vital to distinguish between problem-dependent and metaheuristic-specific components to flexibly apply the required metaheuristic components or other frameworks to the problem if necessary Talbi [57]. The problem components consist of the solution representation design, evaluation, and variation, while the metaheuristic components include solution selection criteria, replacement, and algorithm stopping conditions. Furthermore, Talbi [57] also considered the possibility of additional multi-objective components made compatible with such implementation with a specific fitness assignment and solutions preservation techniques.

Fig. 2 illustrates how the architecture core modules communicate. The model starts in the agent-based problem model, where the Representation, Evaluation, and Variation of solutions occur and are sent to the adopted metaheuristic framework to perform the required global search. Selected solutions are then sent to the multi-objective module to be ranked using a proper sorting technique then the ranked solutions are returned accordingly. The metaheuristic module then decides to request more variation of solutions from the problem module or returns the most optimal solution found.

The architecture's core modules, including the agent-based, metaheuristic and multi-objective components, will be further discussed in the following sections. The detailed design of problem-dependent components is explained in 3.1. The metaheuristic used along with its components is explained in 3.2. Furthermore, the multi-objective components in metaheuristic implementation and its detailed implementation are shown in 3.3.

3.1. The agent-based VRP module

The proposed agent-based module for the VRP comprises three types of agents: customer, vehicle, and assignment agents. The customer agent stores its constraints, for example, time window and demand, and provides them when requested. In addition to its constraints, such as capacity and location, the vehicle agent has specific tasks in optimisation by performing feasibility evaluations of its proposed routes, which are sequences of customer agents. On the other hand, the assignment agent is a dummy agent to govern global and centralised interactions between the agents for the optimisation process to diverge away from the local optima with a metaheuristic framework. This module component's main centralised evaluation process is explained later in 3.1.2, where messages are exchanged between the agents and the necessary local evaluations are performed. Fig. 3 illustrates the proposed module. This proposed module can provide the necessary metaheuristic problem-dependent representation, evaluation, and variation components.

3.1.1. Representation

In early VRP formulations, Dantzig and Ramser [1] represented the problem as Integer Linear Programme, where the decision variable is the connection between two nodes, also known as the "edge representation". Later, when the heuristics approach of both sweep [58] and the two-phase [59] heuristics were introduced to solve this NP-Hard problem, the representation has changed to an ordered sequence of customer nodes known as the "path representation" which has been widely used in metaheuristics [60] hence, it was adopted. A solution representation must represent every possible solution and be efficient when altered where a search path must exist between any solution [57]. Therefore, any customers' permutation is considered a solution, as infeasible constraints can be relaxed regardless of its feasibility

Since the agent-based model consists of vehicles, sequences of customers (can be provided to each vehicle agent), and each vehicle has its start and end node; hence a solution representation of such unique nodes is emitted. This will be known strictly and implicitly from the vehicle agent attributes of current/home locations to be considered when evaluating solutions. The resulted agent-based VRP solution set is shown in Fig. 4, where instances of vehicle agents (V1, V2, …, Vn) are provided with sequences of customers. All customers must be sequenced and the occurrence of a customer across all vehicles is only once.

3.1.2. Evaluation

Since the VRP is a general from the Travelling Salesman Problem (TSP), its original goal is to minimise total travelled distance [61]. However, additional objectives have been considered, such as minimising used vehicles [62] and total waiting time [11]. As a result, multiple objectives approaches have been considered [63]. Furthermore, additional constraints have been incorporated, such as the capacity, route duration, and customer time window constraints, requiring additional feasibility evaluation for these constraints. However, this has resulted in a highly

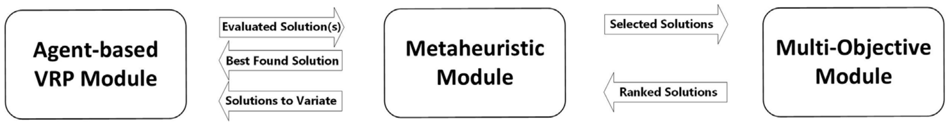

Fig. 2. The core modules workflow.

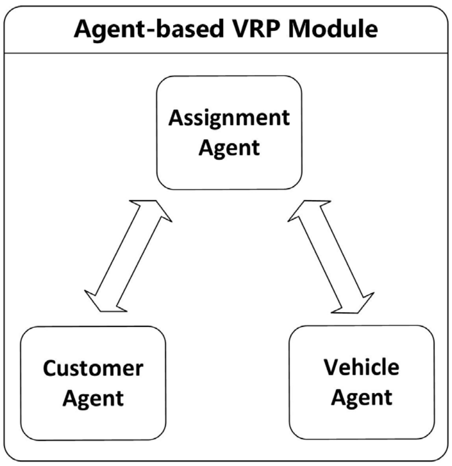

Fig. 3. Proposed agent-based VRP module.

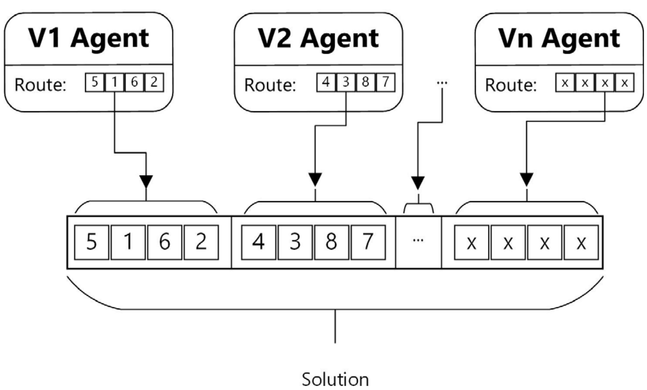

Fig. 4. Solution representation.

constrained problem, and certain constraints have been relaxed to allow the search to explore infeasible regions temporarily [9].

The evaluation process in the proposed agent-based model occurs mainly at the vehicle agent level, where a potential route is provided, and vehicle constraints are checked. Each node information is requested from the respective customer agents, including the time window, demand, service time and location. Accordingly, local route measures such as distance, waiting time, and constraint violations are determined. In contrast, global measures such as the total of these measures and the number of utilised vehicles are determined globally at the assignment agent level from evaluations returned from all vehicle agents for a particular solution. Fig. 5 illustrates the centralised agent messaging for solution evaluation.

In Fig. 5, the evaluation process starts when the assignment agent provides every vehicle agent with its route, part of a solution. Then, the vehicle agent requests the constraints of every customer agent in the route sequence. The required feasibility evaluation will then be performed to be returned to the assignment agent, for which it collects all measures from all the routes of the potential solution and performs a centralised evaluation.

The considered objectives in this evaluation are the total distance, total waiting time, and the number of vehicles used, where all to be minimised, and the respective mathematical representations are:

$$min \sum_{v_r=1}^{V_r} D_{v_r} \tag{1}$$

$$min \sum_{v_r=1}^{V_r} W_{v_r} \tag{2}$$

$$min \sum_{v=1}^{V_r} y_{v_r} \tag{3}$$

where V_r is a set of different routes provided to each vehicle for a particular solution, while the subscript v_r indicates a selected route r provided to vehicle v from the set V_r. D_{v_r} and W_{v_r} are the total distance and waiting time for every route r provided to the vehicle v while y_{v_r} indicate whether vehicle v is idle (0) or utilised (1) if route r is provided. Calculating distance and waiting time measures of a vehicle route is done at the level of the vehicle agent, taking into account the different starting and ending nodes of the route that are unique to the vehicle. The summation of the distances and waiting times are calculated at the assignment agent level. On the other hand, calculating the number of vehicles used can be implicitly determined from the route provided to the vehicle; if empty, then the vehicle is not utilised; otherwise, it is. Distance and waiting time measures are considered zero if an empty route is provided to a vehicle.

One of two strategies can be adopted regarding constraint handling: the Reject or Penalised Strategy. The penalised strategy has been adopted since Cordeau et al. [9] as they solved the MDVRPTW by relaxing the time window, duration, and capacity constraints. However, it allows for the exploration of infeasible solutions, with respect to the relaxed constraints, with a penalty. Gendreau et al. [64] proposed an adaptive penalty strategy to avoid specifying penalties as parameters. Penalties would be initiated randomly and then adapted based on the best previous h iterative solutions evaluated for each constraint violation. If all the previous h solutions violate a constraint, the corresponding penalty for this violation is multiplied by a factor γ; otherwise, divided by it. However, if they were mixed, the penalty value remains. Rochat and Semet [65] randomised the factor γ between 1.5 and 2. Since the relaxed constraints are each vehicle's capacity and duration as well as every customer's late time window, the equations for each violation per potential vehicle route are as follows:

$$VQ_{v_r} = max(0, Q_{v_r} - Q_v) \tag{4}$$

$$Vdur_{v_r} = max(0, dur_{v_r} - dur_v) \tag{5}$$

$$VTW_{v_r} = \sum_{i \in v_r} (max(0, b_i - l_i)) \tag{6}$$

where VQ_{v_r}, $Vdur_{v_r}$ and VTW_{v_r} are violations for capacity, duration and time window constraints, respectively, for a route r provided to vehicle v. Q_{v_r} is the occupied capacity, and dur_{v_r} is the resulted duration, each for route v_r. The subscript i indicates a customer

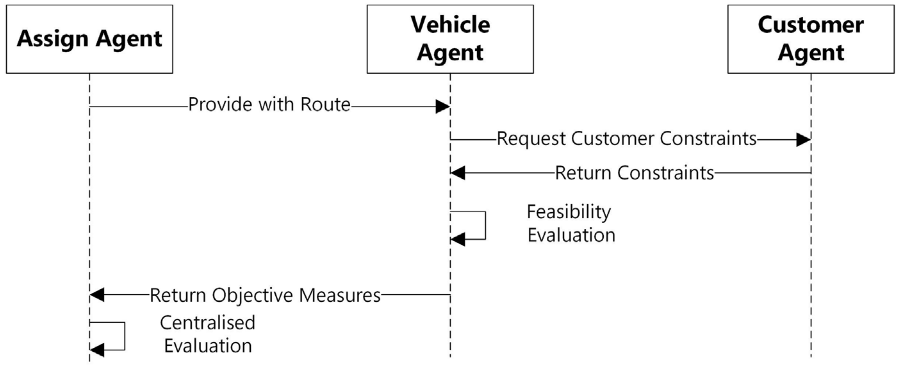

Fig. 5. Centralised evaluation.

in the route v_r where b_i and l_i are its arrival time and late time window.

Since this work takes into consideration vehicle shifts, it is incorporated in the time window violation equation by evaluating a route provided to a vehicle by adding two extra dummy customer nodes at first and the end of the route, indicating their start and end locations, each with a time window of the vehicle shift allowing for shift feasibility evaluation.

Since these measures are localised at the vehicle agent measure, the total violations for each constraint are then calculated by the assignment agent for a particular solution of routes V_r as per the following equations:

$$VQ_{V_r} = \sum_{v_r \in V_r} VQ_{v_r} \quad (7)$$

$$Vdur_{V_r} = \sum_{v_r \in V_r} Vdur_{v_r} \quad (8)$$

$$VTW_{V_r} = \sum_{v_r \in V_r} VTW_{v_r} \quad (9)$$

where VQ_{V_r}, $Vdur_{V_r}$ and VTW_{V_r} are total violations for each capacity, duration, and time windows for a particular solution set V_r. In order to fully apply the penalised strategy, penalties should be multiplied for every total violation to influence the algorithmic evaluation of a particular solution V_r. The following equation represents the total penalised violations for a solution V_r:

$$P_{V_r} = P_Q \ VQ_{V_r} + P_{dur} \ Vdur_{V_r} + P_{TW} \ VTW_{v_r} \quad (10)$$

where P_Q, P_{dur} and P_{TW} are non-negative parameters representing penalties for each capacity, route duration, and time window constraints, respectively. P_{V_r} is the resulted penalty value of the solution V_r. Each penalty is iteratively updated every h iteration and either increased or decreased by multiplying or dividing by a factor γ.

3.1.3. Variation

Variation of VRP solutions in metaheuristics depends if the framework is a single solution-based or population-based metaheuristic. Single solution variations are mainly local heuristics that can be adapted to generate neighbourhoods in high-level heuristics such as TS by modifying one solution with a specific move to generate alternative solutions. These single solution variations can also be classified into two moves [66]: intra-route, variation within one route, and inter-route, variation across multiple routes. This paper adopts the intra-route move of Or-opt [67], where a specific nodes' order is flipped within a route and the inter-route move of CROSS-exchange [68], where specific customer sequences are exchanged between two routes.

On the other hand, population-based variations of the problem are mainly seen in crossover operators in evolutionary algorithms, such as the Genetic Algorithm (GA), where two selected solutions exchange parts of their sequences to generate new offspring, either one or two. However, such a procedure may result in missing or duplicate nodes if the path representation is used. As a result, unique crossover operators have been developed for routing problems such as Partially-Mapped (PMX), Order (OX) and Cycle (CX) crossovers [60]. However, with the introduction of customer time window constraints into VRP, the crossover operators have been adapted to check further solution feasibility and constraint satisfaction. Sequence-Based (SBX) and Route-Based (RBX) have been proposed by Potvin and Bengio [69], where solution repairs are done to ensure feasibility after applying the operator. In later work, Ombuki et al. [70] proposed a crossover operator dubbed Route Cross (RC). In this crossover, randomly selected routes from a parent are removed and the unrouted customers are sorted based on the occurrence in the other parent then iteratively re-inserted in the least cost position. If no feasible position exists, the customers' coverage is relaxed, which allows for missing customers during the search with a penalty. Ombuki et al. [63] further improved RC to Best Cost RC (BCRC), that later adopted by Ghoseiri and Ghannadpour [62] to use for solving multiple objectives problems. The aim is to minimise the vehicles used and total distance, where they limit the randomly selected routes to one in both parents and customer re-insertions are done in random order while not allowing customers to be missed during the search by initiating new routes.

However, this paper adopts the latest developed VRPTW crossover operator BCRC with minor adaptations. The cost calculation for re-insertions considers the penalties of the relaxed constraints, and route initiation is different as every vehicle agent has to be checked due to their heterogeneity, increasing the computational cost.

3.2. The metaheuristic module

This module follows the identification of the problem requirements discussed in Section 3.1. This structure enables an appropriate selection of the metaheuristic components (selection, recombination and stopping condition) for the best problem-solving practice. Metaheuristics are algorithmic frameworks, or "recipes", that provide high-level solution strategies regardless of the problem being solved [71]. Such recipes are mostly favoured when solving VRPs [72], with GA and TS being the most adopted frameworks [73]. Despite that, the choice of using these recipes is up to the researcher(s), the problem on hand dictates a framework to be a population-based metaheuristic to tackle it multi-objectively [74]. Therefore, this work adopts a population-based evolutionary metaheuristic, GA, to solve the problem under study.

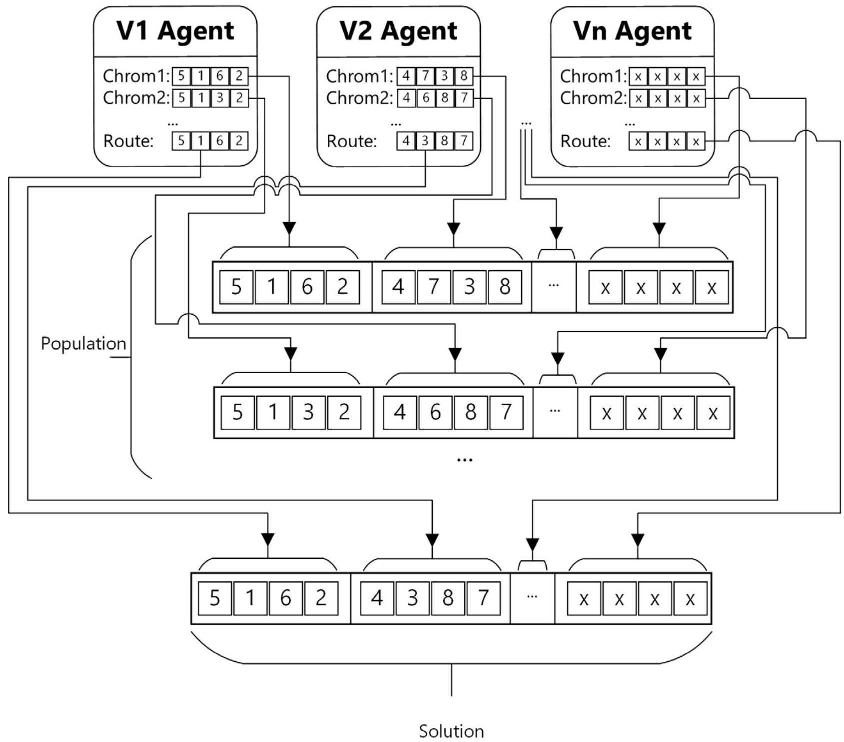

Fig. 6. Population-based solution representation.

GA, first introduced by Holland [75], is a stochastic process that examines pools of solutions (population) by randomly altering selected individuals, skewed to better ones, using crossover and mutation operators to replace the current pool for another exploratory evolution round.

3.2.1. Agent-based adaptation to population representation

Provided that the adopted metaheuristic is a population-based framework, the representation provided in Fig. 4 is not sufficient, given that it only represents one solution. Therefore, a slight adaptation to each vehicle's memory structure is needed by initiating alternative routes, or chromosomes, with proper indexing to create additional individuals. Each individual consists of the same-indexed chromosomes from all vehicle agents, while a population is a group of such individuals. The best-found individual can be stored in fixed memory in the vehicle agent dubbed route. Individuals are selected based on their unique index during selection and recombination, then generate new individuals with a new unique index after applying recombination operations. The population-based representation is illustrated in Fig. 6.

3.2.2. The initialisation, selection, and recombination

To begin the search, GA needs to start with an initial population, and in VRPTW problems, individuals can be created randomly, using a heuristic or a mixture of both. There are two types of route construction heuristics used to generate individuals: greedy or Solomon I1 heuristics, as seen in [62,63], respectively. However, designing such heuristics for constructing routes and sequences of customer agents for each of the vehicle agents will result in decentralised cooperation between the agents, contrary to the scope of this paper, which aims at experimenting with a fully centralised approach. Therefore, a fully randomised initial population was implemented by randomising sequences of customers, and the number of customers is equally distributed across the vehicles for a particular individual solution. The adopted selection method is the Roulette Wheel Selection. Individuals are selected based on their probability determined by their assigned fitness, more on fitness assignment in 3.3.1, resulting in favouring of more fitted individuals.

Regarding recombination, the chosen operators are explained in 3.1.3, BCRC for crossover and Or-opt, with three nodes limit, as mutation. Furthermore, a greedy search heuristic is also implemented to greedily improve specific individuals with a probability of using the CROSS-exchange operator. A similar approach was found in Ghoseiri and Ghannadpour's work [62], where they adopted a hill-climbing improvement for a certain number of individuals and Vidal et al. [76] in their education operator.

3.2.3. The adopted genetic algorithm

The overall framework of the adopted GA is shown in Algorithm 1. The evolutionary process starts by defining the maximum number of generations *Gen*, probabilities of crossover X_{rate}, mutation M_{rate} and local search LS_{rate} and population size P_{Size}. Before the evolutionary search, the initial population *Pop* is first generated, and the population selection size Sel_{size} and the number of crossovers X_{count} are calculated based on the parameters provided. At the start of every generation, a subsequent sub population $Pop_{selected}$ is selected of size Sel_{size} and a subsequent sub population of size two *Parents* using Roulette Wheel Selection, and for several counts X_{count}, crossovers are performed. Offspring are added to the population $Pop_{selected}$. At the end of the crossover operation, a new population *Pop* is set to $Pop_{selected}$. The mutation and local search are applied to every individual with probabilities of M_{rate} and LS_{rate}, respectively. Towards the end of a specific generation *g*, fitness is calculated for every individual *Ind* and ranked accordingly while the best individual is sought and checked with the best found so far. This evolutionary process is repeated until the maximum number of generations is reached.

3.3. The multi-objective module

After addressing the VRP requirements, multiple objectives components (fitness assignment, diversity preservation and

Algorithm 1: Genetic Algorithm

```
Data: Gen, X_rate, M_rate, LS_rate, P_Size
Result: Best found V_r
Generate Pop randomly;
Sel_size = max(int((1 - X_rate) x P_size), 2);
X_count = P_Size - Sel_size;
g = 0;
while g < Gen do
    g := g + 1;
    Pop_selected = Roulette Wheel Selection(Pop, Sel_size);
    Parents = Roulette Wheel Selection(Pop_selected, 2);
    for X_count do
        Apply Crossover on Parents and generate Offspring;
        Add Offspring to Pop_selected;
    end
    Set Pop as Pop_selected;
    for Ind in Pop do
        Apply Mutation on Ind with a probability of M_rate;
        Apply Local Search on Ind with a probability of LS_rate;
    end
    Calculate fitnesses for every Ind in Pop;
    Rank every Ind in Pop based on fitness function;
    Determine the fittest V_r;
end
```

elitism) will be selected and integrated. These additional components are suggested by Talbi [57] when aiming to address a problem for multiple objectives using a metaheuristic. The choice of these three components is discussed in the following subsections.

3.3.1. Fitness assignment

Fitness assignment is when individuals are compared and ranked, in a single value, for a vector of objective functions. Despite being four types of fitness assignment [57], only two have been implemented in VRP: the scalar and dominance-based approaches. The scalar approach, traditionally adopted, adds up all the objectives with defined weights to transform the problem into a single objective function. However, determining such weights as parameters requires previous knowledge of the problem instance from the modeller or decision-maker. Therefore, the dominance-based approach is adopted where the Pareto optimality is sought by sorting solutions based on their vector of objectives. It is apparent in routing problems that the dominance-based approach is preferred to be implemented with population-based algorithms, evolutionary in particular, owing to their ability to deal with a pool of solutions to find their Pareto optimality.

In contrast, the scalar approach is mainly used in single-solution metaheuristics [74]. Therefore, in this paper, the dominance-based Pareto ranking approach is adopted, similar to what has been implemented in [62,63], with modification in evaluating the objective vector to consider the relaxed constraints. The resulted penalties from the violated constraints have to be taken into consideration in each fitness assignment. Therefore, for every objective function, the overall solution penalty, shown in Eq. (10), is added to it, which makes the resulted vector of objective for a particular solution individual V_r:

$$Overall\ \ Objective\ \ Vector = [P_{V_r} + \sum_{v_r=1}^{V_r} D_{v_r}, P_{V_r} + \sum_{v_r=1}^{V_r} W_{v_r}, P_{V_r} + \sum_{v=1}^{V_r} y_{v_r}] \tag{11}$$

Based on the resulted objective vector, dominance sorting of a particular population *Pop* in a generation can be implemented as shown in Algorithm 2. It starts by initialising the first rank $Rank_{current}$ to be 1 and iterates through all the individuals of the population and checks which of them are not dominated by any other individual. Their rank is set to the current rank if they are not dominated. Upon determining this rank of non-dominated individuals, they are then removed from the unsorted population *Pop* and added to the ranked population set Pop_{ranked}. The current rank $Rank_{current}$ is then incremented by 1, and the process is repeated to determine the next non-dominated rank set until no further individuals are left in the unsorted population *Pop*.

Algorithm 2: Pareto Ranking

```
Data: Pop
Result: Pop_ranked
Rank_current = 1;
while Pop ≠ ϕ do
    for Ind in Pop do
        if Ind is non-dominated in Pop then
            Ind Rank = Rank_current;
        end
    end
    for Ind in Pop do
        if Ind Rank = Rank_current then
            Remove Ind from Pop;
            Add Ind to Pop_ranked;
        end
    end
    Rank_current := Rank_current + 1;
end
```

Upon completing the Pareto dominance ranking, fitness calculation for every individual is needed to be based on their rank. Therefore, a complement of the normalised rank for every individual in a population is calculated, based on Eq. (12), to favour the low-ranked individuals; better solutions to this minimisation problem in the selection process given that their fitness will be their probability of selection.

$$Ind_{fitness} = 1 - \frac{(Rank_{Ind} - Rank_{min})}{(Rank_{max} - Rank_{min})} \tag{12}$$

where $Ind_{fitness}$ and $Rank_{Ind}$ are the calculated fitness and rank given to the individual, respectively. While $Rank_{min}$ and $Rank_{max}$ are the minimum and maximum ranks in a given population.

3.3.2. Elitism and diversity

Elitism preserves the best-found solutions across the generations to ensure the quality of the Pareto front at every generation and prevent the deteriorating performance of the algorithm. At the same time, diversity preservation emphasises generating diverse solution sets to prevent biases and stagnation of the search [57].

This work uses an elitism strategy by passing a certain number of good-fit individuals with no alterations to the next generation, as evidenced in Algorithm 1, where only two individuals selected from the population undergo crossover recombination. In contrast, the remaining are passed to the next generation. However, for diversity preservation in the multi-objective VRPTW problem, proper distance preservation in the objective space could not be maintained due to the discrete nature of the number of vehicles objective [63]. As a result, parameterless diversity preservation was neglected in this study.

Table 1
Summary of equations.

Description	Equation	No.
Total distance	$\sum_{v_r=1}^{V_r} D_{v_r}$	(1)
Total waiting time	$\sum_{v_r=1}^{V_r} W_{v_r}$	(2)
Number of vehicles used	$\sum_{v=1}^{V_r} y_{v_r}$	(3)
Capacity violation VQ_{v_r}	$max(0, Q_{v_r} - Q_v)$	(4)
Duration violation $Vdur_{v_r}$	$max(0, dur_{v_r} - dur_v)$	(5)
Time window violations VTW_{v_r}	$\sum_{i \in v_r}(max(0, b_i - l_i))$	(6)
Total capacity violations VQ_{V_r}	$\sum_{v_r \in V_r} VQ_{v_r}$	(7)
Total duration violations $Vdur_{V_r}$	$\sum_{v_r \in V_r} Vdur_{v_r}$	(8)
Total time window violations VTW_{V_r}	$\sum_{v_r \in V_r} VTW_{v_r}$	(9)
Total penalised penalties P_{V_r}	$P_Q\ VQ_{V_r} + P_{dur}\ Vdur_{V_r} + P_{TW}\ VTW_{v_r}$	(10)
Overall objective vector	$[P_{V_r} + \sum_{v_r=1}^{V_r} D_{v_r}, P_{V_r} + \sum_{v_r=1}^{V_r} W_{v_r}, P_{V_r} + \sum_{v=1}^{V_r} y_{v_r}]$	(11)
Individual fitness $Ind_{fitness}$	$1 - \frac{(Rank_{Ind} - Rank_{min})}{(Rank_{max} - Rank_{min})}$	(12)

Table 2
Tuned parameters.

Parameter	Description	Set value
P_{Size}	Population size	100
Gen	Number of generations	150
X_{rate}	Crossover rate	80%
M_{rate}	Mutation rate	20%
LS_{rate}	Greedy local search rate	10%
h	Generations to revisit the penalties	5
γ	Penalties multiply/divide factor	Uniform(1.5, 2.0)
–	Number of parallel experiments	10

3.4. Summary of mathematical equations

Based on the above derivation of mathematical equations, Table 1 provides an overall summary of these equations as per their order.

4. Computational results

In this section, the proposed centralised agent-based metaheuristic optimisation model is validated against the MDVRPTW benchmark instances provided by Cordeau et al. [9]. It is found that these instances are the nearest to the problem under study, as other problem settings would not have direct comparison factors that suit the problem's nature nor provide validated benchmark results for comparison. The proposed approach was first tested on these instances towards initial verification, although their solution approach minimises the travelled distance. In a later study by Chiu et al. [11], they generated multiple criteria from Cordeau et al. [9] work, including total travelled distance, waiting times and the number of vehicles, and the results are compared against them to ensure a fair multiple criteria comparison study. In order to further show the model's ability to solve problems with unique vehicles' attributes, modifications to these instances are proposed. These modifications include randomisations in vehicles' start/end locations, capacities, and ending shifts.

The evolutionary metaheuristic settings are mostly adapted from Ghoseiri and Ghannadpour [62] since similar recombination operators have been adapted; the number of runs, population size, crossover and mutation rates. However, since our search algorithm specifies a probability for an individual to undergo a greedy local search rather than the education parameter from Vidal et al. [76], the greedy search could emerge in better solutions in fewer generations. Therefore, the number of generations is significantly reduced to 150, as seen in Vidal et al. [76]. Additional constraint handling parameters, h and γ, for penalties update, are set to 5 for h given its low sensitivity to the problem, as proven in Gendreau et al. [64], while a random number between 1.5 and 2.0 for γ, as firstly implemented by Rochat and Semet [65]. The experimental setting is repeated 10 times and ran in parallel while the best arbitrary chosen non-dominated solution is reported per instance. The overall parameter settings are summarised in Table 2.

Table 3
Optimisation results on MDVRPTW instances; number of vehicles (V), total travelled distance (TD) and total waiting time (WT).

Inst.	V	TD	WT	CPU (h)
pr01	6	1834.91	0	0.53
pr02	9	2654.15	1	2.41
pr03	13	3534	79	3.42
pr04	17	5276.27	23	4.94
pr05	22	4862.08	268	5.94
pr06	27	6180.17	90	6.63
pr07	7	2601.83	0	0.87
pr08	13	3990.9	92	2.97
pr09	18	5739.3	40	5.96
pr10	26	7187.03	26	6.85
pr11	4	1137.32	0	0.93
pr12	8	1972.7	0	2.26
pr13	11	2566.6	0	4.21
pr14	15	3251.16	0	6.21
pr15	19	3841.19	0	7.00
pr16	23	4395.81	28	8.02
pr17	6	1583.12	5	1.46
pr18	12	2964.17	0	4.20
pr19	16	3683.63	0	6.93
pr20	24	4881.18	14	8.43

The centralised agent-based model has been programmed in Python, where each problem run is conducted on a single core of an Intel(R) Xeon(R) Broadwell CPUs E5-2683 v4 @ 2.10 GHz (32 CPU-cores/node) with the availability of 128 GB of RAM, the multiple cores have been utilised to perform the experiments in parallel. The outputs of solved instances include the three objectives considered in the optimisation, the number of vehicles (V), total travelled distance (TD) and total waiting time (WT). CPU core times are also reported to show the time spent per run. The model can also report the three constraints violations; however, none has been reported under these settings.

4.1. Results on MDVRPTW benchmarks

The approach is tested on 20 MDVRPTW benchmark instances, 10 with tight customer time windows while the remaining are with wider time windows. Table 3 summarises the results for all

Table 4
Optimisation results on MDVRPTW instances compared to best known solutions; percent changes in number of vehicles (V%), total travelled distance (TD%) and total waiting time (WT%).

Instance	V%		TD%		WT(Diff)	
	Cordeau	Chiu	Cordeau	Chiu	Cordeau	Chiu
pr01	−12.50%	16.67%	69.27%	19.09%	−602.10✓	−10.90✓
pr02	−16.67%	11.11%	50.54%	14.96%	−1128.04	−26.04
pr03	−12.50%	−6.67%	46.74%	9.68%	−2162.57	−13.27
pr04	0.00%	11.11%	78.36%	38.35%	−1813.62	−161.92
pr05	−4.35%	−4.35%	55.14%	9.14%	−1948.79	219.41
pr06	0.00%	0.00%	58.30%	22.41%	−2416.14	21.16
pr07	10.00%	37.50%	82.79%	34.25%	−1373.70✓	−8.90✓
pr08	−11.76%	7.14%	85.61%	36.18%	−2332.35	−29.25
pr09	0.00%	15.00%	102.53%	43.00%	−2648.39	−17.99
pr10	−3.45%	0.00%	93.35%	38.53%	−2737.60	−26.30
Avg	−5.12%	8.75%	72.26%	26.56%	−1916.33	−5.40
pr11	0.00%	0.00%	10.26%	−10.15%	−117.00✓	−9.90✓
pr12	0.00%	0.00%	31.47%	11.62%	−294.00✓	0.00✓
pr13	0.00%	0.00%	27.02%	0.46%	−311.80✓	0.00✓
pr14	0.00%	0.00%	44.68%	18.62%	−693.30✓	0.00✓
pr15	0.00%	0.00%	53.05%	21.46%	−713.70✓	0.00✓
pr16	−4.17%	−4.17%	49.32%	20.70%	−878.55	28.15
pr17	0.00%	0.00%	26.64%	−1.18%	−57.28	−1.48
pr18	0.00%	0.00%	63.82%	25.01%	−247.60✓	−9.40✓
pr19	−5.56%	−5.56%	59.40%	28.93%	−767.90✓	0.00✓
pr20	0.00%	0.00%	55.85%	19.70%	−732.21	2.49
Avg	−0.97%	−0.97%	42.15%	13.52%	−481.33	0.99

the instances. The CPU core times are highly dependent on the problem size where it can go up to 6.85 h for tight time window instances (pr01–10) and up to 8.43 h for wider time window instances (pr11–20), which the increased solution space can be explained with the widened time windows. Such high CPU times are due to the extensive greedy local search of a high probable selection of individuals. A selected sample of the resulted maps with their routes is shown in Fig. 7, where a unique colour of arcs represents a route on a map. A sample GA objectives improvement run of instance pr05 is illustrated in Fig. 8, indicating no more improvements were found after around generation 50 for this particular instance run. It is worth mentioning that these evolutionary generation charts update when the best non-dominated solutions are found.

A comparison is needed to compare the model's overall performance in all the instances against the best-known results. It is found that Chiu et al. [11] reported their heuristic, as well as an adapted TS from Cordeau et al. [9], results against the three objectives adopted in this study: vehicles used, total travelled distance and total waiting time. They also reported the total time, but it can be broken down into its original objectives: distance and waiting time by adding them and the servicing times of all customers if served. Since all customers have been served in all cases, reporting the total time is neglected in this paper. Chiu et al. [11] showed better waiting time values, reached zero waiting times in some instances, and reduced the number of vehicles used if they did not remain the same compared to Cordeau et al. [9]. Table 4 compares this work's results to Chiu et al. [11] and Cordeau et al. [9]. Since waiting times can reach zero, showing deviation percentage would not be informative as divisions by zero could occur: therefore, the differences are reports, however, with a ✓symbol indicating if our model's result, for instance, reached zero waiting time.

Based on the comparison table, the proposed approach generally managed to reduce the number of vehicles and total waiting time in some instances, however, at the expense of the total distance objective. Such behaviour is mainly explained by the non-dominance search and sorting of the proposed approach in selecting solutions by not favouring objectives over another, contrary to previous approaches favouring only one objective, distance or waiting time. For instance, with a tight time window, an average reduction of 5.12% in the number of vehicles was achieved only compared to Cordeau's solutions, while it increased by 8.75% for Chiu's. The total distance travelled has significantly increased compared to both solutions, with averages of 72.26% and 26.56%. However, a significant reduction in waiting times is shown compared to Cordeau's with around 1900 time units while, compared to Chiu's, around five-time units reduction were achieved given the total eradication of waiting times in instances pr01 and pr07. A new record of reduced waiting times was achieved in all tight time window instances except pr05 and pr06, which skewed the average.

On the other hand, in wider time window instances, the number of vehicles was reduced by an average of around 1% compared to both previous approaches, while the total distance travelled was increased by 42.15% and 13.52%, respectively. Waiting times have also been decreased significantly compared to Cordeau's solutions with an average of around 481-time units while a very slight average increase of one-time unit compared to Chiu's despite the new record of waiting times in instances pr11, pr17 and pr18 where pr11 and pr18 waiting times have been eradicated. It can be concluded that the proposed approach resulted in better waiting times with an average reduction in all instances of 2.21-time units.

4.2. Results on modified MDVRPTW benchmarks

Although the original 20 MDVRPTW benchmarks are the closest to the problem under study, slight modifications to every vehicle's information are necessary to tackle the problem. Locations in these benchmarks were randomised initially using a uniform distribution, where customer and depot locations should be within $[-100, 100]^2$ and $[-50, 50]^2$ squares, respectively [77]. Since vehicles can be anywhere around the map, it is assumed that their starting locations be within $[-100, 100]^2$ square while their unique depot is within $[-50, 50]^2$ square. The capacity for every vehicle is also randomised following a normal distribution, rounded to the nearest integer, with a 10% deviation of the instance's original capacity Q. Furthermore, vehicle shifts are also considered unique by randomly selecting their ending shift time to be either at the entire shift or reduced by 25%.

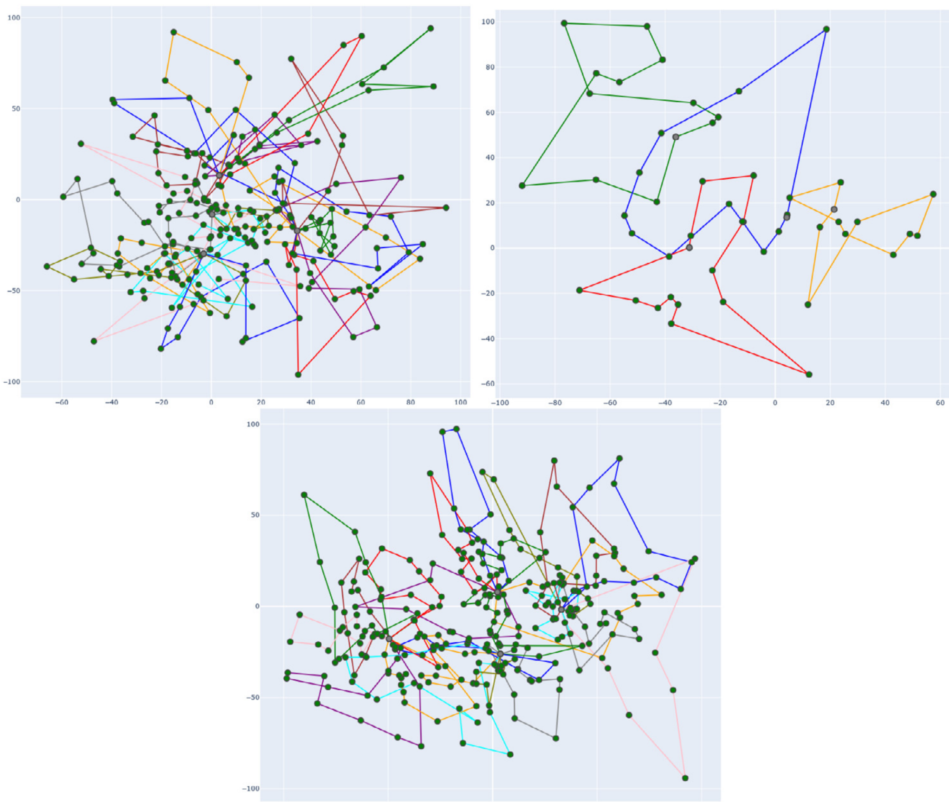

Fig. 7. Sample outputs for pr05 (top left), pr11 (top right) and pr16 (bottom).

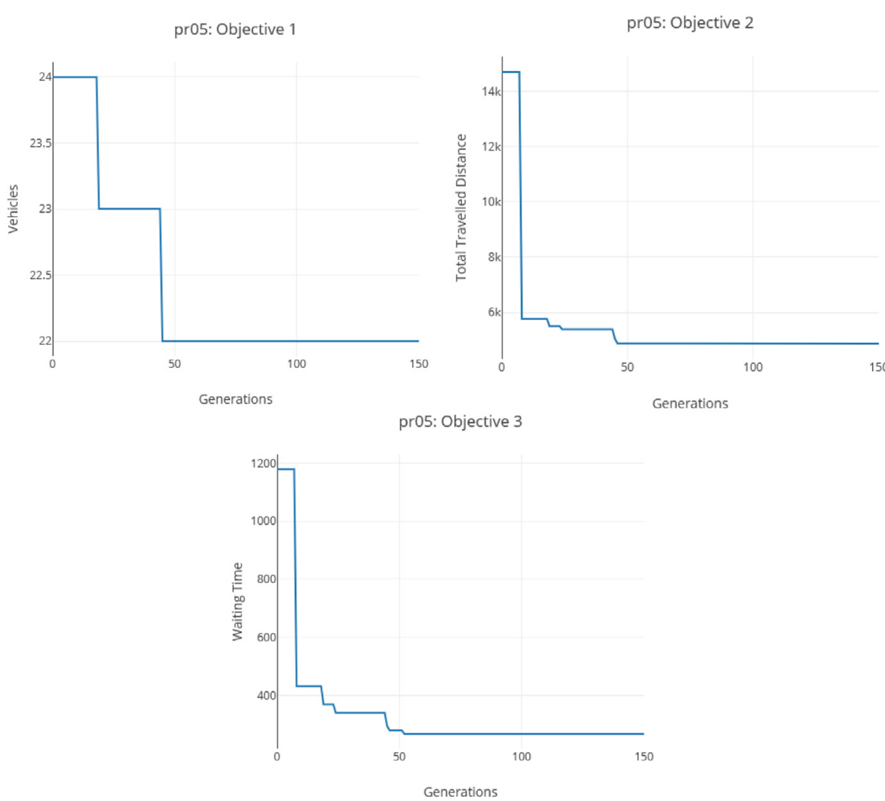

Fig. 8. Pr05 GA run for each objective.

Table 5
Modification scenarios to Cordeau's instances.

Scenario	Change	Vehicles' locations	Depots' location	Capacities	Shifts
1	Location	Uniform dist $[-100, 100]^2$	Uniform dist $[-50, 50]^2$	Not changed	Not changed
2	Capacity	Not changed	Not changed	$\sim N(Q, 0.1Q)$	Not changed
3	Shift	Not changed	Not changed	Not changed	Full or reduced
4	All	Uniform dist $[-100, 100]^2$	Uniform dist $[-50, 50]^2$	$\sim N(Q, 0.1Q)$	Full or reduced

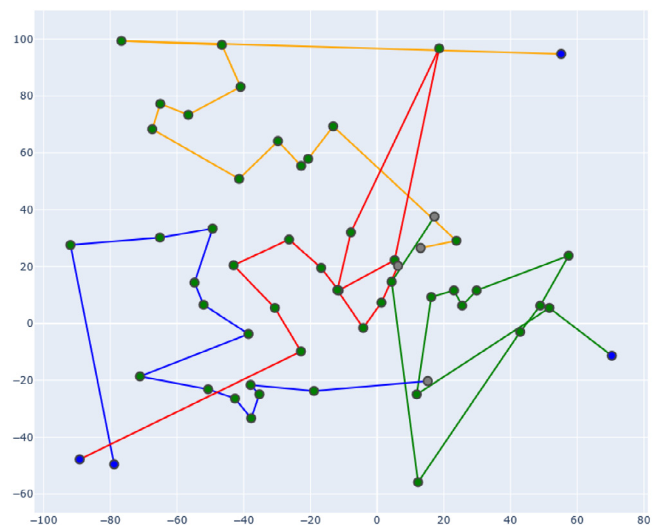

Fig. 9. Sample output for location modified pr11 instance.

Four scenarios are generated based on the way these changes are applied. The first only randomises every vehicle and its home depot locations. Capacities and shifts randomisations are individually applied in the following second and third scenarios, while all modifications are applied in the fourth. The model's results generated in these scenarios are compared to the actual scenario results from Table 3. Table 5 summarises the modifications.

Table 6 summarises the results of scenario 1, where vehicles' locations are modified and their depots'. A sample solution map of the location modified pr11 instance is shown in Fig. 9, where vehicles' starting node is blue, and their home depots are in grey. Generally, it can be seen that there is an average increase in all costs compared to the original scenario:

- Vehicles used increased by around 5% and 3% in tight and wide time window instances.
- Total distance reduced by around 3% in tight time window instances while increased by around 10% in the remaining instances.
- Waiting times have increased by 16.85 and 1.05 of average time units in the respective instances.
- CPU core times have been slightly increased with around 5% and 2% increase.

Results of scenario 2, where every vehicle capacity is randomised, are compared in Table 7 and summarised as follows:

- Vehicles have increased by around 7.84% and 3.16% in tight and wide time window instances.
- Slight changes have resulted in the distance costs, with a reduction of around 2% in tight time window instances and an increase of around 3% in wide time window instances.
- Waiting times have increased in tight time window instances with 44.64 average time units while the other instances reduced by 3.66 average time units.

Table 6
Optimisation results on scenario 1; percent changes in number of vehicles (V%), total travelled distance (TD%) and total waiting time (WT%).

Inst.	V%	TD%	WT(Diff)	CPU(s)
pr01	−16.67%	−14.36%	129.74	6.10%
pr02	0.00%	1.80%	38.87	−11.11%
pr03	7.69%	7.10%	16.56	19.15%
pr04	11.76%	−6.60%	45.82	0.07%
pr05	0.00%	4.90%	−154.27	−15.49%
pr06	3.70%	3.01%	57.50	−8.04%
pr07	14.29%	−8.46%	9.99	15.25%
pr08	7.69%	−1.67%	−5.91	31.11%
pr09	16.67%	−9.14%	−8.14	−0.93%
pr10	7.69%	−6.28%	38.32	14.01%
Avg	5.28%	−2.97%	16.85	5.01%
pr11	0.00%	22.57%	0.00	−0.71%
pr12	0.00%	0.29%	0.00	25.18%
pr13	9.09%	18.62%	2.13	14.30%
pr14	6.67%	20.17%	0.15	−16.27%
pr15	5.26%	9.66%	2.65	−2.63%
pr16	0.00%	8.83%	−0.67	−5.52%
pr17	0.00%	−6.02%	−5.32	23.32%
pr18	0.00%	5.99%	0.00	−1.01%
pr19	6.25%	8.62%	0.00	3.34%
pr20	0.00%	7.83%	11.53	−15.57%
Avg	2.73%	9.65%	1.05	2.44%

Table 7
Optimisation results on scenario 2; percent changes in number of vehicles (V%), total travelled distance (TD%) and total waiting time (WT%).

Inst.	V%	TD%	WT(Diff)	CPU(s)
pr01	0.00%	−2.62%	46.15	0.88%
pr02	0.00%	−2.54%	60.87	−27.51%
pr03	7.69%	7.77%	−30.08	2.01%
pr04	11.76%	−4.63%	88.63	1.87%
pr05	0.00%	6.59%	43.83	10.08%
pr06	0.00%	−6.49%	42.09	−5.83%
pr07	28.57%	7.38%	0.00	9.76%
pr08	15.38%	−4.74%	−62.73	13.35%
pr09	11.11%	−9.93%	138.82	3.07%
pr10	3.85%	−9.45%	118.82	0.55%
Avg	7.84%	−1.87%	44.64	0.82%
pr11	0.00%	−0.62%	0.00	3.68%
pr12	0.00%	−6.48%	0.00	23.24%
pr13	9.09%	16.66%	2.70	3.64%
pr14	6.67%	0.67%	0.00	3.88%
pr15	5.26%	−0.43%	2.60	11.91%
pr16	4.35%	3.83%	−28.15	5.75%
pr17	0.00%	5.60%	−5.32	10.14%
pr18	0.00%	−1.38%	0.00	12.91%
pr19	6.25%	−8.38%	0.00	4.89%
pr20	0.00%	16.66%	−8.45	5.63%
Avg	3.16%	2.61%	−3.66	8.57%

- There is a negligible average increase, less than 1%, in CPU times in the tight time window instances, while a noticeable increase of 8.57% is shown in the remaining instances.

Table 8 summarises the compared results of scenario 3, where every vehicle shift is either at full shift or reduced to a shorter one. As a general insight:

Table 8
Optimisation results on scenario 3; percent changes in number of vehicles (V%), total travelled distance (TD%) and total waiting time (WT%).

Inst.	V%	TD%	WT(Diff)	CPU(s)
pr01	−16.67%	−15.44%	54.23	10.17%
pr02	11.11%	7.02%	14.08	−10.47%
pr03	0.00%	5.52%	−47.96	−0.79%
pr04	5.88%	−22.47%	154.14	−4.98%
pr05	4.55%	14.74%	−130.01	0.88%
pr06	3.70%	4.82%	−29.36	9.28%
pr07	28.57%	−4.79%	61.93	0.72%
pr08	15.38%	−3.49%	−12.90	14.22%
pr09	16.67%	−0.51%	76.18	−2.70%
pr10	7.69%	−10.05%	54.25	5.56%
Avg	7.69%	−2.46%	19.46	2.19%
pr11	0.00%	4.87%	0.00	11.37%
pr12	0.00%	−4.75%	0.00	4.05%
pr13	9.09%	19.01%	0.00	−7.71%
pr14	6.67%	4.10%	0.00	0.42%
pr15	5.26%	−2.97%	0.91	8.28%
pr16	4.35%	−7.60%	−27.20	19.05%
pr17	0.00%	1.54%	−5.32	18.86%
pr18	0.00%	−6.67%	0.00	−7.72%
pr19	6.25%	−3.42%	0.00	−4.60%
pr20	0.00%	6.27%	−9.05	4.48%
Avg	3.16%	1.04%	−4.07	4.65%

Table 9
Optimisation results on scenario 4; percent changes in number of vehicles (V%), total travelled distance (TD%) and total waiting time (WT%).

Inst.	V%	TD%	WT(Diff)	CPU(s)
pr01	−16.67%	−19.19%	81.50	9.77%
pr02	0.00%	−5.41%	28.25	−19.49%
pr03	15.38%	11.16%	−31.13	−0.84%
pr04	5.88%	−13.47%	74.85	−17.20%
pr05	9.09%	17.66%	−61.43	−17.71%
pr06	3.70%	4.55%	175.82	−12.60%
pr07	28.57%	0.96%	34.08	2.73%
pr08	7.69%	−10.79%	−5.53	4.99%
pr09	11.11%	−9.10%	126.94	−7.94%
pr10	3.85%	−11.57%	153.40	−7.78%
Avg	6.86%	−3.52%	57.68	−6.61%
pr11	0.00%	17.14%	1.61	3.63%
pr12	0.00%	1.84%	0.00	12.51%
pr13	9.09%	25.94%	3.00	−11.54%
pr14	6.67%	6.16%	18.83	−22.21%
pr15	5.26%	18.17%	22.97	−19.77%
pr16	4.35%	24.89%	7.93	−17.05%
pr17	0.00%	1.94%	−5.32	8.70%
pr18	0.00%	6.19%	0.00	15.55%
pr19	6.25%	2.92%	0.67	−11.83%
pr20	0.00%	14.92%	11.45	−23.50%
Avg	3.16%	12.01%	6.11	−6.55%

- Similar behaviour to scenario 2 is noticed with more vehicles being utilised and increased by 7.69% and 3.16% in tight and wide time window instances, respectively.
- Minor changes resulted in the total distances, which were reduced by 2.46% in tight time window instances and increased by around 1% in wide time window instances.
- Waiting times have increased in tight time window instances with a 19.46 average time units and reduced by 4.07 average time units in the remaining instances.
- An increase in CPU times with around a 2% and 5% increase in both instances.

Finally, the last scenario's compared results are shown in Table 9, where all previous problem modifications are considered at once. In general, the behaviour is similar to the changes shown in scenarios 2 & 3 and summarised as follows:

- The number of vehicles objective has shown nearly identical behaviour in all previous instances, with an increase of around 7% and 3% in both problem types, respectively.
- Minor changes in the total distance in tight time window instances with around a 4% decrease and a significant increase in distance in wider time window instances with a 12% increase.
- Waiting times have significantly increased in tight time window instances with 57.68 average time units and an increase of 6.11 average time units in waiting times of wider time window instances.
- Noticeable changes can be seen in CPU times which decreased by around 7% in both problem types.

5. Conclusion

This paper studied the VRPTW problem with a different perspective on the heterogeneity of the vehicles that considers different start/end locations, capacities as well as operating shifts. The adopted agent-based modelling approach captured this heterogeneous problem by proposing an architecture that initiates unique instances of vehicles that represented the problem under study. The proposed architecture, including integrating its core modules, successfully emerged appropriate solutions to the unique VRP under study. The proposed architecture managed centralised agent cooperation in solution evaluation and variations. With the aid of a proposed evolutionary metaheuristic framework, the problem is solved by minimising three objectives with the necessary multi-objective components. The model resulted in routes with around 1200 and 2 average time units less in waiting times than Cordeau's and Chiu's solutions of MDVRPTW instances. However, lower waiting times came at the expense of the other objectives, mainly the total travelled distance. It increased by around 57% in Cordeau's and 20% in Chiu's. Adopting of the dominance-based sorting technique in solutions comparison has prevented previous approaches from compromising the waiting times objective by not favouring any of the objectives. MDVRPTW instances are modified to encapsulate the problem under study by randomising locations, capacities and shifts. As a result, four additional scenarios are created: three of which apply each of the changes individually, while the fourth where all are applied. Further experiments were conducted on these scenarios, and the results were compared with the model's output on the original instances and showed costly deviations in all the objectives.

Future research, creating new and generalised benchmarks or systematically modifying existing ones, is needed to capture such practical scenarios by adopting different levels of each of the modifying factors. Further work can include studying additional problem variants, such as pickup and delivery and dynamic VRP, to be compatible with the proposed agent-based problem model. Furthermore, systematic parametric experimentation is needed to find the tuned parameters of the metaheuristic for the problem under study and more experimentation with a combination of the metaheuristic components to suitably find the best algorithm to aid the search for this agent-based model.

CRediT authorship contribution statement

Anees Abu-Monshar: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Ammar Al-Bazi:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G.B. Dantzig, J.H. Ramser, The truck dispatching problem, Manage. Sci. 6 (1) (1959) 80–91, ISBN: 0025-1909.

[2] K. Braekers, K. Ramaekers, I. Van Nieuwenhuyse, The vehicle routing problem: State of the art classification and review, Comput. Ind. Eng. 99 (2016) 300–313.

[3] J.K. Lenstra, A.H.G. Rinnooy Kan, Complexity of vehicle routing and scheduling problems, Networks 11 (2) (1981) 221–227.

[4] G. Laporte, Fifty years of vehicle routing, Transp. Sci. 43 (4) (2009) 408–416, ISBN: 0041-1655.

[5] T.V. Le, A. Stathopoulos, T. Van Woensel, S.V. Ukkusuri, Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence, Transp. Res. C 103 (2019) 83–103.

[6] K. Schopka, H. Kopfer, An adaptive large neighborhood search for the reverse open vehicle routing problem with time windows, in: D. Mattfeld, T. Spengler, J. Brinkmann, M. Grunewald (Eds.), Logistics Management, Springer International Publishing, Cham, 2016, pp. 243–257, Series Title: Lecture Notes in Logistics.

[7] A. Goel, V. Gruhn, A general vehicle routing problem, European J. Oper. Res. 191 (3) (2008) 650–660.

[8] A. Goel, A column generation heuristic for the general vehicle routing problem, in: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, C. Blum, R. Battiti (Eds.), Learning and Intelligent Optimization, vol. 6073, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 1–9.

[9] J.-F. Cordeau, G. Laporte, A. Mercier, A unified tabu search heuristic for vehicle routing problems with time windows, J. Oper. Res. Soc. 52 (8) (2001) 928–936.

[10] P.P. Repoussis, C.D. Tarantilis, G. Ioannou, The open vehicle routing problem with time windows, J. Oper. Res. Soc. 58 (3) (2007) 355–367.

[11] H.N. Chiu, Y.S. Lee, J.H. Chang, Two approaches to solving the multi-depot vehicle routing problem with time windows in a time-based logistics environment, Prod. Plan. Control 17 (5) (2006) 480–493.

[12] C.-J. Ting, C.-H. Chen, Combination of multiple ant colony system and simulated annealing for the multidepot vehicle-routing problem with time windows, Transp. Res. Rec.: J. Transp. Res. Board 2089 (1) (2008) 85–92.

[13] T. Weise, A. Podlich, C. Gorldt, Solving real-world vehicle routing problems with evolutionary algorithms, in: J. Kacprzyk, R. Chiong, S. Dhakal (Eds.), Natural Intelligence for Scheduling, Planning and Packing Problems, Vol. 250, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 29–53.

[14] A. Bettinelli, A. Ceselli, G. Righini, A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows, Transp. Res. C 19 (5) (2011) 723–740.

[15] J. Luo, M.-R. Chen, Improved shuffled frog leaping algorithm and its multi-phase model for multi-depot vehicle routing problem, Expert Syst. Appl. 41 (5) (2014) 2535–2545.

[16] I. Dayarian, T.G. Crainic, M. Gendreau, W. Rei, A column generation approach for a multi-attribute vehicle routing problem, European J. Oper. Res. 241 (3) (2015) 888–906.

[17] R. Cantu-Funes, M.A. Salazar-Aguilar, V. Boyer, Multi-depot periodic vehicle routing problem with due dates and time windows, J. Oper. Res. Soc. 69 (2) (2018) 296–306.

[18] M. Maischberger, J.-F. Cordeau, Solving variants of the vehicle routing problem with a simple parallel iterated tabu search, in: J. Pahl, T. Reiners, S. Voß (Eds.), Network Optimization, Vol. 6701, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 395–400.

[19] Y. Xu, W. Jiang, An improved variable neighborhood search algorithm for multi depot heterogeneous vehicle routing problem based on hybrid operators, Int. J. Control Autom. 7 (3) (2014) 299–316.

[20] J. Wang, T. Weng, Q. Zhang, A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows, IEEE Trans. Cybern. 49 (7) (2019) 2467–2478.

[21] M.E.H. Sadati, B. Çatay, D. Aksen, An efficient variable neighborhood search with tabu shaking for a class of multi-depot vehicle routing problems, Comput. Oper. Res. 133 (2021) 105269.

[22] U.S.S. Dharmapriya, S.B. Siyambalapitiya, A.K. Kulatunga, Artificial intelligence computational techniques to optimize a multi objective oriented distribution operations, in: Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management, Dhaka, Bangladesh, 2010.

[23] M. Adelzadeh, V. Mahdavi Asl, M. Koosha, A mathematical model and a solving procedure for multi-depot vehicle routing problem with fuzzy time window and heterogeneous vehicle, Int. J. Adv. Manuf. Technol. 75 (5–8) (2014) 793–802.

[24] F. Cornillier, F. Boctor, J. Renaud, Heuristics for the multi-depot petrol station replenishment problem with time windows, European J. Oper. Res. 220 (2) (2012) 361–369.

[25] B. Afshar-Nadjafi, A. Afshar-Nadjafi, Multi-depot time dependent vehicle routing problem with heterogeneous fleet and time windows, Int. J. Oper. Res. 26 (1) (2016) 88.

[26] B. Afshar-Nadjafi, A. Afshar-Nadjafi, A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet, J. King Saud Univ., Eng. Sci. 29 (1) (2017) 29–34.

[27] J. Li, Y. Li, P.M. Pardalos, Multi-depot vehicle routing problem with time windows under shared depot resources, J. Combin. Optim. 31 (2) (2016) 515–532.

[28] L. Zhen, C. Ma, K. Wang, L. Xiao, W. Zhang, Multi-depot multi-trip vehicle routing problem with time windows and release dates, Transp. Res. E 135 (2020) 101866.

[29] M.H.F. Zarandi, A. Hemmati, S. Davari, The multi-depot capacitated location-routing problem with fuzzy travel times, Expert Syst. Appl. 38 (8) (2011) 10075–10084.

[30] P. Flisberg, B. Lidén, M. Rönnqvist, A hybrid method based on linear programming and tabu search for routing of logging trucks, Comput. Oper. Res. 36 (4) (2009) 1122–1144.

[31] P. Sombuntham, V. Kachitvichyanukul, S.-I. Ao, H. Katagir, L. Xu, A.H.-S. Chan, Multi-depot vehicle routing problem with pickup and delivery requests, in: AIP Conference Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. 1285, Hong Kong, (China), 2010, pp. 71–85.

[32] R. Kramer, J.-F. Cordeau, M. Iori, Rich vehicle routing with auxiliary depots and anticipated deliveries: An application to pharmaceutical distribution, Transp. Res. E 129 (2019) 162–174.

[33] J.J. Alcaraz, L. Caballero-Arnaldos, J. Vales-Alonso, Rich vehicle routing problem with last-mile outsourcing decisions, Transp. Res. E 129 (2019) 263–286.

[34] N. Norouzi, R. Tavakkoli-Moghaddam, M. Ghazanfari, M. Alinaghian, A. Salamatbakhsh, A new multi-objective competitive open vehicle routing problem solved by particle swarm optimization, Netw. Spat. Econ. 12 (4) (2012) 609–633.

[35] J. Brito, F. Martínez, J. Moreno, J. Verdegay, An ACO hybrid metaheuristic for close–open vehicle routing problems with time windows and fuzzy constraints, Appl. Soft Comput. 32 (2015) 154–163.

[36] Y. Xia, Z. Fu, Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate, Cluster Comput. 22 (S4) (2019) 8725–8733.

[37] J. Brandão, Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows, Comput. Ind. Eng. 120 (2018) 146–159.

[38] S. Hashemi, M. Salari, M. Ranjbar, Multi-trip open vehicle routing problem with time windows: A case study, Int. J. Ind. Eng.: Theory, Appl. Pract. 27 (1) (2020) Number: 1.

[39] N.K. Yu, W. Jiang, R. Hu, B. Qian, L. Wang, Learning whale optimization algorithm for open vehicle routing problem with loading constraints, Discrete Dyn. Nat. Soc. 2021 (2021) e8016356, Publisher: Hindawi.

[40] Y. Niu, Z. Yang, P. Chen, J. Xiao, Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost, J. Cleaner Prod. 171 (2018) 962–971.

[41] L. Shen, F. Tao, S. Wang, Multi-depot open vehicle routing problem with time windows based on carbon trading, Int. J. Environ. Res. Public Health 15 (9) (2018) 2025.

[42] M. Babagolzadeh, A. Shrestha, B. Abbasi, S. Zhang, R. Atefi, A. Woodhead, Sustainable open vehicle routing with release-time and time-window: A two-echelon distribution system, IFAC-PapersOnLine 52 (13) (2019) 571–576.

[43] A. Rahmani, A new closed–open vehicle routing approach in stochastic environments, Int. J. Comput. Math.: Comput. Syst. Theory (2020) 1–17.

[44] A. Abu-Monshar, A. Al-Bazi, V. Palade, An agent-based optimisation approach for vehicle routing problem with unique vehicle location and depot, Expert Syst. Appl. 192 (2022) 116370.

[45] M. Barbati, G. Bruno, A. Genovese, Applications of agent-based models for optimization problems: A literature review, 2012, pp. 6020–6028.

[46] L. Monostori, J. Váncza, S. Kumara, Agent-based systems for manufacturing, 55, 2006, 697–720.

[47] M. Mes, M. van der Heijden, A. van Harten, Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems, European J. Oper. Res. 181 (1) (2007) 59–75.

[48] D. Barbucha, Agent-based guided local search, Expert Syst. Appl. 39 (15) (2012) 12032–12045.

[49] D. Barbucha, Multi-agent approach to the DVRP with GLS improvement procedure, in: I. Czarnowski, R.J. Howlett, L.C. Jain (Eds.), Intelligent Decision Technologies, in: Smart Innovation, Systems and Technologies, Springer, Singapore, 2020, pp. 117–126.

[50] D. Barbucha, Experimental study of the population parameters settings in cooperative multi-agent system solving instances of the VRP, in: N.T. Nguyen (Ed.), Transactions on Computational Collective Intelligence IX, Vol. 7770, Springer Berlin Heidelberg, 2013, pp. 1–28.
[51] J. Vokřínek, A. Komenda, M. Echouek, 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, 2010, pp. 773–780.
[52] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Oper. Res. 35 (2) (1987) 254–265.
[53] P. Kalina, J. Vokřínek, Parallel solver for vehicle routing and pickup and delivery problems with time windows based on agent negotiation, in: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2012, pp. 1558–1563.
[54] S. Martin, D. Ouelhadj, P. Beullens, E. Ozcan, A.A. Juan, E.K. Burke, A multi-agent based cooperative approach to scheduling and routing, European J. Oper. Res. 254 (1) (2016) 169–178.
[55] P. Davidsson, J.A. Persson, J. Holmgren, On the integration of agent-based and mathematical optimization techniques, in: N.T. Nguyen, A. Grzech, R.J. Howlett, L.C. Jain (Eds.), Agent and Multi-Agent Systems: Technologies and Applications, in: Lecture Notes in Computer Science, vol. 4496, Springer Berlin Heidelberg, 2007, pp. 1–10.
[56] F. Glover, Future paths for integer programming and links to artificial intelligence, 13, 1986, 533–549.
[57] E.-G. Talbi, Metaheuristics: From Design to Implementation, John Wiley & Sons, 2009.
[58] B.E. Gillett, L.R. Miller, A heuristic algorithm for the vehicle-dispatch problem, Oper. Res. 22 (2) (1974) 340–349, Pages: 340-349 Publication Title: Operations Research Volume: 22.
[59] N. Christofides, A. Mingozzi, P. Toth, The vehicle routing problem, in: N. Christofides, A. Mingozzi, P. Toth, C. Sandi (Eds.), Combinatorial Optimization, Wiley, Chichester, 1979, pp. 315–338.
[60] Z. Michalewicz, Genetic Algorithms + Data Structures=Evolution Programs, 3rd rev. and extended ed., Springer-Verlag, Berlin ; New York, 1996.
[61] T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures, Omega 34 (3) (2006) 209–219.
[62] K. Ghoseiri, S.F. Ghannadpour, Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, Appl. Soft Comput. 10 (4) (2010) 1096–1107.
[63] B.M. Ombuki, B.J. Ross, F. Hanshar, Multi-objective genetic algorithms for vehicle routing problem with time windows, Appl. Intell. 24 (1) (2006) 17–30.
[64] M. Gendreau, a. Hertz, G. Laporte, A tabu search heuristic for the vehicle routing problem, Manage. Sci. 40 (10) (1994) 1276–1290, arXiv:1405.7020v1 ISBN: 9780387236674.
[65] Y. Rochat, F. Semet, A tabu search approach for delivering pet food and flour in Switzerland, J. Oper. Res. Soc. 45 (11) (1994) 1233–1246.
[66] G. Laporte, F. Semet, 5. Classical Heuristics for the capacitated VRP, in: The Vehicle Routing Problem, in: Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, 2002, pp. 109–128.
[67] I. Or, Traveling Salesman-Type Combinatorial Optimization Problems and Their Relation to the Logistics of Regional Blood Banking (Ph.D. thesis), Department of Industrial Engineering and Management Sciences Northwestern University, Evanston, IL, 1976.
[68] E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, J.Y. Potvin, A tabu search heuristic for the vehicle routing problem with soft time windows, Transp. Sci. 31 (November 2016) (1997) 170–186.
[69] J.-Y. Potvin, S. Bengio, The vehicle routing problem with time windows Part II: Genetic search, INFORMS J. Comput. 8 (2) (1996) 165–172, Publisher: INFORMS.
[70] B.M. Ombuki, M. Nakamura, M. Osamu, A hybrid search based on genetic algorithms and tabu search for vehicle routing, in: 6th IASTED Intl. Conf. on Artificial Intelligence and Soft Computing (ASC 2002), 2002, pp. 176–181.
[71] K. Sörensen, Metaheuristics-the metaphor exposed, Int. Trans. Oper. Res. 22 (1) (2015) 3–18.
[72] J.R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, N. Herazo-Padilla, A literature review on the vehicle routing problem with multiple depots, Comput. Ind. Eng. 79 (2015) 115–129.
[73] R. Elshaer, H. Awad, A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants, Comput. Ind. Eng. 140 (2020) 106242.
[74] N. Jozefowiez, F. Semet, E.-G. Talbi, Multi-objective vehicle routing problems, European J. Oper. Res. 189 (2) (2008) 293–309.
[75] J. Holland, Adaptation in Artificial and Natural Systems, The University of Michigan Press, Ann Arbor, 1975, p. 232.
[76] T. Vidal, T.G. Crainic, M. Gendreau, N. Lahrichi, W. Rei, A hybrid genetic algorithm for multidepot and periodic vehicle routing problems, Oper. Res. 60 (3) (2012) 611–624.
[77] J.-F. Cordeau, M. Gendreau, G. Laporte, A tabu search heuristic for periodic and multi-depot vehicle routing problems, Networks 30 (2) (1997) 105–119.

Anees Abu-Monshar is currently a PhD candidate in simulation and optimisation of logistics systems 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 PhD 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.