Neurocomputing 554 (2023) 126631

Contents lists available at ScienceDirect

Neurocomputing

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

A survey of dynamic pickup and delivery problems

Junchuang Cai, Qingling Zhu, Qiuzhen Lin *, Lijia Ma, Jianqiang Li, Zhong Ming

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR China

A R T I C L E I N F O

Keywords:
Dynamic pickup and delivery problem
Dynamic vehicle routing
Combinational optimization
Real-word benchmarks

A B S T R A C T

Due to the ubiquitous real-world applications of logistics and supply chain management over the past two decades, dynamic pickup and delivery problems (DPDPs), as a subclass of dynamic vehicle routing problems in which objects or people must be collected and delivered in real time, have become a popular research topic in the field of combinatorial optimization. This article provides a comprehensive review of the DPDP literature from 2004 to 2023, in which their corresponding characteristics, principles, and theoretical analysis are discussed in detail. Furthermore, a taxonomy of the related solution methods for DPDPs is given, which can be segmented into four categories: exact methods, heuristics, metaheuristics, and learning-based methods. Moreover, some experimental comparisons and analysis of up-to-date real-word DPDP benchmarks from Huawei Company are included. Finally, a brief conclusion is given to summarize some potential future directions for DPDPs.

1. Introduction

As a generalization of the traveling salesman problem (TSP) [1], the vehicle routing problem (VRP) formulation was first presented in [2], which describes the problem of finding a set of least-cost routes for a fleet of vehicles to satisfy the total demand of a set of customers geographically dispersed in a network. As one of the most widely studied combinatorial optimization problems, the pickup and delivery problem (PDP) is a variant of the VRP in which products are picked up by vehicles from origins and delivered to destinations in a physical environment without considering new incoming orders. Generally, the goal of PDP is to minimize the total travel distances or the number of used vehicles while meeting the requirements of customers distributed at different sites, which has been proven to be an NP-hard problem [3].

Generally, the PDP is a static problem, as all the input data of the problem are known prior to the construction of routes. In contrast, the dynamic pickup and delivery problem (DPDP) is the dynamic counterpart of the PDP, in which the input data are gradually revealed or updated over time when requests are generated. Unlike the PDP, the planning horizon of the DPDP may be indefinite. Therefore, a solution to the DPDP cannot be a static output but rather a dynamic strategy that outlines the actions to be performed over time. However, most studies of PDPs have often focused on the static case and there has been limited research studies on DPDPs.

With the development of computer technology and the ubiquitous applications of DPDPs in real life, the studies of DPDPs have received more research attention in recent years. First, the advancement of information technologies and the increasing availability of data enable us to collect more relevant information for more advanced vehicle routing schemes. A key factor for successfully solving complex DPDPs is to provide reliable and flexible solutions. The recent development of telematics, such as the widespread use of positioning services and mobile communication, allows for the collection of real-time information and precise monitoring of vehicles. This has paved the way for extensive data collection and real-time decision support in DPDPs. Furthermore, today's computing power allows the application of vehicle routing algorithms to provide real-time solutions by incorporating online information and taking potential future events into account. These advances offer the opportunity to improve the quality of solutions for DPDPs, but they must be tackled by efficient optimization algorithms. Second, DPDPs are prevalent in various real-life scenarios, encompassing various domains such as taxi bookings [4], the dial-a-flight problem concerning passenger transportation [5,6], the express delivery routing problem [7], and the meal delivery routing problem [8,9]. These scenarios exemplify the wide applicability of DPDPs in our daily lives. Moreover, with the rapid growth of online shopping, DPDP has become a fundamental problem in the logistics and supply chain management of manufacturing companies, such as Huawei and Tesla. A large amount of products need to be delivered among various factories during the manufacturing process. The orders, containing information about pickup sites, delivery sites, time requirements and amounts of products, are generated dynamically, and then a fleet of vehicles is periodically

* Corresponding author.
E-mail addresses: qiuzhlin@szu.edu.cn (Q. Lin), mingz@szu.edu.cn (Z. Ming).

https://doi.org/10.1016/j.neucom.2023.126631
Received 13 April 2023; Received in revised form 6 July 2023; Accepted 24 July 2023
Available online 27 July 2023
0925-2312/

scheduled to serve these orders. Due to the uncertainty of requirements and production processes in DPDP, most of the delivery requirements cannot be predetermined. Even a small improvement of the logistics efficiency can bring significant benefits for manufacturing enterprises and customers. Also, the energy consumption may be reduced, which is of benefit to the environmental protection.

In the last two decades, there have been a number of studies reviewing dynamic VRPs (DVRPs) [10–19]. However, none of these reviews are devoted to the topic of DPDPs and there only has fragmented introduction about DPDPs among these reviews. Only a review of DPDPs [20] appears in 2010, but there is a gap of more than ten years about the studies of DPDPs. Therefore, this paper aims to update the review of DPDP studies.

In this paper, a taxonomy of the solution methods for DPDPs is provided. From the perspective of solution methods, DPDPs can be segmented into four categories: exact methods, heuristics, *meta*-heuristics, and learning-based methods. Then, the characteristics, principles, and theoretical analysis of various DPDPs are discussed in detail in this paper. Moreover, experimental comparisons and analysis are made on the up-to-date real-world DPDP benchmarks from Huawei company in this paper. Finally, some future directions for DPDPs and a brief conclusion are described.

Fig. 1 presents the structure diagram of this paper, which is mainly composed of four parts that will be introduced in sequence. The rest of this paper is organized as follows. Some background information is given in Section 2, which mainly introduces some basic concepts and definition of DPDPs. Section 3 reviews the existing DPDPs from the perspective of solution methods. The experimental comparisons and analysis on a practical DPDP benchmark set are presented in Section 4. Section 5 provides some future directions for DPDPs, as well as a brief conclusion.

2. Background

In general, some of the solution concepts and features of DVRPs are also applicable to DPDPs. In this section, we discuss some basic concepts and definition of DPDPs, some of which have been previously discussed in [13,20,21].

2.1. Some basic concepts of PDPs

As previously mentioned, PDPs are an essential class of vehicle routing problems in which objects or people must be transported from origins to destinations. These problems exist in a variety of applications, such as logistics, transportation systems, and robotics [22,23]. According to the surveys conducted on PDPs [20,24], PDPs can be classified into three categories.

The first kind is the many-to-many PDP, which has multiple origins and destinations for each pair of product/customer. This type of PDPs often exists in the repositioning of inventory between retail stores or in swapping problems [25]. The second kind is the one-to-many-to-one PDP, which involves delivering some products from a depot to many customers and collecting other products from many customers into the depot. This type of PDPs often exists in the distribution of beverages and the collection of empty cans and bottles, which is also found in forward and reverse logistics systems [26,27]. The last kind is the one-to-one PDP, which only has a given origin and a given destination for each pair of product/customer. This type of PDPs often exists in the supply chain management of manufacturing enterprises. As an example, Fig. 2 (a) shows that the information of three orders is known before routes are constructed. Then, these orders are serviced by scheduling a fleet of vehicles.

2.2. General definition of DPDPs

DPDPs commonly exist in various practical scenarios, such as restaurant meal delivery services and door-to-door transportation services, where customer orders are generated dynamically. Specifically, DPDPs are defined on a complete graph $G(V, E)$, where $V = \{0\} \cup \{i^+ | i \in R\} \cup \{i^- | i \in R\}$ represents the set of nodes and $E = \{(i,j) | i,j \in V, i \neq j\}$ represents the set of arcs between each pair of nodes. Here, $i^+(i^-)$ represents the origin (destination) node of request i, node 0 signifies the depot, R denotes the set of potential requests, and q_i denotes the load. Additionally, each arc $(i,j) \in E$ has a nonnegative length D_{ij} and travel time t_{ij}. A route is a circuit over a set of vertices, starting and finishing at the depot. Notably, the pickup and delivery nodes of a request are exclusive to a single vehicle, with the pickup node naturally preceding

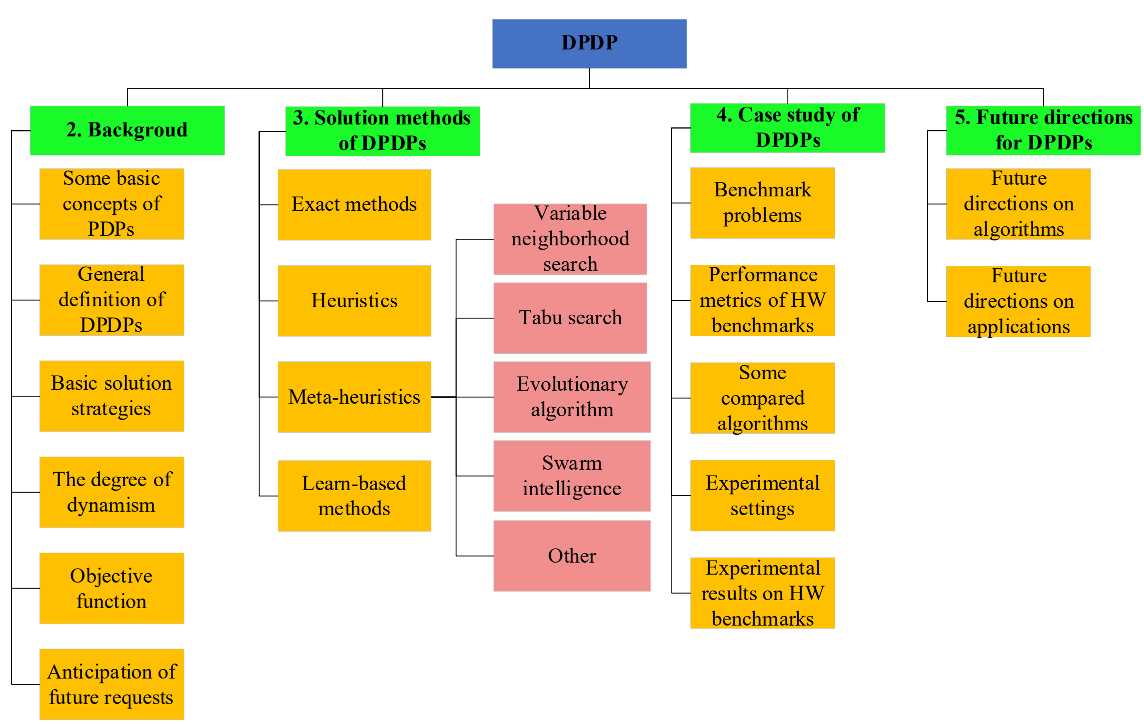

Fig. 1. The structure of this paper.

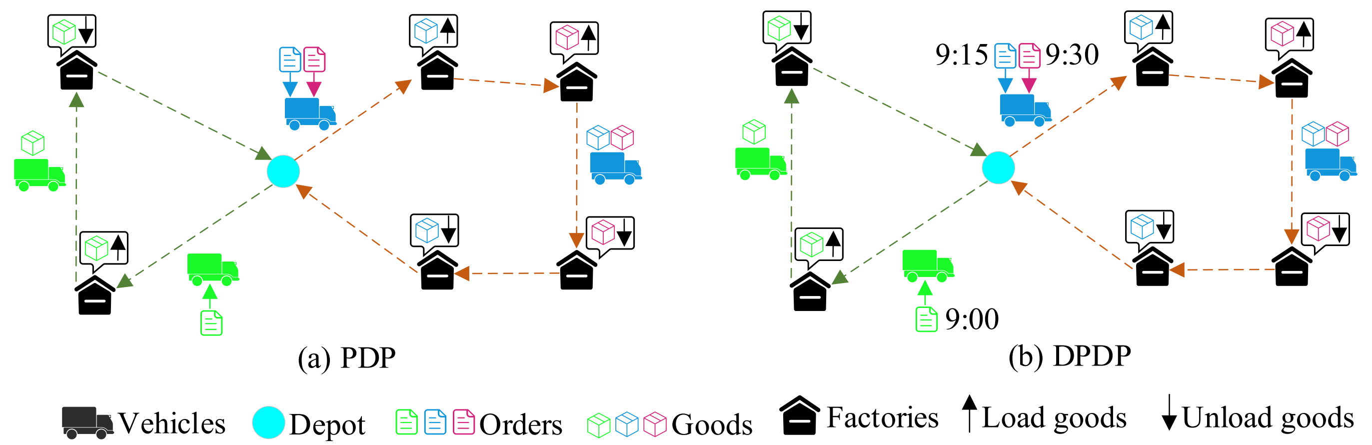

Fig. 2. A single example of PDP and DPDP.

the delivery node. At each time instant t, the request associated to the node is known, and each vehicle k can either be serving, waiting, or moving toward a node.

DPDPs further consider the reality of dynamically generating customer requests in PDPs. For example, as shown in Fig. 2 (b), three orders are generated dynamically at different times. The decisions required in DPDPs can be classified into two types of decision epochs: (i) wait or leave and (ii) accept or reject. A decision epoch of type wait or leave occurs when a vehicle finishes serving at a node. Given the current time t, the system has to decide whether to wait at the vehicle's current location or move towards a node whose request is known at time t. If the decision to wait has been chosen, a series of decision epochs of type wait or leave will be triggered at each time step until the decision to leave has been made, specifying the next node to visit. Finally, order requests are revealed throughout the time horizon, and the system has to determine whether to accept the request when a decision epoch has been triggered by a coming order request.

2.3. Basic solution strategies

In DPDPs, information is gradually revealed over time. A common strategy for solving a DPDP is to adapt an algorithm that solves the static version of the problem. Two strategies can be distinguished. The first strategy tackles a static problem by utilizing linear programming or mixed integer programming whenever new information, such as a new request or cancellation, becomes available [28]. However, this strategy suffers from high computational complexity, as it requires complete reoptimization when new information is revealed. This makes it unsuitable for real-time settings. The second strategy involves dynamically inserting new delivery orders into planned routes using heuristic methods, such as insertion heuristics [29–31], genetic algorithm [32,33], tabu search [34], waiting and buffering strategy [9,35,36], and variable neighborhood search [37]. Specifically, an initial solution is obtained using the available information only once at the beginning of the planning horizon. The current solution is updated with heuristic methods when new information is revealed and this update mechanism is generally fast enough to be applied in real time. More robust optimization methods are applied during the time interval in which new information is revealed to obtain the current solution.

In this paper, we classify DPDPs into four categories according to the types of solution methods: exact methods, heuristics, metaheuristics, and learning-based methods. The detailed introductions of the categories will be further detailed in Section 3.

2.4. The degree of dynamism

Designing an algorithm for solving DPDPs relies on the dynamism of the problem to a great extent. Two levels of dynamism can be observed in different instances of the same problem [15], namely, the frequency of changes and the urgency of requests. The former refers to the rate at which new information is revealed, while the latter is the time gap between the disclosure of a new request and its expected service time. From these two perspectives, the dynamism of DPDPs can be measured by three metrics.

Generally, the planning horizon is assumed to be a given interval $[0, T]$ in DPDPs, which is possibly divided into a finite number of smaller intervals. Let n_s and n_d be the numbers of static and dynamic requests, respectively, and $t_i \in [0, T]$ be the occurrence time of service request i. Moreover, $t_i = 0$ in static requests, while dynamic requests have $t_i \in (0, T]$. The degree of dynamism (dod) is defined by [38] as the ratio between the number of dynamic requests n_d and the total number of requests $(n_s + n_d)$, which is formulated as follows:

$$dod = \frac{n_d}{n_s + n_d}. \tag{1}$$

Thus, dod may vary between 0 and 1 in Eq. (1) and its meaning is straightforward. For example, if dod is equal to 0.2, it means that two out of 10 customers are dynamic. To account for both occurrence times and possible time windows of dynamic requests, Eq. (1) is generalized in [39] according to the observation that the disclosure time of requests is also important [21,40]. In Eq. (1), a problem is more dynamic if immediate requests occur at the end of the planning horizon $[0, T]$. As a result, a new measure of dynamism is given in [39], as follows:

$$dod' = \frac{\sum_{i=1}^{n_s+n_d} t_i / T}{n_s + n_d}. \tag{2}$$

It is worth noting that dod' ranges between 0 and 1. It is equal to 1 if all requests occur at time T, while it is equal to 0 if all requests are revealed in advance. Additionally, the definition of dod' is extended in [39] to take into account how long in advance dynamic requests are known. Let l_i denote the latest time that request i can be served. Then,

$$dod'' = \frac{1}{n_s + n_d} \sum_{i=1}^{n_s+n_d} \left[1 - \frac{l_i - t_i}{T}\right]. \tag{3}$$

Thus, we have $0 \leqslant dod'' \leqslant 1$, and a larger value of dod'' indicates more dynamic of a problem. Finally, the value of advanced information given by disclosure dates is quantified in [41], where experimental research is conducted for a version of the DPDP known as the dynamic stacker crane problem.

2.5. Objective function

Generally, DPDPs have various objectives, including minimizing the total traveling distance, number of used vehicles, service time, expected reachability time, and delays, and maximizing the satisfaction level of the customers or the number of requests served. Response time is a crucial characteristic of dynamic requests in DPDPs, as clients may require immediate attention. Therefore, the objective of the problem is to minimize the delay between the request arrival and its service [17]. Traditional static goals, such as minimizing the total distance travelled or delays, are also applicable in dynamic environments. The common objective functions of DPDPs usually consist of minimizing the total distance, travel time, travel cost, total lateness, and waiting time, and maximizing profit and customer requests.

In addition, some research studies [42,43] propose unique objectives. For instance, the virtual food court delivery problem is examined in [42], where a simple customer order can contain multiple restaurants, unlike traditional business models such as Grubhub and UberEat. The problem is formulated as a mixed-integer linear program with three objectives. The first objective aims to maximize the total earliness to all customers while also penalizing lateness to customers, the second objective minimizes the total time between order items being ready for pickup and delivery, and the last objective minimizes the total time between the pickup and delivery of order items. Moreover, the urban last-mile distribution problem is studied in [43]. Two objectives including minimizing the total delay of shipments and the total travel distance are tackled.

Additional literature examples categorized by objective functions are presented below:

1) Minimized objective functions:
 i. Minimizing the total cost: [6,31,44–51].
 ii. Minimizing the distance: [30,52–55].
 iii. Minimizing the travel time: [56,57].
 iv. Minimizing the total lateness: [9].
2) maximized objective functions:
 i. Maximizing the quality of service: [7,58].
 ii. Maximizing the profit: [4,59–61].
 iii. Maximizing the allocated requests: [62,63].
3) Other: [5,29,42,43,64–68].

2.6. The anticipation of future requests

Probabilistic distribution of future requests is a common characteristic of many DPDPs and DVRPs. For instance, a stochastic and dynamic model for DPDP, referred to as the restaurant meal delivery problem, is presented in [9], where the probability distributions on the time and location of customer requests are known. DPDPs are considered stochastic when some information about future requests is known. Recent efforts have been made to leverage stochastic information to enhance DVRP methods' performance, while research on stochastic DPDPs remains scarce. In [63], an approximate forward dynamic programming approach that integrates future request information into routing decisions is introduced to solve a multivehicle DPDP with time constraints, incorporating key features related to same-day delivery logistics. The sample-scenario approach is adopted, constructing different scenarios based on the current predecision state and random samples of the stochastic parameter set at each decision epoch. To solve a DPDP designed for dial-a-ride systems, a hierarchical multiobjective model-based predictive control approach is presented in [69]. This approach considers two sources of stochasticity: unknown future demand entering the system in real time and network traffic conditions in its spatial and temporal dimensions. The spatial and temporal dimensions of two sources of stochasticity are represented using a speed distribution, which is supposed to be known and obtained from historical data. Anticipatory strategies are also used to address stochastic DPDPs in several references [33,58,70,71].

However, in most real-life situations, DPDPs cannot be treated as stochastic as requests arrive based on an unknown stochastic process. In some complex scenarios, the exact probability distribution of future requests can be approximated using historical data [72–74], as it is not always known in advance. Such problems are widespread in various practical applications, such as cargo delivery [75–77], last-mile transportation [78], and emergency services [79]. In [80], an anticipatory algorithm is proposed to solve an unknown stochastic DPDP, where requests are revealed during the planning horizon. The approach employs a base policy (BP) controlled by a vector of parameters, which is selected by a supervised-learning model using features of the current instance and trained offline by a simulation model to predict the most appropriate BP vector. The algorithm of [80] anticipates the next requests based on the previous requests and a projection of the current trend. In [72], a general framework for solving dynamic stochastic combinatorial optimization problems is presented based on scenario sampling, and a family of algorithms are proposed to handle online combinatorial optimization problems by iteratively solving a series of static problems obtained by sampling each static problem and integrating the results. This framework can be applied to vehicle routing.

Waiting strategies are commonly used in DPDPs to incorporate future requests into route plans. For example, the route plan of vehicle v_k can be represented by $rp_k = \{n_1^k, n_2^k, \ldots, n_{l_k}^k\}$, where n_i^k is the i-th node and l_k is the number of nodes that vehicle v_k should visit. A waiting strategy determines how long the vehicle should wait at each node of rp_k before resuming its route. In DPDPs, waiting at nodes that may have future requests can reduce travel distance or increase the likelihood of servicing requests. In [36], a variety of general and specialized waiting heuristics were adapted and evaluated for the DPDP with time windows. Additionally, a novel waiting strategy based on an intensity measure was proposed, which utilizes past request information without the need for preprocessing. The authors also presented an approach for parameterizing specialized waiting strategies and adapting them to different problem characteristics through direct policy search and simulation-based optimization. The importance of waiting strategies has also been demonstrated by theoretical and experimental results in other literature [29,81–83].

Similarly, the buffering strategy can hold requests for a period before assigning them to vehicles or allow vehicles to move to another location based on the location and frequency of past requests, from which the location of future requests can be easily reached. The buffering strategy is useful for solving various DPDP variants, such as DPDPs with crowdsourcing in last mile delivery [62] and DPDPs with time windows [46,84]. In [51], a buffer is utilized to improve the solution of large-scale DPDPs. The buffer maintains a cache of newly generated orders and periodically dispatches all cached orders at once based on an upper-level agent's dynamic determination of whether to wait longer for more future orders. This approach aims to potentially optimize overall travel distances by providing each vehicle with a larger pool of candidate orders to choose from. To address crowd-sourcing DPDP, a task allocation algorithm was proposed in [62] to incorporate a buffering strategy and consider drivers' right of rejection. This algorithm enables autonomous decentralized decision-making for each agent by leveraging multi-agent reinforcement learning. In [84], a buffering strategy-based periodic optimization approach is introduced for DPDP with time windows. This approach iteratively buffers set of request arrivals and assigns them to available vehicles, with the objective of minimizing operational expenses while maintaining high-quality service levels.

To address the dynamism and uncertainty of the restaurant meal delivery problem, a waiting strategy is combined with a temporal buffer in [9]. The waiting strategy allows the delivery company to wait for additional information regarding ready times and new orders before assigning a driver to an order. The time buffer enables the delivery company to mitigate potential delays caused by ready times and the insertion of new requests into drivers' routes. In addition, the benefits of waiting and buffering strategies are also highlighted in an incapacitated DPDP [35].

3. Solution methods of DPDPs

In this section, a survey of DPDPs is presented. Based on the solution methods, existing DPDPs can be broadly categorized into four main categories: exact methods (as introduced in Section 3.1), heuristics (as introduced in Section 3.2), *meta*-heuristics (as introduced in Section 3.3), and learning-based methods (as introduced in Section 3.4).

3.1. Exact methods

Mixed-integer linear programming (MILP) is widely used for formulating combinatorial optimization problems. Combinatorial optimization problems are typically solved by using branch-and-bound, branch-and-cut, and branch-and-price solution methods after being formulated by MILP. MILP is frequently employed to solve dynamic and deterministic VRPs, primarily through myopic reoptimization. However, exact methods only provide an optimal solution for the current state and cannot guarantee that the solution will remain optimal when new requests become available [17]. The methods presented in this section primarily solve small instances or are used in conjunction with other methods, such as heuristics, metaheuristics and learning-based methods, to obtain high-quality solutions of DPDPs.

There are numerous examples of MILP for DPDPs. In [85], an MILP model is developed to schedule a dynamic pickup and delivery system that utilizes mobile robots to transport materials among different locations. The objective of this problem is to minimize the weighted sum of travel time and finish time. To solve this problem, the MILP model is embedded into a dynamic scheduling framework and periodically solved using commercial solvers. In [28], a dynamic routing algorithm for independent vehicles is proposed, which decomposes the target problem into a series of static optimization problems with a subset of known delivery orders over a rolling-horizon framework. The static optimization problem is then solved using a branch-and-price heuristic, which employs approximation, incomplete optimization techniques, and a sophisticated column management scheme to balance the solution speed and quality in a dynamic environment.

In [44], the authors explore the concept of crowdsourced delivery to utilize excess capacity on existing journeys. The problem aims to minimize the sum of costs associated with ad hoc driver matches and dedicated vehicle matches. A rolling horizon framework is proposed where the service platform creates matches between parcel delivery tasks and ad hoc drivers or a fleet of dedicated vehicles. An exact recursive algorithm is developed to solve the matching problem each time new information becomes available. Computational experiments reveal that the use of ad hoc drivers has the potential to make the last mile more cost-efficient and can provide system-wide vehicle-mile savings of up to 37% when compared to a traditional delivery system with dedicated vehicles.

In [45], the authors studied the economic and environmental sustainability of multicapacity rail-guided vehicles (RGVs) operating on a linear track automated freight handling system (AFHS). The problem is formulated as a capacitated pickup and delivery problem, where air cargo enters and leaves the system dynamically. An MILP model is used to minimize energy consumption. Two routing approaches (a rolling-horizon approach and a rule-based approach) are applied to compute the total energy cost. Similar rolling-horizon and rules-based approaches are reported in [8] and [86,87], respectively. The MILP model is integrated with a rolling-horizon approach to solve the problem. The simulations demonstrate that the rolling-horizon approach can reduce energy costs by up to 15% compared to a heuristic method currently in practice.

3.2. Heuristics

Most of the methods for DPDPs are heuristics, as they are typically fast and offer simple rules to generate good-quality solutions. In the following subsections, we review the main heuristic techniques used to solve DPDPs.

3.2.1. Insertion heuristics

Insertion heuristics are widely used in solving DPDPs due to their ability to address the curse of dimensionality in approximate dynamic programming. Theoretically, they belong to the class of constructive heuristics, wherein a feasible solution is constructed by inserting unscheduled nodes into a partial tour one by one until the final feasible solution is found [88].

For example, in [53], an insertion heuristic is proposed to optimize inventory and delivery strategies for e-groceries in the last-mile distribution. The objective of this problem is to minimize travel distances and maximize the remaining shelf lives of products upon delivery to customers. To solve this problem, a decision support system is developed by incorporating agent-based simulation and dynamic routing procedures to investigate e-grocery inventory and delivery operations. Three different inventory strategies are considered for order picking: first-expired-first-out, last-expired-first-out, and random picking. The insertion heuristic is applied in scheduling and routing of pickups and deliveries, which is started at every minute to construct a feasible solution. It sequentially evaluates the possibilities of adding new requests consisting of the pickup and delivery node to all positions of each vehicle's tour.

In another example, to facilitate efficient urban last-mile distribution, a decision support system is studied in [43], where orders are collected and delivered by a fleet of conventional vehicles owned by a logistics provider and cargo-bikes operated by freelancers. An agent-based simulation is proposed to investigate the corresponding problem setting, in which dynamic optimization procedures are applied to select transshipment points and generate vehicle routes. An insertion heuristic is used in the procedures, which evaluates all potential positions on all feasible vehicle routes considering each potential source of a shipment and selects the best insertion option. Computational experiments demonstrate that having a sufficient number of available cargo bikes and incorporating consolidation strategies is crucial to guarantee timely deliveries.

In [56], the authors examine a single-vehicle DPDP with multiple time-related constraints. The problem involves new requests that arrive over time and are inserted into the vehicle's pickup and delivery plan. The objective is to minimize the total traveling time. To solve this problem, an insertion heuristic based on large neighborhood search (LNS) and heuristic destination and repair (HDR) is proposed. The algorithm operates on an existing solution and a new request that should be inserted into the solution, and then returns a feasible solution to accommodate the new request. Simulations show that LNS outperforms HDR for small problem instances, but the quick convergence of HDR allows it to outperform LNS for larger instances.

In [84], a periodic approach to dynamic pickup and delivery problems with time windows is proposed. The problem involves a continuously repeated decision-making process with potentially urgent request information released over time. The objective is to minimize the sum of three components: total task tardiness, vehicle overtime, and total travel time. The periodic approach presents a two-step scheduling heuristic consisting of cheapest insertion (CI) followed by a local search. At each reoptimization step, CI inserts eligible requests into the available vehicles based on the current state to minimize the objective value.

Moreover, the insertion heuristic is widely applied to address DPDPs in numerous studies, such as [4,30,35,46,52].

3.3. Metaheuristics

In [89], a metaheuristic is defined as a high-level, problem-independent algorithmic framework that provides a set of strategies to develop heuristics for specific optimization problems. This section presents a survey of the main metaheuristics for DPDPs, including variable neighborhood search (VNS), tabu search (TS), evolutionary algorithm (EA), swarm intelligence (SI), and others.

3.3.1. VNS

VNS, first proposed in [90–92], is based on the idea that topological properties of the search space are defined by neighborhood structures, and changing neighborhood structures during the search procedure can obtain high-quality solutions. As an important area of evolutionary computation, VNS combines different local search heuristics with disturbance strategies and has shown effectiveness on a wide variety of combinatorial optimization problems, such as VRPs [93–96].

An example of using VNS to solve DPDPs is presented in [5]. The authors investigate a dial-a-ride problem that arises in the daily operation of the Austrian Red Cross, where each request requires transportation from a patient's home to a hospital or back home from the hospital. To tackle this problem, the authors present dynamic versions of variable neighborhood search (VNS) and a stochastic VNS (S-VNS). Computational experiments show that using stochastic information on return transport leads to average improvements of approximately 15%.

In [6], vehicle speed is considered as an important aspect in the dynamic dial-a-ride problem due to time-dependent and stochastic travel speeds, which frequently lead to missed time windows and poorer service. The authors use stochastic deviations from time-dependent travel speeds, which are deduced from historical accident data. To test the positive effect of using such data instead of average time-dependent travel speeds, the authors propose two pairs of metaheuristic approaches: dynamic stochastic variable neighborhood search (DSVNS) and multiple scenario approach (MSA). The approaches are compared to the corresponding myopic approaches: dynamic VNS (DVNS) and multiple plan approach (MPA). As a result, the DSVNS performs best, but the solution's quality highly depends on the *dod*. For highly and non-dynamic instances, the DVNS works better. However, the MSA is unsuitable for this problem setting, which obtains no improvements on average.

In [37], the authors propose an algorithm named VNSME (Variable Neighborhood Search with Multiple local search strategies and an Efficient disturbance) for addressing a practical DPDP that involves various constraints such as dock, time windows, capacity, and last-in-first-out loading. The VNSME employs a variable neighborhood search approach to find the optimal solution for each new optimization period that is triggered whenever new orders are received. First, the algorithm reconstructs the best solution from the previous period using exhaustive and cheapest insertion heuristics. It then utilizes four different local search strategies (couple-exchange*, block-exchange*, block-relocate*, and multi-relocate*) to explore the neighborhood of the initial solution. Finally, a modified 2-opt-L* method is used to perturb the currently found best solution. The VNSME was evaluated in a well-known competition and the experimental results demonstrated its superior performance in achieving the first rank among 153 participating teams.

3.3.2. TS

TS was first proposed in [97] as an algorithm that explores the solution space by moving to the best neighbor of the current solution, even when this movement deteriorates the objective function. In TS, a tabu list is created to store good-quality candidate solutions. To avoid cycling, solutions in the tabu list are made inaccessible for a certain number of iterations. This approach has proven to be one of the most successful ones in the literature to address DPDPs.

An example of TS is the approach used in [48] to study stochastic and dynamic vehicle routing problems with high computational burden due to the need to consider stochastic information. To overcome the drawback that the heuristic for this problem is relatively simple, the authors extended two versions of TS by different local search variations: a TS using stochastic information when updating the incumbent solution and a TS using stochastic information when selecting moves based on a list of moves determined through a proxy evaluation. The simulations indicate that adding stochastic information to the existing TS can further reduce operating costs for shipping companies by 0.5–2%.

In [57], DPDP with Time Windows (DPDPTW) is investigated, where requests from customers arrive continuously, vehicles may breakdown, and the time windows of customers may change. To address the DPDPTW, a periodic and event-driven rolling horizon procedure is proposed, which is divided into a series of static problems, and each static problem is approached by a quick TS algorithm. The simulation results show that the method is effective and its performance is satisfactory.

Furthermore, in [34], TS heuristics are proposed to optimize the planned routes of vehicles in a context where new requests, with a pickup and a delivery location, occur in real time. Within this framework, new solutions are explored through a neighborhood structure based on ejection chains. Numerical results demonstrate the benefits of these procedures in a real-time context. The impact of a master–slave parallelization scheme, using an increasing number of processors, is also investigated.

3.3.3. EA

Evolutionary computation (EC) is a population-based technique that relies on biological concepts. A number of evolutionary algorithms (EAs) have been designed in the EC field, which can be categorized into generational algorithms and steady-state algorithms [98]. Generational algorithms update the entire population once per iteration, while steady-state algorithms update a few candidate solutions at a time.

Genetic algorithm (GA) is a prominent component of EAs for DPDPs. It is often employed to produce high-quality solutions for optimization and search problems by utilizing biologically inspired operators such as mutation, crossover, and selection [99]. In [100], a study on a dynamic Original Equipment Manufacture (OEM) picking-up (milk-run) routing problem is presented, where tasks that are likely to exceed the time limit in a route are allocated to additional vehicles, thus creating auxiliary dynamic routes comprised of the transferred tasks from the regular trucks. To address this problem, a genetic algorithm is proposed, together with a simulation program designed to define relevant probabilistic parameters. The results of the computational experiments demonstrate that the dynamic formulation significantly enhances the service level compared to the static version.

Furthermore, according to [64], an approach based on GA, aggregation method, and minimum values is presented to optimize the dynamic multipickup and delivery problem with time windows. The primary objective is to minimize the trade-off between the total travel cost and the total tardiness time. In this algorithm, after generating the initial population, the recombination of parental genes to produce new descendants is accomplished through a one-point crossover phase. Additionally, a mutation phase, which randomly exchanges the respective values of two positions within a chromosome at random, is carried out. Moreover, in [65], a genetic algorithm is presented for multiobjective optimization of a dynamic multipickup and delivery problem with time windows. The problem is formulated with the objective of minimizing the compromise between the total travel cost and the total tardiness time.

Memetic algorithm (MA) is another widely used EA for solving DPDPs [101]. MA is an extension of the traditional GA that can provide a sufficiently good solution to an optimization problem by using a local search technique to reduce the likelihood of premature convergence [102]. In [49], the authors study a variant of the VRP in reverse logistics, which is the Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup (DVRPSDP). In this problem, new customers appear during the working day, and each customer requires simultaneous delivery and pickup. An MA is proposed to solve the problem with the objective of minimizing the total cost of the tours. Computational experiments indicate that the MA outperforms the ant colony system algorithm, and 86% of the MA results are better than the GA results. The work of [49] extends the research of [54].

MAs are commonly used in multiobjective optimization of DPDPs, as seen in several examples in the latest literature. According to [66], a multiobjective memetic algorithm (called LSH-MOMA) is proposed to solve one-to-many-to-one DPDP, in which three objectives, namely, route length, response time, and workload, are optimized simultaneously. LSH-MOMA is a synergy of a multiobjective evolutionary algorithm and locality-sensitive hashing (LSH)-based local search. In each generation of LSH-MOMA, LSH-based rectification and local search are applied to repair and improve the individual solutions. Specifically, a population of candidate routes (or individuals) is initially scheduled for all static nodes only before the vehicle leaves the depot. In the course of serving, dynamic requests are received and buffered in a request pool. Then, the pool is checked by LSH-MOMA in a fixed time slice, and dynamic nodes are inserted into the candidate routes by using the mutation operator and local search. The experimental results based on four benchmark DPDPs show that LSH-MOMA is efficient in obtaining optimal tradeoff solutions of the three objectives. In [67], one-to-many-to-one DPDPs with dynamic customer requests and traffic information are formulated, which are similar to the problem in [66]. To solve these problems, a multiobjective memetic algorithm called *prio*LSH-MOMA is proposed, in which a priority and locality-sensitive hashing-based local search is applied to fine-tune the candidate routes during the evolutionary process. In the computational experiment, *prio*LSH-MOMA is evaluated with two DPDPs simulated on real-world maps, demonstrating the efficiency of the proposed algorithm. A similar work solving multiobjective optimization of DPDPs using MAs can also be found in [68].

Moreover, other EAs are also applied to solve DPDPs. In [103], an evolution strategy of [6] is chosen, which uses mutation strength and adjusts it over time depending on the success of the search.

3.3.4. Swarm intelligence

Inspired by natural systems such as ant colonies and flocks of birds, SI refers to the collective behavior of systems and is applied in artificial intelligence research. SI systems are typically composed of a population of simple agents that interact locally with each other and their environment [104]. The most commonly used SI algorithms are ant colony optimization (ACO) [105] and particle swarm optimization (PSO) [106].

In [55], a modified ACO algorithm is proposed to tackle a set of single-vehicle DPDPs that extends the PDP to a set of vehicles already on the road when a new pickup and delivery order arrives. The algorithm dynamically defines new routes of each vehicle while minimizing the overall cost when integrating additional locations. The performance of the proposed algorithm is validated through experiments on three different map sizes.

In [70], a hybrid adaptive predictive control approach is proposed to solve a DPDP. The approach considers an additional stochastic effect within the analytical expression of the objective function for vehicle scheduling and routing, which is the extra cost associated with potential rerouting arising from unknown future requests. The proposed approach utilizes an ad hoc PSO algorithm to efficiently tackle the problem after modelling the DPDP by an adaptive predictive control framework. A simulated numerical example is used to validate the PSO method.

3.3.5. Other

Hybrid metaheuristics can combine multiple metaheuristics, such as VNS, GA, TS, the adaptive large neighborhood search (ALNS), and the multiobjective evolutionary algorithm based on decomposition (MOEA/D). In this section, we introduce hybrid metaheuristics used to solve the DPDP.

In [50], a hybrid metaheuristic called MOEA/D-ES is proposed to solve a practical DPDP [75] with constraints such as docks, time windows, capacity, and last-in-first-out loading. The objective of this DPDP is to minimize the total timeout of orders and the average traveling distance of vehicles. MOEA/D-ES combines MOEA/D and VNS, which evolves iteratively by performing crossover, VNS, and updating reference points and neighborhood solutions after initialization. The experimental results on 40 real-world logistics problem instances validate the high efficiency and effectiveness of MOEA/D-ES.

In [33], a hybrid metaheuristic combining PSO and GA is proposed to solve a DPDP with dial-a-ride service formulated under a hybrid predictive control (HPC) approach. This hybrid metaheuristic is used inside the ad hoc methodology to solve the HPC of the dial-a-ride system and to find good insertion positions of the requests.

In [61], a hybrid metaheuristic is introduced that combines GA, VNS, and TS to address a DPDP with time windows, heterogeneous fleet vehicles, no depot, and dynamic priority. The algorithm is compared against the best-known solutions on the benchmark instances, and the results show that it can improve over 35% (52 out of 146) of the total distance and over 30% (44 out of 146) of the vehicle used.

In [31], a dynamic algorithm framework is presented to tackle a dynamic vehicle routing model based on a DPDP that considers multiple dynamic events in a real-world environment. The framework consists of two stages: construction of an initial solution with the cheapest insert heuristics and improvement of the initial solution with the hybrid metaheuristics of TS and VNS in each subinterval. The experimental results indicate that the model and dynamic algorithm framework can effectively solve the DPDP with time windows.

3.4. Learning-based methods

In recent years, learning-based methods have gained significant attention in addressing various combinatorial optimization problems [107,108], leading to remarkable research breakthroughs. These methods enable the learning of models from training sets to derive optimization strategies. Such strategies can automatically generate solutions for online tasks using either end-to-end or step-by-step approaches [109]. In end-to-end approaches, the model is trained to approximate the mapping function between inputs and solutions, allowing it to directly produce feasible solutions for unknown tasks in real-world applications. Conversely, step-by-step approaches focus on learning optimization strategies that iteratively improve a solution rather than providing a final solution directly. Both approaches exhibit strong learning capabilities and generalization potential.

When addressing DPDP or related dynamic optimization problems, learning-based methods offer several advantages over heuristic methods. A summary of the advantages is given below. Firstly, learning-based methods excel in their ability to learn. Through continuous learning and experience accumulation, they enhance performance by extracting valuable knowledge and patterns from historical data. This optimization of the decision-making process gradually improves problem-solving capabilities. In contrast, heuristic methods rely on expert knowledge to effectively tackle problems [51]. Secondly, learning-based methods possess adaptability as a key advantage [110]. Given the high uncertainty associated with future orders, heuristic algorithms struggle to account for them. In contrast, learning-based methods can leverage data and experience to predict future order demands, enabling them to make informed decisions that adapt to real-time information and order requirements, thereby accommodating dynamic changes. Moreover, learning-based methods typically exhibit strong generalization abilities and can adapt to problems of varying scales and complexities. By increasing training data and refining learning algorithms, they can be extended to larger-scale problems while maintaining a certain level of solution effectiveness. Conversely, heuristic methods necessitate personalized parameter settings when confronted with problems of differing scales and complexities.

In recent years, learning-based methods for solving DPDPs have become a popular topic in computer science. These methods enable computers to learn data, make predictions, and schedule routes. Learning-based methods are capable of improving the "myopic view" of heuristic algorithms, producing competitive results, and performing better under dynamic and stochastic circumstances, as shown in previous work on TSP [111], VRP [112], PDP [113] and DPDP [51,110]. In this section, a survey of learning-based methods for solving DPDPs is illustrated.

In [110], an industry-scale DPDP is investigated, in which the number of vehicles and order requests become large. The objective of the DPDP is to minimize the number of used vehicles and the total travel length of all used vehicles in one episode. To address this problem, a data-driven approach called the spatial–temporal aided double deep graph network (ST-DDGN) is proposed. ST-DDGN forecasts the delivery demands using a spatial–temporal prediction method that guides the neural network to perceive the spatial–temporal distribution of delivery demand when dispatching vehicles. By establishing a graph-based value function, the relationships of individuals such as vehicles are modelled. Additionally, attention-based graph embedding is incorporated with double DQN (DDQN) in ST-DDGN. Extensive experiments using real-world data are conducted to evaluate DDQN, and the results show that STDDGN reduces the number of vehicles used by 11.27% and decreases the total transportation cost by 13.12% on average over the strong baselines, including the heuristic algorithm deployed in the user acceptance test environment and a variety of vanilla deep reinforcement learning methods.

Large-scale DPDPs are dynamically generated in real time with the objective of minimizing the weighted sum of average traveling distances and total overtime, which are also studied in [51]. A novel hierarchical optimization framework is proposed to better solve these problems, consisting of an upper-level agent and a lower-level agent. The upper-level agent dynamically partitions the DPDP into a series of sub-problems with different scales to optimize vehicles' routes towards globally better solutions. Meanwhile, the lower-level agent, similar to traditional metaheuristic algorithms, sequentially manipulates operators (i.e., inner-exchange, inner-relocate, inter-exchange, and inter-relocate) to improve the solution of each PDP. Furthermore, to verify the effectiveness of the proposed framework, a functional simulator is built using real historical data collected from the order dispatching system of the Huawei Supply Chain Business Unit. Computational experiments conducted on the industrial order dispatching system justify the superior performance of the framework over existing baselines.

Moreover, an unknown stochastic DPDP is considered in [80], where a procedure to use a base policy (BP) controlled by a vector of parameters is applied. Then, some features of the current instance are estimated and fed into a supervised-learning model with the objective of predicting the most appropriate BP vector when a new request arrives. The supervised learning model is trained offline using a simulation model.

4. Case study of DPDPs

4.1. Benchmark problems

Currently, there are only a limited number of reference benchmarks available for DPDPs, with most benchmarks being generated from artificial data rather than real-word data. One example of an artificial benchmark is presented in [114], which provides a dataset of instances with varying degrees of dynamism and urgency configurations. The generator produced instances with eight different degrees of dynamism (20%, 30%, 40%, 50%, 70%, 80%, 90%, and 100%) and five different levels of urgency (5, 15, 25, 35, and 45 min). Five instances were produced for each of the 40 possible combinations, resulting in a total of 200 instances. Furthermore, most artificial datasets used for DPDPs are adapted from PDP datasets [31,36,61,71] or randomly generated [7,47,59,60,115] for each static period, which are not necessarily specific to DPDPs.

On the other hand, practical benchmarks such as the HW benchmarks [75] proposed by Huawei in the competition at the International Conference on Automated Planning and Scheduling (ICAPS 2021) are more realistic and challenging. The HW benchmarks introduce enterprise-scale constraints such as dock, time windows, capacity, and Last-In-First-Out (LIFO) loading, based on real system data of 30 days from Huawei's historical records. The datasets consist of four vehicle quantity scales (5, 20, 50, and 100) and eight order quantity scales (50, 100, 300, 500, 1000, 2000, 3000, and 4000), as presented in Table 1. The HW benchmarks consist of 8 problem scales classified by the number of vehicles and orders, with each problem scale containing eight instances, resulting in a benchmark set of 64 instances. The generation process for each problem scale is outlined as follows. Taking the largest problem scale (HW57 - HW64) as an illustration, for each instance within this scale, a random day is initially selected from a pool of 30 sampled days. Subsequently, 4000 orders are randomly chosen from the historical data of the selected day and assigned to be serviced by 100 vehicles. The generation process for the remaining problem scales follows a similar pattern to that of the largest scale. Consequently, the HW benchmarks exhibit a distinct distribution pattern. To facilitate the comparison experiments' description, we categorize these 64 instances into three general problem scales (small, medium, and large) based on the number of orders. Fig. 3 displays the new orders' distributions generated dynamically in one day by the last instance (i.e., HW32, HW48, and HW64) of each problem scale to provide a clearer view of the order distribution for these three scale problems. As shown in Fig. 3, the small problem scales have sparse distributions of new orders in 144 intervals in a day, with the number of new orders in each interval being small. For medium-scale problems, new orders are densely distributed at 144 intervals in a day. In contrast, in large-scale problems, the orders in 144 intervals (10 min each interval) are not only dense, but most intervals have a large number of new orders. Other practical benchmarks of DPDPs can be found in various applications, such as the meal delivery routing problem [8], the same-day courier [56], and the dial-a-ride problem [5,6]. In the following sections, we evaluate the performance of several state-of-the-art algorithms on the HW benchmarks [75].

4.2. Performance metrics of HW benchmarks

The DPDP model for HW benchmarks involves multiple factories that can dynamically generate delivery orders within a day. The model includes two optimization objectives. The first objective is to minimize the total tardiness of orders denoted as f_1. If the arrival time of the order o_i to be delivered to the destination is denoted as a_d^i, then the optimization objective can be expressed as:

$$f_1 = \sum_{i=1}^{N} \max(0, a_d^i - t_l^i) \tag{4}$$

where t_l^i is the committed completion time, and N is the total number of orders.

Table 1
The number of vehicles and orders of different HW instances.

Instance	Vehicle number	Order Number	Scale
HW1 - HW8	5	50	small
HW9 - HW16	5	100	small
HW17 - HW24	20	300	small
HW25 - HW32	20	500	small
HW33 - HW40	50	1000	medium
HW41 - HW48	50	2000	medium
HW49 - HW56	100	3000	large
HW57 - HW64	100	4000	large

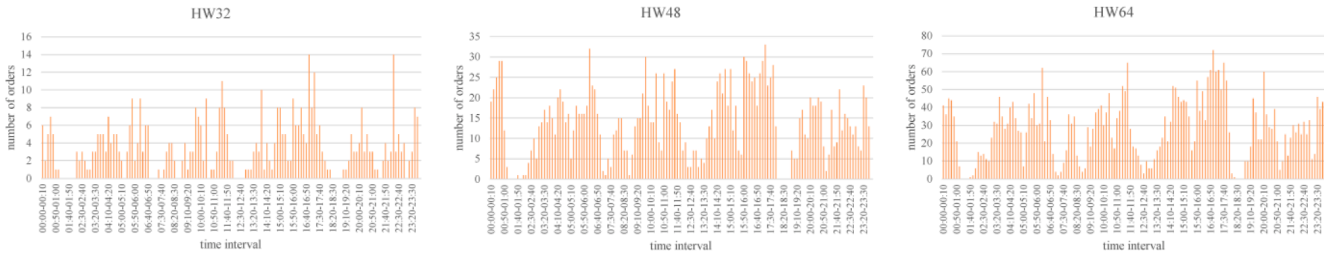

Fig. 3. The distributions of new orders generated by the last instance of three problem scales in the HW benchmark set.

$$f_1 - \sum_{k=1}^{K}\sum_{i=1}^{l_k}\max(0, t_i^k - t_l^{y_i^k})d_i^k = 0 \tag{5}$$

$$t_{i+1}^k = t_i^k + w_i^k + T_{da} + t_s^{y_i^k} + t_{i,i+1}^k \tag{6}$$

In Eq. (5), the first objective function in Eq. (4) is specified. f_1 also represents the total tardiness of the orders serviced by each vehicle. The order visited by vehicle v_k at the i-th node is denoted as y_i^k, and $t_l^{y_i^k}$ represents the committed completion time of that order. Specifically, the tardiness of an order is calculated when vehicle v_k reaches a delivery node (i.e., when d_i^k equals 1). Eq. (6) describes the calculation of the arrival time for vehicle v_k to reach the $(i+1)$-th node. The arrival time at the i-th node for vehicle v_k is denoted as t_i^k, while w_i^k represents the waiting time for the vehicle to be allocated at the dock after reaching node i. T_{da} denotes the fixed running time for a vehicle to travel from the factory to the factory's dock. $t_s^{y_i^k}$ represents the time required to load or unload cargoes for the order y_i^k corresponding to the i-th node visited by vehicle v_k. Lastly, $t_{i,i+1}^k$ indicates the travel time from the i-th node to the $(i+1)$-th node after vehicle v_k completes its service at the i-th node.

The second objective is to minimize the average traveling distance of vehicles denoted as f_2. If the route plan of vehicle v_k is $rp_k = \{n_1^k, n_2^k, \ldots, n_{l_k}^k\}$, where n_i^k stands for the i-th factory in the route of vehicle v_k, and l_k is the total node number travelled by vehicle v_k. The optimization objective can be expressed as:

$$f_2 = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{l_k-1} D_{n_i^k,\, n_{i+1}^k} \tag{7}$$

where $D_{n_i^k,\, n_{i+1}^k}$ is the distance from node n_i^k to node n_{i+1}^k.

The optimization objectives f_1 and f_2 are conflicting. For instance, when multiple orders with the same pickup and delivery nodes are revealed at different times and serviced at once, the minimum value of f_1 conflicts with the maximum value of f_2. Conversely, when earlier orders are postponed and bundled with later orders with the same pickup and delivery nodes, the maximum value of f_1 conflicts with the minimum value of f_2.

In the HW benchmarks, the goal is to dispatch all orders to a fleet of vehicles while minimizing the total tardiness of orders and the average traveling distance of vehicles. Therefore, the total cost is employed as the performance metric:

$$TC = \alpha \times f_1 + f_2 \tag{8}$$

where α signifies the penalty value attributed to a one-hour delay in each order. It is worth noting that the time window constraint in this problem is considered as a soft constraint, resulting in a penalty value weighted by f_1 and α when an order is delivered beyond the committed completion time. For this particular problem, α is assigned a large positive constant value to reflect the emphasis placed by enterprises on ensuring timely order fulfillment, thus enhancing customer satisfaction in real-world scenarios.

4.3. Some compared algorithms

Recently, four state-of-the-art heuristic algorithms have been proposed for solving HW benchmarks, including MOEA/D-ES [50] and the top three algorithms in ICAPS 2021 (i.e., gold-winning, silver-winning and bronze-winning algorithms). Information about ICAPS 2021 is available at https://competition.huaweicloud.com/information/1000041411/circumstance. To investigate the practical DPDP, experimental comparisons of these four algorithms are performed on HW benchmarks. A brief introduction of each compared algorithm is given below:

1) MOEA/D-ES [50]: As illustrated in Fig. 4, MOEA/D-ES is a multi-objective evolutionary algorithm based on MOEA/D with four efficient local search strategies. This method decomposes the target DPDP into many subproblems, which makes solutions diverse and easy to solve. After population initialization, MOEA/D-ES evolves by iteratively performing crossover, VNS, and updating the reference point and neighborhood solutions. In VNS, four local search

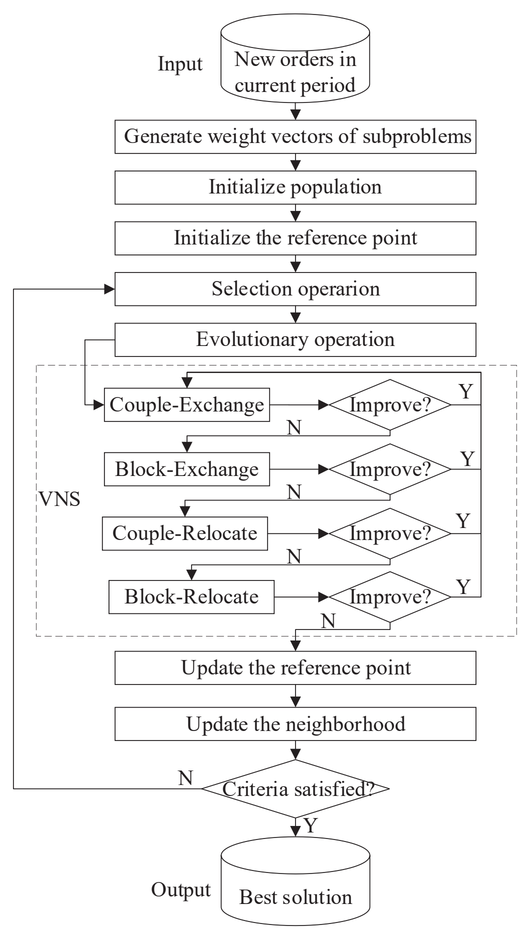

Fig. 4. The framework of MOEA/D-ES.

strategies (i.e., couple-exchange, block-exchange, couple-relocate and block-relocate) are applied.

2) Gold-winning algorithm [37]: The algorithm utilizes the variable neighborhood search method to determine the route plans of vehicles. It continuously swaps the pickup node group and delivery node group among the route plans of different vehicles and exchanges the orders inside a route plan for better results. The pickup (delivery) node group consists of adjacent pickup (delivery) nodes with the same address. Additionally, the delay strategy is applied by this algorithm to obtain a better solution.
3) Silver-winning algorithm: This algorithm adopts a strategy that transforms the DPDP problem into a knapsack problem and packs nodes for each route, taking all practical constraints into consideration at the same time. Specifically, it is developed from the perspective of a threshold check, i.e., allocating the orders by checking whether they reach the threshold of delivery time and vehicle capacity. If so, those orders would be allocated.
4) Bronze-winning algorithm: The algorithm constructs the initial solution for the DPDP by the cheapest insert algorithm, in which the pickup node and delivery node are consecutive in the route for every insertion operation. Moreover, the solution is improved by local search through a ruin-reconstruct strategy.

4.4. Experimental settings

The running time of all the compared algorithms for processing the orders in each interval period is limited to 10 min. Each algorithm runs each instance of the HW benchmarks ten times independently, and the best total cost in Eq. (8) of each algorithm is reserved for comparison. The source codes of the gold-winning, silver-winning, and bronze-winning algorithms can be found at https://competition.huaweicloud.com/information/1000041411/Winning, while MOEA/D-ES is referred to as [50]. To ensure a fair comparison, we have set the corresponding parameters of the considered competitors according to the suggestions provided in their original references. Furthermore, the experimental settings for DPDP align with those of the DPDP competition at ICAPS 2021, guaranteeing a fair comparison. These common settings encompass the following aspects: 1) the penalty value for one-hour tardiness: $\alpha = 10000/3600$, 2) the maximum number of docks in each factory is set to 6, 3) the maximum loading capacity Q is adopted as 15, and 4) the maximum running time is set to 10 min.

Please note that all the compared algorithms are run on Ubuntu 18.04.4 LTS operating system, Intel(R) Core (TM) i5-9500 CPU @ 3.00 GHz $\times$ 4, and 21G memory. Each algorithm is independently run 10 times for each test problem, and other parameters are configured with the same values as their corresponding references.

4.5. The performance of the compared algorithms on HW benchmarks

In this section, we present the experimental results of the compared algorithms on HW benchmarks. To provide a clearer analysis, we divide the results into three categories based on the dataset size: small-scale, medium-scale, and large-scale.

4.5.1. Results on small-scale HW benchmarks

Table 2 presents the best and average performance of four algorithms on small-scale instances of the HW benchmarks (HW1-HW32). The "Gap" and "MTC" columns indicate the gap between $f_{\min}$ and f_{best} and the

Table 2
Comparison between MOEA/D-ES, Gold-winning, Silver-winning, and Bronze-winning algorithms on the small-scale instances of the HW benchmarks over ten runs ("MTC" represents the minimum value of TC in Eq. (8) across ten runs, "Gap" indicates the difference between the solution with the minimum TC obtained by an algorithm across 10 runs and the best solution among all algorithms across 10 runs).

instance	MOEA/D-ES		Gold		Silver		Bronze	
	MTC	Gap	MTC	Gap	MTC	Gap	MTC	Gap
HW1	**1.17E + 2**	**0.0000**	1.34E + 2	0.1449	2.30E + 3	18.584	1.30E + 2	0.1069
HW2	**8.88E + 1**	**0.0000**	9.56E + 1	0.0766	3.05E + 4	342.46	9.14E + 1	0.0293
HW3	**9.41E + 1**	**0.0000**	9.68E + 1	0.0283	3.58E + 4	379.44	9.65E + 1	0.0255
HW4	**9.45E + 1**	**0.0000**	9.46E + 1	0.0008	5.49E + 3	57.095	1.04E + 2	0.1005
HW5	3.31E + 3	0.0004	**3.31E + 3**	**0.0000**	1.69E + 4	4.1124	5.45E + 3	0.6487
HW6	**1.05E + 2**	**0.0000**	1.05E + 2	0.0029	4.76E + 3	44.445	1.18E + 2	0.1266
HW7	**4.32E + 3**	**0.0000**	4.39E + 3	0.0160	1.28E + 4	1.9626	7.36E + 3	0.7035
HW8	**6.39E + 1**	**0.0000**	6.88E + 1	0.0780	7.97E + 2	11.480	7.69E + 2	11.042
HW9	1.65E + 2	0.0868	**1.52E + 2**	**0.0000**	1.81E + 5	1188.0	8.48E + 3	54.708
HW10	**7.46E + 4**	**0.0000**	1.94E + 5	1.6020	1.77E + 6	22.711	1.87E + 5	1.5051
HW11	**1.61E + 2**	**0.0000**	1.98E + 2	0.2291	1.81E + 5	1123.9	3.04E + 3	17.893
HW12	**8.61E + 3**	**0.0000**	5.29E + 4	5.1503	5.67E + 5	64.861	8.42E + 4	8.7805
HW13	**1.73E + 2**	**0.0000**	7.18E + 3	40.587	2.20E + 5	1273.7	2.77E + 2	0.6051
HW14	**1.50E + 2**	**0.0000**	9.39E + 3	61.739	2.37E + 5	1582.1	7.82E + 3	51.237
HW15	1.92E + 4	0.4144	**1.35E + 4**	**0.0000**	7.93E + 5	57.539	1.49E + 5	9.9993
HW16	**1.04E + 4**	**0.0000**	1.68E + 4	0.6072	8.54E + 5	80.901	5.78E + 4	4.5432
HW17	**7.34E + 1**	**0.0000**	8.17E + 1	0.1139	9.65E + 1	0.3153	4.11E + 2	4.6017
HW18	**7.97E + 1**	**0.0000**	8.22E + 1	0.0318	1.12E + 2	0.4060	8.56E + 3	106.45
HW19	**1.06E + 2**	**0.0000**	1.09E + 2	0.0219	4.78E + 2	3.4999	4.15E + 3	38.068
HW20	3.30E + 3	0.0001	**3.30E + 3**	**0.0000**	3.98E + 3	0.2074	1.62E + 4	3.9146
HW21	**9.74E + 1**	**0.0000**	1.13E + 2	0.1641	2.34E + 3	23.037	1.85E + 4	189.03
HW22	**1.64E + 3**	**0.0000**	1.65E + 3	0.0054	3.32E + 3	1.0289	2.11E + 4	11.894
HW23	1.06E + 2	0.0059	**1.05E + 2**	**0.0000**	6.87E + 3	64.198	8.09E + 2	6.6777
HW24	**8.53E + 1**	**0.0000**	9.44E + 1	0.1069	1.29E + 3	14.123	2.00E + 3	22.446
HW25	**5.98E + 3**	**0.0000**	1.03E + 4	0.7144	9.24E + 4	14.444	2.15E + 4	2.5937
HW26	**4.48E + 3**	**0.0000**	8.95E + 3	0.9951	2.75E + 5	60.319	5.40E + 4	11.041
HW27	**1.19E + 2**	**0.0000**	1.34E + 2	0.1240	1.84E + 4	153.26	1.88E + 4	156.61
HW28	7.12E + 3	0.0031	**7.10E + 3**	**0.0000**	9.44E + 3	0.3293	1.49E + 4	1.0982
HW29	**5.92E + 3**	**0.0000**	6.40E + 3	0.0804	1.63E + 5	26.522	4.24E + 4	6.1593
HW30	**1.10E + 2**	**0.0000**	1.19E + 2	0.0809	7.40E + 4	671.54	2.11E + 4	190.76
HW31	**1.45E + 4**	**0.0000**	2.26E + 4	0.5609	1.47E + 5	9.1543	2.33E + 4	0.6095
HW32	**5.69E + 3**	**0.0000**	7.67E + 3	0.3484	6.04E + 4	9.6176	1.96E + 4	2.4455
best/all	26/32		6/32		0/32		0/32	
Avg	5.35E + 3		1.16E + 4		1.80E + 5		2.50E + 4	

minimum total cost (MTC) in Eq. (8), respectively, across ten runs. Here, $f_{\min}$ and f_{best} refer to the minimum total cost obtained by an algorithm on an instance across 10 runs and the best solution found by all algorithms across 10 runs, respectively. In the HW benchmarks, solutions with smaller MTC and Gap values imply lower costs to complete all orders, signifying excellent performance.

Table 2 reveals that MOEA/D-ES significantly outperforms the other three algorithms on small-scale instances of the HW benchmarks. MOEA/D-ES attains the best results on 26 out of 32 instances, whereas the gold-winning, silver-winning, and bronze-winning algorithms achieve the best results on only 6, 0, and 0 out of 32 instances, respectively. The gaps of the six instances on which MOEA/D-ES performs worse than the best solution are relatively small, indicating the robustness of MOEA/D-ES and its capability to reduce optimality gaps. From the "Avg" perspective, the average MTC value of MOEA/D-ES is 5.35E + 3, while the average MTC values of the other three competitors are 1.14E + 4, 1.80E + 5, and 2.50E + 4. The average MTC value obtained by MOEA/D-ES is one order of magnitude smaller than that obtained by the other three competitors, further demonstrating the advantages of MOEA/D-ES. In summary, when dealing with small-scale HW benchmarks, MOEA/D-ES is more efficient than the other three competitors.

4.5.2. Results on medium-scale HW benchmarks

All compared algorithms aim to minimize the TC value of the practical DPDP in Eq. (8). As the problem becomes medium-scale (HW33-HW48), new orders become densely distributed at 144 intervals per day, making it challenging to find the optimal solution. Table 3 presents the best and average TC values of MOEA/D-ES and the three compared algorithms on medium-scale HW benchmarks. The efficacy of MOEA/D-ES can be evaluated from two perspectives: the best and average TC values obtained by MOEA/D-ES in ten independent runs. MOEA/D-ES found the best solution on 13 out of 16 instances, while the gold-winning and silver-winning algorithms could find only 2 and 1 best solutions, respectively. This significant outperformance of MOEA/D-ES is demonstrated by the minimum value of TC. Moreover, the average MTC values of MOEA/D-ES for HW33-HW48 are evidently better than those obtained by the other three algorithms. Specifically, the average MTC value of MOEA/D-ES is 1.65E + 4, while the average MTC values of the other three competitors are 2.65E + 4, 8.53E + 4, and 3.73E + 5. Given that the gold-winning, silver-winning, and bronze-winning algorithms are all single-solution-based algorithms, they may become susceptible to local optima as the complexity of the DPDP increases, which leads to suboptimal solutions. In contrast, MOEAD-ES is a population-based algorithm that preserves diversity in the early stages to avoid local optima and leverages local search in the later stages to discover superior solutions.

4.5.3. Results on large-scale HW benchmarks

In each of the large-scale instances (HW49-HW64) presented in Fig. 3, new orders are densely distributed at 144 intervals per day, and the number of new orders is high at most intervals, making it difficult to find the optimal solution that minimizes the TC value in Eq. (8). Table 4 presents the best and average TC values of MOEA/D-ES and the three compared algorithms on the large-scale HW benchmarks. Notably, MOEA/D-ES found the best solutions on 10 out of 16 instances during ten independent runs, which significantly outperformed the other three algorithms. Moreover, on the large-scale HW benchmarks, the average MTC values of MOEA/D-ES, the gold-winning, silver-winning, and bronze-winning algorithms are 5.52E + 6, 5.67E + 6, 9.79E + 6, and 8.53E + 6, respectively. These results demonstrate the stability of MOEA/D-ES, even when dealing with more challenging large-scale instances. Furthermore, MOEA/D-ES performs consistently better as the problem size increases due to the diversity of solutions provided by its multiobjective approach, which allows it to escape from the local optimal regions of complex and large-scale problems.

The results presented above demonstrate that MOEA/D-ES is a more promising solution for solving the DPDP in HW benchmarks compared to the other three competitors. MOEA/D-ES outperforms the other algorithms due to its ability to balance exploration and exploitation effectively. In contrast to the other competitors that use only one solution throughout the optimization period, MOEA/D-ES first initializes a population of N solutions with different weights, ensuring the diversity of the population. Additionally, the evaluation criterion for total cost in Eq. (8) in the crossover procedure enables the offspring solutions to locate around the global optimal region. The diversity brought by applying different weight vectors to different offspring solutions helps them jump out of local optimality quickly. Moreover, the population in MOEA/D-ES can quickly converge to the optimal solution region, as different subproblems cooperate with each other during the procedures such as updating the reference point and neighboring solutions.

In other words, the remarkable advantages of MOEA/D-ES are achieved due to its ability to explore the feasible search space efficiently, gain high-diversity solutions through independent exploitation and collaboration of N subproblems, and identify more promising regions with a higher probability of finding superior solutions. Therefore, MOEA/D-ES is highly efficient when dealing with HW benchmarks.

Table 3
Comparison between MOEA/D-ES, Gold-winning, Silver-winning, and Bronze-winning algorithms on the medium-scale instances of the HW benchmarks over ten runs ("MTC" represents the minimum value of TC in Eq. (8) across ten runs, "Gap" indicates the difference between the solution with the minimum TC obtained by an algorithm across 10 runs and the best solution among all algorithms across 10 runs).

instance	MOEA/D-ES		Gold		Silver		Bronze	
	MTC	Gap	MTC	Gap	MTC	Gap	MTC	Gap
HW33	**1.39E + 3**	**0.0000**	1.59E + 3	0.1445	7.03E + 3	4.0660	9.50E + 4	67.459
HW34	**6.32E + 3**	**0.0000**	1.05E + 4	0.6596	1.05E + 4	0.6603	5.44E + 4	7.6021
HW35	**8.12E + 1**	**0.0000**	3.44E + 3	41.35	1.58E + 4	193.47	1.04E + 5	1279.0
HW36	**1.34E + 4**	**0.0000**	1.75E + 4	0.3038	2.32E + 4	0.7264	1.13E + 5	7.4088
HW37	1.15E + 4	1.2293	1.12E + 4	1.1845	**5.14E + 3**	**0.0000**	9.57E + 4	17.618
HW38	**1.42E + 4**	**0.0000**	1.50E + 4	0.0617	3.01E + 4	1.1243	9.42E + 4	5.6482
HW39	1.74E + 4	0.1675	**1.49E + 4**	**0.0000**	2.23E + 4	0.4947	8.52E + 4	4.7107
HW40	1.31E + 4	0.2881	**1.02E + 4**	**0.0000**	2.43E + 4	1.3819	1.17E + 5	10.468
HW41	**2.83E + 4**	**0.0000**	2.93E + 4	0.0360	1.24E + 5	3.3880	5.49E + 5	18.4276
HW42	**2.09E + 4**	**0.0000**	5.29E + 4	1.5345	1.64E + 5	6.8623	6.01E + 5	27.8125
HW43	**2.86E + 4**	**0.0000**	6.98E + 4	1.4384	1.67E + 5	4.8378	5.51E + 5	18.2611
HW44	**3.62E + 4**	**0.0000**	7.78E + 4	1.1513	1.91E + 5	4.2797	7.15E + 5	18.7643
HW45	**2.28E + 4**	**0.0000**	3.63E + 4	0.5917	1.81E + 5	6.9433	6.16E + 5	26.0336
HW46	**1.70E + 4**	**0.0000**	3.09E + 4	0.8128	1.76E + 5	9.3420	7.98E + 5	45.8914
HW47	**2.08E + 4**	**0.0000**	2.68E + 4	0.2908	1.04E + 5	4.0061	6.03E + 5	28.0259
HW48	**2.83E + 4**	**0.0000**	2.93E + 4	0.0360	1.24E + 5	3.3880	5.49E + 5	18.4276
best/all	13/16		2/16		1/16		0/16	
Avg	1.65E + 4		2.65E + 4		8.53E + 4		3.73E + 5	

Table 4
Comparison between MOEA/D-ES, Gold-winning, Silver-winning, and Bronze-winning algorithms on the large-scale instances of the HW benchmarks over ten runs ("MTC" represents the minimum value of TC in Eq. (8) across ten runs, "Gap" indicates the difference between the solution with the minimum TC obtained by an algorithm across 10 runs and the best solution among all algorithms across 10 runs).

instance	MOEA/D-ES		Gold		Silver		Bronze	
	MTC	Gap	MTC	Gap	MTC	Gap	MTC	Gap
HW49	**6.30E + 5**	**0.0000**	1.58E + 6	1.5102	1.18E + 6	0.8729	1.59E + 6	1.5236
HW50	8.39E + 5	0.4664	9.31E + 5	0.6283	**5.72E + 5**	**0.0000**	2.51E + 6	3.3881
HW51	**6.46E + 4**	**0.0000**	7.44E + 4	0.1514	5.15E + 5	6.9685	1.34E + 6	19.7337
HW52	4.83E + 5	0.0775	7.61E + 5	0.6983	**4.48E + 5**	**0.0000**	2.03E + 6	3.5313
HW53	**4.98E + 4**	**0.0000**	8.47E + 4	0.6986	3.17E + 5	5.3603	1.93E + 6	37.7234
HW54	1.63E + 6	0.0013	2.07E + 6	0.2680	**1.63E + 6**	**0.0000**	2.00E + 6	0.2270
HW55	1.00E + 6	2.1026	**3.23E + 5**	**0.0000**	7.30E + 5	1.2612	1.98E + 6	5.1331
HW56	**1.60E + 6**	**0.0000**	2.08E + 6	0.2944	1.84E + 6	0.1477	2.16E + 6	0.3473
HW57	**8.95E + 6**	**0.0000**	1.04E + 7	0.1623	2.24E + 7	1.5039	1.65E + 7	0.8444
HW58	**8.52E + 6**	**0.0000**	1.00E + 7	0.1740	1.32E + 7	0.5489	1.12E + 7	0.3142
HW59	9.61E + 6	0.2360	**7.77E + 6**	**0.0000**	1.24E + 7	0.5951	9.54E + 6	0.2272
HW60	**1.03E + 7**	**0.0000**	1.07E + 7	0.0397	2.15E + 7	1.0912	1.32E + 7	0.2839
HW61	**5.69E + 6**	**0.0000**	6.03E + 6	0.0587	9.10E + 6	0.5990	7.21E + 6	0.2669
HW62	1.01E + 7	0.1437	**8.80E + 6**	**0.0000**	1.45E + 7	0.6470	1.09E + 7	0.2381
HW63	**1.33E + 7**	**0.0000**	1.33E + 7	0.0035	3.09E + 7	1.3228	2.50E + 7	0.8793
HW64	**1.56E + 7**	**0.0000**	1.57E + 7	0.0091	2.54E + 7	0.6315	2.74E + 7	0.7600
best/all	10/16		3/16		3/16		0/16	
Avg	5.52E + 6		5.67E + 6		9.79E + 6		8.53E + 6	

5. Future directions and conclusions

5.1. Summary of future directions

Although there have been many publications on DPDPs in the last two decades, numerous challenges and open problems in DPDPs still exist, and new application domains are constantly emerging. In this section, we discuss some future directions for DPDPs, focusing on algorithm design and DPDP applications.

In terms of future directions for solving DPDPs, the scientific literature reviewed can be broadly divided into two streams: reactive solution methods and methods incorporating probabilistic information on future events. Without taking into account probabilistic information about future uncertain events, the first stream focuses on developing efficient methods to obtain dynamic solutions quickly. The second stream, on the other hand, incorporates probabilistic information about future events into methods, with typical hot topics being learning-based methods. The two main streams are briefly discussed below.

1) The research directions of reactive solution methods depend on the design of efficient methods with better responsiveness and solution quality for DPDPs in specific scenarios. Reactive solution methods for DPDPs typically apply heuristics, metaheuristics, or their hybrid versions. For example, in [61], a hybrid metaheuristic with GA, VNS, and TS is introduced to tackle a DPDP in a shared logistics platform, which effectively combines the exploration capabilities of GA with the exploitation capabilities of VNS and TS. Therefore, different heuristics or metaheuristics can be selected to design various efficient algorithms according to the characteristics of DPDPs in different scenarios.
2) Learning-based methods for solving DPDP are a hot research topic. Learning-based methods include some approaches in machine learning [116–120] and reinforcement learning [121–123], which are used successfully in various practical applications, such as traffic flow prediction [124], person reidentification [125–131], multi-model problems [132,133], image segmentation [134–138] and classification [139–142]. Traditional operational research-based methods and metaheuristic methods are computationally expensive and require complex domain knowledge during their design. However, learning-based methods can improve the computing efficiency and ease the algorithm design, as justified by [51,110,143–145]. By leveraging the generalization ability of trained models, these methods have demonstrated significant improvements in computing efficiency and obtained solutions with competitive qualities compared with the state-of-the-art traditional methods. Although learning-based methods have mainly focused on tackling TSPs or VRPs, learning-based methods still show great potential to help solve DPDPs, which is a research topic worthy of further study.

In terms of future directions for applications of DPDPs, this section discusses new business models and logistics concepts that have emerged in recent years, presenting new operational challenges and opportunities for companies. Examples of such applications include online food ordering and delivery services, last mile delivery services, and online retail services. Some of these services have high market value and involve multiple competitors, forcing companies to provide high-quality service while maintaining profitable logistics operations. In the following, we present the opportunities associated with these applications.

1) The DPDP about online food ordering and delivery services is everywhere in large cities, in which food delivery is competed by different companies. This application of DPDP greatly promotes employment while making people's lives more convenient. However, there are disputes over the high quality of service and low delivery prices regarding this problem. There are few scientific papers that address the problem, although it is of practical importance to the operational problem behind the business. There are only a few papers [8,9,42,146] that tackle the problem we find in the literature, none of which considers uncertainty and anticipatory information (such as meal preparation times and delivery orders) through learning-based methods. Therefore, future studies could anticipate delivery information through learning-based algorithms, such as transfer learning or reinforcement learning.
2) Last mile delivery is the transportation of goods from a distribution hub to the final delivery destination, which aims to deliver packages affordably, quickly, and accurately. Last mile delivery has been a widely applied concept in recent years, benefiting courier service companies. However, little research in this stream is related to DPDPs, except for [43,53]. Therefore, DPDP applications in last mile delivery can be a future direction due to their commercial value and academic significance.
3) Recently, online retailer companies have undergone a transformation in their logistics operations to meet the expectations of new customers related to delivery services. Due to the short response time, uncertain and dynamic nature of customers' orders brought

about by recent demands (e.g., same-day delivery and next-day delivery), these companies face operational challenges. In past years, several articles have addressed same-day delivery [146] and attended home delivery [147,148]. However, there are many opportunities and challenges for authors to design better dispatch algorithms, such as order anticipation, postponement and buffer strategies.

5.2. Conclusions

Thanks to faster algorithms and efficient real-time communication technologies, solutions rooted in real-world DPDP applications can now be practically implemented. The volume of research on DPDPs has significantly increased over the last two decades, and useful concepts such as solution methods, dynamism measuring, and anticipation of future requests have been proposed. In this paper, we provide a comprehensive review of the DPDP literature of the past twenty years, discussing the corresponding characteristics, principles, and theoretical analysis in detail. Additionally, we introduce a taxonomy of the related solution methods, categorizing them into four kinds: exact methods, heuristics, metaheuristics, and learning-based methods. To study the characteristics of practical DPDPs, we conduct experimental comparisons and analyse recent real-world DPDP benchmarks from Huawei. Finally, we briefly illustrate some future directions for DPDPs.

CRediT authorship contribution statement

Junchuang Cai: Methodology, Writing – original draft, Software, Data curation. **Qingling Zhu:** Methodology, Validation, Conceptualization. **Qiuzhen Lin:** Supervision, Validation, Conceptualization. **Lijia Ma:** Conceptualization. **Jianqiang Li:** Conceptualization. **Zhong Ming:** Supervision.

Declaration of Competing Interest

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

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported in part by the Guangdong Regional Joint Foundation Key Project under Grant 2022B1515120076, in part by Natural Science Foundation of Guangdong Province under Grant 2023A1515011238, in part by the Shenzhen Science and Technology Program under Grants JCYJ20220531101411027, JCYJ20190808164211203, and JCYJ20190808174801673, in part by the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603, in part by the Guangdong "Pearl River Talent Plan" under Grant 2019JC01X235, in part by the Shenzhen Science and Technology Innovation Commission under Grant R2020A045, in part by the National Natural Science Foundation of China under Grant 62173236 and 62203308, and in part by the China Postdoctoral Science Foundation under Grant 2022M712170.

References

[1] M.M. Flood, The traveling-salesman problem, Oper. Res. 4 (1) (1956) 61–75.
[2] G.B. Dantzig, J.H. Ramser, The truck dispatching problem, Manage. Sci. 6 (1) (1959) 80–91.
[3] J.-F. Cordeau, G. Laporte, S. Ropke, Recent models and algorithms for one-to-one pickup and delivery problems, in: The vehicle routing problem: latest advances and new challenges, Springer, 2008, pp. 327–357.
[4] K. Fagerholt, B. Foss, O. Horgen, A decision support model for establishing an air taxi service: a case study, J. Oper. Res. Soc. 60 (9) (2009) 1173–1182.
[5] M. Schilde, K.F. Doerner, R.F. Hartl, Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports, Comput. Oper. Res. 38 (12) (2011) 1719–1730.
[6] M. Schilde, K.F. Doerner, R.F. Hartl, Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem, Eur. J. Oper. Res. 238 (1) (2014) 18–30.
[7] X. Cheng, S. Liao, Z. Hua, A policy of picking up parcels for express courier service in dynamic environments, Int. J. Prod. Res. 55 (9) (2017) 2470–2488.
[8] D. Reyes, A. Erera, M. Savelsbergh, S. Sahasrabudhe, R. O'Neil, The meal delivery routing problem, Optimization Online 6571 (2018).
[9] M.W. Ulmer, B.W. Thomas, A.M. Campbell, N. Woyak, The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times, Transp. Sci. 55 (1) (2021) 75–100.
[10] U. Ritzinger, J. Puchinger, R.F. Hartl, A survey on dynamic and stochastic vehicle routing problems, Int. J. Prod. Res. 54 (1) (2016) 215–231.
[11] A. Larsen, O. Madsen, M. Solomon, Partially dynamic vehicle routing—models and algorithms, J. Oper. Res. Soc. 53 (6) (2002) 637–646.
[12] L. Brotcorne, G. Laporte, F. Semet, Ambulance location and relocation models, Eur. J. Oper. Res. 147 (3) (2003) 451–463.
[13] G. Ghiani, F. Guerriero, G. Laporte, R. Musmanno, Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies, Eur. J. Oper. Res. 151 (1) (2003) 1–11.
[14] F.T. Hanshar, B.M. Ombuki-Berman, Dynamic vehicle routing using genetic algorithms, Appl. Intell. 27 (1) (2007) 89–99.
[15] S. Ichoua, M. Gendreau, J.-Y. Potvin, Planned route optimization for real-time vehicle routing, in: Dynamic Fleet Management, 2007, pp. 1–18.
[16] A. Larsen, O.B.G. Madsen, M.M. Solomon, Recent developments in dynamic vehicle routing systems, in: The Vehicle Routing Problem: Latest Advances and New Challenges, 2008, pp. 199–218.
[17] V. Pillac, M. Gendreau, C. Guéret, A.L. Medaglia, A review of dynamic vehicle routing problems, Eur. J. Oper. Res. 225 (1) (2013) 1–11.
[18] T. Bekta¸ s, P.P. Repoussis, C.D. Tarantilis, Chapter 11: Dynamic vehicle routing problems, in: *Vehicle, Routing: Problems, Methods, and Applications*, Second Edition, SIAM, 2014, pp. 299–347.
[19] A. Larsen, O. B. Madsen, and M. M. Solomon, Classification of dynamic vehicle routing systems, *Dynamic Fleet Management*. (2007) 19-40.
[20] G. Berbeglia, J.-F. Cordeau, G. Laporte, Dynamic pickup and delivery problems, Eur. J. Oper. Res. 202 (1) (2010) 8–15.
[21] H.N. Psaraftis, Dynamic vehicle routing problems, Vehicle routing: Methods and studies 16 (1988) 223–248.
[22] V. Nenchev, C.G. Cassandras, J. Raisch, Event-driven optimal control for a robotic exploration, pick-up and delivery problem, Nonlinear Anal. Hybrid Syst 30 (2018) 266–284.
[23] E. Osaba, X.-S. Yang, I. Fister Jr, J. Del Ser, P. Lopez-Garcia, A.J. Vazquez-Pardavila, A discrete and improved bat algorithm for solving a medical goods distribution problem with pharmacological waste collection, Swarm Evol. Comput. 44 (2019) 273–286.
[24] K. Treleaven, M. Pavone, E. Frazzoli, Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems, IEEE Trans. Autom. Control 58 (9) (2013) 2261–2276.
[25] S. Anily, R. Hassin, The swapping problem, Networks 22 (4) (1992) 419–433.
[26] H. Min, The multiple vehicle routing problem with simultaneous delivery and pick-up points, Transp. Res. Part A: Gen. 23 (5) (1989) 377–386.
[27] Ç. Koç, G. Laporte, İ. Tükenmez, A review of vehicle routing with simultaneous pickup and delivery, Comput. Oper. Res. 122 (2020), 104987.
[28] M. Savelsbergh, M. Sol, Drive: Dynamic routing of independent vehicles, Oper. Res. 46 (4) (1998) 474–490.
[29] S. Mitrović-Minić, G. Laporte, Waiting strategies for the dynamic pickup and delivery problem with time windows, Transp. Res. B Methodol. 38 (7) (2004) 635–655.
[30] S. Mitrović-Minić, R. Krishnamurti, G. Laporte, Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows, Transp. Res. B Methodol. 38 (8) (2004) 669–685.
[31] B. Sun, Y. Yang, J. Shi, L. Zheng, Dynamic pick-up and delivery optimization with multiple dynamic events in real-world environment, IEEE Access 7 (2019) 146209–146220.
[32] D. Sáez, C.E. Cortés, A. Núñez, Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering, Comput. Oper. Res. 35 (11) (2008) 3412–3438.
[33] D. Muñoz-Carpintero, D. Sáez, C.E. Cortés, A. Núñez, A methodology based on evolutionary algorithms to solve a dynamic pickup and delivery problem under a hybrid predictive control approach, Transp. Sci. 49 (2) (2015) 239–253.
[34] M. Gendreau, F. Guertin, J.-Y. Potvin, R. Séguin, Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries, Transp. Res. Part C: Emerg. Technol. 14 (3) (2006) 157–174.
[35] V. Pureza, G. Laporte, Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows, INFOR: Inf. Syst. Oper. Res. 46 (3) (2008) 165–175.
[36] S. Vonolfen, M. Affenzeller, Distribution of waiting time for dynamic pickup and delivery problems, Ann. Oper. Res. 236 (2) (2016) 359–382.
[37] J. Cai, Q. Zhu, Q. Lin, Variable neighborhood search for a new practical dynamic pickup and delivery problem, Swarm Evol. Comput. 75 (2022), 101182.

[38] K. Lund, O.B. Madsen, J.M. Rygaard. Vehicle routing problems with varying degrees of dynamism, IMM, Institute of Mathematical Modelling, Technical University of Denmark, 1996.
[39] A. Larsen, O. Madsen. The dynamic vehicle routing problem, Institute of Mathematical Modelling, Technical University of Denmark, 2000.
[40] H.N. Psaraftis, Dynamic vehicle routing: Status and prospects, Ann. Oper. Res. 61 (1) (1995) 143–164.
[41] P. Jaillet, M.R. Wagner, Online routing problems: Value of advanced information as improved competitive ratios, Transp. Sci. 40 (2) (2006) 200–210.
[42] Z. Steever, M. Karwan, C. Murray, Dynamic courier routing for a food delivery service, Comput. Oper. Res. 107 (2019) 173–188.
[43] C. Fikar, P. Hirsch, M. Gronalt, A decision support system to investigate dynamic last-mile distribution facilitating cargo-bikes, Int J Log Res Appl 21 (3) (2018) 300–317.
[44] A.M. Arslan, N. Agatz, L. Kroon, R. Zuidwijk, Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers, Transp. Sci. 53 (1) (2019) 222–235.
[45] W. Hu, J. Mao, K. Wei, Energy-efficient rail guided vehicle routing for two-sided loading/unloading automated freight handling system, Eur. J. Oper. Res. 258 (3) (2017) 943–957.
[46] F. Karami, W. Vancroonenburg, G. Vanden Berghe. A buffering-strategy-based re-optimization approach to dynamic pickup and delivery problems with time windows, 2018, in: http://www.patatconference.org/patat2018/proceedings/.
[47] Z. K. Fkaier, and B. F. Chaar, Online K-means based heuristic for the dynamic pickup and delivery problem solving, in: *World Congress on Computer and Information Technology*, (2013) 1-6.
[48] G. Tirado, L.M. Hvattum, Improved solutions to dynamic and stochastic maritime pick-up and delivery problems using local search, Annals of Operations Research 253 (2) (2017) 825–843.
[49] A. Berahhou, Y. Benadada, K. Bouanane, Memetic algorithm for the dynamic vehicle routing problem with simultaneous delivery and pickup, Int. J. Ind. Eng. Comput. 13 (4) (2022) 587–600.
[50] J. Cai, Q. Zhu, Q. Lin, J. Li, J. Chen, and Z. Ming, An efficient multi-objective evolutionary algorithm for a practical dynamic pickup and delivery problem, in: *International Conference on Intelligent Computing*, (2022) 27-40.
[51] Y. Ma, X. Hao, J. Hao, J. Lu, X. Liu, T. Xialiang, M. Yuan, Z. Li, J. Tang, Z. Meng, A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems, Adv. Neural Inf. Proces. Syst. 34 (2021) 23609–23620.
[52] M.A.N. Andini, Y. Satria, H. Burhan, Dynamic pickup and delivery problem with transfer in ridesharing to reduce congestion, J. Phys. Conf. Ser. 1218 (1) (2019).
[53] C. Fikar, A decision support system to investigate food losses in e-grocery deliveries, Comput. Ind. Eng. 117 (2018) 282–290.
[54] A. Berahhou, and Y. Benadada, Dynamic vehicle routing problem with simultaneous delivery and pickup: formulation and resolution, in: *Fifth International Conference on Logistics Operations Management*, (2020) 1-8.
[55] T. Geiser, T. Hanne, R. Dornberger. Best-match in a set of single-vehicle dynamic pickup and delivery problem using ant colony optimization, Business, 2020, pp. 126–131.
[56] T. Andersson, A comparative study on a dynamic pickup and delivery problem: improving routing and order assignment in same-day courier operations, 2021.
[57] Y. Jia, C. Wang, L. Wang. A rolling horizon procedure for dynamic pickup and delivery problem with time windows, 2009, pp. 2087–2091.
[58] G. Ghiani, E. Manni, A. Quaranta, C. Triki, Anticipatory algorithms for same-day courier dispatching, Transportation Research Part E: Logistics and Transportation Review 45 (1) (2009) 96–106.
[59] M. Mes, M. van der Heijden, P. Schuur, Look-ahead strategies for dynamic pickup and delivery problems, OR Spectr. 32 (2) (2010) 395–421.
[60] D. Bertsimas, P. Jaillet, S. Martin, Online vehicle routing: The edge of optimization in large-scale applications, Oper. Res. 67 (1) (2019) 143–162.
[61] Z. Su, W. Li, J. Li, B. Cheng, Heterogeneous fleet vehicle scheduling problems for dynamic pickup and delivery problem with time windows in shared logistics platform: formulation, instances and algorithms, International Journal of Systems Science: Operations & Logistics 9 (2) (2022) 199–223.
[62] J. Mo, S. Ohmori, Crowd sourcing dynamic pickup & delivery problem considering task buffering and drivers' rejection-application of multi-agent reinforcement learning, WSEAS Trans. Bus. Econ. 18 (2021) 636–645.
[63] S.A. Voccia, A.M. Campbell, B.W. Thomas, The same-day delivery problem for online purchases, Transp. Sci. 53 (1) (2019) 167–184.
[64] I.H. Dridi, E.B. Alaia, P. Borne. Heuristic approach for the optimization of the dynamic multi-vehicle pickup and delivery problem with time windows, 2015, pp. 733–739.
[65] I. H. Dridi, R. Kammarti, P. Borne, and M. Ksouri, Multi-objective optimization for the dynamic multi-pickup and delivery problem with time windows. *arXiv preprint arXiv:1101.3396*, 2011.
[66] Z. Zhu, J. Xiao, S. He, Z. Ji, Y. Sun, A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem, Inf. Sci. 329 (2016) 73–89.
[67] J. Xiao, Y. Yang, X. Ma, J. Zhou, and Z. Zhu, Multi-objective memetic algorithm for solving pickup and delivery problem with dynamic customer requests and traffic information, in: *2016 IEEE Congress on Evolutionary Computation*, (2016) 1964-1970.
[68] F. Wang, Y. Gao, and Z. Zhu, Locality-sensitive hashing based multiobjective memetic algorithm for dynamic pickup and delivery problems, in: *2014 IEEE Congress on Evolutionary Computation*, (2014) 661-666.
[69] A. Núñez, B. De Schutter, D. Sáez, and C. E. Cortés, Hierarchical multiobjective model predictive control applied to a dynamic pickup and delivery problem, in: *13th International IEEE Conference on Intelligent Transportation Systems*, (2010) 1553-1558.
[70] C.E. Cortés, D. Sáez, A. Núñez, D. Muñoz-Carpintero, Hybrid adaptive predictive control for a dynamic pickup and delivery problem, Transp. Sci. 43 (1) (2009) 27–42.
[71] C.E. Cortés, A. Núñez, D. Sáez, Hybrid adaptive predictive control for a dynamic pickup and delivery problem including traffic congestion, Int. J. Adapt Control Signal Process. 22 (2) (2008) 103–123.
[72] P.V. Hentenryck, R. Bent, Online stochastic combinatorial optimization, The MIT Press, 2006.
[73] L.M. Hvattum, A. Løkketangen, G. Laporte, Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic, Transp. Sci. 40 (4) (2006) 421–438.
[74] L.M. Hvattum, A. Løkketangen, G. Laporte, A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems, Networks: An Int. J. 49 (4) (2007) 330–340.
[75] J. Hao, J. Lu, X. Li, X. Tong, X. Xiang, M. Yuan, and H. H. Zhuo, Introduction to the dynamic pickup and delivery problem benchmark–ICAPS 2021 competition, *arXiv preprint arXiv:2202.01256*, 2022.
[76] A. Attanasio, J. Bregman, G. Ghiani, E. Manni, Real-time fleet management at Ecourier Ltd, in: Dynamic Fleet Management, Dynamic Fleet Management, Springer US, Boston, MA, 2007, pp. 219–238.
[77] R. H. Möhring, E. Köhler, E. Gawrilow, and B. Stenzel, Conflict-free real-time AGV routing, in: *Operations research proceedings 2004*, (2005) 18-24.
[78] N.F. Cass, E.A. Shove, I. Jones, G. Marsden. The Last Mile and the Next Day, The changing times and spaces of shopping–implications for energy demand, 2017.
[79] A. Haghani, Q. Tian, H. Hu, Simulation model for real-time emergency vehicle dispatching and routing, Transp. Res. Rec. 1882 (1) (2004) 176–183.
[80] G. Ghiani, A. Manni, E. Manni, A scalable anticipatory policy for the dynamic pickup and delivery problem, Comput. Oper. Res. 147 (2022), 105943.
[81] J. Branke, M. Middendorf, G. Noeth, M. Dessouky, Waiting strategies for dynamic vehicle routing, Transp. Sci. 39 (3) (2005) 298–312.
[82] S. Ichoua, M. Gendreau, J.-Y. Potvin, Exploiting knowledge about future demands for real-time vehicle dispatching, Transp. Sci. 40 (2) (2006) 211–225.
[83] B.W. Thomas, Waiting strategies for anticipating service requests from known customer locations, Transp. Sci. 41 (3) (2007) 319–331.
[84] F. Karami, W. Vancroonenburg, G. Vanden Berghe, A periodic optimization approach to dynamic pickup and delivery problems with time windows, J. Sched. 23 (6) (2020) 711–731.
[85] S. Liu, P. H. Tan, E. Kurniawan, P. Zhang, and S. Sun, Dynamic scheduling for pickup and delivery with time windows, in: *2018 IEEE 4th World Forum on Internet of Things*, (2018) 767-770.
[86] K.J. Roodbergen, I.F. Vis, A survey of literature on automated storage and retrieval systems, Eur. J. Oper. Res. 194 (2) (2009) 343–362.
[87] J. Lee, Dispatching rail-guided vehicles and scheduling jobs in a flexible manufacturing system, Int. J. Prod. Res. 37 (1) (1999) 111–123.
[88] S.O. Kimbrough, H.C. Lau, Business analytics for decision making, Chapman and Hall/CRC, 2018.
[89] K. Sörensen, F.W. Glover, Metaheuristics, in: S.I. Gass, M.C. Fu (Eds.), Encyclopedia of Operations Research and Management Science, Springer US, Boston, MA, 2013, pp. 960–970.
[90] N. Mladenović, P. Hansen, Variable neighborhood search, Comput. Oper. Res. 24 (11) (1997) 1097–1100.
[91] P. Hansen, N. Mladenović, An introduction to variable neighborhood search, Meta-heuristics. (1999) 433–458.
[92] P. Hansen, N. Mladenović, Variable neighborhood search: Principles and applications, Eur. J. Oper. Res. 130 (3) (2001) 449–467.
[93] B. Sarasola, K.F. Doerner, V. Schmid, E. Alba, Variable neighborhood search for the stochastic and dynamic vehicle routing problem, Annals of Operations Research 236 (2) (2016) 425–461.
[94] D. Trachanatzi, M. Rigakis, M. Marinaki, Y. Marinakis, A firefly algorithm for the environmental prize-collecting vehicle routing problem, Swarm Evol. Comput. 57 (2020), 100712.
[95] N.R. Sabar, A. Bhaskar, E. Chung, A. Turky, A. Song, A self-adaptive evolutionary algorithm for dynamic vehicle routing problems with traffic congestion, Swarm Evol. Comput. 44 (2019) 1018–1027.
[96] M. Okulewicz, J. Mańdziuk, A metaheuristic approach to solve dynamic vehicle routing problem in continuous search space, Swarm Evol. Comput. 48 (2019) 44–61.
[97] F. Glover, Future paths for integer programming and links to artificial intelligence, Comput. Oper. Res. 13 (5) (1986) 533–549.
[98] S. Luke, Essentials of metaheuristics, Lulu Raleigh, 2013.
[99] M. Mitchell, An introduction to genetic algorithms, MIT press, 1998.
[100] A.G. Novaes, E.T. Bez, P.J. Burin, D.P. Aragão Jr, Dynamic milk-run OEM operations in over-congested traffic conditions, Comput. Ind. Eng. 88 (2015) 326–340.
[101] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, *Caltech concurrent computation program, C3P Report,* (1989) 37.
[102] E. Behmanesh, J. Pannek, A comparison between memetic algorithm and genetic algorithm for an integrated logistics network with flexible delivery path, in: Operations Research Forum, Springer, 2021, pp. 1–24.

[103] A. Beham, M. Kofler, S. Wagner, M. Affenzeller, Agent-based simulation of dispatching rules in dynamic pickup and delivery problems, in: *2009 2nd International Symposium on Logistics and Industrial Informatics*, IEEE, 2009, pp. 1–6.
[104] J. Hu, A.E. Turgut, T. Krajník, B. Lennox, F. Arvin, Occlusion-based coordination protocol design for autonomous robotic shepherding tasks, IEEE Trans. Cognit. Dev. Syst. 14 (1) (2020) 126–135.
[105] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: Proceedings of the first European conference on artificial life, 1991, pp. 134–142.
[106] J. Kennedy, and R. Eberhart, Particle swarm optimization, in: *Proceedings ofICNN'95-international conference on neural networks*, 4 (1995) 1942-1948.
[107] K.A. Smith, Neural networks for combinatorial optimization: a review of more than a decade of research, INFORMS J. Comput. 11 (1) (1999) 15–34.
[108] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (11) (2018) e00938.
[109] B. Li, G. Wu, Y. He, M. Fan, W. Pedrycz, An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem, IEEE/CAA J. Autom. Sin. 9 (7) (2022) 1115–1138.
[110] X. Li, W. Luo, M. Yuan, J. Wang, J. Lu, J. Wang, J. Lü, J. Zeng, Learning to optimize industry-scale dynamic pickup and delivery problems, in: *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 2511–2522.
[111] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, L. Song, Learning combinatorial optimization algorithms over graphs, Adv. Neural Inf. Proces. Syst. 30 (2017).
[112] W. Kool, H. Van Hoof, M. Welling, Attention, learn to solve routing problems!. arXiv preprint arXiv:1803.08475, 2018.
[113] B. Balaji, J. Bell-Masterson, E. Bilgin, A. Damianou, P. M. Garcia, A. Jain, R. Luo, A. Maggiar, B. Narayanaswamy, and C. Ye, Orl: Reinforcement learning benchmarks for online stochastic optimization problems, *arXiv preprint arXiv: 1911.10641*, 2019.
[114] R.R. Van Lon, E. Ferrante, A.E. Turgut, T. Wenseleers, G.V. Berghe, T. Holvoet, Measures of dynamism and urgency in logistics, Eur. J. Oper. Res. 253 (3) (2016) 614–624.
[115] L. Zigranda, R. W. Calvob, E. Traversib, and P. Alizadehd, A Scalable Logic-based Benders Decomposition to Optimize a Dynamic Demand-Responsive Transport System, 2022.
[116] Y. Guo, Z. Wu, D. Shen, Learning longitudinal classification-regression model for infant hippocampus segmentation, Neurocomputing 391 (2020) 191–198.
[117] Z. Zhang, S. Ding, Y. Sun, A support vector regression model hybridized with chaotic krill herd algorithm and empirical mode decomposition for regression task, Neurocomputing 410 (2020) 185–201.
[118] J.-X. Du, D.-S. Huang, G.-J. Zhang, Z.-F. Wang, A novel full structure optimization algorithm for radial basis probabilistic neural networks, Neurocomputing 70 (1–3) (2006) 592–596.
[119] J.-X. Du, D.-S. Huang, X.-F. Wang, X. Gu, Shape recognition based on neural networks trained by differential evolution algorithm, Neurocomputing 70 (4–6) (2007) 896–903.
[120] F. Han, Q.-H. Ling, D.-S. Huang, An improved approximation approach incorporating particle swarm optimization and a priori information into neural networks, Neural Comput. & Applic. 19 (2) (2010) 255–261.
[121] J. Huang, L. Huang, M. Liu, H. Li, Q. Tan, X. Ma, J. Cui, D.-S. Huang, Deep reinforcement learning-based trajectory pricing on ride-hailing platforms, ACM Transactions on Intelligent Systems and Technology (TIST) 13 (3) (2022) 1–19.
[122] P. Chen, J. Pei, W. Lu, M. Li, A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance, Neurocomputing 497 (2022) 64–75.
[123] N. Wang, Y. Gao, C. Yang, X. Zhang, Reinforcement learning-based finite-time tracking control of an unknown unmanned surface vehicle with input constraints, Neurocomputing 484 (2022) 26–37.
[124] H. Li, X. Li, L. Su, D. Jin, J. Huang, D. Huang, Deep spatio-temporal adaptive 3d convolutional neural networks for traffic flow prediction, ACM Transactions on Intelligent Systems and Technology (TIST) 13 (2) (2022) 1–21.
[125] Y. Wu, K. Zhang, D. Wu, C. Wang, C.-A. Yuan, X. Qin, T. Zhu, Y.-C. Du, H.-L. Wang, D.-S. Huang, Person reidentification by multiscale feature representation learning with random batch feature mask, IEEE Transactions on Cognitive and Developmental Systems 13 (4) (2020) 865–874.
[126] D. Wu, C. Wang, Y. Wu, Q.-C. Wang, D.-S. Huang, Attention deep model with multi-scale deep supervision for person re-identification, IEEE Transactions on Emerging Topics in Computational Intelligence 5 (1) (2021) 70–78.
[127] D. Wu, S.-J. Zheng, X.-P. Zhang, C.-A. Yuan, F. Cheng, Y. Zhao, Y.-J. Lin, Z.-Q. Zhao, Y.-L. Jiang, D.-S. Huang, Deep learning-based methods for person re-identification: A comprehensive review, Neurocomputing 337 (2019) 354–371.
[128] D. Wu, K. Zhang, F. Cheng, Y. Zhao, Q. Liu, C.-A. Yuan, and D.-S. Huang, Random occlusion-recovery for person re-identification, *arXiv preprint arXiv:1809.09970*, 2018.
[129] D. Wu, H.-W. Yang, D.-S. Huang, C.-A. Yuan, X. Qin, Y. Zhao, X.-Y. Zhao, J.-H. Sun, Omnidirectional feature learning for person re-identification, IEEE Access 7 (2019) 28402–28411.
[130] D. Wu, S.-J. Zheng, C.-A. Yuan, D.-S. Huang, A deep model with combined losses for person re-identification, Cogn. Syst. Res. 54 (2019) 74–82.
[131] D. Wu, S.-J. Zheng, W.-Z. Bao, X.-P. Zhang, C.-A. Yuan, D.-S. Huang, A novel deep model with multi-loss and efficient training for person re-identification, Neurocomputing 324 (2019) 69–75.
[132] J. Zhang, D.-S. Huang, T.-M. Lok, M.R. Lyu, A novel adaptive sequential niche technique for multimodal function optimization, Neurocomputing 69 (16–18) (2006) 2396–2401.
[133] Q. Lin, W. Lin, Z. Zhu, M. Gong, J. Li, C.A.C. Coello, Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces, IEEE Trans. Evol. Comput. 25 (1) (2020) 130–144.
[134] X. Liang, D. Wu, D.-S. Huang, Image co-segmentation via locally biased discriminative clustering, IEEE Trans. Knowl. Data Eng. 31 (11) (2019) 2228–2233.
[135] X. Liang, D.S. Huang, Image segmentation fusion using weakly supervised trace-norm multi-task learning method, IET Image Proc. 12 (7) (2018) 1079–1085.
[136] X. Liang, L. Zhu, D.-S. Huang, Multi-task ranking SVM for image cosegmentation, Neurocomputing 247 (2017) 126–136.
[137] X.-F. Wang, D.-S. Huang, H. Xu, An efficient local Chan-Vese model for image segmentation, Pattern Recogn. 43 (3) (2010) 603–618.
[138] X. Wang, D.-S. Huang, A Novel Multi-Layer Level Set Method for Image Segmentation, J. Univers. Comput. Sci. 14 (14) (2008) 2427–2452.
[139] J.-X. Mi, D.-S. Huang, B. Wang, X. Zhu, The nearest-farthest subspace classification for face recognition, Neurocomputing 113 (2013) 241–250.
[140] K.-H. Liu, D.-S. Huang, Cancer classification using rotation forest, Comput. Biol. Med. 38 (5) (2008) 601–610.
[141] X.-F. Wang, D.-S. Huang, J.-X. Du, H. Xu, L. Heutte, Classification of plant leaf images with complicated background, Appl. Math Comput. 205 (2) (2008) 916–926.
[142] Y. Zhao, D.-S. Huang, W. Jia, Completed local binary count for rotation invariant texture classification, IEEE Trans. Image Process. 21 (10) (2012) 4492–4497.
[143] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, Adv. Neural Inf. Proces. Syst. 28 (2015).
[144] M. Nazari, A. Oroojlooy, L. Snyder, M. Takác, Reinforcement learning for solving the vehicle routing problem, Adv. Neural Inf. Proces. Syst. 31 (2018).
[145] H. Lu, X. Zhang, S. Yang, A learning-based iterative method for solving vehicle routing problems. International conference on learning representations, 2019.
[146] M. Ulmer, Delivery deadlines in same-day delivery, Logist. Res. 10 (3) (2017) 1–15.
[147] R. Klein, J. Mackert, M. Neugebauer, C. Steinhardt, A model-based approximation of opportunity cost for dynamic pricing in attended home delivery, OR Spectr. 40 (4) (2018) 969–996.
[148] X. Yang, A.K. Strauss, An approximate dynamic programming approach to attended home delivery management, Eur. J. Oper. Res. 263 (3) (2017) 935–945.

Junchuang Cai received the B.S. degrees in computer science and technology from the Shaoguan University, Shaoguan, China, in 2019. He is currently pursuing his Ph.D. degree in College of Computer Science and Software engineering, Shenzhen University. His current research interests include evolutionary algorithms, reinforcement learning, evolutionary transfer optimization, and their applications in the field of logistics.

Qingling Zhu received the B.S. degree from the Nanchang Institution of Technology, Nanchang, China, in 2013, the M.S. degree from Shenzhen University, Shenzhen, China, in 2016 and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2021. He is currently a postdoctor in Shenzhen University. His current research interests include evolutionary multiobjective optimization and machine learning.

Qiuzhen Lin (Member, IEEE) received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from De-partment of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014. He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over 100 research papers since 2008. His research interests include artificial immune system, multi-objective optimization, and dynamic system. Dr. Lin is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Lijia Ma received the B.S. degree in communication engineering from Hunan Normal University, Changsha, China, in 2010, and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2015. From 2015 to 2016, he was a Postdoctoral Fellow with Hong Kong Baptist University, Hong Kong, and Nanyang Technological University, Singapore, from 2016 to 2017. He is an Assistant Professor with the College of Computer and Software Engineering, Shenzhen University, Shenzhen, China. His research interests mainly include evolutionary computation, machine learning, and complex networks.

Jianqiang Li (Member, IEEE) received the B.S. and Ph.D. degrees in automation from the South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.He is a Professor with the College of Computer and Software Engineering, Shenzhen University, Shenzhen, China. He led a project of the National Natural Science Foundation and a project of the Natural Science Foundation of Guangdong Province, China. His major research interests include embedded systems and Internet of Things.

Zhong Ming received the Ph.D. degree in computer science and technology from Sun Yat-sen University, Guangzhou, China, in 2004. He is currently the Executive Director of the Graduate School of Shenzhen University, Shenzhen, China, and a Professor with the National Engineering Laboratory for Big Data System Computing Technology and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen. He has published more than 200 refereed international conference and journal papers (including 40+ ACM/IEEE TRANSACTIONS papers). His research interests include software engineering and artificial intelligence. Prof. Ming was the recipient of the ACM TiiS 2016 Best Paper Award and some other best paper awards.