

Jeong, Byeong-Min; Jang, Dae-Sung; Choi, Han-Lim

## Article

# Decentralized message passing algorithm for heterogeneous multi-depot vehicle routing problems

Operations Research Perspectives

**Provided in Cooperation with:**

Elsevier

*Suggested Citation:* Jeong, Byeong-Min; Jang, Dae-Sung; Choi, Han-Lim (2025) : Decentralized message passing algorithm for heterogeneous multi-depot vehicle routing problems, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 14, pp. 1-15, <https://doi.org/10.1016/j.orp.2025.100341>

This Version is available at:

<https://hdl.handle.net/10419/325818>

### Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

### Terms of use:

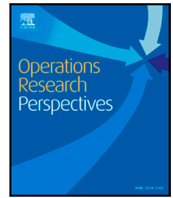
*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by-nc-nd/4.0/>



# Decentralized message passing algorithm for heterogeneous multi-depot vehicle routing problems

Byeong-Min Jeong<sup>a</sup>, Dae-Sung Jang<sup>b</sup> , Han-Lim Choi<sup>a</sup>

<sup>a</sup> KAIST, Daejeon, 34141, Republic of Korea

<sup>b</sup> Korea Aerospace University, Goyang, 10540, Republic of Korea

## ARTICLE INFO

### Keywords:

Heterogeneous multi-depot VRP  
Belief propagation  
Message passing  
Heuristic algorithm

## ABSTRACT

In this paper, a novel message-passing algorithm, named AMP-R, based on belief propagation is proposed to solve the heterogeneous multi-depot vehicle routing problem (HMDVRP) in a distributed manner. Unlike traditional approaches, this is the first attempt to decentralize the solution process for the HMDVRP at the depot level, enabling each depot to independently compute and exchange messages to derive conflict-free solutions. The HMDVRP requires assigning customers to depots and determining routes that minimize total travel cost. By reformulating the problem as a maximum a posteriori estimation in a graphical model comprising depot and customer nodes, The proposed approach enables decentralized message calculation and exchange between depots, effectively addressing various types of the HMDVRP. In this process, it is derived that each message calculation can be reduced to a subset-visit traveling salesman problem or a capacitated vehicle routing problem, and approximation techniques are proposed to address these computational challenges. Furthermore, to ensure solution convergence and feasibility, message buffers and a refinement process are introduced. Extensive simulations demonstrate that the proposed AMP-R algorithm yields near-optimal solutions with computational efficiency, offering practical performance for complex large-scale instances where finding optimal solutions is challenging.

## 1. Introduction

The vehicle routing problem (VRP) [1–3] is a classical combinatorial optimization problem that has significant applications in the field of logistics, transportation, and multi-robot planning. The VRP involves a fleet of vehicles departing from a depot to visit all customers and then returning to the depot. The wide variety of objective functions and constraints in VRP has led to a diverse range of VRP variants. Initially introduced as the truck dispatching problem [4], VRP has evolved to address more complex and realistic scenarios. The variant that considers multiple depots is known as the multi-depot VRP (MDVRP) [5]. Another variant, which accounts for different types of vehicles, is referred to as the heterogeneous fleet VRP (HFVRP) [6,7]. In many cases, vehicles considered in VRP and its variants have limited capacity to serve customer demands, whereas there has also been research on HFVRP that considers uncapacitated vehicles [6]. Additionally, studies have addressed HFVRP scenarios where the number of vehicles departing from each depot is either limited [8–10] or unlimited [11–13]. Examples of heterogeneous fleets include studies such as [9–13], which address vehicles with fixed and variable operating costs, as well as [14], which examines vehicles with different capacities and traveling

costs, and [15,16], which addresses factors such as carbon emissions and recycling costs. Some researches also incorporate time-dependent traveling time [17,18]

The VRP involving both multiple depots and heterogeneous vehicles is known as the heterogeneous MDVRP (HMDVRP). This problem was first introduced in [19], where a multi-level composite heuristic with reduction tests was proposed for both single-depot and multi-depot routing. Several subsequent studies have focused on meta-heuristic solutions to address the HMDVRP. For example, the problem of distributing large quantities of a single product to customers was modeled as the HMDVRP and solved using an ant colony optimization algorithm in [20]. A variable neighborhood search (VNS)-based algorithm for the HMDVRP was introduced in [21]. Another study [22] combined two meta-heuristics, the VNS and the greedy randomized adaptive search procedure, to effectively solve the HMDVRP with backhauls. Similarly, a research [23] proposed two approaches, an enhanced iterated local search and a hybrid genetic search with advanced diversity control, to solve various variants of the HMDVRP. In [24], a modified genetic algorithm was developed for the HMDVRP, integrating genetic operators with local search strategies to improve the efficiency of the solution. In

\* Correspondence to: 76 Hanggongdaehak-ro, Deogyang-gu, Goyang-si, Gyeonggi-do 10540, Republic of Korea.  
E-mail address: [dsjang@kau.ac.kr](mailto:dsjang@kau.ac.kr) (D.-S. Jang).

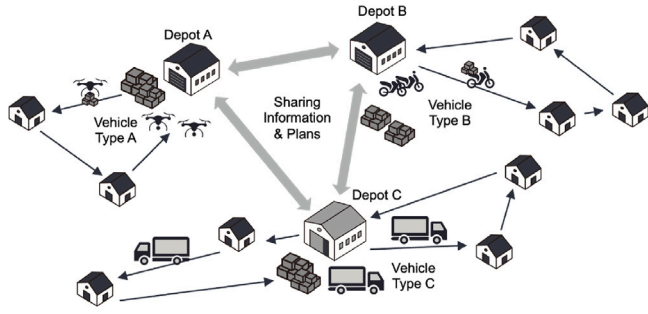


Fig. 1. Conceptual diagram illustrating the distributed collaboration among depots for solving HMDVRP, which can be applied to logistics, delivery, multi-robot mission planning, and warehouse management.

contrast to these meta-heuristic approaches, another study [25] aimed to find the optimal solution for an HMDVRP with split deliveries using a Benders-based branch-and-cut algorithm.

In many studies, as previously mentioned, solutions to the HMDVRP have typically been derived using a centralized planner. Research on solving the MDVRP in a decentralized manner remains limited. For example, one study [26] explored parallel computation to derive solutions using multiple processors. However, it relied on shared memory for the genetic algorithm, which made it challenging to allocate computing agents across separate computers in a distributed setup. Soeanu et al. [27] proposed a decentralized approach in which multiple computing agents, employing an optimization heuristic based on a random number generator, shared intermediate solutions to find near-optimal results. Nonetheless, in this approach, the intermediate solutions needed to be shared among all computing agents to effectively solve the problem. Another study [28] utilized the Vickery–Clarke–Groves mechanism to enable route selection per depot. However, since the underlying heuristic in this study was based on the work of Soeanu et al. [27], it encountered the same limitation. In [29], MDVRP was solved by applying a virtual center, treating it as a single depot VRP, and utilizing an ant colony optimization that supports parallel computation to reduce computational complexity. Additionally, Abu-Monshar et al. [30] proposed the messaging protocol-based heuristics optimization (MPHO) model, allowing solutions to be obtained through interactions between agents. Furthermore, K-means was used for parallel clustering in [31], with the number of clusters matching the number of depots, followed by route planning. These three studies all addressed distributed computing but did not consider communication between agents performing the computations. Therefore, while computation time can be reduced within a single computer, the method of performing computations with physically separated depots was not considered.

To the best of our knowledge, no effective solution has been proposed to address the MDVRP or HMDVRP in a decentralized network where computing agents share limited information (refer Fig. 1). However, in scenarios for large-scale logistics where a large number of customers and depots are spatially distributed, it may not be feasible to effectively aggregate problem information into a centralized computation unit. This is particularly relevant for scenarios involving dynamic conditions, such as mobile robot mission planning where real-time decentralized decision-making is crucial, which can also be modeled as an HMDVRP. In such cases, where conditions are time-varying and computations rely on a distributed network of agents, decentralized algorithms are essential.

In this paper, we propose a belief propagation (BP) [32] based algorithm, named approximate message passing and refinement (AMP-R), to solve the HMDVRP in a distributed manner. To apply the BP-based algorithm, we present a traditional integer programming formulation of the HMDVRP and, for the first time, reformulate it as a stochastic inference problem in a graphical model, where the objective is

to find decision variables that maximize the joint probability of the model. BP is an approximation method originally developed for solving stochastic inference problems and has been applied to find optimal or near-optimal solutions for various combinatorial optimization problems using its max-product or max-sum forms [33–38]. The graphical model and the BP-based AMP-R algorithm proposed in this paper provide a theoretical foundation for formulating and solving variants of the HMDVRP in a distributed manner, accounting for vehicle heterogeneity, capacity constraints, and the vehicle limit of each depot. The reformulation of HMDVRP as a probabilistic inference problem contributes to a novel perspective on solving VRP-related combinatorial optimization problems through message-passing approaches. The AMP-R algorithm enables decentralized computation by exchanging messages between computational agents, making it suitable for practical applications, such as large-scale logistics and multi-robot planning. This decentralized computation framework allows AMP-R to be effectively applied regardless of whether the computation units are depots, vehicles, or other levels of distribution. Since the message calculation required for solving the HMDVRP is NP-hard, this paper proposes approximation techniques for efficient computation. Additionally, message buffers and a refinement process are incorporated to enhance convergence and ensure solution feasibility in decentralized environments. Monte Carlo simulations demonstrate that the proposed approach outperforms a centralized greedy algorithm and provides practical solutions for non-trivial large-scale problems where finding the optimal solution is challenging.

The structure of the rest of this paper is as follows. Section 2 introduces the mathematical formulations of the HMDVRP in the frameworks of integer programming and stochastic inference. Section 3 describes the proposed message-passing algorithm designed to efficiently obtain solutions for the HMDVRP. This Section presents the derivation of the max-product message computation equations for solving the HMDVRP represented in a graphical model and discusses approximation techniques and problem transformations used to calculate these messages efficiently. Additionally, it explains the refinement step to enhance solution feasibility and the distributed implementation of the algorithm. Section 4 presents a numerical evaluation of the proposed algorithm's performance across various HMDVRP variants, using Monte Carlo simulations and comparisons with greedy and optimal solutions. Section 5 concludes the paper.

## 2. Mathematical formulations

This section sequentially explains two mathematical representations for describing HMDVRP. Section 2.1 introduces conventional integer programming to explicitly state the constraints of the HMDVRP addressed in this study, and presents the key notations commonly used in both representations. Section 2.2 describes a method to express HMDVRP as a statistical inference problem in a graphical model. It demonstrates that this problem can represent the constraints through the design of conditional probability distribution functions and is equivalent to finding a solution that maximizes the objective function of the integer program.

### 2.1. Definition of HMDVRP and integer programming

The HMDVRP consists of a set of customers,  $C = \{c_1, \dots, c_{n_c}\}$ , a set of depots,  $D = \{d_1, \dots, d_{n_d}\}$ , and a set of vehicles,  $V = \{v_1, \dots, v_{n_v}\}$ . Here,  $n_c$ ,  $n_d$ , and  $n_v$  represent the number of customers, depots, and vehicles, respectively. Each customer  $c \in C$  has a unique demand  $l_c$ , and each vehicle  $v \in V$  has a unique capacity  $L_v$  to handle this demand. The set  $U = C \cup D$  includes all customers and depots, and for convenience, the elements of this set are referred to as nodes. The cost or expenditure  $e_{ijk}$  refers to the cost incurred when vehicle  $k \in V$  travels from one node  $i \in U$  to another node  $j \in U$ . The objective of the HMDVRP is to minimize the total travel cost of the vehicles while satisfying several

constraints, which include: (i) Each customer can only be served by one vehicle and must be visited exactly once. (ii) Each vehicle  $v$  must depart from and return to its assigned depot  $d^v$ . (iii) In the capacitated version of the problem, the total demand of the customers visited by each vehicle must not exceed its capacity  $L_v$ . (iv) Each depot  $d \in \mathbf{D}$  has a predefined number of vehicles  $n_{v,d}$ , and the sum of all vehicles across the depots satisfies  $\sum_{d \in \mathbf{D}} n_{v,d} = n_v$ . The number of vehicles is considered unlimited if  $n_{v,d} \geq n_c$  for all depots.

Since the definition and formulation of HMDVRP vary slightly among studies, the objective and constraints of the HMDVRP addressed in this study are explicitly formulated as an integer programming problem, as shown in (1)–(8). This formulation is based on [39,40] and incorporates the Dantzig–Fulkerson–Johnson subtour elimination constraint [41]. Note that  $i, j$ , and  $k$  represent the nodes and vehicles themselves, not merely their index numbers.

$$\min \sum_{i \in \mathbf{U}} \sum_{j \in \mathbf{U}} \sum_{k \in \mathbf{V}} x_{ijk} e_{ijk} \quad (1)$$

$$\text{s.t.} \sum_{i \in \mathbf{U}} \sum_{k \in \mathbf{V}} x_{iik} = 0, \quad (2)$$

$$\sum_{i \in \mathbf{U}} \sum_{k \in \mathbf{V}} x_{ijk} = 1, \quad \forall j \in \mathbf{C}, \quad (3)$$

$$\sum_{i \in \mathbf{U}} x_{ijk} = \sum_{i \in \mathbf{U}} x_{jik} \leq 1, \quad \forall j \in \mathbf{C}, \forall k \in \mathbf{V}, \quad (4)$$

$$\sum_{i \in \mathbf{D} \setminus d^k} \sum_{j \in \mathbf{C}} x_{ijk} + x_{jik} = 0, \quad \forall k \in \mathbf{V}, \quad (5)$$

$$\sum_{i \in \mathbf{U}} \sum_{j \in \mathbf{C} \setminus i} x_{ijk} l_j \leq L_k, \quad \forall k \in \mathbf{V}, \quad (6)$$

$$\sum_{i \in \mathbf{S}} \sum_{j \in \mathbf{S}} x_{ijk} \leq |\mathbf{S}| - 1, \quad \mathbf{S} \subseteq \mathbf{C}, |\mathbf{S}| > 1, \quad \forall k \in \mathbf{V}, \quad (7)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i \in \mathbf{U}, \forall j \in \mathbf{U}, \forall k \in \mathbf{V}. \quad (8)$$

$x_{ijk}$  is the decision variable of the integer program. If  $x_{ijk} = 1$ , the path<sup>1</sup> from node  $i$  to node  $j$  is taken by vehicle  $k$ ; otherwise,  $x_{ijk} = 0$ . Eq. (1) represents the objective, which is the sum of the costs of all selected paths. In the homogeneous MDVRP, the path cost  $e_{ijk}$  becomes indistinguishable, i.e.,  $e_{ij}$ . Eq. (2) signifies that a path returning to the same node cannot be selected, and while Eq. (3) ensures that each customer node is visited exactly once. Eq. (4) implies that the number of paths entering and leaving any node must be equal and must be less than or equal to 1. Eq. (5) ensures no path is drawn from depots other than the one to which each vehicle belongs. Eq. (6) enforces the capacity constraint of each vehicle, and Eq. (7) is the subtour elimination constraint, which guarantees a closed tour of selected paths for each vehicle. For an uncapacitated problem, (6) is removed. Furthermore, if vehicles in each depot are uncapacitated and homogeneous, and the path costs satisfy the triangle inequality, only one vehicle is required for each depot in an optimal solution ( $n_{v,d} = 1, \forall d \in \mathbf{D}$ ), which leads to  $n_d = n_v$ .

## 2.2. Graphical model

To apply the BP algorithm, the HMDVRP is formulated as a maximum a posteriori (MAP) state estimation problem on a graphical model. A graphical model consists of nodes with random variables and edges that are associated with functions indicating conditional dependencies between those variables. In this study, we utilize the approach presented in [42] that applies BP to the set cover problem, and represent the HMDVRP using a pairwise Markov random field (MRF), a type of

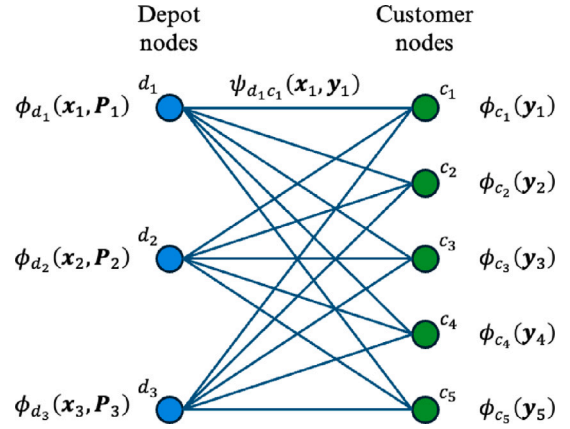


Fig. 2. Example graphical model for representing the HMDVRP.

graphical model. In a pairwise MRF, each undirected edge connecting a pair of nodes is associated with a potential function of the random variables at the nodes. The MAP state estimation problem involves finding the values of the variables that maximize the joint probability of all the random variables in the graphical model, which are defined by the potential functions.

Fig. 2 shows an example of representing the HMDVRP as a graphical model. Nodes are created for depots and customers, and edges connect each depot node to the nodes of customers that can be visited from the depot. The graph is bipartite in that the edges are placed only between the depot node group  $\mathbf{D} = \{d_1, \dots, d_{n_d}\}$  and the customer node group  $\mathbf{C} = \{c_1, \dots, c_{n_c}\}$  with no edges within either group. Each node is assigned variables that represent the allocation of vehicles to customers and their visiting order. Each edge has a potential function based on the variables assigned to its incident nodes, and each node has a potential function based on its own variables. Unlike in the integer program in Section 2.1, the graphical model is formulated at the depot-level rather than the vehicle-level. This approach is more natural if BP is applied in a decentralized environment, assuming that computations are performed at the depots. However, if every vehicle has computational capabilities, a graphical model can be constructed with the nodes for the vehicles, allowing BP to be applied through message passing between vehicles.

In the graphical model, the  $i$ th depot node has two types of variables: a binary vector  $\mathbf{x}_i \in \{0, 1\}^{n_c}$  that represents the assignment of customers, and a set  $\mathbf{P}_i = \{\mathbf{p}_v : \forall v \in \mathbf{V}_{d_i}\}$  of variable-length lists  $\mathbf{p}_v$ , which represent the order visiting nodes for each vehicle, where  $\mathbf{V}_{d_i} = \{v : \forall v, d^v = d_i\}$  denotes the set of vehicles at  $d_i$ . Also, a customer node has a binary vector  $\mathbf{y}_j \in \{0, 1\}^{n_d}$ , indicating which depot it is assigned to. Note that, unlike in Section 2.1,  $i$  and  $j$  here represent the index numbers (integers) of depot and customer nodes.  $\mathbf{x}_i[j]$  and  $\mathbf{y}_j[i]$  refer to the  $j$ th element of the binary vector at the  $i$ th depot node and the  $i$ th element of the binary vector at the  $j$ th customer node, respectively.  $\mathbf{x}_i[j] = \mathbf{y}_j[i] = 1$  indicates that a vehicle from the  $i$ th depot visits  $j$ th customer, while  $\mathbf{x}_i[j] = \mathbf{y}_j[i] = 0$  indicates otherwise. The list  $\mathbf{p}_v$  is an ordered list of nodes, representing the route a vehicle  $v$  takes, and it always starts and ends with the depot  $d^v$  to which the vehicle belongs. For example, if vehicle  $v$  belonging to depot  $d_1$  visits customers  $c_3$ ,  $c_7$ , and  $c_2$  in order, then  $\mathbf{p}_v = (d_1, c_3, c_7, c_2, d_1)$ .

According to the Hammersley–Clifford theorem [43], any strictly positive distribution of a graphical model factorizes based on its cliques. In a bipartite graph of which the maximal clique size is 2, the joint probability distribution  $P(\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_d}\}; \mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n_c}\}; \mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n_d}\})$  of all variables in the graph is the product of all potential functions defined at its edges and nodes:

$$P(\mathbf{X}, \mathbf{Y}, \mathbf{P}) = \frac{1}{Z} \prod_{i,j} \psi_{d_i c_j}(\mathbf{x}_i, \mathbf{y}_j) \prod_i \phi_{d_i}(\mathbf{x}_i, \mathbf{P}_i) \prod_j \phi_{c_j}(\mathbf{y}_j) \quad (9)$$

<sup>1</sup> In this paper, to avoid confusion with the term ‘edge’ used in graphical models, we define a ‘path’ as a direct connection between two nodes, and a ‘route’ as a series of paths that sequentially visit multiple nodes. It is important to note that this terminology differs from the conventional meanings of ‘path’ and ‘edge’ in graph theory.



where  $Z$  is the normalization constant for  $P$ . By appropriately designing the potential functions that constitute the joint probability  $P$  as shown below, it is possible to incorporate the objective and various constraints of the HMDVRP. This allows the problem of finding the MAP state that maximizes  $P$  to become equivalent to finding the optimal solution of the HMDVRP. Specifically, the potential functions are designed so that its total product  $P$  becomes non-zero only if the solution  $(X, Y, P)$  is feasible and  $P$  is maximized when corresponding total cost of vehicle tours is minimized.

$$X^*, Y^*, P^* = \operatorname{argmax}_{X, Y, P} P(X, Y, P) \quad (10)$$

$$\psi_{d_i, c_j}(x_i, y_j) = \begin{cases} 1, & x_i[j] = y_j[i], \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

$$\phi_{d_i}(x_i, P_i) = \zeta(x_i, P_i) \eta(P_i) e^{-w(P_i)}, \quad (12)$$

$$\zeta(x_i, P_i) = \begin{cases} 1, & x_i[j] = 1, \forall j \in \mathbb{J}(P_i) \text{ and } x_i[j] = 0, \forall j \notin \mathbb{J}(P_i) \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

$$\eta(P_i) = \begin{cases} 1, & \sum_{k=2}^{|p_v|-1} l(p_v[k]) \leq L_v, \forall v \in V_{d_i}, \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

$$w(P_i) = \sum_{v \in V_{d_i}} \sum_{k=1}^{|p_v|-1} e(p_v[k], p_v[k+1], v), \quad (15)$$

$$\phi_{c_j}(y_j) = \begin{cases} 1, & \sum_i y_j[i] = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

In Eq. (11), the potential  $\psi_{d_i, c_j}(x_i, y_j)$ , defined for each edge connecting a depot node to a customer node, equals 1 only when the values of  $x_i[j]$  and  $y_j[i]$  match; otherwise, it equals 0. The joint probability  $P$  cannot attain its maximum if even a single potential function is 0. Thus,  $\psi_{d_i, c_j}(x_i, y_j)$  ensures the consistency between the two variables representing the assignment between the depot and the customer. The potential function  $\phi_{d_i}(x_i, P_i)$  of the depot node represents the consistency among the node variables (Eq. (13)), the capacity constraints (Eq. (14)), and the costs of the tours formed by the vehicles (Eq. (15)). The function  $\zeta(x_i, P_i)$  in Eq. (13) ensures that for each customer in the set  $P_i$ , which comprises the list of customers visited by the depot's vehicles, the corresponding element in  $x_i$  is set to 1; otherwise, it is set to 0.  $\mathbb{J}(P_i)$  denotes the set of all customer indices that appear in the  $p_v$  lists within  $P_i$ . In Eq. (14),  $p_v[k]$  refers to the  $k$ th element in the list of nodes visited by vehicle  $v$  belonging to the depot, and  $l(\cdot)$  is a function representing the demand of a customer node. Therefore, a set  $P_i$  that makes  $\eta(P_i)$  non-zero satisfies the capacity constraints for all vehicles of the depot. In Eq. (15),  $w(P_i)$  signifies the sum of path costs of vehicles in  $V_{d_i}$ , forming their tours according to their respective  $p_v$  lists.  $e(a, b, v) \equiv e_{a, b, v}$  represents the cost when vehicle  $v$  travels from node  $a$  to node  $b$ . From Eqs. (12) and (15), for a feasible solution  $(X, Y, P)$  that satisfies all other constraints, assuming the total cost of the tours is  $W$ , the value of the joint probability  $P$  is  $e^{-W}$ . Therefore,  $P$  reaches its maximum value with the minimum path cost  $W^*$  of the HMDVRP. Eq. (16) ensures that exactly one element in  $y_j$  is set to 1, ensuring that the  $j$ th customer is visited exactly once.

### 3. Solution approach

This section provides a detailed explanation of how the distributed algorithm proposed in this study for solving HMDVRP is derived from BP rules and how it generates a feasible approximate solution through the proposed procedure. Section 3.1 presents the message update and MAP state computation rules when the max-product algorithm, a BP method for estimating the MAP state, is applied to the graphical model introduced in Section 2.2. Section 3.2 derives a simplified message representation by reducing the dimensionality of message computations between depot and customer nodes, reflecting the characteristics of the potential functions. As a result, it is shown that computing the messages transmitted from depot nodes to customer nodes requires

solving VRP-like problems. Section 3.3 outlines the overall procedure of the proposed algorithm and describes the refinement step required to ensure the feasibility of the solution. Section 3.4 explains the approximate computation method for the messages transmitted from depot nodes to customer nodes, which is a core component of the algorithm. A representative route, commonly used in computing messages transmitted to multiple customer nodes, is defined, and a greedy method is proposed to approximately solve the VRP problem required for the message computation. Additionally, a transformation technique applicable to the uncapacitated case is introduced. Finally, Section 3.5 describes how the proposed method is implemented in a distributed network structure at the depot level.

#### 3.1. The max-product algorithm

The max-product algorithm [32] is a message-passing algorithm used to solve the MAP state estimation problem in a graphical model. It iteratively exchanges messages between the nodes in the graphical model to determine the MAP state. Messages sent from a node are computed based on those received from neighboring nodes and are used to determine the estimate of the posterior distribution, called the *belief*. In each iteration  $q$ , the algorithm exchanges messages and calculates the belief for each node, and the value that maximizes the belief is chosen as the MAP state estimate for that node's variable. Thus, the max-product algorithm can determine the MAP state of all variables in the graph in a distributed manner through localized message exchanges.

Since the graphical model for the HMDVRP is a bipartite graph, the messages can be categorized into two types: ones sent from depots to customers (Eq. (17)) and others sent from customers to depots (Eq. (18)).

$$m_{d_i, c_j}^q(y_j) = \frac{1}{z_{d_i, c_j}} \max_{x_i, P_i} \psi_{d_i, c_j}(x_i, y_j) \phi_{d_i}(x_i, P_i) \prod_{c_k \in \mathcal{C} \setminus c_j} \tilde{m}_{c_k, d_i}^{q-1}(x_i, P_i) \quad (17)$$

$$m_{c_j, d_i}^q(x_i, P_i) = \frac{1}{z_{c_j, d_i}} \max_{y_j} \psi_{d_i, c_j}(x_i, y_j) \phi_{c_j}(y_j) \prod_{d_k \in \mathcal{D} \setminus d_i} \tilde{m}_{d_k, c_j}^{q-1}(y_j) \quad (18)$$

The message from a node  $a$  to another node  $b$  is a distribution for the variable of receiver  $b$ . For example,  $m_{d_i, c_j}^q(y_j)$  in Eq. (17) is required to be computed for all possible values of  $y_j$ . At this stage, for each possible value of  $y_j$ , the maximum value of the product of functions involving  $x_i$  and  $P_i$  on the right-hand side of Eq. (17) is computed. This product consists of the edge  $\psi_{d_i, c_j}$  and node potentials  $\phi_{d_i}$ , as well as the messages received from all neighboring nodes of  $d_i$ , excluding  $c_j$ . Since the message represents a probability distribution, the sum of the message values over all  $y_j$  must equal 1. Accordingly, the values obtained from the max operation are normalized by dividing by their sum, denoted as  $z_{d_i, c_j}$ , to ensure that the total sum of the values in the message equals 1. In a similar manner,  $m_{c_j, d_i}^q$  in Eq. (18) is computed by finding the maximum value over  $y_j$  for each possible value of the variable  $x_i$  and  $P_i$  of  $d_i$ , followed by normalization.

On the right-hand side of Eqs. (17) and (18),  $\tilde{m}$  represents the average of the messages stored in *buffer*. Rather than sending the messages computed in each iteration directly, as in Eqs. (19) and (20), the most recent  $N$  messages are stored in buffer, and their average is transmitted to the neighbor nodes. This approach is intended to improve the convergence of the algorithm by incorporating a damping effect, similar to that of the damped BP [44].

$$\tilde{m}_{d_i, c_j}^q(y_j) = \frac{1}{N} \sum_{n=q-N+1}^q m_{d_i, c_j}^n(y_j) \quad (19)$$

$$\tilde{m}_{c_j, d_i}^q(x_i) = \frac{1}{N} \sum_{n=q-N+1}^q m_{c_j, d_i}^n(x_i) \quad (20)$$

The belief  $b^q$  of each node is calculated by multiplying the messages received from all neighboring nodes with the node potential  $\phi$ . Similar to the messages, this must be computed for all possible values of the node's variables. The belief for the depot nodes can be obtained

as shown in (21), and this contains the assignment and the vehicle schedules for visiting the customers. In (22), the values of  $\hat{x}_i^q$  and  $\hat{P}_i^q$  that maximize this belief are the MAP state estimates, providing the solution to the HMDVRP.

$$b_{d_i}^q(x_i, P_i) = \frac{1}{z_{d_i}} \phi_{d_i}(x_i, P_i) \prod_{c_k \in C} \tilde{m}_{c_k d_i}^{q-1}(x_i) \quad (21)$$

$$\hat{x}_i^q, \hat{P}_i^q = \operatorname{argmax}_{x_i, P_i} b_{d_i}^q(x_i, P_i) \quad (22)$$

### 3.2. Message calculation

In the max-product algorithm, the most computationally intensive task is generally the calculation of messages, as shown in Eqs. (17) and (18). Each message represents a distribution over all possible values of the receiving node's variable, and to compute this distribution, a max operation is required over the transmitting node's variable for each of the possible values. As a result, the computational burden increases as the number of possible values for the node variables grows. For example, the message sent from a depot node to a customer node is a function of  $y_j$ , and the max operation needs to be carried out  $2^{n_d}$  times, which corresponds to the number of possible combinations of the binary vector variable  $y_j$ .

However, in the graphical model described in Section 2.2, the number of required operations is significantly smaller, with only two computations needed per message. Due to the potential function  $\psi_{d_i c_j}(x_i, y_j)$  in Eq. (17), the result of the max operation on the right-hand side is independent of the values of  $y_j$  except for the  $i$ th element  $y_j[i]$ , and the max operation yields a non-zero value only when  $x_i[j] = y_j[i]$ . Therefore, there is no need to perform max operations for values other than the two possible cases for  $y_j[i]$ , which are 0 and 1. Similarly, in Eq. (18), the right-hand side is determined by the value of  $x_i[j]$ , which is one of the variables of the receiving node  $d_i$ , and is independent of the other components of the  $x_i$  vector or the elements of  $P_i$ . Thus, as shown in Eqs. (23) and (24), it is sufficient to compute only for the two possible values (0 and 1) of the binary vector component of the receiver's variable, significantly reducing the computational complexity for each message.

$$m_{d_i c_j}^q(y_j[i] = 0) = \frac{1}{z_{d_i c_j}} \max_{x_i, P_i} \phi_{d_i}(x_i, P_i) \prod_{c_k \in C \setminus c_j} \tilde{m}_{c_k d_i}^{q-1}(x_i[k]) \quad (23)$$

$$m_{c_j d_i}^q(x_i[j] = 1) = \frac{1}{z_{c_j d_i}} \max_{y_j} \phi_{c_j}(y_j) \prod_{d_k \in D \setminus d_i} \tilde{m}_{d_k c_j}^{q-1}(y_j[k]) \quad (24)$$

Now, Eq. (23), which describes the message sent from a depot to a customer, is examined in detail. The right-hand side of the equation is computed for both binary values of  $y_j[i]$ . Specifically, when  $y_j[i] = x_i[j] = 0$ , the product of potential  $\phi_{d_i}$  and messages is maximized over the assignment  $x_i$  and the set of vehicle routes  $P_i$  that do not include the  $j$ th customer  $c_j$ . Conversely, when  $y_j[i] = x_i[j] = 1$ , the maximization is performed over  $x_i$  and  $P_i$  that necessarily include  $c_j$ . Based on this, the message for each binary value of  $y_j[i]$  can be summarized as follows:

$$m_{d_i c_j}^q(y_j[i] = 0) = \frac{1}{z_{d_i c_j}} \max_{x_i, P_i^{j,0}} \eta(P_i^{j,0}) e^{-w(P_i^{j,0})} \prod_{c_k \in P_i^{j,0}} \tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1) \times \prod_{c_k \notin \{P_i^{j,0} \cup c_j\}} \tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0) \quad (25)$$

$$m_{d_i c_j}^q(y_j[i] = 1) = \frac{1}{z_{d_i c_j}} \max_{x_i, P_i^{j,1}} \eta(P_i^{j,1}) e^{-w(P_i^{j,1})} \prod_{c_k \in \{P_i^{j,1} \setminus c_j\}} \tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1) \times \prod_{c_k \notin P_i^{j,1}} \tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0) \quad (26)$$

In these two equations,  $P_i^{j,0}$  represents the set of vehicle routes within the depot that do not include customer  $c_j$ , while  $P_i^{j,1}$  represents the set containing at least one route that includes  $c_j$ . The consistency between  $x_i$  and  $P_i$  is enforced, and thus  $\zeta(x_i, P_i) = 1$  and is omitted

from the equations. In addition, if a customer node  $c_k$  is included in  $P_i^{j,0}$  or  $P_i^{j,1}$ , the received message for  $x_i[k] = 1$  is multiplied, and if not, the message for  $x_i[k] = 0$  is used.

Since the messages are normalized by  $z_{d_i c_j}$  after the max operation in Eqs. (25) and (26) to ensure that their sum equals 1, multiplying or dividing both equations by the same value does not affect the message values. Therefore, dividing both equations by  $\prod_{c_k \in C \setminus c_j} \tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0)$  simplifies them as follows:

$$m_{d_i c_j}^q(y_j[i] = 0) = \frac{1}{z_{d_i c_j}} \max_{x_i, P_i^{j,0}} \eta(P_i^{j,0}) e^{-w(P_i^{j,0})} \prod_{c_k \in P_i^{j,0}} \frac{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1)}{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0)} \quad (27)$$

$$m_{d_i c_j}^q(y_j[i] = 1) = \frac{1}{z_{d_i c_j}} \max_{x_i, P_i^{j,1}} \eta(P_i^{j,1}) e^{-w(P_i^{j,1})} \times \prod_{c_k \in \{P_i^{j,1} \setminus c_j\}} \frac{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1)}{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0)} \quad (28)$$

It can be observed that Eqs. (27) and (28) represent the same type of optimization problem, differing only in whether the receiving node  $c_j$  is included or not, while both aim to find the solution  $x_i$  and  $P_i$  (specifically,  $P_i^{j,0}$  or  $P_i^{j,1}$ ) that maximize the right-hand side. Here,  $x_i$  is a variable indicating whether customers are assigned to depot  $d_i$ , and  $P_i$  is the set of routes representing the order in which vehicles from the depot visit the assigned customers. Additionally,  $\eta(P_i)$  is a function that reflects the capacity constraint, and  $w(P_i)$  is the sum of the route costs for the selected routes  $P_i$ . The last term takes the product of the ratios of received messages only for the customers included in  $P_i$ . Therefore, the max operation for calculating the message becomes a VRP-like problem, where customers assigned to this depot are selected by considering the ratio of messages received from each customer, and the minimum-cost routes visiting the selected customers are found. Since visiting a customer always results in a negative route cost, the exponential term becomes less than 1. However, there may exist a non-trivial solution that maximizes the right-hand side of the equation if there is a customer whose message ratio is sufficiently greater than 1 so that it offsets the decrease in the exponential.

Table 1 summarizes how the depot-to-customer message computation for each variant of the HMDVRP discussed in this paper corresponds to different optimization problems. Unlike in standard VRP, the message computations in Eqs. (27) and (28) involve visiting only a subset of customers rather than all of them, which is indicated by prefixing each problem with *subset-visit* (SV). When vehicle capacity is constrained, the problem is classified as either the subset-visit capacitated VRP (SV-CVRP) or the subset-visit capacitated VRP with a limited number of vehicles (SV-CVRP-LV), depending on whether the number of vehicles per depot is limited. In cases where vehicle capacity is not constrained, it can be treated as a type of the TSP where each vehicle's depot is considered a distinct starting point, and this is referred to as the subset-visit multi-TSP (SV-mTSP). In this case, the edges of the TSP are initially assigned positive path costs, but the negative of the logarithm of the received message ratio in Eqs. (27) and (28) is added to the costs of the paths visiting each customer node (refer Section 3.4.2). A non-trivial subset-visit solution exists only if some of these path costs are negative.

In solving the HMDVRP using the max-product algorithm, a form of BP, it becomes apparent that each time a message is calculated, it requires solving another NP-hard problem. Since BP is used as a heuristic to approximate a solution for the HMDVRP, solving these NP-hard problems exactly for each message calculation would be highly inefficient. Therefore, this paper proposes an approach, named AMP, to compute these messages approximately, as detailed in Section 3.4. For uncapacitated cases, when the vehicles within a depot are homogeneous and the path costs satisfy the triangular inequality, there always exists an optimal solution where only one vehicle departs from each depot. A method for converting such an SV-TSP to an asymmetric

**Table 1**  
Variants of HMDVRP covered in this study.

Capacity	Number of vehicles	Problem type in message calculation	Notes
Capacitated	Unlimited	SV-CVRP: subset-visit capacitated VRP	Always be able to visit all customers.
Capacitated	Limited	SV-CVRP-LV: subset-visit capacitated VRP with a limited number of vehicles	
Uncapacitated	Unlimited/limited	SV-mTSP: subset-visit multi-TSP with negative edge costs	If vehicles in each depot are homogeneous and the path costs satisfy the triangle inequality, one vehicle is sufficient for each depot, i.e. $n_{v,d} = 1$ .

TSP (ASP) and then applying a state-of-the-art TSP heuristic is also presented in Section 3.4.2.

The message sent from a customer to a depot is significantly simplified because only one binary element of  $\mathbf{y}_j$  can be nonzero, due to the potential function  $\phi_{c_j}(\mathbf{y}_j)$  in (16). Taking into account the consistency between the two node variables,  $\mathbf{y}_j[i] = \mathbf{x}_i[j]$ , the message equation in (24) can be expressed for each binary value of the receiving node variable  $\mathbf{x}_i[j]$ :

$$m_{c_j d_i}^q(\mathbf{x}_i[j] = 0) = \frac{1}{z_{c_j d_i}} \max_{s \neq i} \tilde{m}_{d_s c_j}^{q-1}(\mathbf{y}_j[s] = 1) \times \prod_{d_k \in \mathcal{D} \setminus \{d_i, d_s\}} \tilde{m}_{d_k c_j}^{q-1}(\mathbf{y}_j[k] = 0) \quad (29)$$

$$m_{c_j d_i}^q(\mathbf{x}_i[j] = 1) = \frac{1}{z_{c_j d_i}} \prod_{d_k \in \mathcal{D} \setminus d_i} \tilde{m}_{d_k c_j}^{q-1}(\mathbf{y}_j[k] = 0) \quad (30)$$

Since only one element can be set to 1 in the customer node variable, in Eq. (29) the message where  $\mathbf{y}_j[s] = 1$  appears only for one depot node  $d_s$  and maximization is performed over that index  $s$ , excluding the receiving node's index  $i$ . In contrast, in Eq. (30), since the  $i$ th element of  $\mathbf{y}_j$  is set to 1, only the messages corresponding to  $\mathbf{y}_j[k] = 0$  are multiplied. By diving Eqs. (29) and (30) by  $\prod_{d_k \in \mathcal{D} \setminus d_i} \tilde{m}_{d_k c_j}^{q-1}(\mathbf{y}_j[k] = 0)$  and applying the normalization, the simplified customer-to-depot message is derived as below, where  $s^*$  maximizes  $\tilde{m}_{d_s^* c_j}^{q-1}(\mathbf{y}_j[s^*] = 1)$ .

$$m_{c_j d_i}^q(\mathbf{x}_i[j] = 0) = \frac{\tilde{m}_{d_{s^*} c_j}^{q-1}(\mathbf{y}_j[s^*] = 1)}{\tilde{m}_{d_{s^*} c_j}^{q-1}(\mathbf{y}_j[s^*] = 0) + \tilde{m}_{d_{s^*} c_j}^{q-1}(\mathbf{y}_j[s^*] = 1)} \quad (31)$$

$$m_{c_j d_i}^q(\mathbf{x}_i[j] = 1) = \frac{\tilde{m}_{d_{s^*} c_j}^{q-1}(\mathbf{y}_j[s^*] = 0)}{\tilde{m}_{d_{s^*} c_j}^{q-1}(\mathbf{y}_j[s^*] = 0) + \tilde{m}_{d_{s^*} c_j}^{q-1}(\mathbf{y}_j[s^*] = 1)} \quad (32)$$

### 3.3. Algorithm structure

The HMDVRP has specific parameters related to depots, vehicles, and customers. For depots, the parameters include their locations and the types of vehicles they possess. The primary parameters for vehicles are their capacity and locations, while the parameters for customers are their demands. The cost function for the path between a depot node and a customer node, or between customer nodes, typically utilizes the distance between two points. However, when considering vehicle heterogeneity, this function can be more complex. The cost may vary for the same path depending on the type of vehicle.

The algorithm (refer Algorithm 1) begins by initializing the message values ( $\tilde{m}_{d_i c_j}^0$  and  $\tilde{m}_{c_j d_i}^0$ ) to 0.5, representing the initial probabilities of any depot choosing a customer and any customer choosing a depot as 50%. The algorithm then iterates through a predetermined number ( $N_{iter}$ ) of message calculations. In each iteration, a representative set of routes  $\mathbf{R}_i$  is generated for each depot, and the messages ( $\tilde{m}_{d_i c_j}^q$  and  $\tilde{m}_{c_j d_i}^q$ ) are calculated for each depot–customer pair. The sets of routes  $\mathbf{P}_i^{j,0}$  and  $\mathbf{P}_i^{j,1}$  in (27) and (28) differ for every depot–customer pair. The method for calculating  $\mathbf{R}_i$  as a substitute for  $\mathbf{P}_i$  and using it to compute the messages from depots to customers,  $\tilde{m}_{d_i c_j}^q$ , is described in detail in Section 3.4. These messages are averaged using a buffer to

### Algorithm 1 Centralized Algorithm Structure of AMP-R

```

1: Initialize parameters about depot, vehicle, customer.
2: Initialize messages  $\tilde{m}_{d_i c_j}^0$  and  $\tilde{m}_{c_j d_i}^0$  with  $\frac{1}{2}$ .
3: for  $q = 1:N_{iter}$  do
4:   for each  $d_i \in \mathcal{D}$  do
5:     Build  $\mathbf{R}_i$ .
6:     for each  $c_j \in \mathcal{C}$  do
7:       Calculate  $\tilde{m}_{d_i c_j}^q(\mathbf{y}_j[i])$  using Eqs. (40), (41), and (19).
8:       Calculate  $\tilde{m}_{c_j d_i}^q(\mathbf{x}_i[j])$  using Eqs. (31), (32), and (20).
9:       if  $c_j \in \mathbf{R}_i$  then  $\hat{\mathbf{x}}_i[j] = 1$ , else  $\hat{\mathbf{x}}_i[j] = 0$ .
10:    end for
11:     $L_{d_i}^{rem} = L_{d_i} - \sum_j l(c_j) \hat{\mathbf{x}}_i[j]$ .
12:  end for
13: end for
14: Get  $\hat{\mathbf{x}}_i, \forall d_i \in \mathcal{D}$  from Refinement (Algorithm 2) with  $\hat{\mathbf{x}}_i, L_{d_i}^{rem}$ , and  $\tilde{m}_{d_i c_j}^{N_{iter}}$ .
15: for each  $d_i \in \mathcal{D}$  do
16:   Obtain  $\hat{\mathbf{P}}_i$  by solving the CVRP for  $c_j$ s satisfying  $\hat{\mathbf{x}}_i[j] = 1$ .
17: end for

```

enhance convergence, resulting in  $\tilde{m}_{d_i c_j}^q$ . The messages from customers to depots are calculated and averaged in a similar manner. The binary variable  $\hat{\mathbf{x}}_i[j]$  indicates whether a customer is assigned to a depot. If customer  $c_j$  is included in  $\mathbf{R}_i$ , then  $\hat{\mathbf{x}}_i[j]$  becomes one. To satisfy the constraints of the HMDVRP, the conditions  $\sum_i \hat{\mathbf{x}}_i[j] = 1$  and  $\sum_j \hat{\mathbf{x}}_i[j] = 1$  must be met; these constraints ensure that each customer is served by a vehicle from a specific depot. However, these are not guaranteed to be met by message calculation. Therefore, a Refinement step is necessary to derive a solution that satisfies the constraints of the HMDVRP. Capacity constraints must also be satisfied, making it necessary to monitor the sum of the remaining capacities of all vehicles belonging to depot  $d_i$ . The total remaining capacity for depot  $d_i$  is represented as  $L_{d_i}^{rem}$ .

### Algorithm 2 Refinement Step.

```

1:  $\hat{\mathbf{x}}_i[j] = 0, M(i, j) = \tilde{m}_{d_i c_j}^{N_{iter}}(\mathbf{y}_j[i]), \forall d_i \in \mathcal{D}, \forall c_j \in \mathcal{C}$ .
2: for each  $c_j \in \mathcal{C}$  do
3:   if  $\sum_i \hat{\mathbf{x}}_i[j] > 1$  then
4:      $\hat{\mathbf{x}}_{i^*}[j] = 1$  where  $i^* = \arg\max_i M(i, j) \hat{\mathbf{x}}_i[j]$ .
5:      $L_{d_{i^*}}^{rem} = L_{d_{i^*}}^{rem} + l(c_j), \forall i : \hat{\mathbf{x}}_i[j] = 1, i \neq i^*$ .
6:      $M(i, j) = 0, \forall i$ .
7:   end if
8: end for
9: while  $M(i, j) \neq 0, \exists i, j$  do
10:  ( $i^*, j^*$ ) =  $\arg\max_{i,j} M(i, j)$ 
11:  if  $L_{d_{i^*}}^{rem} > l(c_{j^*}) n_{v, d_{i^*}}^{c_j}$  then
12:     $\hat{\mathbf{x}}_{i^*}[j^*] = 1, L_{d_{i^*}}^{rem} = L_{d_{i^*}}^{rem} - l(c_{j^*})$ .
13:     $M(i, j^*) = 0, \forall i$ .
14:  else
15:     $M(i^*, j^*) = 0$ .
16:  end if
17: end while

```

The Refinement step (Algorithm 2) uses  $\hat{\mathbf{x}}_i[j]$  to calculate  $\hat{\mathbf{x}}_i[j]$  that satisfies  $\sum_i \hat{\mathbf{x}}_i[j] = 1$  and  $\sum_j \hat{\mathbf{x}}_i[j] = 1$ . If  $\sum_i \hat{\mathbf{x}}_i[j]$  is equal to or

greater than 1, only the depot  $d_i^*$  with the highest message value takes the customer  $j$ , i.e.,  $\tilde{x}_{i^*}[j] = 1$ . If  $\tilde{x}_i[j] = 1$  but  $\tilde{x}_{i^*}[j] = 0$ , this indicates that, although a depot may initially consider itself the best candidate to serve a particular customer  $c_j$  based on its own calculations, comparing messages with neighboring depots reveals that it would be more efficient for another depot to serve  $c_j$ . Therefore, the remaining capacity of the depot  $d_i$  is increased by the demand of the customer  $c_j$ , and this amount of demand is added to  $L_{d_i}^{\text{rem}}$ .

After this step, unassigned customers must be allocated to appropriate depots using message values, for which a matrix  $M$  is defined to contain these values.  $M(i, j)$  is initially set to the message value from the last iteration,  $\tilde{m}_{d_i c_j}^{\text{Niter}}(\mathbf{y}_j[i])$ . If a customer  $c_j$  is already assigned elsewhere,  $M(i, j)$  is set to zero for all depots. Once the matrix is built, the pair  $(i^*, j^*)$  with the maximum message value is identified. If the remaining capacity of the depot is sufficient, the customer is assigned to that depot. This results in  $\tilde{x}_{j^*}[j^*] = 1$ . As customer  $c_{j^*}$  is newly assigned to depot  $d_{i^*}$ ,  $L_{d_{i^*}}^{\text{rem}}$  is decreased.  $M(k, j^*)$ s for all  $k$ s are set to zero to prevent this customer from being selected again. If the capacity is insufficient, only  $M(i^*, j^*)$  is set to zero, allowing  $j^*$  to be selected by another depot.

When considering the remaining capacity, it is essential that the available capacity exceeds the product of the customer's demand and the number of vehicles at the depot. Even if the capacity appears sufficient, an even distribution among vehicles may pose a risk that an assigned customer is not included in a route. To mitigate this risk, the remaining capacity is evaluated comprehensively. Therefore, for capacitated and limited problems, the sum of vehicle capacities must be significantly higher than the sum of all customer demands for a solution to be feasible. Once the allocation is completed without conflicts,  $\tilde{x}$  is finalized, allowing determination of which customers are assigned to each depot. Finding the optimal route for each depot then becomes a simple CVRP, which can be solved using the LKH (Lin–Kernighan–Helsgaun) heuristic [45,46].

### 3.4. Message approximation techniques

To send a message from a depot node to each customer node, one must solve either the SV-CVRP or SV-TSP for the cases where each customer is included and excluded, as shown in Eqs. (27) and (28). Repeating this process for every receiving customer node is highly inefficient. Therefore, by solving the SV-CVRP for all customers once, and then partially modifying the resulting route by adding or removing specific customers, the approximate messages from a single depot can be quickly calculated. The representative route  $\mathbf{R}_i$  appearing in Algorithm 1 serves as an approximate solution to  $\hat{P}_i^q$  in Eq. (33), which is the optimal route of the SV-CVRP for all customers.

$$\hat{x}_i^q, \hat{P}_i^q = \underset{\mathbf{x}_i, \mathbf{P}_i}{\operatorname{argmax}} \eta(\mathbf{P}_i) e^{-w(\mathbf{P}_i)} \prod_{c_k \in \mathbf{P}_i} \frac{\tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 1)}{\tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 0)} \quad (33)$$

Note that Eq. (33) is derived by applying the same procedure used in Eqs. (27) and (28) to the equation for the MAP state estimate, i.e., Eq. (22). Thus, the proposed AMP-R algorithm first computes an approximate MAP state estimate that maximizes the belief in each iteration, then partially adjusts this estimate to calculate approximate messages for each customer. The representative route  $\mathbf{R}_i$  is constructed by greedily selecting customers one by one until the vehicle reaches its capacity, making it applicable to all types of the HMDVRP shown in Table 1. However, when addressing uncapacitated problems where the vehicles in each depot are homogeneous and path costs satisfy the triangular inequality, the computation of  $\mathbf{R}_i$  reduces to the SV-TSP. For this case, the SV-TSP is converted to an asymmetric TSP (ATSP), allowing the application of highly effective heuristics like LKH.

#### 3.4.1. Greedy method to obtain $\mathbf{R}_i$

This section describes the proposed greedy selection heuristic for finding solutions to the SV-CVRP or SV-TSP. The procedure of the greedy selection is described in Algorithm 3. The representative routes  $\mathbf{R}_i$ , precisely a set of routes, is incrementally constructed by adding customers one by one. Since the goal is to find  $\mathbf{R}_i$  that approximates  $\hat{P}_i^q$ , which maximizes the right-hand side of Eq. (33), each new customer added to  $\mathbf{R}_i$  is chosen greedily to yield the largest increase in the right-hand side of Eq. (33). This process is repeated until the capacity of a single vehicle from the depot  $d_i$  is filled, after which a new vehicle is selected, and routes are formed until the entire fleet of  $d_i$  is deployed.

When customer  $c_j$  is added to  $\mathbf{R}_i$ , the increase in the operand of the max operation in Eq. (33), denoted as  $r_{ij}(\mathbf{R}_i)$ , is estimated as shown in Eq. (34).

$$r_{ij}(\mathbf{R}_i, c_j) = \frac{e^{-w(\mathbf{R}_i \cup \{c_j\})} \tilde{m}_{c_j d_i}^{q-1}(\mathbf{x}_i[j] = 1)}{e^{-w(\mathbf{R}_i)} \tilde{m}_{c_j d_i}^{q-1}(\mathbf{x}_i[j] = 0)} \quad (34)$$

$w(\mathbf{R}_i \cup \{c_j\})$  represents the total path cost of the route after adding  $c_j$  to  $\mathbf{R}_i$ . It is chosen as the minimal increase in path cost when  $c_j$  is inserted between the nodes already included in  $\mathbf{R}_i$ . Each time, the customer with the highest  $r_{ij}(\mathbf{R}_i, c_j)$  is selected and added to  $\mathbf{R}_i$ . If  $\max_{c_j \notin \mathbf{R}_i} r_{ij}(\mathbf{R}_i, c_j) < 1$ , then adding any customer does not further increase the right-hand side of Eq. (33), so the process terminates even if capacity and vehicles remain, and  $\mathbf{R}_i$  is returned.

#### Algorithm 3 Build $\mathbf{R}_i$ using Greedy Method.

---

```

1: while True do
2:    $\mathbf{C}_{\text{rem}} = [k : \forall c_k \in \mathbf{C}, \forall c_k \notin \mathbf{R}_i, L_{d_i}^{\text{rem}} > l(c_k)]$ 
3:   if  $\mathbf{C}_{\text{rem}} = \emptyset$  then break;
4:    $k^* = \operatorname{argmax}_{k \in \mathbf{C}_{\text{rem}}} r_{ik}(\mathbf{R}_i, c_k)$ 
5:   if  $r_{ik^*}(\mathbf{R}_i, c_{k^*}) > 1$  then
6:      $\mathbf{R}_i = \mathbf{R}_i \cup \{c_{k^*}\}$ 
7:   else
8:     break;
9:   end if
10: end while

```

---

#### 3.4.2. Transformation of SV-TSP to ATSP

For the uncapacitated HMDVRP, if certain conditions are met, Eq. (33) becomes equivalent to the SV-TSP. In this case, by modifying the graph, the problem can be transformed into an ATSP, allowing the application of various TSP solutions. To clarify how the path cost of the SV-TSP is defined, the logarithm of the right-hand side of Eq. (33) is taken and multiplied by  $-1$  under the assumption that the capacity constraint holds.

$$\hat{x}_i^q, \hat{P}_i^q = \underset{\mathbf{x}_i, \mathbf{P}_i}{\operatorname{argmin}} w(\mathbf{P}_i) - \sum_{c_k \in \mathbf{P}_i} \ln \left( \tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 1) - \tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 0) \right) \quad (35)$$

$w(\mathbf{P}_i)$  denotes the sum of all path costs in  $\mathbf{P}_i$ , and the second term originates from the messages received from each customer node included in  $\mathbf{P}_i$ . The latter term is added only for nodes among the given customers that have been selected to visit. Therefore,  $\ln \tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 1) - \ln \tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 0)$  can be regarded as the benefit of visiting the corresponding customer node  $c_k$ . The path cost in the SV-TSP is obtained by subtracting  $\ln \tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 1) - \ln \tilde{m}_{c_k d_i}^{q-1}(\mathbf{x}_i[k] = 0)$  from the original path cost (referred to as edge cost in a general graph) for each customer. For this reason, unlike the standard TSP, the SV-TSP allows paths with negative costs, and a route satisfying Eq. (35) can be formed, which visits only a subset of the given customers.

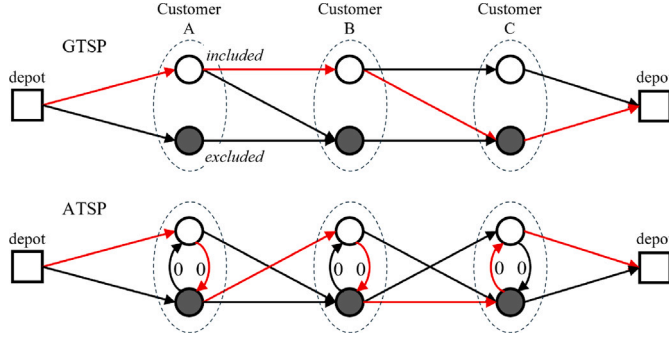
To transform the SV-TSP to the ATSP, it should first be transformed into a problem known as the generalized TSP (GTSP) or one-in-a-set TSP. In the GTSP, multiple groups of nodes are given, and the goal is to find a Hamiltonian cycle that visits exactly one node in each group. To convert the SV-TSP to the GTSP, each customer node is divided into



**Table 2**  
Edge costs in GTSP and ATSP for transformation from SV-TSP.

Edge		GTSP cost	ATSP cost
Tail ( $c_j$ or $d_i$ )	Head ( $c_k$ or $d_i$ )		
Depot	<i>included</i>	$e(d_i, c_k, v) - \ln m_{c_k d_i}(1) + \ln m_{c_k d_i}(0)$	$e(d_i, c_k, v) - \ln m_{c_k d_i}(1) + \ln m_{c_k d_i}(0) + M$
	<i>excluded</i>	0	$M$
<i>included</i>	Depot	$e(c_j, d_i, v)$	0
<i>excluded</i>	Depot	0	$e(c_j, d_i, v)$
<i>included</i>	<i>included</i>	$e(c_j, c_k, v) - \ln m_{c_k d_i}(1) + \ln m_{c_k d_i}(0)$	–
<i>excluded</i>	<i>included</i>	–	$e(c_j, c_k, v) - \ln m_{c_k d_i}(1) + \ln m_{c_k d_i}(0) + M$
<i>included</i>	<i>excluded</i>	$e(c_j, d_i, v)$	$M$
<i>excluded</i>	<i>excluded</i>	0	$e(c_j, d_i, v) + M$

For conciseness,  $m_{c_k d_i}(a)$  denotes  $m_{c_k d_i}(\mathbf{x}_i[k] = a)$ .



**Fig. 3.** Example tours comparing GTSP and ATSP: the edges corresponding to the SV-TSP solution (Depot - A - B - Depot) are marked in red.

an *included* node and an *excluded* node, with both nodes belonging to the same group. Therefore, a GTSP solution selects either an *included* node or an *excluded* node for each customer and constructs a route that returns to the depot. Visiting the *included* node implies that the corresponding customer is part of the representative route  $R_i$  in the SV-TSP, while selecting the *excluded* node indicates otherwise. Every path cost toward a customer node in the SV-TSP is assigned to the path toward the *included* node of the customer in the GTSP. Path costs directed toward *excluded* nodes, as well as those from *excluded* nodes back to the depot, are all set to zero. Additionally, to ensure that the GTSP solution translates to a valid solution for the SV-TSP, the selected *included* nodes in the GTSP solution must be visited consecutively. Thus, all paths leading from an *excluded* node to an *included* node are removed, ensuring that once an *excluded* node is visited, only *excluded* nodes can be visited in all subsequent groups. Finally, the cost of returning from a customer node to the depot in the SV-TSP is copied to both the path from the corresponding *included* node to the depot and the paths from the *included* node to the *excluded* nodes in the other groups. Table 2 summarizes the path costs for each path type in the GTSP, and an example graph is shown in Fig. 3.

To transform the GTSP to the ATSP, the Noon-Bean transformation [47] is applied. First, connect the *included* and *excluded* nodes within each customer group with paths that have a cost of zero. Next, swap the tails (starting nodes) of paths originating from each group. In other words, change paths originating from the *included* node to start from the *excluded* node in the same group, and vice versa for paths with the *excluded* node as the tail. Finally, add a large number  $M$  to the cost of paths entering each group. This ensures that the ATSP solution visits each group's nodes only once, with  $M$  set to be larger than the sum of all path costs in the original graph. Since the depot node remains unsplit and  $M$  is added to all paths originating from the depot, there is no need to add  $M$  to the paths returning to the depot. The resulting costs assigned in the ATSP are summarized in the last column of Table 2.

By transforming the SV-TSP to the ATSP in this way, the ATSP solution forms a Hamiltonian cycle, as shown in Fig. 3, visiting all *included* and *excluded* nodes and returning to the depot. The number of times paths entering groups are chosen equals the number of customers. When a group is visited via its *included* node, it leaves through the *excluded* node of the same group via a zero-cost path, indicating that the corresponding customer is selected in the SV-TSP. Also, once a group is first entered through an *excluded* node, the solution always exits through the *included* node of that group. From that point on, only paths leading into each group's *excluded* node can be selected, representing groups corresponding to customers not chosen in the SV-TSP. The overall procedure for solving SV-TSP by converting it into an ATSP to obtain  $R$  in the uncapacitated HMDVRP case is summarized in Algorithm 4.

#### Algorithm 4 Solving SV-TSP using LKH Solver.

- 1: **Step 1: Copy Customer Nodes**
- 2: Duplicate  $c$  as  $c_{in}$  (*included*) and  $c_{ex}$  (*excluded*)  $\forall c \in C$
- 3: **Step 2: Connect Depot to Customers**
- 4: Create directed edges from  $d_i$  to  $c_{in}$  and  $c_{ex}$  and directed edges from  $c_{in}$  and  $c_{ex}$  to  $d_i$   $\forall c \in C$
- 5: **Step 3: Connect Customer Nodes**
- 6: Create directed edges between:  $(c_{k,ex}, c_{j,ex}), (c_{j,in}, c_{k,ex}), (c_{j,ex}, c_{k,in})$   $\forall c_j, c_k \in C$
- 7: Create edges between  $c_{j,in}$  and  $c_{j,ex}$   $\forall c_j \in C$
- 8: **Step 4: Assign Edge Costs based on Table 2**
- 9: **Step 5: Solve ATSP using LKH Solver**
- 10: **Step 6: Construct Final Path**
- 11: **for** each customer node in ATSP solution **do**
- 12:   **if**  $c_{j,in}$  is visited before  $c_{j,ex}$  **then**
- 13:      $R_i = R_i \cup \{c_j\}$
- 14:   **end if**
- 15: **end for**

#### 3.4.3. Calculating messages with $R_i$

Once the representative route  $R_i$  is derived, it is used to calculate the messages to be sent to each customer node adjacent to the depot  $d_i$  node in the graphical model. The messages in Eqs. (27) and (28) are defined with the optimal assignment and route, either necessarily excluding or including the receiving node  $c_j$ , respectively. Since  $R_i$  is an approximate solution of the subset-visit VRP-like problems for given customers, a certain receiving node  $c_j$  may or may not be included. If  $c_j \in R_i$ ,  $R_i$  becomes an approximate solution of  $P_i^{j,1}$  that maximizes the right side of Eq. (28), enabling direct calculation of  $m_{d_i, c_j}^q(y_j[i] = 1)$ . However, to compute  $m_{d_i, c_j}^q(y_j[i] = 0)$  in Eq. (27), an approximate solution of  $P_i^{j,0}$  that excludes  $c_j$  is needed, which can be obtained by removing  $c_j$  from  $R_i$ . On the other hand, when  $c_j \notin R_i$ ,  $R_i$  can be used directly to compute  $m_{d_i, c_j}^q(y_j[i] = 0)$ , while  $m_{d_i, c_j}^q(y_j[i] = 1)$  requires adding  $c_j$  to  $R_i$ .

Let the approximate solutions of  $P_i^{j,0}$  and  $P_i^{j,1}$  that maximize Eqs. (27) and (28) be denoted by  $R_i^{j,0}$  and  $R_i^{j,1}$ , respectively. For

problems without capacity constraints, every customer  $c_k$  satisfying  $r_{ik}(\mathbf{R}_i, c_k) \geq 1$  has already been added when calculating  $\mathbf{R}_i$  since adding the customer increases the right hand side of Eq. (33). Thus,  $\mathbf{R}_i^{j,0}$  and  $\mathbf{R}_i^{j,1}$  can be directly obtained by simply removing  $c_j$  from or adding it to  $\mathbf{R}_i$ . In contrast, for capacity-constrained cases, other customers  $c_k$  may exist that can be added within the remaining capacity after excluding  $c_j$  from  $\mathbf{R}_i$  and that also maximize  $r_{ik}(\mathbf{R}_i - \{c_k\})$ . Let the set of these customers be  $S_{\text{add}}$ .  $\mathbf{R}_i^{j,0}$  is constructed by inserting the customers in  $S_{\text{add}}$  into the route  $\mathbf{R}_i - \{c_j\}$  at positions that least increase the total path cost, similar to the approach in Section 3.4.1. Furthermore, when adding  $c_j$  to obtain  $\mathbf{R}_i^{j,1}$ , some customers already included in  $\mathbf{R}_i$  may need to be removed due to capacity limitations. Let  $S_{\text{del}}$  denote the set of customers removed in greedy order of lowest  $r_{ik}$  values until sufficient residual capacity for  $c_j$  is secured. Eqs. (36) and (37) summarize the definitions of  $\mathbf{R}_i^{j,0}$  and  $\mathbf{R}_i^{j,1}$  using this approach.

$$\mathbf{R}_i^{j,0} = \begin{cases} (\mathbf{R}_i - \{c_j\}) \cup S_{\text{add}}, & c_j \in \mathbf{R}_i \\ \mathbf{R}_i, & c_j \notin \mathbf{R}_i \end{cases} \quad (36)$$

$$\mathbf{R}_i^{j,1} = \begin{cases} \mathbf{R}_i, & c_j \in \mathbf{R}_i \\ (\mathbf{R}_i \cup \{c_j\}) - S_{\text{del}}, & c_j \notin \mathbf{R}_i \end{cases} \quad (37)$$

The union signifies the greedy insertion of customers, while the difference represents the removal of elements. Eqs. (27) and (28) can now be reformulated as follows using  $\mathbf{R}_i^{j,0}$  and  $\mathbf{R}_i^{j,1}$ .

$$m_{d_i c_j}^q(y_j[i] = 0) = \frac{1}{z_{d_i c_j}} e^{-w(\mathbf{R}_i^{j,0})} \prod_{c_k \in \mathbf{R}_i^{j,0}} \frac{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1)}{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0)} \quad (38)$$

$$m_{d_i c_j}^q(y_j[i] = 1) = \frac{1}{z_{d_i c_j}} e^{-w(\mathbf{R}_i^{j,1})} \prod_{c_k \in (\mathbf{R}_i^{j,1} - \{c_j\})} \frac{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1)}{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0)} \quad (39)$$

Since the representative route  $\mathbf{R}_i$  is an approximation of the right-hand side of Eq. (33),  $\mathbf{R}_i^{j,0}$  and  $\mathbf{R}_i^{j,1}$  are also treated as approximate solutions, thus removing the max operation. In Eq. (39), the product of the message ratios is taken for customers in  $\mathbf{R}_i^{j,1}$  excluding the receiving node  $c_j$ . Using Eqs. (27) and (28), it can be shown that regardless of whether  $c_j$  is included in  $\mathbf{R}_i$ , the customers in  $\mathbf{R}_i^{j,1} - \{c_j\}$  are always included in  $\mathbf{R}_i^{j,0}$ . Furthermore, the difference between the customers in the two routes equals  $S_{\text{add}}$  when  $c_j \in \mathbf{R}_i$  and  $S_{\text{del}}$  when  $c_j \notin \mathbf{R}_i$ . Therefore, by dividing both equations by the product of message ratios for customers in  $\mathbf{R}_i^{j,1} - \{c_j\}$ , the message equations can be simplified as follows:  $S$  denotes  $S_{\text{add}}$  or  $S_{\text{del}}$ , depending on whether  $c_j$  belongs to  $\mathbf{R}_i$ .

$$m_{d_i c_j}^q(y_j[i] = 0) = \frac{1}{z_{d_i c_j}} e^{-w(\mathbf{R}_i^{j,0})} \prod_{c_k \in S} \frac{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 1)}{\tilde{m}_{c_k d_i}^{q-1}(x_i[k] = 0)} \quad (40)$$

$$m_{d_i c_j}^q(y_j[i] = 1) = \frac{1}{z_{d_i c_j}} e^{-w(\mathbf{R}_i^{j,1})} \quad (41)$$

### 3.5. Decentralized implementation

In this section, the decentralized version of the AMP-R algorithm is explained. The AMP-R exchanges messages between nodes in the graphical model of the HMDVRP, which is represented in a bipartite graph consisting of depot and customer nodes. While each depot can act as a computational agent, each customer cannot. Therefore, all message computations are processed at the depots. The depots calculate both  $m_{d_i c_j}$ , the messages sent from the depots to the customers, and  $m_{c_j d_i}$ , the messages sent from the customers to the depots (refer Fig. 4). Exchanging these messages simulates the passing of messages within the overall graphical model.

The decentralized approach presented in Algorithm 5 closely resembles the centralized approach described in Algorithm 1. Each depot individually generates its own routes  $\tilde{\mathbf{P}}_i$  using the decentralized algorithm, which includes the computations and information exchanges

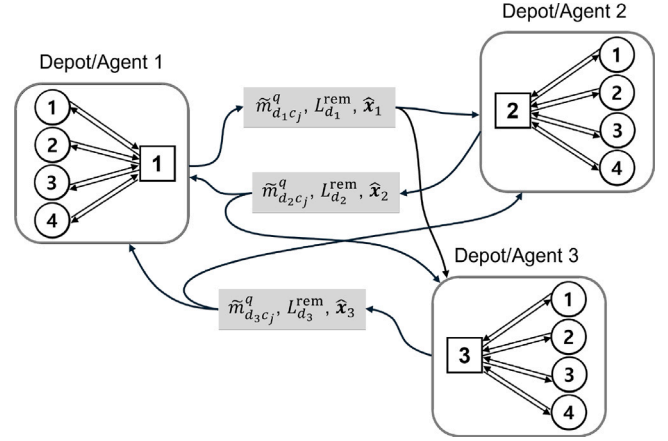


Fig. 4. Example of the distributed computation and message exchange in the AMP algorithm: within each depot, all messages between the depot and customer nodes are computed and exchanged; then, depot-to-customer messages  $\tilde{m}_{d_i c_j}^q(y_j[i])$ , remaining capacity  $L_{d_i}^{rem}$ , and assignment  $\hat{x}_i$  are shared with other depots.

### Algorithm 5 Decentralized Algorithm Structure of AMP-R for Depot $d_i$

- 1: Initialize parameters about depot, vehicle, customer.
- 2: Initialize messages  $\tilde{m}_{d_i c_j}^0$  and  $\tilde{m}_{c_j d_i}^0$  with  $\frac{1}{2}$ .
- 3: **for**  $q = 1:N_{\text{iter}}$  **do**
- 4:   Build  $\mathbf{R}_i$ .
- 5:   **for each**  $c_j \in C$  **do**
- 6:     Calculate  $\tilde{m}_{d_i c_j}^q(y_j[i])$  using Eqs. (40), (41), and (19).
- 7:     Calculate  $\tilde{m}_{c_j d_i}^q(x_i[j])$  using Eqs. (31), (32), and (20).
- 8:     **if**  $c_j \in \mathbf{R}_i$  **then**  $\hat{x}_i[j] = 1$ , **else**  $\hat{x}_i[j] = 0$ .
- 9:   **end for**
- 10:    $L_{d_i}^{rem} = L_{d_i} - \sum_j l(c_j) \hat{x}_i[j]$ .
- 11:   Save  $\tilde{m}_{c_j d_i}^q(x_i[j])$ ,  $\forall c_j \in C$
- 12:   Broadcast  $\tilde{m}_{d_i c_j}^q(y_j[i])$ ,  $L_{d_i}^{rem}$  and  $\hat{x}_i$ ,  $\forall c_j \in C$
- 13:   Receive  $\tilde{m}_{d_i c_j}^q(y_j[k])$ ,  $L_{d_i}^{rem}$  and  $\hat{x}_k$ ,  $\forall d_k \in D \setminus d_i$ ,  $\forall c_j \in C$
- 14: **end for**
- 15: Get  $\tilde{\mathbf{x}}_i$  from Refinement (Algorithm 2) with  $\hat{x}_i$ ,  $L_{d_i}^{rem}$ , and  $\tilde{m}_{d_i c_j}^{N_{\text{iter}}}$ ,  $\forall d_i \in D$ ,  $\forall c_j \in C$ .
- 16: Obtain  $\tilde{\mathbf{P}}_i$  by solving the CVRP for  $c_j$ s satisfying  $\tilde{x}_i[j] = 1$ .

of the depots. Initially, each depot receives the same information about other depots, vehicles, and customers. All outgoing and incoming messages for a specific depot  $d_i$  are initialized to 0.5. Each depot then performs the following steps for a predefined number  $N_{\text{iter}}$  of iterations. First, each depot  $d_i$  computes routes and messages in the same way as the centralized AMP-R algorithm, but only the messages sent to and from the depot  $d_i$  are computed. Depots then broadcast and share depot-to-customer messages  $\tilde{m}_{d_i c_j}^q$ , remaining capacity  $L_{d_i}^{rem}$ , and the assignment result  $\hat{x}_i$ . The shared messages are used to compute customer-to-depot messages  $\tilde{m}_{c_j d_i}^q$  in the next step. After exchanging and calculating messages for  $N_{\text{iter}}$  iterations, each depot performs the same refinement process as in the centralized algorithm. Since all required information is shared, this produces identical results for all depots, ensuring no conflicts. With the allocated customers for which  $\tilde{x}_i[j] = 1$  after the refinement, the set of routes  $\tilde{\mathbf{P}}_i$  for each depot is derived by solving the CVRP independently.

### 4. Numerical simulation

This section describes the numerical simulations conducted to evaluate the performance of the proposed AMP-R algorithm and presents the corresponding results. Since the HMDVRP considered in this study

encompasses multiple variants, Section 4.1 first categorizes these variants and introduces the algorithms used for comparative evaluation in each case. Section 4.2 details the conditions for generating problem instances, and Section 4.3 presents the comparative evaluation results obtained through Monte Carlo simulations for each problem type. Finally, Section 4.4 analyzes the performance of the AMP-R algorithm on MDVRP benchmark instances to compare with other algorithms.

#### 4.1. Baseline

The algorithm presented in this paper can solve various variants of the HMDVRP in a decentralized manner. The criteria used to classify the problems shown in Table 1 are the capacity and the number of vehicles assigned to each depot. Taking into account heterogeneity, there are a total of eight possible cases based on three conditions: (1) uncapacitated and capacitated cases, (2) unlimited and limited number of vehicles, and (3) homogeneous and heterogeneous cases. However, as shown in Table 1, when the problem is uncapacitated and the path cost satisfies the triangle inequality, only one vehicle per depot is required, making it unnecessary to limit the number of vehicles in uncapacitated cases. Consequently, excluding the vehicle number limitation in the uncapacitated cases leaves only six types of problems. These are expressed using notation as *MDVRP-(u)*, *HMDVRP-(u)*, *MDVRP-(c)*, *HMDVRP-(c)*, *MDVRP-(c)-LV*, *HMDVRP-(c)-LV*, where “H” indicates that heterogeneity is considered. The suffix *(u)* signifies that capacity is disregarded, while the suffix *(c)* indicates that capacity is considered. Lastly, *LV* means that the number of the vehicle that each depot can hold is limited.

In this paper, two approaches are presented to approximate the messages. The first method, discussed in Section 3.4.1, uses a greedy method to compute the messages, which can be applied to both the SV-TSP and the SV-CVRP. This approach is named AMP-R-Greedy. The second method, described in Section 3.4.2, approximates the message by replacing the SV-TSP with the ATSP, deriving a near-optimal solution using the LKH heuristic. This approach is named AMP-R-LKH and is applicable only to the uncapacitated case of variants of the HMDVRP. Additionally, the message approximation approach can be selected at each iteration step. Therefore, it is possible to initially apply the greedy method and then switch to the LKH in later steps. This hybrid approach is named AMP-R-Mix. To evaluate the performance of the AMP-R algorithm, additional benchmarks are necessary. One simple benchmark can be the greedy algorithm applied to solve the MDVRP. The greedy algorithm used for the HMDVRP variants is similar to the heuristic in [48], using a constructive approach that repeatedly selects the best customer–vehicle pair, followed by a 2-opt heuristic to improve each vehicle’s route. For an optimal solution, the problem modeled in Section 2.1 can be solved using the Gurobi solver. Additionally, in uncapacitated problems, the MDVRP can be transformed into a GTSP, which can then be converted into an ATSP using the Noon-Bean transformation [47].

Thus, for *MDVRP-(u)* and *HMDVRP-(u)*, the performance of AMP-R-Greedy, AMP-R-LKH, AMP-R-Mix, the greedy method, the LKH heuristic, and the optimal solution obtained by the Gurobi solver are compared. For the remaining four problem types, the performance of AMP-R-Greedy, the greedy method, and the optimal solution from the Gurobi solver are compared. By comparing the results from the AMP-R algorithms with the benchmarks, the effectiveness of the AMP-R approaches in various types of the HMDVRP is demonstrated.

#### 4.2. Problem description

This paper presents a comprehensive evaluation of various algorithms for solving the HMDVRP under different constraints. To ensure a fair comparison, a consistent experimental setup was designed for all problem instances. The experiments were conducted on a 100 m  $\times$  100 m map with four depots and a varying number of customers (10,

**Table 3**

Simulation settings for vehicle capacities according to vehicle type and the number of customers.

Type of vehicle	$N_c = 10$	$N_c = 20$	$N_c = 40$	$N_c = 60$	$N_c = 80$
Type 1 (velocity : 1 m/s)	100	200	400	600	800
Type 2 (velocity : 1.2 m/s)	75	150	300	450	600
Type 3 (velocity : 0.5 m/s)	200	400	800	1200	1600

**Table 4**

Convergence ratio and steps for varying buffer sizes.

Buffer size	1	2	5	10	15	20
Convergence ratio (%)	2	40	76	96	94	86
Steps for convergence	2.0	14.6	31.8	56.3	66.2	77.1

20, 40, 60, and 80) randomly distributed within the map. Three types of vehicles were considered, each with different capacity and speed characteristics: high capacity with low speed, low capacity with high speed, and medium capacity with medium speed. These vehicles were randomly assigned to the depots, and each depot has the same type of vehicles. The vehicle capacity values for the varying test conditions are presented in Table 3, and customer demands were randomly generated between 20 and 30 units. This type of numerical simulation is capable of reflecting various real-world scenarios that may arise in logistics and transportation domains. The simulated scenarios emphasize realistic elements such as multiple types of vehicles with different capacities and speeds, their assigned depots, and variations in the number of customers.

For each variant and problem size, 50 random instances were generated. The performance of each algorithm was evaluated based on the average total travel time for all vehicles and the computation time across these instances. Due to the extensive computation time required, Gurobi was used to obtain solutions only for problems with up to 40 customers, with a 20 min time limit for each problem.

#### 4.3. Simulation results

The results of our experiments are presented in graphical form, illustrating the trade-offs between solution quality and computation time for each algorithm across different problem types and sizes. This comprehensive analysis allows for a thorough evaluation of the effectiveness and computational efficiency of the proposed decentralized approach in solving various HMDVRP instances.

In the *MDVRP-(u)* case, AMP-R-Greedy was applied to solve 50 instances of the HMDVRP while varying the buffer size. The number of steps for message passing was set to 200. Table 4 shows the convergence ratio and the number of steps required to reach convergence. Convergence was determined when the change in the message value was less than  $10^{-6}$ . The convergence ratio was calculated by dividing the number of cases that converged within 200 steps by the total number of test instances. The *steps to convergence* indicates the step count at which convergence was achieved. From Table 4, a buffer size of 10 demonstrated the highest convergence ratio; thus, the buffer size was set to 10, and the number of message exchange steps was reduced to 100.

The simulation was carried out using the decentralized version of the AMP algorithm, but all computations were performed on a single computer. Therefore, the reported computation time below can be regarded as the total calculation time used by all depots. While actual implementation would need to account for communication delays, it is expected that the computation time could be significantly reduced, roughly proportional to the number of depots.

Figs. 5 through 10 depict the results of the comparative numerical tests for the following cases, respectively: *MDVRP-(u)*, *HMDVRP-(u)*, *MDVRP-(c)*, *HMDVRP-(c)*, *MDVRP-(c)-LV*, *HMDVRP-(c)-LV*. On the abscissa of the graphs, the number of depots and customers is represented;

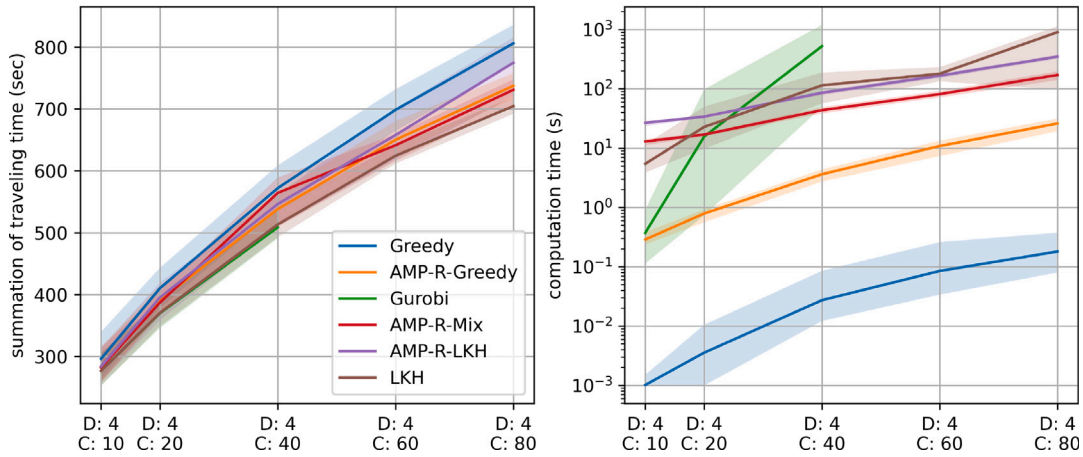
**MDVRP-(u)(Uncapacitated, Unlimited, Homogeneous)**

Fig. 5. Simulation Result : MDVRP-(u) (Uncapacitated, Unlimited, Homogeneous) case.

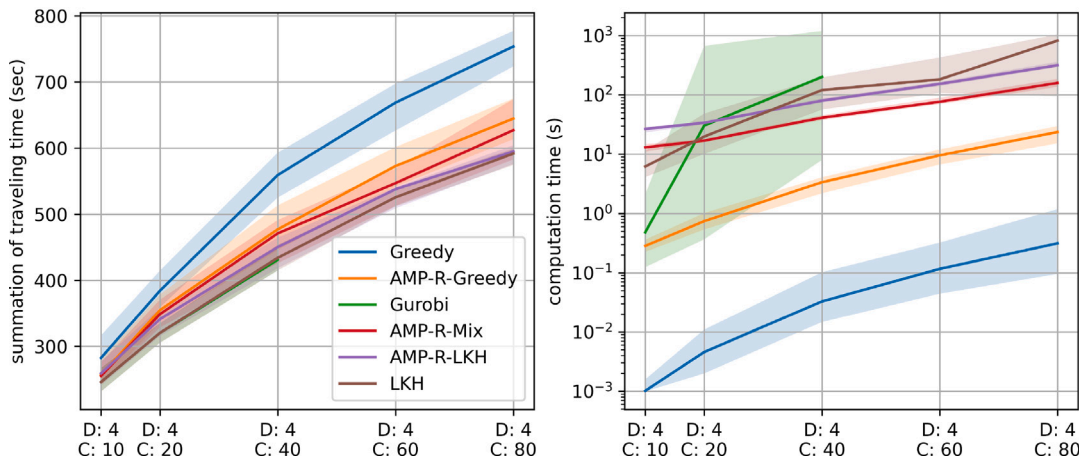
**HMDVRP-(u)(Uncapacitated, Unlimited, Heterogeneous)**

Fig. 6. Simulation Result : HMDVRP-(u) (Uncapacitated, Unlimited, Heterogeneous) case.

for instance,  $D : 4, C : 10$  indicates a scenario with 4 depots and 10 customers. The ordinate of the left graph in each figure represents the average total travel time from a solution, while the ordinate of the right one displays average computation time. The shaded region in the left graph illustrates the 25th to 75th percentile, while the shaded region in the right graph shows the 1st to 99th percentile. For the computation time of the greedy algorithm, extremely short times have been rounded up to 0.001 s to better align with the graph scale. From the graphs, it can be observed that although the Gurobi is significantly faster than the LKH for small-sized instances, such as with 4 depots and 10 customers, computation time sharply increases as the number of customers reaches 20. This increase makes it challenging to obtain solutions within the 20 min time limit for larger problem instances.

Fig. 5 shows the computation results for the MDVRP-(u), while Fig. 6 presents the results for the HMDVRP-(u) case. Overall, the AMP-R-Greedy, the AMP-R-LKH, and the AMP-R-Mix demonstrate superior performance compared to the greedy algorithm. In heterogeneous conditions, each depot has different types of vehicles, making the message values clearly distinguishable. This distinction helps aid message convergence, leading to better results. In contrast, for homogeneous cases, exact calculations result in more similar messages across depots, which can rather hinder convergence. When convergence is not achieved, adjustments during the refinement phase can ensure compliance with the MDVRP constraints, though the forced nature of these adjustments can

limit solution quality improvements. As a result, under homogeneous conditions, the AMP-R-Greedy or the AMP-R-Mix approaches typically yield better results, with the AMP-R-LKH rarely outperforming the others. However, in heterogeneous cases, the AMP-R-LKH consistently provides the best performance across all instances.

Fig. 7 shows the computational results for the MDVRP-(c), while Fig. 8 presents the results for the HMDVRP-(c). In the capacitated scenarios, the Gurobi solver required significantly more computation time, and except for the  $D : 4, C : 10$  problem, it was unable to produce valid solutions within the twenty minutes time limit. Consequently, for the  $D : 4, C : 40$  case, Gurobi's performance was worse than that of AMP-R-Greedy. However, AMP-R-Greedy successfully found solutions for the  $D : 4, C : 80$  problem in a relatively short time (a maximum of 1 min), achieving performance that surpassed the greedy method. Additionally, as in other cases, better results were obtained in heterogeneous settings compared to greedy solutions.

Fig. 9 presents the computational results for the MDVRP-(c)-LV, and Fig. 10 shows the results for the HMDVRP-(c)-LV. In general, the results in Fig. 9 are similar to those in Fig. 7, and the results in Fig. 10 resemble those in Fig. 8. This similarity arises from adjusting the vehicles' capacity, which ensured a consistent number of vehicles dispatched from each depot, even as the customer count increased. Once again, better results were obtained in heterogeneous settings compared to the greedy approach.



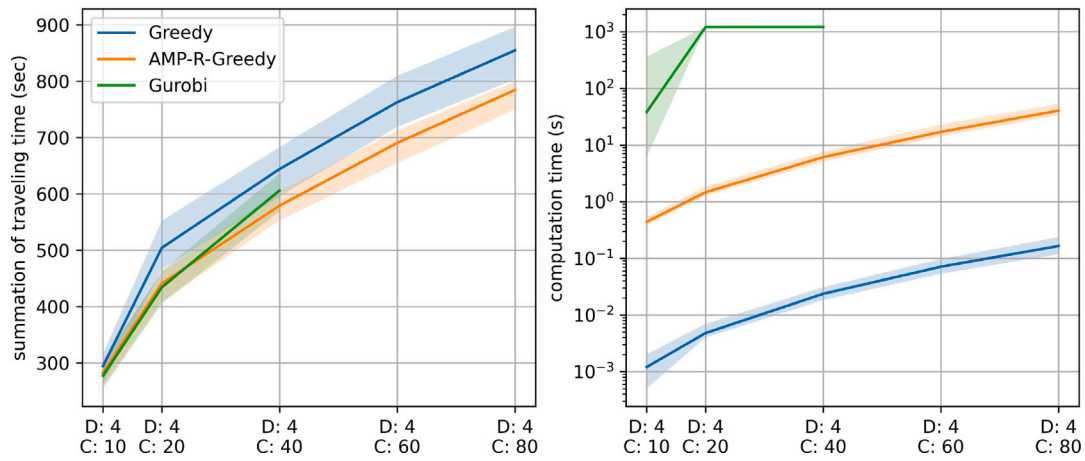
**MDVRP-(c)(Capacitated, Unlimited, Homogeneous)**

Fig. 7. Simulation Result : MDVRP-(c)(Capacitated, Unlimited, Homogeneous) case.

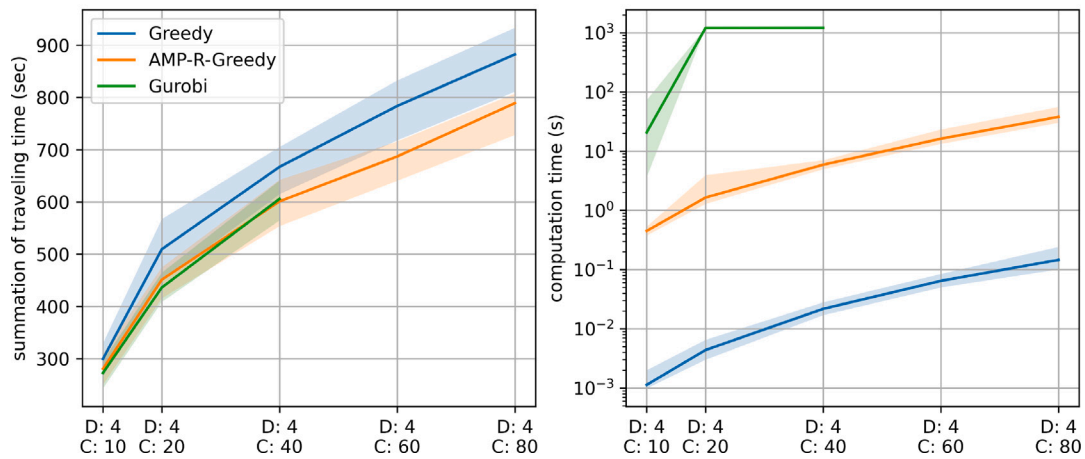
**HMDVRP-(c)(Capacitated, Unlimited, Heterogeneous)**

Fig. 8. Simulation Result : HMDVRP-(c)(Capacitated, Unlimited, Heterogeneous) case.

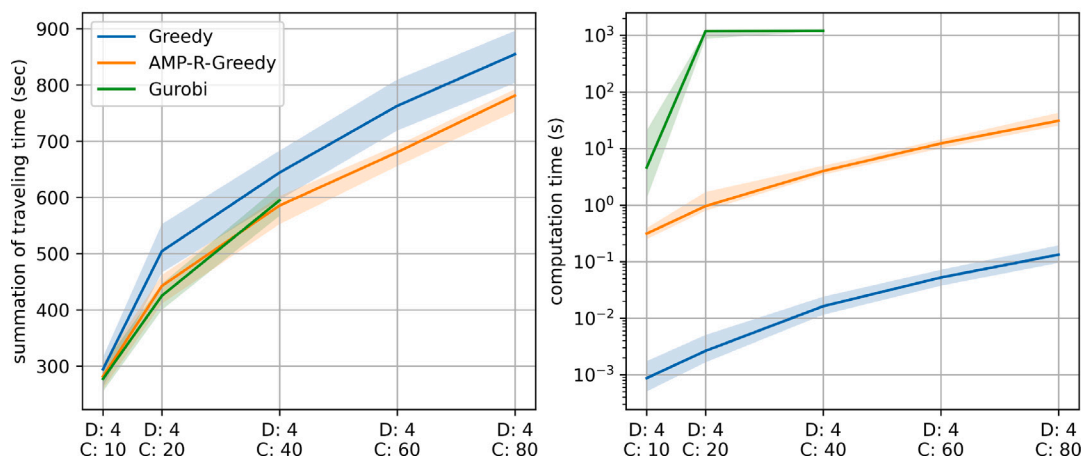
**MDVRP-(c)-LV(Capacitated, Limited, Homogeneous)**

Fig. 9. Simulation Result : MDVRP-(c)-LV(Capacitated, Limited, Homogeneous) case.

### HMDVRP-(c)-LV(Capacitated, Limited, Heterogeneous)

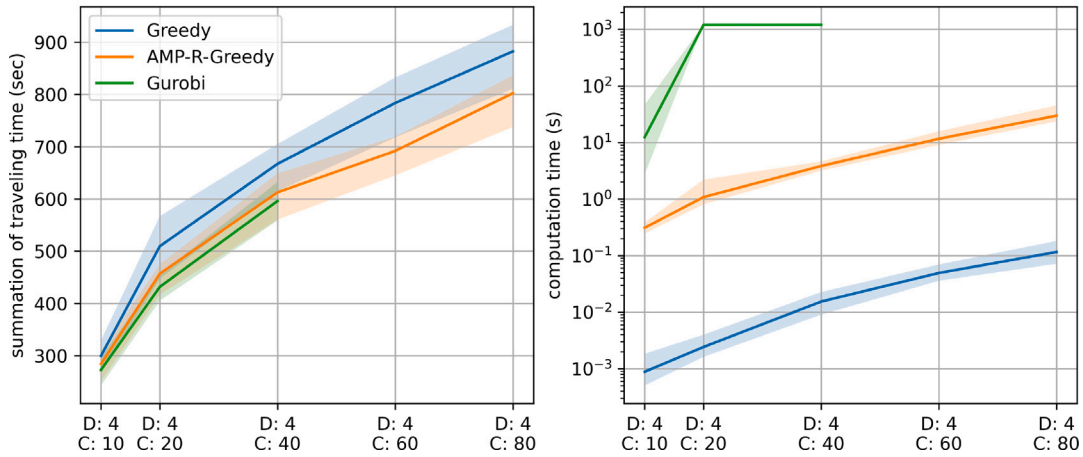


Fig. 10. Simulation Result : HMDVRP-(c)-LV(Capacitated, Limited, Heterogeneous) case.

**Table 5**  
Benchmark test results: total cost.

Problem	$N_d$	$N_c$	$L_v$	Gillett and Johnson [50]	FIND [51]	PIACO [29]	AMP-R-Greedy
1	4	50	80	593.2	576.86	576.86	611.21
2	4	50	160	486.2	473.53	473.53	520.0
3	5	75	140	652.4	641.18	641.18	674.83
4	2	100	100	1066.7	1003.86	1001.49	1062.40
5	2	100	200	778.9	750.26	750.26	783.63
6	3	100	100	912.2	876.5	876.5	911.98
7	4	100	100	939.5	892.58	885.69	954.46

#### 4.4. Benchmark test

In the previous subsections, the results of the proposed AMP-R algorithm were compared with the optimal solutions obtained using Gurobi, a greedy heuristic, and LKH across various HMDVRP variants. Beyond these general solution methods, evaluation against dedicated solvers specifically designed for HMDVRP is also necessary. However, no suitable benchmark studies for HMDVRP exist, and therefore, a comparative analysis was conducted using results from centralized approaches reported for the MDVRP benchmark problems [49]. The benchmark problems originally proposed by Christofides and Eilon consist of 2 to 5 depots and 50 to 100 customers, as summarized in Table 5, with all vehicles having identical capacities. For these benchmark problems, the results of heuristic methods [50], the tabu search based FIND algorithm [51], and the parallel improved ant colony optimization (PIACO) [29] have been previously reported, and their results are summarized in Table 5. The last column of the table presents the results of applying AMP-R-Greedy to each problem. Since AMP-R-LKH cannot be applied to capacitated problems, it was excluded from the comparison. The results show that the two metaheuristic methods, FIND [51] and PIACO [29], achieved the lowest-cost solutions, and the similarity of these globally optimized results suggests that they are close to the optima. AMP-R-Greedy demonstrated comparable performance to the heuristic algorithm proposed by Gillett and Johnson [50]. Although AMP-R-Greedy performs slightly worse than centralized metaheuristic methods, it provides solutions within 10% of the near-optima through decentralized computation.

#### 4.5. Managerial implications

This study proposes the AMP-R algorithm based on BP and demonstrates through simulations under various conditions that the HMDVRP can be effectively solved via decentralized computation. The managerial implications of this research are as follows. Companies seeking to

optimize their logistics and supply chains often operate complex networks with multiple distribution depots. The types and characteristics of transportation systems available at each facility can vary significantly. Problems involving the coordination of diverse transportation systems and geographically distributed depots for optimized delivery can be modeled as an HMDVRP. By solving such problems within a reasonable time frame, logistics companies can enhance resource utilization and operational efficiency. The AMP-R algorithm enables effective solutions to be derived through the exchange of assignment information between depots without centralized computation. This allows for improvements in overall supply chain efficiency without the need for unnecessary sharing of sensitive details, such as internal parameters or specifics about other depots and vehicles. In particular, when the scale of the supply chain is large, making centralized computation burdensome, and information sharing is restricted due to security concerns, the proposed method can provide significant advantages by enabling decentralized cooperation. Moreover, the algorithm can progressively improve the solution through continuous iterations without reconstructing the entire problem in response to partial changes in the network. This highlights its practicality as a solution that offers both continuity and stability in dynamic environments.

#### 5. Conclusion

This paper presented the AMP-R algorithm, a novel message-passing method based on belief propagation, for effectively addressing various forms of the HMDVRP in both centralized and decentralized settings. By modeling the HMDVRP as a statistical inference problem in a graphical model and applying the max-product BP approach, simplified message equations were derived to facilitate distributed computation. Despite the simplification, solving the SV-TSP and SV-CVRP in the message computation remains computationally intensive; therefore, an approximation technique was developed to efficiently compute the messages. Message buffers and a refinement step were incorporated to enhance convergence and ensure compliance with HMDVRP constraints

at each depot. In heterogeneous scenarios, transforming the SV-TSP into an ATSP yielded near-optimal solutions, demonstrating the practicality of the AMP-R algorithm to problem variations. Experimental results indicated that the AMP-R algorithm significantly outperformed traditional greedy approaches, even under conditions where finding optimal solutions was computationally impractical. Moreover, across six different HMDVRP variations, the solutions obtained within limited computation times closely approximated optimal results.

In ideal decentralized environments with minimal communication delays, computation can be effectively distributed among depots, further enhancing overall efficiency and scalability. The AMP-R algorithm is thus expected to make a significant contribution to logistics and task planning studies, offering a framework for providing practical solutions in decentralized vehicle routing problems. The distributed nature of the AMP-R algorithm makes it particularly well-suited for large-scale logistics and multi-robot planning problems where real-time decision-making in decentralized settings is essential. The ability to distribute computations across multiple depots also has the potential to enhance robustness against a constrained network with limited bandwidth or intermittent connectivity, such as autonomous vehicles operating in urban areas or remote environments. Theoretical guarantees on convergence and optimality in arbitrary HMDVRP instances remain an open problem, as BP-based methods may not always converge to the optimal solution.

#### CRedit authorship contribution statement

**Byeong-Min Jeong:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Dae-Sung Jang:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Han-Lim Choi:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

#### Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Dae-Sung Jang reports financial support was provided by Gyeonggi-do Regional Research Center. Han-Lim Choi reports financial support was provided by Defense Acquisition Program Administration. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This research was partly supported by Future Challenge Program through the Agency for Defense Development funded by the Defense Acquisition Program Administration, Republic of Korea (UC190075RD) and Gyeonggi-do Regional Research Center (GRRC), Republic of Korea funded by Gyeonggi Province (GRRC KAU 2023-B01).

#### Data availability

Data will be made available on request.

#### References

- Braekers K, Ramaekers K, Van Nieuwenhuyse I. The vehicle routing problem: State of the art classification and review. *Comput Ind Eng* 2016;99:300–13.
- Konstantakopoulos GD, Gayialis SP, Kechagias EP. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Oper Res* 2022;22(3):2033–62.
- Rios BHO, Xavier EC, Miyazawa FK, Amorim P, Curcio E, Santos MJ. Recent dynamic vehicle routing problems: A survey. *Comput Ind Eng* 2021;160:107604.
- Dantzig GB, Ramser JH. The truck dispatching problem. *Manag Sci* 1959;6(1):80–91.
- Laporte G, Nobert Y, Taillefer S. Solving a family of multi-depot vehicle routing and location-routing problems. *Transp Sci* 1988;22(3):161–72.
- Salhi S, Fraser M. An integrated heuristic approach for the combined location vehicle fleet mix problem. *Stud Locat Anal* 1996;8:3–22.
- Golden B, Assad A, Levy L, Gheysens F. The fleet size and mix vehicle routing problem. *Comput Oper Res* 1984;11(1):49–66.
- Tarantilis CD, Kiranoudis CT, Vassiliadis VS. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *J Oper Res Soc* 2003;54(1):65–71.
- Li F, Golden B, Wasil E. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Comput Oper Res* 2007;34(9):2734–42.
- Avci M, Topaloglu S. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Syst Appl* 2016;53:160–71.
- Taillard ÉD. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO- Oper Res* 1999;33(1):1–14.
- Wassan NA, Osman IH. Tabu search variants for the mix fleet vehicle routing problem. *J Oper Res Soc* 2002;53(7):768–82.
- Choi E, Tcha D-W. A column generation approach to the heterogeneous fleet vehicle routing problem. *Comput Oper Res* 2007;34(7):2080–95.
- Yu VF, Anh PT, Gunawan A, Han H. A simulated annealing with variable neighborhood descent approach for the heterogeneous fleet vehicle routing problem with multiple forward/reverse cross-docks. *Expert Syst Appl* 2024;237:121631.
- Garside AK, Ahmad R, Muhtazaruddin MNB. A recent review of solution approaches for green vehicle routing problem and its variants. *Oper Res Perspect* 2024;12:100303.
- Wang Y, Peng S, Zhou X, Mahmoudi M, Zhen L. Green logistics location-routing problem with eco-packages. *Transp Res Part E: Logist Transp Rev* 2020;143:102118, URL <https://www.sciencedirect.com/science/article/pii/S1366554520307663>.
- Xiao Y, Konak A. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transp Res Part E: Logist Transp Rev* 2016;88:146–66, URL <https://www.sciencedirect.com/science/article/pii/S1366554516000193>.
- Wei Y, Wang Y, Hu X. The two-echelon truck-unmanned ground vehicle routing problem with time-dependent travel times. *Transp Res Part E: Logist Transp Rev* 2025;194:103954, URL <https://www.sciencedirect.com/science/article/pii/S1366554524005453>.
- Salhi S, Sari M. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European J Oper Res* 1997;103(1):95–112.
- Benslimane MT, Benadada Y. Ant colony algorithm for the multi-depot vehicle routing problem in large quantities by a heterogeneous fleet of vehicles. *INFOR Inf Syst Oper Res* 2013;51(1):31–40.
- Salhi S, Imran A, Wassan NA. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Comput Oper Res* 2014;52:315–25, Recent advances in Variable neighborhood search.
- Kocaturk F. Heuristic approaches for multi depot vehicle routing problems with heterogeneous vehicle fleet. (Ph. D. Thesis), 2022.
- Vidal T, Crainic TG, Gendreau M, Prins C. Implicit depot assignments and rotations in vehicle routing heuristics. *European J Oper Res* 2014;237(1):15–28.
- Bolaños RI, Escobar JW, Echeverri MG. A metaheuristic algorithm for the multi-depot vehicle routing problem with heterogeneous fleet. *Int J Ind Eng Comput* 2018;9(4):461–78.
- Zhang Q, Wang Z, Huang M, Yu Y, Fang S-C. Heterogeneous multi-depot collaborative vehicle routing problem. *Transp Res Part B: Methodol* 2022;160:1–20.
- Li J, Lv X, Liu L. A parallel genetic algorithm with GPU accelerated for large-scale mdvrp in emergency logistics. In: 2011 14th IEEE international conference on computational science and engineering. 2011, p. 602–5.
- soeanu A, Ray S, Debbabi M, Berger J, Boukhtouta A, Ghanmi A. A decentralized heuristic for multi-depot split-delivery vehicle routing problem. In: 2011 IEEE international conference on automation and logistics. ICAL, 2011, p. 70–5.
- Saleh M, Soeanu A, Ray S, Debbabi M, Berger J, Boukhtouta A. Mechanism design for decentralized vehicle routing problem. *Proc the 27th Annu ACM Symp Appl Comput* 2012.
- Yu B, Yang ZZ, Xie JX. A parallel improved ant colony optimization for multi-depot vehicle routing problem. *J Oper Res Soc* 2011;62(1):183–8.
- Abu-Monshar A, Al-Bazi A, Palade V. An agent-based optimisation approach for vehicle routing problem with unique vehicle location and depot. *Expert Syst Appl* 2022;192:116370.
- Han X. Path planning algorithm for the multiple depot vehicle routing problem based on parallel clustering. *Sci Program* 2023;2023(1):7588595.
- Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. *Networks Plau Inference* 1988.
- Bayati M, Shah D, Sharma M. Max-product for maximum weight matching: Convergence, correctness, and lp duality. *IEEE Trans Inform Theory* 2008;54(3):1241–51.

- [34] Gamarnik D, Shah D, Wei Y. Belief propagation for min-cost network flow: Convergence and correctness. *Oper Res* 2012;60(2):410–28.
- [35] Lazic N, Frey B, Aarabi P. Solving the uncapacitated facility location problem using message passing algorithms. In: Teh YW, Titterton M, editors. *Proceedings of the thirteenth international conference on artificial intelligence and statistics. Proceedings of machine learning research*, vol. 9, Chia Laguna Resort, Sardinia, Italy: PMLR; 2010, p. 429–36.
- [36] Li W, Xu L, Schuurmans D. Facility locations revisited: An efficient belief propagation approach. In: 2010 IEEE international conference on automation and logistics. 2010, p. 408–13.
- [37] Sanghavi S, Shah D, Willsky AS. Message passing for maximum weight independent set. *IEEE Trans Inform Theory* 2009;55(11):4822–34.
- [38] Mézard M, Tarzia M. Statistical mechanics of the hitting set problem. *Phys Rev E— Stat Nonlinear, Soft Matter Phys* 2007;76(4).
- [39] Lim A, Wang F. Multi-depot vehicle routing problem: A one-stage approach. *IEEE Trans Autom Sci Eng* 2005;2(4):397–402.
- [40] Ramos TRP, Gomes MI, Póvoa APB. Multi-depot vehicle routing problem: a comparative study of alternative formulations. *Int J Logist Res Appl* 2020;23(2):103–20.
- [41] Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling-salesman problem. *J Oper Res Soc Am* 1954;2(4):393–410.
- [42] Jang D-S, Choi H-L. Decentralized message passing algorithm for distributed minimum sensor cover. *J Aerosp Inf Syst* 2017;14(7):373–90.
- [43] Besag J. Spatial interaction and the statistical analysis of lattice systems. *J R Stat Soc Ser B Stat Methodol* 1974;36(2):192–225.
- [44] Bishop CM, Nasrabadi NM. *Pattern recognition and machine learning*. Springer; 2006.
- [45] Helsgaun K. An effective implementation of the lin–kernighan traveling salesman heuristic. *European J Oper Res* 2000;126(1):106–30.
- [46] Helsgaun K. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. Technical report, Roskilde University; 2017.
- [47] Noon CE, Bean JC. An efficient transformation of the generalized traveling salesman problem. *INFOR Inf Syst Oper Res* 1993;31(1):39–44.
- [48] Campbell AM, Savelsbergh M. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transp Sci* 2004;38(3):369–78.
- [49] Christofides N, Eilon S. An algorithm for the vehicle-dispatching problem. *J Oper Res Soc* 1969;20(3):309–18.
- [50] Gillett BE, Johnson JG. Multi-terminal vehicle-dispatch algorithm. *Omega* 1976;4(6):711–8.
- [51] Renaud J, Laporte G, Boctor FF. A tabu search heuristic for the multi-depot vehicle routing problem. *Comput Oper Res* 1996;23(3):229–35.