Int. J. Production Economics 256 (2023) 108751

Contents lists available at ScienceDirect

International Journal of Production Economics

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

Review

Dynamic vehicle routing with random requests: A literature review

Jian Zhang [a,*], Tom Van Woensel [b]

[a] Department of Electronic Business, South China University of Technology, Guangzhou 510006, PR China
[b] Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, 5600MB, Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Dynamic vehicle routing
Dynamic requests
Stochastic information
Literature review

ABSTRACT

Stimulated by the growing demand for logistics services and the advances in information technologies, research interest in the dynamic vehicle routing problem with random requests (DVRPRR) increased over recent decades. The DVRPRR differs from classical vehicle routing in that the customer requests are not fully known in advance but arrive dynamically during the execution of planned routes. Various approaches were developed to deal with this dynamism and the potential stochasticity. This paper provides a comprehensive and in-depth review of the existing DVRPRR literature. We propose a novel taxonomy that identifies four DVRPRR variants with different request types and planning horizons. We also analyze the research on each variant from the perspectives of problem settings, decision strategies, solution approaches, etc. Finally, we summarize the state of the art of the DVRPRR and suggest promising directions for future research.

1. Introduction

The vehicle routing problem (VRP) is a generalization of the well-known NP-hard traveling salesman problem (TSP). It is intensively studied since the late 1950s due to its higher complexity compared to the TSP and its importance in improving logistics services. The basic VRP aims to find optimal vehicle routes to serve geographically dispersed customers at the minimum total travel costs. The basic VRP can be extended to different VRP variants with different side constraints. For example, the capacitated VRP considers package size and limited vehicle capacity; the VRP with time windows requires each customer to be visited within a specific time interval; the periodic VRP has a multi-day planning horizon during which each customer must be visited a required number of times.

Most VRP variants, including those mentioned above, are *static and deterministic* (SD): all problem inputs are known beforehand and do not change over time. By contrast, some VRP variants are *static and stochastic* (SS) because they incorporate stochastic parameters that are realized only during route executions. The solution to an SS VRP is usually an a priori route optimized based on the probabilistic information of the stochastic parameters (Pillac et al., 2013; Ritzinger et al., 2016). A typical SS VRP is the VRP with stochastic demands, in which the actual demand of each customer is not revealed until a service vehicle arrives at the customer's location (e.g, Florio et al., 2020a,b). The other common sources of uncertainty involved in SS VRPs include stochastic customers, travel times, and service times (Gendreau et al., 2016).

Although the two categories of static VRPs (SVRPs) account for the majority of problems studied in the vehicle routing literature (Toth and Vigo, 2014; Braekers et al., 2016), in the last decades, dynamic VRPs (DVRPs) attracted increasing attention from the research community (Psaraftis et al., 2016). DVRPs are characterized by the property that the problem inputs are partially known in advance and dynamically change during the operation period. The arising of DVRPs, on the one hand, reflects the real demand from the current logistics and transportation industry. Dynamism is an intrinsic property of many real-world applications, as the exact information related to request arrivals, customer locations, time windows, etc., is not always fully available when the routes are planned. On the other hand, the applications of DVRPs are boosted by the recent advances in communication and information technologies, which allow decision makers to receive updated information in real-time and dynamically adapt the ongoing route plans to the changing environment.

A large amount of literature on the DVRP has evolved during the past 40 years. Pillac et al. (2013) review the applications and solution approaches of DVRPs from the literature up to 2012. They classify DVRPs into two categories according to the information quality regarding the dynamic issues, as presented in Table 1. Specifically, a DVRP is considered to be *dynamic and deterministic* (DD) if the decision maker does not have any probabilistic information on the unknown dynamic events that may occur in the future. In a DD VRP, the route plan is updated based on the revealed information only. On the contrary, if the probability distributions of the dynamic issues are available, the decision maker can use this information to anticipate future events and proactively adapt the planned vehicle routes to potential changes. In this case, the DVRP is *dynamic and stochastic* (DS).

* Corresponding author at: Department of Electronic Business, South China University of Technology, Guangzhou 510006, PR China.
E-mail addresses: jianzhang@scut.edu.cn (J. Zhang), t.v.woensel@tue.nl (T. Van Woensel).

https://doi.org/10.1016/j.ijpe.2022.108751
Received 29 November 2021; Received in revised form 6 October 2022; Accepted 12 December 2022
Available online 19 December 2022
0925-5273/

Table 1
VRP taxonomy by information evolution and quality (Pillac et al., 2013).

		Information quality	
		No probabilistic information	With probabilistic information
Information evolution	Inputs known in advance	Static and deterministic (SD)	Static and stochastic (SS)
	Inputs change dynamically	Dynamic and deterministic (DD)	Dynamic and stochastic (DS)

In a later survey, Bektaş et al. (2014) classify DVRPs based on the sources of dynamism, namely requests, travel times, and vehicle availability, and provide a deep review of the solution approaches for different DVRP variants. Psaraftis et al. (2016) propose a comprehensive DVRP taxonomy with 11 criteria. They review the DVRP papers over three decades, discuss the technological and methodological advances, and suggest directions for further research. More recently, Rios et al. (2021) review the characteristics and solution methods of DVRPs from the literature between 2015 and 2021. While the aforementioned reviews cover both the two categories of DVRPs, Ritzinger et al. (2016) and Soeffker et al. (2022) focus exclusively on the category of DS VRPs. Ritzinger et al. (2016) review the preprocessed and online decision-making approaches for the DS VRPs with three stochastic aspects: travel times, demands, and customers. Soeffker et al. (2022) investigate the information models and decision models for DS VRPs and review the prescriptive analytics methodologies for integrating the two types of models.

Our aim in this paper is to provide a comprehensive literature review on the DVRPs with random arrivals of customer requests, which are the most common source of dynamism in DVRPs (Pillac et al., 2013; Ritzinger et al., 2016; Rios et al., 2021). Psaraftis et al. (2016) report that dynamic requests are involved in around 80% of DVRPs. We propose a novel taxonomy to classify the reviewed research into several groups, and investigate each group of research from the perspectives of applications, decision strategies, problem instances, solution approaches, etc. The DVRPs with other sources of dynamism (e.g., demands and travel times) are not studied in depth but are also discussed to provide a broad overview for the DVRP research. Our main contribution can be summarized as follows:

- We systematically review the research on DVRPs, with a special emphasis given to the problems with the most common and representative source of dynamism: the random (dynamic) requests. We collect the papers studying DD and DS VRPs published in operations research journals from 1980 to 2022, and compare these papers in more than 20 aspects. Compared to the previous reviews, this review has a broader scope regarding time span, problem categories, and perspectives.
- We propose a novel taxonomy to classify the DVRP based on three criteria (source of dynamism, request type, and planning horizon) which are critical to the design of decision strategies and solution approaches. Our DVRP taxonomy is more detailed than those which only identify different dynamic aspects (Bektaş et al., 2014; Ritzinger et al., 2016) and is more compact than the taxonomy of Psaraftis et al. (2016) since we exclude the relatively unimportant factors (e.g., objective functions) that impose no significant change on problem characteristics.
- We set a new perspective for reviewing DVRPs—decision strategies—which change the problem characteristics significantly by imposing relaxations or restrictions on the scheduling and routing decisions. We identify the strategies (e.g., rejection, diversion, and waiting) that can be applied to different DVRP variants, analyze their effects, and investigate their implementation methods.
- Based on a comprehensive and in-depth review of the existing research, we derive useful insights regarding the decision strategies, methods of exploiting stochastic information, and solution approaches for DVRPs, and suggest promising directions for further research.

The remainder of this review paper is organized as follows. Section 2 introduces the proposed DVRP taxonomy, which identifies four different DVRP variants with random requests, namely the VRP with dynamic service requests, the dynamic pickup and delivery problem, the same-day delivery problem, and the dynamic multi-period VRP. Section 3 describes our review methodology and presents a conceptual framework covering all problem attributes reviewed in this paper. Concerning these attributes, we review in detail the four DVRP variants in Sections 4 to 7. Section 8 discusses the DVRPs with other sources of dynamism and compares them with the DVRPs with random requests. Finally, Section 9 summarizes this review, presents the main insights gained from the existing literature, and discusses the potential future research opportunities.

2. DVRP taxonomy

2.1. Dynamic aspects

Motivated by a wide variety of real-world applications, many DVRP variants are introduced and studied during the past four decades. Based on the dynamic aspects, we first classify most DVRPs into three groups: the *DVRP with random requests* (DVRPRR), the *DVRP with random demands* (DVRPRD), and the *DVRP with random travel times* (DVRPRTT). In the DVRPRR, the customers waiting for services are not fully known beforehand but place requests dynamically during the planning horizon. In the DVRPRD, customers' locations are known for certain in advance. Still, the amount of product that should be delivered to or picked up from each customer is revealed only when a capacitated vehicle arrives at the customer's location. In the DVRPRTT, the travel times between locations or the travel speeds of vehicles are time-dependent due to random fluctuations in traffic conditions.

It is worth noting that, according to Table 1, the category of a VRP depends not only on the information quality (with or without stochastic information) but also on the way the problem is solved (statically or dynamically). The problems with random demands or travel times are SS VRPs when the focus is on computing optimal a priori routes with stochastic information (e.g., Gómez et al., 2016; Florio et al., 2020a,b), while the DVRPRD and DVRPRTT exclusively refer to the problems that are solved by dynamic approaches (e.g., Goodson et al., 2016; Kim et al., 2016).

Besides the three common DVRP groups mentioned above, the other DVRPs are characterized by relatively uncommon dynamic aspects, such as unexpected vehicle breakdowns (e.g., Li et al., 2009), occasional drivers (e.g, Archetti et al., 2016), changing customer locations (e.g., Wang et al., 2021), and uncertain time windows (e,g, Srour et al., 2018; Györgyi and Kis, 2019). Some DVRPs involve multiple dynamic aspects. For example, Sarasola et al. (2016) study a DVRP with random pickup requests and stochastic demands, and Ferrucci and Bock (2014) address a DPDP with random travel times. In this review, we classify a DVRP with multiple dynamic aspects into the DVRPRR as long as it includes random requests.

2.2. Request types

Random arrivals of customer requests are the most common dynamic aspects in DVRPs. Some DVRPRRs studied in the literature are fully dynamic, meaning that all requests are dynamically received during the service period and that no customer information is known for certain beforehand (e.g., Voccia et al., 2019; Ulmer, 2020a). The other DVRPRRs are partially dynamic because some customers place their requests before the service period begins. The requests that the service provider knows in the beginning are commonly referred to as *static requests* (or advance/early requests), and the other requests can be called *dynamic requests* (or immediate/late requests).

To describe the dynamism of a DVRPRR, Lund et al. (1996) define the *degree of dynamism* (DoD):

$$\delta = \frac{n_d}{n_d + n_s} \in [0, 1] \tag{1}$$

This definition indicates that the dynamism of a DVRPRR is increasing in n_d, the number of dynamic requests, and is decreasing in n_s, the number of static requests. Further, Larsen et al. (2002) propose the *effective degree of dynamism* which takes into account the release times of dynamic requests; Lin et al. (2014) modify the definition of DoD by incorporating the number of request cancellations; van Lon et al. (2016) propose to measure the dynamism of fully dynamic problems by the frequency that new information arrives. In this review, we adopt the DoD defined by Lund et al. (1996) for its simplicity and wide acceptance in the literature.

The DoD mainly influences the computational burden of online decision-making, as the more dynamic requests generally require the route plan to be updated more frequently. However, changes in the DoD do not lead to significant differences in decision strategies. In many studies, the DoD is varied in a large interval to test the same problem settings and solution approaches (e.g., Larsen et al., 2002; Hong, 2012; Ulmer, 2019). Hence, we do not classify the DVRPRR according to the DoD. Instead, we distinguish three types of dynamic requests concerning the *precedence constraints*, which pose restrictions on the sequences in which vehicles visit locations.

- A *service request* is not associated with any precedence constraints. A newly arrived service request can be committed to an en-route vehicle, and the latter does not need to visit a specific pickup location or return to the depot before serving the request. Dynamic service requests mainly arise in the many-to-one pickup services in which packages are collected from customer locations and transported to a single depot, the door-to-door services in which technicians for repair works visit customers, and the delivery services in which a single type of goods (e.g., newspapers, see Ferrucci et al., 2013) are delivered to customer locations.
- A *pickup and delivery (P&D) request* consists of a pair of locations subject to precedence constraints: the service vehicle must visit the pickup location before going to the delivery location. Dynamic P&D requests can be found in various applications such as many-to-many freight transportation, taxi scheduling, ride-sharing, and on-demand meal deliveries.
- A *delivery request* is a special P&D request: its delivery location is random, but its pickup location is at a fixed depot. To serve such a request, a vehicle must first load the ordered goods at the depot and then deliver the goods to the customer. Dynamic delivery requests are mostly related to online retailing, where the products sold by an e-commerce platform are delivered to customers on the same day (one-to-many deliveries). However, when the goods to be delivered are of the same type, vehicles can be fully loaded with this type of goods before departure and do not need to return to the depot for replenishment until all loaded goods are delivered. In this case, the requests are considered service requests rather than delivery requests.

According to the type of dynamic requests, the DVRPRR can be further classified into the *VRP with dynamic service requests* (VRPDSR), the *dynamic pickup and delivery problem* (DPDP), and the *same-day delivery problem* (SDDP). We note that some DVRPRRs involve *unpaired* pickup and delivery requests, which means that each customer can only place either a pickup or a delivery request. It is commonly assumed in these DVRPRRs that all delivery requests must be placed in advance and that all dynamic requests involve pickups only (e.g., Larsen et al., 2004; Hvattum et al., 2006; Ghiani et al., 2012; Ninikas and Minis, 2014; Zhu et al., 2016). Hence, we regard these problems as VRPDSRs instead of DPDPs.

Fig. 1 presents an example for each of the three DVRPRR variants. In the VRPDSR presented in Fig. 1(a), the pre-decision state shows that a vehicle has served requests A and B and is traveling to the location of request C when two new requests, G and H, appear. The two new requests are inserted into the planned route such that the travel distance is minimized, as shown in the post-decision state. This update may violate precedence constraints if the service requests are changed to P&D requests. In the DPDP presented in Fig. 1(b), the new request D's delivery location must be inserted between its pickup location and the depot. Fig. 1(c) presents an example of the SDDP. Since the goods to be delivered must first be picked up at the depot, the new delivery requests cannot be inserted into the ongoing route as in the VRPDSR and DPDP; instead, a new route starting and ending at the depot has to be planned to accommodate the new requests.

2.3. Planning periods

The three DVRPRR variants mentioned above are considered within a single planning period, which usually corresponds to a single day's service period (no longer than 24 h). We identify one more DVRPRR variant which is characterized by multiple correlated planning periods: the *dynamic multi-period VRP* (DMPVRP). In most DMPVRPs, the rejected requests, which can be of any type, do not disappear permanently but are postponed to the next day for reconsideration. Postponing requests saves vehicles' service capability and operation costs on the same day, but incurs losses in customer satisfaction and increases the difficulty of route planning on the next day. Hence, the decision maker should balance the workload on each day and minimize the total costs (or maximize the total revenue) over the entire planning horizon. Moreover, in some DMPVRPs, the familiarity between drivers and customers changes daily. The route planner needs to take the familiarity-dependent service times into account and/or proactively develop the familiarity to improve long-term performance.

2.4. The DVRP taxonomy

In summary, we distinguish DVRPs by the three criteria mentioned above: dynamic aspects, request types, and planning periods, each having a significant influence on decision-making strategies. Our complete DVRP taxonomy is presented in Fig. 2, where the DVRPRR and its four variants are shown in solid-line blocks, while the other DVRP variants that are not studied in depth in this review are shown in dashed-line blocks.

3. Research methodology

We systematically review the DVRPRR research output published in peer-reviewed operations research journals since the year 1980. Our research protocol is summarized in Table B.1 in Appendix B. We adopt a research method consisting of two steps. First, we search for the research papers focusing on at least one of the four DVRPRR variants from Scopus. We exclude conference papers, technical notes, and book chapters in this step. We also exclude most papers from computer science journals because these papers mainly focus on developments of metaheuristic algorithms while just taking DVRPRRs as test instances. Papers providing innovative contributions to the model formulations, decision strategies, and/or solution approaches for the DVRPRR are included. Moreover, it is important to note that the DVRPRR only refers to the *truly* dynamic problems where decision-making and route executions take place simultaneously during the same period. In some problems similar to the DVRPRR, such as the *attended home delivery* (e.g., Ulmer and Thomas, 2020; Vinsensius et al., 2020), decisions

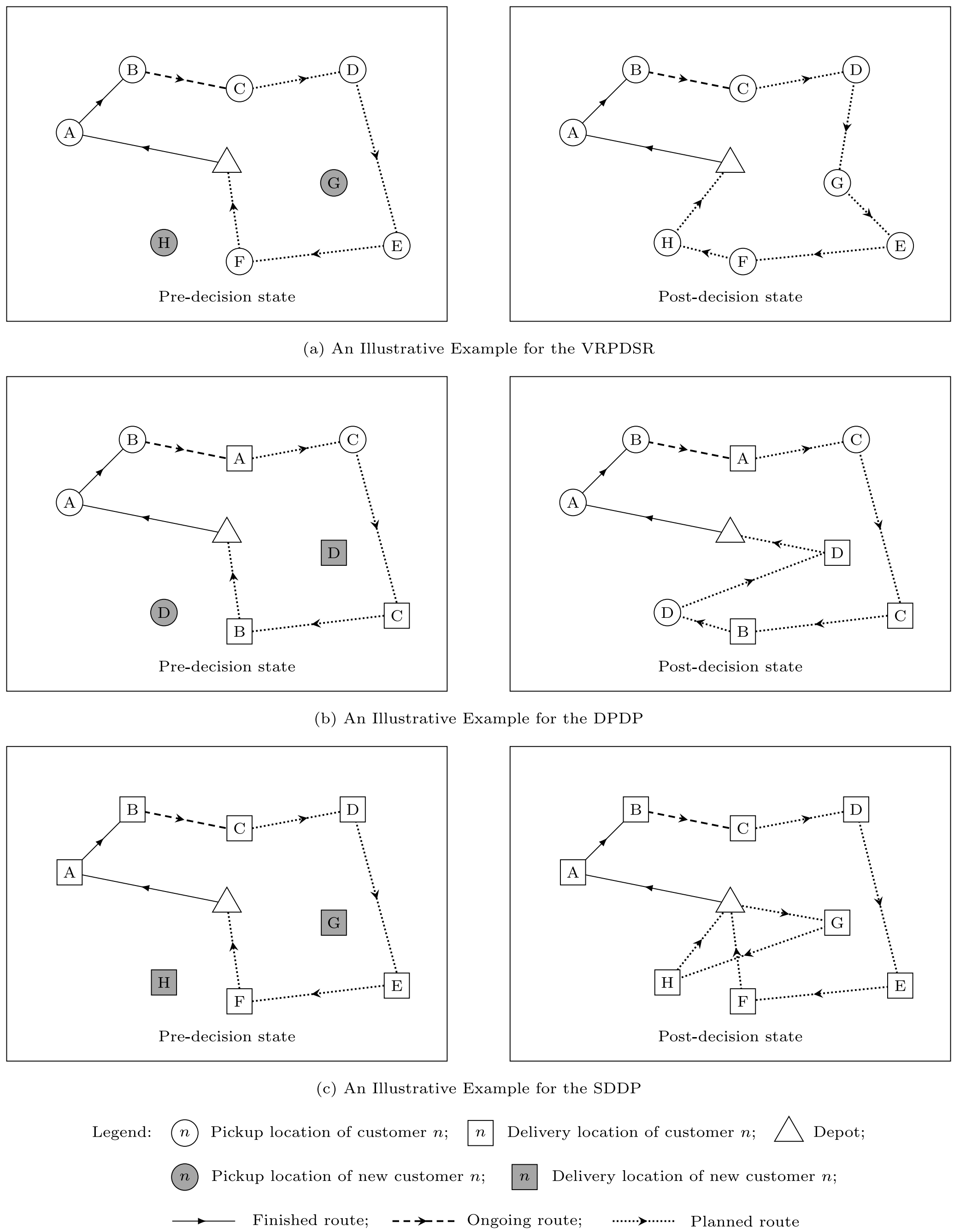

Fig. 1. Illustrative examples of the three DVRPRR variants with different request types.

are made within an order phase during which customers place their requests, while vehicles do not leave the depot until the order phase ends and a subsequent delivery phase starts. These problems are not strictly dynamic and hence are excluded from this review. In a second step, we perform a snowball search from the reference lists of the previously found papers (backward search) and the ignored papers that cite at least one previously found paper (forward search). The inclusion and exclusion criteria adopted in this step are the same as those in the first step.

The conceptual framework in Fig. 3 illustrates the three groups of problem attributes reviewed in this paper. Regarding the problem characteristics, we classify DVRPRRs by their request types and planning periods, compare their logistical contexts, side constraints, and optimization objectives, and look into how different decision strategies can be implemented. As for solution approaches, we review the mathematical models and algorithms developed to formulate and solve DVRPRRs and discuss how the stochastic information can be exploited. Moreover, we summarize the test instances solved in the reviewed papers to compare the efficiency and scalability of different solution approaches. In each of the following sections, we review a specific DVRPRR variant in detail concerning the problem attributes illustrated in Fig. 3.

4. Vehicle routing problems with dynamic service requests

The VRPDSR is the most basic version of the DVRPRR. A full classification of the existing papers on the VRPDSR is presented in a comprehensive table in our online supplementary materials. We divide this classification into three parts. The first part, Table 2, classifies the

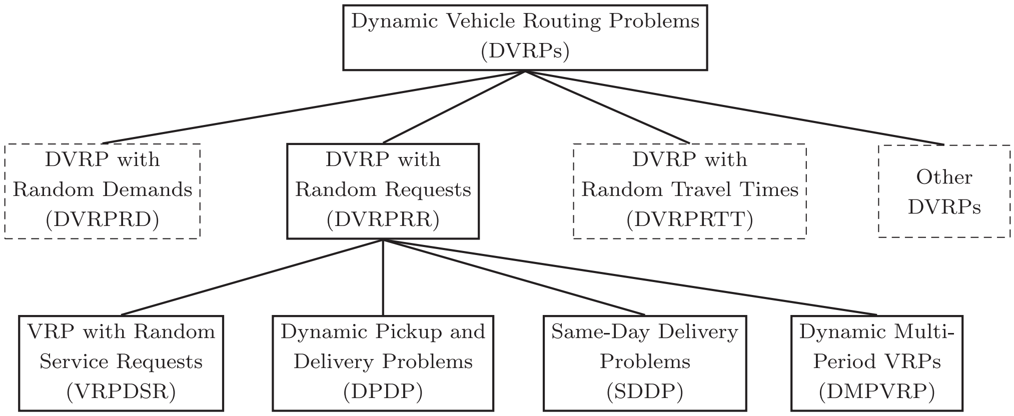

Fig. 2. DVRP taxonomy.

relevant papers concerning the main problem characteristics. In Table 2, columns "CAT" and "Context" specify the problem category (DD or DS) and the logistical context of each paper, respectively, and "N/S" means "not specified". In the "Strategies" columns, we specify for each paper whether dynamic requests can be rejected (REJ), whether each customer is given a prompt response indicating acceptance or rejection (PR), whether each vehicle can be diverted from its next customer to visit (DIV), and whether a waiting strategy is adopted (WAI). For the papers which do not allow rejections, column "PR" is marked with "—" because no responses are sent to customers. A checkmark in parentheses in column "REJ" indicates that dynamic requests can only be *implicitly* rejected by offering customers high prices that are greater than their willingness-to-pay (Zou et al., 2021). Similarly, a "(✓)" in column "DIV" means that an *implicit* diversion strategy is adopted. That is, in a road network, a vehicle can be diverted to a newly revealed customer, but it cannot change its direction before it arrives at the next intersection (where there may be no customer) determined by its current route (Thomas and White III, 2004; Zhang et al., 2022a).

The "Side constraints" columns in Table 2 indicate the fleet setting (single, multiple, or infinite vehicles) and whether service time window (STW), depot deadline (DDL), and finite vehicle capacity (CAP) are considered in each paper. A hard time window or depot deadline is a strict constraint that must be respected, while a soft time constraint may be violated and incur a penalty cost of lateness in the objective function. A fuzzy time window is extended from a hard time window by adding the best service time between the earliest and latest service times and a fuzzy membership function for evaluating the service level (Lin et al., 2014). The vehicle capacity constraints in VRPDSRs are all hard. The last column of Table 2 indicates the optimization objectives, where the multiple objectives scalarized (e.g., weighted sum) as a single cost or reward function are connected by "+" and "–". In contrast, the primary and secondary objectives that are optimized sequentially are separated by ",".

4.1. Applications

The earliest VRPDSR studied in the literature is also called *dynamic traveling repairman problem* (DTRP). Bertsimas and van Ryzin (1993) introduce a typical application of the DTRP: a utility firm is in charge of the maintenance of a geographically dispersed network of electricity, gas, or water facilities and operates a fleet of repair vehicles based at a single depot to respond to the failures which are random in both time and location. The main optimization objective of the DTRP is to minimize the response time and/or travel costs.

During the last 20 years, most VRPDSRs arise in the context of on-demand door-to-door pickup services. A typical application of the VRPDSR is the courier pickup service in which a fleet of vehicles or a single vehicle is dispatched from a depot to collect parcels from customer locations (e.g., Sarasola et al., 2016; Ulmer et al., 2018a). Some VRPDSR applications involve unpaired pickup and delivery requests, such as the one-to-many-to-one problems studied by Ninikas and Minis (2014) and Zhu et al. (2016), but only pickup requests are accepted during the online phase. Zou et al. (2021) study a VRPDSR based on a shared transportation system in which each vehicle is considered an individual service participant. Each time a customer places a dynamic request for collecting shipments, the system manager needs to properly allocate the service cost among vehicles and let the customer select a suitable vehicle to fulfill the request.

Only a few VRPDSRs are based on pure delivery services. The VRPDSR studied by Ferrucci and Bock (2015, 2016) is motivated by the urgent newspaper delivery service, where some subscribers have not received their newspapers due to unexpected delivery failures or errors and request subsequent deliveries of newspapers. Schyns (2015) addresses an airport refueling problem in which a fleet of trucks is dispatched to refuel the aircraft, which arrive dynamically at different parking spaces. In the VRPDSRs with delivery requests only, all customers request the same type of goods so that an en-route vehicle can be sent to serve a newly received request without returning to the depot (as long as it still has enough goods on board). This feature allows the service provider to handle the delivery requests in the same way as handling pickup requests.

4.2. Decision strategies

While the basic decisions that need to be made in the VRPDSR are the construction and updates of vehicle routes, the service performance can be improved by additional decision strategies such as rejecting requests, diverting en-route vehicles, and ordering vehicles to wait. In this section, we discuss these strategies adopted in the existing research.

Rejection

In many VRPDSRs, service providers can reject the dynamic requests they are unable or unwilling to serve. Under this setting, a new request should be accepted before being inserted into vehicle routes. The decisions on acceptance or rejection are usually irreversible. Once a request is accepted, it becomes a mandatory request that must be served within the current service period, and once a request is rejected, it will not be considered anymore. Around 50% of the existing papers on VRPDSRs allow rejections of dynamic requests. The logistical contexts of these papers are mainly courier and general pickup services. In the VRPDSRs involving essential or emergency services, such as repair, delivery of urgent goods, and aircraft refueling (within an airport), request rejections are not permitted.

Generally, there are two situations in which a new request is rejected. In the first situation, the service provider cannot find a feasible

Table 2
Classification of the literature on the VRPDSR — Part 1: Problem characteristics.

Literature	CAT	Context	Strategies				Side constraints				Objective(s)
			REJ	PR	DIV	WAI	VEH	STW	DDL	CAP	
Bertsimas and van Ryzin (1991)	DD	Repair	×	–	×	×	Single	×	×	×	min W
Bertsimas and van Ryzin (1993)	DD	Repair	×	–	×	×	Multi	×	×	Hard	min(C+W)
Gendreau et al. (1999)	DD	Courier pickup	✓	✓	×	×	Multi	Soft	Hard	×	min(C+L)
Ichoua et al. (2000)	DD	Courier pickup	✓	✓	✓	×	Multi	Soft	Hard	×	min(C+L)
Larsen et al. (2002)	DD	Repair	×	–	×	×	Single	×	×	×	min C
Bent and Van Hentenryck (2004)	DS	Multiple	✓	✓	×	×	Multi	Hard	Hard	Hard	max S
Larsen et al. (2004)	DS	Courier pickup	×	–	×	✓	Single	Soft	×	×	min(C+L)
Thomas and White III (2004)	DS	Pickup	✓	×	(✓)	✓	Single	×	×	×	min(C−S)
Branke et al. (2005)	DS	N/S	✓	×	✓	✓	Multi	×	Hard	×	max P
Chen and Xu (2006)	DD	Pickup	×	–	✓	×	∞	Hard	Hard	Hard	min C
Hvattum et al. (2006)	DS	Pickup	✓	×	×	✓	Multi	Hard	Hard	Hard	min(V+C−U)
Ichoua et al. (2006)	DS	Courier pickup	✓	✓	×	✓	Multi	Soft	Hard	×	max S, min(C+L)
Hvattum et al. (2007)	DS	Pickup	✓	×	×	×	Multi	Hard	×	Hard	min U, V, C
Thomas (2007)	DS	Pickup	✓	×	×	✓	Single	×	Hard	×	max S
Branchini et al. (2009)	DS	Pickup	✓	✓	✓	✓	Multi	Soft	Hard	Hard	max(S−L−C−V)
Garrido and Riff (2010)	DD	N/S	×	–	N/S	×	Multi	×	×	Hard	min C
Ghiani et al. (2012)	DS	Pickup	✓	×	×	✓	Single	×	Hard	×	max S
Hong (2012)	DD	N/S	✓	✓	✓	×	Multi	Hard	Hard	Hard	min(C+V)
Ferrucci et al. (2013)	DS	Urgent delivery	×	–	✓	✓	Multi	Soft	×	×	min(W+L)
Lin et al. (2014)	DD	Courier pickup	×	–	×	×	Multi	Fuzzy	Hard	Hard	min C
Ninikas and Minis (2014)	DD	Pickup	×	–	×	×	∞	Hard	Hard	Hard	min C
Euchi et al. (2015)	DD	N/S	×	–	N/S	×	Multi	×	×	Hard	min C
Ferrucci and Bock (2015)	DS	Urgent delivery	×	–	✓	×	Multi	Soft	×	×	min(W+L+D)
Schyns (2015)	DD	Plane refueling	×	–	×	×	Multi	Hard	×	Hard	min W
Ferrucci and Bock (2016)	DS	Urgent delivery	×	–	✓	✓	Multi	Soft	×	×	min(W+L)
Sarasola et al. (2016)	DD	Courier pickup	×	–	N/S	×	∞	×	Soft	Hard	min(C+L)
Zhu et al. (2016)	DD	N/S	✓	×	×	×	Single	×	×	Hard	min(C+W−S)
Ulmer et al. (2018a)	DS	Courier pickup	✓	×	×	✓	Single	×	Hard	×	max S
Xu et al. (2018)	DD	N/S	×	–	N/S	×	Multi	×	×	Hard	min C
Ulmer et al. (2019a)	DS	Courier pickup	✓	×	×	×	Single	×	Hard	×	max S
Ulmer (2019)	DS	N/S	✓	×	×	×	Single	×	Hard	×	max S
Bono et al. (2021)	DS	Delivery	×	–	×	×	Multi	Soft	×	Hard	min(C+L)
Xiang et al. (2021)	DD	Multiple	×	–	N/S	×	Multi	×	×	Hard	min C
Zou et al. (2021)	DD	Pickup	(✓)	✓	N/S	×	Multi	Hard	×	Hard	min C
Zhang et al. (2022a)	DS	Pickup	✓	✓	(✓)	×	Multi	×	×	×	max S

Acronyms				Objectives			
CAT	Category	REJ	Rejection	W	Waiting time of customers	C	Travel cost/time/distance
PR	Prompt reply	DIV	Diversion	L	Penalty of lateness	S	Number/revenue of served requests
WAI	Waiting	VEH	Number of vehicles	P	Probability that a new request can be served		
STW	Service time window			V	Number/cost of vehicles used	U	Number/penalty of unserved requests
DDL	Depot deadline	CAP	Vehicle capacity	D	Penalty of en-route diversions		

vehicle route to accommodate the new request. This situation can occur in almost all VRPDSRs. In the second situation, which mainly occurs in the DS VRPDSRs, a new request that can be feasibly served is rejected if it consumes too much time or vehicle resources and reduces the expected rewards in the future. Regardless of the problem category (DD or DS), when rejections are allowed, maximizing the number or revenue of the accepted requests (or minimizing the penalty for rejections) is commonly incorporated into the objective function, and vehicles' service capability is usually restricted by a hard time window and/or capacity constraints. By contrast, in the VRPDSRs where rejections are forbidden, the authors primarily focus on minimizing the total travel costs and/or service lateness, and the time and vehicle constraints should be relaxed to guarantee feasibility. For instance, Ninikas and Minis (2014) and Sarasola et al. (2016) assume that the fleet size is sufficiently large to serve all requests, while Ferrucci and Bock (2015) define the depot deadline and service time windows as soft constraints.

In the VRPDSRs with rejections, customers expect prompt responses from service providers to know whether they will be served in the current service period. Making such prompt decisions is challenging because it requires highly efficient solution approaches. In most relevant papers, the decisions on acceptance or rejection are not immediately made upon request arrivals; instead, decision epochs are triggered at predefined time points or when a vehicle arrives at a customer location. In Table 2, only eight papers emphasize customer responsiveness: most of them run reoptimization procedures continuously, while Hong (2012), Zou et al. (2021), and Zhang et al. (2022a) make immediate decisions each time a new request arrives. Unlike most of the relevant papers in which service providers make decisions explicitly, Zou et al. (2021) propose several cost-sharing mechanisms which take customers' decisions into account. After placing a new request, the customer receives from the service provider an immediate response indicating its shared cost. The customer then accepts the service if the shared cost is lower than its level of willingness to pay; otherwise, it declines the service. In these mechanisms, the service provider implicitly rejects the infeasible or unprofitable requests by sending high shared costs to customers.

Diversions

In most VRPDSRs, a vehicle that has left its previous location cannot be diverted before it finishes servicing the next customer in its planned

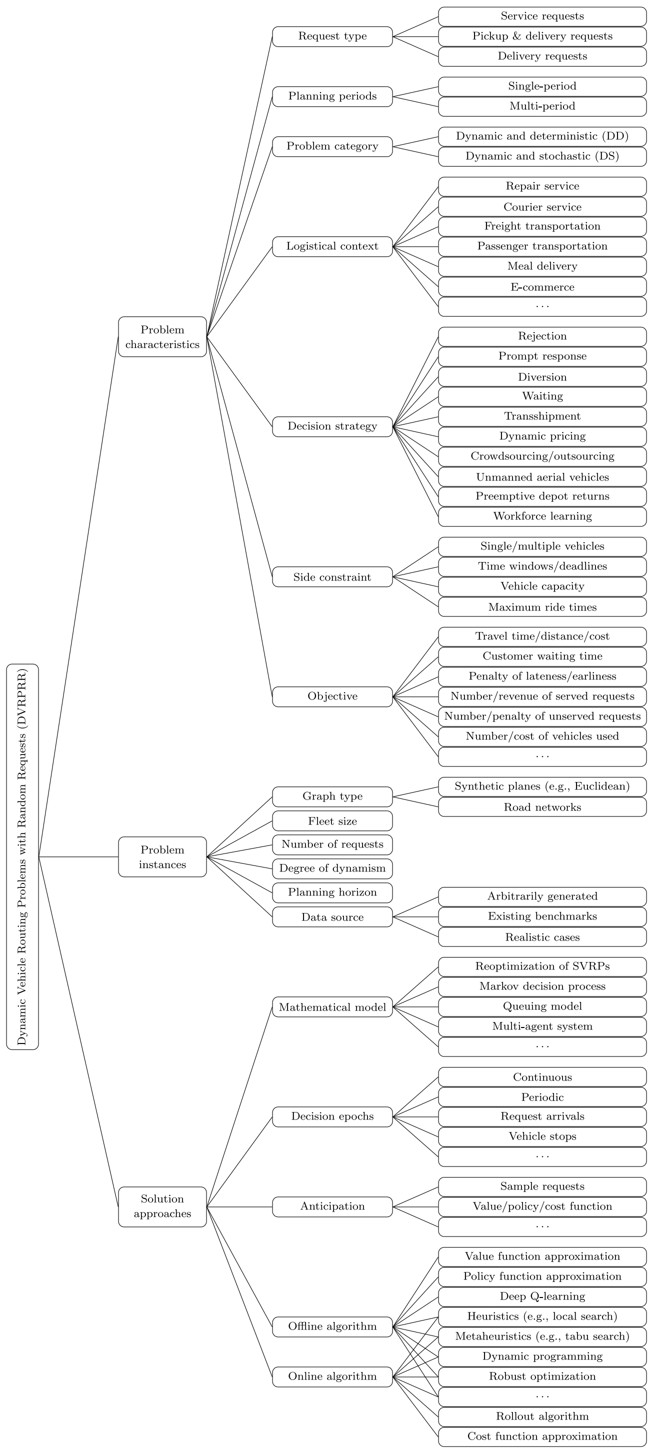

Fig. 3. Conceptual framework of the DVRPRR.

route. For instance, Gendreau et al. (1999), Hvattum et al. (2006), and Lin et al. (2014) set each vehicle's next planned location as a fixed starting node when updating route plans. In Thomas (2007) and Ulmer et al. (2018a, 2019a), decisions are made only when the single vehicle arrives at a customer location, and hence diverting the vehicle is intrinsically prohibited. Diversion strategies, on the contrary, allow new requests to be inserted between vehicles' current locations and next destinations and are adopted in 23% of VRPDSRs. Diverting a vehicle is likely to be beneficial especially when a new request appears in the vicinity of the vehicle's current location (an example is provided in Fig. 4), and such a decision should be made quickly because the vehicle moves fast and the diversion opportunity may be lost soon (Gendreau and Potvin, 1998).

Diversion strategies do not necessarily require stochastic information and hence can be applied to both DD and DS VRPDSRs. The first diversion strategy for the VRPDSR is proposed by Ichoua et al. (2000). When reoptimizing the route plan, Ichoua et al. (2000) set vehicles' current locations, rather than their next destinations, as fixed origins. Experimental results show that allowing diversions reduces the total travel distance, the lateness of services, and the number of rejected requests. In another VRPDSR, Branchini et al. (2009) find that adopting a diversion strategy leads to slight reductions in travel distance and request rejections. For most of the other VRPDSRs with diversion strategies, no comparative experiments are conducted to evaluate the benefits of diversions.

Despite the potential benefits, diversions may interrupt the ongoing vehicle routes too often and cause increased workloads and distractions for drivers. Ferrucci and Bock (2015) is the only work that addresses the trade-off between diversions' benefits and negative effects. They assume that each diversion incurs a fixed penalty cost to abandon the diversion opportunities with insignificant reductions in service delays. Their sensitivity analysis reveals that, compared to zero penalties, a small value of penalty cost leads to substantially fewer diversions and shorter waiting times for customers, and that a large penalty value can further reduce the number of diversions but significantly increases customer inconvenience.

Waiting

A common characteristic of almost all VRPDSRs is that vehicles may be required to wait at some locations under some circumstances. We consider a waiting policy to be *reactive* as long as it follows a deterministic decision rule and does not rely on any stochastic information, no matter if it is applied to a DD or DS problem. For example, in Ninikas and Minis (2014), if a vehicle arrives at a customer location too early, it must wait there until the beginning of the customer's time window; Ferrucci and Bock (2015) require each vehicle with no pending requests to idle at the location of the last customer that it has served. By contrast, a waiting policy is *proactive* if it prescribes waiting decisions based on the stochastic information of dynamic requests, and such a policy is called a waiting strategy in this review. A waiting strategy may require a vehicle to wait proactively, even if it has not finished all committed services and the next customer's time window is already open, to prevent the vehicle from leaving the regions where many new requests are likely to unfold in the future.

Nearly 1/3 of the VRPDSR papers develop waiting strategies within DS formulations. Most of these waiting strategies are based on heuristic decision rules because making optimal waiting decisions is computationally prohibitive for reasonably large problems. Thomas and White III (2004) point out that a VRPDSR with a single vehicle and waiting decisions can be reduced to a TSP and hence is NP-hard. They employ the standard dynamic programming method to solve the problem, and the results show that the computation time increases drastically as the number of dynamic requests increases from 1 to 7. For a VRPDSR with exactly one dynamic request and a given a priori tour, Branke et al. (2005) prove that the problem of optimally allocating the available waiting time (depot deadline minus the travel time of the a priori tour) to customer locations is NP-complete, even if the arrival time of the single dynamic request is known. They derive optimal waiting strategies only for the simple cases with at most two vehicles and turn to heuristic strategies for the general cases with more vehicles.

To guarantee practicability, a typical heuristic waiting strategy makes a decision only when a vehicle is initially at the depot or visits a customer, and its decision space is usually restricted to only two options: waiting at the current location and moving to the next destination. A vehicle that is moving from one location to another

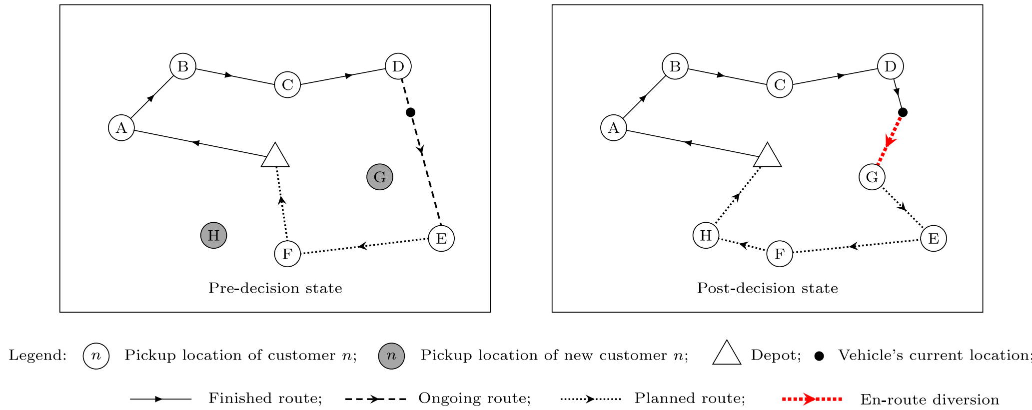

Fig. 4. An illustrative example of En-route diversion in a VRPDSR.

is never interrupted by an order of waiting. The general idea of a heuristic waiting strategy is to let a vehicle wait as long as it is feasible (i.e., no violation of the depot deadline or the time windows of pending requests) and the probability of receiving new requests in the neighborhood is sufficiently high. Moreover, for a VRPDSR with multiple vehicles, a good waiting strategy should pursue a balanced coverage of the entire service area and avoid clustering vehicles in a small zone. This can be achieved by setting an upper bound for the number of vehicles in each subregion (Ichoua et al., 2006) or proactively directing vehicles to the subregion with the least number of vehicles (Branchini et al., 2009).

Some intuitively appealing waiting strategies are designed based on the analytical results of simple cases (e.g., precisely one dynamic request and single or two vehicles), such as the *variable* waiting strategy proposed by Branke et al. (2005) and the *center-of-gravity longest waiting* strategy of Thomas (2007) and Ghiani et al. (2012). Another way to derive heuristic waiting strategies is to incorporate waiting decisions into a mathematical model implicitly or explicitly and solve the model with heuristic algorithms. For instance, Ferrucci et al. (2013), Ferrucci and Bock (2016) divide the service area into clusters based on historical request data, and use a dummy request with a specific starting time to represent each cluster. During the execution of the route plan, if a vehicle's next destination is a dummy request, the vehicle is required to wait at its current location as long as it can arrive at the dummy request no later than the starting time. In the Markov decision process (MDP) formulated by Ulmer et al. (2018a), the decision space includes an explicit waiting decision that orders the vehicle to stay at its current location for a certain period of time. In a later work, however, Ulmer et al. (2019a) disallow waiting because it significantly increases the computational burden and does not improve the solution quality substantially.

In addition to the decisions on when and how long a vehicle should wait at its current location, some waiting strategies may guide the waiting vehicles to strategic waiting points, which are determined in advance based on the spatial distribution of customers. When developing such a waiting strategy, it is important to restrict the repositioning decisions to an appropriate level, so as to trade off the potential profit margin and the increase in travel costs. In the waiting strategy proposed by Larsen et al. (2004), if a vehicle will arrive at the next customer's location earlier than the opening of a time window, it either waits at its current location or moves to one of several predefined idle points. Specifically, the authors suggest that the waiting vehicle should be directed to the nearest idle point, as long as the probability of receiving a new request there is higher than a small threshold. Experimental results show that this repositioning policy reduces service lateness and only leads to a slight increase in the travel distance. In Branchini et al. (2009), when a vehicle has no request left to serve and its next destination is the depot, it is sent to a strategic waiting place if it has sufficient spare time to return to the depot at least 30 min before the deadline. The benefit of this repositioning policy is validated through an extensive computational study.

4.3. Problem instances

There is no well-accepted benchmark dedicated to the VRPDSR. Thirty percent of VRPDSR instances originate directly from realistic cases, such as the airport refueling operations studied by Schyns (2015) and the urgent newspaper delivery services considered in Ferrucci and Bock (2016). Hong (2012) adopt the DVRP data set designed by Lackner (2004). The other instances are mostly derived from well-known SVRP benchmarks, such as Solomon (1987), Beasley (1990), and Dumas et al. (1995). To convert an SVRP benchmark to a VRPDSR instance, the customer set should be divided into a subset of static requests and a subset of dynamic requests, and assumptions need to be made for the fleet size, planning horizon, the probability distribution of arrival times, etc. (see e.g., Ninikas and Minis, 2014; Zou et al., 2021). The interested reader is also referred to Kilby et al. (1998) and Montemanni et al. (2005) which present detailed methods of creating dynamic instances (DoD = 0.50) from SVRP benchmarks.

Table 3 summarizes the VRPDSR instances solved in the literature. The "Graph" column indicates the graph type: Euclidean plane, road network, grid network, or airport. For each non-Euclidean graph, the number of nodes is presented in parentheses. The "Avail." column indicates the availability of VRPDSR instances, where "(✓)" means that the links to the data sets were provided but are no longer accessible. The "Vehicle(s)" column shows the number of vehicles considered in each paper, where ">1" means that the fleet size is greater than one and finite but its exact value is not specified. For example, Gendreau et al. (1999) and Ichoua et al. (2000) set the fleet size to the minimum number of vehicles in the best-known solution for Solomon's benchmark, while Bent and Van Hentenryck (2004) determine the fleet size by solving Solomon's benchmark again (by local search) and adding two more vehicles. The "Request(s)" column indicates the total number of requests, including both static and dynamic ones. The "Horizon" column presents the time length of the planning horizon (during which dynamic requests arrive and vehicle routes are updated). The main data sources are given in the last column "Source(s)", where "Arbitrary" means that the instances are newly generated in an arbitrary manner.

4.4. Solution approaches

Table 4 summarizes the solution approaches for VRPDSRs. We refer the interested reader to our online supplementary materials for a complete classification of the VRPDSR literature (a combination of Tables 2, 3, and 4). In Table 4, columns "Formulation" indicate the mathematical model (SVRP reoptimization, MDP, queuing model, etc.),

Table 3
Classification of the literature on the VRPDSR — Part 2: Problem instances.

Literature	Avail.	Graph	Vehicle(s)	Request(s)	DoD (%)	Horizon (h)	Source(s)
Bertsimas and van Ryzin (1991)	×	Euclidean	1	N/S	100	∞	Arbitrary
Bertsimas and van Ryzin (1993)	×	Euclidean	>1	N/S	100	∞	Arbitrary
Gendreau et al. (1999)	×	Euclidean	>1	100	50	{0.25, 1}	Solomon
Ichoua et al. (2000)	×	Euclidean	>1	100	{50, 75}	0.25	Solomon
Larsen et al. (2002)	×	Euclidean	1	{20, 30, 40}	{0, 5, …, 100}	8	Arbitrary
Bent and Van Hentenryck (2004)	×	Euclidean	>1	100	{50, 80}	0.5	Solomon
Larsen et al. (2004)	×	Euclidean	1	{54, …, 79}	{11, …, 23}	8	Realistic
Thomas and White III (2004)	×	Road (131)	1	{1, 3, 5, 7}	100	4	Realistic
Branke et al. (2005)	×	Euclidean	>1	{51, …, 200}	{0.5, …, 1.96}	N/S	Beasley
Chen and Xu (2006)	×	Euclidean	∞	100	{25, …, 75}	{1/3, 2/3}	Solomon
Hvattum et al. (2006)	×	Euclidean	>1	≈130	≈50	10	Realistic
Ichoua et al. (2006)	×	Euclidean	{4, 6}	{180, 240}	75	5	Arbitrary
Hvattum et al. (2007)	×	Euclidean	>1	133	≈50	10	Arbitrary
Thomas (2007)	×	Euclidean	1	{40, 50}	{6, 7.5, 25, 50}	N/S	Solomon, Dumas
Branchini et al. (2009)	×	Grid (22500)	70	{1060, 1080, 1101}	{66, 80}	9	Realistic
Garrido and Riff (2010)	✓	Euclidean	50	{50, …, 199}	50	{5/24, 5/12}	Kilby, Montemanni
Ghiani et al. (2012)	✓	Euclidean	1	{40, 50}	{25, 50}	N/S	Solomon, Dumas
Hong (2012)	(✓)	Euclidean	>1	100	{10, 30, …, 90}	N/S	Lackner
Ferrucci et al. (2013)	×	Road (N/S)	{8, 10, 12}	150	>90	4	Realistic
Lin et al. (2014)	×	Euclidean	>1	{25, 50, 100}[a]	20	N/S	Solomon
Ninikas and Minis (2014)	(✓)	Euclidean	∞	100	{25, 50, 75}	N/S	Solomon
Euchi et al. (2015)	✓	Euclidean	50	{50,…,199}	50	5/12	Kilby, Montemanni
Ferrucci and Bock (2015)	×	Road (N/S)	{8, 10, 12}	150	>90	4	Realistic
Schyns (2015)	×	Airport (N/S)	5	30	100	3	Realistic
Ferrucci and Bock (2016)	×	Road (N/S)	{8, 10, 12}	150	>90	4	Realistic
Sarasola et al. (2016)	✓	Euclidean	∞	{50, …, 199}	50	N/S	Kilby
Zhu et al. (2016)	×	Euclidean	1	{30, 50}	50	N/S	Realistic
Ulmer et al. (2018a)	×	Euclidean	1	100	{50, 75}	6	Arbitrary
Xu et al. (2018)	(✓)	Euclidean	>1	{50,…,480}	{33.3, 50, 66.7}	5/12	Montenammi
Ulmer et al. (2019a)	×	Euclidean	1	100	{25, 50, 75}	6	Arbitrary
Ulmer (2019)	×	Euclidean	1	60	{25, 50, 75, 100}	6	Arbitrary
Bono et al. (2021)	✓	Euclidean	{2, 4, 10}	{10, 20, 50}	{10, 25, 50, 75}	8	Arbitrary
Xiang et al. (2021)	✓	N/S	>1	{50, …, 1000}	50	N/S	Kilby, realistic
Zou et al. (2021)	×	Euclidean	>1	≈82	≈69.5	N/S	Solomon
Zhang et al. (2022a)	✓	Road (16 080)	{2,…,20}	{160,…,947}	{75, 85, 90, 95}	10	Realistic

[a]The authors assume that 16% of customers cancel their requests during the service period.

decision epochs (time points at which an SVRP is reoptimized or a decision is made), and anticipation method adopted in each work. The algorithms that are employed to solve the mathematical models are summarized in columns "Algorithms", where "Offline" refers to the algorithms implemented at the beginning of the service period to compute decision policies (e.g., Thomas and White III, 2004; Ulmer et al., 2018a) or to optimize the initial service routes for static requests (e.g., Ghiani et al., 2012; Lin et al., 2014), while "Online" refers to the algorithms implemented during the service period to deal with the dynamic requests and update route plans.

The formulations for most VRPDSRs can be classified into two types: SVRP reoptimization and MDP. The former decomposes the VRPDSR into a sequence of SVRPs, each corresponding to a decision epoch, and then solves the SVRPs repeatedly by heuristic or metaheuristic approaches. As shown in Tables 2 and 4, nearly 2/3 of VRPDSRs in the literature are solved by reoptimizing their SVRP counterparts, either continuously, periodically, or at event-triggered decision epochs. Around 40% of SVRP reoptimization frameworks incorporate stochastic information of dynamic requests. One way to exploit stochastic information is randomly generating sample requests (or dummy requests) so as to guide vehicles to request-likely regions (e.g., Ferrucci and Bock, 2016). The other way is to design proactive waiting strategies based on the spatio-temporal distribution of dynamic requests (e.g., Branchini et al., 2009). Over the past 5 years, there is an increasing trend in the amount of work modeling VRPDSRs as MDPs. As a classical model for sequential decision-making, MDP is intrinsically stochastic as it aims at maximizing the expected total reward over the entire planning horizon. Solving an MDP usually requires approximating a value function that captures the expected reward-to-go (e.g., Ulmer et al., 2018a), while Zhang et al. (2022a) propose a new solution method that anticipates the future reward by knapsack-based models. In the following, we discuss the VRPDSR formulations and their solution methods in detail.

Reoptimization of SVRPs

The first continuous reoptimization framework is proposed by Gendreau et al. (1999) for a DD VRPDSR. The authors maintain in an adaptive memory a set of feasible solutions which are initially generated by a stochastic insertion heuristic. Throughout the planning horizon, they employ a parallel tabu search metaheuristic to continuously improve these solutions. When a new request arrives, the adaptive memory is updated by keeping the routes where the new request can be feasibly inserted and discarding the others; if the new request cannot be inserted into any route stored in the adaptive memory, it is rejected. The same framework is adopted by Ichoua et al. (2000) for a similar DD VRPDSR with consideration of en-route diversions.

In the first study on DS VRPDSR, Bent and Van Hentenryck (2004) generalize the continuous reoptimization framework of Gendreau et al. (1999) to a *multiple plan approach* (MPA). The MPA maintains a distinguished route plan and a solution pool, in which new plans that are compatible with the distinguished plan are continuously generated by a local search heuristic, and the incompatible ones are continuously discarded. The authors show that it is beneficial to use a consensus function to select the distinguished plan that shares the most similarities with the other routes in the solution pool. More importantly, Bent and Van Hentenryck (2004) extend the MPA to a *multiple*

Table 4
Classification of the literature on the VRPDSR — Part 3: Solution approaches.

Literature	Formulation			Algorithms	
	Framework/Model	Decision epochs	Anticipation	Offline	Online
Bertsimas and van Ryzin (1991)	Queuing	Request arrivals	–	Analytical study	
Bertsimas and van Ryzin (1993)	Queuing	Request arrivals	–	Analytical study	
Gendreau et al. (1999)	Reopt-SVRPs	Continuous	–	Tabu search	
Ichoua et al. (2000)	Reopt-SVRPs	Continuous	–	Tabu search+diversion strategy	
Larsen et al. (2002)	Dynamic systems	Request arrivals	–	Simulation study	
Bent and Van Hentenryck (2004)	Reopt-SVRPs	Continuous	Samples	Multiple scenario approach+local search	
Larsen et al. (2004)	Reopt-SVRPs	Arrivals & stops	Waiting	3-opt procedure+waiting heuristics	
Thomas and White III (2004)	MDP	Vehicle stops	Value function	Dynamic programming	–
Branke et al. (2005)	WDP	–	Waiting	Exact	CH+waiting heuristics
Chen and Xu (2006)	Reopt-SVRPs	Periodic	–	CGBH	
Hvattum et al. (2006)	Reopt-SVRPs	Periodic	Samples	Dynamic stochastic hedging heuristic	
Ichoua et al. (2006)	Reopt-SVRPs	Arrivals & stops	Samples	Tabu search+waiting heuristics	
Hvattum et al. (2007)	Reopt-SVRPs	Periodic	Samples	Branch-and-regret heuristic	
Thomas (2007)	MDP	Vehicle stops	Value function	GRASP	Waiting heuristics
Branchini et al. (2009)	Reopt-SVRPs	Continuous	Waiting	Constructive heuristic	CH+AGLS+waiting
Garrido and Riff (2010)	Reopt-SVRPs	Periodic	–	Evolutionary hyper-heuristic	
Ghiani et al. (2012)	MDP	Vehicle stops	Value function	GRASP	Waiting heuristics
Hong (2012)	Reopt-SVRPs	Request arrivals	–	Large neighborhood search	
Ferrucci et al. (2013)	Reopt-SVRPs	Periodic	Samples	Tabu search	
Lin et al. (2014)	Reopt-SVRPs	Request arrivals	–	HNS+VNS	Heuristics
Ninikas and Minis (2014)	Reopt-SVRPs	SRR/NRR	–	B&P/CGBH	
Euchi et al. (2015)	Reopt-SVRPs	Periodic	–	Ant colony algorithm	
Ferrucci and Bock (2015)	Reopt-SVRPs	Periodic	Samples	Tabu search	
Schyns (2015)	Reopt-SVRPs	Request arrivals	–	–	Ant colony algorithm
Ferrucci and Bock (2016)	Reopt-SVRPs	Periodic	Samples	Tabu search	
Sarasola et al. (2016)	Reopt-SVRPs	Periodic	–	Variable neighborhood search	
Zhu et al. (2016)	Reopt-SVRPs	Periodic	–	Multiobjective memetic algorithm	
Ulmer et al. (2018a)	MDP	Vehicle stops	Value function	nVFA+CH	CH
Xu et al. (2018)	Reopt-SVRPs	Periodic	–	Ant colony algorithm	
Ulmer et al. (2019a)	MDP	Vehicle stops	Value function	nVFA+CH	Rollout+CH
Ulmer (2019)	MDP	Vehicle stops	Value function	nVFA+CH	Rollout+CH
Bono et al. (2021)	Multi-agent MDP	Vehicle stops	Policy function	Reinforcement learning	
Xiang et al. (2021)	Reopt-SVRPs	Periodic	–	Ant colony algorithm	
Zou et al. (2021)	Reopt-SVRPs	Arrivals & periodic	–	Cost sharing heuristics	
Zhang et al. (2022a)	MDP	Request arrivals	Knapsack model	Potential-based lookahead policies	

Acronyms

Reopt-SVRPs	Reoptimization of static VRPs	MDP	Markov decision process	WDP	Waiting drivers problem
SRR/NRR	Single-/N-request reoptimization	CH	Constructive heuristic	CGBH	Column-generation-based heuristic
GRASP	Greedy randomized adaptive search procedure			AGLS	Adaptive granular local search
HNS	Hybrid neighborhood search	VNS	Variable neighborhood search	B&P	Branch-and-price
nVFA	Non-parametric value function approximation				

scenario approach (MSA) by considering the possible future requests (i.e., dummy requests) which are randomly generated from known probability distributions. Their computational study shows that MSA produces significantly better route plans than MPA, thus validating the merits of exploiting stochastic information.

Another continuous reoptimization framework is proposed by Branchini et al. (2009). To compute the initial routes for serving static requests, the authors develop a constructive heuristic whose objective is to scatter vehicles in the service region. During the service period, they employ another fast constructive heuristic (cheapest insertion) to quickly determine whether to accept each newly arrived request and run an adaptive granular local search procedure to improve the ongoing vehicle routes continuously. Their online solution approaches do not explicitly anticipate the future, but the local search procedure is integrated with a waiting strategy (see Section 4.2), in which the strategic waiting places of idle vehicles are determined according to the expected spatial distribution of dynamic requests.

Compared to continuous reoptimization, periodic reoptimization is a simpler and more commonly used framework characterized by equidistant decision epochs. At each epoch, an SVRP including all revealed requests can be formulated by taking a "snapshot" of the VRPDSR. Newly received requests are not scheduled immediately but are buffered for consideration at the next decision epoch. Thus, the decision maker follows a rolling-horizon procedure and solves a sequence of SVRPs. Most periodic reoptimization frameworks have tight time frames. Chen and Xu (2006) solve an SVRP every 60 or 120 s and each time a route plan must be computed within 5 or 10 s. They propose a column-generation-based heuristic (CGBH) that generates columns dynamically over time and solves a restricted set-partitioning model at each decision epoch. In Ferrucci and Bock (2015), a tabu search metaheuristic is executed every 20 s and has only 10 s of time to solve each SVRP.

If stochastic information is used, the periodically solved SVRPs can be added with dummy requests. The dynamic stochastic hedging heuristic (DSHH) and the branch-and-regret heuristic (BRH) proposed by Hvattum et al. (2006, 2007) are two examples. They are similar to the MSA (Bent and Van Hentenryck, 2004) in that they all solve multiple sample scenarios at each decision epoch and produce multiple route plans, but MSA determines the distinguished plan by a consensus function, while DSHH and BRH formulate the final plan through iterative procedures. Moreover, the local search algorithm in MSA improves the route plans continuously, while DSHH and BRH are executed every hour for a short period of time. A comparison of these anticipatory algorithms is provided in Hvattum et al. (2007).

Reoptimizations of sequential SVRPs can also be triggered by events. For example, in an aircraft refueling truck routing problem, Schyns (2015) reacts to each newly received task by resolving an SVRP immediately with an ant colony algorithm. The arrival-triggered mechanism allows them to improve service responsiveness, which is an important objective under strict flight schedules. Furthermore, Ninikas and Minis (2014) propose an N-request reoptimization (NRR) framework in which the sequential SVRPs are reoptimized when the number of pending dynamic requests reaches a predefined value $N > 1$. They find that infrequent reoptimization (large N) performs better if the customer-to-vehicle assignments are permanent, and that frequent reoptimization (small N) leads to superior solutions if all unserved requests are allowed to be reassigned. Their experiments also indicate that NRR outperforms periodic reoptimization in the cases where dynamic requests arrive non-uniformly. The reoptimization framework proposed by Zou et al. (2021) can be seen as a combination of arrival-triggered and periodic reoptimization. In Zou et al. (2021), vehicle routes are updated by a constructive heuristic each time a new dynamic request is confirmed, and are also periodically reoptimized by a routing heuristic.

Markov decision process

The first MDP model for the VRPDSR is formulated by Thomas and White III (2004). The authors address an anticipatory route selection problem, in which a single vehicle travels from a fixed origin to a fixed destination, meanwhile, dynamic pickup requests arise randomly in the road network. Each time the vehicle arrives at an intersection, a decision determining the next intersection to visit should be made. The MDP suffers from the well-known "curses of dimensionality" (Powell, 2011). That is, real-world problems have huge state, action, and outcome spaces, making the exact solution approaches computationally intractable. In Thomas and White III (2004), only the simple cases with no more than seven dynamic requests are solved by the standard dynamic programming method. For similar but larger MDPs with a single vehicle and at most 25 dynamic requests, Thomas (2007) and Ghiani et al. (2012) use waiting heuristics (see Section 4.2), which are motivated by the model properties in the single-request case, to update the route of the vehicle.

To overcome the curses of dimensionality, Ulmer et al. (2018a) propose an offline ADP algorithm based on the non-parametric value function approximation (nVFA) framework introduced by Powell et al. (2012). They employ this algorithm to solve an MDP for a single-vehicle dynamic pickup problem. They formulate the value function of the MDP on post-decision states and project each post-decision state to a feature vector capturing the current time point and the vehicle's free time budget (depot deadline minus the vehicle's planned return time). Thus, the high-dimensional state space is aggregated to a two-dimensional vector space. The value function mapped to the vector space is approximated via extensive offline simulations, and an efficient online decision policy is then derived. Ulmer et al. (2019a) combine the offline nVFA of Ulmer et al. (2018a) with an online rollout algorithm which is first proposed by Bertsekas and Tsitsiklis (1996) and then formalized by Goodson et al. (2017). At each decision epoch, the rollout algorithm estimates the rewards-to-go of a set of decisions by simulating the nVFA policy over several sample paths, then selects the decision with the highest estimated reward-to-go. The rollout algorithm improves the performance of the nVFA policy but increases the online computation time. Hence, Ulmer et al. (2019a) conduct experiments with different numbers of online and offline sample paths to examine the trade-offs between performance and efficiency.

For a VRPDSR with homogeneous vehicles, Bono et al. (2021) propose a sequential multi-agent MDP in which each vehicle is modeled as an agent. They employ a reinforcement learning method to train a parameterized decision policy online, which produces robust routes efficiently. In recent work, Zhang et al. (2022a) propose potential-based lookahead policies to solve an MDP formulated for the VRPDSR. They develop novel knapsack-based approximation methods to predict the expected reward-to-go, without performing extensive online or offline simulations. For large-scale test instances based on a real-world street network, they demonstrate that the proposed policies outperform a set of rollout policies and a VFA policy in terms of both solution quality and computational efficiency.

The only comparisons between MDPs and reoptimization frameworks are made by Ghiani et al. (2012) and Ulmer (2019) under simple problem settings (single vehicle and no diversion, service time windows, or capacity constraints). Ghiani et al. (2012) show that the MSA (i.e., continuous reoptimization with sample requests) proposed by Bent and Van Hentenryck (2004) requires a sufficiently large number of samples and more computational efforts to achieve the same solution quality as the anticipatory insertion approach (i.e., waiting heuristics designed based on MDP properties). Ulmer (2019) compares a rollout-nVFA policy to an arrival-triggered reoptimization framework without anticipation, and suggests that the former should be used when the DoD is high and the geographical dispersion of customers is wide, while the latter is more suitable for the cases with low DoDs and clustered customers.

Other approaches

Bertsimas and van Ryzin (1991, 1993) and Larsen et al. (2002) evaluate several arbitrary heuristic policies for the DTRP by analytical approaches (queuing theories) and simulations. The stochastic information in these works is mainly used for performance evaluation rather than policy design. Hence, we classify Bertsimas and van Ryzin (1991, 1993) and Larsen et al. (2002) as DD VRPDSRs in this review. Branke et al. (2005) formulate a DS VRPDSR with a single dynamic request as a waiting driver problem (WDP). In the WDP, an optimal a priori route covering all static requests is given, and the available waiting time should be distributed over the locations of static customers to maximize the probability that the dynamic request can be feasibly inserted. The authors prove the NP-completeness of the WDP and propose several waiting heuristics to solve the WDP.

5. Dynamic pickup and delivery problems

By far, the DPDP is the most intensively studied DVRPRR variant in the literature. The latest review on this topic is Berbeglia et al. (2010), while our online appendix fully classifies the relevant papers published from the early 1980s to the 2020s. Similar to the previous section, we divide our complete classification into three tables. First, Table 5 presents the problem characteristics of DPDPs. In column "REJ" (rejection), a "(✓)" indicates that the service provider adopts a dynamic pricing strategy to incentivize customers to accept or reject services, while a "(×)" indicates that requests cannot be rejected but can be subcontracted to third-party companies at high costs (no responses need to be sent to customers). In column "WAI" (waiting), a "(✓)" refers to a reactive or implicit waiting policy (more details are discussed in Section 5.3). As for the fleet setting ("VEH"), "Dyn" means that the number of available vehicles (e.g., ad hoc drivers) dynamically changes during the service period. Uncommon problem features are listed in column "Others", such as transshipment (TRA) strategies and random travel times (RTT). The optimization objectives of DPDPs are shown in the last column, where "Multiple" means that more than three objectives are optimized.

Table 5
Classification of the literature on the DPDP — Part 1: Problem characteristics.

Literature	CAT	Context	Strategies				Side constraints					Others	Objective(s)
			REJ	PR	DIV	WAI	VEH	STW	VTW	MRT	CAP		
Psaraftis (1980)	DD	Passenger	×	–	✓	×	Single	×	×	×	H	–	min W
Psaraftis (1983)	DD	Passenger	×	–	✓	×	Single	H	×	×	×	–	min W
Shen et al. (1995)	DD	Courier	×	–	✓	×	Multi	S	×	×	×	–	min(C+L)
Swihart and Papastavrou (1999)	DD	Passenger	×	–	×	×	Single	×	×	×	1, H	–	min W
Mitrović-Minić et al. (2004)	DD	Courier	×	–	N/S	(✓)	∞	H	×	×	×	–	min C
Mitrović-Minić and Laporte (2004)	DD	Courier	×	–	N/S	(✓)	∞	H	×	×	×	–	min C
Yang et al. (2004)	DS	Freight	✓	✓	N/S	×	Multi	S	×	×	1	–	max(R−C−L)
Coslovich et al. (2006)	DD	Passenger	✓	✓	×	×	Single	H	×	S	×	–	max(R−W)
Fabri and Recht (2006)	DD	N/S	✓	✓	N/S	×	Multi	H	×	×	H	–	min C
Gendreau et al. (2006)	DD	Courier	×	–	×	×	Multi	S	S	×	×	–	min(C+L)
Mes et al. (2007)	DD	Freight	×	–	×	×	Multi	S	×	×	1	–	max(R−C−L)
Cheung et al. (2008)	DD	Freight	✓	✓	N/S	×	Multi	H	×	×	H	RTT	min C
Goel and Gruhn (2008)	DD	Freight	(×)	–	N/S	×	Multi	H	H	×	H	–	min(R−C)
Pureza and Laporte (2008)	DD	N/S	✓	✓	×	(✓)	∞	H	H	×	H	RTT	min(U+V+C)
Sáez et al. (2008)	DS	Passenger	×	–	✓	×	Multi	×	×	×	H	–	min W
Cortés et al. (2009)	DS	Passenger	×	–	✓	×	Multi	×	×	×	H	–	min W
Ghiani et al. (2009)	DS	Courier	×	–	×	✓	Multi	S	×	×	1, ×	–	min L
Beaudry et al. (2010)	DD	Patient	×	–	×	×	Multi	S	H	H	H	–	min(C+L+E)
Bock (2010)	DD	Freight	(×)	–	×	×	Multi	S	×	×	H	TRA&RTT	Multiple
Máhr et al. (2010)	DD	Freight	✓	✓	N/S	×	Multi	H	×	×	1	–	min(T+U)
Mes et al. (2010)	DS	Freight	×	–	×	✓	Multi	S	×	×	1	DP	max(R−C−L)
Kergosien et al. (2011)	DD	Patient	(×)	–	✓	×	Multi	H	S	×	1	–	min(C+O+L)
Schilde et al. (2011)	DS	Patient	×	–	×	×	Multi	S	S	H	H	ERT	min(L+V+C)
Zhang et al. (2011)	DS	Drayage	×	–	×	×	∞	H	×	H	1	–	min V, min C
Berbeglia et al. (2012)	DD	Passenger	✓	✓	×	×	Multi	S	×	S	S	–	Multiple
Hyytiä et al. (2012)	DS	Passenger	×	–	N/S	×	Multi	×	×	×	H	–	min(C+W)
Sheridan et al. (2013)	DS	Passenger	×	–	✓	✓	Multi	×	×	×	1	–	min W
Tirado et al. (2013)	DS	Freight	(×)	–	N/S	×	Multi	H	×	×	H	–	min(C+O)
Ferrucci and Bock (2014)	DD	Courier	×	–	✓	×	Multi	S	×	H	H	RTT	min(C+L)
Schilde et al. (2014)	DD	Patient	×	–	N/S	×	Multi	S	S	S	H	RTT	Multiple
Zolfagharinia and Haughton (2014)	DD	Freight	✓	×	✓	×	Multi	S	×	H	1	–	Multiple
Ma et al. (2015)	DD	Passenger	(✓)	✓	N/S	×	Multi	H	×	×	H	DP	min C
Muñoz-Carpintero et al. (2015)	DS	Passenger	×	–	✓	×	Multi	×	×	×	H	–	min W
Sayarshad and Chow (2015)	DS	Passenger	(✓)	✓	✓	×	Multi	×	×	×	H	DP	max B
van Lon et al. (2016)	DD	Courier	×	–	✓	×	Multi	S	S	×	×	–	min(C+L)
Vonolfen and Affenzeller (2016)	DS	Courier	×	–	N/S	✓	∞	H	×	×	H	–	min(V+C)
Zolfagharinia and Haughton (2016)	DS	Freight	✓	×	✓	✓	Multi	S	×	H	1	–	Multiple
Tirado and Hvattum (2017)	DS	Freight	(×)	–	N/S	×	Multi	H	×	×	H	–	min(C+O)
Zolfagharinia and Haughton (2017)	DD	Freight	(×)	–	✓	×	Multi	S	×	×	1	–	Multiple
Hyland and Mahmassani (2018)	DD	Passenger	×	–	✓	×	Multi	×	×	×	1	–	min(C+W)
Sayarshad and Gao (2018)	DS	Passenger	(✓)	✓	✓	×	Multi	×	×	×	H	DP	max B
Arslan et al. (2019)	DD	E-commerce	×	–	×	×	Dyn	H	H	H	×	CS	min C
Bertsimas et al. (2019)	DD	Passenger	✓	✓	✓	×	Multi	H	×	×	1	–	max R
He et al. (2019)	DD	Meal	×	–	✓	×	Multi	S, U	×	×	H	–	min(C+W)
Liu (2019)	DD	Meal	×	–	✓	×	Multi	S, U	×	H	H	UAV	Multiple
Steever et al. (2019)	DS	Meal	×	–	✓	×	Multi	S, U	H	×	H	–	min(W+L)
Duan et al. (2020)	DS	Passenger	✓	✓	✓	✓	Multi	H	×	×	1	–	max R
Karami et al. (2020)	DD	N/S	×	–	×	×	Multi	S	S	×	×	–	min(L+C)
Los et al. (2020)	DD	Freight	✓	✓	✓	(✓)	Dyn	H	H	×	H	–	max(R−C)
Tafreshian et al. (2021)	DS	Passenger	✓	×	×	×	Multi	H	×	H	H	–	max R
Ulmer et al. (2021)	DS	Meal	×	–	×	(✓)	Multi	S, U	×	×	×	–	min L
Ghiani et al. (2022)	DS	N/S	×	–	×	×	Multi	S	×	×	×	–	min L

(*continued on next page*)

5.1. Applications

While the VRPDSR mostly involves courier or freight transportation, the logistical contexts of the DPDP are more diverse and 50% of them are related to the transportation of goods (parcels, freight, food, etc.). Ferrucci and Bock (2014) address an express courier service application with dynamic P&D requests, traffic congestion, and vehicle disturbances. Zolfagharinia and Haughton (2014) consider a long-haul full-truckload transportation problem and investigate the benefits of the advance information regarding the dynamic P&D requests. The

Table 5 (*continued*).

Literature	CAT	Context	Strategies				Side constraints					Others	Objective(s)
			REJ	PR	DIV	WAI	VEH	STW	VTW	MRT	CAP		
Haferkamp and Ehmke (2022)	DS	Passenger	✓	✓	×	×	Multi	H	×	×	×	–	max R
Kullman et al. (2022)	DS	Passenger	✓	✓	✓	✓	Multi	H	×	×	1	EV	max(R−C)

Acronyms

CAT	Category	REJ	Rejection	PR	Prompt reply	DIV	Diversion
WAI	Waiting	VEH	Number of vehicles	STW	Service time window	VTW	Vehicle time window
MRT	Max ride time	CAP	Vehicle capacity	TRA	Transshipment	DP	Dynamic pricing
ERT	Expected return trip	RTT	Random travel time	CS	Crowdsourcing	UAV	Unmanned aerial vehicle
EV	Electric vehicle						

Side constraints

H	Hard constraints	S	Soft constraints	U	Uncertain release times	1	Vehicle capacity equals 1

Objectives

W	Waiting/ride time of customers	C	Travel cost/time/distance	L	Penalty of lateness
R	Revenue/number of served requests	U	Number/penalty of unserved requests	V	Number/cost of vehicles used
P	Probability of accepting new requests	E	Penalty of earliness	T	Travel time of empty vehicles
O	Outsourcing/subcontracting cost	B	Social welfare/benefit		

DPDP studied by Tirado and Hvattum (2017) originates from a maritime transportation application in which a fleet of vessels transports cargoes between a set of ports. In recent years, there is an increasing research interest in crowdsourced delivery, e-commerce, and online food ordering and delivery services. For example, Arslan et al. (2019) introduce a crowdsourced DPDP in which the orders of an e-commerce platform are fulfilled by self-owned vehicles as well as ad hoc drivers with uncertain availability. In the restaurant meal delivery problem studied by Ulmer et al. (2021), customers order meals from different restaurants via a website or mobile application, and with delivery deadlines and uncertain ready times of meals taken into account, the service provider operates a fleet of vehicles to pick up the ordered meals from restaurants and deliver to different customer locations.

Around 43% of DPDPs deal with the transportation of people (passengers, patients, etc.), and these problems can also be referred to as dynamic dial-a-ride problems (DARPs). A typical DARP application is taxi routing. Bertsimas et al. (2019) study a large-scale online taxi routing problem in New York City, where each dynamic request is associated with a pickup location and a drop-off location, and a fleet of taxis serves the requests while respecting the pickup time windows. Duan et al. (2020) address an autonomous taxi dispatching problem, in which each taxi manages its own long-term route autonomously, meanwhile a centralized dispatcher optimizes the short-term routes of all taxis. As in most taxi routing problems, Bertsimas et al. (2019) and Duan et al. (2020) assume that each taxi can serve at most one customer at a time, whereas in the DARPs based on bus or shuttle services, ride sharing is allowed (e.g., Tafreshian et al., 2021). Moreover, the taxi sharing system considered by Ma et al. (2015) differs from most taxi routing problems in that the passengers that are already on board a taxi can decide whether to share the ride with a new passenger and get compensated for the detour. Besides passenger transportation, some DPDP research deals with the transportation of patients in hospitals (Beaudry et al., 2010), between care units (Kergosien et al., 2011), or between patients' homes and hospitals (Schilde et al., 2011, 2014).

5.2. Comparison of problem characteristics with the VRPDSR

The DPDP can be seen as an extension of the VRPDSR. Each customer in the VRPDSR has only one location to visit, whereas, in the DPDP, each customer specifies a pickup location and a delivery location, which are paired by a precedence constraint requiring the former to be visited prior to the latter. The decision strategies adopted in the VRPDSR, including rejections of requests, en-route diversions, and waiting strategies, are also applicable to the DPDP. In addition, four new strategies have been introduced and studied for the DPDP:

(1) The *transshipment* strategy allows orders to be transferred from one vehicle to another.

(2) The *dynamic pricing* strategy sets flexible service prices for customers.

(3) The *crowdsourcing* strategy addresses the scheduling and routing of crowdsourced vehicles.

(4) The *unmanned aerial vehicle* (UAV) strategy utilizes drones for P&D services.

Moreover, there are some slight differences in the side constraints of the DPDP and the VRPDSR:

(1) In meal P&D problems, the ready times of meals are usually subject to uncertainty (He et al., 2019; Liu, 2019; Steever et al., 2019; Ulmer et al., 2021).

(2) About 22% of DPDPs are subject to vehicle availability constraints, which can be a deadline before which vehicles must return to a fixed location (e.g., Arslan et al., 2019) or a period of time during which a vehicle can serve customers (e.g., Los et al., 2020). All these constraints are referred to as vehicle time windows in Table 5.

(3) About 22% of DPDPs are subject to ride/driving time constraints. For example, Beaudry et al. (2010), Schilde et al. (2011, 2014), and Berbeglia et al. (2012) specify maximum ride times for patients and passengers to reduce their inconvenience. Ferrucci and Bock (2014) and Zolfagharinia and Haughton (2014, 2016) limit the driving times of drivers to control workloads. Moreover, in the meal P&D problem with drones studied by Liu (2019), a drone's maximum travel distance is limited by its battery capacity.

(4) All VRPDSRs listed in Table 2 are lighter-than-truckload (LTL) problems, meaning that multiple orders can be loaded on the same vehicle, as long as no capacity constraints are violated. Such order consolidations are not allowed in 28% of DPDPs, as indicated by column "CAP" in Table 5. These DPDPs include the full-truckload (FTL) problems where each truck is assumed to have a capacity of one (e.g., Zolfagharinia and Haughton, 2017) and the taxi routing problems where each taxi can transport at most one passenger at a time (e.g., Bertsimas et al., 2019).

Besides the above differences, *expected return trips* and *random travel times* are two special features of some DPDPs. The former, considered in a patient transportation problem, addresses the probability that the patients who have previously been transported to a hospital may need to be transported back home in the future (Schilde et al., 2011). The latter captures the variations in vehicles' travel speeds caused by rush hours, accidents, construction sites, and poor weather conditions. It is worth mentioning that the DPDP studied by Schilde et al. (2014) accounts for stochastic travel speeds, but is regarded as a DD problem in this review since there is no probabilistic information on request arrivals.

5.3. Decision strategies

Rejections

Rejections of dynamic requests are allowed in 28% of DPDPs. Regardless of the problem formulation (DD or DS), most rejection strategies adopted in DPDPs are reactive in the sense that a dynamic request is rejected only if it cannot be feasibly inserted into any vehicle's route. The only two anticipatory rejection strategies are proposed by Yang et al. (2004) and Haferkamp and Ehmke (2022). Vehicle capacity and hard time windows are the common constraints that limit vehicles' service capability. In 80% of the DPDPs with rejections, new requests are immediately accepted or rejected by service providers.

Two types of implicit rejection strategies are studied for DPDPs. The first type refers to the cases where dynamic P&D requests can be partially outsourced to third-party service providers. Goel and Gruhn (2008) assume that the unscheduled requests at each decision epoch (which occurs every hour) are served by external carriers. They aim at maximizing the revenue of serving requests with the self-owned fleet and minimizing transportation costs. In Bock (2010), Kergosien et al. (2011), Tirado and Hvattum (2017), and Zolfagharinia and Haughton (2017), subcontracting a request is assumed to incur a high cost, which is incorporated into the objective functions to be minimized. In contrast to the explicit rejections, subcontracting decisions have little impact on customer satisfaction and hence do not need to be made immediately. The second type of implicit rejection strategies is associated with dynamic pricing. Under these strategies, customers confirm or cancel their requests according to their willingness-to-pay and the service prices quoted by the service provider.

Diversions

Diversions of en-route vehicles are enabled in 43% of DPDPs. In LTL transportation, ride-sharing, and meal P&D services, an en-route vehicle can be diverted from its planned destination at any time, as long as no constraints (e.g., vehicle capacity and time windows) are violated (e.g., Ferrucci and Bock, 2014; van Lon et al., 2016; Liu, 2019; Steever et al., 2019). In the FTL transportation and taxi scheduling problems, only the empty vehicles that are moving to pickup locations can be diverted because the nonempty vehicles must first transport the on-board freight or passengers to their destinations (e.g., Sheridan et al., 2013; Zolfagharinia and Haughton, 2014, 2016, 2017).

Under the diversion strategy of Sheridan et al. (2013), a full reassignment involving all empty vehicles and pending requests is carried out each time new customer information is revealed. The simulation study of Sheridan et al. (2013) confirms that the diversion strategy significantly reduces the expected customer waiting times. However, for an FTL transportation problem, Zolfagharinia and Haughton (2017) show that diverting en-route vehicles does not lead to significant reductions in total costs. In the other relevant papers, the benefits of diversions in DPDPs are not studied.

Waiting

Similar to VRPDSRs, a DPDP may benefit from a waiting strategy that requires vehicles to proactively wait at some locations for the possibility of receiving new requests in the vicinity. Thirteen percent of the existing DPDP research studies waiting strategies. In Ghiani et al. (2009), the available waiting time is distributed over the nodes included in the planned routes such that the expected service delays are minimized. To reduce computational complexity, the authors define a maximum total waiting time and require the waiting time at each node to be a multiple of a given value. When a vehicle has no request left to serve, it is repositioned to one of several stochastic medians of the service area. In Mes et al. (2010), when a vehicle finishes a delivery, it may move to the pickup location of the next planned request immediately, or wait at the current or a different location. The value functions (i.e., expected rewards-to-go) of these decisions are approximated in offline simulations. Sheridan et al. (2013) assume that a set of vehicle waiting stations have been determined in advance, and they direct each vehicle that has served all its requests to the closest station with the fewest waiting vehicles. The waiting strategy of Duan et al. (2020) is dependent on the route plans covering real requests and predicted requests (retrieved from historical data). Under their strategy, assigning a predicted request to a vehicle means that the vehicle will move to the origin of the predicted request and wait there. Zolfagharinia and Haughton (2016) make waiting decisions using the MSA proposed by Bent and Van Hentenryck (2004). In each sample scenario, if a vehicle is assigned a sample request, it is suggested to wait at the delivery location of the previous real request. In the distinguished plan to be executed, a vehicle is required to wait if the number of scenarios with waiting suggestions is greater than a predefined threshold.

As mentioned in Section 4.2, the reactive waiting policies that do not make use of any stochastic information are not considered as waiting strategies in this review. Nevertheless, several reactive waiting policies (marked by "(✓)" in Table 5) developed for DPDPs with service time windows deserve some discussion. Mitrović-Minić and Laporte (2004) propose the *drive-first* and *wait-first* waiting policies. Drive-first means that each vehicle always moves to the next planned location as early as possible and that the vehicle waits at the next location if the time window is not opened. Wait-first requires each vehicle to wait at its current location until the latest feasible departure time that does not cause any time window violations. Mitrović-Minić and Laporte (2004) also propose two dynamic waiting policies under which vehicles follow the drive-first policy when traveling between the locations that are close to each other and follow the wait-first policy (or wait for a shorter period of time) when the next location is far away. In Pureza and Laporte (2008), vehicles are first scheduled to arrive at each customer location as early as possible, but no earlier than the opening of the time window. Given that triangular inequalities may not hold with dynamic travel times, Pureza and Laporte (2008) then employ a fastest-paths-based waiting policy to exploit the chances of reducing the travel times between two nodes by adding an intermediate node. If a faster path can be found in this way, the vehicle can wait for more time at the upstream node. Besides, Los et al. (2020) introduce a waiting policy that requires vehicles to wait at each location for 20% of the available waiting time. These reactive waiting policies are easy to implement because of their simple structures and independence of stochastic information.

Mitrović-Minić and Laporte (2004) show that the wait-first policy outperforms the drive-first policy because the new requests received during the waiting period of a vehicle can be used to build better routes. Moreover, Vonolfen and Affenzeller (2016) compare the waiting policies/strategies of Mitrović-Minić and Laporte (2004), Branke et al. (2005), and Ichoua et al. (2006) (see Section 4.2) on a set of DPDP instances with service time windows. They also propose an intensity-based waiting strategy which determines the waiting time at each location by a function of three factors: travel time to the next planned location, customer intensity of the current location (computed from historical data), and total available waiting time. They find that the performances of these waiting policies/strategies are affected by instance settings. Overall, the intensity-based strategy and the variable strategy of Branke et al. (2005) are more robust than the other tested waiting policies/strategies.

Table 5 also identifies the implicit waiting policies (also marked by "(✓)") which do not directly require vehicles to wait but postpone the decisions on request-to-vehicle assignments. In Pureza and Laporte (2008), only the urgent new requests are immediately assigned to vehicles, while the non-urgent requests are postponed to be considered when a vehicle finishes the service at a pickup or delivery location. As such, more requests are considered when the route plans are reoptimized and better performance can thereby be achieved. Similarly, for a meal P&D problem, Ulmer et al. (2021) postpone the assignments of the "*not yet important*" requests. Under their policy, a request can be assigned to a vehicle only when the restaurant of this request is the next planned destination of the vehicle. In addition, Ulmer et al. (2021) add a time buffer (tuned by sample average approximation) to the planned arrival times at customer locations so as to absorb the uncertainty related to ready times and request arrivals. Their experimental results validate that the postponement and time buffer policies reduce the total delays concerning customers' service deadlines.

Transshipment

In most DPDPs, the pickup, transportation, and delivery of a request must be performed by the same vehicle. The only exception is the multimodal DPDP addressed by Bock (2010), in which loads can be transshipped between self-owned and third-party vehicles. To reduce the computational complexity, the author restricts the transshipment locations to a finite set of hubs and depots. The inconveniences caused by transshipments, such as extra loading and unloading works and hub utilization, are minimized in an objective function which also includes the vehicle travel costs and the time window violation costs. The experimental results of Bock (2010) show that the transshipment strategy allows more flexibility in handling the dynamism in request arrivals and is of significant importance for coping with unexpected traffic jams and vehicle breakdowns.

Dynamic pricing

Dynamic pricing strategies which regulate the prices offered to customers are applied to 7% of DPDPs. Mes et al. (2010) propose a procurement auction mechanism to coordinate the requests and vehicles in a DPDP. In this mechanism, every newly arrived request initiates a reverse Vickrey auction for vehicles to bid for the right of serving the request. The bidding price submitted by each vehicle is determined by the direct cost of inserting the new request into the planned route and the opportunity cost capturing the expected loss in future profits. The vehicle which offers the lowest bid wins the auction at the price of the second lowest bid. The main conclusion of Mes et al. (2010) is that taking the opportunity cost into account significantly increases the total revenue of the service provider in various market settings.

Ma et al. (2015) propose a dynamic pricing strategy to deal with the ride-sharing issues in a dynamic DARP. They allow the passengers that are already on board a taxi to decide whether they want to share the ride with new passengers. Specifically, each time a new passenger is assigned to a non-empty taxi, the passengers on board are informed of the changes in their fares and travel times, based on which they accept or reject this ride-sharing. To regulate fare changes, Ma et al. (2015) introduce three monetary constraints to ensure: (1) passengers on board do not pay more after the joining of a new passenger, (2) passengers on board are compensated for the increases in their travel times, and (3) taxi drivers' incomes (depending on the total travel distances) are increased. Moreover, Ma et al. (2015) develop a cloud-mobile-based ride-sharing system to implement their dynamic pricing strategy.

Sayarshad and Chow (2015) and Sayarshad and Gao (2018) combine dynamic DARPs with dynamic pricing as well. Each time a new request arrives, they solve a TSP-like problem for each vehicle to accommodate the new request. Then, for each updated route, a price reflecting the expected average service lateness is computed using queuing theory. Finally, the new request is assigned to the vehicle whose updated route leads to the highest social welfare, which increases with the number of served customers and decreases with customers' expenses and service delays. Sayarshad and Chow (2015) highlight the advantage of their single-server-queue-based non-myopic pricing strategy against a marginal pricing strategy that only considers the route lengths of vehicles. Sayarshad and Gao (2018) show that social welfare can further be improved by basing the pricing strategy on a more realistic multiple-server queue.

Crowdsourcing

Crowdsourcing can be interpreted as a strategy that allows individuals (i.e., the crowd) to participate in the service of a company. Because of the development of e-commerce, many retailers and logistics companies are now facing the challenge of satisfying the increasing demands for faster and cheaper delivery services in a cost-effective way. To overcome this challenge, some e-commerce platforms, such as Amazon and Walmart, adopt crowdsourcing in their last-mile deliveries (Alnaggar et al., 2021). Among the papers listed in Table 5, Arslan et al. (2019) is the only one that addresses crowdsourced DPDP. In their problem, Arslan et al. (2019) assume that P&D requests are mainly served by ad hoc drivers (i.e., individuals that participate in crowdsourcing) and that the service provider owns a fleet of dedicated vehicles to serve the requests that cannot be served by ad hoc drivers. The problem is characterized by the uncertainty in the ad hoc drivers' available time windows, origins, and destinations. Moreover, the service provider must respect each ad hoc driver's departure time flexibility and stop willingness (i.e., number of P&D locations that the driver is willing to visit), which both have significant impacts on system performance. Through extensive numerical experiments, Arslan et al. (2019) find that crowdsourcing reduces transportation costs compared to a conventional system with dedicated vehicles only. Furthermore, they suggest that crowdsourcing brings the highest benefit when in-store customers serve as ad-hoc drivers to deliver the same store's online orders.

Unmanned aerial vehicles (drones)

Compared to the conventional vehicles that are considered in the majority of VRP variants, battery-powered unmanned aerial vehicles (UAVs) or drones have the advantages of high speed and autonomy, low emission and noise, independence from traffic congestion, etc. Hence, UAVs or drones are suitable for the transportation of light items in last-mile deliveries. Liu (2019) studies a DPDP in which a fleet of drones is used to deliver meals from restaurants to customers. In Liu (2019), the movements of drones are straight lines on a Euclidean plane and are not restricted by a road network, but the payload and battery capacity of drones are limited. Each drone can carry a small number of standard-sized orders, and different types of orders (e.g., hot and cold food) cannot be loaded on the same drone. Battery capacity is measured by the product of payload and flight time (e.g., 90 min with the maximum payload) and the energy consumption rate of a drone is increasing in its payload. In addition to the locations of pickups (restaurants) and deliveries (customers), the DPDP of Liu (2019) includes a set of charging depots where the drones whose battery levels are lower than a threshold can be quickly swapped with fully charged batteries. With these special features (associated with the utilization of drones) taken into account, the author models the DPDP as a mixed-integer program (MIP) that is periodically reoptimized.

5.4. Problem instances

From the DPDP instances summarized in Table 6, it can be seen that the graph types of DPDPs are more various than those of VR-PDSRs. Column "Graph" shows that DPDP instances are formulated on Euclidean planes, Manhattan-style planes (taxicab geometry), synthetic planes (geometry not specified), hospital campuses, and networks of grids/roads/cities/ports/airports (with numbers of nodes given in parentheses). We can also observe that the DPDP instances in which every vehicle can serve at most one request at a time tend to be larger than the other DPDP instances in terms of the number of nodes, vehicles, and requests. This implies that disallowing order consolidation may reduce the complexity of a DPDP. Column "Horizon" shows that most DPDP instances have a short planning horizon of several hours, while in a few instances with the contexts of maritime or long-haul transportation, the planning horizons are several weeks or one year. Regarding the data sources, there is no canonical test benchmark for

Table 6
Classification of the literature on the DPDP — Part 2: Problem instances.

Literature	Avail.	Graph	Vehicle(s)	Requests	DoD (%)	Horizon (h)	Source(s)
Psaraftis (1980)	✓	Euclidean	1	10	40	4/3	Arbitrary
Shen et al. (1995)	×	Road (20 000)	12	140	100	6	Realistic
Swihart and Papastavrou (1999)	×	Euclidean	1	N/S	100	N/S	Arbitrary
Mitrović-Minić et al. (2004)	×	Synthetic	∞	{100, 500, 1000}	100	10	Realistic
Mitrović-Minić and Laporte (2004)	×	Synthetic	∞	{100, ..., 1000}	100	10	Realistic
Yang et al. (2004)	×	Euclidean	10	1000	100	N/S	Arbitrary
Coslovich et al. (2006)	×	Synthetic	1	{25, 30, 50}	{20, 30, 33}	N/S	Arbitrary
Fabri and Recht (2006)	×	Synthetic	{10, 20, 100}	{100, 400, 1000}	100	N/S[a]	Caramia
Gendreau et al. (2006)	×	Synthetic	{10, 20}	{96, ..., 248}	100	{4, 7.5}	Arbitrary
Mes et al. (2007)	×	Nodes ({11, 20})	{20, 21, 22}	{1800, 5760}	100	{24, 144}	Arb., real.
Cheung et al. (2008)	×	Euclidean	12	{29, ..., 204}	{1.96, ..., 7.41}	N/S	Arbitrary
Goel and Gruhn (2008)	×	Airports (39)	{100, 250, 500}	{500, 1250, 2500}	40	10	Realistic
Pureza and Laporte (2008)	×	Synthetic	∞	{50, 100, 200}	{0, ..., 100}	N/S	Li & Lim
Sáez et al. (2008)	×	Euclidean	9	120	100	2	Realistic
Cortés et al. (2009)	×	Euclidean	9	120	100	3	Realistic
Ghiani et al. (2009)	×	Grid (25)	{18, 36, 54}	{100, 200, 300}	100	8	Arbitrary
Beaudry et al. (2010)	×	Hospital (N/S)	11	{182, ..., 302}	{98.6, ..., 93.7}	≈9.75	Realistic
Bock (2010)	×	Road (5600)	{38, ..., 55}	{124, ..., 372}	<100	{2, 4}	Realistic
Máhr et al. (2010)	×	Road (N/S)	40	65	{0, 25, ..., 100}	24	Realistic
Mes et al. (2010)	×	Grid ({3, 9})	10	{600, 800, 900}	100	200	Arbitrary
Kergosien et al. (2011)	×	N/S	15	130	{65, ..., 75}	15	Realistic
Schilde et al. (2011)	×	Road (N/S)	>1	≈{150, ..., 719}	≈{60.0, ..., 71.0}	10	Realistic
Zhang et al. (2011)	×	Road (N/S)	∞	{50, 75}	{10, 24, 40}	12	Realistic
Berbeglia et al. (2012)	×	Euclidean	{4, 5, ..., 8}	{40, ..., 96}	75	12	Ropke
Hyytiä et al. (2012)	×	Euclidean	{120, ..., 330}	36 000	100	10	Arbitrary
Sheridan et al. (2013)	×	Euclidean	{1, 10, 20, 100}	N/S[b]	100	24	Automod
Tirado et al. (2013)	×	Ports ({16, 32})	{6, 12, 24}	N/S	<100	8640	Realistic
Ferrucci and Bock (2014)	×	Road (126 500)	{70, 140, 210}	{400, 600}	{33.3, 50}	2	Realistic
Schilde et al. (2014)	×	Road (N/S)	>1	{215, ..., 762}	{0, 15, ..., 90, 100}	10	Realistic
Zolfagharinia and Haughton (2014)	×	Cities (50)	{6, 20}	{45, 90, 150, 300}	ALI[c]	504	Realistic
Ma et al. (2015)	(✓)	Road (106 579)	7088	N/S[d]	100	0.5	Realistic
Muñoz-Carpintero et al. (2015)	×	Euclidean	{7, 9, 11}	8	100	2	Arbitrary
Sayarshad and Chow (2015)	×	Euclidean	{1, 2, ..., 5}	13	100	N/S	Hyytiä
van Lon et al. (2016)	×	Euclidean	10	360	100	12	Arbitrary
Vonolfen and Affenzeller (2016)	✓	Euclidean	∞	50	{0, 10, ..., 100}	N/S	Pankratz
Zolfagharinia and Haughton (2016)	×	Cities (50)	6	{45, 90}	ALI[c]	504	Realistic
Tirado and Hvattum (2017)	×	Ports ({16, 32})	{6, 12, 24}	N/S	<100	8640	Realistic
Zolfagharinia and Haughton (2017)	×	Cities (50)	6	{45, 90}	ALI[c]	504	Realistic
Hyland and Mahmassani (2018)	×	Manhattan	{130, ..., 800}	{4000, ..., ≈96 000}	100	{4, 24}	Arb., real.
Sayarshad and Gao (2018)	✓	Road (N/S)	70	99	100	1	Realistic
Arslan et al. (2019)	×	Euclidean	100	100	100	10	Arbitrary
Bertsimas et al. (2019)	✓	Road (4324)	4000	26 109	100	1.5	Realistic
He et al. (2019)	×	Road (N/S)	5	298	100	2.5	Realistic
Liu (2019)	✓	Euclidean	{10, 50}	{20, 353}	100	6	Arbitrary
Steever et al. (2019)	×	Road (N/S)	{15, 40}	≈{27, 65}	100	4	Realistic
Duan et al. (2020)	✓	Road (5023)	{300, 500, 700}	48 592	100	24	Realistic
Karami et al. (2020)	×	Euclidean	10	180	100	10	van Lon
Los et al. (2020)	✓	Euclidean	{75, 100, 150}	1000	100	10	Arbitrary
Tafreshian et al. (2021)	✓	Road (184)	100	≈50 000	100	3	Realistic
Ulmer et al. (2021)	✓	Euclidean	{15, ..., 120}	{180, ..., 2400}	100	7	Realistic
Ghiani et al. (2022)	×	Euclidean	{6, ..., 25}	{25, ..., 940}	100	8	Arbitrary
Haferkamp and Ehmke (2022)	✓	Road (N/S)	{2, 6, ..., 18}	{36, 108, ..., 324}	100	{0.6, 1.8, ..., 5.4}	Realistic
Kullman et al. (2022)	✓	Taxi zones (61)	{14, ..., 140}	{1400, ..., 14 000}	100	24	Realistic

[a]The authors assume an average arrival rate of 1 request per minute.
[b]An arrival rate is defined, but the time unit is not specified.
[c]Advanced load information: every request is known to the service provider 24, 48, or 72 hours before its earliest pickup time.
[d]Approximately in $[10^4, 10^5]$.

the DPDP, and 56% of DPDP instances are based on realistic cases. For example, the taxicab data of New York City are the basis of many dynamic DARP instances. Besides, 33% of DPDP instances are arbitrarily generated or derived from the static P&D problem instances of Li and Lim (2001) and Ropke et al. (2007). Sheridan et al. (2013) generate their instances using the Automod software, while the other authors reuse the existing DPDP instances created by Caramia et al. (2002), Pankratz (2005), Hyytiä et al. (2012), and van Lon et al. (2016).

5.5. Solution approaches

The solution approaches developed for DPDPs are summarized in Table 7. The formulations of DPDPs have some similarities with those of VRPDSRs: (1) 2/3 of DPDPs are formulated as sequentially reoptimized SVRPs, among which 1/3 incorporate stochastic information, mainly by generating sample requests; (2) there is an increasing number of papers studying DS DPDPs with MDP formulations in recent years. In addition to the two most common formulations, queuing models and multi-agent systems are also formulated for DPDPs: the former are used to analyze the performances of specific decision policies in DARPs with stochastic information, while the latter are suitable for the DPDPs in which every vehicle can make decisions autonomously.

Overall, nearly 60% of DPDP solution approaches do not make use of any stochastic information. Although most authors do not explain why using DD formulations, one reason could be that many DPDP instances are derived from real-world applications with no available stochastic information. Other possible reasons could be that realistic

Table 7
Classification of the literature on the DPDP — Part 3: Solution approaches.

Literature	Formulation			Algorithms	
	Framework/Model	Decision epochs	Anticipation	Offline	Online
Psaraftis (1980)	Reopt-SVRPs	Request arrivals	–	Backward dynamic programming	
Psaraftis (1983)	Reopt-SVRPs	Request arrivals	–	Forward dynamic programming	
Shen et al. (1995)	Expert system	Request arrivals	–	Neural network	–
Swihart and Papastavrou (1999)	Queuing	Request arrivals	–	Analytical study	
Mitrović-Minić et al. (2004)	Reopt-SVRPs	Periodic	–	–	CH+tabu search
Mitrović-Minić and Laporte (2004)	Reopt-SVRPs	Periodic	–	–	CH+tabu search
Yang et al. (2004)	Reopt-SVRPs	Request arrivals	Opportunity cost	–	CPLEX
Coslovich et al. (2006)	Reopt-SVRPs	Request arrivals	–	CH	2-phase insertion
Fabri and Recht (2006)	Reopt-SVRPs	Request arrivals	–	–	Local search
Gendreau et al. (2006)	Reopt-SVRPs	Continuous	–	–	ECN+tabu search
Mes et al. (2007)	Multi-agent	Request arrivals	–	–	Auction+CH
Cheung et al. (2008)	Reopt-SVRPs	New information	–	Heuristic+GA	Heuristic
Goel and Gruhn (2008)	Reopt-SVRPs	Periodic	–	LNS/RVNS	
Pureza and Laporte (2008)	Reopt-SVRPs	Arrivals & stops	–	CH+implicit waiting heuristic	
Sáez et al. (2008)	HAPC	Request arrivals	Samples	–	GA
Cortés et al. (2009)	HAPC	Request arrivals	Samples	–	PSO
Ghiani et al. (2009)	Reopt-SVRPs	Request arrivals	Samples	–	CH/local search
Beaudry et al. (2010)	Reopt-SVRPs	Request arrivals	–	CH+tabu search	
Bock (2010)	Reopt-SVRPs	Continuous	–	CH+variable neighborhood search	
Máhr et al. (2010)	Multi-agent	Request arrivals	–	Auction+CH+improvement heuristics	
Mes et al. (2010)	Multi-agent	Request arrivals	Value function	nVFA	Auction+CH
Kergosien et al. (2011)	Reopt-SVRPs	Request arrivals	–	Tabu search	
Schilde et al. (2011)	Reopt-SVRPs	Continuous	Samples	CH+variable neighborhood search	
Zhang et al. (2011)	Reopt-SVRPs	Periodic	Samples	Branch-and-price+CH	
Berbeglia et al. (2012)	Reopt-SVRPs	Continuous	–	CH	Tabu search+CP
Hyytiä et al. (2012)	MDP+queuing	Request arrivals	Value function	–	CH
Sheridan et al. (2013)	Queuing	Arrivals & idles	Waiting	–	DNN
Tirado et al. (2013)	Reopt-SVRPs	Arrivals & stops	Samples	MSA/BRH+tabu search	
Ferrucci and Bock (2014)	Reopt-SVRPs	Continuous	–	Tabu search	
Schilde et al. (2014)	Reopt-SVRPs	Continuous	–	CH+variable neighborhood search	
Zolfagharinia and Haughton (2014)	Reopt-SVRPs	Periodic	–	Gurobi	
Ma et al. (2015)	Reopt-SVRPs	Request arrivals	–	–	CH
Muñoz-Carpintero et al. (2015)	HAPC	Request arrivals	Samples	–	Evolutionary algorithm
Sayarshad and Chow (2015)	MDP+queuing	Request arrivals	Pricing	–	TSPPD heuristic
van Lon et al. (2016)	Reopt-SVRPs	Request arrivals	–	CH+2-opt reoptimization	
Vonolfen and Affenzeller (2016)	Reopt-SVRPs	Request arrivals	Waiting	Training	CH+tabu search+waiting
Zolfagharinia and Haughton (2016)	Reopt-SVRPs	Periodic	Samples	Deadhead coefficient policy+MSA	
Tirado and Hvattum (2017)	Reopt-SVRPs	Arrivals & stops	Samples	MSA/BRH+local search	
Zolfagharinia and Haughton (2017)	Reopt-SVRPs	Periodic	–	Gurobi	
Hyland and Mahmassani (2018)	Reopt-SVRPs	Periodic	–	–	Gurobi
Sayarshad and Gao (2018)	MDP+queuing	Request arrivals	Pricing	–	TSPPD heuristic
Arslan et al. (2019)	Reopt-SVRPs	New information	–	–	Recursive algorithm
Bertsimas et al. (2019)	Reopt-SVRPs	Periodic	–	–	Backbone algorithm
He et al. (2019)	Multi-agent	Request arrivals	–	–	CH
Liu (2019)	Reopt-SVRPs	Periodic	–	–	MIP-based heuristic
Steever et al. (2019)	Reopt-SVRPs	Request arrivals	Equity & Dispersion	–	Auction-based heuristic
Duan et al. (2020)	Reopt-SVRPs	Hybrid	Predicted requests	–	CPLEX+decentralization
Karami et al. (2020)	Reopt-SVRPs	Periodic	–	–	CH+local search
Los et al. (2020)	Multi-agent	Request arrivals	–	–	Auction+CH
Tafreshian et al. (2021)	Reopt-SVRPs	Periodic	Samples	Local search	Cost–benefit analysis
Ulmer et al. (2021)	MDP	Hybrid	Time buffer	SAA	Anticipatory assignment
Ghiani et al. (2022)	MDP	Request arrivals	Policy function	–	Policy function approximation
Haferkamp and Ehmke (2022)	MDP	Request arrivals	Samples	Tuning	LNS
Kullman et al. (2022)	MDP	Hybrid	Q-values	–	Deep reinforcement learning

Acronyms

Reopt-SVRPs	Reoptimization of static VRPs	HAPC	Hybrid adaptive predictive control	MDP	Markov decision process
CH	Constructive heuristic	ECN	Ejection chain neighborhood	GA	Genetic algorithm
RVNS	Reduced variable neighborhood search			LNS	Large neighborhood search
nVFA	Non-parametric value function approximation			PSO	Particle swarm optimization
DNN	Dynamic nearest neighborhood	MSA	Multiple scenario approach	BRH	Branch-and-regret heuristic
TSPPD	TSP with P&D	SAA	Sample average approximation	MIP	Mixed-integer programming

instances are already hard to solve due to their large sizes (see Section 5.4), and that exploiting stochastic information further increases the computational complexity.

The existing research on DS DPDPs has confirmed the benefits of exploiting the probabilistic information on future P&D requests. For a wide variety of DPDP applications, including FTL transportation (Ghiani et al., 2009; Zolfagharinia and Haughton, 2016), LTL transportation (Ghiani et al., 2009; Vonolfen and Affenzeller, 2016; Steever et al., 2019), DARPs with ride-sharing (Sáez et al., 2008; Cortés et al., 2009; Muñoz-Carpintero et al., 2015; Sayarshad and Chow, 2015; Haferkamp and Ehmke, 2022) and without ride-sharing (Sheridan et al., 2013), extensive computational studies have validated that anticipatory (or proactive) scheduling and routing policies outperform myopic (or reactive) ones.

Reoptimization of SVRPs

Continuous SVRP reoptimization is a popular solution framework for DPDPs among the research before 2015. The continuous reoptimization frameworks adopted in Gendreau et al. (2006) and Berbeglia et al. (2012) are similar to the one proposed by Gendreau et al. (1999) for a VRPDSR. They all run tabu search metaheuristics continually, but Gendreau et al. (2006) also adopt an ejection chain neighborhood algorithm to improve the planned routes and Berbeglia et al. (2012) use a constraint programming model to check the feasibility of serving each new request. Bock (2010) proposes a new continuous reoptimization framework based on rolling horizons. This framework decomposes the entire service period into a sequence of short uniform time intervals, called anticipation horizons, and maintains a relevant plan which is actually executed and a theoretical plan which is continuously improved. At the beginning of each anticipation horizon, a static problem is formulated by freezing the decisions that will be executed in the current horizon according to the relevant plan and incorporating the new information revealed during the last horizon. Then, an analytical plan is constructed and continually improved by solving the static problem using a constructive heuristic and variable neighborhood search (VNS). At the end of each anticipation horizon, the theoretical plan replaces the relevant plan if it outperforms the latter. The same framework is adopted by Ferrucci and Bock (2014) with a different improvement heuristic (tabu search). Another continuous reoptimization approach is the DS VNS algorithm proposed by Schilde et al. (2011, 2014). This algorithm is dynamic in the sense that new requests are checked and inserted to the current solution periodically. It is stochastic because it generates samples of future requests or traffic accidents to compare the qualities of the current solution and a candidate solution. The experimental results of Schilde et al. (2011, 2014) show that, for the considered dynamic patient DARPs, the DS VNS algorithm outperforms the MSA of Bent and Van Hentenryck (2004).

Periodic decision epochs are adopted in 22% of the research on DPDPs. For instance, Zolfagharinia and Haughton (2016) make decisions every 12 h for a DPDP with long-haul FTL transportation. They aim at maximizing an objective function which is defined as the total revenue minus the total actual costs (including travel, waiting, and lateness costs) and minus the product of a coefficient and the deadhead costs (i.e., total empty travel costs from all vehicles' last delivery locations to the depot). As the coefficient value has significant impacts on decisions, Zolfagharinia and Haughton (2016) name their policy as a deadhead coefficient policy. They further enhance this policy with an MSA to utilize stochastic information and make waiting decisions (see Section 5.3). Compared to Zolfagharinia and Haughton (2016) and Bertsimas et al. (2019) make periodic decisions much more frequently (i.e., every 30 s) in a taxi scheduling problem. At each decision epoch, they formulate an MIP whose solution is mainly composed of a subset of arcs from a directed flow graph. To cope with realistically sized instances, they introduce a K-neighborhood method to remove the arcs associated with high costs and propose a backbone algorithm to identify the potentially good arcs that may participate in optimal solutions. Thus, the flow graph is sparsified and near-optimal solutions can be quickly computed by solving the MIP with the sparse flow graph.

In about 57% of DPDPs, decisions are made when a new P&D request arrives. In an arrival-triggered reoptimization framework, Yang et al. (2004) propose an anticipatory policy based on opportunity costs. For each new pickup or delivery location to be inserted into the ongoing routes, they estimate the expected distance from this location to the pickup location of a random future customer. This expected distance is incorporated into their cost function so that the influence of the current decision on the opportunities of serving future requests is anticipated. In Vonolfen and Affenzeller (2016), the arrival of each new request triggers a reoptimization procedure consisting of three steps: first, a constructive heuristic is executed to insert the new request to the best position in the current route plan; second, a tabu search heuristic is executed for a fixed number of iterations to improve the new route plan; third, the waiting time at each location is updated by the intensity-based waiting strategy (see Section 5.3). Before these decisions are made during the online phase, the parameters of the waiting strategy are trained via offline simulations. For a meal P&D problem, Steever et al. (2019) develop an auction-based heuristic to perform arrival-triggered reoptimizations. For each vehicle, the heuristic computes the optimal route including the new request and regards the associated objective value as the vehicle's bid price. Then, the vehicle with the highest bid price wins the auction. Steever et al. (2019) also propose a proactive policy that determines the auction winner according to bid prices and two additional metrics: equity and dispersion. Equity captures the expected distance from each restaurant to the nearest vehicle in the future, while dispersion is measured by the average future distance between each possible pair of vehicles. Test results show that using the two metrics can facilitate the services for future demands.

Some 13% of DPDPs are formulated with "hybrid" (i.e., different types of) decision epochs. Pureza and Laporte (2008) and Tirado et al. (2013), Tirado and Hvattum (2017) perform reoptimizations when a new request arrives or when a vehicle stops at a customer location. Cheung et al. (2008) update route plans when new information regarding request arrival or traffic condition is revealed. They accept or reject each new request immediately, and in case of acceptance, employ a heuristic to insert the new request and refine the route plan. When new travel time data becomes available, the previously accepted requests which cause infeasibility are removed from planned routes and processed as new requests. In Arslan et al. (2019), a decision epoch is triggered when a request or an ad hoc (crowdsourced) driver arrives. The authors define a job as a set of one or multiple requests and formulate the DPDP as a matching problem of jobs and drivers. They propose an efficient recursive algorithm to determine the feasible jobs and associated routes. Duan et al. (2020) propose a partially decentralized dispatching system for an autonomous taxi scheduling problem. They consider each revealed request as either a short-term or a long-term request, according to the gap between the request's earliest departure time and the current time point. Each newly arrived short-term request is buffered for consideration by a central dispatcher, while each new long-term request is temporarily accepted by the nearest taxi that can serve it feasibly. Every two minutes, the central dispatcher collects all short-term requests and reoptimizes the global route plan. Each taxi acts as a decentralized dispatcher and works as follows: (1) respond immediately to the long-term requests that appear nearby, (2) check and adjust the taxi's route planned by the central dispatcher, and (3) send accepted long-term requests to the central dispatcher when they become short-term requests.

Queuing theory and Markov decision process

Swihart and Papastavrou (1999) employ queuing theory to analyze the performances of several heuristic policies for single-vehicle DPDPs. Sheridan et al. (2013) extend the nearest neighbor (NN) policy studied by Swihart and Papastavrou (1999) to a dynamic nearest neighbor (DNN) policy for multi-vehicle single-capacity DPDPs. The

NN policy simply requires each vehicle that has finished a delivery to move to the nearest pickup location, while under the DNN policy, each customer awaiting service is guaranteed to be assigned to the closest vehicle at any time and, if this condition does not hold due to the arrival of a new customer or the completion of the last planned service of a vehicle, reassignments of awaiting customers to empty vehicles are allowed. Simulation results show that the DNN policy outperforms the "first-come, first-served" and NN policies.

Hyytiä et al. (2012) formulate a dynamic DARP as an MDP, in which each vehicle is considered as an M/M/1 queue with known service times (from pickup to delivery). They add each newly arrived request immediately to the optimal vehicle by the cheapest insertion heuristic, where the optimal vehicle is determined by a non-myopic policy derived from the analytical results of the multi-queue system. Sayarshad and Chow (2015) extend (Hyytiä et al., 2012) by incorporating a dynamic pricing strategy (see Section 5.3). Further, Sayarshad and Gao (2018) extend (Sayarshad and Chow, 2015) using a more realistic multi-server queue (M/M/s) system and obtain better results. Sayarshad and Chow (2015), Sayarshad and Gao (2018) also propose a TSPPD (TSP with P&D) algorithm to update route plans. When a new request is assigned to a vehicle, the TSPPD algorithm first computes a TSP tour starting at the vehicle's current location and covering the delivery locations of all the requests assigned to the vehicle; then, it inserts the pickup locations of the assigned requests one by one into the TSP tour while ensuring that all constraints are respected and that the increase in route length is minimized.

Ulmer et al. (2021) present a route-based MDP for a DPDP. According to Ulmer et al. (2020b), an MDP is route-based if its state and decision variables indicate the complete routes of vehicles, while an MDP is not route-based if its states and decisions only specify vehicles' current positions and next locations to visit, respectively. Route-based formulation increases the dimensions of the state and decision spaces, but facilitates the applications of conventional vehicle routing algorithms (e.g., constructive heuristics). In Ulmer et al. (2021), a decision epoch is triggered when a new request arrives or when a pending request has been postponed for a certain period of time. The state of the MDP includes the current time point and all information regarding the planned routes and revealed requests. Combining the postponement (i.e., implicit waiting) and time buffer policies discussed in Section 5.3, Ulmer et al. (2021) propose an anticipatory assignment approach to decide, at each decision epoch, which pending requests should be postponed and which should be assigned. Based on these decisions, the assignments and route updates are performed by an insertion heuristic which aims to minimize the increase in service delay. Recently, Haferkamp and Ehmke (2022) formulate a dynamic DARP as an MDP but solve it with an arrival-triggered reoptimization framework.

Multi-agent system

A multi-agent system (MAS) is a decentralized model in which multiple intelligent agents interact autonomously and pursue their objectives. About 9% of DPDPs are formulated as MASs, and in these problems, it is straightforward to consider the requests and vehicles as independent agents. In the MAS proposed by Máhr et al. (2010), the containers to be transported and the trucks performing transportation services are modeled as agents. Each time a container is revealed, its associated agent organizes a Vickrey auction and collects quotes from truck agents, meanwhile truck agents use a constructive heuristic to compute their marginal costs of transporting the container and submit these costs as their bids. The container agent makes a contract with the winner if the second-best bid is lower than its desired price; otherwise, it reorganizes auctions at random time points until a contract is made or it can no longer be feasibly served. Besides, agents can perform re-auctions and container exchanges randomly to improve the route plan. Comparing the experimental results of their MAS and a centralized reoptimization framework, Máhr et al. (2010) find that the MAS performs better when the service times at customer locations are highly uncertain, but when the arrivals of new containers are the dominant uncertainty, the centralized reoptimization framework leads to better results. In an earlier study, Mes et al. (2007) formulate a similar MAS with Vickrey auctions and show that the MAS performs no worse than two traditional hierarchical heuristics: *local control* and *serial scheduling*. Later, Mes et al. (2010) present an extended MAS which is incorporated with a waiting strategy and a dynamic pricing strategy (see Section 5.3).

The studies discussed above all address dynamic FTL P&D problems, while He et al. (2019) and Los et al. (2020) apply MAS to DPDPs with LTL transportation. For a food ordering and delivery problem, He et al. (2019) propose an MAS including three types of agents: customers, restaurants, and an online platform. Customer agents place orders and provide feedback to the platform agent, and their objective is to optimize food quality and waiting time. The platform agent provides customers with each restaurant's expected food quality and waiting time, which are estimated from historical feedback, and sends customers' orders to restaurants. The platform agent also updates the routes of delivery riders by an insertion heuristic, with the objective of minimizing the waiting times of customers and the travel distances of riders. Restaurant agents send the estimated ready time of each order to the platform and, based on historical records, relocate themselves in order to maximize the number of received orders. Los et al. (2020) propose an MAS for a collaborative transportation problem. In this MAS, interactions between order agents and vehicle agents are organized by auctions. The authors test three information sharing strategies that require vehicles to reveal their full route plans, their locations only, or no information at all, and three bidding strategies under which vehicles' marginal costs are always hidden, always shared, or only shared in some conditions.

Other approaches

Shen et al. (1995) introduce an expert system to assist the decision-making of a vehicle dispatcher in a DPDP. They first collect the decisions of the dispatcher during a simulation and use these data to train a neural network. Then, the trained neural network can provide the dispatcher with suggested decisions in response to newly arrived P&D requests. Sáez et al. (2008) and Cortés et al. (2009) propose a hybrid adaptive predictive control (HAPC) framework based on the modern control theory. This framework formalizes the DPDP as a dynamic system consisting of a set of state, input, and output variables that are related by discrete time-domain functions. With probable future requests taken into account, the objective function of the HAPC minimizes the total waiting and ride times of passengers over a rolling prediction horizon. To solve the non-linear MIP inherent in the HAPC framework efficiently, Sáez et al. (2008) and Cortés et al. (2009) develop a genetic algorithm and a particle swarm optimization algorithm, respectively. In a later work, Muñoz-Carpintero et al. (2015) propose several generic evolutionary algorithms to solve the HAPC.

6. Same-day delivery problems

The SDDP is a new DVRPRR variant with all customers requesting deliveries of goods from a single depot to their locations. More than 90% of the studies on the SDDP are published within the past 5 years. A full classification of the existing SDDP papers is provided in our online supplementary materials. Table 8 summarizes the characteristics of the SDDPs studied in the literature.

6.1. Applications

The rapid growth of e-commerce gave rise to the increasing research interest in the SDDP in recent years. Compared to the traditional brick-and-mortar stores, the greatest disadvantage of e-commerce is its lack of "instant gratification" (Ulmer and Streng, 2019). To narrow down this

Table 8
Classification of the literature on the SDDP — Part 1: Problem characteristics.

Literature	CAT	Context	Strategies							Side constraints				Objective(s)
			REJ	PR	WAI	PDR	DP	CS	UAV	VEH	STW	DDL	CAP	
Azi et al. (2012)	DS	E-grocery	✓	✓	×	×	×	×	×	Multi	Hard	×	×	max(R-C)
Klapp et al. (2018a)	DS	E-commerce	✓	×	✓	×	×	×	×	Single	×	Hard	×	min(C+U)
Klapp et al. (2018b)	DS	E-commerce	✓	×	✓	×	×	×	×	Single	×	Hard	×	min(C+U)
Ulmer and Thomas (2018)	DS	E-commerce	✓	✓	×	×	×	×	✓	Multi	Hard	Hard	×	max S
Ulmer and Streng (2019)	DS	E-commerce	×	–	✓	×	×	×	×	Multi	×	×	Hard	min W
Ulmer et al. (2019b)	DS	E-commerce	✓	×	×	✓	×	×	×	Single	×	Hard	×	max S
van Heeswijk et al. (2019)	DS	UCC	×	–	✓	×	×	×	×	∞	Hard	×	Hard	min(C+V+H)
Voccia et al. (2019)	DS	E-commerce	✓	×	✓	×	×	×	×	Multi	Hard	Hard	×	max S
Dayarian and Savelsbergh (2020)	DS	E-commerce	×	–	✓	×	×	✓	×	Dyn	Soft	×	×	min L, min C
Dayarian et al. (2020)	DD	E-commerce	✓	×	✓	×	×	×	✓	Single	Hard	×	×	max S, min C
Klapp et al. (2020)	DS	E-commerce	✓	✓	✓	×	×	×	×	Single	×	×	×	min(C+U)
Ulmer (2020a)	DS	E-commerce	(✓)	✓	×	×	✓	×	×	Multi	Hard	Hard	×	max R
Chen et al. (2022)	DS	E-commerce	✓	✓	×	×	×	×	✓	Multi	Hard	Hard	×	max S

Acronyms

CAT	Category	REJ	Rejection	PR	Prompt reply	WAI	Waiting
PDR	Preemptive depot return	DP	Dynamic pricing	CS	Crowdsourcing	UAV	Unmanned aerial vehicle (drone)
VEH	Number of vehicles	STW	Service time window	DDL	Depot deadline	CAP	Vehicle capacity
UCC	Urban consolidation center						

Objectives

R	Revenue of served requests	C	Travel cost/time/distance	U	Number/penalty of unserved requests
S	Number of served requests	W	Waiting time of customers	V	Number/cost of vehicles used
H	Handling cost (incurred each time a vehicle visits a customer)			L	Penalty of lateness

gap between online and offline shopping, many e-commerce platforms, such as Amazon, Instacart, Walmart, and Google, now offer same-day delivery services (Klapp et al., 2020). On the one hand, same-day delivery provides online shoppers with "near-instant gratification" and hence is a powerful tool for improving sales; on the other hand, it poses a considerable challenge to the logistics service providers. Consequently, many studies have been performed to improve the efficiency and cost-effectiveness of same-day delivery services.

Azi et al. (2012) present to the best of our knowledge the first study on the SDDP. The problem studied by Azi et al. (2012) is inspired by e-groceries where customers request perishable goods to be delivered within specific time windows on the same day. Later, Klapp et al. (2018a,b, 2020) consider SDDPs as dynamic dispatch wave problems. They define a series of equidistant time points (i.e., waves) as potential dispatch times of a single vehicle, and optimize the dispatch decisions to serve the dynamically revealed requests. Ulmer and Streng (2019) optimize the same-day delivery services performed by autonomous vehicles which can only travel directly between the depot and a parcel locker. Ulmer (2020a) studies an SDDP with dynamic pricing, where customers can choose delivery deadlines (the later, the cheaper) according to their willingness-to-pay. Dayarian and Savelsbergh (2020) consider a crowdsourced SDDP and exploit the benefits of employing in-store customers to deliver online orders. As the UAV technology advances, an increasing number of papers focus on the joint same-day delivery of drones and conventional vehicles Ulmer and Thomas (2018), Chen et al. (2022), and Dayarian et al. (2020) introduce a drone resupply system in which trucks are used to deliver goods to customers and drones are used to resupply the delivery trucks. Besides e-commerce applications, the SDDP considered by van Heeswijk et al. (2019) deals with freight deliveries from an urban consolidation center to multiple retailers with dynamic orders.

6.2. Comparison of problem characteristics with the VRPDSR

In the VRPDSR, a new service request can be committed to a vehicle that is currently serving a customer or moving from one location to another, as long as the time and capacity constraints are not violated. This is impossible in the SDDP since goods must be loaded onto vehicles at the depot before being delivered to customers. Given this fact, once a vehicle is dispatched from the depot, the customers it can serve during the current delivery route cannot be changed. As a result, the diversion strategies are not applicable to the SDDP. The delivery requests also bring changes to the waiting strategies: vehicles in the VRPDSR may be required to wait at the depot, customer locations, or strategic waiting points, whereas in the SDDP, the depot is the only location where waiting is likely to be beneficial.

A new strategy that can be applied to the SDDP is the *preemptive depot return*. That is, a vehicle can return to the depot for replenishment before the goods already loaded on board are all delivered. Moreover, motivated by the current practice of online retailers, some SDDPs are incorporated with dynamic pricing, crowdsourcing, and/or drone deliveries. As for the side constraints, the physical capacity of vehicles is usually not a constraint because the orders in SDDPs are mostly small parcels. However, the capacity of drones is usually highly limited, and a "×" in the "CAP" column of Table 8 indicates that only the traditional vehicles (e.g., trucks) are assumed to have infinite capacity. Besides, a "Dyn" in the "VEH" column means that the number of available vehicles changes dynamically due to the random arrivals of crowdsourced drivers.

6.3. Comparison of problem characteristics with the DPDP

The SDDP can be seen as a special case of the DPDP where all customers' pickup locations are at the same depot but their delivery locations are random. With respect to the strategies, the SDDP differs from the DPDP mainly in that diversions are not applicable and that waiting is only beneficial at the depot. Moreover, transshipment and preemptive depot return are two unique strategies for the DPDP and the SDDP, respectively. Regarding the side constraints, many DPDPs are subject to unit vehicle capacity constraints, while in SDDPs, vehicle capacity is not constraining or can accommodate multiple orders. However, the assumption of direct dispatch adopted by Ulmer and Streng (2019) is similar to a unit vehicle capacity constraint because it does not allow goods with different destinations to be loaded simultaneously on a vehicle. In Ulmer and Streng (2019), every dispatched vehicle can carry multiple customers' orders, but it must deliver all loaded orders to only one parcel locker and then return to the depot immediately.

6.4. Decision strategies

Rejections

Rejections of infeasible (and unprofitable, if stochastic information is used) delivery requests are considered in 9 out of the 13 SDDPs. Klapp et al. (2018a,b) determine at fixed time points whether to dispatch a delivery vehicle and which subset of the revealed requests to serve. They assume that the revealed but unserved requests are not permanently rejected until the end of the day, and hence no prompt responses are sent to customers. Voccia et al. (2019) and Dayarian et al. (2020) assume that requests can be served by either self-owned vehicles or a third party. They outsource a pending request when it can no longer be feasibly served due to its time window. Their outsourcing decisions are equivalent to rejections because the routes or costs of the third party are not explicitly modeled.

The other rejection strategies for SDDPs guarantee immediate replies to new requests. To evaluate the profitability of accepting a new request, Azi et al. (2012) construct and maintain multiple solutions, each initially covering the dummy requests from an independently generated sample scenario. A new request is accepted if inserting it into all solutions increases of the overall solution quality. The authors demonstrate that their acceptance rule with anticipation of future requests leads to higher net profits than a myopic acceptance rule. Klapp et al. (2020) develop a proactive acceptance rule as well and show that it increases the overall request acceptance rate compared to myopic rules. They also find that making immediate decisions on new requests results in slightly higher total costs than a policy that postpones the rejections to the end of the day, like those adopted in Klapp et al. (2018a,b). Ulmer (2020a) does not explicitly reject requests, but offers each customer a set of delivery deadlines that are differently priced, and lets the customer to decide whether to accept one of the offered options.

Waiting

The decision maker in an SDDP is faced with the trade-off between early dispatch and order consolidation. Dispatching a vehicle as early as possible reduces the waiting times of the revealed customers, but the opportunities of consolidating future requests are lost and the transportation costs may significantly increase. Eight out of the 13 SDDP papers deal with this trade-off by explicitly incorporating waiting decisions into mathematical models. In Klapp et al. (2018a,b, 2020), Ulmer and Streng (2019), van Heeswijk et al. (2019), and Dayarian and Savelsbergh (2020), the authors define a sequence of equidistant time points, sometimes called *dispatch waves*, as their decision epochs and the available times for dispatching vehicles. At any wave, an idle vehicle at the depot can be dispatched to deliver the pending orders or wait until the next wave for more orders to arrive. Voccia et al. (2019) make waiting decisions when a vehicle returns to the depot or finishes a waiting period. With consideration of service time windows, they prove that delaying the departure of a vehicle to the latest feasible departure time does not reduce the number of delivery requests that can be served by the vehicle. To solve the models with waiting decisions, Ulmer and Streng (2019) and van Heeswijk et al. (2019) use policy function approximation (PFA) and value function approximation (VFA) policies, respectively, while Klapp et al. (2018a,b, 2020) jointly optimize the dispatching and waiting decisions by solving a sequence of stochastic integer programs, and Voccia et al. (2019) and Dayarian and Savelsbergh (2020) identify the beneficial moments for issuing waiting decisions based on sample scenarios (see Section 6.6 for more detailed discussions).

Although allowing waiting does not provide significant performance improvements in some applications (Ulmer and Thomas, 2018; Ulmer et al., 2019b) and increases computational complexity, the extensive numerical experiments performed by Ulmer and Streng (2019), Voccia et al. (2019), and Dayarian and Savelsbergh (2020) show that appropriate waiting strategies can increase the number of served requests and reduce service lateness and total costs. Furthermore, Ulmer and Streng (2019) and Voccia et al. (2019) find that the maximum-consolidation strategies, which only dispatch full vehicles or always require vehicles to wait until the latest feasible departure times, do not perform as well as the strategies trading off early dispatch and consolidation.

Dynamic pricing

Motivated by the current practice of e-commerce retailers, Ulmer (2020a) studies an SDDP in which customers can select their delivery deadlines from multiple options ranging from one hour to four hours. The resources required to serve each request depend not only on the customer's location and the ongoing vehicle routes but also on the deadline chosen by the customer. Hence, Ulmer (2020a) proposes an anticipatory dynamic pricing strategy to incentivize customers to select the deadline options that are efficient and profitable to fulfill. Specifically, each time a customer places a new request, the service provider immediately computes the price of each deadline option and sends the prices to the customer. The customer then selects the most suitable option according to his/her willingness-to-pay. Given a request and a deadline option, the price is set to the maximum of the opportunity cost and the budget cost: the former captures the decrease in the value function if the request will be served within the given deadline, while the latter is defined as the product of the willingness-to-pay and a coefficient, whose optimal value is tuned during an offline policy search algorithm (similar to PFA). In case all deadline options are more expensive than a customer's willingness-to-pay, the customer rejects all options and turns to next-day delivery. The experimental results of Ulmer (2020a) show that incorporating dynamic pricing into dynamic routing leads to significant increases in the total revenue and the number of customers served the same day.

Preemptive depot returns

A common assumption in most SDDPs is that the route of a vehicle is fixed once the vehicle leaves the depot. Under this assumption, an en-route vehicle cannot return to the depot before the loaded orders are all delivered. The concept of preemptive depot return originates from the literature on the DVRPRD, in which vehicles may return to the depot to replenish capacity before the loaded goods are totally consumed and a "route failure" occurs (e.g., Goodson et al., 2016). In an SDDP, it may be beneficial to let a non-empty vehicle return preemptively when the vehicle is close to the depot and a new customer appears near the vehicle's planned route. Comparing the illustrative examples in Figs. 1(c) and 5, we can observe that enabling preemptive depot return leads to a shorter travel distance.

Among the existing SDDP papers, only (Ulmer et al., 2019b) study the application of preemptive depot returns in a single-vehicle SDDP. In Ulmer et al. (2019b), the vehicle's route plan is composed of an ongoing route, a first depot return (for replenishment), a planned route, and a second depot return. At each decision epoch, the route plan is updated in three steps: (1) remove the first depot return, (2) insert the newly accepted orders, and (3) insert a new depot return between the vehicle's current location and the delivery location of the first order that has not been loaded. Thus, the vehicle is allowed to return to the depot before the loaded orders are all delivered. In their computational experiments, Ulmer et al. (2019b) demonstrate that enabling preemptive depot returns increases the number of served requests. However, for a multi-vehicle SDDP with service deadlines, Ulmer (2020a) shows that preemptive returns do not bring significant benefits.

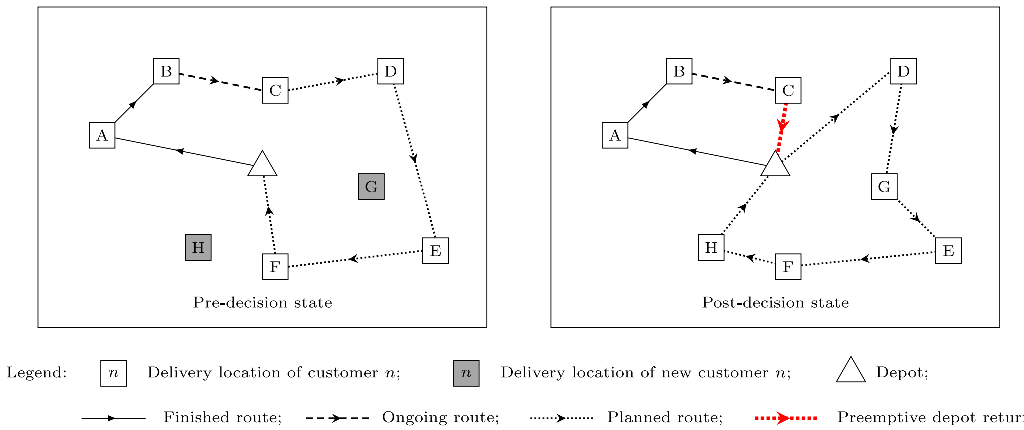

Fig. 5. An illustrative example of preemptive depot return in an SDDP.

Crowdsourcing

Dayarian and Savelsbergh (2020) study an SDDP with a special form of crowdsourcing: a single store employs in-store customers and self-owned vehicles to deliver online orders. This SDDP is similar to the crowdsourced DPDP studied by Arslan et al. (2019) in that the arrivals of online orders and crowdsourced drivers (in-store customers) are both dynamic and uncertain. Dayarian and Savelsbergh (2020) assume that each in-store customer is associated with a home location, a delivery capacity, a minimum time between announcing his/her availability and becoming ready to make a delivery, and a maximum waiting time in the store. To control the inconvenience caused by crowdsourced deliveries, the store confines the service of each in-store customer within a coverage area and pays compensation to each in-store customer. The coverage area is an ellipse region within which delivering an online order increases the in-store customer's travel time by no more than a maximum detour time, and the compensation consists of a fixed cost and an extra cost proportional to the detour time. The computational study of Dayarian and Savelsbergh (2020) reveals that employing in-store customers as crowdsourced deliverers is beneficial in terms of reducing service lateness, especially in the situations where the store is constrained by a small self-owned fleet, tight service deadlines, and limited information on the future. The experimental results also show that considering the stochastic information on the arrivals of online orders and in-store customers improves the solution quality significantly.

Unmanned aerial vehicles (drones)

Ulmer and Thomas (2018) and Chen et al. (2022) study the usage of a heterogeneous fleet consisting of vehicles and drones in same-day delivery. In contrast to the DPDP addressed by Liu (2019) where all services are performed by drones, the SDDPs studied by Ulmer and Thomas (2018) and Chen et al. (2022) require balancing the utilization of vehicles and drones and exploiting their different advantages: vehicles are slower but have a much larger capacity, while drones are faster but their payload is highly limited (a drone can serve only one customer in each tour). Ulmer and Thomas (2018) suggest assigning the customers that are far from the depot to drones and using vehicles to serve the customers close to the depot. They tune a threshold in offline simulations to determine whether a customer is far enough so that serving it with a drone is beneficial. In Chen et al. (2022), the values of assigning customers to vehicles and drones are learned in a neural network.

Dayarian et al. (2020) introduce a truck-drone delivery system in which drones are used to resupply delivery trucks rather than to deliver goods directly to customers. While trucks are executing their routes, drones are dispatched from the depot to transport newly arrived orders to trucks. Focusing on a simplified case with a single truck and single drone, Dayarian et al. (2020) investigate three resupply strategies: (1) in the truck-only strategy, the drone is not used at all and the truck must return to the depot to load new orders after finishing a delivery route; (2) in the restricted resupply strategy, the truck is resupplied by the drone at one of a set of predetermined meeting locations after all loaded goods are delivered; (3) in the flexible resupply strategy, the truck and the drone can be directed to a meeting location for resupply at any time. The experimental results show that enabling drone resupplies facilitates the services to the customers far from the depot and significantly increases the percentage of orders delivered on time. Moreover, the flexible resupply strategy allows more orders to be feasibly fulfilled than the restricted resupply strategy but results in more drone dispatches and higher transportation costs.

6.5. Problem instances

Table 9 compares the SDDP instances solved in the literature. More than 80% of SDDP instances are adapted from SVRP benchmarks or arbitrarily generated on synthetic graphs. It is worth noting that the instances of Ulmer and Thomas (2018), Ulmer and Streng (2019), involving up to 800 and 1000 dynamic requests, respectively, are significantly larger than the other SDDP instances. The difficulties in solving these large instances are alleviated by the assumption of "*direct delivery*". In Ulmer and Thomas (2018), no routing decision needs to be made for the requests assigned to drones because each drone can carry only one order at a time and has to serve each assigned request in an individual round trip between the depot and the customer's location. Similarly, Ulmer and Streng (2019) assume that each autonomous delivery vehicle can visit only one parcel locker in each trip, thus avoiding routing subproblems.

6.6. Solution approaches

Table 10 summarizes the solution approaches that are developed to solve SDDPs. Most SDDPs are modeled as MDPs, and only three papers adopt SVRP reoptimization frameworks. Moreover, almost all SDDP studies incorporate stochastic information and highlight its value in improving the solution quality. The anticipations on future delivery requests can be used to develop non-myopic acceptance rules and waiting strategies, as well as to guide the decisions on routing, preemptive depot returns, and dynamic pricing. Dayarian and Savelsbergh (2020) consider the probabilistic information of both delivery requests and crowdsourced deliverers to improve same-day delivery services. Most MDPs for SDDPs are solved by approximating value functions, policy functions, or Q-values, which return the expected rewards-to-go of given states, policies, or state-decision pairs, respectively, while some authors transform MDPs to SVRPs and reoptimize them with sample requests (Klapp et al., 2018a, 2020; Voccia et al., 2019). The only DD SDDP is studied by Dayarian et al. (2020), in which the authors use a reoptimization framework without sample requests to optimize the strategies of using a drone to resupply a delivery truck.

Table 9
Classification of the literature on the SDDP — Part 2: Problem instances.

Literature	Avail.	Graph	Vehicle(s)	Drone(s)	Requests	DoD (%)	Horizon (h)	Source(s)
Azi et al. (2012)	×	Synthetic	{3, 5}	0	{72, 108, 144}	100	3	Arbitrary
Klapp et al. (2018a)	✓	Manhattan	1	0	{30, 40}[a]	{16.7, …, 87.5}	N/S[b]	Arbitrary
Klapp et al. (2018b)	×	1-D line	1	0	≈{2, …, 87}[a]	≈{54, …, 98}	N/S[b]	Arbitrary
Ulmer and Thomas (2018)	×	Euclidean	{1, …, 20}	{1, …, 5}	{300, …, 800}	100	8	Arbitrary, realistic
Ulmer and Streng (2019)	×	Road (13)	{4, 6, 8, 10}	0	{100, 200, …, 1000}	100	8	Realistic
Ulmer et al. (2019b)	×	Euclidean	1	0	{30, …, 100}	{25, 50, 75}	8	Arbitrary
van Heeswijk et al. (2019)	×	Multiple[c]	∞	0	{150, 400}	{0, 50, 100}	N/S[b]	Solomon, realistic
Voccia et al. (2019)	✓	Manhattan	{1, …, 13}	0	{48, 96, 144, 192}	100	8	Solomon
Dayarian and Savelsbergh (2020)	×	Euclidean	{7, 8, 9, 10}[d]	0	N/S[e]	100	8	Arbitrary
Dayarian et al. (2020)	×	Euclidean	1	1	≈{216, 432}	100	8	Arbitrary
Klapp et al. (2020)	×	Manhattan	1	0	≈{10, …, 115}[a]	≈{62.5, …, 100}	N/S[b]	Arbitrary
Ulmer (2020a)	×	Euclidean	{1, 2, 3}	0	{60, 120, 180}	100	7	Arbitrary
Chen et al. (2022)	×	Euclidean	{2, 3, 4}	{5, 10, 15}	500	100	7	Arbitrary

[a]Locations of all potential customers are known in advance.
[b]Time units are not specified.
[c]Manhattan-style planes (taxicab geometry) and a realistic road network with 11 locations.
[d]This is the size of the company-owned fleet, excluding the crowdsourced drivers.
[e]The authors only specify the ratio between the arrival rates of online orders and crowdsourced drivers.

Table 10
Classification of the literature on the SDDP — Part 3: Solution approaches.

Literature	Formulation			Algorithms	
	Framework/Model	Decision epochs	Anticipation	Offline	Online
Azi et al. (2012)	Reopt-SVRPs	Arrivals & departures	Samples	CH+ALNS	
Klapp et al. (2018a)	MDP	Periodic	Samples	B&C+local search	
Klapp et al. (2018b)	MDP	Periodic	Value function	DP/ALP	
Ulmer and Thomas (2018)	MDP	Request arrivals	Policy function	PFA+CH	CH+rollout
Ulmer and Streng (2019)	MDP	Periodic	Policy function	PFA	–
Ulmer et al. (2019b)	MDP	Vehicle stops	Value function	nVFA+CH	CH
van Heeswijk et al. (2019)	MDP	Periodic	Value function	pVFA+CPLEX	CPLEX
Voccia et al. (2019)	MDP	Hybrid	Samples	–	SSP+VNS
Dayarian and Savelsbergh (2020)	Reopt-SVRPs	Periodic & returns	Samples	–	SSP+Tabu search
Dayarian et al. (2020)	Reopt-SVRPs	Drone returns	–	–	Drone resupply heuristics
Klapp et al. (2020)	MDP	Arrivals & periodic	Samples	B&C+local search	
Ulmer (2020a)	MDP	Request arrivals	Value function	mVFA+CH+PFA	CH
Chen et al. (2022)	MDP	Request arrivals	Q-values	DQL+NN+CH	CH

Acronyms

Reopt-SVRPs	Reoptimization of static VRPs	MDP	Markov decision process	CH	Constructive heuristic
ALNS	Adaptive large neighborhood search	B&C	Branch-and-cut	DP	Dynamic programming
ALP	Approximate linear programming	PFA	Policy function approximation	SSP	Sample-scenario planning
nVFA	Non-parametric value function approximation			VNS	Variable neighborhood search
pVFA	Parametric value function approximation			DQL	Deep Q-learning
mVFA	Meso-parametric value function approximation			NN	Neural network

Reoptimization of SVRPs

Azi et al. (2012) propose a reoptimization framework with decision epochs triggered by two types of events: arrivals of new requests and departures of vehicles from the depot. Before the service period starts, multiple scenarios of sample requests are generated and an initial solution is constructed for each scenario. When an event occurs, the obsolete part of each solution is removed and the rest part of each solution is updated as follows. First, each newly arrived request is tentatively added to each solution by an insertion heuristic and, if feasible, leads to an increase or a decrease in the solution's quality (i.e., revenue minus travel distance). The new request is accepted only if it increases the sum of all solutions' qualities; otherwise, it is rejected immediately. Moreover, each time a new request is accepted or a vehicle leaves the depot, as well as at the beginning of the service period, all solutions are improved by an adaptive large neighborhood search algorithm for a certain number of iterations.

Dayarian and Savelsbergh (2020) propose a reoptimization framework in which decisions are made at equidistant time points and each time a vehicle returns to the depot. Their reoptimization procedure at each decision epoch consists of two steps. First, a tabu search metaheuristic is employed to construct routes to deliver all pending orders with the vehicles and in-store customers (crowdsourced deliverers) currently available at the store. Second, the profitability of postponing the semi- and non-urgent orders, whose latest departure times are later than the next decision epoch, are evaluated with multiple scenarios of sample requests. For each order, the cost and lateness of delivering it at the current decision epoch are compared to the expected cost and lateness of later delivery. If the former is greater than or equal to the latter, the considered order is postponed and removed from the current route plan.

In the truck-drone delivery system introduced by Dayarian et al. (2020), each time the drone returns to the depot, the decision maker chooses the optimal resupply option that allows the largest subset of pending orders to be delivered on time. For the restricted resupply strategy under which the truck is resupplied only after having delivered all loaded goods, each feasible resupply option is associated with a specific meeting location, while for the flexible resupply strategy which allows resupplies to take place at any time, each resupply option specifies a meeting location and the last loaded order that the truck delivers before resupply. For both strategies, the truck's new route in each resupply option is obtained by solving a TSP with time windows using a two-phase heuristic: in the first phase, a minimal-cost route that covers all pending orders and leads to the least service delay is constructed; in the second phase, the smallest subset of orders is removed from the route such that all remaining orders are delivered on time.

Markov decision process

About 77% of SDDPs are formulated as MDPs, but four of them are in fact decomposed into sequential SVRPs to be repeatedly solved (Klapp et al., 2018a,b, 2020; Voccia et al., 2019). Klapp et al. (2018b) study a single-vehicle SDDP where customers are located on a one-dimensional line. In their settings, all customer locations are known in advance but the time points at which customers place requests are probabilistic. The authors define a sequence of dispatch waves (i.e., discrete time points) as decision epochs. At each wave, they compute an a priori solution for the rest of the service period by dynamic programming or approximate linear programming. Klapp et al. (2018a) extend the one-dimensional SDDP of Klapp et al. (2018b) by considering a two-dimensional geographical distribution of customers. Accordingly, the static a priori problems to be solved change from relatively simple dynamic programs or linear programs to TSPs. The authors employ a branch-and-cut algorithm, which is complemented by local search heuristics, as the solution approach at each wave. For the same SDDP, Klapp et al. (2020) reoptimize a priori problems not only at dispatch waves but also when any new request arrives. To guarantee immediate responses to new customers, they develop a local-search-based heuristic to make acceptance/rejection decisions efficiently. The computational experiments of Klapp et al. (2018a,b, 2020) confirm that their policies outperform a static policy which only solves an a priori problem at the beginning and uses simple recourse rules (e.g., order skipping) during the service period.

Voccia et al. (2019) formulate an MDP for a multi-vehicle SDDP with service time windows. They solve the MDP in an event-triggered reoptimization framework. When a new request arrives: if there is an idle vehicle at the depot, the new request is directly assigned to the idle vehicle; otherwise, the new request is buffered to be processed at the next decision epoch, which is triggered by a vehicle returning to the depot or finishing a waiting period. Inspired by the MSA proposed by Bent and Van Hentenryck (2004) and Voccia et al. (2019) develop a sample-scenario planning (SSP) approach to solve the SVRP at each decision epoch. The SSP generates multiple sample scenarios of future requests and computes a route plan for each scenario by VNS. It then determines the distinguished plan by a newly designed consensus function.

To deal with the intractably large scales of the MDPs for SDDPs, many authors employ ADP approaches to compute decision policies. Ulmer et al. (2019b) adapt the nVFA proposed by Ulmer et al. (2018a) to solve a single-vehicle SDDP with preemptive depot returns. They aggregate the post-decision state space to a three-dimensional vector space based on three features: the current time point, the planned time of the first depot return, and the free time budget (depot deadline minus the planned time of the second depot return). They also restrict the action space by using an insertion heuristic to make routing decisions. Through extensive offline simulations, the value function mapped to the vector space is approximated and stored in a dynamic lookup table. A different approach for approximating value functions, called parametric value function approximation (pVFA), is employed by van Heeswijk et al. (2019) to address a multi-vehicle SDDP. After aggregating the post-decision states to low-dimensional feature vectors, van Heeswijk et al. (2019) formulate a linear function of the feature vectors to approximate the value function. They perform offline simulations to tune the coefficients of the linear function, with the aim of minimizing the approximation error. Comparing the two VFA approaches, the linear functional structure of pVFA is easier to implement than the lookup table of nVFA, and pVFA provides more reliable approximations for all parts of the value function; however, nVFA better captures the complex structure of the value function and generally achieves more accurate approximations (Ulmer and Thomas, 2020). To exploit the advantages of nVFA and pVFA, Ulmer (2020a) draws on a meso-parametric VFA (mVFA) which combines nVFA and pVFA in a linear function. He selects three features to aggregate the state space: the current time point, the free time budget, and the flexibility which measures the average gap between the planned delivery times and the deadlines of all pending orders. Experimental results show that the mVFA produces accurate and efficient approximations and leads to better results than nVFA and pVFA.

PFA is another class of ADP approaches that can effectively solve the curses of dimensionality. PFA usually relies on an analytical function to select an action for a given state, and is suitable for the cases where the structure of a good decision rule can be intuitively designed (Powell et al., 2012). For an SDDP with both vehicles and drones, Ulmer and Thomas (2018) base their PFA on the observation that it is beneficial to serve remote customers by drones and nearby customers by vehicles. They define a threshold to divide customers into two groups: if the vehicle travel time from the depot to a customer is greater than the threshold, the customer is served by a drone as long as it is feasible; otherwise, the customer is preferably assigned to a vehicle. Later, Ulmer and Streng (2019) design a PFA policy with a threshold for trading off early dispatches and consolidations. In Ulmer and Streng (2019), a fleet of autonomous vehicles delivers goods from a central depot to a set of parcel lockers. Each dispatched vehicle only visits one locker and then returns to the depot immediately. Each locker has a finite capacity and each delivered order stays in the locker for a stochastic period of time until being picked up by the customer. The authors observe that vehicle dispatch is efficient if the number of orders to be delivered is sufficiently large. Hence, at each decision epoch, they select the parcel locker to which the largest number of orders can be shipped without violating the capacity constraint. If this number is smaller than the threshold, all idle vehicles are required to wait at the depot; otherwise, a vehicle is dispatched to the selected locker. In the PFA policies of Ulmer and Thomas (2018) and Ulmer and Streng (2019), the optimal threshold value is determined by simulating the problem offline with a set of candidate values (e.g., an arbitrary interval discretized by an arbitrary step), and no substantial computation is required during the online phase.

Recently, Chen et al. (2022) apply deep Q-learning to dynamic vehicle routing. For an SDDP with vehicles and drones, they use neural networks to decide whether a new request should be accepted and whether an accepted request should be assigned to a vehicle or a drone. Based on analytical results, they define the inputs of neural networks as a time feature indicating the current time, a set of fleet features indicating the finish times of ongoing and planned routes, and two action features indicating the distance between the new request and the depot as well as the additional travel time if the new request is assigned to a vehicle. Given a state, the neural networks output the Q-values of all feasible decisions, based on which the best decision is selected. The coefficients of the neural networks are trained in offline simulations. Thus, immediate decisions can be made in real time. Experimental results show that the deep Q-learning policy outperforms the PFA policy of Ulmer and Thomas (2018), especially when resources are limited.

7. Dynamic multi-period vehicle routing problems

The last DVRPRR variant reviewed in this paper is the DMPVRP, which is not characterized by a specific request type but by the existence of multiple consecutive planning periods (days). Table 11 summarizes the main problem characteristics of the DMPVRPs studied in

Table 11
Classification of the literature on the DMPVRP — Part 1: Problem characteristics.

Literature	CAT	Request type	Context	Strategies		Side constraints				Objective(s)
				RT	Others	VEH	STW	VTW	CAP	
Savelsbergh and Sol (1998)	DD	P&D	Courier	✓	–	Multi	Hard	Hard	Hard	min V, min C
Angelelli et al. (2007a)	DD	Service	N/S	×	–	Single	Hard	×	×	min C
Angelelli et al. (2007b)	DD	Service	N/S	×	–	Single	Hard	×	×	min C
Zhong et al. (2007)	DS	Delivery	Courier	×	WL	Multi	Same-day	Hard	×	min(V+C+T)
Andreatta and Lulli (2008)	DS	Delivery	Blood	×	–	Single	Hard	×	×	min C
Angelelli et al. (2009)	DD	Pickup	Courier	✓	OUT & DIV	Multi	Hard	Hard	×	min O, min C
Angelelli et al. (2010)	DD	Pickup	Courier	✓	REJ & DIV	Multi	Hard	Hard	×	min U, min C
Wen et al. (2010)	DD	Delivery	Agriculture	×	–	Multi	×	Hard	Hard	min(C+W−B)
Albareda-Sambola et al. (2014)	DS	Service	N/S	×	–	Multi	Hard	×	Hard	max(R−C)
Cordeau et al. (2015)	DD	Delivery	Auto-carrier	×	–	Multi	Soft	Hard	Hard	min(C+V+H+L)
Chen et al. (2016)	DD	Service	Technician	×	WL	Multi	Same-day	×	×	min M
Ulmer et al. (2018b)	DS	Service	Multiple	✓	–	Single	Hard	Hard	×	max S
Ulmer et al. (2020a)	DS	Delivery	Retail	×	WL	Multi	Same-day	Hard	×	min(C+T+F)
Ulmer (2020b)	DS	Service	Multiple	✓	–	Single	Hard	Hard	×	max S
Laganà et al. (2021)	DD	Delivery	Courier	×	–	Multi	Hard	×	Hard	min C
Subramanyam et al. (2021)	DS	Delivery	Gas	×	OUT	Multi	Hard	×	Hard	min O, min C

Acronyms				Objectives			
CAT	Category	RT	Real-time	V	Number/cost of vehicles used	C	Travel cost/time/distance
VEH	Number of vehicles	STW	Service time window	T	Total service time	O	Outsourcing cost
VTW	Vehicle time window	CAP	Vehicle capacity	U	Number of unserved requests	W	Waiting time of customers
P&D	Pickup and delivery	WL	Workforce learning	B	Balance of daily workload	R	Revenue of served requests
OUT	Outsourcing	DIV	Diversion	H	Handling cost	L	Penalty of lateness
REJ	Rejection			M	Makespan	S	Number of same-day services
				F	Driver–customer familiarity		

the literature (a full classification of DMPVRPs is presented in the supplementary materials). Based on the decision-making frequency, DMPVRPs can be divided into two types. The first type is the DMPVRP without real-time decision-making. In this type of problem, route plans are updated only at the beginning of each day, and no decisions are made during the service period of each day. Accordingly, the newly arrived requests cannot be served or rejected on the same day but have to wait until the next day for the service provider to make a decision. Moreover, diversion and waiting strategies are not applicable. The second type of DMPVRP is incorporated with real-time decision-making. This type of problem is more dynamic because route plans are updated multiple times each day, thus allowing the newly arrived requests to be either serviced on the same day or postponed.

7.1. Applications

Table 11 shows that the DMPVRP may arise in a wide variety of logistical contexts. Andreatta and Lulli (2008) present a blood delivery problem with regular and urgent requests: the former can be postponed for one day, while the latter must be served on the same day. The blood bank needs to determine on each day which regular requests to postpone so as to minimize the total delivery costs. Similarly, Angelelli et al. (2010) study a dynamic courier pickup problem with postponable and unpostponable requests. The DMPVRP formulated by Wen et al. (2010) originates from an agricultural product delivery application, and aims at optimizing travel costs, customers' waiting times, as well as the balance of daily workload over the planning horizon. Cordeau et al. (2015) address a DMPVRP in which a fleet of auto-carriers delivers vehicles in response to dynamic customer requests. The authors maintain a rolling planning horizon consisting of multiple days and take into account the heterogeneity of auto-carriers and soft time windows of customers.

The DMPVRPs studied by Chen et al. (2016) and Ulmer et al. (2020a) are motivated by technician routing and retail delivery, respectively. The two problems both incorporate workforce learning: Chen et al. (2016) assume that the time required by a technician to complete a task depends on the technician's experience in performing the same or similar tasks; Ulmer et al. (2020a) assume that a driver becomes familiar with a customer's location after the first visit and will need less time to complete a second visit. More recently, Subramanyam et al. (2021) study a multi-day gas cylinder delivery problem with the objective of minimizing the usage of outsourced vehicles and the travel costs of self-owned vehicles.

7.2. DMPVRP without real-time decision-making

Nearly 70% of the papers studying DMPVRPs do not consider intra-day real-time decisions. In these papers, it is assumed that all new requests arrive at the beginning of each day (e.g., Angelelli et al., 2007a,b) or that the new requests accumulated during the course of each day are not considered until the beginning of the next day (e.g., Subramanyam et al., 2021). As the route plan is updated on a daily basis, a DMPVRP without real-time decision-making is similar to a VRPDSR, DPDP, or SDDP which is reoptimized periodically. However, in the latter, the service periods of any two consecutive days are independent, whereas those in the former are correlated. A common correlation is that the known requests that are not served during the current day have to be reconsidered on the next day. Taking this correlation into account, the decision maker should determine which requests to serve on the same day and which to postpone, so as to balance the fleet utilization on each day and minimize (or maximize) the total costs (or revenues) over the entire planning horizon. Furthermore, some authors consider different priority classes of requests. In Andreatta and Lulli (2008), a blood center delivers blood to hospitals in response to two types of requests: urgent requests that must be served on the same day and regular requests that should be served within two days. Laganà et al. (2021) classify parcel delivery requests into three dynamic priority classes: urgent, prominent, and unimportant. They assume that the unserved unimportant requests become prominent in two days, that the unserved prominent requests become urgent in one day, and that the urgent requests cannot be postponed.

In some non-real-time DMPVRPs, request postponements are not allowed, that is, all requests are treated as urgent requests. The consecutive days in these problems are correlated by the increase in workforce experience or driver–customer familiarity. Zhong et al. (2007) assume that a driver needs less time to finish delivery tasks if he/she is more familiar with customers' locations. This familiarity can be enhanced if the service area assigned to each driver does not change; however, adjusting the delivery routes on a daily basis improves operational efficiency. Zhong et al. (2007) balance this trade-off in a strategic planning stage, where they identify a set of core areas, each of which is always assigned to a fixed driver, and a flex zone, which includes the locations that are close to the depot and can be reassigned to different drivers. For a technician routing problem, Chen et al. (2016) assume that each request requires a specific skill and that its service time depends on the technician's experience. After serving a request, the technician gains experience and will spend less time on the next request which requires the same skill. The authors take into account the heterogeneity in technicians' learning rates on different skills and aim at minimizing the daily makespan. Ulmer et al. (2020a) address the familiarity between drivers and customers in a retail distribution problem. They assume the driver–customer familiarity to be binary, that is, the familiarity between a driver and a customer is established as long as the driver visits the customer once. They demonstrate that appropriate investments in the development of familiarity is beneficial for the improvement of service performance.

7.3. DMPVRP with real-time decision-making

The DMPVRPs with real-time decision-making can be regarded as multi-period extensions of DPDPs or VRPDSRs. Savelsbergh and Sol (1998) develop a route planning module for a multi-period DPDP. They reoptimize the route plan not only every day at midnight but also every hour during working hours. Angelelli et al. (2009, 2010) address a multi-period VRPDSR. Before the beginning of each day, they compute for each vehicle a route starting and ending at the depot and covering the unpostponable requests only. During the course of the day, they reoptimize the route plan repeatedly with consideration of all requests. In Angelelli et al. (2009), the requests that cannot be feasibly served are outsourced at high costs, while in Angelelli et al. (2010), the infeasible requests are rejected by the service provider immediately. The numerical experiments of Angelelli et al. (2010) reveal that the policy with outsourcing outperforms the policy with immediate acceptances and rejections. En-route diversions are allowed in Angelelli et al. (2009, 2010), but their benefits are not examined. Ulmer et al. (2018b) and Ulmer (2020b) study a single-vehicle multi-period VRPDSR, in which all requests must be served on the same day or the next day by the service provider. The authors make decisions at the beginning of each day's service period and each time the vehicle finishes serving a customer. Their computational studies show that considering both multi-day and intra-day anticipations increases the number of requests that are served on the same day.

7.4. Problem instances

Table 12 compares the DMPVRP instances solved in the literature. Column "Horizon" presents the number of days that are simulated in each paper; for the papers addressing real-time decision-making, this column also presents the length of the service period on each day. Column "Requests/day" compares the (expected) number of new requests per day. We do not compare the DoD in Table 12 for two reasons: first, in the non-real-time DMPVRPs, the subproblem on each day has a DoD of 0 because all requests to be served are known in the beginning; second, in the real-time DMPVRPs, the static requests on each day are in fact the postponed dynamic requests from the previous days, and the number of static requests depends not only on the problem settings but also on the decision policy. Hence, the DoD does not appropriately reflect the dynamism of the DMPVRP. The last column "Source(s)" shows that there is no widely accepted DMPVRP benchmark. Most authors formulate DMPVRP instances arbitrarily, from realistic cases, or based on SVRP instances such as Solomon (1987), VRPLIB (provided by the Operations Research Group of the University of Bologna), Lysgaard et al. (2004), and Bosco et al. (2013), while Ulmer et al. (2018b) and Ulmer (2020b) extend the single-period VRPDSR instances of Ulmer et al. (2018a) to multi-period settings.

7.5. Solution approaches

The solution approaches for the DMPVRP are summarized in Table 13. It can be seen that 11 out of the 16 papers solve DMPVRPs by reoptimizing their static counterparts, while the other 5 papers model DMPVRPs as MDPs. Note that we distinguish between "Daily" and "Periodic" in column "Decision epochs": "Daily" implies that all decisions are made at the beginning of each day (i.e., real-time decision-making is not considered), while "Periodic" means that the time interval between two consecutive decision epochs is shorter than one day (i.e., real-time decision-making is considered).

Nearly half of DMPVRPs are formulated as DS models and the stochastic information in these models are exploited in different ways. In MDP models, Andreatta and Lulli (2008), Ulmer et al. (2018b) and Ulmer (2020b) anticipate the rewards-to-go by the exact or approximate value functions; Chen et al. (2016) propose a myopic solution approach to solve their MDP model; Ulmer et al. (2020a) use a cost function to estimate the values of investing in driver–customer familiarity, and validate that anticipatory development of familiarity reduces operational costs. Among the 11 reoptimization frameworks for DMPVRPs, only three incorporate stochastic information. Zhong et al. (2007) use the expected spatial distribution of customers to strategically optimize the design of core areas and flex zone. Albareda-Sambola et al. (2014) propose an adaptive policy that postpones a pending request if the probability that new requests arise in the vicinity is high. Subramanyam et al. (2021) propose a robust reoptimization framework to tackle the uncertainty without probability distributions. Next, we discuss the solution frameworks for DMPVRPs in detail.

Reoptimization of SVRPs

The DMPVRPs with and without real-time decision-making can both be decomposed into sequential SVRPs. Angelelli et al. (2007a,b) perform analytical studies on a single-vehicle DMPVRP in which routes are reoptimized on a daily basis. They analyze several simple online policies theoretically and evaluate their competitive ratio (i.e., the ratio between the solution quality of the online policies and that of an exact offline algorithm which requires all information to be known in advance). Later, Angelelli et al. (2009, 2010) propose dynamic policies to enable real-time decision-making in a multi-vehicle DMPVRP. On each day, they first compute an initial route plan covering all unpostponable requests, then reoptimize the route plan at equidistant time points to incorporate the dynamically arriving requests. For the cases where rejections of infeasible requests are allowed, they set the reoptimization interval to a sufficiently small value to guarantee immediate responses to new requests. Though no stochastic information is considered, Angelelli et al. (2009, 2010) define a look-ahead period within which the planned routes and costs are reoptimized by the SVRP at each decision epoch. They solve the sequential SVRPs by a VNS metaheuristic and demonstrate that the solution quality improves as the look-ahead period increases from 0 to 2 days. Similar concepts of look-ahead periods, also called rolling horizons, are adopted by Savelsbergh and Sol (1998), Wen et al. (2010), and Cordeau et al. (2015) in DD DMPVRPs.

Some special approaches are proposed for the real-time DS DMPVRPs. Zhong et al. (2007) divide the whole service area into a number of cells and define for each cell a normally distributed random variable representing the number of requests. With these random variables, the authors formulate a chance-constrained programming (CCP) model to plan the core areas (each assigned to a fixed driver) strategically. They convert the CCP model to its deterministic counterpart and solve it

Table 12
Classification of the literature on the DMPVRP — Part 2: Problem instances.

Literature	Avail.	Graph	Horizon (d or d × h)	Vehicle(s)	Requests/day	Source(s)
Savelsbergh and Sol (1998)	×	Road (N/S)	10 × 14	≈100	≈500	Realistic
Angelelli et al. (2007a)	×	1-D line/Euclidean	{2, 3, …}	1	No computational study performed	
Angelelli et al. (2007b)	×	1-D line	{2, 3, …}	1	No computational study performed	
Zhong et al. (2007)	×	Euclidean	30	>1	≈{38, …, 760}	Arbitrary
Andreatta and Lulli (2008)	×	Nodes (6)	∞	1	≈10/3	Arbitrary
Angelelli et al. (2009)	×	Euclidean	10 × 10	3	{100, 200, …, 500}	Solomon
Angelelli et al. (2010)	×	Euclidean	5 × 10	3	{100, 200, …, 500}	Solomon
Wen et al. (2010)	(✓)	Euclidean	{10, 15}	>1	80	Realistic
Albareda-Sambola et al. (2014)	×	Euclidean	{6, 10}	>1	{25/6, …, 86}	Solomon, Wen
Cordeau et al. (2015)	(✓)	Road (N/S)	23	137	≈774	Realistic
Chen et al. (2016)	✓	Euclidean	29	18	{71, …, 150}	VRPLIB
Ulmer et al. (2018b)	×	Euclidean	{3, …, 50}×6	1	{22.5, …, 75}	Ulmer
Ulmer et al. (2020a)	×	Euclidean	120	10	70	Arbitrary
Ulmer (2020b)	×	Euclidean	3 × 6	1	{50, 75}	Ulmer
Laganà et al. (2021)	×	Nodes ({7, …, 22})	7	{3, 4, …, 7}	≈{33, …, 100}	Bosco
Subramanyam et al. (2021)	✓	Euclidean	30	>1	⩽{12.5, 15, 17.5}	Lysgaard

Table 13
Classification of the literature on the DMPVRP — Part 3: Solution approaches.

Literature	Formulation			Algorithms	
	Framework/Model	Decision epochs	Anticipation	Offline	Online
Savelsbergh and Sol (1998)	Reopt-SVRPs	Periodic	–	Branch-and-price	
Angelelli et al. (2007a)	Reopt-SVRPs	Daily	–	Analytical study (competitive analysis)	
Angelelli et al. (2007b)	Reopt-SVRPs	Daily	–	Analytical study (competitive analysis)	
Zhong et al. (2007)	Reopt-SVRPs	Daily	Distribution	Tabu search	CH+DP
Andreatta and Lulli (2008)	MDP	Daily	Value function	DP	–
Angelelli et al. (2009)	Reopt-SVRPs	Periodic	–	Variable neighborhood search	
Angelelli et al. (2010)	Reopt-SVRPs	Periodic	–	Variable neighborhood search	
Wen et al. (2010)	Reopt-SVRPs	Daily	–	Three-phase heuristic[a]	
Albareda-Sambola et al. (2014)	Reopt-SVRPs	Daily	Distribution	Variable neighborhood search	
Cordeau et al. (2015)	Reopt-SVRPs	Daily	–	Rolling horizon iterative local search	
Chen et al. (2016)	MDP	Daily	–	Record-to-record travel heuristic	
Ulmer et al. (2018b)	MDP	Daily & vehicle stops	Value function	nVFA+CH	CH
Ulmer et al. (2020a)	MDP	Daily	Cost function	–	CFA+CH
Ulmer (2020b)	MDP	Daily & vehicle stops	Value function	nVFA+CH	CH+rollout
Laganà et al. (2021)	Reopt-SVRPs	Daily	–	CH+ALNS	
Subramanyam et al. (2021)	Reopt-SVRPs	Daily	Uncertainty sets	Robust optimization+B&C	

Acronyms					
Reopt-SVRPs	Reoptimization of static VRPs	MDP	Markov decision process	CH	Constructive heuristic
DP	Dynamic programming	nVFA	Non-parametric value function approximation		
CFA	Cost function approximation	ALNS	Adaptive large neighborhood search	B&C	Branch-and-cut

[a]Three-phase heuristic: 1, customer selection; 2, VNS; 3, post-optimization.

with a tabu search metaheuristic. Based on the results of strategic planning, they optimize the daily operational routing problem by an insertion heuristic and dynamic programming. Unlike Zhong et al. (2007) and Albareda-Sambola et al. (2014) which rely on the spatio-temporal distribution of customers, Subramanyam et al. (2021) propose a robust optimization framework that does not rely on any probabilistic information. With all customer locations known in advance, the authors define a vector of random binary variables to describe the presence of potential requests in the future and assume that the realizations of random variables are within a finite uncertainty set constructed from historical data. They develop a branch-and-cut algorithm to solve an integer program at the end of each day, thus obtaining a route plan for the next 5 days which is feasible for all possible realizations of random variables. In a computational study, Subramanyam et al. (2021) compare the robust solutions to the nominal solutions which are obtained without considering any potential future requests. The results show that the robust solutions significantly reduce the frequency of vehicle capacity violations and only lead to slight increases in total costs.

Markov decision process

For a DMPVRP with daily decision epochs, Andreatta and Lulli (2008) propose an exact MDP in which the state variables indicate the request status at each customer location. They also propose an aggregate MDP whose state variables only include the number of urgent and regular requests. The aggregate MDP has a much smaller state space than the exact MDP but does not capture all problem details. To exploit the advantages of both models, the authors first solve the aggregate MDP in advance by dynamic programming and obtain the estimated cost-to-go from each aggregate state. Then, on each day, they select the action that minimizes the sum of the exact MDP's cost function and the estimated cost-to-go. Numerical experiments on a 6-node network show that the resulting policy coincides with the optimal policy in most states. Chen et al. (2016) formulate an MDP for a multi-period technician routing problem with experience-based learning. Due to the curses of dimensionality, the authors ignore stochastic information and solve the MDP myopically.

Ulmer et al. (2020a) extend the work of Chen et al. (2016) by considering stochastic information and investments in driver–customer familiarity. They model the DMPVRP as an MDP and propose an anticipatory investment policy based on an ADP approach named cost function approximation (CFA). The basic idea of CFA is to artificially manipulate the cost function to anticipate the cost-to-go. To this end, Ulmer et al. (2020a) introduce a group of additional terms that are subtracted from the cost function, each estimating the potential benefit of investing in the familiarity of a given driver–customer pair on a given day. Such an additional term equals 0 if the driver and customer are already familiar with each other or if the customer's location is not in the familiar zone of the driver; otherwise, it takes a positive value and decreases with the total number of drivers that are familiar with the customer. Unlike VFA, PFA, and rollout approaches, CFA does not update the Bellman equation recursively; it simply finds the best action that minimizes the manipulated cost function at each decision epoch. In a computational study, Ulmer et al. (2020a) validate that the CFA policy improves the overall driver–customer familiarity and reduces operational costs.

Ulmer et al. (2018b) extend the MDP formulation and nVFA approach proposed by Ulmer et al. (2018a) to address a single-vehicle real-time DMPVRP. Recall that Ulmer et al. (2018a) deal with a VRPDSR and aggregate each post-decision state to a two-dimensional vector featured by the current point of time and the free time budget of the current day (see Section 4.4). Ulmer et al. (2018b) add two more features to allow multi-day anticipation in a finite planning horizon: the index of the current day and the free time budget of the next day. They perform offline simulations to obtain an approximate value function, based on which they decide at each decision epoch the subset of new requests to be accepted for same-day services. For each subset, they update the route plans of the current and next days by a cheapest insertion heuristic and compute the two time budgets accordingly. They demonstrate in a computational study that their nVFA policy outperforms a myopic policy and the policy of Angelelli et al. (2009), and that incorporating multi-day anticipation improves the fairness of service and leads to constantly high service performance over all days in the planning horizon. Ulmer (2020b) studies the same DMPVRP as Ulmer et al. (2018b). Similar to the way that Ulmer et al. (2019a) combine an offline nVFA with an online rollout algorithm, Ulmer (2020b) proposes a VFA-based limited horizon rollout algorithm. The latter differs from the former in that the rollout algorithm only simulates a fixed number of decision epochs into the future, rather than simulating the entire horizon. The experimental results of Ulmer (2020b) show that the proposed algorithm achieves a good trade-off between solution quality and efficiency.

8. DVRPs with other sources of dynamism

Besides the four DVRPRR variants reviewed in the previous sections, the VRPs with random demands or travel times have also received considerable attention. In these problems, the locations of customers are deterministic and known in advance, but the actual demands of customers are not known until vehicles arrive, or the travel times in the road network are uncertain. Many of these problems fall into the category of SS VRP, as their solutions are computed a priori, typically by stochastic programming, with recourse actions determining the restocking policies or penalizing the unsatisfied demands and violated time windows. In contrast to these static problems, the DVRPRD and the DVRPRTT involve reoptimization of route plans during route execution. The latest review explicitly covering the DVRPRD and the DVRPRTT is provided by Ritzinger et al. (2016). In this section, we briefly overview the recent research on these two types of dynamic problems.

8.1. DVRP with random demands

The logistical contexts of DVRPRDs are diverse, such as delivering products to hospitals, restaurants, and vending machines, and collecting money, packages, wastes, and recycled materials from banks, homes, and industrial plants (Novoa and Storer, 2009). Except for the DVRPRDs with dynamic requests (e.g., Hvattum et al., 2007; Sarasola et al., 2016), the service provider in a DVRPRD cannot benefit from rejection or waiting strategies, since all known customers must be visited and there are no new customers placing orders during the service period. The DVRPRD usually requires a restocking policy to tackle the route failures. Under the simplest detour-to-depot policy, a vehicle returns to the depot for replenishment only when it fails to serve a customer due to insufficient load or capacity. In order to prevent route failures and reduce costs, most DVRPRD papers exploit the stochastic demand information and develop proactive restocking strategies which allow vehicles to return to the depot before running out of stock or capacity (similar to the preemptive depot return strategy discussed in Section 6.4). Furthermore, Achamrah et al. (2021) study the dynamic inventory routing problem with stochastic demands, which can be regarded as a DVRPRD with inventory control decisions. To mitigate the shortage of products, Achamrah et al. (2021) optimize a transshipment policy that allows products to be transshipped between customers, and a substitution policy that provides substitute products to customers in case the ordered products are insufficient.

Compared to the four DVRPRR variants, the DVRPRD has a lower dynamism but higher stochasticity, since there are no request arrivals and the stochastic demand information is usually available. Achamrah et al. (2021) propose a rolling horizon framework to reoptimize an inventory routing problem with stochastic demands. To solve the sequential SVRPs, the authors develop a genetic algorithm that is accelerated by deep reinforcement learning. Most other DVRPRDs are formulated as MDPs. Goodson et al. (2013, 2016) model the DVRPRD with duration limits as an MDP, and propose a family of a-priori-based and restocking-based rollout policies, in which the routing subproblems are solved by local search or VNS. These rollout policies are then formalized by Goodson et al. (2017) and employed to solve VRPDSRs (e.g., Ulmer et al., 2019a), SDDPs (e.g., Ulmer and Thomas, 2018), and DMPVRPs (e.g., Ulmer, 2020b). In order to address the curses of dimensionality of the MDP for multi-vehicle DVRPRDs, Zhang et al. (2022b) propose a pVFA framework to aggregate the state space, and design the so-called recourse reduction and neighborhood reduction strategies to prune the decision space. Finally, most DVRPRD instances are randomly generated (Hvattum et al., 2007; Novoa and Storer, 2009; Secomandi and Margot, 2009) or derived from existing SVRP benchmarks (Goodson et al., 2013, 2016; Zhang et al., 2022b).

8.2. DVRP with random travel times

The DVRPRTTs studied in the literature are also motivated by different applications, including freight pickup and delivery (Güner et al., 2017; Rifki et al., 2020), dial-a-ride with ride-sharing (Liang et al., 2020), drone-assisted parcel delivery (Liu et al., 2022), etc. Almost all DVRPRTTs incorporate stochastic travel time information and have service time windows associated with customers. Lorini et al. (2011) and Respen et al. (2019) evaluate the diversion opportunities in DVRPRTTs, while Güner et al. (2017) and Vodopivec and Miller-Hooks (2017) develop proactive waiting strategies to exploit the dynamic traffic information. Liu et al. (2022) introduce a flying sidekick TSP with stochastic travel times, in which a pair of truck and drone is employed to deliver parcels: the truck travels in the road network and serves a subset of customers, meanwhile the drone is dispatched from the truck to deliver parcels to the other customers. We recall that random travel times are also considered in some DPDPs (see Table 5), but these problems mostly do not incorporate stochastic information.

Liang et al. (2020) solve their DVRPRTT in a rolling horizon reoptimization framework with equidistant decision epochs, and propose a Lagrangian relaxation algorithm to reoptimize the sequential SVRPs (modeled as integer nonlinear programs). The other DVRPRTTs are mostly modeled as MDPs. To solve these MDPs, Toriello et al. (2014) and Yu and Yang (2019) develop approximate linear programming approaches; Kim et al. (2016) propose rollout-based approaches to avoid curses of dimensionality; Köster et al. (2018) introduce a heuristic policy which aims at minimizing the expected travel time and avoiding the potentially critical areas in the city; Liu et al. (2022) develop a reinforcement-learning approach based on deep Q-network and advance actor-critic algorithms. Most DVRPRTT instances solved in the recent literature are based on real-world road networks (Kim et al., 2016; Güner et al., 2017; Köster et al., 2018; Liang et al., 2020; Liu et al., 2022), while Toriello et al. (2014), Respen et al. (2019), and Yu and Yang (2019) generate instances randomly or based on SVRP benchmarks.

9. Conclusions, insights and future outlook

The research interest in the DVRP significantly increased during the last 40 years. Previous DVRP reviews either only include the DS VRPs or cover all types of dynamism. In this review, we focus on the most common dynamic aspect—the random arrivals of customer requests—and review the existing DVRPRR research works adopting both DD and DS formulations. In total, we discussed 118 research papers published in peer-reviewed academic journals. Given that the type of dynamic customer requests and the length of planning horizon have important effects on decision-making, we classify the DVRPRRs based on these two criteria into four main variants: the VRPDSR (35 papers), the DPDP (54 papers), the SDDP (13 papers), and the DMPVRP (16 papers). For each variant, we review the logistical contexts, applicability of different decision strategies, test instances, mathematical models, and solution techniques. We are convinced that this comprehensive and in-depth review provides researchers and practitioners with important methodological and managerial insights as well as useful recommendations for future research directions.

9.1. Main insights

The insights gained from this literature review are summarized as follows.

Applications. DVRPRRs arise from various real-world applications. Courier and technician services, freight and passenger transportation, e-commerce, and meal delivery account for the majority of the logistical contexts of the reviewed papers. Nearly all e-commerce-related problems are considered as SDDPs, while urban mobility and meal delivery problems are only studied as DPDPs. Since 2018, there has been increasing research interest in the problems related to e-commerce and meal delivery, many of which are associated with new technologies and/or business models such as drone delivery/resupply and crowdsourcing.

Decision Strategies. The implemented decision strategies depend on the type of dynamic requests.

(1) En-route diversions of vehicles can be enabled in VRPDSRs and DPDPs but do not apply to SDDPs. Allowing vehicles to be diverted from their next destinations leads to higher flexibility for decision-making and may shorten travel distance and service delay, but too frequent diversions may distract drivers and increase their workload. Therefore, when implementing a diversion strategy, it is recommended to limit the frequency of diversions (e.g., by a penalty term in the objective function) to balance the positive and negative effects.

(2) Preemptive depot returns can be implemented in SDDPs to improve solution quality but are meaningless for VRPDSRs and DPDPs. Enabling preemptive depot returns significantly increases the problem complexity, especially the dimension of the decision space, and the existing research in this respect is limited to single-vehicle settings.

(3) Whether dynamic requests can be rejected mainly depends on the logistical context rather than the request type. Nearly 36% of the reviewed papers explicitly consider request rejections, but only 55% of them accept or reject new requests immediately upon request arrival. Making immediate decisions favors customers' interests, but imposes tight time frames on decision-making and requires highly efficient online decision policies.

(4) Waiting strategies are compatible with all DVRPRR variants. But we note that, in VRPDSRs and DPDPs, vehicles can wait at many different locations. In contrast, in SDDPs, the only potentially beneficial waiting location is the depot. A waiting strategy can be integrated with a repositioning policy to strategically direct vehicles to areas where many new requests are expected to appear in the future. Implementing a waiting strategy requires the decision maker to determine when, where, and for how long each vehicle should wait, based on the spatio-temporal distribution of dynamic requests. Due to the large decision space, the existing waiting strategies are mostly based on some intuitively appealing decision rules or heuristics.

(5) The other decision strategies, such as dynamic pricing, drone delivery, and crowdsourcing, are relatively new to the DVRPRR and received limited attention in recent research.

Stochasticity. Overall, about 50% of the reviewed works consider the stochastic information of dynamic requests. There are two common ways of exploiting stochastic information: (1) generating sample requests or scenarios from Monte-Carlo simulations and incorporating them into route reoptimizations; (2) computing a value function to anticipate the expected reward-/cost-to-go from a given problem state or computing a policy function to approximate the expected rewards/costs of making decisions following a given policy. In recent years, there is an increasing research interest in novel anticipation methodologies based on neural networks, deep learning, robust optimization, etc. In most papers adopting DS formulations, the benefits of considering stochastic information are validated through comparisons against myopic policies. However, stochastic models are generally more difficult to solve than deterministic ones, which may limit the utilization of stochastic information in large-scale problems.

Test Instances. There are still no widely accepted test benchmarks for any of the four DVRPRR variants. In the reviewed literature, 39% of test instances are based on realistic problems (e.g., the New York City taxicab data), around 32% are arbitrarily generated, and the rest are mostly derived from existing SVRP benchmarks (e.g., Solomon's instances). Single-vehicle settings and synthetic graphs (e.g., Euclidean planes) are adopted in 20% and 69% of test instances, respectively. Only 19% of test instances are publicly available.

Solution Approaches. Most mathematical models used to formulate DVRPRRs can be classified into two categories: reoptimization of SVRPs (62%) and MDP (25%). The selection between the two types of models seems to be independent of the request type. Other mathematical models, such as MAS, HAPC, and queuing models, are adopted in some research on DPDPs. Nearly all models are solved by heuristics (e.g., insertion heuristics, CGBH), metaheuristics (e.g., local search, tabu search, and VNS), and ADP (e.g., VFA, PFA, and rollout) because of the high complexity and tight time frames of DVRPRRs. Applications of exact solution approaches (e.g., dynamic programming, branch-and-cut, and branch-and-price) are limited to a few unrealistically small instances. Due to the problem settings diversity and the lack of canonical test benchmarks, the specific conditions under which a certain model or algorithm outperforms the others are unclear.

9.2. Future research opportunities

Based on a detailed review of the existing literature on the DVRPRR, the previous section summarizes the relevant research and shows that some issues related to the DVRPRR are still not sufficiently addressed.

Moreover, the advances in information and communication technologies, the emergence of autonomous and electric vehicles (EVs), as well as the development of e-commerce and sharing economy will bring new features to the DVRPRR and lead to increased problem scales and complexity. In this section, we discuss the remaining challenges and research trends regarding the DVRPRR and derive recommendations for future research.

9.2.1. Applications

New vehicle technologies. Drones (UAVs) and unmanned vehicles will be more extensively used in the future for their environmental friendliness and high levels of autonomy and flexibility, but these vehicles have limited payload and battery capacity and their scheduling and routing strategies are significantly different from those of the conventional vehicles. To better exploit the advantages of the new types of vehicles, future research could focus on the dynamic drone/unmanned vehicle delivery problems with consideration of order consolidations, battery recharging/swapping operations, decentralized optimization, and/or joint use of unmanned and conventional vehicles.

Sustainability. The sustainability of logistics systems has become a critical issue as governments and customers are more and more concerned by the environmental impacts of logistics operations. However, few studies have considered energy consumption or carbon emission in the DVRPRR. Moreover, EVs are gaining more market share because of their advantages of no direct emission and high energy efficiency, but the utilization of EVs in logistics has mostly been studied with static settings (Asghari and Mirzapour Al-e hashem, 2020, 2021). Hence, dynamic green VRPs focusing on minimizing environmental impacts and dynamic EV routing problems with vehicle charging policies deserve to be studied in the future.

Sharing economy. The rise of sharing economy brings novel business models, such as crowdsourcing and ride-sharing, to city logistics and urban mobility. These models reduce operational costs by outsourcing workload partially or totally to the crowd or by allowing customers to share vehicles and travel expenses. In the DVRPRRs with crowdsourcing, the uncertain availability of crowdsourced drivers/riders increases problems' dynamism and stochasticity (e.g., Arslan et al., 2019). In ride-sharing systems, the service provider needs to trade off the reduction in total travel costs and the increase in customers' ride times and inconvenience caused by ride-sharing (e.g., Ma et al., 2015). The DVRPRRs based on these new business models have received limited attention and are promising directions for future research.

9.2.2. Decision strategies

Improvement of responsiveness. As mentioned earlier, many DVRPRRs are subject to strict time or vehicle constraints so the service provider has to reject some dynamic requests. In these problems, customers expect to know the service provider's decisions as early as possible, in order to have enough time to find an alternative service option in case their requests are rejected. However, making immediate acceptance/rejection decisions is a challenging task because the available online computation time is highly limited (within one or a few minutes). This review reveals that, in nearly 50% of the DVRPRRs with rejections, responses to customers are mostly delayed because decisions are made only at equidistant time points (e.g., every hour) or when a vehicle stops somewhere. Moreover, many existing approaches with immediate decisions are developed for the problems without stochastic information or order consolidation, or are only tested on small- and medium-scale artificial instances with no more than 500 requests. Hence, more computationally efficient solution approaches that can cope with large-scale stochastic instances and can better balance solution quality and efficiency need to be developed.

Dynamic pricing. In addition to the prompt responses to customers, another strategy that can further improve customer satisfaction is dynamic pricing. Under this strategy, each customer is offered multiple service options with different prices and/or deadlines. The service provider does not explicitly reject any request, but prices the service options differently based on service capability and potential profit, so that customers are encouraged to choose the most profitable option, or give up all options if serving the request is too costly or infeasible. Obviously, the available service options should be provided to each customer shortly after the customer places a request. Due to the tight time frames and large decision space, dynamic pricing is considered in only six DVRPRRs and needs to be further studied in the future.

Flexible routing options. Transshipment and preemptive depot return are two routing strategies that enable a higher routing flexibility and have only been studied in a DPDP (Bock, 2010) and an SDDP (Ulmer et al., 2019b), respectively. They can better cope with the dynamic events or allow more requests to be served on the same day, but in the meantime, they lead to a much larger decision space and higher computational complexity. A future research direction could be to study how to implement these flexible strategies efficiently and in which cases these strategies achieve the most significant performance improvement.

9.2.3. Solution approaches

Various mathematical formulations and algorithms have been developed to model and solve the four DVRPRR variants. Although none of them is widely accepted to be a standard or dominant approach, we recommend several representative solution frameworks that future research could refer to. The effectiveness of these frameworks has been sufficiently validated for a wide variety of dynamic problems, and most solution approaches for the four DVRPRR variants are based on or similar to at least one of these frameworks.

(1) The real-time control system proposed by Bock (2010) is a typical rolling-horizon reoptimization framework. As discussed in Section 5.5, this framework discretizes the service period into a sequence of anticipation horizons and solves a series of SVRPs which are formulated at the beginning of each horizon based on the buffered disturbances (e.g., request arrivals). When applying this framework to a DVRPRR, the length of the anticipation horizon should be set appropriately according to the disturbance rate, the performance of the reoptimization algorithms, and whether immediate responses need to be provided to new customers. Moreover, stochastic information can be exploited in this framework by integrating sample requests into the SVRPs (e.g., Ferrucci and Bock, 2016).

(2) The multiple scenario approach (MSA) proposed by Bent and Van Hentenryck (2004) is another reoptimization framework with rolling horizons (see Section 4.4). There are two major differences between the MSA and the framework of Bock (2010). First, the MSA maintains and reoptimizes multiple route plans at any moment, with each plan based on an independently generated scenario of sample requests. Second, when formulating the SVRPs to be reoptimized, the MSA does not partially freeze the ongoing routes within an anticipation horizon, thus allowing higher decision flexibility. The multi-scenario structure of MSA improves the robustness of the solutions against the stochasticity and dynamism in the environment, and requires a proper consensus function to derive a high-quality and executable route plan from the solution pool. If no stochastic information is employed, the MSA reduces to the multiple plan approach (MPA).

(3) The route-based MDP constructed by Ulmer et al. (2020b) is a generalized modeling framework for DS VRPs. In contrast to a conventional MDP which only captures vehicles' current locations and next moves, Ulmer et al. (2020b) propose to incorporate vehicles' planned routes into the state and decision variables of the MDP. The incorporation of complete routing information leads to larger state and decision spaces, but facilitates the application of the state-of-the-art route-based methodologies from the rich body of VRP literature. The route-based MDP has been applied to all four DVRPRR variants, and can easily be extended to address special problem characteristics, such as dynamic pricing and drone delivery.

(4) The unified framework formalized by Powell (2019) for stochastic sequential decision problems is a more generalized formulation that is applicable to the DVRP. Powell (2019) provides a universal canonical sequential optimization model, summarizes the methods of modeling stochasticity, and classifies the policies for state-dependent and state-independent problems. For more detailed elaboration on this modeling framework and its application in transportation and logistics, we also refer the interested reader to Powell (2022a,b).

Despite the large body of research that has been carried out to solve the DVRPRR, more work needs to be done to further evaluate and compare the existing solution approaches, improve the utilization of stochastic information, and achieve a better trade-off between solution quality and efficiency. More specifically, we suggest the following future research directions:

Better exploitation of stochastic information. Although many studies have validated the benefits of stochastic information in DVRPRRs, most DS approaches assume that the arrival times and locations of dynamic requests follow an artificial spatio-temporal distribution. Research on deriving anticipatory information from realistic historical data is relatively limited. To better capture the dynamism of real-world problems and the spatio-temporal heterogeneity of dynamic requests, more advanced anticipation approaches could be studied, such as the clustering algorithm of Ferrucci and Bock (2016), the neural networks and deep learning methods of Bono et al. (2021) and Chen et al. (2022), the recent data mining and machine learning techniques, etc. Besides, robust optimization, which is only studied by Subramanyam et al. (2021), provides a promising avenue for anticipating without relying on probability distributions.

MDP vs reoptimization. MDP and sequential reoptimization of SVRPs represent the mainstream of the solution frameworks for DVRPRRs. However, their advantages and disadvantages remain unclear. Only Ghiani et al. (2012) and Ulmer (2019) compared the two frameworks for single-vehicle VRPDSRs without time windows or capacity constraints. More comparative studies could be performed to examine the two frameworks under different problem settings. It is also worth comparing the quality of anticipations based on value/policy functions (mostly adopted in MDPs) and sample requests/scenarios (mostly adopted in reoptimization frameworks).

Multi-objective optimization. Among the DVRPRRs reviewed in this paper, 54% have at least two objectives to optimize. Almost all these dynamic bi-/multi-objective optimization problems are solved by weighted-sum methods (combining objectives into a scalar function) or lexicographic methods (ordering objectives according to their importance). Only three papers evaluate the Pareto optimum of the solutions produced by weighted-sum methods (Hyytiä et al., 2012; Klapp et al., 2018a, 2020), and Zhu et al. (2016) is the only paper that aims at finding the Pareto optimal set. Future research could be devoted to developing efficient algorithms for approximating the Pareto optimal sets for the bi-/multi-objective DVRPRRs, and the methods of selecting appropriate solutions for execution from the Pareto optimal sets are also worth investigating.

9.2.4. Test instances

As there is no canonical DVRP benchmark and more than 80% of DVRPRR instances are not publicly available, a large number of test instances derived from various data sources are used for computational studies in the literature, making it difficult to directly compare the performance of different decision strategies and solution approaches. Therefore, there is still a need for open benchmark instances for the four DVRPRR variants and we recommend researchers to publish their test instances to allow comparability and reproducibility of their results.

9.3. Limitations of this review

Due to the vast number of existing DVRP research papers, we limited our scope to the DVRPs with the most common source of dynamism: the dynamic customer requests. The DVRPs with other dynamic aspects, such as time-dependent travel times and uncertain customer demands, are also critical and challenging optimization problems faced by logistics service providers, but are not reviewed in depth in this paper. A broader review of the past and recent research on all DVRP variants could be performed in the future, and some recent conference papers and preprints (e.g., Okulewicz and Mańdziuk, 2020) that we excluded from this review may also be worth investigating. Statistical analysis of the literature on the DVRPRR could also be done to further validate the insights obtained from this review.

Finally, several research topics that are similar to the DVRPRR or incorporate the DVRPRR as a subproblem are not covered in this review. For example, in attended home delivery services, planned routes are dynamically updated to accommodate new requests, but no more requests can be placed after vehicles start their routes (e.g., Ulmer and Thomas, 2020; Vinsensius et al., 2020). In shared mobility systems, service providers should dynamically reposition bikes or cars to guarantee their availability to future customers (e.g., Nourinejad and Roorda, 2014; Ghosh et al., 2017). In delivery workforce scheduling and delivery service sizing problems, the primary decisions are to optimize the shift schedule of drivers or the size of the service area, while the secondary decisions are dynamic vehicle routing (e.g., Ulmer and Savelsbergh, 2020; Ulmer et al., 2022). Future literature reviews could investigate the links between the DVRPRR and these related problems.

Data availability

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

Appendix A. Abbreviations and acronyms

Abbreviations and acronyms	
ADP	Approximate dynamic programming
BRH	Branch-and-regret heuristic
CCP	Chance-constrained programming
CFA	Cost function approximation
CGBH	Column-generation-based heuristic
DARP	Dial-a-ride problem
DD	Dynamic and deterministic
DMPVRP	Dynamic multi-period vehicle routing problem
DNN	Dynamic nearest neighbor
DoD	Degree of dynamism
DPDP	Dynamic pickup and delivery problem
DS	Dynamic and stochastic
DSHH	Dynamic stochastic hedging heuristic
DTRP	Dynamic traveling repairman problem
DVRP	Dynamic vehicle routing problem
DVRPRD	Dynamic vehicle routing problem with random demands
DVRPRR	Dynamic vehicle routing problem with random requests
DVRPRTT	Dynamic vehicle routing problem with random travel times
EV	Electric vehicle
FTL	Full-truckload
HAPC	Hybrid adaptive predictive control
LTL	Lighter-than-truckload

(*continued on next page*)

MAS	Multi-agent system
MDP	Markov decision process
MIP	Mixed-integer program
MPA	Multiple plan approach
MSA	Multiple scenario approach
NN	Nearest neighbor
NRR	N-request reoptimization
P&D	Pickup and delivery
PFA	Policy function approximation
SD	Static and deterministic
SDDP	Same-day delivery problem
SS	Static and stochastic
SSP	Sample-scenario planning
SVRP	Static vehicle routing problem
TSP	Traveling salesman problem
TSPPD	Traveling salesman problem with pickup and delivery
UAV	Unmanned aerial vehicle
VFA	Value function approximation
mVFA	Meso-parametric value function approximation
nVFA	Non-parametric value function approximation
pVFA	Parametric value function approximation
VNS	Variable neighborhood search
VRP	Vehicle routing problem
VRPDSR	Vehicle routing problem with dynamic service requests
WDP	Waiting driver problem

Appendix B. Research protocol

See Table B.1.

Table B.1
Research protocol.

Background	
• Research topic	DVRPRR: dynamic vehicle routing problem with random requests
• Need for a review	(1) Many real-world transportation and logistics problems can be regarded as DVRPRRs
	(2) Technological advances facilitate the dynamic and real-time routing of vehicles
	(3) There is a vast number of research papers on DVRPRRs with various problem settings, decision strategies, mathematical models, and solution techniques
• Research questions	(1) How to classify DVRPRRs and what are the key classification criteria?
	(2) What problem settings and decision strategies are suitable to each DVRPRR variant?
	(3) What are the state-of-the-art solution approaches for DVRPRRs and what instances can they solve?
	(4) How stochastic information can be exploited in DVRPRRs and what are the benefits?
• Objectives	(1) Provide a comprehensive and up-to-date literature review on the DVRPRR
	(2) Summarize the achievements and limitations of the existing research and suggest promising future research opportunities
Search method	
• Step 1: Search from scientific database	
– Main database	Scopus
– Search fields	Article title; Abstract; Keywords
– Search strings	(1) VRPDSR: "dynamic vehicle routing" OR ("vehicle routing" AND "dynamic requests")
	(2) DPDP: "dynamic pickup and delivery" OR "dynamic dial-a-ride" OR ("dynamic vehicle routing" AND ("pickup and delivery" OR "dial-a-ride"))
	(3) SDDP: "same-day delivery" OR ("dynamic vehicle routing" AND "delivery")
	(4) DMPVRP: "dynamic vehicle routing" AND ("multiple periods" OR "multi-period")
• Step 2: Snowballing	(1) Backward search from the reference lists of the papers found in the first step
	(2) Forward search from the papers citing at least one paper found in the first step
Selection criteria	
• Time span	From 1980 to 2022
• Language	Limited to English
• Inclusion criteria	Research papers focusing on at least one of the four DVRPRR variants from peer-reviewed academic journals
• Exclusion criteria	(1) Conference papers, technical reports, book chapters, preprints
	(2) Papers focusing on the DVRP variants other than the DVRPRR (e.g., DVRPRTT and DVRPRD)
	(3) Papers focusing on (meta-)heuristics rather than DVRPRRs (mostly from computer science journals)
	(4) Papers focusing on VRPs that are not truly dynamic (e.g., attended home delivery problems)
Papers reviewed	Total: 118
	(1) VRPDSR: 35
	(2) DPDP: 54
	(3) SDDP: 13
	(4) DMPVRP: 16

Appendix C. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ijpe.2022.108751.

References

Achamrah, F.E., Riane, F., Limbourg, S., 2021. Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. Int. J. Prod. Res. Article in Press.

Albareda-Sambola, M., Fernández, E., Laporte, G., 2014. The dynamic multiperiod vehicle routing problem with probabilistic information. Comput. Oper. Res. 48, 31–39.

Alnaggar, A., Gzara, F., Bookbinder, J.H., 2021. Crowdsourced delivery: A review of platforms and academic literature. Omega 98, 102139.

Andreatta, G., Lulli, G., 2008. A multi-period TSP with stochastic regular and urgent demands. European J. Oper. Res. 185 (1), 122–132.

Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M.G., 2009. Short term strategies for a dynamic multi-period routing problem. Transp. Res. C 17 (2), 106–119.

Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M.G., 2010. Comparison of policies in dynamic routing problems. J. Oper. Res. Soc. 61 (4), 686–695.

Angelelli, E., Grazia Speranza, M., Savelsbergh, M.W., 2007a. Competitive analysis for dynamic multiperiod uncapacitated routing problems. Networks 49 (4), 308–317.

Angelelli, E., Savelsbergh, M.W., Speranza, M.G., 2007b. Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. Oper. Res. Lett. 35 (6), 713–721.

Archetti, C., Savelsbergh, M., Speranza, M.G., 2016. The vehicle routing problem with occasional drivers. European J. Oper. Res. 254 (2), 472–480.

Arslan, A.M., Agatz, N., Kroon, L., Zuidwijk, R., 2019. Crowdsourced delivery—A dynamic pickup and delivery problem with ad hoc drivers. Transp. Sci. 53 (1), 222–235.

Asghari, M., Mirzapour Al-e hashem, S.M.J., 2020. New advances in vehicle routing problems: A literature review to explore the future. In: Derbel, H., Jarboui, B., Siarry, P. (Eds.), Green Transportation and New Advances in Vehicle Routing Problems. Springer, Cham, pp. 1–42.

Asghari, M., Mirzapour Al-e hashem, S.M.J., 2021. Green vehicle routing problem: A state-of-the-art review. Int. J. Prod. Econ. 231, 107899.
Azi, N., Gendreau, M., Potvin, J.-Y., 2012. A dynamic vehicle routing problem with multiple delivery routes. Ann. Oper. Res. 199 (1), 103–112.
Beasley, J.E., 1990. OR-Library: distributing test problems by electronic mail. J. Oper. Res. Soc. 41 (11), 1069–1072.
Beaudry, A., Laporte, G., Melo, T., Nickel, S., 2010. Dynamic transportation of patients in hospitals. OR Spectrum 32 (1), 77–107.
Bektaş, T., Repoussis, P.P., Tarantilis, C.D., 2014. Chapter 11: Dynamic vehicle routing problems. In: Vehicle Routing: Problems, Methods, and Applications, second ed. SIAM, pp. 299–347.
Bent, R.W., Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Oper. Res. 52 (6), 977–987.
Berbeglia, G., Cordeau, J.-F., Laporte, G., 2010. Dynamic pickup and delivery problems. European J. Oper. Res. 202 (1), 8–15.
Berbeglia, G., Cordeau, J.-F., Laporte, G., 2012. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. INFORMS J. Comput. 24 (3), 343–355.
Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neuro-Dynamic Programming. Athena Scientific, Belmont, MA.
Bertsimas, D., Jaillet, P., Martin, S., 2019. Online vehicle routing: The edge of optimization in large-scale applications. Oper. Res. 67 (1), 143–162.
Bertsimas, D.J., van Ryzin, G., 1991. A stochastic and dynamic vehicle routing problem in the Euclidean plane. Oper. Res. 39 (4), 601–615.
Bertsimas, D.J., van Ryzin, G., 1993. Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. Oper. Res. 41 (1), 60–76.
Bock, S., 2010. Real-time control of freight forwarder transportation networks by integrating multimodal transport chains. European J. Oper. Res. 200 (3), 733–746.
Bono, G., Dibangoye, J.S., Simonin, O., Matignon, L., Pereyron, F., 2021. Solving multi-agent routing problems using deep attention mechanisms. IEEE Trans. Intell. Transp. Syst. 22 (12), 7804–7813.
Bosco, A., Laganà, D., Musmanno, R., Vocaturo, F., 2013. Modeling and solving the mixed capacitated general routing problem. Optim. Lett. 7 (7), 1451–1469.
Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I., 2016. The vehicle routing problem: State of the art classification and review. Comput. Ind. Eng. 99, 300–313.
Branchini, R.M., Armentano, V.A., Løkketangen, A., 2009. Adaptive granular local search heuristic for a dynamic vehicle routing problem. Comput. Oper. Res. 36 (11), 2955–2968.
Branke, J., Middendorf, M., Noeth, G., Dessouky, M., 2005. Waiting strategies for dynamic vehicle routing. Transp. Sci. 39 (3), 298–312.
Caramia, M., Italiano, G.F., Oriolo, G., Pacifici, A., Perugia, A., 2002. Routing a fleet of vehicles for dynamic combined pick-up and deliveries services. In: Operations Research Proceedings 2001. Springer, pp. 3–8.
Chen, X., Thomas, B.W., Hewitt, M., 2016. The technician routing problem with experience-based service times. Omega 61, 49–61.
Chen, X., Ulmer, M.W., Thomas, B.W., 2022. Deep Q-learning for same-day delivery with vehicles and drones. European J. Oper. Res. 298 (3), 939–952.
Chen, Z.-L., Xu, H., 2006. Dynamic column generation for dynamic vehicle routing with time windows. Transp. Sci. 40 (1), 74–88.
Cheung, B.K.-S., Choy, K., Li, C.-L., Shi, W., Tang, J., 2008. Dynamic routing model and solution methods for fleet management with mobile technologies. Int. J. Prod. Econ. 113 (2), 694–705.
Cordeau, J.-F., Dell'Amico, M., Falavigna, S., Iori, M., 2015. A rolling horizon algorithm for auto-carrier transportation. Transp. Res. B 76, 68–80.
Cortés, C.E., Sáez, D., Núñez, A., Muñoz-Carpintero, D., 2009. Hybrid adaptive predictive control for a dynamic pickup and delivery problem. Transp. Sci. 43 (1), 27–42.
Coslovich, L., Pesenti, R., Ukovich, W., 2006. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European J. Oper. Res. 175 (3), 1605–1615.
Dayarian, I., Savelsbergh, M., 2020. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. Prod. Oper. Manage. 29 (9), 2153–2174.
Dayarian, I., Savelsbergh, M., Clarke, J.-P., 2020. Same-day delivery with drone resupply. Transp. Sci. 54 (1), 229–249.
Duan, L., Wei, Y., Zhang, J., Xia, Y., 2020. Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode. Transp. Res. C 111, 397–420.
Dumas, Y., Desrosiers, J., Gelinas, E., Solomon, M.M., 1995. An optimal algorithm for the traveling salesman problem with time windows. Oper. Res. 43 (2), 367–371.
Euchi, J., Yassine, A., Chabchoub, H., 2015. The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. Swarm Evol. Comput. 21, 41–53.
Fabri, A., Recht, P., 2006. On dynamic pickup and delivery vehicle routing with several time windows and waiting times. Transp. Res. B 40 (4), 335–350.
Ferrucci, F., Bock, S., 2014. Real-time control of express pickup and delivery processes in a dynamic environment. Transp. Res. B 63, 1–14.
Ferrucci, F., Bock, S., 2015. A general approach for controlling vehicle en-route diversions in dynamic vehicle routing problems. Transp. Res. B 77, 76–87.
Ferrucci, F., Bock, S., 2016. Pro-active real-time routing in applications with multiple request patterns. European J. Oper. Res. 253 (2), 356–371.
Ferrucci, F., Bock, S., Gendreau, M., 2013. A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. European J. Oper. Res. 225 (1), 130–141.
Florio, A.M., Hartl, R.F., Minner, S., 2020a. New exact algorithm for the vehicle routing problem with stochastic demands. Transp. Sci. 54 (4), 1073–1090.
Florio, A.M., Hartl, R.F., Minner, S., 2020b. Optimal a priori tour and restocking policy for the single-vehicle routing problem with stochastic demands. European J. Oper. Res. 285 (1), 172–182.
Garrido, P., Riff, M.C., 2010. DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. J. Heuristics 16 (6), 795–834.
Gendreau, M., Guertin, F., Potvin, J.-Y., Séguin, R., 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transp. Res. C 14 (3), 157–174.
Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, E., 1999. Parallel tabu search for real-time vehicle routing and dispatching. Transp. Sci. 33 (4), 381–390.
Gendreau, M., Jabali, O., Rei, W., 2016. 50Th anniversary invited article—future research directions in stochastic vehicle routing. Transp. Sci. 50 (4), 1163–1173.
Gendreau, M., Potvin, J.-Y., 1998. Dynamic vehicle routing and dispatching. In: Fleet Management and Logistics. Springer US, pp. 115–126.
Ghiani, G., Manni, A., Manni, E., 2022. A scalable anticipatory policy for the dynamic pickup and delivery problem. Comput. Oper. Res. 147, 105943.
Ghiani, G., Manni, E., Quaranta, A., Triki, C., 2009. Anticipatory algorithms for same-day courier dispatching. Transp. Res. E 45 (1), 96–106.
Ghiani, G., Manni, E., Thomas, B.W., 2012. A comparison of anticipatory algorithms for the dynamic and stochastic traveling salesman problem. Transp. Sci. 46 (3), 374–387.
Ghosh, S., Varakantham, P., Adulyasak, Y., Jaillet, P., 2017. Dynamic repositioning to reduce lost demand in bike sharing systems. J. Artificial Intelligence Res. 58, 387–430.
Goel, A., Gruhn, V., 2008. A general vehicle routing problem. European J. Oper. Res. 191 (3), 650–660.
Gómez, A., Mariño, R., Akhavan-Tabatabaei, R., Medaglia, A.L., Mendoza, J.E., 2016. On modeling stochastic travel and service times in vehicle routing. Transp. Sci. 50 (2), 627–641.
Goodson, J.C., Ohlmann, J.W., Thomas, B.W., 2013. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. Oper. Res. 61 (1), 138–154.
Goodson, J.C., Thomas, B.W., Ohlmann, J.W., 2016. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. Transp. Sci. 50 (2), 591–607.
Goodson, J.C., Thomas, B.W., Ohlmann, J.W., 2017. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. European J. Oper. Res. 258 (1), 216–229.
Güner, A.R., Murat, A., Chinnam, R.B., 2017. Dynamic routing for milk-run tours with time windows in stochastic time-dependent networks. Transp. Res. E 97, 251–267.
Györgyi, P., Kis, T., 2019. A probabilistic approach to pickup and delivery problems with time window uncertainty. European J. Oper. Res. 274 (3), 909–923.
Haferkamp, J., Ehmke, J.F., 2022. Effectiveness of demand and fulfillment control in dynamic fleet management of ride-sharing systems. Networks 79 (3), 314–337.
He, Z., Han, G., Cheng, T., Fan, B., Dong, J., 2019. Evolutionary food quality and location strategies for restaurants in competitive online-to-offline food ordering and delivery markets: An agent-based approach. Int. J. Prod. Econ. 215, 61–72.
van Heeswijk, W., Mes, M., Schutten, J., 2019. The delivery dispatching problem with time windows for urban consolidation centers. Transp. Sci. 53 (1), 203–221.
Hong, L., 2012. An improved LNS algorithm for real-time vehicle routing problem with time windows. Comput. Oper. Res. 39 (2), 151–163.
Hvattum, L.M., Løkketangen, A., Laporte, G., 2006. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transp. Sci. 40 (4), 421–438.
Hvattum, L.M., Løkketangen, A., Laporte, G., 2007. A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. Networks 49 (4), 330–340.
Hyland, M., Mahmassani, H.S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transp. Res. C 92, 278–297.
Hyytiä, E., Penttinen, A., Sulonen, R., 2012. Non-myopic vehicle and route selection in dynamic DARP with travel time and workload objectives. Comput. Oper. Res. 39 (12), 3021–3030.
Ichoua, S., Gendreau, M., Potvin, J.-Y., 2000. Diversion issues in real-time vehicle dispatching. Transp. Sci. 34 (4), 426–438.
Ichoua, S., Gendreau, M., Potvin, J.-Y., 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. Transp. Sci. 40 (2), 211–225.
Karami, F., Vancroonenburg, W., Berghe, G.V., 2020. A periodic optimization approach to dynamic pickup and delivery problems with time windows. J. Sched. 23 (6), 711–731.
Kergosien, Y., Lente, C., Piton, D., Billaut, J.-C., 2011. A tabu search heuristic for the dynamic transportation of patients between care units. European J. Oper. Res. 214 (2), 442–452.
Kilby, P., Prosser, P., Shaw, P., 1998. Dynamic VRPs: A Study of Scenarios. Technical Report, University of Strathclyde.

Kim, G., Ong, Y.S., Cheong, T., Tan, P.S., 2016. Solving the dynamic vehicle routing problem under traffic congestion. IEEE Trans. Intell. Transp. Syst. 17 (8), 2367–2380.

Klapp, M.A., Erera, A.L., Toriello, A., 2018a. The dynamic dispatch waves problem for same-day delivery. European J. Oper. Res. 271 (2), 519–534.

Klapp, M.A., Erera, A.L., Toriello, A., 2018b. The one-dimensional dynamic dispatch waves problem. Transp. Sci. 52 (2), 402–415.

Klapp, M.A., Erera, A.L., Toriello, A., 2020. Request acceptance in same-day delivery. Transp. Res. E 143, 102083.

Köster, F., Ulmer, M.W., Mattfeld, D.C., Hasle, G., 2018. Anticipating emission-sensitive traffic management strategies for dynamic delivery routing. Transp. Res. D 62, 345–361.

Kullman, N.D., Cousineau, M., Goodson, J.C., Mendoza, J.E., 2022. Dynamic ride-hailing with electric vehicles. Transp. Sci. 56 (3), 775–794.

Lackner, A., 2004. Dynamische Tourenplanung mit Ausgewählten Metaheuristiken: Eine Untersuchung am Beispiel des Kapazitätsrestriktiven Dynamischen Tourenplanungsproblems mit Zeitfenstern, Vol. 47. Cuvillier Verlag.

Laganà, D., Laporte, G., Vocaturo, F., 2021. A dynamic multi-period general routing problem arising in postal service and parcel delivery systems. Comput. Oper. Res. 129, 105195.

Larsen, A., Madsen, O., Solomon, M., 2002. Partially dynamic vehicle routing—models and algorithms. J. Oper. Res. Soc. 53 (6), 637–646.

Larsen, A., Madsen, O.B., Solomon, M.M., 2004. The a priori dynamic traveling salesman problem with time windows. Transp. Sci. 38 (4), 459–472.

Li, H., Lim, A., 2001. A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001. pp. 160–167.

Li, J.-Q., Mirchandani, P.B., Borenstein, D., 2009. Real-time vehicle rerouting problems with time windows. European J. Oper. Res. 194 (3), 711–727.

Liang, X., de Almeida Correia, G.H., An, K., van Arem, B., 2020. Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times. Transp. Res. C 112, 260–281.

Lin, C., Choy, K.L., Ho, G.T., Lam, H., Pang, G.K., Chin, K.-S., 2014. A decision support system for optimizing dynamic courier routing operations. Expert Syst. Appl. 41 (15), 6917–6933.

Liu, Y., 2019. An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. Comput. Oper. Res. 111, 1–20.

Liu, Z., Li, X., Khojandi, A., 2022. The flying sidekick traveling salesman problem with stochastic travel time: A reinforcement learning approach. Transp. Res. E 164, 102816.

van Lon, R.R., Ferrante, E., Turgut, A.E., Wenseleers, T., Berghe, G.V., Holvoet, T., 2016. Measures of dynamism and urgency in logistics. European J. Oper. Res. 253 (3), 614–624.

Lorini, S., Potvin, J.-Y., Zufferey, N., 2011. Online vehicle routing and scheduling with dynamic travel times. Comput. Oper. Res. 38 (7), 1086–1090.

Los, J., Schulte, F., Spaan, M.T., Negenborn, R.R., 2020. The value of information sharing for platform-based collaborative vehicle routing. Transp. Res. E 141, 102011.

Lund, K., Madsen, O.B., Rygaard, J.M., 1996. Vehicle Routing Problems with Varying Degrees of Dynamism. IMM, Institute of Mathematical Modelling, Technical University of Denmark.

Lysgaard, J., Letchford, A.N., Eglese, R.W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Math. Program. 100 (2), 423–445.

Ma, S., Zheng, Y., Wolfson, O., 2015. Real-time city-scale taxi ridesharing. IEEE Trans. Knowl. Data Eng. 27 (7), 1782–1795.

Máhr, T., Srour, J., de Weerdt, M., Zuidwijk, R., 2010. Can agents measure up? A comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty. Transp. Res. C 18 (1), 99–119.

Mes, M., van der Heijden, M., Schuur, P., 2010. Look-ahead strategies for dynamic pickup and delivery problems. OR Spectrum 32 (2), 395–421.

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

Mitrović-Minić, S., Krishnamurti, R., Laporte, G., 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transp. Res. B 38 (8), 669–685.

Mitrović-Minić, S., Laporte, G., 2004. Waiting strategies for the dynamic pickup and delivery problem with time windows. Transp. Res. B 38 (7), 635–655.

Montemanni, R., Gambardella, L.M., Rizzoli, A.E., Donati, A.V., 2005. Ant colony system for a dynamic vehicle routing problem. J. Comb. Optim. 10 (4), 327–343.

Muñoz-Carpintero, D., Sáez, D., Cortés, C.E., Núñez, A., 2015. A methodology based on evolutionary algorithms to solve a dynamic pickup and delivery problem under a hybrid predictive control approach. Transp. Sci. 49 (2), 239–253.

Ninikas, G., Minis, I., 2014. Reoptimization strategies for a dynamic vehicle routing problem with mixed backhauls. Networks 64 (3), 214–231.

Nourinejad, M., Roorda, M.J., 2014. A dynamic carsharing decision support system. Transp. Res. E 66, 36–50.

Novoa, C., Storer, R., 2009. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. European J. Oper. Res. 196 (2), 509–515.

Okulewicz, M., Mańdziuk, J., 2020. Dynamic vehicle routing problem: A Monte Carlo approach. arXiv preprint, arXiv:2006.09996.

Pankratz, G., 2005. Dynamic vehicle routing by means of a genetic algorithm. Int. J. Phys. Distrib. Logist. Manage. 35 (5), 362–383.

Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. European J. Oper. Res. 225 (1), 1–11.

Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality, second ed. In: Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, New Jersey.

Powell, W.B., 2019. A unified framework for stochastic optimization. European J. Oper. Res. 275 (3), 795–821.

Powell, W.B., 2022a. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons, Ltd, Hoboken, New Jersey.

Powell, W.B., 2022b. Designing lookahead policies for sequential decision problems in transportation and logistics. IEEE Open J. Intell. Transp. Syst. 3, 313–327.

Powell, W.B., Simao, H.P., Bouzaiene-Ayari, B., 2012. Approximate dynamic programming in transportation and logistics: A unified framework. EURO J. Transp. Logist. 1 (3), 237–284.

Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. 14 (2), 130–154.

Psaraftis, H.N., 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transp. Sci. 17 (3), 351–357.

Psaraftis, H.N., Wen, M., Kontovas, C.A., 2016. Dynamic vehicle routing problems: Three decades and counting. Networks 67 (1), 3–31.

Pureza, V., Laporte, G., 2008. Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. INFOR: Inf. Syst. Oper. Res. 46 (3), 165–175.

Respen, J., Zufferey, N., Potvin, J.-Y., 2019. Impact of vehicle tracking on a routing problem with dynamic travel times. RAIRO-Oper. Res. 53 (2), 401–414.

Rifki, O., Chiabaut, N., Solnon, C., 2020. On the impact of spatio-temporal granularity of traffic conditions on the quality of pickup and delivery optimal tours. Transp. Res. E 142, 102085.

Rios, B.H.O., Xavier, E.C., Miyazawa, F.K., Amorim, P., Curcio, E., Santos, M.J., 2021. Recent dynamic vehicle routing problems: A survey. Comput. Ind. Eng. 160, 107604.

Ritzinger, U., Puchinger, J., Hartl, R.F., 2016. A survey on dynamic and stochastic vehicle routing problems. Int. J. Prod. Res. 54 (1), 215–231.

Ropke, S., Cordeau, J.-F., Laporte, G., 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks 49 (4), 258–272.

Sáez, D., Cortés, C.E., Núñez, A., 2008. 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), 3412–3438.

Sarasola, B., Doerner, K.F., Schmid, V., Alba, E., 2016. Variable neighborhood search for the stochastic and dynamic vehicle routing problem. Ann. Oper. Res. 236 (2), 425–461.

Savelsbergh, M., Sol, M., 1998. Drive: Dynamic routing of independent vehicles. Oper. Res. 46 (4), 474–490.

Sayarshad, H.R., Chow, J.Y., 2015. A scalable non-myopic dynamic dial-a-ride and pricing problem. Transp. Res. B 81, 539–554.

Sayarshad, H.R., Gao, H.O., 2018. A scalable non-myopic dynamic dial-a-ride and pricing problem for competitive on-demand mobility systems. Transp. Res. C 91, 192–208.

Schilde, M., Doerner, K.F., Hartl, R.F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput. Oper. Res. 38 (12), 1719–1730.

Schilde, M., Doerner, K.F., Hartl, R.F., 2014. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. European J. Oper. Res. 238 (1), 18–30.

Schyns, M., 2015. An ant colony system for responsive dynamic vehicle routing. European J. Oper. Res. 245 (3), 704–718.

Secomandi, N., Margot, F., 2009. Reoptimization approaches for the vehicle-routing problem with stochastic demands. Oper. Res. 57 (1), 214–230.

Shen, Y., Potvin, J.-Y., Rousseau, J.-M., Roy, S., 1995. A computer assistant for vehicle dispatching with learning capabilities. Ann. Oper. Res. 61 (1), 189–211.

Sheridan, P.K., Gluck, E., Guan, Q., Pickles, T., Balcıog, B., Benhabib, B., et al., 2013. The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. Transp. Res. A 49, 178–194.

Soeffker, N., Ulmer, M.W., Mattfeld, D.C., 2022. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. European J. Oper. Res. 298, 801–820.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.

Srour, F.J., Agatz, N., Oppen, J., 2018. Strategies for handling temporal uncertainty in pickup and delivery problems with time windows. Transp. Sci. 52 (1), 3–19.

Steever, Z., Karwan, M., Murray, C., 2019. Dynamic courier routing for a food delivery service. Comput. Oper. Res. 107, 173–188.

Subramanyam, A., Mufalli, F., Laínez-Aguirre, J.M., Pinto, J.M., Gounaris, C.E., 2021. Robust multiperiod vehicle routing under customer order uncertainty. Oper. Res. 69 (1), 30–60.

Swihart, M.R., Papastavrou, J.D., 1999. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. European J. Oper. Res. 114 (3), 447–464.

Tafreshian, A., Abdolmaleki, M., Masoud, N., Wang, H., 2021. Proactive shuttle dispatching in large-scale dynamic dial-a-ride systems. Transp. Res. B 150, 227–259.
Thomas, B.W., 2007. Waiting strategies for anticipating service requests from known customer locations. Transp. Sci. 41 (3), 319–331.
Thomas, B.W., White III, C.C., 2004. Anticipatory route selection. Transp. Sci. 38 (4), 473–487.
Tirado, G., Hvattum, L.M., 2017. Improved solutions to dynamic and stochastic maritime pick-up and delivery problems using local search. Ann. Oper. Res. 253 (2), 825–843.
Tirado, G., Hvattum, L.M., Fagerholt, K., Cordeau, J.-F., 2013. Heuristics for dynamic and stochastic routing in industrial shipping. Comput. Oper. Res. 40 (1), 253–263.
Toriello, A., Haskell, W.B., Poremba, M., 2014. A dynamic traveling salesman problem with stochastic arc costs. Oper. Res. 62 (5), 1107–1125.
Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications, second ed. MOS-SIAM Series on Optimization, Philadelphia.
Ulmer, M.W., 2019. Anticipation versus reactive reoptimization for dynamic vehicle routing with stochastic requests. Networks 73 (3), 277–291.
Ulmer, M.W., 2020a. Dynamic pricing and routing for same-day delivery. Transp. Sci. 54 (4), 1016–1033.
Ulmer, M.W., 2020b. Horizontal combinations of online and offline approximate dynamic programming for stochastic dynamic vehicle routing. CEJOR Cent. Eur. J. Oper. Res. 28 (1), 279–308.
Ulmer, M.W., Erera, A., Savelsbergh, M., 2022. Dynamic service area sizing in urban delivery. OR Spectrum 44, 763—-793.
Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Hennig, M., 2019a. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. Transp. Sci. 53 (1), 185–202.
Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Thomas, B.W., 2020b. On modeling stochastic dynamic vehicle routing problems. EURO J. Transp. Logist. 9 (2), 100008.
Ulmer, M.W., Mattfeld, D.C., Köster, F., 2018a. Budgeting time for dynamic vehicle routing with stochastic customer requests. Transp. Sci. 52 (1), 20–37.
Ulmer, M., Nowak, M., Mattfeld, D., Kaminski, B., 2020a. Binary driver-customer familiarity in service routing. European J. Oper. Res. 286 (2), 477–493.
Ulmer, M.W., Savelsbergh, M., 2020. Workforce scheduling in the era of crowdsourced delivery. Transp. Sci. 54 (4), 1113–1133.
Ulmer, M.W., Soeffker, N., Mattfeld, D.C., 2018b. Value function approximation for dynamic multi-period vehicle routing. European J. Oper. Res. 269 (3), 883–899.
Ulmer, M.W., Streng, S., 2019. Same-day delivery with pickup stations and autonomous vehicles. Comput. Oper. Res. 108, 1–19.
Ulmer, M.W., Thomas, B.W., 2018. Same-day delivery with heterogeneous fleets of drones and vehicles. Networks 72 (4), 475–505.
Ulmer, M.W., Thomas, B.W., 2020. Meso-parametric value function approximation for dynamic customer acceptances in delivery routing. European J. Oper. Res. 285 (1), 183–195.
Ulmer, M.W., Thomas, B.W., Campbell, A.M., Woyak, N., 2021. The restaurant meal delivery problem: dynamic pickup and delivery with deadlines and random ready times. Transp. Sci. 55 (1), 75–100.
Ulmer, M.W., Thomas, B.W., Mattfeld, D.C., 2019b. Preemptive depot returns for dynamic same-day delivery. EURO J. Transp. Logist. 8 (4), 327–361.
Vinsensius, A., Wang, Y., Chew, E.P., Lee, L.H., 2020. Dynamic incentive mechanism for delivery slot management in e-commerce attended home delivery. Transp. Sci. 54 (3), 567–587.
Voccia, S.A., Campbell, A.M., Thomas, B.W., 2019. The same-day delivery problem for online purchases. Transp. Sci. 53 (1), 167–184.
Vodopivec, N., Miller-Hooks, E., 2017. An optimal stopping approach to managing travel-time uncertainty for time-sensitive customer pickup. Transp. Res. B 102, 22–37.
Vonolfen, S., Affenzeller, M., 2016. Distribution of waiting time for dynamic pickup and delivery problems. Ann. Oper. Res. 236 (2), 359–382.
Wang, F., Liao, F., Li, Y., Yan, X., Chen, X., 2021. An ensemble learning based multi-objective evolutionary algorithm for the dynamic vehicle routing problem with time windows. Comput. Ind. Eng. 154, 107131.
Wen, M., Cordeau, J.-F., Laporte, G., Larsen, J., 2010. The dynamic multi-period vehicle routing problem. Comput. Oper. Res. 37 (9), 1615–1623.
Xiang, X., Tian, Y., Zhang, X., Xiao, J., Jin, Y., 2021. A pairwise proximity learning-based ant colony algorithm for dynamic vehicle routing problems. IEEE Trans. Intell. Transp. Syst..
Xu, H., Pu, P., Duan, F., 2018. Dynamic vehicle routing problems with enhanced ant colony optimization. Discrete Dyn. Nat. Soc. 2018, 1295485.
Yang, J., Jaillet, P., Mahmassani, H., 2004. Real-time multivehicle truckload pickup and delivery problems. Transp. Sci. 38 (2), 135–148.
Yu, G., Yang, Y., 2019. Dynamic routing with real-time traffic information. Oper. Res. 19 (4), 1033–1058.
Zhang, J., Luo, K., Florio, A.M., Van Woensel, T., 2022a. Solving large-scale dynamic vehicle routing problems with stochastic requests. European J. Oper. Res. Article in Press.
Zhang, G., Smilowitz, K., Erera, A., 2011. Dynamic planning for urban drayage operations. Transp. Res. E 47 (5), 764–777.
Zhang, X., Zhang, J., Fan, X., 2022b. Offline approximate value iteration for dynamic solutions to the multivehicle routing problem with stochastic demand. Comput. Oper. Res. 105884.
Zhong, H., Hall, R.W., Dessouky, M., 2007. Territory planning and vehicle dispatching with driver learning. Transp. Sci. 41 (1), 74–89.
Zhu, Z., Xiao, J., He, S., Ji, Z., Sun, Y., 2016. A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem. Inform. Sci. 329, 73–89.
Zolfagharinia, H., Haughton, M., 2014. The benefit of advance load information for truckload carriers. Transp. Res. E 70, 34–54.
Zolfagharinia, H., Haughton, M., 2016. Effective truckload dispatch decision methods with incomplete advance load information. European J. Oper. Res. 252 (1), 103–121.
Zolfagharinia, H., Haughton, M.A., 2017. Operational flexibility in the truckload trucking industry. Transp. Res. B 104, 437–460.
Zou, H., Dessouky, M.M., Hu, S., 2021. An online cost allocation model for horizontal supply chains. Transp. Res. C 122, 102888.